Explorations in Controlled Image Captioning

By

Omid Mohamad Nezami

A thesis submitted to Macquarie University
for the degree of
Doctor of Philosophy
Department of Computing
August 2020

MACQUARIE
University
SYDNEY · AUSTRALIA

Declaration

I certify that the work in this thesis entitled Explorations in Controlled Image Captioning has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree to any other university or institution other than Macquarie University. I also certify that the thesis is an original piece of research and it has been written by me. Any help and assistance that I have received in my research work and the preparation of the thesis itself have been appropriately acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

I note that this research has an ethics committee approval with the protocol number of 5201700785.

Omid Mohamad Nezami

Acknowledgements

First, I am extremely grateful to my principal supervisor, Prof. Mark Dras, for his immense knowledge and patient guidance. The completion of this thesis was possible with the unconditional support provided by him. He has been a tremendous mentor for me. His advice on both research and career options has been invaluable.

I would like to express my sincere gratitude to my associate supervisor, Dr. Len Hamey, for all the support and guidance he gave me during my PhD study and related research. I would also like to greatly thank my former supervisor, Prof. Deborah Richards, for her continuous support and encouragement.

I wish to gratefully thank my supervisors at Data61, Dr. Stephen Wan and Dr. Cecile Paris, for their insightful comments and encouragement. It has been a great honour to complete my PhD study under their supervision.

I would especially like to thank Dr. Peter Anderson and Akshay Chaturvedi. I consider it as a great opportunity to work with them during my PhD study.

In regards to visiting the ADAPT centre, Dublin, Ireland, I gratefully thank Dr. Teresa Lynn and her colleagues for insightful discussions to develop my PhD project.

My sincere thanks goes to the department administrators and the science IT staff for all their help and support during my PhD study.

I would like to acknowledge that this research has been funded by International Macquarie University Research Excellence Scholarship (iMQRES) and Data61 Top-Up Scholarship.

Last but not least, I dedicate this thesis to my parents, wife and siblings for their moral support and their encouragements during my PhD study.

List of Publications

The main papers and articles as the basis of Chapters 3, 4, 5, 6 and 7 in the thesis:

- Omid Mohamad Nezami, Mark Dras, Peter Anderson, Len Hamey (2018). Face-Cap: Image Captioning using Facial Expression Analysis. *Proceedings of the 2018 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2018),* Dublin, Ireland.
- Omid Mohamad Nezami, Mark Dras, Stephen Wan, Cecile Paris (2018). Senti-Attend: Image Captioning using Sentiment and Attention. *arXiv preprint arXiv:1811.09789.*
- Omid Mohamad Nezami, Mark Dras, Stephen Wan, Cecile Paris, Len Hamey (2019). Towards Generating Stylized Image Captions via Adversarial Training. *Proceedings of the 2019 Pacific Rim International Conference on Artificial Intelligence (PRICAI 2019),* Cuvu, Fiji.
- Omid Mohamad Nezami, Mark Dras, Len Hamey, Deborah Richards, Stephen Wan, Cecile Paris (2019). Automatic Recognition of Student Engagement using Deep Learning and Facial Expression. *Proceedings of the 2019 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2019),* Wuerzburg, Germany.

- Omid Mohamad Nezami, Akshay Chaturvedi, Mark Dras, Utpal Garain (2020). Pick-Object-Attack: Type-Specific Adversarial Attack for Object Detection. *arXiv preprint arXiv:2006.03184*. [1]

- Omid Mohamad Nezami, Mark Dras, Stephen Wan, Cecile Paris (2020). Image Captioning using Facial Expression and Attention. *Journal of Artificial Intelligence Research (JAIR),* vol. 68, pp. 661-689.

As the primary author of these papers and articles, I just wanted to say thank the co-authors for their invaluable contributions.

[1] The first two authors contributed equally to this work.

Abstract

Benefiting from advances in machine vision and natural language processing, current image captioning systems are able to generate natural-sounding descriptions of source images. Most systems deal only with factual descriptions, although there are extensions where the captions are ‘controlled’, in the sense that they are directed to incorporate particular additional information, such as selected stylistic properties. This thesis seeks the understand and improve these controlled image captioning models, applying extra visually-grounded and non-grounded information. First, we target the emotional content of images as extra visually-grounded information, which is an important facet of human generated captions, to generate more descriptive image captions. Second, we target stylistic patterns as non-grounded information, which is an important property of written communication. Finally, as a more general instance of perturbing the input, we examine how image captions are affected by the injection of perturbations in the source image, introduced by adversarial attacks that we propose on an object detector. Specifically, the major contributions of the thesis are described as follows:

- We propose several novel image captioning models to incorporate emotional features that learned from an external dataset. Before applying the features for image captioning, we show the transferability and the effectiveness of the features for another task: *automatic engagement recognition*. For this, we propose a novel model for engagement recognition, initialized with the features, using our newly collected dataset. In the image captioning models, we specifically use one-hot encoding and attention-based representations of facial expressions present in images as our emotional features. We find that injecting facial features as a fixed one-hot encoding can lead to improved captions, with the best results if the injection is at the initial time step of an encoder-decoder architecture with a specific loss function to remember the encoding. An attention-based distributed representation at each time step provides the best results.

- We present several novel image captioning models using attention-based encoder-decoder architectures to generate image captions with style. Following previous work, our first kind of model is trained in a two-stage fashion: pretraining on a large factual dataset and then training on a stylistic dataset. For this, we design an adversarial training mechanism leading to generated captions that better match human references than previous work on the same dataset, and that are also stylistically diverse. Our second kind of model is trained in an end-to-end fashion, which incorporates both high-level and word-level embeddings representing stylistic information, and leads to the highest-scoring captions according to standard metrics; this end-to-end approach is an effective strategy for incorporating this kind of information.

- We introduce a novel adversarial attack against Faster R-CNN, as a high performing and widely used object detector. Our version of Faster R-CNN is used in the state-of-the-art image captioning system to generate bounding boxes including detected objects present in the image. In contrast to existing attack that changes all bounding boxes, our attack aims to change the label of a particular detected object in both targeted and non-targeted scenarios, while preserving the labels of other detected objects; it achieves this aim with a high rate of success. In terms of understanding the effect of noise injection into the input, we find that although the injected perturbations that attack all bounding boxes or only a specific object type score similarly on standard visual perceptibility metrics, the impact on generated captions is dramatically different.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Motivation

As a branch of computer science, artificial intelligence (AI) is concerned with simulating behaviours of intelligent beings. To do so, computers need to perceive, learn and respond to different world states intelligently. For example, computer vision (CV) is a field of AI that aims to develop methods to enable machines to see, identify and understand visual content in a manner motivated by human visual capacities. Natural language processing (NLP) as another field of AI assists computers to process, understand and generate human languages. It also aims to enable the interaction between humans and machines using language. Currently, deep neural networks are the state-of-the-art in machine learning for many applications. Due to their complex structures including multiple layers, they are able to capture different levels of information [15–17]. In terms of concrete applications, this has led to the state-of-the-art

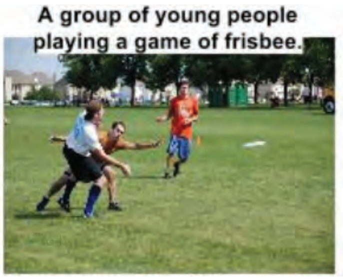


Figure 1.1: An example of machine generated image captions [1].

for different types of image-grounded communications for combining image and language representations. This thesis focuses on one type of image-grounded communication using deep neural networks: *automatic image captioning* — producing image descriptions by understanding the visual content from the field of CV and generating captions from the field of NLP.

Automatic image captioning aims to describe visual content in a form of natural language which is grammatically correct and semantically correlated with visual content [1, 7, 8, 18–20]. For example, in Figure 1.1, "a group of young people playing a game of frisbee" describes the image in terms of attributes, objects and actions in a correct grammatical form. This is useful when humans need automatically generated interpretations for images [21], e.g. for assisting visually impaired people who cannot recognize visual content [1], designing complex search engines using natural language queries [18], interpreting images in newspaper articles [22] and generating automatic comments for images [23]. As shown in Figure 1.2, current image captioning systems usually contain an encoder-decoder architecture where a convolutional neural network (CNN), as a popular deep learning approach in CV, captures visual content and a long short-term memory (LSTM) network, as a popular deep learning approach in NLP, generates image captions. The systems are inspired by the encoder-decoder model used in neural machine translation [24] for translating a source language to a target language.

Most image captioning systems are trained using a few image captioning datasets including MSCOCO [25], Flickr30k [4, 26] and Flickr8k [18]. Recently, various kinds of additions have been made to image captioning, usually outside of these datasets, such as describing novel objects and concepts [27, 28]. These kinds of models use an external corpus or source to generate keywords or phrases which have rarely appeared or do not exist in the datasets. For

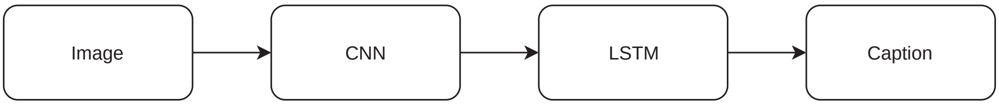
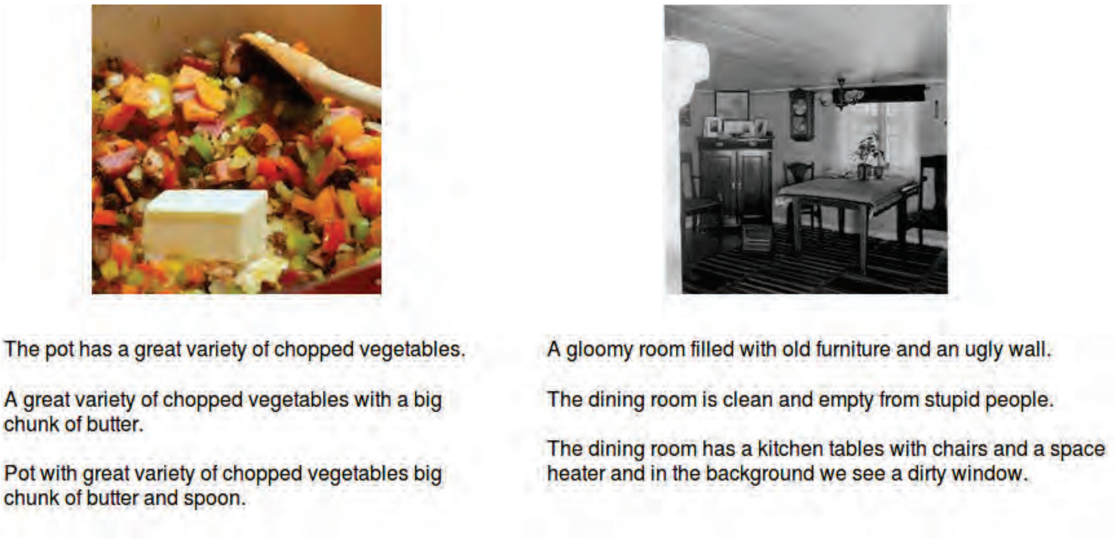


Figure 1.2: A usual encoder decoder model for image captioning. CNN is a convolutional neural network to capture visual content and LSTM is a long short-term memory to generate image captions.

example, they use labeled images from classification or object detection tasks [8]. This aims to enhance the ability of current image captioning systems to generate a wider set of objects or concepts recognized from visual data [27–29]. Additionally, style-bearing image captioning has been investigated using an external stylistic dataset. In describing an image, style could refer to captions being "positive" and "negative" [2], or "romantic" and "humorous" [30]. Adding style makes captions more interesting, more attractive and more expressive [30, 31]. It also has practical applications such as enhancing the engagement level of users interacting with chatbot platforms [32] and automatically generated comments of online photos and videos [23]. These different kinds of additions usually aim to encourage diversity in the generated captions and have been studied for both single-sentence captioning and paragraph generation tasks [33, 34]. In terms of approach, these extensions to image captioning often start with a standard architecture like the one in Figure 1.2, and include components to 'push' or direct the caption to incorporate the desired addition. Following the terminology of Hu *et al.* [35], who proposed techniques for the more general problem of 'controlled generation of text', where some attributes of the text are required to be included, the thesis refers to 'controlled image captioning'.

The particular case of controlling or directing image captions to incorporate style is one focus of this thesis. Image captioning systems have mostly aimed to describe the content of an image using a neutral sentence [1, 7, 8, 19, 20] such as the caption "a group of young people playing a game of frisbee" for Figure 1.1. However, Mathews *et al.* [2] argued that purely objective captions may not fully engage humans, and consequently pioneered an approach to incorporating stylistic or non-factual information into image captions, as in Figure 1.3; later related work, such as by Gan *et al.* [30] and Chen *et al.* [31], has expanded on the idea. A potential problem is that 'pushing' captions to incorporate such non-grounded information could lead to a divergence from accurately describing visual content. Figure 1.4



The pot has a great variety of chopped vegetables.

A great variety of chopped vegetables with a big chunk of butter.

Pot with great variety of chopped vegetables big chunk of butter and spoon.

A gloomy room filled with old furniture and an ugly wall.

The dining room is clean and empty from stupid people.

The dining room has a kitchen tables with chairs and a space heater and in the background we see a dirty window.

Figure 1.3: Human generated captions with positive (left) and negative (right) sentiments, as a kind of styles [2].

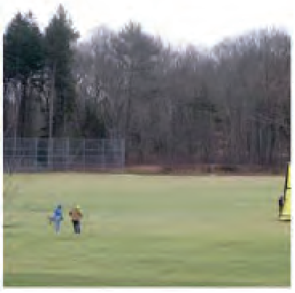


Figure 1.4: "A dead man is playing frisbee on a field". An example of automatically generated image captions with style that does not describe visual content correctly. Our proposed models can change this to "A group of stupid people are playing frisbee on a field" [3].

is an automatically generated image caption having style: the caption wrongly includes "dead" to describe the image. In considering how to make sure too great a divergence from visual content does not occur, however, it is necessary not to restrict captions so much that diversity is lost. This is particularly the case in the usual two-stage architectures that we discuss below and throughout the thesis: in these, the neutral caption is produced first, and the stylistic aspect added later, in a manner that can lead to the 'forgetting' of the basic neutral content.

A related but different kind of augmentation of standard captions that has not previously been tackled is the inclusion of emotional aspects of an image. For example, Figure 1.5 has

igure has been suppressed due to copyright reasons

Figure 1.5: A human generated image caption extracted from Flickr30k dataset [4] including emotional content. “Two females one wearing blue the other black smiling and laughing.”

as a core part of the caption content that the women are smiling and laughing. Incorporating emotional content, as a general characteristic of intelligent systems, plays a critical role in generating desirable outcomes and making communications more expressive [36–39]. This differs from the above sentiment-infused captioning in that is incorporating visually grounded content; the stylistically enhanced captions typically embody descriptions of an image that represent an *observer’s* view towards the image (e.g. *a cuddly cat* for a positive view of an image, versus *a sinister cat* for a negative one). Although they differ in terms of the specific augmentations, both cases — incorporating style and incorporating emotion — raise the question of how to direct captions to incorporate this extra information in a way that does not lead to captions that less accurately capture the other aspects of the image.

To investigate this overall topic of how best to direct image captions to incorporate extra information, this thesis looks at three different aspects. For the first aspect, we target the emotional content of images as an extra layer of information on top of an image captioning system. We propose several novel methods for facial expression analyses and image captioning systems, and examine which best adds facial emotion descriptions to captions. For the second aspect, we work on the problem of infusing captions with sentiment first defined by Mathews *et al.* [2], and again propose several novel models to examine which best adds sentiment while staying faithful to the visual content but not sacrificing diversity. These first two applications prompt a third, more general kind of investigation: seeing how captions can be changed in response to changes in images. We specifically study the impact of this kind of changes on generated captions by introducing an adversarial attack, having applications in assessing the robustness of deep learning-based approaches [40–42], against the object detection task playing an important role in the state-of-the-art image captioning system [8] to perceive visual content. The following section describes each of these three aspects in more detail.

1.2 Research Questions and Scope

1.2.1 Image Captioning using Emotional Content

Image captioning is the process of generating a natural language description of an image. Most current image captioning models, however, do not take into account the emotional aspect of the image, which can be relevant to activities and interpersonal relationships represented therein. Towards developing image captioning models that can produce captions incorporating this, we use facial expression features extracted from images that include human faces. Specifically, we aim to improve the descriptive ability of captioning models using these features. Moreover, covering the emotional content of images, recognized using facial expression analyses, is an important aspect of generating human-like descriptions. As shown in the example human generated caption in Figure 1.5, an effective image captioning system needs to detect the emotional content of the image to generate the relevant words *smiling* and *laughing*.

To propose image captioning models incorporating emotional content, we first need to train a model to extract facial expression features for use in our image captioning models. Thus, our first research question, as a precursor to the work on image captioning, is:

RQ 1. *How can a facial expression recognition model be trained to generate representative and transferable features for other tasks?*

To address this question, we train a state-of-the-art facial expression recognition (FER) model for two purposes: as a basis for evaluating transfer learning using FER for another task and to advance engagement recognition — that is, determining whether a person is interested and engaged in a task — for education. In fact, we study the transferability of the facial features extracted by the FER model for engagement recognition which plays a critical role for building intelligent educational interfaces [11, 43, 44]. Facial information has recently been investigated as one of the main sources for recognizing engagement [45]. This motivates us to propose a novel engagement recognition model initialized by the weights of the FER model.

We also collect a new dataset for training the engagement recognition model for facilitating research in this domain. Annotating a dataset for engagement recognition is a difficult task because of the complexities [46] and ambiguities [47] of defining engagement. This has led to a small amount of work for engagement recognition which can result in a bad performance due to poor quality annotation. Therefore, we construct an engagement recognition dataset by expert annotators with backgrounds in psychology to better incorporate the psychological phenomena of engagement in the annotation task. Moreover, we use a large FER dataset to provide an initial state for our engagement model to effectively recognize engagement. To evaluate whether the FER representation is usefully transferable, we compare our engagement model against baselines that do not use this representation.

Then, we propose image captioning models to employ FER features extracted by the FER model. Thus, our second research question is:

RQ 2. *Given the existing image captioning datasets, can incorporating the recognized emotions from facial expression analyses produce better image captions?*

To address this question, we propose several different novel image captioning models to generate image captions incorporating FER features. We extract a subset of the Flickr30K image caption dataset [4] that includes human faces, and use this subset for training our proposed models. We first incorporate high-level facial expression features as fixed one-hot encoding representations inspired by previous work for incorporating similar representations [35, 48]. As a second approach, we incorporate low-level facial expression features such as convolutional features produced by the FER model. We employ an attention mechanism to generate more effective image captions using this fine-grained facial information. We mainly build our models based on two state-of-the-art image captioning systems with attention [7, 8]. To evaluate the generated captions by the models, we look at the usual metrics for measuring the overall aggregate quality of captions. But because we are also interested in the way particular models' incorporation of this additional emotion information might direct the captions to be (for example) less diverse or less well-aligned to the original captions, we propose several other measures to analyse linguistic properties of the generated captions.

1.2.2 Image Captioning with Stylistic Information

Although most image captioning models generate factual captions, some recent work has aimed at generating style-bearing image captions [2, 30, 31, 49]. Style is usually referred to a set of attributes and characteristics which are distinct from the semantic aspects of text [50]. For example it can be "romantic" or "humorous" [30]; or "positive" or "negative" [2]. As shown in Figure 1.3, a caption with positive sentiment for the left image is "The pot has a great variety of chopped vegetables" and a caption with negative sentiment for the right image is "A gloomy room filled with old furniture and an ugly wall".

In image captioning with style, the generated captions should include word choices reflecting the targeted style and also describe the visual content correctly. Previous work has applied two-stage training: learning to describe visual content using a factual caption dataset, and then learning a chosen style using a (smaller) stylistic dataset. However, this can lead to image captions with the lack of diversity in terms of stylistic patterns since the second stage of training usually focuses on adding style-bearing information using a small number of captions. For example, captions frequently use the negative adjective "dead", even if not really suitable (Figure 1.4). This is because of the popularity of "dead" in the small stylistic dataset. Moreover, collecting a large-scale dataset is difficult since it requires annotating a large number of images with style-bearing captions. Motivated by these issues, our third research question is:

RQ 3. *What kind of two-stage image captioning model can better generate captions with diverse stylistic patterns?*

To address this question, we propose an image captioning model with style. We train the model in two stages similar to the SentiCap model [2], as the starting point for image captioning with style. However, our proposed model is equipped with an adversarial training mechanism which consists of a caption discriminator and an attention-based caption generator. The discriminator guides the generator to produce image captions similar to human-generated captions with style, having highly diversified stylistic patterns. To measure the effectiveness of our proposed model, we use novel linguistic metrics showing the diversity of generated stylistic patterns in addition to the usual image captioning metrics.

We observe from the above that, using these two stages of training, training an image captioning model which is able to distinguish between stylistic and factual aspects of captions is a hard task. In the second stage, the previous models usually focus on transferring style by directly modifying word prediction [2] or word embedding [30, 31], without considering the visual content. Mathews *et al.* [49] observes that this addition of style can come at the expense of accurate visual description. Thus, most work on generating stylistic image captions adds style in a way that can detract from the visual grounding, leading to less accurately grounded captions. Motivated by this, our forth research question is:

RQ 4. *How can an image captioning model be trained in an end-to-end fashion to generate diverse stylistic captions which are still faithful to visual content?*

To address this question, we propose a model trained in an end-to-end training fashion: in addition to transferring style, this encourages the preservation of semantic relationships between captions and images. Our proposed model uses an attention mechanism to make a strong connection between visual regions and generated words or phrases. Moreover, the model embeds the targeted style to capture the overall style of the generated caption and the word-level style of each generated word. These are inspired by Zhou *et al.* [51] and Ghosh *et al.* [52] in controlled text generation. We will use both of these and see whether they are complementary to generate image captions with style.

1.2.3 Image Captioning of Adversarial Images

The above two tasks look at directing captions to include additional information which can be visually-grounded or non-grounded. We aim to explore and characterise the effects of this kind of directed addition of content; one approach is to change different parts of the visual content and see the impact on the generated captions, through the mechanism of adversarial attack construction.

The vulnerability of deep learning-based approaches to adversarial attacks has been extensively investigated in the previous work [40, 41]. For example, adversarial examples (images with imperceptible perturbations) are used to mislead image classifiers [40, 42, 53–57]. Some adversarial attacks recently studied attacking object detectors [58, 59] such as

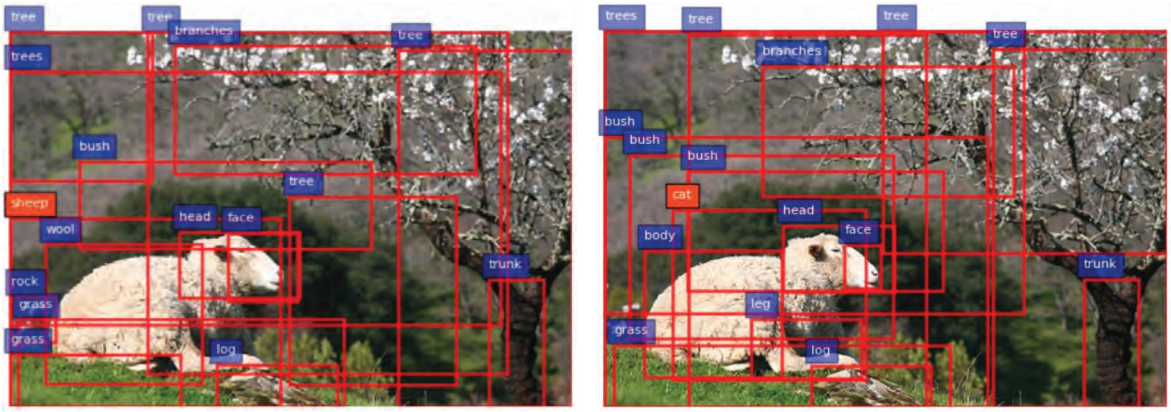


Figure 1.6: Example of our adversarial attack adding imperceptible perturbations to the first image (left) resulting in the second image (right). It succeeds in changing a predicted class from “sheep” to “cat”. It also succeeds in changing the generated caption from “A sheep lying in the grass next to tree” to “A cat is lying down in the grass”.

Faster R-CNN [60]. Faster R-CNN generates the coordinates of bounding boxes contain objects and classifies the objects at same time. It is the main part of the state-of-the-art image captioning system [8] serving to generate visual features. Motivated by this, our fifth research question is:

RQ 5. *How is an adversarial attack against object detection in an image possible, such that it changes the label of a particular object, and what impact does that have on the captions generated by a state-of-the-art image captioning model?*

To address this question, we propose an adversarial attack for a version of Faster R-CNN, used in the state-of-the-art image captioning model [8], and study the impact of this attack on the generated captions by the model. The attack specifically targets changing the label of a particular object, by adding noise that is imperceptible by standard metrics to the object’s bounding boxes, while preserving the labels of other detected objects in the image. Figure 1.6 shows an example of our proposed attack, where the label of an object is changed from “sheep” to “cat”. The generated caption is also changed from “A sheep lying in the grass next to a tree” to “A cat is lying down in the grass”. We will also study the impact of other more general attacks with similarly imperceptible noise on the generated captions, such as attacking all detected objects.

1.2.4 Thesis Structure and Contributions

By considering the research questions, the thesis's aims can be divided into three main sections: First, to propose novel image captioning architectures that incorporate additional visually-grounded information, recognized using a state-of-the-art facial expression recognition (FER) model, to further improve the description of visual content. Second, to present novel image captioning systems using non-grounded information and learning from an image captioning dataset with style to generate more engaging image captions. Third, to introduce an adversarial attack against Faster R-CNN generating bottom-up features for the state-of-the-art image captioning model [8]. In addition to examining the reliability of Faster R-CNN, this is useful to study how this attack can impact on the generated captions by the model.

In order to achieve these aims, we first address **RQ 1** by training a state-of-the-art FER model and applying this for the engagement recognition task in Chapter 3. Then, the model is used to extract FER features for automatic image captioning. To address **RQ 2**, in Chapter 4, we propose several image captioning models using the FER features. We present several novel image captioning models with style to address **RQ 3** and **RQ 4** in Chapters 5 and 6, respectively. Here, more diverse and correlated stylistic image captions are targeted by the captioning models. Finally, we introduce an adversarial attack for object detection and study its impacts on image captioning in Chapter 7, where we address **RQ 5**. The structure and contributions of the thesis are described as follows:

- **Chapter 2: Background** gives a general overview on image captioning and necessary technical background related to the thesis.

- **Chapter 3: Automatic Recognition of Student Engagement using Deep Learning and Facial Expression** To our knowledge, the work in this chapter is the first time a rich face representation model has been used to capture basic facial expressions and initialize an engagement recognition model, resulting in positive outcomes. This shows the effectiveness of applying basic facial expression data in order to recognize engagement. To our knowledge, this is the first study which models engagement using deep learning techniques. We have collected a new dataset we call Engagement Recognition (ER) dataset to facilitate research on engagement recognition from images. To handle the complexity and ambiguity of the engagement concept, our data is annotated in two steps by expert people, separating the behavioral and emotional dimensions of engagement. The final engagement label in the ER dataset is the combination of the two dimensions. The proposed model outperforms a comprehensive range of baseline approaches, and shows that facial expression recognition models can produce useful transferable representations of human faces.

- **Chapter 4: Image Captioning using Facial Expression and Attention** In this chapter, we propose Face-Cap and Face-Attend models to effectively employ facial expression features to generate image captions. To our knowledge, this is the first study to apply facial expression analyses in image captioning tasks. The generated captions using the models are evaluated by all standard image captioning metrics. The results show the effectiveness of the models comparing to a comprehensive list of image captioning models using the FlickrFace11K dataset, a subset of images from the Flickr 30K dataset [4] that includes human faces. We further assess the quality of the generated captions in terms of the characteristics of the language used, such as variety of expression. Our analysis suggests that the captions generated by the models improve over other image captioning models by better describing the actions performed in the image.

- **Chapter 5: Towards Generating Stylized Image Captions via Adversarial Training** To generate human-like stylistic captions in a two-stage architecture, we propose Attend-GAN using both the designed attention-based caption generator and the adversarial training mechanism in this chapter. Attend-GAN achieves results which are significantly better than the state-of-the-art and a comprehensive range of baseline models for generating image captions with styles. We show how Attend-GAN can result in stylistic captions which are strongly correlated with visual content. Attend-GAN exhibits significant variety in generating adjectives and adjective-noun pairs.

- **Chapter 6: Image Captioning using Sentiment and Attention** In this chapter, we propose an attention-based image captioning model, that we name Senti-Attend, trained in an end-to-end fashion, applying two complementary representations of sentiment information, high-level and word-level, to generate sentiment-bearing descriptions. Senti-Attend outperforms the state-of-the-art in generating sentiment-bearing descriptions. We also show via ablation experiments that all components are useful in producing these results. We show that Senti-Attend can generate sentiment-bearing captions which include highly diversified adjectives with sentiment and can preserve the semantic correlation between an image and its generated caption.

- **Chapter 7: Type-Specific Adversarial Attack for Object Detection** The work in this chapter is the first study to successfully apply both targeted and non-targeted attacks against Faster R-CNN on different types of images. This is the first work which studies an adversarial attack against Faster R-CNN in a constrained setting where only pixels within a specified object type are changed. Our proposed attack changes the label of a particular object in an image with high success rates while preserving the labels of other detected objects. We propose an attack which works for arbitrary images and can be straightforwardly generalised to change the labels of multiple detected objects. We show that the proposed attack adds imperceptible perturbations to the image. This is the first work to study the effect of attacking Faster R-CNN on the state-of-the-art image captioning system [8] which uses bottom-up, object-based, features. We show that it leads to many fewer changes in captions than a method based on Xie *et al.* [58] which modifies all the objects. In fact, this shows that small changes across all image, like those generated by Xie *et al.*, can produce major shifts in captions, but some techniques (like ours, changing only one object) can result in very small changes in captions.

- **Chapter 8: Conclusions and Future Work** recaps the thesis's findings and presents potential directions as future work.

Figure 1.7 shows the dependencies of different parts and sections related to Chapters 2, 3, 4, 5, 6 and 7 in the thesis.

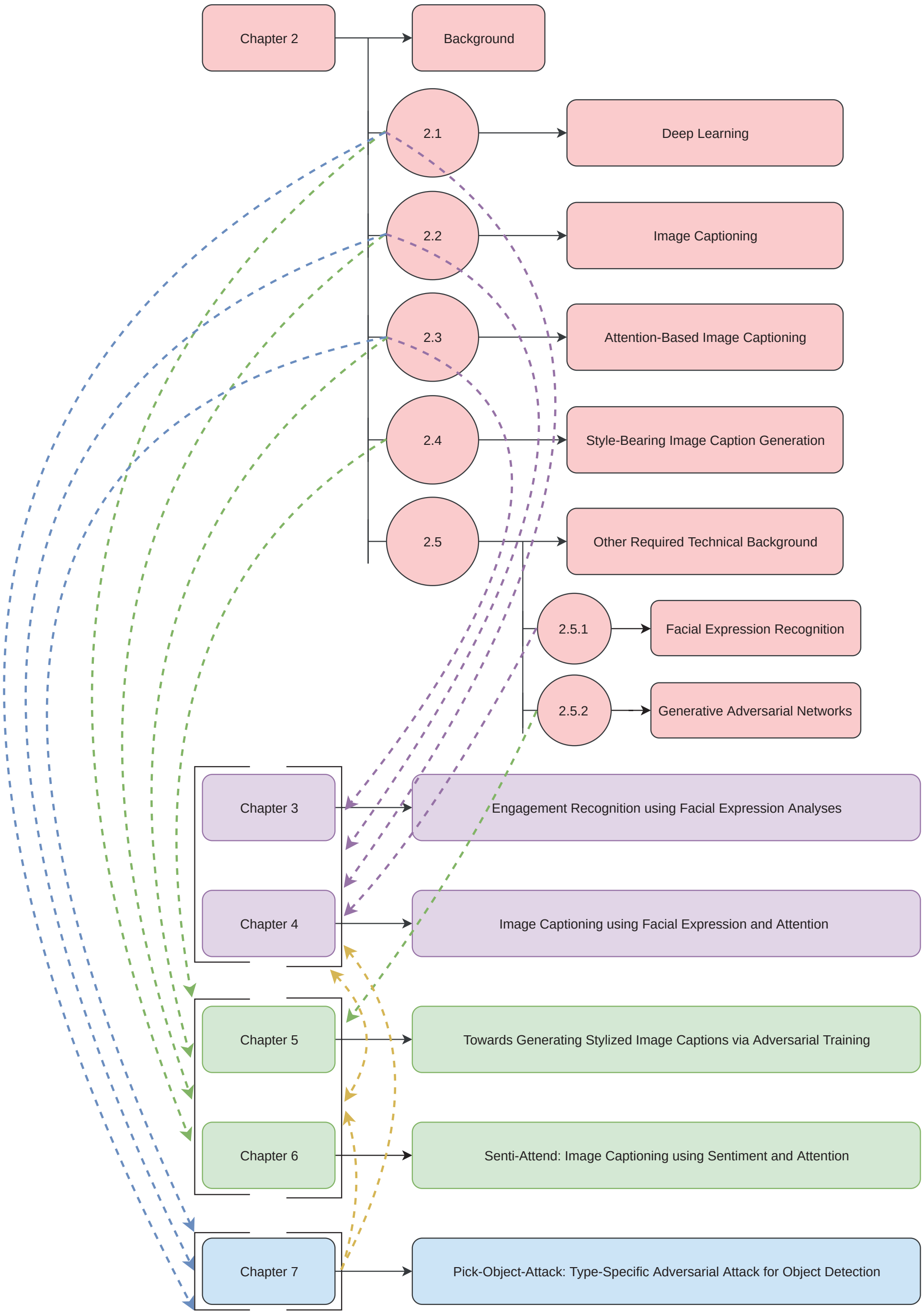

Chapter 2
Background
2.1
Deep Learning
2.2
Image Captioning
2.3
Attention-Based Image Captioning
2.4
Style-Bearing Image Caption Generation
2.5
Other Required Technical Background
2.5.1
Facial Expression Recognition
2.5.2
Generative Adversarial Networks
Chapter 3
Engagement Recognition using Facial Expression Analyses
Chapter 4
Image Captioning using Facial Expression and Attention
Chapter 5
Towards Generating Stylized Image Captions via Adversarial Training
Chapter 6
Senti-Attend: Image Captioning using Sentiment and Attention
Chapter 7
Pick-Object-Attack: Type-Specific Adversarial Attack for Object Detection

Figure 1.7: Dependencies shown with different colors of parts and sections in the thesis.

2

Background

In this chapter, we briefly overview related work and technical background required for the thesis. We first review some major developments in deep learning and the main fields relevant to the thesis including vision and language processing in §2.1. Then, as the main focus of the thesis, we explain about image captioning combining visual and language processing techniques in §2.2. We describe attention-based image captioning in §2.3 which is the basis of our proposed attention-based image captioning models in the following chapters. In §2.4, we explain style-bearing image caption generation as a common part between Chapters 5 and 6 where we propose novel image captioning systems directing generated captions to use style. Finally, we discuss other related work in §2.5 including facial expression recognition, as a common part between Chapters 3 and 4, and generative adversarial networks, as a part of Chapter 5.

2.1 Deep Learning

Conventional machine learning approaches require domain expertise and human knowledge to transform the input data to feature vectors when building classification or predication models. Deep learning, as a subdomain of machine learning (ML) and artificial intelligence (AI), aims to automatically provide representations from input data to make classification or predication easier [15–17]. It refers to a deep version of artificial neural networks (NNs) generating distributed representations for different concepts and notations with inspiration from biological systems [61]. NNs contain neurons as the elementary processing units activated from the previous weighted neurons or the input data [62]. Deep learning is a version of representation learning relating lower and simpler concepts to higher and more complex concepts in a form of different representation layers [17, 63]. For instance, for an image, the lower layers typically end up representing simple concepts such as edges in different regions of the image and the higher layers may represent different parts of an object existing in the image (Figure 2.1).

Deep learning for both versions of NNs including feedforward neural networks (FNNs) and recurrent neural networks (RNNs) has recently been successful in a wide range of fields including visual [64], language [65] and speech processing [66]. RNNs are able to record and capture sequences of data patterns in addition to processing parallel information as in FNNs [62]. The success of such networks has been possible because of large scale datasets including ImageNet [67] and powerful systems including GPUs with high computing abilities.

In this thesis, we use deep learning approaches for both visual and language processing which are explained in the next sections. After these, we explain image captioning as the combination of visual and language processing techniques as the main focus of this thesis.

2.1.1 Visual Processing

The thesis focuses on image classification and object detection as specific visual processing tasks. These aim to identify visual content with different purposes: image classification assigns a label to the whole image while object detection first detects bounding boxes including

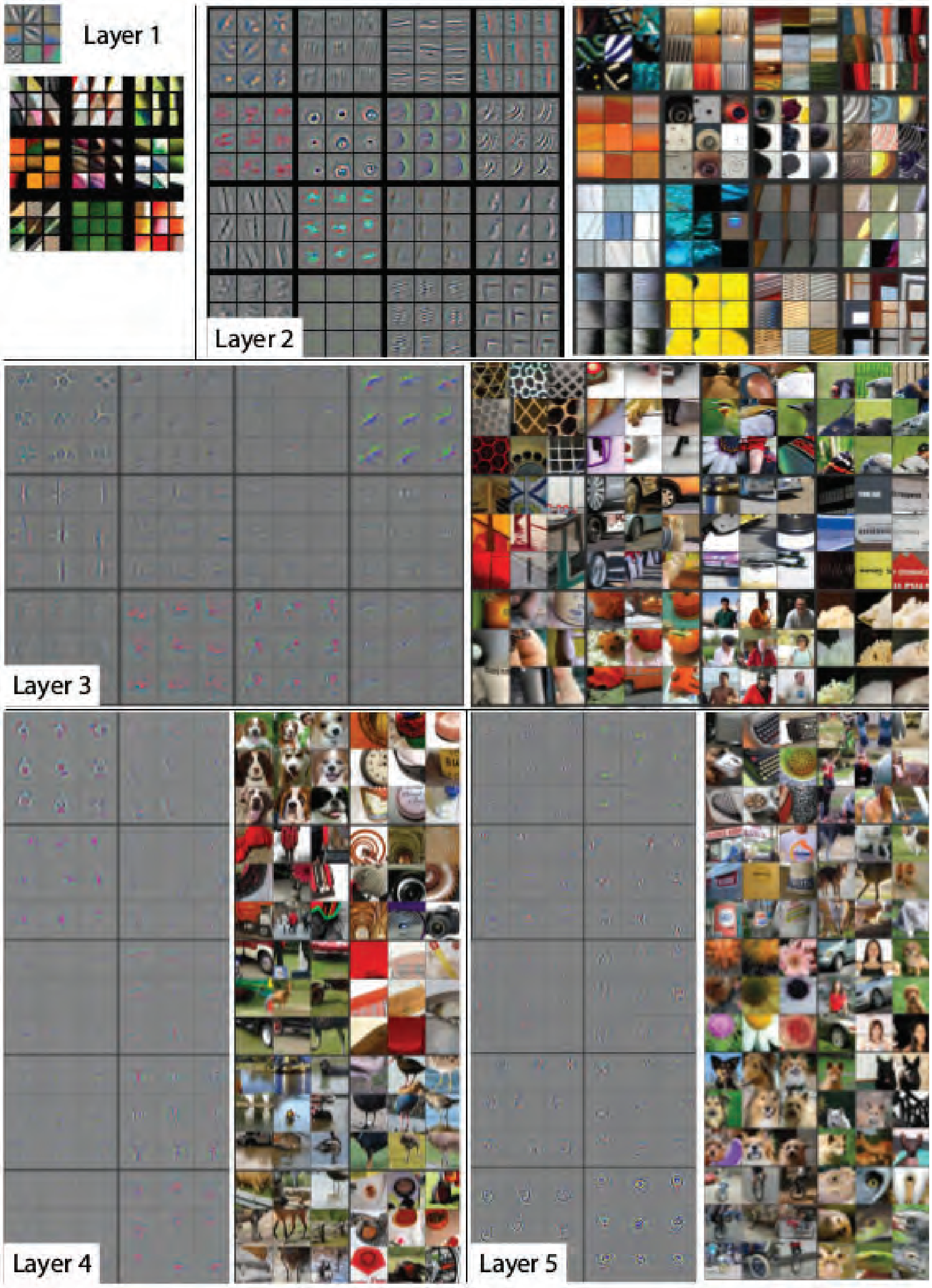


Figure 2.1: Visualizing the features extracted by different layers in a fully trained convolutional neural network with their corresponding image regions [5].

objects in an image and then assign labels to the bounding boxes. The recent tremendous advancements in visual processing have started with recent progress on Convolutional Neural networks (CNNs) [68] such as AlexNet, the first large scale CNN which achieved the highest performance in the visual recognition challenge 2012 on the ImageNet dataset [64].

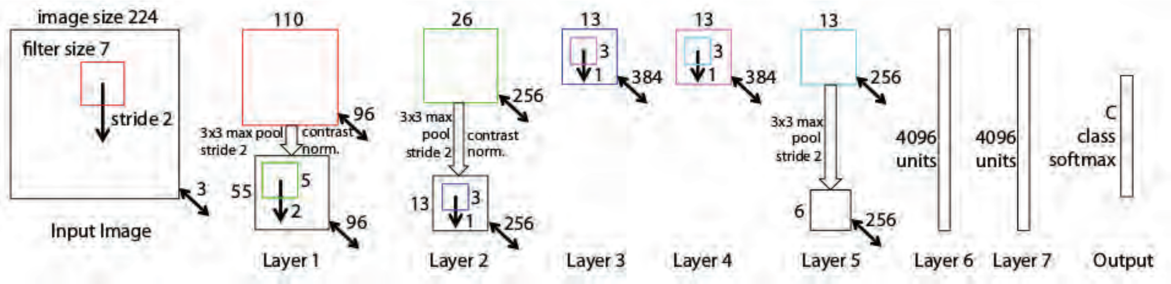


Figure 2.2: A full CNN architecture including convolutional layers specified with their filter size and stride length [5]. The small number on the lower right corner of each layer denotes the depth of the layer. Layers 6 and 7 are fully-connected layers. The output layer has a softmax function generating the probability distribution for targeted classes.

CNNs include a number of convolutional layers considering an image as a matrix of pixel values. The layers apply different filters to the image to extract different feature maps (extracted matrices corresponding to the image) inspired by the visual cortex of biological systems [62]. The filters are small matrices with specific sizes including trainable weights. They scan with pre-defined stride lengths over the matrices corresponding to the image. The feature maps are the element-wise multiplications between the weights and every location of the matrices, which are then summed to generate a single value. The value is passed to an activation function such as rectified linear unit (ReLU). Figure 2.2 shows a full CNN architecture including convolutional layers. As a part of CNN architectures, pooling layers aim to reduce the spatial size of the feature maps by applying different operations. For example, the max pooling applies a filter with a particular size to specify the maximum value across the area covered by the filter (layers 1, 2 and 5 in the figure). Layers 6 and 7 in the figure are fully-connected layers with ReLU. The last layer is a fully connected layer with a softmax function calculating the probability distribution across different classes. The weights of different filters of convolutional layers are learned through a process called backpropagation. After achieving the end of a CNN model, a loss value is calculated which is showing the difference between the prediction and the ground-truth label. For example, cross-entropy loss, as a popular loss function, is calculated using Equation 2.1.

$$\text{crossentropy} = -\sum_{i=1}^{n} (y_i \log(y_i')) \tag{2.1}$$

where y_i and y_i' are the ground truth and the prediction, respectively, and n is the number of classes. Then, gradients are calculated using the loss value for each layer to update the weights.

Different improvements have been made on CNNs [5, 10, 69, 70]. For example, in addition to generating a better performance, Zeiler *et al.* [5] showed how CNNs can learn low- to high-level information by visualizing the feature maps generated by each convolutional layer (Figure 2.1). They did this by proposing deconvolutional networks mapping the feature maps to the input image. Simonyan *et al.* [10] and Szegedy *et al.* [69] have shown that the depth of CNNs plays a critical role in the achieved performance. For example, Simonyan *et al.* [10] have proposed very deep CNN architectures such as VGG networks including up to 19 layers. In addition to the depth of CNNs, He *et al.* [70] proposed new architectures called Residual networks to handle some training issues such as vanishing gradient attached to very deep neural networks. Residual networks include residual modules skipping the training phases of selected layers by applying skip-connections.

2.1.2 Language Processing

The language processing domain has seen considerable improvements across different tasks using deep learning approaches [65, 71–74]. The modern image captioning platform [1, 7], which is the focus of this thesis, is inspired by deep learning-based machine translation systems for sequence to sequence learning (seq2seq) [24, 75]. This platform is usually based on Long Short-Term Memory (LSTM) Networks [76, 77] or their variants such as the Gated Recurrent Unit (GRU) providing a simpler version of LSTM by combining its different gates [75]. LSTM networks are a form of gated units showing more effective results in comparison with the previous versions of RNNs and including particular hidden units to remember input data for a long period of time (Figure 2.3).

LSTM networks have four important gates to calculate their memory cell ($\boldsymbol{c}_x$) and hidden unit ($\boldsymbol{h}_x$): the input gate ($\boldsymbol{i}_t$), which controls the input word ($\boldsymbol{w}_{t-1}$) embedded in M dimensions ($\boldsymbol{w}_x \in \mathbb{R}^M$); the forget gate ($\boldsymbol{f}_t$), which forgets the previous memory ($\boldsymbol{c}_{t-1}$); the output gate ($\boldsymbol{o}_t$), which decides on transferring knowledge from the current memory to the current hidden state; and the input modulation gate ($\tilde{\boldsymbol{c}}_t$), which adjusts the new information for the memory

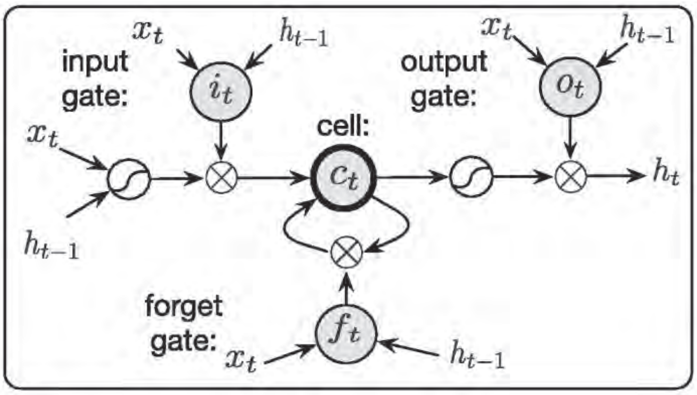


Figure 2.3: An LSTM architecture showing different gated units [6]. The input word and the hidden state at the current time step t are shown with $\boldsymbol{x}_t$ and $\boldsymbol{h}_t$, respectively.

(Equation 2.2).

$$\begin{aligned}
\boldsymbol{i}_t &= \sigma(\boldsymbol{H}_i \boldsymbol{h}_{t-1} + \boldsymbol{W}_i \boldsymbol{w}_{t-1} + \boldsymbol{b}_i) \\
\boldsymbol{f}_t &= \sigma(\boldsymbol{H}_f \boldsymbol{h}_{t-1} + \boldsymbol{W}_f \boldsymbol{w}_{t-1} + \boldsymbol{b}_f) \\
\boldsymbol{o}_t &= \sigma(\boldsymbol{H}_o \boldsymbol{h}_{t-1} + \boldsymbol{W}_o \boldsymbol{w}_{t-1} + \boldsymbol{b}_o) \\
\tilde{\boldsymbol{c}}_t &= \boldsymbol{H}_g \boldsymbol{h}_{t-1} + \boldsymbol{W}_g \boldsymbol{w}_{t-1} + \boldsymbol{b}_g \\
\boldsymbol{c}_t &= \boldsymbol{f}_t \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \tanh(\tilde{\boldsymbol{c}}_t) \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \tanh(\boldsymbol{c}_t)
\end{aligned} \tag{2.2}$$

where σ is the sigmoid function. $\boldsymbol{H}_x, \boldsymbol{W}_x$, and $\boldsymbol{b}_x$ are the trainable weights and biases.

In seq2seq systems, LSTM networks perform the role of an encoder compressing the input sequence of words into a dense feature representation and a decoder generates the output sequence of words using the representation. This encoder-decoder system has been equipped with an attention mechanism by Bahdanau *et al.* [78]. The attention mechanism can make the dense feature representation richer for the decoding phase and provide customized or weighted connections between the representation and each output element. This also makes remembering long input sequences more effective. In fact, the attention mechanism enables the decoder to selectively attend to the required information from the input sequence. Vaswani *et al.* [79] has built on the work of Bahdanau *et al.* [78] by introducing Transformers. They replace recurrent units in the encoder-decoder framework with multi-head attention mechanisms to handle the issues attached to recurrent units such as their sequential nature and the difficulty to learn dependencies over long sequences.

2.2 Image Captioning

Image captioning is the generation of automatic descriptions for visual content such as entities, actions and visual scenes. It has a lot of important applications such as helping blind people to understand and perceive visual content [1], building complex search engines to access the existing information in images [18] and providing automatically generated comments for images or videos in different platforms [23].

Earlier image captioning systems can be classified into two main categories including template-based and retrieval-based systems [21, 22]. Template-based models first detect visual objects, their attributes and relations and then fill pre-defined templates' slots [80]. These approaches even combined more complex structures to incorporate the relations among visual objects or phrases related to them [81, 82]. However, hand-designed features and pre-defined templates are the basis of this kind of approaches, with corresponding limitations on the text generated. Retrieval-based models generate captions using available ones corresponding to similar images [18]. In this work, the authors framed image captioning as a retrieval and ranking task where captions are selected from a large set of human-written captions [18, 83–85], which added some diversity compared to previous methods. These approaches usually aim to share an embedding space for images and captions. For an image, the nearest captions are selected in the shared embedding space. However, the approaches do not aim to generate new image captions. Thus, they are not able to deal with unseen objects or different compositions of seen objects. Moreover, these earlier image captioning systems do not incorporate the detection and generation steps using an end-to-end training approach.

In response to these issues, modern image captioning systems, explained in the next section, are currently the most popular ones. They tackle image captioning as a generation task using deep learning models.

2.2.1 Modern Image Captioning Systems

Modern image captioning systems usually use an encoder-decoder paradigm [1, 7, 86]. They apply a top-down approach where a Convolutional Neural Network (CNN) learns the image content (encoding) followed by a Long Short-Term Memory (LSTM) generates the

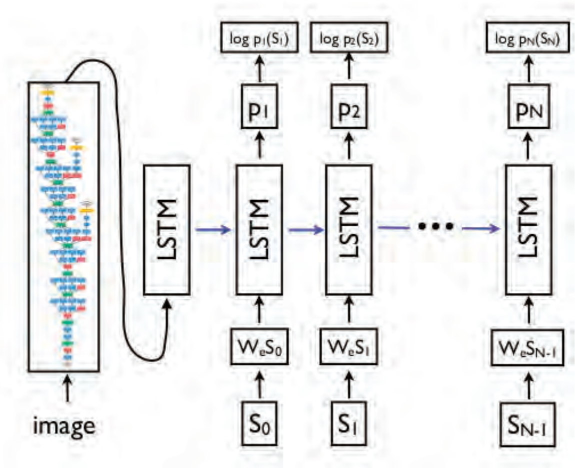


Figure 2.4: The Show-Tell image captioning model [1]. $\boldsymbol{S}_x$ is the input word and $\boldsymbol{p}_x$ is the probability distribution of the next generated word. The embedded input word is shown by $\boldsymbol{W_e S_x}$.

image caption (decoding). This follows the paradigm employed in machine translation tasks, using deep neural networks [24], to translate an image into a caption. Figure 2.4 shows the architecture of Show-Tell [1] as the first image captioning system to use an encoder-decoder framework. As shown in the figure, the model contains an LSTM network with gated units similar to Equation 2.2. However, the features of the input image, extracted by a CNN model, are fed into the initial step of the LSTM network.

This top-down mechanism directly converts the extracted visual features into image captions [87–91]. However, attending to fine-grained and important fragments of visual data, required to provide a better image description, is usually difficult using a top-down paradigm. To solve this problem, a combination of top-down and bottom-up approaches, inspired by the earlier image captioning models, is proposed by You *et al.* [19]. The bottom-up approach overcomes this limitation by generating the relevant words and phrases, which can be detected from visual data with any image resolution, and combining them to form image captions [80, 82, 92, 93].

Moreover, to attend to fine-grained fragments, attention-based image captioning models have been proposed recently [7]. These kinds of approaches usually analyze different regions of an image in different time steps of a caption generation process, in comparison to the initial encoder-decoder image captioning system [1] which considers only the whole image as the initial step for generating image captions. They can also take the spatial information of an image into account when generating the relevant words and phrases in the image caption. The current state-of-the-art models in image captioning are attention-based systems [7, 8, 19, 20], explained in the next section.

2.3 Attention-Based Image Captioning

Visual attention is an important aspect of the visual processing system of humans. It dynamically attends to salient spatial locations in an image with special properties or attributes which are relevant to particular objects. It is different from dealing with the whole image as a set of static extracted features, and assists humans to concentrate more on a targeted object or region at each time step. Although visual attention has been extensively studied in Psychology and Neuroscience [94–97], it has only more recently been adopted in different artificial intelligence fields including machine learning, computer vision and natural language processing. In image captioning, a typical attention mechanism is a Top-Down approach, while the region-based features obtained using an object detector are referred to as Bottom-Up features. The combination of these called Top-Down and Bottom-up attention [8].

2.3.1 Top-Down Image Captioning

The first image captioning model with attention was proposed by Xu *et al.* [7]. It is a variant of Show-Tell (Figure 2.4), referred to as Show-Attend-Tell. The model uses visual content extracted from the convolutional layers of CNNs, referred to as spatial features, as the input of a *spatial* attention mechanism to selectively attend to different parts of an image at every time step in generating an image caption (Figure 2.5). This work is inspired by the work of Bahdanau *et al.* [78], since extended by Vaswani *et al.* [79], who employed attention in the task of machine translation; by Mnih *et al.* [98]; and by Ba *et al.* [99] who applied attention in the task of object recognition. Image captioning with attention differs from previous encoder-decoder image captioning models by concentrating on the salient parts of an input image to generate its equivalent words or phrases simultaneously. Xu *et al.* [7] proposed two types of attention including a hard (stochastic) mechanism and a soft (deterministic) mechanism. In the soft attention mechanism, a weighted matrix is calculated to weight different parts of an

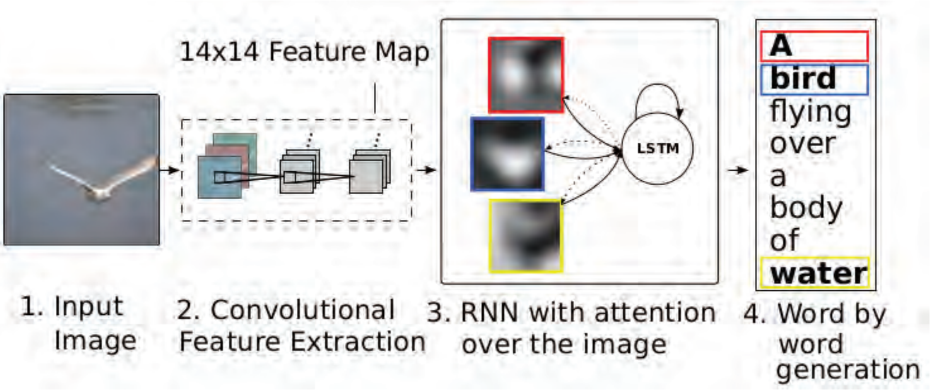

14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Figure 2.5: The Show-Attend-Tell image captioning model [7].

image as the input to the decoder (interpreted as probability values for considering the parts of the image). The hard attention mechanism, in contrast, picks a sampled annotation vector corresponding to a particular part of an image at each time step as the input to the decoder.

This kind of image captioning models usually includes an LSTM with a visual attention-based mechanism. The model uses the spatial features, $\boldsymbol{a} = \{\boldsymbol{a}_1, ..., \boldsymbol{a}_K\}, \boldsymbol{a}_i \in \mathbb{R}^D$, where $\boldsymbol{a}_i$ is a part of the features belonging to a specific region in the image, and generates an image caption, $\boldsymbol{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}, \boldsymbol{x}_i \in \mathbb{R}^V$. K and D are the dimensions of the spatial features. T denotes the maximum length of the generated captions and V is the size of the vocabulary. Here, $\boldsymbol{a}$ is usually extracted from a convolutional layer of a CNN model. The objective function of an attention-based image captioning model is usually defined as:

$$L(\theta) = -\sum_{1\le t\le T} \log(\boldsymbol{p}(\boldsymbol{x}_t | \boldsymbol{h}_t, \hat{\boldsymbol{a}}_t)) + \sum_{1\le k\le K} (1 - \sum_{1\le t\le T} a_{tk})^2 \tag{2.3}$$

where $\boldsymbol{\theta}$ are the parameters of the model; and $\boldsymbol{p}(\boldsymbol{x}_t | \boldsymbol{h}_t, \hat{\boldsymbol{a}}_t)$, the likelihood of the next generated word, is the output of a multilayer perceptron with softmax:

$$\boldsymbol{p}(\boldsymbol{x}_t | \boldsymbol{h}_t, \hat{\boldsymbol{a}}_t) = \text{softmax}(\boldsymbol{h}_t \boldsymbol{W}_h + \hat{\boldsymbol{a}}_t \boldsymbol{W}_a + \boldsymbol{b}) \tag{2.4}$$

where the learned weights and bias are $\boldsymbol{W}_x$ and $\boldsymbol{b}$, respectively. The last factor in Equation 2.3 is a penalty value to guide the model to include all regions of the input image at the final step of the caption generation process. $\boldsymbol{h}_t$ is the current hidden state calculated using an LSTM and $\hat{\boldsymbol{a}}_t$ is the attention-based content estimated using $\boldsymbol{h}_t$. The LSTM calculates $\boldsymbol{h}_t$ using:

$$\begin{aligned}
\boldsymbol{i_t} &= \sigma(\boldsymbol{H}_i\boldsymbol{h}_{t-1} + \boldsymbol{W}_i\boldsymbol{w}_{t-1} + \boldsymbol{A}_i\hat{\boldsymbol{a}}_t + \boldsymbol{b}_i) \\
\boldsymbol{f_t} &= \sigma(\boldsymbol{H}_f\boldsymbol{h}_{t-1} + \boldsymbol{W}_f\boldsymbol{w}_{t-1} + \boldsymbol{A}_f\hat{\boldsymbol{a}}_t + \boldsymbol{b}_f) \\
\boldsymbol{o_t} &= \sigma(\boldsymbol{H}_o\boldsymbol{h}_{t-1} + \boldsymbol{W}_o\boldsymbol{w}_{t-1} + \boldsymbol{A}_o\hat{\boldsymbol{a}}_t + \boldsymbol{b}_o) \\
\tilde{\boldsymbol{c}}_t &= \boldsymbol{H}_g\boldsymbol{h}_{t-1} + \boldsymbol{W}_g\boldsymbol{w}_{t-1} + \boldsymbol{A}_g\hat{\boldsymbol{a}}_t + \boldsymbol{b}_g \\
\boldsymbol{c}_t &= \boldsymbol{f}_t\boldsymbol{c}_{t-1} + \boldsymbol{i}_t \tanh(\tilde{\boldsymbol{c}}_t) \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \tanh(\boldsymbol{c}_t)
\end{aligned} \tag{2.5}$$

where $\boldsymbol{H_x}, \boldsymbol{W_x}, \boldsymbol{A_x}$, and $\boldsymbol{b}_x$ denote the trainable weights and biases. $\hat{\boldsymbol{a}}_t$ is calculated using:

$$\hat{\boldsymbol{a}}_t = \sum_{1 \leq j \leq K} e'_{j,t}\boldsymbol{a}_j \tag{2.6}$$

where $e'_{j,t}$ are our attention weights normalized using a softmax over the output (e_t) of our attention module:

$$\begin{aligned}
\boldsymbol{e}'_t &= \text{softmax}(\boldsymbol{e}_t) \\
e_{j,t} &= \boldsymbol{W}_e^T \tanh(\boldsymbol{W}'_a\boldsymbol{a}_j + \boldsymbol{W}'_h\boldsymbol{h}_t)
\end{aligned} \tag{2.7}$$

$\boldsymbol{W}_e^T$ and $\boldsymbol{W}'_x$ are the trainable weights of the attention module. Rennie *et al.* [20] extended the work of Xu *et al.* [7] by applying the CIDEr metric [100], a standard performance metric for image captioning, to optimize their caption generator compared to using maximum likelihood estimation loss (Equation 2.3). Their approach was inspired by a Reinforcement Learning approach [101, 102] called self-critical sequence training, which involves normalizing the reward signals calculated using the CIDEr metric.

2.3.2 Top-Down and Bottom-Up Image Captioning

Yu *et al.* [103] and You *et al.* [19] applied a notion of *semantic* attention to detected visual attributes, learned in an end-to-end fashion, where bottom-up approaches were combined with top-down approaches to take advantage of both paradigms. For instance, they acquired a list of semantic concepts or attributes, regarded as a bottom-up mechanism, and used the list with visual features, as an instance of top-down information, to generate an image caption. Semantic attention is used to attend to semantic concepts detected from various parts of a given image. Here, the visual content was only used in the initial time step similar to Vinyals *et al.* [1]. In other time steps, semantic attention was used to select the extracted semantic concepts. That is, semantic attention differs from spatial attention, which attends to spatial features in every time step, and does not preserve the spatial information of the detected concepts.

To preserve spatial information, salient regions can be localized using spatial transformer networks [104], which get the spatial features as inputs. This is similar to Faster R-CNN's generation of bounding boxes [60], but it is trained in an end-to-end fashion using bilinear interpolation instead of a Region of Interest pooling mechanism [89]. Drawing on this idea, Anderson *et al.* [8] applied a pre-trained Faster R-CNN and an attention mechanism to discriminate among different visual-based regions regarding the spatial features. Specifically, they combined bottom-up and top-down approaches where a pre-trained Faster R-CNN is used to extract the salient regions from images, instead of using the detected objects as high-level semantic concepts in the work of You *et al.* [19]; and an attention mechanism is used to generate spatial attention weights over the convolutional feature maps representing the regions. Faster R-CNN, as an object detection model, is pre-trained on the Visual Genome dataset [33]; this pre-training on a large dataset is analogous to pre-training a classification model on the ImageNet dataset [105]. Jin *et al.* [106] previously used salient regions with different scales which are extracted by applying selective search [107] instead of applying Faster R-CNN. Then, they made the inputs of their spatial attention mechanism by resizing and encoding the regions in the task of image captioning.

The bottom-up and top-down system of Anderson *et al.* [8] first uses the pre-trained Faster R-CNN to extract bounding boxes including objects, as a bottom-up approach, from images and then uses a top-down mechanism to attend to the bounding boxes to generate their corresponding words and phrases (Figure 2.6). The top-down mechanism includes two LSTMs. The first LSTM is to calculate attention weights which is defined as (a short form of Equation 2.5):

$$\boldsymbol{h}_{t,a} = \mathrm{LSTM}(\boldsymbol{h}_{t,a-1}, [\bar{\boldsymbol{a}}, \boldsymbol{h}_{t,l-1}, \boldsymbol{w}_{t-1}]) \tag{2.8}$$

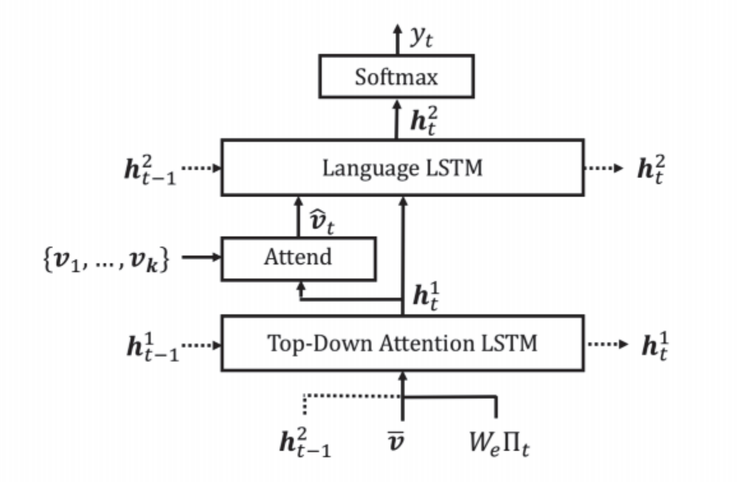


Figure 2.6: The bottom-up and top-down model including two LSTMs to attend to bottom-up features $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ obtained from Faster R-CNN [8].

where $\bar{\boldsymbol{a}} = \frac{1}{K} \sum_{1 \le i \le K} \boldsymbol{a}_i$ is calculated as the mean-pooled visual features and $\boldsymbol{h}_{t,a}$ is the current hidden state of the LSTM. $\boldsymbol{h}_{t,a}$ is used to calculate attention weights as in Equation 2.7. $\boldsymbol{h}_{t,l-1}$ is the previous hidden state of the second LSTM which plays the role of the language model:

$$\boldsymbol{h}_{t,l} = \text{LSTM}(\boldsymbol{h}_{t,l-1}, [\hat{\boldsymbol{a}}_t, \boldsymbol{h}_{t,a}]) \tag{2.9}$$

$\boldsymbol{h}_{t,l}$ is used to calculate the probability distribution of the next generated word as in Equation 2.4 and the objective function as in Equation 2.3.

In our image captioning systems proposed in the next chapters, we use different spatial attention mechanisms weighting the convolutional features representing salient regions of images. This allows our image captioning models to generate captions which are highly correlated with visual content. In Chapter 4, we propose novel attention-based image captioning models to apply the recognized emotions from visual content to enrich image captions. In Chapters 5 and 6, we propose novel attention-based image captioning models to generate more correlated and more diverse sentiment-bearing content. These are from the literature of image captioning with style explained in the next section. In Chapter 7, we propose an adversarial attack against a version of Faster R-CNN used in the state-of-the-art image captioning model [8] and show its impact on the generated captions.

2.4 Style-Bearing Image Caption Generation

2.4.1 Controlled Image Captioning with Style

Most image captioning systems concentrate on describing visual content without adding any extra information, giving rise to factual linguistic descriptions. However, there are also stylistic aspects of language which play an essential role in enriching written communication and engaging users during interactions. Style helps in clearly conveying visual content [49], and making the content more attractive [30, 31]. It also conveys personality-based [50] and emotion-based attributes which can impact on decision making [2]. Incorporating style into the description of an image is effective in boosting the engagement level of humans in dealing with automatically-generated comments for photos and videos in social media platforms [23] and helping chatbot platforms to interact like humans [32].

There are a few models that incorporated style or other non-factual characteristics into the generated captions. In addition to describing the visual content, these models learn to generate different forms or styles of captions. For instance, StyleNet proposed by Gan *et al.* [30] used a novel type of LSTM called Factored-LSTM to transfer factual and style-based information from the input caption. Factored-LSTM includes three matrices which are all trained on a factual dataset to transfer the factual content of the caption, but only one of them is trained on a stylized dataset to transfer the style of the caption. Factored-LSTM is inspired from the Show-Tell image captioning model [1] (Figure 2.4) which uses visual content as an initial state of a caption generation process. It is defined as:

$$\begin{aligned}
\boldsymbol{i}_t &= \sigma(\boldsymbol{H}_i\boldsymbol{h}_{t-1} + \boldsymbol{A}_{ia}\boldsymbol{N}_{ia}\boldsymbol{Z}_{ia}\boldsymbol{w}_{t-1} + \boldsymbol{b}_i) \\
\boldsymbol{f}_t &= \sigma(\boldsymbol{H}_f\boldsymbol{h}_{t-1} + \boldsymbol{A}_{fa}\boldsymbol{N}_{fa}\boldsymbol{Z}_{fa}\boldsymbol{w}_{t-1} + \boldsymbol{b}_f) \\
\boldsymbol{o}_t &= \sigma(\boldsymbol{H}_o\boldsymbol{h}_{t-1} + \boldsymbol{A}_{oa}\boldsymbol{N}_{oa}\boldsymbol{Z}_{oa}\boldsymbol{w}_{t-1} + \boldsymbol{b}_o) \\
\tilde{\boldsymbol{c}}_t &= \boldsymbol{H}_g\boldsymbol{h}_{t-1} + \boldsymbol{A}_{ga}\boldsymbol{N}_{ga}\boldsymbol{Z}_{ga}\boldsymbol{w}_{t-1} + \boldsymbol{b}_g \\
\boldsymbol{c}_t &= \boldsymbol{f}_t\boldsymbol{c}_{t-1} + \boldsymbol{i}_t \tanh(\tilde{\boldsymbol{c}}_t) \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \tanh(\boldsymbol{c}_t)
\end{aligned} \tag{2.10}$$

The key components of Factored-LSTM are the three different kinds of trainable matrices linked to the input word: $\boldsymbol{A}_{xa} \in \mathbb{R}^{U\times S}$, $\boldsymbol{N}_{xa} \in \mathbb{R}^{S\times S}$, and $\boldsymbol{Z}_{xa} \in \mathbb{R}^{S\times M}$, where U is the size of the hidden state. $\boldsymbol{N}_{xa}$ is the stylistic matrix with S dimensions for adjusting the targeted style. Factored-LSTM aims to optimize two different tasks. First, it learns to describe visual content using a factual dataset. All trainable parameters in Equation 2.10 are updated at this stage. Second, it learns to transfer style using a stylistic dataset. Here, it only updates the stylistic parameters ($\boldsymbol{N}_{xa}$) in Equation 2.10. In both stages, it is given the ground truth captions as well as the visual features. Style here refers to non-grounded information added to factual captions which can be *romantic* or *humorous*. StyleNet particularly uses a mechanism to transfer style using different matrices to record the factual and stylistic information, which is inspired from Anne *et al.* [27] transferring knowledge belonging to seen objects to describe unseen objects; and multi-task training [108], using an external dataset belonging to factual image captioning for stylistic image captioning. However, in the second stage of the training, since it learns only from the stylistic image captioning dataset, it focuses mainly on the stylistic aspect for generating image captions. This can lead to ignoring visual content since there is no mechanism to regulate the application of the stylistic and factual information.

To deal with this issue, Mathews *et al.* [2] proposed the SentiCap system which is able to differentiate between the factual and stylistic information by training two different LSTMs. The system is trained on the SentiCap dataset including two sets of stylistic image captions: positive and negative sentiment-bearing captions. Here, the notion of sentiment is drawn from Natural Language Processing [109], with sentiment either *negative* or *positive*. The SentiCap system is a full switching architecture incorporating both factual and sentiment-bearing caption paths using two LSTMs, which are acting in parallel, to play the role of the caption generator in this work. One LSTM aims to capture the factual aspect and the other one aims to capture the sentiment-bearing aspect of the generated caption. During the caption generation process, SentiCap weights the generated probability distributions of words using the LSTMs to generate image captions. It does this by predicting the sentiment levels of the words, learned from the ground-truth labels describing the sentiment levels of different words in the stylistic dataset. However, this extra word-level ground truth labels for style are not available for all style-bearing datasets such as the dataset used by Gan *et al.* [30].

Recently, Chen *et al.* [31] applied an attention mechanism to weight the stylistic and the factual information in Factored-LSTM to consider this. The attention mechanism controls between applying semantic and style-based information from the input caption. Using this work, the new version of Factored-LSTM is defined as:

$$
\begin{aligned}
\boldsymbol{i}_t &= \sigma((g_{ht}\boldsymbol{S}_{hi} + (1-g_{ht})\boldsymbol{H}_i)\boldsymbol{h}_{t-1} + ((g_{wt}\boldsymbol{S}_{wi} + (1-g_{wt})\boldsymbol{W}_i)\boldsymbol{w}_{t-1} + \boldsymbol{b}_i) \\
\boldsymbol{f}_t &= \sigma((g_{ht}\boldsymbol{S}_{hf} + (1-g_{ht})\boldsymbol{H}_f)\boldsymbol{h}_{t-1} + ((g_{wt}\boldsymbol{S}_{wf} + (1-g_{wt})\boldsymbol{W}_f)\boldsymbol{w}_{t-1} + \boldsymbol{b}_f) \\
\boldsymbol{o}_t &= \sigma((g_{ht}\boldsymbol{S}_{ho} + (1-g_{ht})\boldsymbol{H}_o)\boldsymbol{h}_{t-1} + ((g_{wt}\boldsymbol{S}_{wo} + (1-g_{wt})\boldsymbol{W}_o)\boldsymbol{w}_{t-1} + \boldsymbol{b}_o) \\
\tilde{\boldsymbol{c}}_t &= (g_{ht}\boldsymbol{S}_{hg} + (1-g_{ht})\boldsymbol{H}_g)\boldsymbol{h}_{t-1} + ((g_{wt}\boldsymbol{S}_{wg} + (1-g_{wt})\boldsymbol{W}_g)\boldsymbol{w}_{t-1} + \boldsymbol{b}_g \\
\boldsymbol{c}_t &= \boldsymbol{f}_t\boldsymbol{c}_{t-1} + \boldsymbol{i}_t \tanh(\tilde{\boldsymbol{c}}_t) \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \tanh(\boldsymbol{c}_t)
\end{aligned}
\tag{2.11}
$$

where $\boldsymbol{S}_x$ is a matrix aiming to record the stylistic information and $\boldsymbol{H}_x$ and $\boldsymbol{W}_x$ are matrices to record the factual information. Similar to the original Factored LSTM (Equation 2.10), only matrices related to style ($\boldsymbol{S}_x$) are updating in the second stage of training. g_{ht} and g_{wt} are learned using the attention mechanism to differentiate between the stylistic and the factual information during the caption generation process. They are the main differences between this version of Factored-LSTM and the original Factored-LSTM [30]. However, all these approaches need two-stage training: training on factual image captions and training on style-bearing image captions. Therefore, they do not support an end-to-end training.

To address this issue, You *et al.* [48] designed two new schemes, Direct Inject and Sentiment Flow, to better employ sentiment in generating image captions. For Direct Inject, an additional dimension was added to the input of an LSTM to express sentiment and this sentiment unit is injected at every time step of the generation process. The Sentiment Flow approach of You *et al.* [48] injects the sentiment unit only at the initial time step of a designated sentiment cell trained in a similar learning fashion to the memory cell in LSTMs. This work is inspired by Radford *et al.* [110] who identified a sentiment unit in RNN-based systems. Radford *et al.* [110] showed that the sentiment unit directly impacts on the generation process so that specifying the sentiment unit with different values leads to outputs with different sentiments. By comparing with larger encoding vectors for sentiment, they also showed that most of the information that the generative model needs to provide sentiment-bearing content is in the sentiment unit.

As mentioned, differentiating between stylistic and factual information is a challenging task and current image captioning models with style are usually trained using a small amount of image caption paired data with style. These lead to stylized image captions but at the cost of less correlated visually-grounded captions. Both transferring features, such as positive or negative sentiments [2] and hilarious or romantic styles [30], and preserving the semantic content at the same time are difficult because they require learning disentangled representations, which is challenging even with a large amount of paired data. Moreover, collecting a large number of paired images with stylistic captions is very costly [49]. Recently, Mathews *et al.* [49] proposed an image captioning model which is able to learn from a large amount of unpaired data and keep the relevance between images and their corresponding captions. The basis of this model is a semantic representation dividing the semantic and stylistic aspects of image captions. However, evaluating the model is more difficult because there is no image caption paired data to measure the relevance between images and captions.

In Chapters 5 and 6, we propose image captioning models with style to employ sentiment-bearing content in generating image captions. To keep the connection between images and captions and generate stylized image captions, our models consist of attention-based architectures, attending to visual content not stylistic and factual information like Chen *et al.* [31]. Our attention mechanisms learn the correlation between different generated words and different regions in images to preserve the factual information while adding the stylistic information. We apply different training mechanisms including an adversarial training mechanism in Chapter 5, inspired by generative adversarial networks which we explain in §2.5.2, and an end-to-end training mechanism combined with an embedding approach to capture high-level and word-level sentiment information in Chapter 6, inspired by the literature of controlled natural language generation which we explained in the next section.

2.4.2 Controlled Natural Language Generation

While not specific to image captioning, an area of direct relevance to §2.4.1 is natural language generation (NLG), which aims to generate understandable texts in different languages by combining computational linguistics and artificial intelligence techniques [111]. A recent survey is given by Gatt *et al.* [112]. In this section, we only focus on one particular aspect that

Hu *et al.* [35] refer to as controlled generation of text: generating texts that are controllable for particular attributes.

In recent years, researchers in the NLG domain have tried to make a control over the generated text in terms of different attributes. In some applications, the purpose is to generate a target sentence having a specific attribute with similar content compared to its corresponding source sentence. Hu *et al.* [35] used Variational Autoencoders (VAEs) to control a generated sentence in terms of its attributes including sentiment and tense: they conditioned the sentence encoded space on these attributes. VAEs [113] include an encoder encoding the input sample into the latent variables and a decoder generating the samples from the variables, differing from standard autoencoders by using KL divergence loss. The loss matches the prior and posterior of the variables to make generating an acceptable sentence from the prior variables possible. Here, the attributes are included into the latent variables produced by the encoder. In this kind of task, a type of RNNs, usually LSTM networks, plays the role of both the encoder and the decoder. In the conversation generation task, Zhou *et al.* [51] used emotion categories to control the responses in terms of emotional values. As a part of their system, they fed an embedded emotion category as an input to their decoder. They also designed internal and external memories to record the emotion dynamics and distinguish emotional words versus other words, respectively. The targeted emotion is erased in the internal memory in each time step until the sentence is generated completely. The external memory learns to capture the emotional values of different words in the vocabulary during the training phase to assist the generator to assign more probabilities to the words from the targeted emotion. Ghosh *et al.* [52] proposed a model conditioning a conversational text generation module on emotions. The model can control a generated sentence without previous knowledge about the words' polarities in the existing vocabulary. It is also able to generate emotional sentences with a customized amount of emotional content. To do so, it learns an embedding space for the emotional values of different words during the training phase. Then, during the prediction phase, these embedding values multiplied to an emotion strength parameter deciding the impact of emotional information on the next generated word.

Our image captioning models aim to have controls on image caption generation using extra knowledge such as encoded emotional information detected from visual content (Chapter

4) and encoded sentiment-bearing information incorporated from text (Chapters 5 and 6). In Chapter 6, we embed sentiment-bearing information: high-level embedding captures the overall sentiment-bearing value of the generated caption and word-level embedding captures the sentiment-bearing value linked to each generated word.

2.5 Other Required Technical Background

In this section, we explain other related work outside of image captioning that is required for the next chapters.

2.5.1 Facial Expression Recognition

As one of our goals, we aim to implement a model for facial expression recognition from images to incorporate in our proposed image captioning systems. Facial expression is a form of non-verbal communication conveying attitudes, affects, and intentions of individuals. It happens as the result of changes over time in facial features and muscles [114]. It is also one of the most important communication means for showing emotions and transferring attitudes in human interactions. Indeed, research on facial expressions started more than a century ago when Darwin published his book titled, “The expression of the emotions in man and animals” [115]. Since then a large body of work has emerged on recognizing facial expressions, usually using a purportedly universal framework of a small number of standard emotions (*happiness*, *sadness*, *fear*, *surprise*, *anger*, and *disgust*) or this set including a *neutral* expression [114, 116–120] or more fine-grained facial features such as facial action units, defined as the deformations of facial muscles [121]. Recently, recognizing facial expressions has been paid special attention because of its practical applications in different domains such as education, health-care and virtual reality [114, 122]. It is worth mentioning that the automatic recognition of facial expressions is a difficult task because different people express their attitudes in different ways and there are close similarities among various types of facial expressions [123], as shown in Figure 2.7.

Deep Learning-Based Facial Expression Recognition To find effective representations, deep learning-based methods have been recently successful in this domain. Due to their

igure has been suppressed due to copyright reasons

Figure 2.7: Examples from the Facial Expression Recognition 2013 dataset [9] including seven standard facial expressions.

complex architectures including multiple layers, they can capture hierarchical structures from low- to high-level representations of facial expression data. Tang [124], the winner of the 2013 Facial Expression Recognition (FER) challenge [9], trained a CNN with a linear support vector machine (SVM) to detect facial expressions. He replaced the softmax layer of the CNN with a linear SVM and showed a consistent improvement compared to the previous work. Instead of cross-entropy loss, his approach optimizes a margin-based loss to maximize margins among data points belonging to diverse classes.

CNNs are also used for feature extraction and transfer learning in this domain. Kahou *et al.* [125] applied a CNN model to recognize facial expressions and won the 2013 Emotion Recognition in the Wild (EmotiW) Challenge. Their approach uses a combination of deep neural networks to learn from diverse data modalities including video frames, audio data and spatio-temporal information [126]. The CNN model, as the best model in this work, aims to recognize emotions from static video frames. Then, the recognized emotions are combined across a video clip by a frame aggregation technique and classified using an SVM with a radial basis kernel function. Yu *et al.* [127] used an ensemble of CNNs to detect facial expressions in a transfer learning framework. On their target samples, they applied a set of face detection approaches to optimally detect faces and remove irrelevant data. They used a multiple neural network training framework to learn a set of weights assigned to the responses of the CNNs in addition to averaging and voting over the responses. Kim *et al.* [128] combined aligned and non-aligned faces to enhance the recognition performance of facial expressions where they automatically detected facial landmarks from faces to rotate and align faces. Then, they trained a CNN model using this combination of faces. Zhang *et al.* [129] proposed a CNN-based method to recognize social relation traits (e.g. friendly, competitive and dominant) from detected faces in an image. The method includes a CNN model to recognize facial expressions projected into a shared representation space. The space combines the extracted features from two detected faces in an image and generates the predictions of social traits.

The models mentioned above usually use conventional CNN architectures to report the performance on different facial expression recognition datasets including the FER-2013 dataset [9], which is a publicly available dataset with a large number of human faces collected in the wild condition. Pramerdorfer *et al.* [130] instead used an ensemble of very deep architectures of CNNs such as VGGnet, Inception and ResNet by identifying the bottlenecks of the previous state-of-the-art facial expression recognition models on the FER-2013 dataset and achieving a new state-of-the-art result on the dataset.

The quality of these recent models is high: it is at least as good as human performance [9]. Moreover, the idea of applying VGGnet in facial expression recognition tasks motivates our work to make a facial expression recognition model reproducing the state-of-the-art result on FER-2013 dataset. We aim to check the ability of the model to learn representations that can be useful in other tasks such as engagement recognition in Chapter 3. We use the model to extract facial features from human faces to apply in our image captioning models in Chapter 4.

2.5.2 Generative Adversarial Networks

Goodfellow *et al.* [131] introduced Generative Adversarial Networks (GANs), whose training mechanism consists of a generator and a discriminator; they have been applied with great success in different applications [132–135]. The discriminator is trained to recognize real and synthesized samples generated by the generator. In contrast, the generator wants to generate realistic data to mislead the discriminator in distinguishing the source of data. GANs are successfully used to learn from data in computer vision recently [132, 136]. However, they were originally established for a continuous data space [131, 133] rather than a discrete data distribution such as a text generation task. A major reason is that the task makes the generator's optimization difficult because of its non-differentiable nature in generating (discrete) sequences of words.

To handle this, a form of reinforcement learning is usually applied, where the sentence generation process is formulated as a reinforcement learning problem [137]; the discriminator provides a reward for the next action (in our context the next generated word), and the generator uses the reward to calculate gradients and update its parameters, as proposed by Yu *et al.* [133]. They also applied Monte Carlo search to complete a partially generated sentence to generate intermediate rewards where the discriminator can only evaluate a complete generated sentence. The most popular objective function to train a sequence generator is the maximum likelihood estimation (MLE) mechanism which suffers from a gap between teacher-forcing in the training phase and self-feeding in the testing phase. This is called exposure bias. The generator usually uses MLE to calculate the likelihood of the current generated word with respect to the ground-truth sequence so far during training; however, it can only use its previous generated words during testing which leads to this gap. In this domain, using the rewards received from the discriminator to incorporate an additional term, as a regularization term, to the MLE optimization can reduce this gap. In addition to GANs, Shen *et al.* [138] employed the Professor-Forcing algorithm to handle this gap [139] where the sequence of a RNN's hidden states, which includes the information of output words, is matched as an alternative to the sequence of words and distributed more smoothly.

Liang *et al.* [140] applied a GAN framework to generate paragraphs describing visual content, where their discriminators (one for distinguishing between generated sentences and another one for distinguishing between generated paragraphs) are trained to recognize real paragraphs from synthesized ones and their generator seeks the generation of realistic and varied paragraphs to mislead the discriminators. In this work, the discriminators are based on the Wasserstein GAN (WGAN) [141], an improved version of earlier GANs, and the generator is inspired from the work of Xu *et al.* [7] explained in §2.3.1. WGAN provides continuous outputs to generate meaningful gradients and prevent vanishing gradients in comparison with the traditional GANs providing non-continuous outputs.

Here, as a text discriminator, a GAN typically builds a classifier model aiming to distinguish between the generated texts by machines and humans:

$$L_D(\boldsymbol{\phi}) = [\mathbb{E}_{x\sim\mathbb{P}_H}(\log D_\phi(\boldsymbol{x})) + \mathbb{E}_{\bar{\mathrm{x}}\sim\mathbb{P}_G}(\log(1 - D_\phi(\bar{\mathbf{x}})))] \tag{2.12}$$

where $\boldsymbol{\phi}$ are the parameters of the discriminator D_ϕ; $\mathbb{P}_H$ is a collection of the generated text by humans; and $\mathbb{P}_G$ is a collection of the generated text by the generator. The human generated text is given by $\boldsymbol{x}$ and the machine generated text is given by $\bar{\boldsymbol{x}}$. WGAN, instead, aims to optimize a different objective function:

$$L_D(\boldsymbol{\phi}) = \mathbb{E}_{x\sim\mathbb{P}_H}[D_\phi(\boldsymbol{x})] - \mathbb{E}_{\bar{\mathrm{x}}\sim\mathbb{P}_G}[D_\phi(\bar{\mathbf{x}})] \tag{2.13}$$

In addition to the supervised setting, Liang *et al.* [140] applied a semi-supervised setting where their paragraph generator only uses a single sentence with annotation and generates the reset of the paragraph using the discriminators, showing a considerable improvement. Later, Wang and Wan [135] applied a similar framework to generate sentiment-bearing text (although not conditioned on any input, such as the images in our captioning task). They trained several generators to generate sentences with different sentiment values and a discriminator distinguishing among real sequences with different sentiments, and fake sequences. The discriminator is trained for a multiclass classification generating a probability distribution over k real sentiment-bearing labels and one fake label for generated sentences. The discriminator provides reward signals using WGAN.

In these kinds of work, the discriminator, which usually learns by a combination of real and generated data, specifies the score of a generated sequence and is different from the work of Bahdanau *et al.* [142] and Rennie *et al.* [20] requiring task-related evaluation metrics such as BLEU and CIDEr, respectively. In image captioning, Luo *et al.* [143] used a combination of CIDEr and a score generated by a pre-trained retrieval model, evaluating the match between the generated caption and its corresponding image, to calculate the loss value and optimize their image captioning model. To handle the non-differentiable value of the combination reward, they employed a reinforcement learning algorithm with baseline inspired from Rennie *et al.* [20]. Here, the baseline is used to reduce the variance of the calculated gradients as in the work of Ranzato *et al.* [144]. Using a pre-trained retrieval model, the trained image captioning model can be optimized by a specific objective function discriminating images and their corresponding captions rather than depending on only human generated captions as in other image captioning systems using GANs (e.g. [145]).

Our work in Chapter 5 uses a GAN framework, for the first time applying it to image captioning with style, where the style is independent from the content of images.

2.6 Summary

In this chapter, we have discussed key technical background required for the thesis. After giving a high-level introduction to deep learning approaches in §2.1, we provided an overview over image captioning models in §2.2, as the central application of this thesis, followed by attention-based image captioning, as a common part among Chapters 4, 5, 6 and 7, and style-bearing image caption generation, as a common part between Chapters 5 and 6. Lastly, we described other required technical background in §2.5 such as facial expression recondition, used in Chapters 3 and 4, and generative adversarial networks, used in Chapter 5.

Part I

Image Captioning using Emotional Content

3

Automatic Recognition of Student Engagement using Deep Learning and Facial Expression

Engagement is a key indicator of the quality of learning experience, and one that plays a major role in developing intelligent educational interfaces. Any such interface requires the ability to recognise the level of engagement in order to respond appropriately; however, there is very little existing data to learn from, and new data is expensive and difficult to acquire. What we explore here, which previous work on automatically recognising engagement has not, is the idea that representations from other facial expression recognition tasks might usefully generalise. This chapter presents a deep learning model to improve engagement recognition from images that overcomes the data sparsity challenge by pre-training on readily available basic facial expression data, before training on specialised engagement data. In the first of two steps, a facial expression recognition model is trained to provide a rich face representation using deep learning. In the second step, we use the model's weights to initialize a deep learning-based model to recognize engagement; we term this the engagement model. We train the model on our new engagement recognition dataset with 4627 engaged and disengaged samples. We find that the engagement model outperforms effective deep learning architectures that we apply for the first time to engagement recognition, as well as approaches using histogram of oriented gradients and support vector machines. This confirms the effectiveness of applying facial expression recognition features for recognizing engagement.[1] In the following chapter, we aim to use facial expression features for image captioning, where we incorporate the features to improve the generated captions. We will propose different image captioning models to effectively apply the features to generate more descriptive image captions.

3.1 Introduction

Engagement is a significant aspect of human-technology interactions and is defined differently for a variety of applications such as search engines, online gaming platforms, and mobile health applications [47]. According to Monkaresi *et al.* [45], most definitions describe engagement as attentional and emotional involvement in a task.

This chapter deals with engagement during learning via technology. Investigating engagement is vital for designing intelligent educational interfaces in different learning settings including educational games [11], massively open online courses (MOOCs) [43], and intelligent tutoring systems (ITSs) [44]. For instance, if students feel frustrated and become disengaged (see disengaged samples in Figure 3.1), the system should intervene in order to

[1] The content of this chapter is based on the following publication:

Omid Mohamad Nezami, Mark Dras, Len Hamey, Deborah Richards, Stephen Wan, Cecile Paris (2019). Automatic Recognition of Student Engagement using Deep Learning and Facial Expression. *Proceedings of the 2019 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2019),* Wuerzburg, Germany.

igure has been suppressed due to copyright reasons

Figure 3.1: Engaged (left) and disengaged (right) samples collected in our studies. We blurred the children's eyes for ethical issues, even though we have their parents consent at the time.

bring them back to the learning process. However, if students are engaged and enjoying their tasks (see engaged samples in Figure 3.1), they should not be interrupted even if they are making some mistakes [146]. In order for the learning system to adapt the learning setting and provide proper responses to students, we first need to automatically measure engagement. This can be done by, for example, using context performance [44], facial expression [147] and heart rate [45] data. Recently, engagement recognition using facial expression data has attracted special attention because of widespread availability of cameras [45].

This chapter aims at quantifying and characterizing engagement using facial expressions extracted from images. In this domain, engagement detection models usually use typical features which are designed for general purposes, such as Gabor features [147], histogram of oriented gradients [43] and facial action units [148]. To the best of our knowledge, there is no work in the literature investigating the design of specific and high-level features for engagement. Therefore, providing a rich engagement representation model to distinguish engaged and disengaged samples remains an open problem (Challenge 1). Training such a rich model requires a large amount of data which means extensive effort, time, and expense would be required for collecting and annotating data due to the complexities [46] and ambiguities [47] of the engagement concept (Challenge 2).

To address the aforementioned challenges, we design a deep learning model which includes two essential steps: basic facial expression recognition, and engagement recognition. In the first step, a convolutional neural network (CNN) is trained on the dataset of the Facial Expression Recognition Challenge 2013 (FER-2013) to provide a rich facial representation model, achieving the state-of-the-art performance. In the next step, the model is applied to initialize our engagement recognition model, designed using a separate CNN, learned on our newly collected dataset in the engagement recognition domain. As a solution to Challenge 1, we train a deep learning-based model that provides the representation model specifically for engagement recognition. As a solution to Challenge 2, we use the FER-2013 dataset, which is around eight times larger than our collected dataset, as external data to pre-train the engagement recognition model and compensate for the shortage of engagement data.

3.2 Related Work

Engagement has been detected in three different time scales: the entire video of a learning session, 10-second video clips, and images. In the first category, Grafsgaard *et al.* [149] studied the relation between facial action units (AUs) and engagement in learning contexts. They collected videos of web-based sessions between students and tutors. After finishing the sessions, they requested each student to fill out an engagement survey used to annotate the student's engagement level. Then, they used linear regression methods to find the relationship between different levels of engagement and different AUs. However, their approach does not characterize engagement in fine-grained time intervals which are required for making an adaptive educational interface.

As an attempt to solve this issue, Whitehill *et al.* [147] applied linear support vector machines (SVMs) and Gabor features, as the best approach in this work, to classify four engagement levels: not engaged at all, nominally engaged, engaged in task, and very engaged. In this work, the dataset includes 10-second videos annotated into the four levels of engagement by observers, who analyzed the videos. Monkaresi *et al.* [45] used heart rate features in addition to facial features to detect engagement. They used a face tracking engine to extract facial features and WEKA (a classification toolbox) to classify the features into engaged or not engaged classes. They annotated their dataset, including 10-second videos, using self-reported data collected from students during and after their tasks. Bosch *et al.* [148] detected engagement using AUs and Bayesian classifiers. The generalizability of the model was also investigated across different times, days, ethnicities and genders [150]. Furthermore, in interacting with intelligent tutoring systems (ITSs), engagement was investigated based on a personalized model including appearance and context features [44]. Engagement was considered in learning with massively open online courses (MOOCs) as an e-learning environment [151]. In such settings, data are usually annotated by observing video clips or filling self-reports. However, the engagement levels of students can change during 10-second video clips, so assigning a single label to each clip is difficult and sometimes inaccurate.

In the third category, HOG features and SVMs have been applied to classify images using three levels of engagement: not engaged, nominally engaged and very engaged [43]. This work is based on the experimental results of Whitehill *et al.* [147], who showed that engagement patterns are mostly recorded in images. Bosch *et al.* [148] also confirmed that video clips can not provide extra information by reporting similar performances using different lengths of video clips in detecting engagement. However, competitive performances are not reported in this category.

We focus on the third category to recognize engagement from images. To do so, we collected a new dataset annotated by Psychology students, who can potentially better recognize the psychological phenomena of engagement, because of the complexity of analyzing student engagement. To assist them with recognition, brief training was provided prior to commencing the task and delivered in a consistent manner via online examples and descriptions. We did not use crowdsourced labels, as in the work of Kamath *et al.* [43], as it resulted in low quality annotation and poor model performance. Furthermore, we captured more effective labels by following an annotation process to simplify the engagement concept into the behavioral and the emotional dimensions, in line with the domain literature. We requested annotators to label the dimensions for each image and make the overall annotation label by combining these. Our aim is for this dataset to be useful to other researchers interested in detecting engagement from images. Given this dataset, we introduce a novel model to recognize engagement using deep learning. The model includes two important phases. First, we train a deep model to recognize basic facial expressions. Second, the model is applied to initialize the weights of our engagement recognition model trained using our newly collected dataset.

3.3 Facial Expression Recognition from Images

3.3.1 Facial Expression Recognition Dataset

In this section, we use the facial expression recognition 2013 (FER-2013) dataset [9]. The dataset includes images, labeled *happiness*, *anger*, *sadness*, *surprise*, *fear*, *disgust*, and *neutral*. It contains 35,887 samples (28,709 for the training set, 3589 for the public test set and 3589 for the private test set), collected by the Google search API. The samples are in grayscale at the size of 48-by-48 pixels.

We split the training set into two parts after removing 11 completely black samples: 3589 for validating and 25,109 for training our facial expression recognition model. To compare with related work [127, 128, 130], we do not use the public test set for training or validation, but use the private test set for performance evaluation of our facial expression recognition model.

3.3.2 Facial Expression Recognition using Deep Learning

We train the VGG-B model [10], using the FER-2013 dataset, with one less Convolutional (Conv.) block as shown in Figure 3.2. This results in eight Conv. and three fully connected layers. We also have a max pooling layer after each Conv. block with stride 2. We normalize each FER-2013 image so that the image has a mean 0.0 and a norm 100.0 [124]. Moreover, for each pixel position, the pixel value is normalized to mean 0.0 and standard-deviation 1.0

Image has been suppressed due to copyright reasons

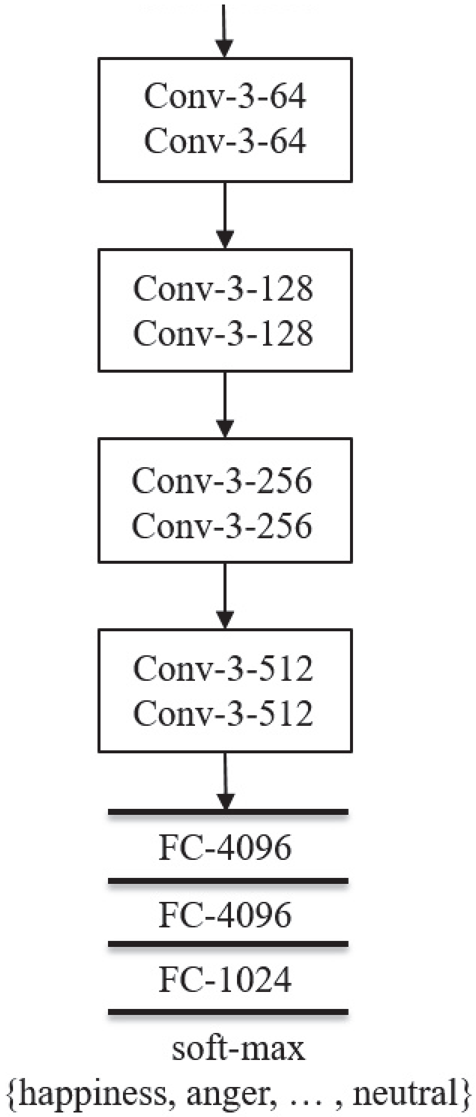


Figure 3.2: The architecture of our facial expression recognition model adapted from VGG-B framework [10]. Each rectangle is a Conv. block including two Conv. layers. The max pooling layers are not shown for simplicity.

using our training part. The model has a similar performance to the work of Pramerdorfer *et al.* [130] generating the state-of-the-art on the FER-2013 dataset. The model's output layer has a softmax function generating the categorical distribution probabilities over seven facial expression classes in FER-2013. We aim to use this model as a part of our engagement

igure has been suppressed due to copyright reasons

Figure 3.3: The interactions of a student with Omosa [11], captured in our studies. On the left side, the Omosa environment is shown where students can fill in a report or interact with virtual agents.

recognition model.

3.4 Engagement Recognition from Images

3.4.1 Engagement Recognition Dataset

Data Collection To recognize engagement from face images, we construct a new dataset that we call the Engagement Recognition (ER) dataset. The data samples are extracted from videos of students, who are learning scientific knowledge and research skills using a virtual world named Omosa [11]. Samples are taken at a fixed rate instead of random selections, making the dataset samples representative, spread across both subjects and time. In the interaction with Omosa, the goal of students is to determine why a certain animal kind is dying out by talking to characters, observing the animals and collecting relevant information, Figure 3.3 (top). After collecting notes and evidence, students are required to complete a workbook, Figure 3.3 (bottom).

igure has been suppressed due to copyright reasons

Figure 3.4: Examples without detectable faces because of high face occlusions.

The videos of students were captured from our studies in two public secondary schools involving twenty students (11 girls and 9 boys) from Years 9 and 10 (aged 14–16), whose parents agreed to their participation in our ethics-approved studies. We collected the videos from twenty individual sessions of students recorded at 20 frames per second (fps), resulting in twenty videos and totalling around 20 hours. After extracting video samples, we applied a convolutional neural network (CNN) based face detection algorithm [152] to select samples including detectable faces. The face detection algorithm cannot detect faces in a small number of samples (less than 1%) due to their high face occlusion (Figure 3.4). We removed the occluded samples from the ER dataset.

Data Annotation We designed custom annotation software to request annotators to independently label 100 samples each. The samples are randomly selected from our collected data and are displayed in different orders for different annotators. Each sample is annotated by at least six annotators.[2] Following ethics approval, we recruited Psychology students to undertake the annotation task, who received course credit for their participation. Before starting the annotation process, annotators were provided with definitions of behavioral and emotional dimensions of engagement, which are defined in the following paragraphs, inspired

[2] The Fleiss' kappa of the six annotators is 0.59, indicating reasonable inter-coder agreement. (The agreement of labelling the emotional dimension is lower than the behavioural dimension which can be because of the higher level of subjectivity in the emotional one.) We also calculated the correction or accuracy of each annotator versus the rest of annotators. The average value is 74.58% showing an approximate level of human performance to label our samples. This performance is not very high and it shows that engagement recognition is even a challenging task for humans.

Table 3.1: The adapted relationship between the behavioral and emotional dimensions from Woolf *et al.* [13] and Aslan *et al.* [14].

Behavioral	Emotional	Engagement
On-task	Satisfied	Engaged
On-task	Confused	Engaged
On-task	Bored	Disengaged
Off-task	Satisfied	Disengaged
Off-task	Confused	Disengaged
Off-task	Bored	Disengaged

by the work of Aslan *et al.* [14].

Behavioral dimension:

- *On-Task*: The student is looking towards the screen or looking down to the keyboard below the screen.
- *Off-Task*: The student is looking everywhere else or eyes completely closed, or head turned away.
- *Can't Decide*: If you cannot decide on the behavioral state.

Emotional dimension:

- *Satisfied*: If the student is not having any emotional problems during the learning task. This can include all positive states of the student from being neutral to being excited during the learning task.
- *Confused*: If the student is getting confused during the learning task. In some cases, this state might include some other negative states such as frustration.
- *Bored*: If the student is feeling bored during the learning task.
- *Can't Decide*: If you cannot decide on the emotional state.

igure has been suppressed due to copyright reasons

Figure 3.5: An example of our annotation software where the annotator is requested to specify the behavioral and emotional dimensions of the displayed sample.

During the annotation process, we show each data sample followed by two questions indicating the engagement's dimensions. The behavioral dimension can be chosen among *on-task*, *off-task*, and *can't decide* options and the emotional dimension can be chosen among *satisfied*, *confused*, *bored*, and *can't decide* options. In each annotation phase, annotators have access to the definitions to label each dimension. A sample of the annotation software is shown in Figure 3.5. In the next step, each sample is categorized as engaged or disengaged by combining the dimensions' labels using Table 3.1. For example, if a particular annotator labels an image as *on-task* and *satisfied*, the category for this image from this annotator is *engaged*. Then, for each image we use the majority of the engaged and disengaged labels to specify the final overall annotation. If a sample receives the label of *can't decide* more than twice (either for the emotional or behavioral dimensions) from different annotators, it is removed from ER dataset (around 7% of the annotated samples). Labeling this kind of samples is a difficult task for annotators, notwithstanding the good level of agreement that was achieved, and finding solutions to reduce the difficulty remains as a future direction of our work. Using this approach, we have created ER dataset consisting of 4627 annotated images including 2290 engaged and 2337 disengaged.

igure has been suppressed due to copyright reasons

Figure 3.6: Randomly selected images of ER dataset including engaged and disengaged.

Table 3.2: The statistics of ER dataset and its partitions.

State	Total	Train	Valid	Test
Engaged	2290	1589	392	309
Disengaged	2337	1635	323	379
Total	4627	3224	715	688

Dataset Preparation We apply the CNN based face detection algorithm to detect the face of each ER sample. If there is more than one face in a sample, we choose the face with the biggest size. Then, the face is transformed to grayscale and resized into 48-by-48 pixels, which is an effective resolution for engagement detection [147]. Figure 3.6 shows some examples of the ER dataset. We split the ER dataset into training (3224), validation (715), and testing (688) sets, which are subject-independent (the samples in these three sets are from different subjects). Table 3.2 demonstrates the statistics of these three sets.

3.4.2 Engagement Recognition using Deep Learning

We define two Convolutional Neural Network (CNN) architectures as baselines, one designed architecture and one that is similar in structure to VGGnet [10]. The key model of interest in this paper is a version of the latter baseline that incorporates facial expression recognition. For completeness, we also include another baseline that is not based on deep learning, but rather uses support vector machines (SVMs) with histogram of oriented gradients (HOG) features. For all the models, every sample of the ER dataset is normalized so that it has a zero mean and a norm equal to 100.0. Furthermore, for each pixel location, the pixel values are normalized to mean zero and standard deviation one using all ER training data.

HOG+SVM We trained a method using the histogram of oriented gradients (HOG) features extracted from ER samples and a linear support vector machine (SVM), which we call the HOG+SVM model. The model is similar to that of Kamath *et al.* [43] for recognizing engagement from images and is used as a baseline model in this work. HOG [153] applies gradient directions or edge orientations to express objects in local regions of images. For example, in facial expression recognition tasks, HOG features can represent the forehead's wrinkling by horizontal edges. A linear SVM is usually used to classify HOG features. In this work, C, determining the misclassification rate of training samples against the objective function of SVM, is fine-tuned, using the validation set of the ER dataset, to the value of 0.1.

Convolutional Neural Network We use the training and validation sets of the ER dataset to train a Convolutional Neural Networks (CNNs) for this task from scratch (the CNN model); this constitutes another of the baseline models in this work. The model's architecture is shown in Figure 3.7. The model contains two convolutional (Conv.) layers, followed by two max pooling (Max.) layers with stride 2, and two fully connected (FC) layers, respectively. A rectified linear unit (ReLU) activation function [154] is applied after all Conv. and FC layers. The last step of the CNN model includes a softmax layer, followed by a cross-entropy loss, which consists of two neurons indicating engaged and disengaged classes. To overcome model over-fitting, we apply a dropout layer [155] after every Conv. and hidden FC layer. Local response normalization [64] is used after the first Conv. layer. As the optimizer algorithm, stochastic gradient descent with mini-batching and a momentum of 0.9 is used. Using Equation 3.1, the learning rate at step t (a_t) is decayed by the rate (r) of 0.8 in the decay step (s) of 500. The total number of iterations from the beginning of the training phase

Image has been removed due to copyright reasons

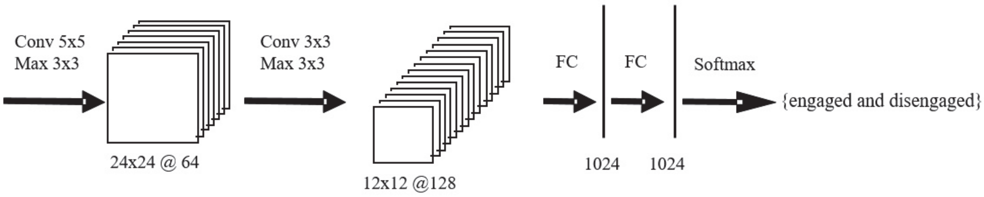


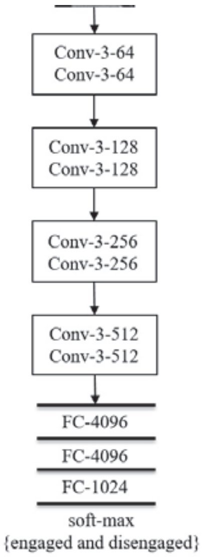
Figure 3.7: The architecture of the CNN Model. We denote convolutional, max-pooling, and fully-connected layers with “Conv”, “Max”, and “FC”, respectively.

is global step (g).

$$a_t = a_{t-1} \times r^{\frac{g}{s}} \tag{3.1}$$

Very Deep Convolutional Neural Network Using the ER dataset, we train a deep model which has eight Conv. and three FC layers similar to VGG-B architecture [10], but with two fewer Conv. layers. The model is trained using two different scenarios. Under the first scenario, the model is trained from scratch initialized with random weights; we call this the VGGnet model (Figure 3.8), and this constitutes the second of our deep learning baseline models. Under the second scenario, which uses the same architecture, the model's layers, except the softmax layer, are initialized by the trained model of §3.3.2, the goal of which is to recognize basic facial expressions; we call this the engagement model (Figure 3.9), and this is the key model of interest in our paper. In this model, all layers' weights are updated and fine-tuned to recognize engaged and disengaged classes in the ER dataset. For both VGGnet and engagement models, after each Conv. block, we have a max pooling layer with stride 2. In the models, the softmax layer has two output units (engaged and disengaged), followed by a cross-entropy loss. Similar to the CNN model, we apply a rectified linear unit (ReLU) activation function [154] and a dropout layer [155] after all Conv. and hidden FC layers. Furthermore, we apply local response normalization after the first Conv. block. We use the same approaches to optimization and learning rate decay as in the CNN model.

Images have been suppressed due to copyright reasons



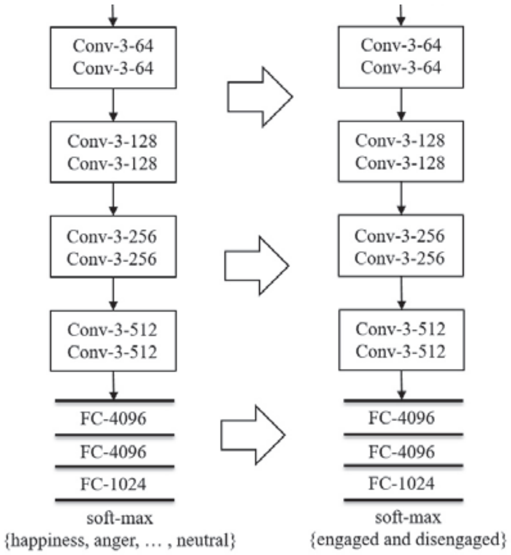


Figure 3.8: The architecture of the VGGnet model on ER dataset. “Conv” and “FC” are convolutional and fully connected layers.

Figure 3.9: The facial expression recognition model on FER-2013 dataset (left). The engagement model on ER dataset (right).

3.5 Experiments

3.5.1 Evaluation Metrics

In this work, the performance of all models are reported on the both validation and test splits of the ER dataset. We use three performance metrics including classification accuracy, F1 measure and the area under the ROC (receiver operating characteristics) curve (AUC). In this work, classification accuracy specifies the number of positive (engaged) and negative (disengaged) samples which are correctly classified and are divided by all testing samples (Equation 3.2).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{3.2}$$

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively. F1 measure is calculated using Equation 3.3.

$$F1 = 2\times\frac{p\times r}{p+r} \tag{3.3}$$

where p is precision defined as $\frac{TP}{TP+FP}$ and r is recall defined as $\frac{TP}{TP+FN}$. AUC is a popular

Table 3.3: The results of the models (%) on the validation set of ER dataset.

Method	Accuracy	F1	AUC
HOG+SVM	67.69	75.40	65.50
CNN	72.03	74.94	71.56
VGGnet	68.11	70.69	67.85
Engagement	**77.76**	**81.18**	**76.77**

Table 3.4: The results of the models (%) on the test set of ER dataset.

Method	Accuracy	F1	AUC
HOG+SVM	59.88	67.38	62.87
CNN	65.70	71.01	68.27
VGGnet	66.28	70.41	68.41
Engagement	**72.38**	**73.90**	**73.74**

metric in engagement recognition task [45, 147, 148]; it is an unbiased assessment of the area under the ROC curve. An AUC score of 0.5 corresponds to chance performance by the classifier, and AUC 1.0 represents the best possible result.

3.5.2 Implementation Details

In the training phase, for data augmentation, input images are randomly flipped along their width and cropped to 48-by-48 pixels (after applying zero-padding because the samples were already in this size). Furthermore, they are randomly rotated by a specific max angle. We set learning rate for the VGGnet model to 0.001 and for other models to 0.002. The batch size is set to 32 for the engagement model and 28 for other models. The best model on the validation set is used to estimate the performance on the test partition of the ER dataset for all models in this work.

3.5.3 Results

Overall Metrics We summarize the experimental results on the validation set and the test set of the ER dataset in Table 3.3 and Table 3.4, respectively. On the sets, the engagement model substantially outperforms all baseline models using all evaluation metrics (it also has a comparable performance against the human performance (74.58%) calculated in §3.4.1), showing the effectiveness of using a trained model on basic facial expression data to initialize

Table 3.5: Confusion matrix of the HOG+SVM model (%).

		Predicted	
		Engaged	Disengaged
GT	Engaged	92.23	7.77
	Disengaged	66.49	33.51

Table 3.6: Confusion matrix of the CNN model (%).

		Predicted	
		Engaged	Disengaged
GT	Engaged	93.53	6.47
	Disengaged	56.99	43.01

Table 3.7: Confusion matrix of the VGGnet model (%).

		Predicted	
		Engaged	Disengaged
GT	Engaged	89.32	10.68
	Disengaged	52.51	47.49

Table 3.8: Confusion matrix of the engagement model (%).

		Predicted	
		Engaged	Disengaged
GT	Engaged	87.06	12.94
	Disengaged	39.58	60.42

an engagement recognition model [3]. All deep models including CNN, VGGnet, and engagement models perform better than the HOG+SVM method, showing the benefit of applying deep learning to recognize engagement. On the test set, the engagement model achieves 72.38% classification accuracy, which outperforms VGGnet by 5%, and the CNN model by more than 6%; it is also 12.5% better than the HOG+SVM method. The engagement model achieved 73.90% F1 measure which is around 3% improvement compared to the deep baseline models and 6% better performance than the HOG+SVM model. Using the AUC metric, as the most popular metric in engagement recognition tasks, the engagement model achieves 73.74% which improves the CNN and VGGnet models by more than 5% and is around 10% better than the HOG+SVM method. There are similar improvements on the validation set.

[3] We could as an alternative pre-train the CNN model with FER. However, although the CNN model has a better performance on the validation set, it cannot generalize on the test set as well as the VGGnet model. Moreover, we mostly focused on the VGG architecture since it has led to the state of the art result.

igure has been suppressed due to copyright reasons

Figure 3.10: Representative engaged (left) and disengaged (right) samples that are correctly classified using the engagement model with high confidence. For example, the predicted probabilities (the confidence levels) for engaged samples from top to bottom are 72.10%, 70.76%, 83.16% and 83.82%, respectively. They for disengaged samples from top to bottom are 99.73%, 71.11%, 81.01% and 70.81%, respectively. The agreement level among annotators is also high. This means that annotators can label these kinds of samples with less difficulties. Out of six annotators, five or all of them usually agreed for choosing either engaged or disengaged labels.

Confusion Matrices We show the confusion matrices of the HOG+SVM, CNN, VGGnet, and engagement models on the ER test set in Table 3.5, Table 3.6, Table 3.7, and Table 3.8, respectively. The tables show the proportions of predicted classes with respect to the ground-truth (GT) classes, allowing an examination of precision per class. It is interesting that the effectiveness of deep models comes through their ability to recognize disengaged samples compared to the HOG+SVM model. Moreover, the tables demonstrate that all models have more difficulties to detect disengaged samples compared to engaged ones. Thus, changing the objective function to assign different error costs to engaged and disengaged samples can be useful and we want to investigate this topic in the future work.

Disengaged samples have a wider variety of body postures and facial expressions than

igure has been suppressed due to copyright reasons

Figure 3.11: Engaged (left) and disengaged (right) examples that are correctly detected using the engagement model with low confidence. For example, the predicted probabilities (the confidence levels) for engaged samples from top to bottom are 57.02% and 69.43% respectively. They for disengaged samples from top to bottom are 56.79% and 62.53% respectively. As shown, there is more visual likeness between engaged and disengaged examples compared to the previous samples. Labeling these kinds of examples is difficult for annotators. For some of these examples, out of six annotators, only four ones agreed to label the examples as engaged or disengaged.

igure has been suppressed due to copyright reasons

Figure 3.12: These samples are wrongly predicted as engaged (left) and disengaged (right) using the engagement model. For example, the predicted probabilities for wrong classes (the confidence levels) for left samples from top to bottom are 53.77%, 65.64% and 73.48% respectively. They for right samples from top to bottom are 55.41%, 51.82% and 70.81% respectively. The ground-truth labels of the left samples are disengaged and the right samples are engaged. Here, similar to the previous figure, labeling engaged or disengaged samples is also difficult for annotators which shows some challenging examples in engagement recognition tasks.

engaged sample (see engaged and disengaged examples in Figure 3.10). Due to complex structures, deep learning models are more powerful in capturing these wider variations. The VGGnet model, which has a more complex architecture compared to the CNN model, can also detect disengaged samples with a higher probability. Since we pre-trained the engagement model on basic facial expression data including considerable variations of samples, this model is the most effective approach to recognize disengaged samples achieving 60.42% precision which is around 27% improvement in comparison with the HOG+SVM model (See Figure 3.11 and 3.12 which are showing some challenging examples to recognize engagement).

3.6 Summary

Reliable models that can recognize engagement during a learning session, particularly in contexts where there is no instructor present, play a key role in allowing learning systems to intelligently adapt to facilitate the learner. There is a shortage of data for training systems to do this; the first contribution of the work is a new dataset, labelled by annotators with expertise in psychology, that we hope will facilitate research on engagement recognition from visual data. In this chapter, we have used this dataset to train models for the task of automatic engagement recognition, including for the first time deep learning models. The next contribution has been the development of a model, called the engagement model, that can address the shortage of engagement data to train a reliable deep learning model. The engagement model has two key steps. First, we pre-train the model using basic facial expression data, of which is relatively abundant. Second, we train the model to produce a rich deep learning-based representation for engagement, instead of commonly used features and classification methods in this domain. We have evaluated this model with respect to a comprehensive range of baseline models to demonstrate its effectiveness, and have shown that it leads to a considerable improvement against the baseline models using all standard evaluation metrics.

In terms of representation of facial expressions, we have found that the representation learned from our facial expression recognition model is transferable to quite a different task in response to **RQ 1** discussed in Chapter 1, suggesting it could be useful for the image captioning task we set up in the next chapter.

4

Image Captioning using Facial Expression and Attention

Benefiting from advances in machine vision and natural language processing techniques, current image captioning systems are able to generate detailed visual descriptions. For the most part, these descriptions represent an objective characterisation of the image, although some models do incorporate subjective aspects related to the observer's view of the image, such as sentiment (discussed in §2.4 and Chapters 5 and 6); current models, however, usually do not consider the emotional content of images during the caption generation process. This chapter addresses this issue by proposing novel image captioning models which use facial expression features to generate image captions. To do this, we draw on the representation of facial expression features that we found in Chapter 3 to be transferable from emotion recognition to other tasks.[1].

4.1 Introduction

Image captioning systems aim to describe the content of an image using Computer Vision and Natural Language Processing approaches which have led to important and practical applications such as helping visually impaired individuals [1]. This is a challenging task because we have to capture not only the objects but also their relations and the activities displayed in the image to generate a meaningful description. The impressive progress in deep neural networks and large image captioning datasets has recently resulted in a considerable improvement in generating automatic image captions [1, 7, 8, 19, 20, 89, 156–158].

However, current image captioning methods often overlook the emotional aspects of the image, which play an important role in generating captions that are more semantically correlated with the visual content. For example, Figure 4.1 shows three images with their corresponding human-generated captions including emotional content. The first image at left has the caption of "a dad smiling and laughing with his child" using "smiling" and "laughing" to describe the emotional content of the image. In a similar fashion, 'angry" and "happy" are applied in the second and the third images, respectively.[2] These examples demonstrate how image captioning systems that recognize emotions and apply them can generate richer, more expressive and more human-like captions. This desideratum of incorporating emotional content is one that is general to intelligent systems, which researchers like Lisetti *et al.* [36] have identified as necessary to generate more effective and adaptive outcomes. In this work, we seek to demonstrate this desideratum holds also for image captioning systems.

[1] The content of this chapter is based on the following publications:

Omid Mohamad Nezami, Mark Dras, Peter Anderson, Len Hamey (2018). Face-Cap: Image Captioning using Facial Expression Analysis. *Proceedings of the 2018 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2018),* Dublin, Ireland.

Omid Mohamad Nezami, Mark Dras, Stephen Wan, Cecile Paris (2020). Image Captioning using Facial Expression and Attention. *Journal of Artificial Intelligence Research (JAIR),* vol. 68, pp. 661-689.

[2] We note that while the second and third images use adjectives that correspond directly to labels used in the facial expression recognition models discussed in Chapter 3, the description of the first image represents a broader notion of emotion. The representation of facial expression thus needs to be transferable, as in Chapter 3.

Images have been suppressed due to copyright reasons

A dad smiling and laughing with his child. Two men with angry faces drink out of white cups. Two happy people pose for a photo.

Figure 4.1: The examples of Flickr 30K dataset [4] with emotional content. The green color indicates words with strong emotional values.

Although detecting emotions from visual data has been an active area of research in the recent years [114, 120], designing an effective image captioning system to employ emotions in describing an image is still an open and challenging problem.

As discussed in §2.4, a few models have incorporated sentiment or other non-factual information into image captions [2, 30, 31]; they typically require the collection of a supplementary dataset, from which a sentiment vocabulary is derived, drawing on work in Natural Language Processing [109] where sentiment is usually characterized as one of positive, neutral or negative. Mathews *et al.* [2], for instance, constructed a sentiment image-caption dataset via crowdsourcing, where annotators were asked to include either positive sentiment (e.g. *a cuddly cat*) or negative sentiment (e.g. *a sinister cat*) using a fixed vocabulary; their model was trained on both this and a standard set of factual captions. These kinds of approaches typically embody descriptions of an image that represent an *observer's* view towards the image (e.g. *a cuddly cat* for a positive view of an image, versus *a sinister cat* for a negative one); they do not aim to capture the emotional content of the image, as in Figure 4.1.

To capture the emotional content of the image, we propose two groups of models: Face-Cap and Face-Attend. Face-Cap feeds in a fixed one-hot encoding vector similar to Hu *et al.* [35] and You *et al.* [48]. In comparison, we represent the aggregate facial expressions of the input image at different time steps of our caption generator, which employs a long short-term memory (LSTM) architecture. To construct the vector, we train a state-of-the-art facial expression recognition (FER) model which automatically recognizes facial expressions (e.g. happiness, sadness, fear, and so on), as in Chapter 3. However, the recognized facial expressions are not always reliable because the FER model is not 100% accurate. This can result in an image captioning architecture that propagates errors. Moreover, these facial expression classes do not necessarily align with more fine-grained facial expression representations such as action units (AUs), one framework for characterising different facial muscle movements [159]. Hence, we propose an alternative representation that uses more fine-grained facial expression features (e.g. convolutional features) which could potentially be more useful than the one-hot encoding representation. We also recognize from design choices that there might be images that Face-Cap may not perform well on (e.g. images including multiple faces such as Figure 4.1, because we have a single encoding representation of emotion for the whole image) and an attention mechanism might better localise emotional features in a way useful for image captioning. Thus, Face-Attend employs an attention mechanism to selectively attend to facial features, for different detected faces in an image, extracted from the last convolutional layer of the FER model. Face-Attend uses two LSTMs to incorporate facial features along with general visual content in generating image descriptions.

4.2 Approach

In this section, we describe Face-Cap and Face-Attend, our proposed models for generating image captions using facial expression analyses. The models are inspired by two popular image captioning models, specifically Show-Attend-Tell [7] and Up-Down-Captioner [8].

Show-Attend-Tell is a well-known and widely used image captioning system that incorporates an attention mechanism to attend to spatial visual features. It demonstrates a significant improvement over earlier image captaining models that do not have an attention mechanism; we discussed it in §2.3. From this starting point, we propose the Face-Cap model which similarly attends to visual features and additionally uses facial expression analyses in generating image captions. Face-Cap incorporates a one-hot encoding vector as a representation of the facial expression analysis, similar to the representations used for sentiment by Hu *et al.* [35] and You *et al.* [48].

Up-Down-Captioner is the current state-of-the-art image captioning model, defining a new architecture to incorporate attended visual features in generating image captions; we also discussed this in §2.3. In this model, the features directly relate to the objects in the image and two LSTMs (one for generating attention weights and another one for a language model) are used to generate image captions. We propose Face-Attend based on this kind of architecture, as we can apply more fine-grained facial expression features and use two LSTMs to attend to the features in addition to the general visual features. Because Up-Down-Captioner already incorporates attention on objects in the image, our models derived from this allow us to examine the effectiveness of the facial expression features beyond just recognition of the face as an object.

In what follows, we describe our datasets and our facial expression recognition model that are used by Face-Cap and Face-Attend models. We then explain the models in detail.

4.2.1 Datasets

Facial Expression Recognition The setup here is similar to §3.3. To train our facial expression recognition model, we use the facial expression recognition 2013 (FER-2013) dataset [9]. It includes images labeled with standard facial expression categories (*happiness*, *sadness*, *fear*, *surprise*, *anger*, *disgust* and *neutral*). It consists of 35,887 examples (28,709 for training, 3589 for public and 3589 for private test), collected by means of the Google search API. The examples are in grayscale at the size of 48-by-48 pixels. We split the training set of FER-2013 into two sections after removing 11 completely black examples: 25,109 for training and 3589 for validating the model. Similar to other work in this domain [127, 128, 130], we use the private test set of FER-2013 for the performance evaluation of the model after the training phase. To compare with the related work, we do not apply the public test set either for training or for validating the model.

Image Captioning To train Face-Cap and Face-Attend, we have extracted a subset of the Flickr 30K dataset with image captions [4] that we name FlickrFace11K. It contains 11,696 images including human faces detected using a convolutional neural network-based face detector [152].[3] Each image has five ground-truth captions. We observe that the Flickr 30K dataset is a good source for our dataset, because it has a larger portion of images that include human faces, in comparison with other image caption datasets such as the MSCOCO

[3] The new version (2018) of Dlib library is applied.

dataset [160]. We split the FlickrFace11K samples into 8696 for training, 2000 for validation and 1000 for testing. Since we aim to train a facial expression recognition model on FER-2013 and use it as a facial expression feature extractor on the samples of FlickrFace11K, we need to adapt and make the samples consistent with the FER-2013 data. This is inspired by the domain adaptation topic [161, 162] to address differences between the source and target domains. To this end, the face detector is used to pre-process the faces of FlickrFace11K. The faces are cropped from each sample. Then, we transform each face to grayscale and resize it into 48-by-48 pixels, which is the same as in the FER-2013 data.

4.2.2 Facial Expression Recognition Model

Again, the setup here is similar to §3.3. We train a facial expression recognition (FER) model using the VGG-B architecture [10], but we remove the last convolutional block, including two convolutional layers, and the last max pooling layer from the architecture. We use 3 × 3 kernel sizes for all remained convolutional layers. We use a batch normalization layer [163] after every remained convolutional block. Our FER model gives a similar performance to the state-of-the-art under a similar experimental setting, as described in Pramerdorfer [130]; this is higher than reported human performance [9]. The framework of our FER model is shown in Figure 4.2.

From the FER model, we extract two classes of facial expression features to use in our image captioning models. The first class of features is the output of the final softmax layer of our FER model, $\boldsymbol{a}_i = (a_{i,1}, \ldots, a_{i,7})$, representing the probability distribution of the facial expression classes for the i^{th} face in the image. For the image as a whole, we construct a vector of facial expression features $\boldsymbol{s} = \{s_1, \ldots, s_7\}$ used in our image captioning model as in Equation 4.1.

$$s_k = \begin{cases} 1 & \text{for } k = \arg\max \sum_{1 \leq i \leq n} a_{i,j}, \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

where n is the number of faces in the image. That is, s is a one-hot encoding, which we refer to as the facial encoding vector, of the aggregate facial expressions of the image.

The second class of features consists of convolutional features extracted from the FER

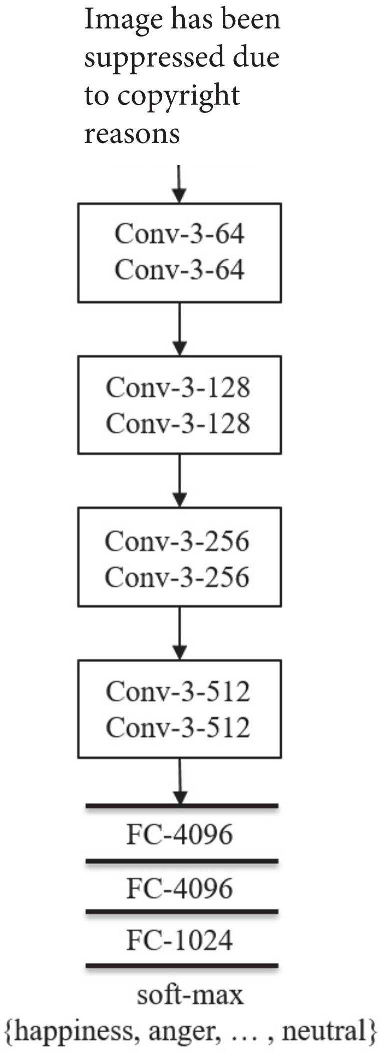


Figure 4.2: Our facial expression recognition module. Each convolutional block is shown with a rectangle including two convolutional layers. FC indicates fully-connected layers. Max pooling layers after convolutional blocks are not shown for simplicity.

model, giving a more fine-grained representation of the faces in the image. For each face, we extract the last convolutional layer of the model, giving 6 × 6 × 512 features. We convert these into a 36 × 512 representation. We restrict ourselves to a maximum of three faces: in our FlickrFace11K dataset, 96.5% of the images have at most three faces. If one image has more than three faces, we select the three faces with the biggest bounding box sizes. We then concatenate the features of the three faces leading to 108 × 512 dimensions, $\boldsymbol{f} = \{\boldsymbol{f}_1, ..., \boldsymbol{f}_{K^\star}\}, \boldsymbol{f}_i \in \mathbb{R}^D$, where $K^\star$ is 108 and D is 512; we refer to these as facial features. If a sample includes fewer than three faces, we fill in dimensions with zero values.

4.2.3 Image Captioning Models

Our image captioning models aim to generate an image caption, $\boldsymbol{x} = \{x_1, \ldots, x_T\}$, where x_i is a word and T is the length of the caption, using facial expression analyses. As a representation of the image, all our models use the last convolutional layer of the VGG-E architecture [10]. In addition to our proposed facial features, the VGG-E network trained on ImageNet [105] produces a $14 \times 14 \times 512$ feature map. We convert this into a 196×512 representation, $\boldsymbol{c} = \{\boldsymbol{c}_1, ..., \boldsymbol{c}_K\}, \boldsymbol{c}_i \in \mathbb{R}^D$, where K is 196 and D is 512; we refer to this as the visual features. The specifics of the image captioning models are explained below.

Face-Cap These models essentially extend the Show-Attend-Tell architecture of [7]. Like these models, we use a long short-term memory (LSTM) network as our caption generator. The LSTM incorporates the emotional content of the image in the form of the facial encoding vector defined in Equation 4.1. We propose two variants, Face-Cap-Repeat and Face-Cap-Memory, that differ in terms of how the facial encoding vector is incorporated.

Face-Cap-Repeat In Face-Cap-Repeat, in each time step (t), the LSTM uses the previous word embedded in M dimensions ($\boldsymbol{w}_{t-1} \in \mathbb{R}^M$ selected from an embedding matrix learned without pre-training from random initial values), the previous hidden state ($\boldsymbol{h}_{t-1}$), the attention-based features ($\hat{\boldsymbol{c}}_t$), and the facial encoding vector ($\boldsymbol{s}$) to calculate input gate ($\boldsymbol{i}_t$), forget gate ($\boldsymbol{f}_t$), output gate ($\boldsymbol{o}_t$), input modulation gate ($\boldsymbol{g}_t$), memory cell ($\boldsymbol{m}_t$), and hidden state ($\boldsymbol{h}_t$).

$$\begin{aligned}
\boldsymbol{i}_t &= \sigma(\boldsymbol{W}_i \boldsymbol{w}_{t-1} + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{C}_i \hat{\boldsymbol{c}}_t + \boldsymbol{S}_i \boldsymbol{s} + \boldsymbol{b}_i) \\
\boldsymbol{f}_t &= \sigma(\boldsymbol{W}_f \boldsymbol{w}_{t-1} + \boldsymbol{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{C}_f \hat{\boldsymbol{c}}_t + \boldsymbol{S}_f \boldsymbol{s} + \boldsymbol{b}_f) \\
\boldsymbol{o}_t &= \sigma(\boldsymbol{W}_o \boldsymbol{w}_{t-1} + \boldsymbol{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{C}_o \hat{\boldsymbol{c}}_t + \boldsymbol{S}_o \boldsymbol{s} + \boldsymbol{b}_o) \\
\boldsymbol{g}_t &= \tanh(\boldsymbol{W}_g \boldsymbol{w}_{t-1} + \boldsymbol{U}_g \boldsymbol{h}_{t-1} + \boldsymbol{C}_g \hat{\boldsymbol{c}}_t + \boldsymbol{S}_g \boldsymbol{s} + \boldsymbol{b}_g) \\
\boldsymbol{m}_t &= \boldsymbol{f}_t \boldsymbol{m}_{t-1} + \boldsymbol{i}_t \boldsymbol{g}_t \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \tanh(\boldsymbol{m}_t)
\end{aligned} \tag{4.2}$$

where $\boldsymbol{W}, \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{S}$, and $\boldsymbol{b}$ are learned weights and biases and σ is the logistic sigmoid activation function. From now on, we show this LSTM equation using the shorthand of Equation 4.3.

$$\boldsymbol{h}_t = \text{LSTM}(\boldsymbol{h}_{t-1}, [\hat{\boldsymbol{c}}_t, \boldsymbol{w}_{t-1}, \boldsymbol{s}]) \tag{4.3}$$

To calculate $\hat{\boldsymbol{c}}_t$, for each time step t, Face-Cap-Repeat weights visual features (c) using a soft attention mechanism as in Equation 4.4 and 4.5.

$$\begin{aligned} e_{i,t} &= \boldsymbol{W}_e^T \tanh(\boldsymbol{W}_c \boldsymbol{c}_i + \boldsymbol{W}_h \boldsymbol{h}_{t-1}) \\ \boldsymbol{e}'_t &= \mathrm{softmax}(\boldsymbol{e}_t) \end{aligned} \tag{4.4}$$

where $e_{i,t}$ are unnormalized weights for the visual features ($\boldsymbol{c}_i$) and $\boldsymbol{e}'_t$ are the normalized weights using a softmax layer at time step t. Our trained weights are represented by $\boldsymbol{W}_x$. Finally, our attention-based features ($\hat{\boldsymbol{c}}_t$) are calculated using:

$$\hat{\boldsymbol{c}}_t = \sum_{1 \leq i \leq K} \boldsymbol{e}'_{i,t} \boldsymbol{c}_i \tag{4.5}$$

To initialize the LSTM's hidden state ($\boldsymbol{h}_0$), we feed the facial features through a standard multilayer perceptron, shown in Equation 4.6.

$$\boldsymbol{h}_0 = \mathrm{MLP}_{init}(\boldsymbol{s}) \tag{4.6}$$

We use the current hidden state ($\boldsymbol{h}_t$) to calculate the negative log-likelihood of s in each time step (Equation 4.7); we call this the face objective function.

$$L_f = -\sum_{1 \leq i \leq 7} s_i \log(\boldsymbol{p}_e(i|\boldsymbol{h}_t)) \tag{4.7}$$

where a multilayer perceptron generates $\boldsymbol{p}_e(i|\boldsymbol{h}_t)$, which is the categorical probability distribution of the current hidden state across the facial expression classes. We adapt this from [48], who use this objective function for injecting ternary-valued sentiment (positive, neutral, negative) into captions. This loss is estimated and averaged, over all steps, during the training phase.

The general objective function of Face-Cap-Repeat is defined as:

$$L_{g1} = -\sum_{1 \leq t \leq T} \log(\boldsymbol{p}_x(\boldsymbol{x}_t \,|\, \hat{\boldsymbol{c}}_t, \boldsymbol{h}_t)) + \sum_{1 \leq k \leq K} (1 - \sum_{1 \leq t \leq T} c_t)^2 \tag{4.8}$$

A multilayer perceptron and a softmax layer is used to calculate p_x, the probability of the next generated word (we apply argmax over this to generate the next word):

$$\boldsymbol{p}_x(\boldsymbol{x}_t \,|\, \hat{\boldsymbol{c}}_t, \boldsymbol{h}_t) = \mathrm{softmax}(\boldsymbol{W}'_c \hat{\boldsymbol{c}}_t + \boldsymbol{W}'_h \boldsymbol{h}_t + \boldsymbol{b}') \tag{4.9}$$

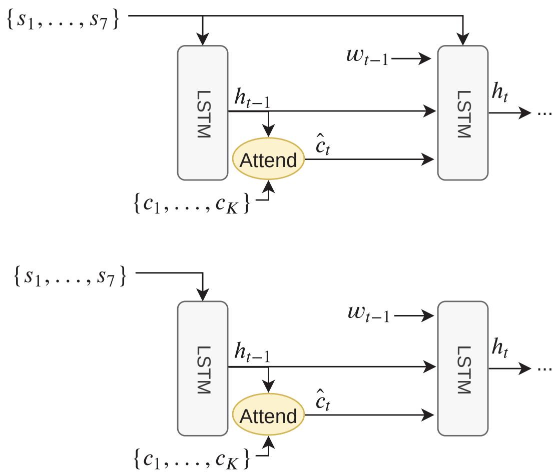


Figure 4.3: The frameworks of Face-Cap-Repeat (top), and Face-Cap-Memory (bottom). Attend is our attention mechanism attending to the visual features, $\{\mathbf{c}_1,\ldots,\mathbf{c}_K\}$.

where the learned weights and bias are given by $\boldsymbol{W}'$ and $\boldsymbol{b}'$. The last term in Equation 4.8 is to encourage Face-Cap-Repeat to equally pay attention to different sets of c when a caption generation process is finished. Here, we sum the values of L_{g1} and L_f to calculate the final loss.

Face-Cap-Memory The above Face-Cap-Repeat model feeds in the facial encoding vector at the initial step (Equation 4.6) and at each time step (Equation 4.3), shown in Figure 4.3 (top). The LSTM uses the vector for generating every word because the vector is fed at each time step. Since not all words in the ground truth captions will be related to the vector — for example in Figure 4.1, where the majority of words are not directly related to the facial expressions — this mechanism could lead to an overemphasis on these features.

Our second variant of the model, Face-Cap-Memory, is as above except that the $\boldsymbol{s}$ term is removed from Equation 4.3: we do not apply the facial encoding vector at each time step and we only apply it at the initial time step per Equation 4.6 (Figure 4.3 (bottom)). We rely on Equation 4.7 to memorize this facial expression information. Using this mechanism, the LSTM can effectively take the information in generating image captions and ignore the information when it is irrelevant. To handle an analogous issue for sentiment, [48] implemented a sentiment cell, working similarly to the memory cell in the LSTM, initialized by the ternary sentiment. They then fed the visual features to initialize the memory cell and hidden state of the LSTM. Similarly, Face-Cap-Memory uses the facial features to initialize the memory cell and hidden state. Using the attention mechanism, our model applies the visual features in generating every caption word.

Face-Attend Here, we apply two LSTMs to attend to our more fine-grained facial features ($\boldsymbol{f}$) explained in Section 4.2.2, in addition to our visual features ($\boldsymbol{c}$). We propose two variant architectures for combining these features, Dual-Face-Att and Joint-Face-Att, explained below.

Dual-Face-Att The framework of Dual-Face-Att is shown in Figure 4.4. To generate image captions, Dual-Face-Att includes two LSTMs: one, called F-LSTM, to attend to facial features and another one, called C-LSTM, to attend to visual content. Both LSTMs are defined as in Equation 4.10, but with separate training parameters.

$$\boldsymbol{h}_{t,z} = \text{LSTM}(\boldsymbol{h}_{t,z-1}, [\hat{\boldsymbol{z}}_t, \boldsymbol{w}_{t-1}]) \tag{4.10}$$

In both LSTMs, to calculate $\hat{\boldsymbol{z}}_t$ at each time step (t), features $\boldsymbol{z}$ (the facial features ($\boldsymbol{f}$) for F-LSTM and the visual features ($\boldsymbol{c}$) for C-LSTM) are weighted using a soft attention mechanism, but with separately learned parameters.

$$\begin{aligned} e_{i,t,z} &= \boldsymbol{W}_{e,z}^{T} \tanh(\boldsymbol{W}_z \boldsymbol{z}_i + \boldsymbol{W}_{h,z} \boldsymbol{h}_{t,z-1}) \\ \boldsymbol{e}'_{t,z} &= \text{softmax}(\boldsymbol{e}_{t,z}) \end{aligned} \tag{4.11}$$

where $e_{i,t,z}$ and $\boldsymbol{e}'_{t,z}$ are unnormalized weights for features $\boldsymbol{z}_i$, and normalized weights using a softmax layer, respectively. Our trained weights are $\boldsymbol{W}_z$. Finally, our attention-based features ($\hat{\boldsymbol{z}}_t$) are calculated using:

$$\hat{\boldsymbol{z}}_t = \sum_{1 \le i \le K_z} e'_{i,t,z} \boldsymbol{z}_i \tag{4.12}$$

K_z is $K^{\star}$ for F-LSTM and K for C-LSTM. The initial LSTM's hidden state ($\boldsymbol{h}_{0,z}$) is computed using a standard multilayer perceptron:

$$\boldsymbol{h}_{0,z} = \text{MLP}_{init,z}(\frac{1}{K_z} \sum_{1 \le i \le K_z} \boldsymbol{z}_i) \tag{4.13}$$

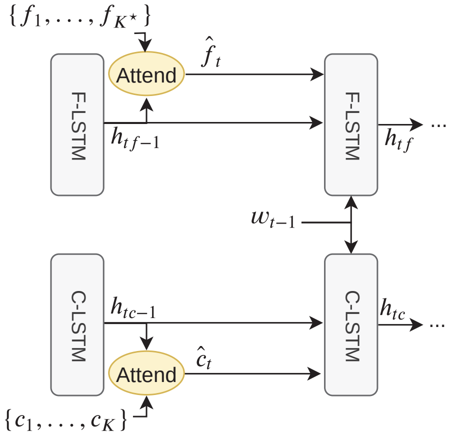


Figure 4.4: Dual-Face-Att model enables generating image captions with both facial features $\{\boldsymbol{f}_1,\ldots,\boldsymbol{f}_{K^\star}\}$ and visual content $\{\boldsymbol{c}_1,\ldots,\boldsymbol{c}_K\}$.

The objective function of Dual-Face-Att is defined using Equation (4.14).

$$\begin{aligned} L_{g2} = -\lambda[\sum_{1\le t\le T}\log(\boldsymbol{p}_{x,c}(\boldsymbol{x}_t\,|\,\hat{\boldsymbol{c}}_t,\boldsymbol{h}_{t,c})) + \sum_{1\le k\le K}(1-\sum_{1\le t\le T} c_{t,k})^2] - \\ (1-\lambda)[\sum_{1\le t\le T}\log(\boldsymbol{p}_{x,f}(\boldsymbol{x}_t\,|\,\hat{\boldsymbol{f}}_t,\boldsymbol{h}_{t,f})) + \beta_1 \sum_{1\le k\le K^*}(1-\sum_{1\le t\le T} f_{t,k})^2] \end{aligned} \quad (4.14)$$

where a multilayer perceptron and a softmax layer, for each LSTM, are used to calculate $\boldsymbol{p}_{x,f}$ and $\boldsymbol{p}_{x,c}$ (the probabilities of the next generated word on the basis of facial expression features and visual features, respectively):

$$\begin{aligned} \boldsymbol{p}_{x,f}(\boldsymbol{x}_t\,|\,\hat{\boldsymbol{f}}_t,\boldsymbol{h}_{t,f}) &= \text{softmax}(\boldsymbol{W}_f\hat{\boldsymbol{f}}_t + \boldsymbol{W}_{h,f}\boldsymbol{h}_{t,f} + \boldsymbol{b}_f) \\ \boldsymbol{p}_{x,c}(\boldsymbol{x}_t\,|\,\hat{\boldsymbol{c}}_t,\boldsymbol{h}_{t,c}) &= \text{softmax}(\boldsymbol{W}_c\hat{\boldsymbol{c}}_t + \boldsymbol{W}_{h,c}\boldsymbol{h}_{t,c} + \boldsymbol{b}_c) \end{aligned} \quad (4.15)$$

λ and β_1 are regularization constants (these are hyperparameters). The ultimate probability of the next generated word is:

$$\boldsymbol{p}_x(\boldsymbol{x}_t\,|\,\hat{\boldsymbol{f}}_t,\boldsymbol{h}_{t,f},\hat{\boldsymbol{c}}_t,\boldsymbol{h}_{t,c}) = \lambda\boldsymbol{p}_{x,f}(\boldsymbol{x}_t\,|\,\hat{\boldsymbol{f}}_t,\boldsymbol{h}_{t,f}) + (1-\lambda)\boldsymbol{p}_{x,c}(\boldsymbol{x}_t\,|\,\hat{\boldsymbol{c}}_t,\boldsymbol{h}_{t,c}) \quad (4.16)$$

Joint-Face-Att The above Dual-Face-Att model uses two LSTMs: one for attending to visual features and another one for attending to facial features. In the model, both LSTMs also play the role of language models (Equation 4.16) and directly impact on the prediction of

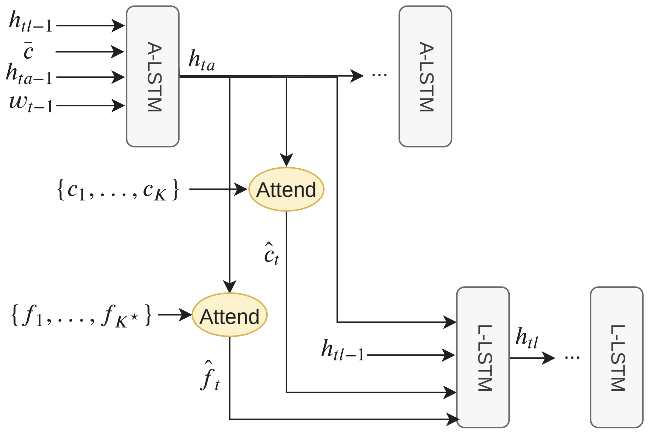


Figure 4.5: Joint-Face-Att model enables generating image captions with two LSTMs for learning attention weights and generating captions, separately. (This is a two-layer LSTM and L-LSTM has the role of generating $\boldsymbol{w}_t$.)

the next generated word. However, the recent state-of-the-art image captioning model of [8] achieved better performance by using two LSTMs with differentiated roles: one for attending only to visual features and a second one purely as a language model. Inspired by this, we define our Joint-Face-Att variant to use one LSTM, which we call A-LSTM, to attend to image-based features, both facial and visual; and a second one, which we call L-LSTM, to generate language (Figure 4.5). Here, we calculate the hidden state of A-LSTM using:

$$\boldsymbol{h}_{t,a} = \mathrm{LSTM}(\boldsymbol{h}_{t,a-1}, [\bar{\boldsymbol{c}}, \boldsymbol{h}_{t,l-1}, \boldsymbol{w}_{t-1}]) \tag{4.17}$$

where $\bar{\boldsymbol{c}} = \frac{1}{K}\sum_{1\le i\le K} \boldsymbol{c}_i$ is the mean-pooled visual features and $h_{t,l-1}$ is the previous hidden state of L-LSTM. We also calculate the hidden state of L-LSTM using:

$$\boldsymbol{h}_{t,l} = \mathrm{LSTM}(\boldsymbol{h}_{t,l-1}, [\hat{\boldsymbol{f}}_t, \hat{\boldsymbol{c}}_t, \boldsymbol{h}_{t,a}]) \tag{4.18}$$

where $\hat{\boldsymbol{f}}_t$ and $\hat{\boldsymbol{c}}_t$ are the attended facial features and visual features, respectively. They are defined analogously to Equation 4.4 and 4.5, but $\boldsymbol{h}_{t,z-1} = \boldsymbol{h}_{t,a}$ with different sets of trainable parameters. $\boldsymbol{h}_a$ and $\boldsymbol{h}_l$ are similarly initialized as follows using two standard multilayer perceptrons:

$$\begin{aligned} \boldsymbol{h}_{0,l} &= \mathrm{MLP}_{init,l}(\frac{1}{K}\sum_{1\le i\le K} \boldsymbol{c}_i) \\ \boldsymbol{h}_{0,a} &= \mathrm{MLP}_{init,a}(\frac{1}{K}\sum_{1\le i\le K} \boldsymbol{c}_i) \end{aligned} \tag{4.19}$$

The objective function of Joint-Face-Att is:

$$L_{g3} = -[\sum_{1\le t\le T} \log(\boldsymbol{p}_x(\boldsymbol{x}_t \,|\, \hat{\boldsymbol{c}}_t, \hat{\boldsymbol{f}}_t, \boldsymbol{h}_{t,l})) + \sum_{1\le k\le K} (1 - \sum_{1\le t\le T} c_{t,k})^2 + \beta_2 \sum_{1\le k\le K^\star} (1 - \sum_{1\le t\le T} f_{t,k})^2] \tag{4.20}$$

where β_2 is a regularization constant and $\boldsymbol{p}_x$ is the probability of the next generated word calculated as follows:

$$\boldsymbol{p}_x(\boldsymbol{x}_t \,|\, \hat{\boldsymbol{c}}_t, \hat{\boldsymbol{f}}_t, \boldsymbol{h}_{t,l}) = \mathrm{softmax}(\boldsymbol{W}_{c,l}\hat{\boldsymbol{c}}_t + \boldsymbol{W}_{f,l}\hat{\boldsymbol{f}}_t + \boldsymbol{W}_{h,l}\boldsymbol{h}_{t,l} + \boldsymbol{b}_l) \tag{4.21}$$

where $\boldsymbol{W}_{x,l}$ and $\boldsymbol{b}_l$ are trainable weights and bias, respectively.

4.3 Experimental Setup

In the following sections, we describe the evaluation setup and discuss the experimental results. At the end, we analyse the failure cases.

4.3.1 Evaluation Metrics

Overall Metrics Following previous work, we evaluate our image captioning model using standard evaluation metrics including BLEU [164], ROUGE [165], METEOR [166], CIDEr [100], and SPICE [167]. Larger values are better results for all metrics. BLEU calculates a weighted average for n-grams with different sizes as a precision metric:

$$BLEU_n(x,y) = \frac{\sum_{g_n \in x} \min\left(c_x(g_n), \max_{j=1,\ldots,|y|} c_{y_j}(g_n)\right)}{\sum_{w_n \in x} c_x(g_n)} \tag{4.22}$$

where x is the generated caption, y is a set of reference captions, g_n is n-gram and c_z is the count of g_n in caption z. ROUGE is a recall-oriented metric that calculates F-measures using the matched n-grams between the generated captions and their corresponding reference summaries:

$$ROUGE_n(x,y) = \frac{\sum_{j=1}^{|y|} \sum_{g_n \in y_j} \min\left(c_x(g_n), c_{y_j}(g_n)\right)}{\sum_{j=1}^{|y|} \sum_{g_n \in y_j} c_{y_j}(g_n)} \tag{4.23}$$

METEOR uses a weighted F-measure matching synonyms and stems in addition to standard n-gram matching. CIDEr uses a n-gram matching, calculated using the cosine similarity, between the generated captions and the consensus of the reference captions:

$$CIDEr_n(x,y) = \frac{1}{|y|}\sum_{j=1}^{|y|}\frac{\boldsymbol{T}^n(x)\boldsymbol{T}^n(y_j)}{||\boldsymbol{T}^n(x)||\,||\boldsymbol{T}^n(y_j)||} \tag{4.24}$$

where $\boldsymbol{T}^n(z)$ is a vector that is formed by Term Frequency Inverse Document Frequency (TF-IDF) for all n-grams in z and $||\boldsymbol{T}^n(z)||$ is its magnitude. Finally, SPICE calculates F-score for semantic tuples derived from scene graphs:

$$SPICE(x,y) = F_1(x,y) = \frac{2P(x,y)R(x,y)}{P(x,y)+R(x,y)} \tag{4.25}$$

where $P(x,y) = \frac{|Map(x) \otimes Map(y)|}{Map(x)}$, $R(x,y) = \frac{|Map(x) \otimes Map(y)|}{Map(y)}$ and $Map(z)$ is the mapping from z to tuples. Here, $\otimes$ is defined as a binary operator to return matching tuples between x and y.

Linguistic Metrics To analyze what it is about the captions themselves that differs under the various models, with respect to our aim of injecting information about emotional states of the faces in images, we first extracted all generated adjectives, which are tagged using the Stanford part-of-speech tagger software [168]. Perhaps surprisingly, emotions do not manifest themselves in the adjectives in our models: the adjectives used by all systems are essentially the same.

To investigate this further, we took the NRC emotion lexicon[4] [169] and examined the occurrence of words in the captions that also appeared in the lexicon. This widely-used lexicon is characterised as "a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust)" whose labels have been manually annotated through crowd-sourcing. The labels are based on word associations — annotators were asked "which emotions are associated with a target term" — rather than whether the word *embodies* an emotion; the lexicon thus contains a much larger set of words than is useful for our purposes. (For example, the most frequent word overall in the reference captions that appears in the lexicon is *young*, which presumably has some positive

[4] https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

emotional associations.) In addition, the set of emotions used in lexicon labels does not exactly correspond to our set. We therefore do not propose to use this lexicon purely automatically, but instead to help in understanding the use of emotion-related words.

Among the reference captions, as noted above the most frequent word from the emotion lexicon was *young*, followed by *white*, *blue* and *black*; all of these presumably have some emotional association, but do not generally embody an emotion. The first word embodying the expression of an emotion is the verb *smiling*, at rank 8, with other similar verbs following closely (e.g. *laughing*, *enjoying*). The highest ranked emotion-embodying adjective is *happy* at rank 26, with a frequency of around 15% of that of *smiling*; other adjectives were much further behind. It is clear that verbs form a more significant expression of emotion in this particular dataset than do adjectives.

To come up with an overall quantification of the different linguistic properties of the generated captions under the models, we therefore focussed our investigation on the differences in distributions of the generated verbs. To do this, we calculated three measures. The first is entropy (in the information-theoretic sense), which can indicate which distributions are closer to deterministic and which are more spread out (with a higher score indicating more spread out): in our context, it will indicate the amount of variety in selecting verbs. We calculated entropy using the standard Equation 4.26.

$$\text{Entropy} = -\sum_{1 \leq i \leq V} p(v_i) \times \log_2(p(v_i)) \tag{4.26}$$

where V indicates the number of the unique generated verbs and $p(v_i)$ is the probability of each generated verb (v_i), estimated as the Maximum Likelihood Estimate from the sample.

As a second measure, we looked at the four most frequent verbs (Top$_4$), which are the same for all models (*is*, *sitting*, *are*, *standing*) — these are verbs with relatively little semantic content, and for the most part act as syntactic props for the content words of the sentence. The amount of probability mass left beyond those four verbs is another indicator of variety in verb expression.

The two measures above are concerned only with variety of verb choice and not with verbs linked specifically to emotions or facial expressions. For a third measure, therefore, we look at selected individual verbs linked to actions that relate to facial emotion expression, either direct or indirect. Our measure is the rank of the selected verb among all those chosen by a model; higher (i.e. lower-numbered) ranked verbs mean that the model more strongly prefers this verb. Our selected verbs are among those that ranked highly in the reference captions and also appeared in the emotion lexicon.

4.3.2 Systems for Comparison

The starting points for our Face-Cap and Face-Attend models are Show-Attend-Tell [7] and Up-Down-Captioner [8], respectively. We therefore use these models, trained on the Flickr-Face11K dataset, as baselines to examine the effect of adding facial expression information. We call these baseline models Show-Att-Tell and Up-Down. (Moreover, Anderson *et al.* [8] has the state-of-the-art results for image captioning.)

We further look at two additional models to investigate the impact of the face loss function in using the facial encoding in different schemes. We train the Face-Cap-Repeat model, which uses the facial encoding in every time step, without calculating the face loss function (Equation 4.7); we refer to this (following the terminology of You *et al.* [48]) as the Step-Inject model. The Face-Cap-Memory model, which applies the facial encoding in the initial time step, is also modified in the same way; we refer to this as the Init-Flow model.

4.3.3 Implementation Details

The size of the word embedding layer, initialized via a uniform distribution, is set to 300 except for Up-Down and Joint-Face-Att which is set to 512. We fixed 512 dimensions for the memory cell and the hidden state in this work. We use the mini-batch size of 100 and the initial learning rate of 0.001 to train each image captioning model except Up-Down and Joint-Face-Att where we set the mini-batch size to 64 and the initial learning rate to 0.005. We used different parameters for Up-Down and Joint-Face-Att in comparison with other models because using similar parameters led to worse results for all models. The Adam optimization algorithm [170] is used for optimizing all models. During the training phase, if the model does not have an improvement in METEOR score on the validation set in two successive epochs, we divide the learning rate by two (the minimum learning rate is set to 0.0001) and the previous trained model with the best METEOR is reloaded. This method of learning rate decay is inspired by Wilson *et al.* [171], who advocated tuning the learning rate decay for Adam. In addition to learning rate decay, METEOR is applied to select the best model on the validation set because of a reasonable correlation between METEOR and human judgments [167]. Although SPICE can have higher correlations with human judgements, METEOR is quicker to calculate than SPICE, which requires dependency parsing, and so more suitable for a training criterion. The epoch limit is set to 30. We use the same vocabulary size and visual features for all models. λ and β_1 in Equation 4.14 are empirically set to 0.8 and 0.2, respectively. β_2 in Equation 4.20 is also set to 0.4. Multilayer perceptrons in Equation 4.6, 4.13 and 4.19 use tanh as an activation function.

4.4 Results

4.4.1 Overall Metrics

The FlickrFace11K splits are used for training and evaluating all image captioning models in this paper. Table 4.1 summarizes the results on the FlickrFace11K test set. Dual-Face-Att and Joint-Face-Att outperform other image captioning models using all the evaluation metrics. For example, Dual-Face-Att achieves 17.6 for BLEU-4 which is 1.9 and 0.4 points better that Show-Att-Tell (the first baseline model) and Face-Cap-Memory (the best of the Face-Cap models), respectively. Joint-Face-Att also achieves a BLEU-4 score of 17.7 which is 0.4 better than Up-Down, the baseline model it builds on, and 0.5 better than Face-Cap-Memory. Dual-Face-Att and Joint-Face-Att show very close results, with Dual-Face-Att demonstrating a couple of larger gaps in performance, in the BLEU-1 and ROUGE-L metrics. Among the Face-Cap models, Face-Cap-Memory is clearly the best.

4.4.2 Linguistic Metrics

Table 4.2 shows that Dual-Face-Att can generate the most diverse distribution of the verbs compared to other models because it has the highest Entropy. It also shows that Dual-Face-Att

Table 4.1: The results of different image captioning models (%) on FlickrFace11K test split. B-N is the BLEU-N metric. The best performances are bold.

Model	**B-1**	**B-2**	**B-3**	**B-4**	**METEOR**	**ROUGE-L**	**CIDEr**	**SPICE**
Show-Att-Tell	56.0	35.4	23.1	15.7	17.0	43.7	21.9	9.3
Up-Down	57.9	37.3	25.0	17.3	17.5	45.1	24.4	10.1
Step-Inject	58.4	37.6	24.8	17.0	17.5	45.0	22.8	9.9
Init-Flow	56.6	36.5	24.3	16.9	17.2	44.8	23.1	9.8
Face-Cap-Repeat	57.1	36.5	24.1	16.5	17.2	44.8	23.0	9.7
Face-Cap-Memory	58.9	37.9	25.1	17.2	17.4	45.5	24.7	10.0
Dual-Face-Att	**59.4**	**38.2**	25.4	17.6	**17.6**	**45.8**	**24.9**	10.1
Joint-Face-Att	58.6	38.1	**25.6**	**17.7**	**17.6**	45.5	24.8	**10.2**

has the lowest (best) proportion of the probability mass taken up by Top$_4$, leaving more for other verbs. In contrast to the results of the standard image captioning metrics shown in Table 4.1, Dual-Face-Att and Joint-Face-Att show very different behaviour: Dual-Face-Att is clearly superior. Among the Face-Cap models, as for the overall metrics, Face-Cap-Memory is the best, and is in fact better than Joint-Face-Att. (As a comparison, we also show Entropy and Top$_4$ for all reference captions (5 human-generated captions per image): human-generated captions are still much more diverse than the best models.)

Table 4.3 shows a sample of verbs, explained as the third measure in §4.3.1, such as *singing*, *reading* and *laughing*. The baseline Show-Att-Tell model ranks all of those relatively low, where our other baseline Up-Down and our models incorporating facial expressions do better. Only Face-Cap-Memory (the best of our Face-Cap models by overall metrics) and our Face-Attend models manage to use verbs like *laughing* and *reading*.

In Figure 4.6, we compare some generated captions by different image captioning models using four representative images. The first one shows that Dual-Face-Att correctly uses *smiling* and *laughing* to capture the emotional content of the image. Step-Inject, Init-Flow, Face-Cap-Repeat and Face-Cap-Memory are also successful in generating *smiling* for the image. For the second sample, Dual-Face-Att and Joint-Face-Att use the relevant verb *singing* to describe the image, while other models cannot generate the verb. Similarly, Dual-Face-Att

Table 4.2: The Entropies of all generated verbs and the probability mass of the Top$_4$ generated verbs (*is*, *are*, *sitting*, and *standing*). Reference means the ground-truth captions.

Model	**Entropy**	**Top$_4$**
Reference	6.9963	32.63%
Show-Att-Tell	2.7864	77.05%
Up-Down	2.7092	79.24%
Step-Inject	2.9059	74.80%
Init-Flow	2.6792	78.78%
Face-Cap-Repeat	2.7592	77.68%
Face-Cap-Memory	2.9306	73.65%
Dual-Face-Att	**3.0154**	**71.14%**
Joint-Face-Att	2.8074	77.69%

Table 4.3: Comparison of different image captioning models in ranking example generated verbs. These verbs are mostly selected from the emotion lexicon and are highly ranked in the reference captions. Higher ranks mean better results.

Model	**Smiling**	**Looking**	**Singing**	**Reading**	**Eating**	**Laughing**
Reference	11	10	27	35	24	40
Show-Att-Tell	19	n/a	15	n/a	24	n/a
Up-Down	14	**13**	9	n/a	15	n/a
Step-Inject	11	18	10	n/a	15	n/a
Init-Flow	10	21	12	n/a	14	n/a
Face-Cap-Repeat	12	20	9	n/a	14	n/a
Face-Cap-Memory	**9**	18	15	22	**13**	27
Dual-Face-Att	14	16	9	19	19	25
Joint-Face-Att	15	**13**	**8**	**15**	17	**23**

generates the verb *reading* for the third image. Moreover, most models can correctly generate *smiling* for the forth image except Show-Att-Tell and Up-Down which do not use the facial

Image has been removed due to copyright reasons

SAT: Two women and a man are posing for a picture.

UD: A group of people are posing for a picture.

SI: Two men and a woman are smiling.

IF: Two men and a woman are smiling at the camera.

FR: Two men and a woman are smiling.

FM: Two men and a woman are smiling at a camera.

DFA: Three women are smiling and laughing.

JFA: A group of people are posing for a picture.

Image has been removed due to copyright reasons

SAT: A woman with a black shirt and black pants is standing in front of a microphone.

UD: A man in a black shirt and a woman in a black shirt and a woman in a black shirt.

SI: A man with a beard and a beard is playing a guitar.

IF: A man in a black shirt and a black hat is playing a guitar.

FR: A woman with a black shirt and a black hat is holding a microphone.

FM: A woman in a black shirt is holding a microphone.

DFA: A woman in a black dress is singing into a microphone.

JFA: A woman in a black shirt is singing into a microphone.

Image has been removed due to copyright reasons

SAT: A man in a white shirt is sitting at a table with a computer.

UD: A man in a yellow shirt is sitting at a table with a book in his lap.

SI: A man in a yellow shirt is working on a computer.

IF: A woman in a yellow shirt is sitting at a table with a computer.

FR: A man in a yellow shirt is sitting at a table with a computer.

FM: A woman in a yellow shirt is working on a computer.

DFA: A woman in a yellow shirt is reading a book.

JFA: A man in a yellow shirt is working on a computer.

Image has been removed due to copyright reasons

SAT: Two young girls are sitting in a chair.

UD: A woman in a striped shirt is holding a small child in a striped shirt.

SI: A woman with a brown shirt and a blond woman in a blue shirt are smiling.

IF: A woman with a white shirt and a young girl in a blue shirt are sitting in a chair.

FR: A woman and a woman are smiling at the camera.

FM: A woman and a young girl are smiling.

DFA: A man and a woman are smiling at the camera.

JFA: A woman in a striped shirt is smiling at the camera.

Figure 4.6: Example generated captions using SAT (Show-Att-Tell), UD (Up-Down) SI (Step-Inject), IF (Init-Flow), FR (Face-Cap-Repeat), FM (Face-Cap-Memory), DFA (Dual-Face-Att) and JFA (Joint-Face-Att) models.

information. Init-Flow also cannot generate *smiling* because it uses the facial information only at initial step which provides a weak emotional signal for the model. Here, Dual-Face-Att can generate the most accurate caption (“A man and a woman are smiling at the camera”) for the image, while other models generate some errors. For example, Face-Cap-Memory generates

Image has been removed due to copyright reasons

SAT: A man in a black shirt is sitting at a table with a woman in a black shirt and a.

UD: A man in a black shirt is sitting at a table with a book in his hand.

SI: A woman in a black shirt is sitting at a table with a glass of wine in a kitchen.

IF: A woman in a black shirt is sitting at a table with a computer.

FR: A woman in a black shirt is standing in front of a bar with a man in a black shirt.

FM: A man in a black shirt is sitting at a table with a glass of wine.

DFA: Two women are sitting at a table with a laptop and a laptop.

JFA: A man in a black shirt is sitting at a table with a woman in a blackshirt.

Image has been removed due to copyright reasons

SAT: Two men are sitting on a rock and one is holding a large tree.

UD: A little boy in a white shirt is holding a small child in his arms.

SI: A woman in a black shirt is holding a child in a blue dress.

IF: A young boy and a boy are sitting on a rock.

FR: A young boy in a blue shirt is holding a small child in a field.

FM: A young boy and a boy are sitting on a rock with a dog.

DFA: A young boy and a boy are sitting on a rock and smiling.

JFA: A young man in a black shirt is holding a small child.

Figure 4.7: Example generated captions including some errors. (SAT (Show-Att-Tell), UD (Up-Down) SI (Step-Inject), IF (Init-Flow), FR (Face-Cap-Repeat), FM (Face-Cap-Memory), DFA (Dual-Face-Att) and JFA (Joint-Face-Att))

"A woman and a young girl are smiling", which does not describe the man in the image.

Figure 4.7 shows two examples including some improper words/phrases. For the first image, Dual-Face-Att generates "Two women are sitting at a table with a laptop and a laptop". This caption wrongly includes *laptop* and *two women*. Here, other models are more successful in generating relevant image captions. For the second image, Joint-Face-Att incorrectly generates "holding a small child" and Face-Cap-Memory wrongly generates "a dog".

4.5 Summary

In this chapter, we have presented several image captioning models incorporating information from facial features. The joint image captioning models, Dual-Face-Att and Joint-Face-Att models, learned to apply both facial features and visual content to generate image captions that produce the highest results as measured by standard metrics on the FlickrFace11K dataset. They use attention mechanisms to adaptively take into account the presented facial expressions in images to generate more descriptive image captions. The example generated captions show that the models can generate more diverse image captions in addition to having a higher ability to employ facial expression features to describe images.

Moreover, our proposed approaches applying facial expression features achieved more effective results in comparison with our baseline models without the features. This shows the effectiveness of applying the features in image captioning in response to **RQ 2** discussed in Chapter 1. As an exploration of which ways are better to control image captioning using facial expression features, we injected the features at different time steps of a caption generation process. Among the models using the one-hot encoding version of the features, Face-Cap-Memory achieved the best results by injecting the features only at initial time step and employing a specific loss function to remember the features. Among all models, Dual-Face-Att and Joint-Face-Att achieved the best results by applying an attention-based version of the features at every time step. This provides an adaptive set of the features in contrast to applying a fixed one-hot encoding.

Part II

Image Captioning with Stylistic Information

5

Towards Generating Stylized Image Captions via Adversarial Training

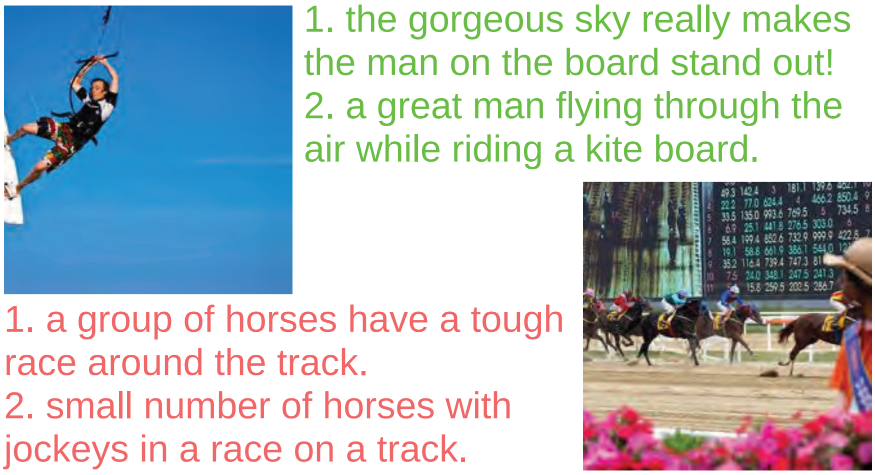
While most image captioning aims to generate objective descriptions of images, the last few years have seen work on generating visually grounded image captions which have a specific style (e.g., incorporating positive or negative sentiment). However, because the stylistic component is typically the last part of training, current models usually pay more attention to the style at the expense of accurate content description. In addition, there is a lack of variability in terms of the stylistic aspects. To address these issues, we propose an image captioning model called Attend-GAN which has two core components: first, an attention-based caption generator to strongly correlate different parts of an image with different parts of a caption; and second, an adversarial training mechanism to assist the caption generator to add diverse stylistic components to the generated captions. Because of these components, Attend-GAN can generate correlated captions as well as more human-like variability of stylistic patterns.[1] In this chapter, we study style-bearing image captioning using two steps including training on a factual image caption dataset and then training on a style-bearing image caption dataset. In the following chapter, we will propose an style-bearing image captioning model trained in end-to-end fashion combining both factual and style-bearing datasets.

5.1 Introduction

Deep learning has facilitated the task of supplying images with captions. Current image captioning models [1, 7, 8] have gained considerable success due to powerful deep learning architectures and large image-caption datasets including the MSCOCO dataset [160]. These models mostly aim to describe an image in a factual way. However, when humans produce descriptions of images, they often go beyond the purely factual, and incorporate some subjective properties like sentiment or stylistic effects, depending on broader context or goals; a widely discussed example was the photo of Donald Trump at the 2018 G7 Summit where he is seated with arms crossed in front of Angela Merkel; various commentators described the photo in terms intended to be negative ("eyes glaring") or positive ("alpha male"). Researchers in image captioning have similarly proposed models that allow the generation of captions with a particular style [30, 49] or sentiment [2, 31] such as positive and negative sentiment, as in the captions of Figure 5.1. Users often find such captions more expressive and more attractive [30]; they have the practical purpose of enhancing the engagement level of users in social applications (e.g., chatbots) [23], and can assist people to make interesting image captions in social media content [30]. Moreover, Mathews *et al.* [2] found that they are more common in the descriptions of online images, and can have a role in transferring visual content clearly [49]. We have given an overview of this in §2.4, but recap key points here.

[1] The content of this chapter is based on the following publication:

Omid Mohamad Nezami, Mark Dras, Stephen Wan, Cecile Paris, Len Hamey (2019). Towards Generating Stylized Image Captions via Adversarial Training. *Proceedings of the 2019 Pacific Rim International Conference on Artificial Intelligence (PRICAI 2019),* Cuvu, Fiji.



1. the gorgeous sky really makes the man on the board stand out!
2. a great man flying through the air while riding a kite board.

1. a group of horses have a tough race around the track.
2. small number of horses with jockeys in a race on a track.

Figure 5.1: Examples of positive (green) and negative (red) captions.

In stylistically enhanced descriptions, the content of images should still be reflected correctly. Moreover, the descriptions should fluently include stylistic words or phrases. To meet these criteria, previous models have used two-stage training: first, training on a large factual dataset to describe the content of an image; and then training on a small stylistic dataset to apply stylistic properties to a caption. The models have different strategies for integrating the learned information from the datasets. Gan *et al.* [30] proposed a new type of LSTM network, factored LSTM, to learn both factual and stylistic information. The factored LSTM has three matrices instead of one multiplied to the input caption: all matrices are learned on the factual dataset to preserve the factual aspect of the input caption and one is learned on the stylistic dataset to transfer the style aspect of the input caption. However, there is no specific mechanism to regulate and switch between factual and stylistic information. To solve this issue, SentiCap has two Long Short-Term Memory (LSTM) networks: one learns from a factual dataset and the other one learns from a stylistic dataset [2]. It also detects the current word's sentiment level using the ground-truth sentiment label of each word. Then, it uses the sentiment level to weight the probability distribution of the predicted word by two LSTMs to regulate their predictions. However, these ground-truth word-level labels are not available for all stylistic datasets such as the dataset of Gan *et al.* [30]. To address this, Chen *et al.* [31] applied an attention-based model which is similar to the factored LSTM, but it has an attention mechanism to differentiate attending to the factual and stylistic information of the input caption.

Since the stylistic dataset is usually small, preserving the correlations between images and captions as well as generating a wide variety of stylistic patterns is difficult even with approaches proposed by Mathews *et al.* [2] and Chen *et al.* [31] to regulate between factual and stylistic information. An imperfect caption from the system of Mathews *et al.* [2] — "a dead man doing a clever trick on a skateboard at a skate park" — illustrates the problem: the man is not actually dead; this is just a frequently used negative adjective.

Recently, Mathews *et al.* [49] dealt with this by applying a large stylistic dataset to separate the semantic and stylistic aspects of the generated captions. However, evaluation in this work was more difficult because the dataset includes stylistic captions which are not aligned to images. To address this challenge without any large stylistic dataset, we propose Attend-GAN, an image captioning model using an attention mechanism and a Generative Adversarial Network (GAN); our particular goal is to better apply stylistic information in the sort of two-stage architecture in previous work. Similar to the previous work, we first train a caption generator on a large factual dataset, although Attend-GAN uses an attention-based version attending to different image regions in the caption generation process [8]. Because of this, each word of a generated caption is conditioned upon a relevant fine-grained region of the corresponding image, ensuring a direct correlation between the caption and the image. Then we train a caption discriminator to distinguish between captions generated by our caption generator, and real captions, generated by humans. In the next step, on a small stylistic dataset, we implement an adversarial training mechanism to guide the generator to generate sentiment-bearing captions. To do so, the generator is trained to fool the discriminator by generating correlated and highly diversified captions similar to human-generated ones. The discriminator also periodically improves itself to further challenge the generator. Because GANs are originally designed to face continuous data distributions not discrete ones like texts [131], we use a gradient policy [133] to guide our caption generator using the rewards received from our caption discriminator for the next generated word.

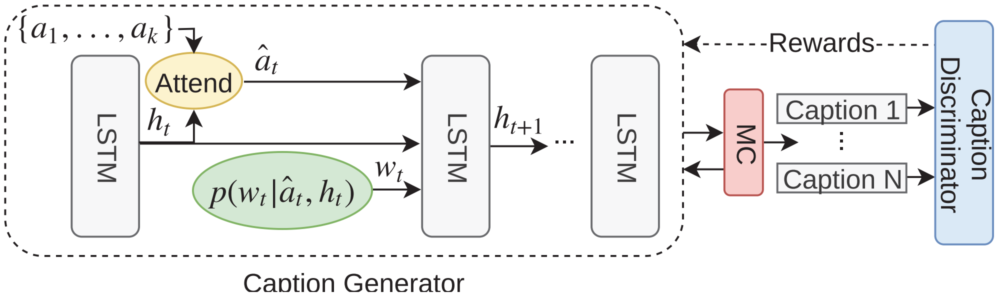


Figure 5.2: The architecture of the Attend-GAN model. $\{\boldsymbol{a}_1, ..., \boldsymbol{a}_K\}$ are spatial visual features generated by ResNet-152 network. Attend and MC modules are our attention mechanism and Monte Carlo search, respectively.

5.2 Attend-GAN Model

The purpose of our image captioning model is to generate sentiment-bearing captions. Our caption generator employs an attention mechanism, described in §5.2.1, to attend to fine-grained image regions $\boldsymbol{a} = \{\boldsymbol{a}_1, ..., \boldsymbol{a}_K\}, \boldsymbol{a}_i \in \mathbb{R}^D$, where the number of regions is K with D dimensions, in different time steps so as to generate an image caption $\boldsymbol{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}, \boldsymbol{x}_i \in \mathbb{R}^N$, where the size of our vocabulary is N and the length of the generated caption is T. We also propose a caption discriminator, explained in §5.2.2, to distinguish between the generated captions and human-produced ones. We describe our training in §5.2.3. Our proposed model is called Attend-GAN (Figure 5.2).

5.2.1 Caption Generator

The goal of our caption generator $G_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \hat{\boldsymbol{a}}_t)$ is to generate an image caption to achieve a maximum reward value from our caption discriminator $D_{\boldsymbol{\phi}}(\boldsymbol{x}_{1:T})$, where $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are the parameters of the generator and the discriminator, respectively. The objective function of the generator, which is dependent on the discriminator, is to minimize:

$$L_1(\boldsymbol{\theta}) = \sum_{1 \leq t \leq T} G_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \hat{\boldsymbol{a}}_t).Z_{D_{\phi}}^{G_{\theta}}(\boldsymbol{x}_{1:t}) \tag{5.1}$$

where $Z_{D_{\phi}}^{G_{\theta}}(\boldsymbol{x}_{1:t})$ is the reward value of the partially generated sequence, $\boldsymbol{x}_{1:t}$, and is estimated using the discriminator [2]. This is inspired by the literature of reinforcement learning explained in Chapter 2. Here, Z is an instantaneous reward at each time step in generating the image caption. It can be interpreted as a score value that $\boldsymbol{x}_{1:t}$ is real. Since the discriminator can only generate a reward value for a complete sequence, Monte Carlo (MC) search is applied, which uses the generator to roll out the remaining part of the sequence at each time step. We apply MC search N times, and calculate the average reward (to decrease the variance of the next generated words):

$$Z_{D_\phi}^{G_\theta}(\boldsymbol{x}_{1:t}) = \begin{cases} \frac{1}{N}\sum_{n=1}^{N} D_\phi(\boldsymbol{x}_{1:T}^{n}),\ \boldsymbol{x}_{1:T}^{n} \in MC_{G_\theta}(\boldsymbol{x}_{1:t;N}) & \text{if } t < T \\ D_\phi(\boldsymbol{x}_{1:t}) & \text{if } t = T \end{cases} \tag{5.2}$$

$\boldsymbol{x}_{1:T}^{n}$ is the n-th MC completed sequence at current time step t. In addition to Equation 5.1, we calculate the maximum likelihood estimation (MLE) of the generated word with respect to the attention-based content ($\hat{\boldsymbol{a}}_t$) and the hidden state ($\boldsymbol{h}_t$) at the current time of our LSTM, which is the core of our caption generator, as the second objective function:

$$L_2(\theta) = -\sum_{1\le t\le T} \log(\boldsymbol{p}_w(\boldsymbol{x}_t \,|\, \hat{\boldsymbol{a}}_t, \boldsymbol{h}_t)) + \lambda_1 \sum_{1\le k\le K} (1 - \sum_{1\le t\le T} a_{tk})^2 \tag{5.3}$$

$\boldsymbol{p}_w$ is calculated using a multilayer perceptron with a softmax layer on its output and indicates the probabilities of the possible generated words:

$$\boldsymbol{p}_w(\boldsymbol{x}_t \,|\, \hat{\boldsymbol{a}}_t, \boldsymbol{h}_t) = \text{softmax}(\hat{\boldsymbol{a}}_t \boldsymbol{W}_a + \boldsymbol{h}_t \boldsymbol{W}_h + \boldsymbol{b}_w) \tag{5.4}$$

$\boldsymbol{W}_x$ and $\boldsymbol{b}_w$ are the learned weights and biases. The last term in Equation 5.3 is to encourage our caption generator to equally consider diverse regions of the given image at the end of the caption generation process. λ_1 is a regularization parameter. $\boldsymbol{h}_t$ is calculated using our

[2] Equation 5.1 does not have anything particular to capture sentiment-bearing information; however, it is effective to generate realistic captions. In the second stage of training, apart from our model learning from sentiment-bearing image captions, the discriminator learns from the captions to generate rewards.

LSTM:

$$
\begin{aligned}
\boldsymbol{i}_t &= \sigma(\boldsymbol{H}_i \boldsymbol{h}_{t-1} + \boldsymbol{W}_i \boldsymbol{w}_{t-1} + \boldsymbol{A}_i \hat{\boldsymbol{a}}_t + \boldsymbol{b}_i) \\
\boldsymbol{f}_t &= \sigma(\boldsymbol{H}_f \boldsymbol{h}_{t-1} + \boldsymbol{W}_f \boldsymbol{w}_{t-1} + \boldsymbol{A}_f \hat{\boldsymbol{a}}_t + \boldsymbol{b}_f) \\
\boldsymbol{g}_t &= \tanh(\boldsymbol{H}_g \boldsymbol{h}_{t-1} + \boldsymbol{W}_g \boldsymbol{w}_{t-1} + \boldsymbol{A}_g \hat{\boldsymbol{a}}_t + \boldsymbol{b}_g) \\
\boldsymbol{o}_t &= \sigma(\boldsymbol{H}_o \boldsymbol{h}_{t-1} + \boldsymbol{W}_o \boldsymbol{w}_{t-1} + \boldsymbol{A}_o \hat{\boldsymbol{a}}_t + \boldsymbol{b}_o) \\
\boldsymbol{c}_t &= \boldsymbol{f}_t \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \boldsymbol{g}_t \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \tanh(\boldsymbol{c}_t)
\end{aligned}
\tag{5.5}
$$

Here, $\boldsymbol{i}_t$, $\boldsymbol{f}_t$, $\boldsymbol{g}_t$, $\boldsymbol{o}_t$, and $\boldsymbol{c}_t$ are the parameters of the LSTM and represent input, forget, modulation, output, and memory gates, respectively. $\boldsymbol{w}_{t-1}$ is the embedded previous word in M dimensions, $\boldsymbol{w}_x \in \mathbb{R}^M$. $\boldsymbol{H}_x, \boldsymbol{W}_x, \boldsymbol{A}_x$, and $\boldsymbol{b}_x$ are learned weights and biases; and σ is the Sigmoid function. Using $\boldsymbol{h}_t$, our soft attention module generates unnormalized weights $e_{j,t}$ for each image region $\boldsymbol{a}_j$. Then, the weights are normalized using a softmax layer, $\boldsymbol{e}'_t$:

$$
e_{j,t} = \boldsymbol{W}_e^T \tanh(\boldsymbol{W}'_a \boldsymbol{a}_j + W'_h \boldsymbol{h}_t), \boldsymbol{e}'_t = \mathrm{softmax}(\boldsymbol{e}_t) \tag{5.6}
$$

$\boldsymbol{W}_e^T$ and $\boldsymbol{W}'_x$ are our trained weights. Finally, $\hat{\boldsymbol{a}}_t$, our attention-based content, is calculated using Equation 5.7:

$$
\hat{\boldsymbol{a}}_t = \sum_{1 \le j \le K} e'_{j,t} \boldsymbol{a}_j \tag{5.7}
$$

During the adversarial training, the objective function of the caption generator is a combination of Equation 5.1 and Equation 5.3:

$$
L_G(\theta) = \lambda_2 L_1(\theta) + L_2(\theta) \tag{5.8}
$$

λ_2 is a balance parameter. The discriminator cannot be learned effectively from a random initialization of the generator; we therefore pretrain the generator with the MLE objective function[3]:

$$
L_G(\theta) = L_2(\theta) \tag{5.9}
$$

[3] We cannot train the generator with Equation 5.8 from the beginning since the generator generates poor captions and the discriminator cannot learn from the captions to offer valid rewards in Equation 5.1.

5.2.2 Caption Discriminator

Our caption discriminator is inspired by the Wasserstein GAN (WGAN) [141] which is an improved version of the GAN [131]. The WGAN generates continuous values and solves the problem of the GAN generating non-continuous outputs leading to some training difficulties (e.g. vanishing gradients). The objective function of our WGAN is:

$$L_D(\boldsymbol{\phi}) = \mathbb{E}_{x \sim \mathbb{P}_H}[D_\phi(\boldsymbol{x})] - \mathbb{E}_{\overline{\mathrm{x}} \sim \mathbb{P}_G}[D_\phi(\overline{\mathbf{x}})] \tag{5.10}$$

where $\boldsymbol{\phi}$ are the parameters of the discriminator (D_ϕ); $\mathbb{P}_H$ is the set of the generated captions by humans; and $\mathbb{P}_G$ is the set of the generated captions by the generator. D_ϕ is implemented via a Convolutional Neural Network (CNN) that calculates the score value of the input caption. To feed a caption to our CNN model, we first embed all words in the caption into M embedding dimensions, $\{\boldsymbol{w}'_1, \ldots, \boldsymbol{w}'_T\}, \boldsymbol{w}'_i \in \mathbb{R}^M$, and build a 2-dimensional matrix for the caption, $\boldsymbol{S} \in \mathbb{R}^{T \times M}$ [133]. Our CNN model includes Convolutional (Conv.) layers with P different kernel sizes $\{\boldsymbol{k}_1, \ldots, \boldsymbol{k}_P\}, \boldsymbol{k}_i \in \mathbb{R}^{C \times M}$, where C indicates the number of the words ($C \in [1, T]$). Applying each Conv. layer to $\boldsymbol{S}$ results a number of feature maps, $\boldsymbol{v}_{ij} = \boldsymbol{k}_i \otimes \boldsymbol{S}_{j:j+C-1} + \boldsymbol{b}_j$, where $\otimes$ is a convolution operation and b_j is a bias vector. We apply a batch normalization layer [163], and a nonlinearity, a rectified linear unit (ReLU), respectively. Then, we apply a max-pooling layer, $\boldsymbol{v}^*_i = \max \boldsymbol{v}_{ij}$. Finally, a fully connected layer is applied to output the score value of the caption. The weights of our CNN model are clipped to be in a compact space.

5.2.3 Attend-GAN Training

As shown in Algorithm 1, we first pre-train our caption generator for a specific number of epochs. Then, we apply the best generator model to generate sample captions. The real captions are selected from the ground truth. In Step 3, our caption discriminator is pre-trained using a combination of the generated and real captions for a specific number of epochs. Here, both the caption generator and discriminator are pre-trained on a factual dataset. In Step 4, we start our adversarial training on a sentiment-bearing dataset with positive or negative sentiment. We continue the training of the caption generator and discriminator for g-steps and d-steps, respectively. Using this mechanism, we improve both the caption generator

Algorithm 1: Attend-GAN Training Mechanism.

```
Pre-train the caption generator ($G_\theta$) using Equation 5.9.
Use $G_\theta$ to generate sample captions $\mathbb{P}_G$ and select ground-truth captions $\mathbb{P}_H$.
Pre-train the caption discriminator ($D_\phi$) using Equation 5.10 and the combination of
  $\mathbb{P}_G$ and $\mathbb{P}_H$.
repeat
    for $g$ steps do
        Apply $G_\theta$ to generate image captions.
        Calculate $Z_{D_\phi}^{G_\theta}$ using Equation 5.2.
        Update $\boldsymbol{\theta}$, the parameters of $G_\theta$, using Equation 5.8.
    end for
    for $d$ steps do
        Generate sample captions $\mathbb{P}_G$ by $G_\theta$ and select human-generated captions $\mathbb{P}_H$.
        Update $\boldsymbol{\phi}$, the parameters of $D_\phi$, using Equation 5.10.
    end for
until Attend-GAN converges
```

and discriminator. Here, the caption generator applies the received rewards from the caption discriminator to update its parameters using Equation 5.8.

5.3 Experimental Setup

5.3.1 Datasets

Microsoft COCO Dataset We use the MSCOCO image-caption dataset [160] to train our models. Specifically, we use the training set of the dataset including 82K+ images and 413K+ captions.

SentiCap Dataset To add sentiment to the generated captions, our models are trained on the SentiCap dataset [2] including sentiment-bearing image captions. The dataset has two separate sections of sentiments: *positive* and *negative*. 2,873 captions paired with 998 images are for training and 2019 captions paired with 673 images are for testing in the positive section. 2,468 captions paired with 997 images are for training and 1,509 captions paired with 503 images are for testing in the negative section.

5.3.2 Evaluation Metrics

Overall Metrics Attend-GAN is evaluated using standard image captioning metrics: METEOR [166], BLEU [164], CIDEr [100] and ROUGE-L [165]. SPICE has not previously been used in the literature; however, it is reported for future comparisons because it has shown a close correlation with human-based evaluations [167]. Larger values of these metrics indicated better results.

Linguistic Metrics Most work in image captioning considers only the standard overall metrics above. However, we are also interested in understanding the linguistic properties of our captions related to sentiment. To analyze the quality of language generated by our models, we extract all generated adjectives using the Stanford part-of-speech tagger software [168], and identify the adjectives with strong sentiment values which are found in the list of the adjective-noun pairs (ANPs) of the SentiCap dataset (for example, *cuddly*, *sunny*, *shy* and *dirty*). Then, we calculate Entropy of the distribution of these adjectives as a measure of variety in lexical selection (higher scores mean more variety) using Equation 5.11.

$$\text{Entropy} = -\sum_{1 \leq j \leq U} \log_2[p(A_j)] \times p(A_j) \tag{5.11}$$

where $p(A_j)$ is the probability of the adjective (A_j) and U indicates the number of all unique adjectives. Moreover, we calculate the total probability mass of the four most frequent adjectives (Top_4) generated by our models. Here, lower values mean that the model allocates more probability to other generated adjectives, also indicating greater variety.

5.3.3 Models for Comparison

Our models are compared with a range of baseline models from Mathews *et al.*[2]: CNN+RNN, which is only trained using the MSCOCO dataset; ANP-Replace, which adds the most common adjectives to a randomly chosen noun; ANP-Scoring, which applies multi-class logistic regression to select an adjective for the chosen noun; RNN-Transfer, which is CNN+RNN fine-tuned on the SentiCap dataset; and their key system SentiCap, which uses two LSTM modules to learn from factual and sentiment-bearing caption. We also compare with SF-LSTM+Adapt, which applies an attention mechanism to weight factual and sentiment-based information [31]. The results of all these models in Table 5.1 are obtained from the corresponding references. Moreover, we first train our attention-based model only on the factual dataset MSCOCO (we name this model Attend-GAN$_{-SA}$). This helps us to isolate the effect of applying the attention-based model without sentiment information. Second, we train our model additionally on the SentiCap dataset but without our caption discriminator (Attend-GAN$_{-A}$). This helps us to specify the effect of applying sentiment information in the model without the discriminator. Finally, we train our full model using the caption discriminator (Attend-GAN).

5.3.4 Implementation Details

Encoder In this work, we apply ResNet-152 [70] as our visual encoder model pre-trained using the ImageNet dataset [67]. In comparison with other CNN models, ResNet-152 has shown more effective results on different image-caption datasets [156]. We specifically use its Res5c layer to extract the spatial features of an image. The layer gives us $7 \times 7 \times 2048$ feature map converted to 49×2048 representing 49 semantic-based regions with 2048 dimensions.

Vocabulary Our vocabulary has 9703 words, coming form both the MSCOCO and SentiCap datasets, for all our models. Each word is embedded into a 300 dimensional vector.

Generator and Discriminator The size of the hidden state and the memory cell of our LSTM is set to 512. For the caption generator, we use the Adam function [170] for optimization and set the learning rate to 0.0001. We set the the size of our mini-batches to 64. To optimize the caption discriminator, we use the RMSprop solver [172] and clip the weights to $[-0.01, 0.01]$. The mini-batches are fixed to 80 for the discriminator. We apply Monte Carlo search 5 times (Equation 5.2). We set λ_1 and λ_2 to 1.0 and 0.1 in Equation 5.3 and 5.8, respectively. During the adversarial training, we alternate between Equation 5.8 and 5.10 to optimize the generator and the discriminator, respectively. We particularly operate a single gradient descent phase on the generator (g steps) and 3 gradient phases (d steps) on the discriminator every time. The models are trained for 20 epochs to converge. The METEOR metric is used to select the model with the best performance on the validation sets of positive and negative datasets of SentiCap because it has a close correlation with human judgments and is less computationally expensive than SPICE which requires dependency parsing [167].

5.4 Results

5.4.1 Overall Metrics

Comparison with the State-of-the-art All models in Table 5.1 used the same training/test folds of the SentiCap dataset to make them comparable. In comparison with the state-of-the-art, our full model (Attend-GAN) achieves the best results for all image captioning metrics in both positive and negative parts of the SentiCap dataset. We report the average results to show the average improvements of our models over the state-of-the-art model. Attend-GAN achieved large gains of 6.15, 6.45, 3.00, and 2.95 points with respect to the best previous model using BLEU-1, ROUGE-L, CIDEr and BLEU-2 metrics, respectively. Other metrics show smaller but still positive improvements.

Comparison with Our Baseline Models Our models are compared in Table 5.1 in terms of image captioning metrics. Attend-GAN outperforms Attend-GAN$_{-A}$ over all metrics across both positive and negative parts of the SentiCap dataset; the discriminator is thus an important part of the architecture. Attend-GAN outperforms Attend-GAN$_{-SA}$ for all metrics except, by a small margin, CIDEr and ROUGE-L. Recall that Attend-GAN$_{-SA}$ is trained only on the large MSCOCO (with many captions), and so is in a sense encouraged to have diverse captions; second-stage training for Attend-GAN$_{-A}$ and Attend-GAN leads to more focussed captions relevant to SentiCap. As CIDEr and ROUGE-L are the two recall-oriented

Table 5.1: The compared performances on different sections of SentiCap and their average. BLEU-N performance metric is shown by B-N. (The best performances are bold.)

Senti	Model	B-1	B-2	B-3	B-4	ROUGE-L	METEOR	CIDEr	SPICE
Pos	CNN+RNN	48.7	28.1	17.0	10.7	36.6	15.3	55.6	_
	ANP-Replace	48.2	27.8	16.4	10.1	36.6	16.5	55.2	_
	ANP-Scoring	48.3	27.9	16.6	10.1	36.5	16.6	55.4	_
	RNN-Transfer	49.3	29.5	17.9	10.9	37.2	17.0	54.1	_
	SentiCap	49.1	29.1	17.5	10.8	36.5	16.8	54.4	_
	SF-LSTM + Adap	50.5	30.8	19.1	12.1	38.0	16.6	60.0	_
	Attend-GAN$_{-SA}$	56.1	32.5	19.4	11.8	44.8	17.1	63.0	15.9
	Attend-GAN$_{-A}$	55.8	33.4	20.1	12.4	44.2	18.6	61.1	15.7
	Attend-GAN	56.9	33.6	20.3	12.5	44.3	18.8	61.6	15.9
Neg	CNN+RNN	47.6	27.5	16.3	9.8	36.1	15.0	54.6	_
	ANP-Replace	48.1	28.8	17.7	10.9	36.3	16.0	56.5	_
	ANP-Scoring	47.9	28.7	17.7	11.1	36.2	16.0	57.1	_
	RNN-Transfer	47.8	29.0	18.7	12.1	36.7	16.2	55.9	_
	SentiCap	50.0	31.2	20.3	13.1	37.9	16.8	61.8	_
	SF-LSTM + Adap	50.3	31.0	20.1	13.3	38.0	16.2	59.7	_
	Attend-GAN$_{-SA}$	55.4	32.4	19.4	11.9	44.4	17.0	63.4	15.6
	Attend-GAN$_{-A}$	54.7	32.6	20.4	12.9	43.2	17.7	60.4	16.1
	Attend-GAN	56.2	34.1	21.3	13.6	44.6	17.9	64.1	16.2
Avg	CNN+RNN	48.15	27.80	16.65	10.25	36.35	15.15	55.10	_
	ANP-Replace	48.15	28.30	17.05	10.50	36.45	16.25	55.85	_
	ANP-Scoring	48.10	28.30	17.15	10.60	36.35	16.30	56.25	_
	RNN-Transfer	48.55	29.25	18.30	11.50	36.95	16.60	55.00	_
	SentiCap	49.55	30.15	18.90	11.95	37.20	16.80	58.10	_
	SF-LSTM + Adap	50.40	30.90	19.60	12.70	38.00	16.40	59.85	_
	Attend-GAN$_{-SA}$	55.75	32.45	19.40	11.85	**44.60**	17.05	**63.20**	15.75
	Attend-GAN$_{-A}$	55.25	33.00	20.25	12.65	43.70	18.15	60.75	15.90
	Attend-GAN	**56.55**	**33.85**	**20.80**	**13.05**	44.45	**18.35**	62.85	**16.05**

Table 5.2: Entropy and Top_4 of the generated adjectives using different models.

Senti	Model	Entropy	Top_4
Pos	Attend-GAN$_{-SA}$	2.2457	93.33%
	Attend-GAN$_{-A}$	3.0324	72.11%
	Attend-GAN	3.5671	62.33%
Neg	Attend-GAN$_{-SA}$	2.2448	91.67%
	Attend-GAN$_{-A}$	4.1040	48.44%
	Attend-GAN	3.9562	50.51%
Avg	Attend-GAN$_{-SA}$	2.2453	92.50%
	Attend-GAN$_{-A}$	3.5682	60.28%
	Attend-GAN	**3.7617**	**56.42%**

metrics, they suffer in this two-stage process, illustrating the issue we noted in §5.1. The discriminator, however, removes almost all of this penalty, as well as boosting the other metrics beyond Attend-GAN$_{-SA}$. Furthermore, §5.4.2 will illustrate how Attend-GAN$_{-SA}$ produces unsatisfactory captions in terms of sentiment.

5.4.2 Linguistic Metrics

Table 5.2 shows that Attend-GAN achieves the best results on average for Entropy (highest score) and Top_4 (lowest) compared to other models, by a large margin with respect to Attend-GAN$_{-SA}$. It is not surprising that Attend-GAN$_{-SA}$ has the lowest variability of use of sentiment-bearing adjectives because it does not use the stylistic dataset. As demonstrated by the improvement of Attend-GAN over Attend-GAN$_{-A}$, the discriminator helps in generating a greater diversity of adjectives.

The top-10 adjectives generated by our models are shown in Table 5.3. *white* is generated for both negative and positive sections because they are common in both sections. Attend-GAN and Attend-GAN$_{-A}$ produce a natural ranking of sentiment-bearing adjectives for both sections. For example, these models rank *nice* as the most positive adjective, and *lonely* as the

Table 5.3: The top-10 adjectives that are generated by our models and are in the adjective-noun pairs of the SentiCap dataset.

Senti	Model	Top 10 Adjectives
Pos	Attend-GAN$_{-SA}$	white, black, small, blue, different, little, busy, _, _, _
	Attend-GAN$_{-A}$	nice, beautiful, happy, busy, great, sunny, good, cute, pretty, white
	Attend-GAN	nice, beautiful, happy, great, good, sunny, busy, white, pretty, delicious
Neg	Attend-GAN$_{-SA}$	black, white, small, blue, different, tall, little, _, _ , _
	Attend-GAN$_{-A}$	lonely, dead, broken, stupid, dirty, bad, cold, little, crazy, lazy
	Attend-GAN	lonely, stupid, broken, dirty, dead, cold, bad, white, crazy, little

most negative. As Attend-GAN$_{-SA}$ does not use the stylistic dataset, it generates a similar and limited ($<$ 10) range of adjectives for both.

Figure 5.3 shows sample sentiment-bearing captions generated by our models for the positive and negative sections of the SentiCap dataset. For instance, for the first two images, Attend-GAN correctly applies positive sentiments to describe the corresponding images (e.g., “nice street”, “tasty food”). Here, Attend-GAN$_{-A}$ also succeeds in generating captions with positive sentiments, but less well. In the third image, Attend-GAN uses “pretty woman” to describe the image which is better than the “beautiful court” of Attend-GAN$_{-A}$: for this image, all ground-truth captions have positive sentiment for the noun “girl” (e.g. “a beautiful girl is running and swinging a tennis racket”); none of them describes the noun “court” with a sentiment-bearing adjective as Attend-GAN$_{-A}$ does. For all images, since Attend-GAN$_{-SA}$ is not trained using the SentiCap dataset, it does not generate any caption with sentiment. For the fourth image, Attend-GAN generates “a group of stupid people are playing frisbee on a field”, applying “stupid people” to describe the image negatively. Here, one of the ground-truth captions exactly includes “stupid people” (“two stupid people in open field watching yellow tent blown away”). Attend-GAN$_{-A}$, like our flawed example from §5.1, refers instead inaccurately to a dead man. For the fifth image (as for the first image), Attend-GAN has incorporates more (appropriate) sentiment in comparison to Attend-GAN$_{-A}$. It generates “rough hill” and “cold day”, while Attend-GAN$_{-A}$ only generates the former. It also uses “skier” which is more appropriate than “person”. In the last image, Attend-GAN adds

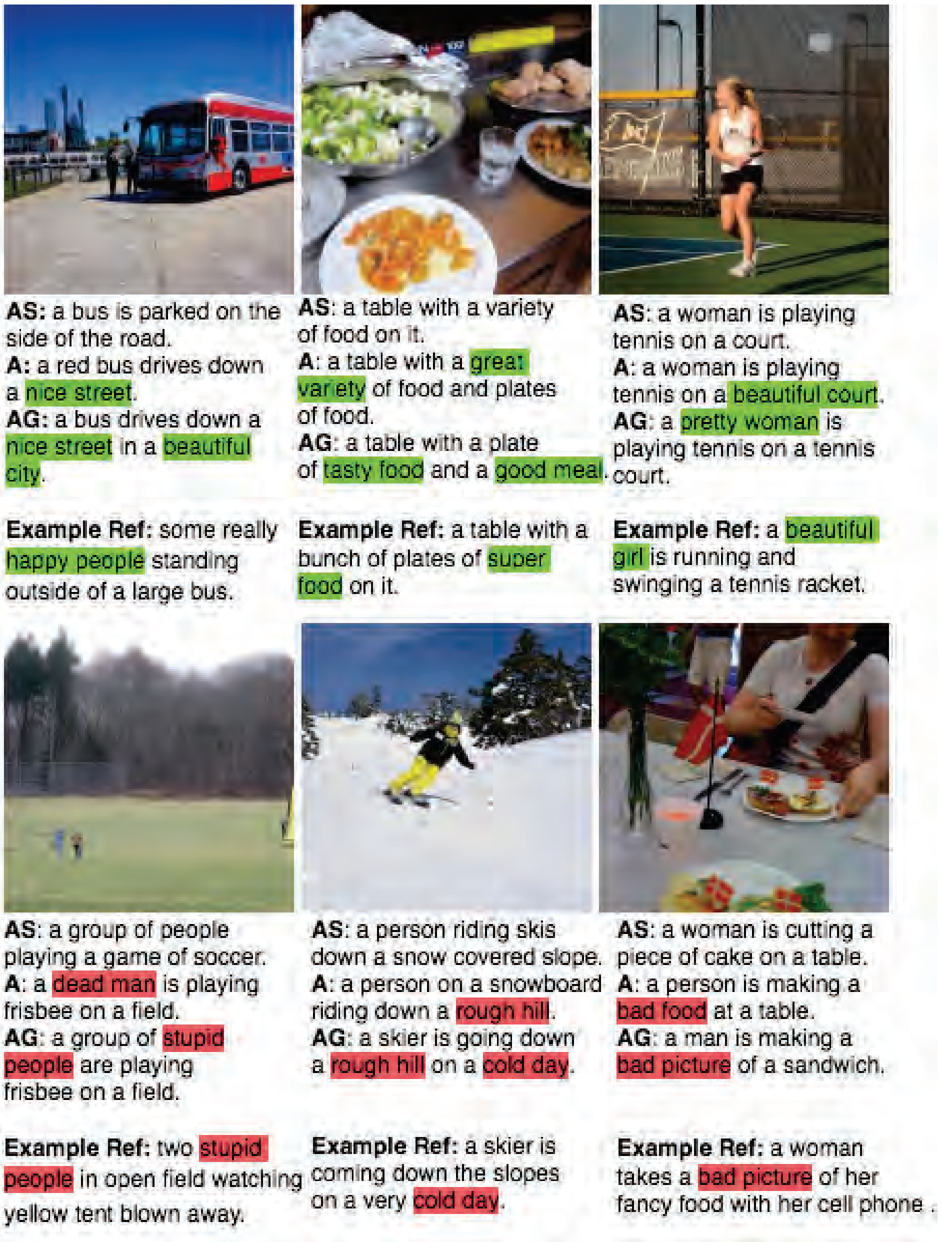

AS: a bus is parked on the side of the road.
A: a red bus drives down a nice street.
AG: a bus drives down a nice street in a beautiful city.
Example Ref: some really happy people standing outside of a large bus.
AS: a table with a variety of food on it.
A: a table with a great variety of food and plates of food.
AG: a table with a plate of tasty food and a good meal.
Example Ref: a table with a bunch of plates of super food on it.
AS: a woman is playing tennis on a court.
A: a woman is playing tennis on a beautiful court.
AG: a pretty woman is playing tennis on a tennis court.
Example Ref: a beautiful girl is running and swinging a tennis racket.
AS: a group of people playing a game of soccer.
A: a dead man is playing frisbee on a field.
AG: a group of stupid people are playing frisbee on a field.
Example Ref: two stupid people in open field watching yellow tent blown away.
AS: a person riding skis down a snow covered slope.
A: a person on a snowboard riding down a rough hill.
AG: a skier is going down a rough hill on a cold day.
Example Ref: a skier is coming down the slopes on a very cold day.
AS: a woman is cutting a piece of cake on a table.
A: a person is making a bad food at a table.
AG: a man is making a bad picture of a sandwich.
Example Ref: a woman takes a bad picture of her fancy food with her cell phone .

Figure 5.3: Examples on the positive (first 3) and negative (last 3) datasets (AS for Attend-GAN$_{-SA}$, A for Attend-GAN$_{-A}$, AG for Attend-GAN and Example Ref for an example from the reference captions). Green and red colors indicate the generated positive and negative adjective-noun pairs in SentiCap, respectively.

"bad picture" and Attend-GAN$_{-A}$ generates "bad food". One of the ground-truth captions exactly includes "bad picture" (in Appendix A, Figures A.1 and A.2 give more examples showing the effectiveness of the Attend-GAN model compared to other models to generate sentiment-bearing image captions).

5.5 Summary

In this chapter, we proposed Attend-GAN, an attention-based image captioning model using an adversarial training mechanism. Our model is capable of generating stylistic captions which are strongly correlated with images and contain diverse stylistic components. Attend-GAN achieves the state-of-the-art performance on the SentiCap dataset. It outperforms our baseline models and generates stylistic captions with a high level of variety. It does this by adding stylistic information in the second stage of training using a small stylistic dataset similar to the previous work [2, 30, 31]; however, it uses the adversarial training mechanism to effectively regulate between factual and stylistic information by comparing the generated captions with human-generated ones. The mechanism guides Attend-GAN to apply diverse stylistic patterns at proper time, in response to **RQ 3** discussed in Chapter 1, while maintaining the relationship between generated captions and visual content.

6

Senti-Attend: Image Captioning using Sentiment and Attention

As noted in the previous chapters, there has been much recent work on image captioning models that describe the factual aspects of an image. Recently, some models have incorporated non-factual aspects into the captions, such as sentiment or style. However, such models typically have difficulty in balancing the semantic aspects of the image and the non-factual dimensions of the caption, in part because of the usual two-stage architecture of such systems; in addition, it can be observed that humans may focus on different aspects of an image depending on the chosen sentiment or style of the caption. While there has been some recent purely text-based work on generating text with aspect-based sentiment, this visually grounded sentiment presents different challenges. To address this, we design an attention-based model, named Senti-Attend, to better add sentiment to image captions. The model embeds and learns sentiment with respect to image-caption data, and uses both high-level and word-level sentiment information during the learning process. Senti-Attend outperforms the state-of-the-art work in image captioning with sentiment using standard evaluation metrics. An analysis of generated captions also shows that our model does this by a better selection of the sentiment-bearing adjectives and adjective-noun pairs.[1]

6.1 Introduction

As discussed in §5.1, researchers in image captioning have proposed models that allow the generation of captions with a particular style [30, 49] or sentiment [2, 31] for different applications. In sentiment-bearing image captioning, the focus of the caption could be different depending on the desired sentiment. Figure 6.1 contains an example image from the dataset of human-authored sentiment-infused captions from Mathews *et al.* [2], where all three negative captions focus on the mugs, and the positive captions focus on the light or the kitchen generally. Previous work in this domain usually does not use attention mechanisms [7, 8, 20]; however, the state-of-the-art image captioning models do, and the above observation suggests it would be useful here. And while some purely textual generation systems do in some respects focus on elements on the source — e.g. generation of reviews with aspect-based sentiment [173], or using an attention mechanism [174] — these are quite different from the visually grounded aspects requiring spatial attention. The Senti-Attend model that we propose in this chapter, therefore, incorporates spatial attention into the generation of sentiment-bearing captions.

Moreover, we note that previous work similar to our model proposed in Chapter 5, usually apply a two-stage training mechanism: training on a large factual dataset and then training on a small stylistic dataset. As we discussed in the previous chapter, this usually leads to a small and limited set of stylistic patterns because the image captioning model usually learns

[1] The content of this chapter is based on the following paper:

Omid Mohamad Nezami, Mark Dras, Stephen Wan, Cecile Paris (2018). Senti-Attend: Image Captioning using Sentiment and Attention. *arXiv preprint arXiv:1811.09789*.

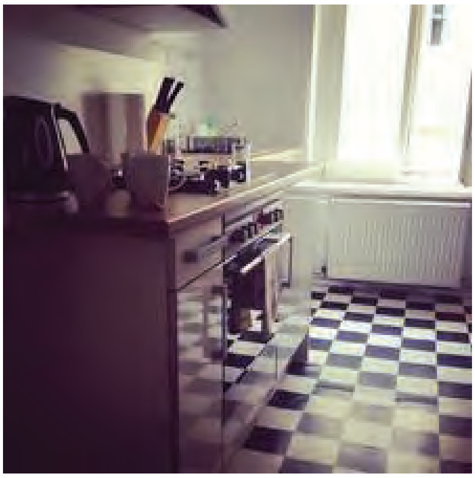


Figure 6.1: An image with different foci for positive ("a beautiful well-appointed kitchen") or negative ("ugly mugs") sentiments.

from a small scale stylistic dataset in the second stage of training. The model also has a difficulty to regulate between factual and stylistic information since it mostly concentrates on generating stylistic image captions in the second stage. Thus, Senti-Attend uses an end-to-end training mechanism which needs one stage training for generating image captions with stylistic information. It learns from the combination of factual and stylistic datasets by assigning different labels to factual and stylistic captions during training. This helps Senti-Attend to differentiate between applying semantic and stylistic information in one phase training from a large combined dataset. To do so, Senti-Attend needs to encode stylistic information in a way that is distinguishable from factual information and does not 'damage' this information. This is mostly inspired by the literature of controlled natural language generation discussed in §2.4.2. However, the previous work usually encoded and injected sentiment using one-hot vectors at different time steps [35, 48], which can have the effect of forcing sentiment into a generated description of an image that is not semantically suited to the image. Thus, the focus of the encoding part of Senti-Attend model is to embed the sentiment information in real-valued vectors, allowing the model to learn where sentiment can be applied without changing semantic relationship between the image and the generated caption. The model incorporates two kinds of sentiment embedding, which turn out to be complementary. The high-level embedding captures overall sentiment; it is fed into the long short-term memory (LSTM) network that handles the caption generation. The word-level embedding, in contrast, captures a notion of sentiment linked to the words in the vocabulary.

6.2 Approach

Our image captioning model has an attention-based encoder-decoder mechanism to generate sentiment-bearing captions; we call our model Senti-Attend. Our model takes as the first input an image encoded into K image feature sets, $\boldsymbol{a} = \{\boldsymbol{a}_1, ..., \boldsymbol{a}_K\}$. Each set has D dimensions to represent a region of the image, termed spatial features, $\boldsymbol{a}_i \in \mathbb{R}^D$. a is usually generated using a convolutional layer of a convolutional neural network (CNN). As the second input, we have the targeted sentiment category ($\boldsymbol{s}$) to generate the image description with specific sentiment. The model takes these inputs and generates a caption $\boldsymbol{x}$ encoded as a sequence of 1-of-N encoded words.

$$\boldsymbol{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}, \boldsymbol{x}_i \in \mathbb{R}^N \tag{6.1}$$

where N is the size of the vocabulary and T is the length of the caption.

6.2.1 Spatial Features

ResNet-152 [70] is used as the CNN model. It has been pre-trained on the ImageNet dataset [10]. For use in our image captioning model, we use 7 × 7 × 2048 features from the Res5c layer of the CNN model. Then, we reshape the features into 49 × 2048 dimensions.

6.2.2 Targeted Sentiment Category

Our model aims to achieve an image description that is relevant to a targeted sentiment category similar to our previous model described in Chapter 5. Similarly, sentiment categories are { Positive, Negative }, as per the SentiCap work [2]. However, we have a further sentiment category { Neutral }, which is for generating captions without dominant sentiment values. In our problem statement, we assume that the targeted sentiment category is already specified, as previous work does. Because our model uses the sentiment category to describe an image, we can change the sentiment category to generate a different caption with a new sentiment value.

We embed the sentiment categories to give real value vectors, which are randomly initialized. Then, our model learns a sentiment category's vector during training time. Using this mechanism, we allow our system to learn the sentiment information in an adaptive way with respect to visual and text data. Specifically, we use one sentiment embedding vector ($\boldsymbol{E}_1$) as an additional input to a long short-term memory (LSTM) network. In addition, we use another embedding vector ($\boldsymbol{E}_2$) as a supplementary energy term to predict the next word's probability [2]. $\boldsymbol{E}_x$, the embedding vector of the sentiment category, has F dimensions, $\boldsymbol{E}_x \in \mathbb{R}^F$. $\boldsymbol{E}_1$ is to model the high-level representation of the sentiment concept in the generated caption. $\boldsymbol{E}_2$ represents desired word-level sentiment.

6.2.3 Captioning Model

Senti-Attend uses a sentiment term specifying the targeted sentiment and an attention mechanism attending to the spatial visual features. It aims to minimize the following cross entropy loss (the first loss):

$$L_1 = -\sum_{1 \le t \le T} \log(\boldsymbol{p}_1(\boldsymbol{x}_t | \boldsymbol{h}_t, \hat{\boldsymbol{a}}_t, \boldsymbol{E}_2)) + \sum_{1 \le k \le K} (1 - \sum_{1 \le t \le T} a_{tk})^2 \tag{6.2}$$

$$\boldsymbol{p}_1 = \text{softmax}(\boldsymbol{h}_t \boldsymbol{W}_h + \hat{\boldsymbol{a}}_t \boldsymbol{W}_a + \boldsymbol{E}_2 \boldsymbol{W}_e + \boldsymbol{b}) \tag{6.3}$$

where $\boldsymbol{p}_1$, as the categorical probability distribution across all words in the vocabulary, is from the output of a multilayer perceptron; $\boldsymbol{x}_t$ is the next targeted word, from the ground truth caption; $\boldsymbol{E}_2$ is the second embedded vector of sentiment; $\boldsymbol{h}_t$ is the hidden state of the LSTM, calculated using Equation 6.6; $\hat{\boldsymbol{a}}_t$ is the attention-based content (Equation 6.7); and $\boldsymbol{W}_x$ and $\boldsymbol{b}$ are our trained weights and bias, respectively. The loss function contains a regularization term (the last part), to encourage the system to take equal notice to different parts of the image by the end of a caption generation process.

[2] We found E_1 effective as the input of the decoder and E_2 as an external factor impacting on the probability distribution of the next generated word. We have also tried applying E_1 to initiate the decoder, but, we didn't achieve good results.

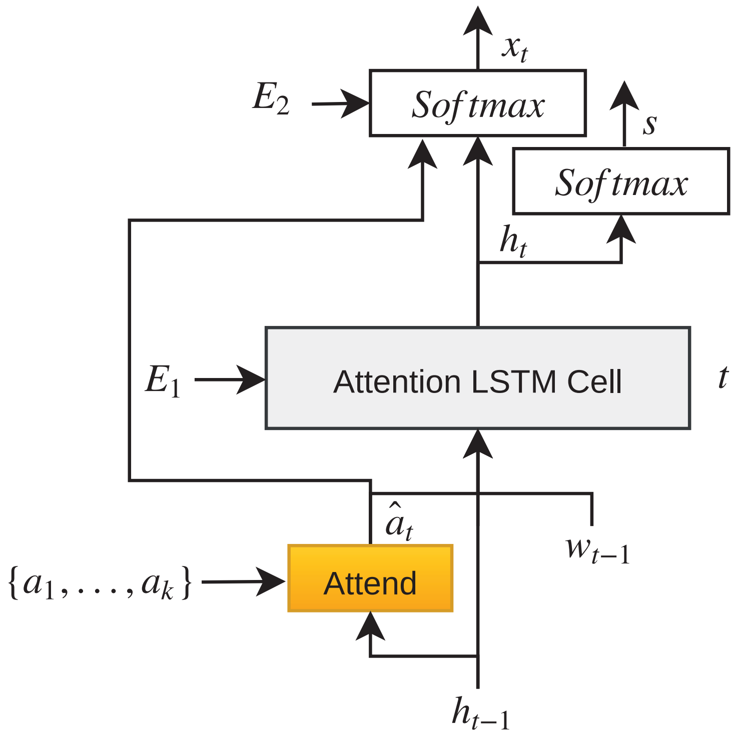


Figure 6.2: The framework of the Senti-Attend model. Spatial image features $\{\boldsymbol{a}_1,...,\boldsymbol{a}_k\}$ are attended using our attention-based component. They are the outputs of our ResNet-152 network.

Senti-Attend also includes the following sentiment-specific cross entropy loss that we call the second loss:

$$L_2 = -\frac{1}{L} \sum_{1 \le t \le L} \log(\boldsymbol{p}_2(\boldsymbol{s}|\boldsymbol{h}_t)) \tag{6.4}$$

$$\boldsymbol{p}_2(\boldsymbol{s}|\boldsymbol{h}_t) = \text{softmax}(\boldsymbol{h}_t \boldsymbol{W}_s + \boldsymbol{b}_s) \tag{6.5}$$

where $\boldsymbol{p}_2(\boldsymbol{s}|\boldsymbol{h}_t)$ is the categorical probability distribution of the current state ($\boldsymbol{h}_t$) across three sentiment classes { Positive, Neutral, Negative }, obtained from a multilayer perceptron; $\boldsymbol{s}$ is the targeted sentiment class (as the ground-truth sentiment label compared to $\boldsymbol{E}_x$ as a learned sentiment embedding vector); and $\boldsymbol{W}_s$ and $\boldsymbol{b}_s$ are our trained weights and bias. Using this loss function, Senti-Attend model can learn the targeted sentiment class at the end of each time step. The model architecture is shown in Figure 6.2.

Our LSTM decoder is defined by Equations 6.6:

$$\begin{aligned}
\boldsymbol{i}_t &= \sigma_g(\boldsymbol{W}_i \boldsymbol{w}_{t-1} + \boldsymbol{H}_i \boldsymbol{h}_{t-1} + \boldsymbol{A}_i \hat{\boldsymbol{a}}_t + \boldsymbol{B}_i \boldsymbol{E}_1 + \boldsymbol{b}_i) \\
\boldsymbol{g}_t &= \sigma_h(\boldsymbol{W}_g \boldsymbol{w}_{t-1} + \boldsymbol{H}_g \boldsymbol{h}_{t-1} + \boldsymbol{A}_g \hat{\boldsymbol{a}}_t + \boldsymbol{B}_g \boldsymbol{E}_1 + \boldsymbol{b}_g) \\
\boldsymbol{o}_t &= \sigma_g(\boldsymbol{W}_o \boldsymbol{w}_{t-1} + \boldsymbol{H}_o \boldsymbol{h}_{t-1} + \boldsymbol{A}_o \hat{\boldsymbol{a}}_t + \boldsymbol{B}_o \boldsymbol{E}_1 + \boldsymbol{b}_o) \\
\boldsymbol{f}_t &= \sigma_g(\boldsymbol{W}_f \boldsymbol{w}_{t-1} + \boldsymbol{H}_f \boldsymbol{h}_{t-1} + \boldsymbol{A}_f \hat{\boldsymbol{a}}_t + \boldsymbol{B}_f \boldsymbol{E}_1 + \boldsymbol{b}_f) \\
\boldsymbol{c}_t &= \boldsymbol{f}_t \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \boldsymbol{g}_t \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \sigma_h(\boldsymbol{c}_t)
\end{aligned} \tag{6.6}$$

where $\boldsymbol{i}_t$, $\boldsymbol{g}_t$, $\boldsymbol{o}_t$, $\boldsymbol{f}_t$ and $\boldsymbol{c}_t$ are input gate, input modulation gate, output gate, forget gate and memory cell, respectively. Here, $\boldsymbol{w}_{t-1}$ is a real value vector with M dimensions to represent the previous word, $\boldsymbol{w}_x \in \mathbb{R}^M$, embedded using our model. $\boldsymbol{E}_1$ is the first embedded vector of sentiment. It is used to condition the caption generation process using the targeted sentiment, which we refer as high-level sentiment. $\boldsymbol{W}_x, \boldsymbol{H}_x, \boldsymbol{A}_x, \boldsymbol{B}_x$, and $\boldsymbol{b}_x$ are trained weights and biases. σ_g and σ_h are the logistic sigmoid function and the hyperbolic tangent function, respectively.

$\hat{\boldsymbol{a}}_t$ is estimated using:

$$\hat{\boldsymbol{a}}_t = \sum_{1 \leq j \leq K} e'_{j,t} \boldsymbol{a}_j \tag{6.7}$$

where our generated attention weights are specified with $e'_{j,t}$ that we normalize using a softmax function applied on the generated scores ($\boldsymbol{e}_t$) of our attention-based component:

$$\begin{aligned}
\boldsymbol{e}'_t &= \text{softmax}(\boldsymbol{e}_t) \\
e_{j,t} &= \boldsymbol{W}_e^T \tanh(\boldsymbol{W}'_a a_j + \boldsymbol{W}'_h \boldsymbol{h}_t)
\end{aligned} \tag{6.8}$$

We specify the trainable parameters of the component with $\boldsymbol{W}_e^T$ and $\boldsymbol{W}'_x$.

6.3 Experimental Setup

6.3.1 Datasets

Microsoft COCO dataset This dataset [25], which is the largest image captioning dataset, is used to train the Senti-Attend model. We use the specified training portion of the dataset which includes 413K+ captions for 82K+ images. This dataset can help our model to generate generic image captions.

SentiCap dataset This dataset [2] is used to train our model to generate captions with sentiment. It includes manually generated captions with two parts: *positive* and *negative* sentiments. The training portion of the positive part consists of 2873 captions for 998 images and the testing portion of the positive part consists of 2019 captions for 673 images. The training portion of the negative part contains 2468 captions for 997 images and the testing portion of the negative part contains 1509 captions for 503 images.

Similar to Chapter 5, we use both of these datasets to train our model. However, for the training phase, we combine the training sets of the Microsoft COCO and the SentiCap dataset so that all captions in the Microsoft COCO dataset are assigned the *neutral* label in terms of sentiment values. We separately report our results on the negative and positive test parts similar to other work in this domain [2, 31, 48].

6.3.2 Evaluation Metrics

Overall Metrics As in §5.3.2, we evaluate Senti-Attend model using SPICE [167], CIDEr [100], METEOR [166], ROUGE-L [165] and BLEU [164], which are standard evaluation metrics. Larger values are better results for all metrics.

Linguistic Metrics As in §5.3.2, we calculate entropy for measuring the diversity of the generated adjectives, which are in the set of the adjective-noun pairs (ANPs) provided in the SentiCap dataset. Moreover, to consider how well the model's captions attend to objects chosen by humans to add a particular sentiment to, we also calculate how often a model's chosen nouns correspond to the constituent nouns of the ANPs in the reference captions. To do this we consider the SPICE metric [167]. In its full form, SPICE maps reference and generated captions to scene graphs that are derived from dependency trees, and then calculates their overlap; this proves to give a higher correlation with human judgements than the other standard metrics. For our evaluation goal here, we are only interested in overlap in objects chosen for reference and generated captions, so we implement a simplified version of SPICE that only considers noun matches, SPICE$_N$; this includes the SPICE functionality of matching WordNet synonyms [167] (for example, *street* and *road*). Relatedly, we calculate precision using the generated ANPs by each model. We divide the number of the generated ANPs found in the (sentiment-appropriate) reference captions by the overall generated ANPs. Larger values show better results for these metrics.

Human Sentiment Evaluation In addition to the above, we gather human judgements about the sentiment of the generated captions, as it is possible that neither the overall metrics nor the linguistic metrics capture the sentiment of the caption as a whole. As this kind of human evaluations is not commonly done in previous work, there is no standard approach to draw on. We use an approach similar to that of Mathews *et al.* [2] in validating the quality of their SentiCap dataset. In that work, each caption was given to three Amazon Mechanical Turkers, and asked whether the sentiment was positive, negative or neutral. Here we randomly select 50 positive and 50 negative captions generated by Senti-Attend, with additional captions randomly sampled from the MSCOCO and SentiCap ground-truth datasets such that each Turker had to annotate 20 each of positive, neutral and negative captions. These ground-truth captions allow benchmarking both for the Senti-Attend captions and with respect to the validation data of Mathews *et al.* [2].

6.3.3 Models for Comparison

To evaluate our model's performance in generating sentiment-bearing captions, we compare it with high-performing approaches in this domain, as described in Chapter 2. Mathews *et al.*[2] introduced the task and the SentiCap dataset, and the SentiCap model for generating captions. Chen *et al.*[31] proposed SF-LSTM+Adapt to weight the factual and sentimental dimensions of the input captions. You *et al.*[48] proposed two models for incorporating sentiment into image captioning tasks: Direct Inject, adding a new dimension to the input; and Sentiment Flow, using a new architecture that injects sentiment in different caption generation steps. Attend-GAN [3], as the state-of-the-art, incorporates sentiment in a different fashion, using an adversarial training mechanism (this is our previous approach explained in Chapter 5).

For model-internal comparisons, we use our attention-based image captioning model without sentiment inputs (the Attend model); this allows us to assess the effect of spatial attention alone, without sentiment (only images as inputs). We further define three variants of our general Senti-Attend system (Figure 6.2). First, we use one-hot embedding representations for the ternary sentiment instead of the two embedding vectors ($\boldsymbol{E}_1$ and $\boldsymbol{E}_2$). We also do not use the second loss function. We call this model Senti-Attend$_{-E_1E_2L_2}$ (showing the impact of adding sentiment in addition to images as inputs). We speculate that the distributed representation of sentiment, analogous to those used in Ghosh *et al.*[52] and Zhou *et al.*[51], will allow a model to be more selective about where to apply sentiment, where this one-hot variant in contrast might force sentiment inappropriately. Second, we only use the first embedding vector ($\boldsymbol{E}_1$) instead of both embedding vectors, and again do not use the second loss function: we call this the Senti-Attend$_{-E_2L_2}$ model. Third, we only leave out the second loss function, giving the Senti-Attend$_{-L_2}$ model. We name our full approach, which uses both embedding vectors and the second loss function, the Senti-Attend model. All Senti-Attend models have a similar architecture applying high-level and word-level sentiment information.

6.3.4 Implementation Details

In this work, we set the size of the memory cell and the hidden state of the LSTM to 2048 dimensions for all models except the Attend model. The model has the memory cell and the hidden state with size of 1024 dimensions (the Attend model can generate a better performance using this size). The word embedding and the sentiment embedding vectors have 512 and 256 dimensions, respectively. The Adam optimization function with a learning rate of 0.001 is applied to optimize our network [170]. We set the size of mini-batches to 180 for all models except the Attend model, which has mini-batches of size 100 (the Attend model can generate a better performance using this batch size). The size of our vocabulary is fixed to 9703 for all models. Since METEOR is more closely correlated with human judgments and is calculated more quickly than SPICE (METEOR does not need dependency parsing) [167], it is used to select the best model on the validation set.

6.4 Results

6.4.1 Overall Metrics

In Table 6.1, we report the results of Senti-Attend compared to the state-of-the-art (the results of the state-of-the-art are obtained from the cited references) on the SentiCap dataset. We use the same training/test split, so numbers are comparable. The Senti-Attend model achieved better results compared to the previous state-of-the-art by all standard evaluation metrics except BLEU-4 for the negative test set, where the model is very marginally (0.1) lower than the Sentiment Flow model; all the average results outperform the recent state-of-the-art

Table 6.1: Our image captioning results (%) compared to the state-of-the-art models on the SentiCap test split. Pos, Neg, and Avg show the results on the positive test set, the negative test set and their average. The best performances are bold. (B-N: BLEU-N, R: ROUGE-L, M: METEOR, C: CIDEr, S: SPICE).

Senti	Model	B-1	B-2	B-3	B-4	R	M	C	S
Pos	SentiCap	49.1	29.1	17.5	10.8	36.5	16.8	54.4	_
	SF-LSTM+Adapt	50.5	30.8	19.1	12.1	38.0	16.6	60.0	_
	Direct Inject	51.2	30.6	18.8	11.6	38.4	17.2	61.1	_
	Sentiment Flow	51.1	31.4	19.4	12.3	38.6	16.9	60.8	_
	Attend-GAN	56.9	33.6	20.3	12.5	44.3	18.8	61.6	15.9
	Ours: Attend	57.1	33.8	20.6	13.1	45.3	17.8	69.2	17.2
	Ours: Senti-Attend	57.6	34.2	20.5	12.7	45.1	18.9	68.6	16.7
Neg	SentiCap	50.0	31.2	20.3	13.1	37.9	16.8	61.8	_
	SF-LSTM+Adapt	50.3	31.0	20.1	13.3	38.0	16.2	59.7	_
	Direct Inject	52.2	33.6	22.2	14.6	39.8	17.1	68.4	_
	Sentiment Flow	51.0	33.0	21.9	14.8	39.4	17.0	70.1	_
	Attend-GAN	56.2	34.1	21.3	13.6	44.6	17.9	64.1	16.2
	Ours: Attend	56.5	33.5	20.2	12.5	45.0	17.7	67.7	16.3
	Ours: Senti-Attend	58.6	35.4	22.3	14.7	45.7	19.0	71.9	17.4
Avg	SentiCap	49.55	30.15	18.90	11.95	37.20	16.80	58.10	_
	SF-LSTM+Adapt	50.40	30.90	19.60	12.70	38.00	16.40	59.85	_
	Direct Inject	51.70	32.10	20.50	13.10	39.10	17.15	64.75	_
	Sentiment Flow	51.05	32.20	20.65	13.55	39.00	16.95	65.49	_
	Attend-GAN	56.55	33.85	20.80	13.05	44.45	18.35	62.85	16.05
	Ours: Attend	56.80	33.65	20.40	12.80	45.15	17.75	68.45	16.75
	Ours: Senti-Attend	**58.10**	**34.80**	**21.40**	**13.70**	**45.40**	**18.95**	**70.25**	**17.05**

Table 6.2: The image captioning results (%) of our proposed models on the SentiCap test split. Pos, Neg, and Avg show the results on the positive test set, the negative test set and their average. The best performances are bold. (B-N: BLEU-N, R: ROUGE-L, M: METEOR, C: CIDEr, S: SPICE).

Senti	Model	B-1	B-2	B-3	B-4	R	M	C	S
Pos	Attend	57.1	33.8	20.6	13.1	45.3	17.8	69.2	17.2
	Senti-Attend$_{-E_1E_2L_2}$	56.0	32.3	18.7	11.1	43.6	18.7	60.8	15.9
	Senti-Attend$_{-E_2L_2}$	57.1	34.1	21.0	13.2	45.5	18.1	69.9	16.8
	Senti-Attend$_{-L_2}$	56.4	33.2	19.7	12.3	44.2	18.0	65.0	16.1
	Senti-Attend	57.6	34.2	20.5	12.7	45.1	18.9	68.6	16.7
Neg	Attend	56.5	33.5	20.2	12.5	45.0	17.7	67.7	16.3
	Senti-Attend$_{-E_1E_2L_2}$	55.8	32.5	19.5	11.9	43.7	18.1	62.3	16.5
	Senti-Attend$_{-E_2L_2}$	56.6	34.0	21.1	13.6	45.2	18.1	69.9	16.4
	Senti-Attend$_{-L_2}$	56.6	34.2	21.5	14.1	45.4	18.0	71.3	16.8
	Senti-Attend	58.6	35.4	22.3	14.7	45.7	19.0	71.9	17.4
Avg	Attend	56.80	33.65	20.40	12.80	45.15	17.75	68.45	16.75
	Senti-Attend$_{-E_1E_2L_2}$	55.90	32.40	19.10	11.50	43.65	18.40	61.55	16.20
	Senti-Attend$_{-E_2L_2}$	56.85	34.05	21.05	13.40	45.35	18.10	69.90	16.60
	Senti-Attend$_{-L_2}$	56.50	33.70	20.60	13.20	44.80	18.00	68.15	16.45
	Senti-Attend	**58.10**	**34.80**	**21.40**	**13.70**	**45.40**	**18.95**	**70.25**	**17.05**

Attend-GAN on the SentiCap dataset on all metrics, which itself showed large improvements over previous approaches (although not on all metrics in that case).

Table 6.2 shows the comparison between our model variants, to assess the contributions of elements of the main model. The Senti-Attend model has achieved the best performances for all evaluation metrics across positive, negative and average test sets. The Senti-Attend$_{-L_2}$ and Senti-Attend$_{-E_2L_2}$ models have comparable performance (although we observe in §6.4.2 below that Senti-Attend$_{-L_2}$ produces better sentiment-bearing captions by our linguistic criteria). These two models, which are using the sentiment embedding vectors, outperform Senti-Attend$_{-E_1E_2L_2}$ for all evaluation metrics except METEOR, showing the effectiveness

Table 6.3: The Entropy of the generated adjectives and the SPICE of the generated nouns (SPICE$_N$) using our models on the SentiCap test split. We also calculated the precision of the generated ANPs using the models.

Senti	Model	Entropy	SPICE$_N$	Precision
Pos	Attend	2.2042	15.8%	40.0%
	Senti-Attend$_{-E_1E_2L_2}$	3.2840	13.7%	39.1%
	Senti-Attend$_{-E_2L_2}$	3.2795	15.3%	32.0%
	Senti-Attend$_{-L_2}$	3.2691	14.4%	35.1%
	Senti-Attend	3.2040	15.0%	39.3%
Neg	Attend	2.1513	15.5%	40.0%
	Senti-Attend$_{-E_1E_2L_2}$	3.5681	18.1%	46.2%
	Senti-Attend$_{-E_2L_2}$	3.5895	16.7%	37.9%
	Senti-Attend$_{-L_2}$	3.5396	17.7%	46.1%
	Senti-Attend	3.7954	17.7%	51.4%
Avg	Attend	2.17775	15.65%	40.00%
	Senti-Attend$_{-E_1E_2L_2}$	3.42605	15.90%	42.65%
	Senti-Attend$_{-E_2L_2}$	3.43450	16.00%	34.95%
	Senti-Attend$_{-L_2}$	3.40435	16.05%	40.60 %
	Senti-Attend	**3.49970**	**16.35%**	**45.35%**

of our embedding approach. Here, we report the results of the Attend model to show the performance of our attention-based system without sentiment; however, the model is not effective in terms of sentiment-bearing captions (Table 6.3).

6.4.2 Linguistic Metrics

Table 6.3 shows the Attend model, which does not have sentiment inputs, achieves the lowest Entropy and SPICE$_N$. The Senti-Attend model gives the highest Entropy and SPICE$_N$, and so has both the highest variability in adjective choice for describing sentiment, and also attends best to the objects (represented by nouns) that human annotators choose to apply sentiment to

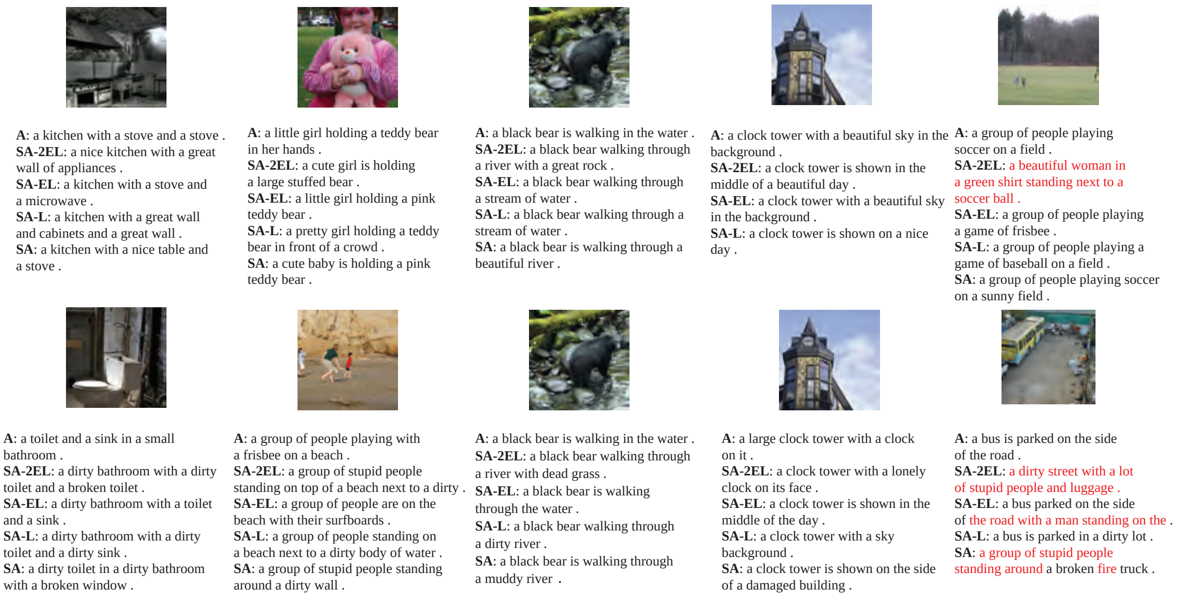
Table 6.4: Top-10 generated adjectives using our models.

Senti	Model	Top 10 Adjective
Pos	Attend	white, black, small, blue, little, tall, different, _, _, _
	Senti-Attend$_{-E_1E_2L_2}$	nice, beautiful, great, happy, good, busy, white, sunny, blue, black
	Senti-Attend$_{-E_2L_2}$	white, black, blue, small, nice, little, beautiful, tall, sunny, happy
	Senti-Attend$_{-L_2}$	great, nice, beautiful, white, black, blue, busy, right, healthy, happy
	Senti-Attend	beautiful, nice, sunny, great, busy, white, blue, good, happy, calm
Neg	Attend	white, black, small, blue, little, tall, different, busy, _, _
	Senti-Attend$_{-E_1E_2L_2}$	dirty, stupid, lonely, bad, broken, cold, dead, little, crazy, white
	Senti-Attend$_{-E_2L_2}$	white, black, blue, small, little, dead, tall, busy, lonely, dirty
	Senti-Attend$_{-L_2}$	dirty, white, black, dead, lonely, stupid, little, blue, broken, cold
	Senti-Attend	lonely, stupid, dead, dirty, broken, white, cold, black, bad, weird

in captions; in addition, it generates the best precision value for the generated ANPs.

SPICE$_N$ further shows that Senti-Attend$_{-E_1E_2L_2}$ is less effective than other Senti-Attend models at choosing appropriate nouns to describe with sentiment. This accords with our intuition that the one-hot sentiment encoding does indeed force sentiment into captions in ways that are not optimal. Of the remaining models, Senti-Attend$_{-L_2}$ has the best SPICE$_N$ after Senti-Attend. Senti-Attend$_{-E_2L_2}$ has the best Entropy after Senti-Attend and a good value of SPICE$_N$; however, it does not have a good precision, indicating that having separate word-level sentiment is useful. As noted in the previous section, it achieved comparable performance to Senti-Attend$_{-E_1E_2L_2}$ for all overall evaluation metrics. This shows that producing captions that combine good performances for evaluation metrics (i.e. producing good captions in general), a high variety of sentiment terms, and good alignment with objects chosen to apply sentiment to, is a challenging issue in this domain. Nevertheless, the full Senti-Attend model is able to do well on all these competing objectives.

By way of examples, Table 6.4 shows the top-10 generated adjectives using our models. We observe that colours are repeated for both positive and negative generated captions (as they are in both positive and negative ground-truth captions): these commonly appear in non-sentiment-based captioning system outputs. Among our sentiment models, Senti-Attend$_{-E_2L_2}$



A: a kitchen with a stove and a stove .
SA-2EL: a nice kitchen with a great wall of appliances .
SA-EL: a kitchen with a stove and a microwave .
SA-L: a kitchen with a great wall and cabinets and a great wall .
SA: a kitchen with a nice table and a stove .

A: a little girl holding a teddy bear in her hands .
SA-2EL: a cute girl is holding a large stuffed bear .
SA-EL: a little girl holding a pink teddy bear .
SA-L: a pretty girl holding a teddy bear in front of a crowd .
SA: a cute baby is holding a pink teddy bear .

A: a black bear is walking in the water .
SA-2EL: a black bear walking through a river with a great rock .
SA-EL: a black bear walking through a stream of water .
SA-L: a black bear walking through a stream of water .
SA: a black bear is walking through a beautiful river .

A: a clock tower with a beautiful sky in the background .
SA-2EL: a clock tower is shown in the middle of a beautiful day .
SA-EL: a clock tower with a beautiful sky in the background .
SA-L: a clock tower is shown on a nice day .

A: a group of people playing soccer on a field .
SA-2EL: a beautiful woman in a green shirt standing next to a soccer ball .
SA-EL: a group of people playing a game of frisbee .
SA-L: a group of people playing a game of baseball on a field .
SA: a group of people playing soccer on a sunny field .

A: a toilet and a sink in a small bathroom .
SA-2EL: a dirty bathroom with a dirty toilet and a broken toilet .
SA-EL: a dirty bathroom with a toilet and a sink .
SA-L: a dirty bathroom with a dirty toilet and a dirty sink .
SA: a dirty toilet in a dirty bathroom with a broken window .

A: a group of people playing with a frisbee on a beach .
SA-2EL: a group of stupid people standing on top of a beach next to a dirty .
SA-EL: a group of people are on the beach with their surfboards .
SA-L: a group of people standing on a beach next to a dirty body of water .
SA: a group of stupid people standing around a dirty wall .

A: a black bear is walking in the water .
SA-2EL: a black bear walking through a river with dead grass .
SA-EL: a black bear is walking through the water .
SA-L: a black bear walking through a dirty river .
SA: a black bear is walking through a muddy river .

A: a large clock tower with a clock on it .
SA-2EL: a clock tower with a lonely clock on its face .
SA-EL: a clock tower is shown in the middle of the day .
SA-L: a clock tower with a sky background .
SA: a clock tower is shown on the side of a damaged building .

A: a bus is parked on the side of the road .
SA-2EL: a dirty street with a lot of stupid people and luggage .
SA-EL: a bus parked on the side of the road with a man standing on the .
SA-L: a bus is parked in a dirty lot .
SA: a group of stupid people standing around a broken fire truck .

Figure 6.3: Examples of generated captions using our models. The top row contains positive generated captions and the bottom row contains negative ones (A for Attend, SA-2EL for Senti-Attend$_{-E_1E_2L_2}$, SA-EL for Senti-Attend$_{-E_2L_2}$, SA-L for Senti-Attend$_{-L_2}$, and SA for Senti-Attend). The last column includes some captions with some inconsistent parts.

ranks them high, whereas the others have sentiment-appropriate adjectives ranked higher, indicating that just injecting sentiment into the LSTM (Figure 6.2) is insufficient. Attend has similar generated adjectives for both sentiments because it does not use any sentiment signal.

Figure 6.3 shows a number of the example generated captions using our models. The first two columns include positive and negative captions for different images (positive and negative captions for the same images do not necessarily appear in the test set). For example, in the first column, Senti-Attend generates the caption “a dirty toilet in a dirty bathroom with a broken window”. The caption is compatible with the negative sentiment of the corresponding image. Other Senti-Attend models are also successful in generating negative captions for the image, but with less variability of expression. For instance, Senti-Attend$_{-E_1E_2L_2}$ generates “a dirty bathroom with a dirty toilet and a broken toilet”. Senti-Attend can generate “a kitchen with a nice table and a stove” caption, which positively describes its corresponding image. Senti-Attend$_{-L_2}$ and Senti-Attend$_{-E_1E_2L_2}$ can also generate positive captions. Here, Attend cannot generate sentiment-bearing captions. In the second column, we have similar generated captions. Columns three and four include positive and negative captions for similar images. The captions show that Senti-Attend can manipulate and control the sentiment value of the captions using the targeted sentiment. For example, in the third column, Senti-Attend generates "a black bear is walking through a beautiful river" and "a black bear is walking through a muddy river" for the positive and negative sentiments, respectively. Senti-Attend$_{-E_1E_2L_2}$ can generate both types of sentiments, as well. Senti-Attend$_{-L_2}$ can generate a negative caption properly. In the forth column, all Senti-Attend models are successful in generating positive captions. The Attend model also generates a positive caption properly. Here, generating negative captions for the image is even challenging for humans. However, Senti-Attend and Senti-Attend$_{-E_1E_2L_2}$ models can effectively generate positive ("nice day" and "beautiful day") and negative ("damaged building" and "lonely clock") ANPs for the image. They choose different nouns for different sentiments which are compatible with the corresponding image. The last column shows some captions with some errors. For example, Senti-Attend$_{-E_1E_2L_2}$ generates "a beautiful woman in a green shirt standing next to a soccer ball", which is not even semantically compatible with the image. Senti-Attend can handle both topics properly although it has some errors in the generated captions (in Appendix A, Figures A.3 to A.8 provide many examples showing the effectiveness of Senti-Attend in comparison with other models).

6.4.3 Human Sentiment Evaluation

Table 6.5 contains the results. The first three rows are our benchmarks, using ground-truth captions. Clear majorities choose the correct sentiment with these. The results are, however, lower than those of Mathews *et al.* [2], where around 95% had all three voters agreeing on the correct sentiment for neutral and positive captions, and just over 80% for negative captions. Where we presented captions individually, it seems Mathews *et al.* [2] may have presented a positive or negative caption together with a neutral one, and the contrast there may have prompted stronger agreement. The levels of agreement for Senti-Attend captions are as expected lower than for ground-truth captions, but the correct sentiment is still a strong majority for positive captions and a plurality for negative captions, with negative captions also having shown the lowest agreement for ground-truth. In both these cases, the next highest

Table 6.5: Human evaluation results (%). These are given as both as proportion of overall votes, broken down by sentiment; and by the proportion of captions that received a majority vote for a particular sentiment (or None, where there was no majority). The GT rows are for the ground-truth captions.

Image Captions	Sentiment	Propn Overall Votes			Propn Majority Votes			
		Positive	*Neutral*	*Negative*	*Positive*	*Neutral*	*Negative*	*None*
MSCOCO GTs	Neutral	16.50%	76.50%	7.00%	7.35%	85.29%	2.94%	4.42%
SentiCap GTs	Positive	88.00%	6.00%	6.00%	100.00%	0.00%	0.00%	0.00%
SentiCap GTs	Negative	16.00%	14.00%	70.00%	10.53%	5.26%	73.68%	10.53%
Senti-Attend	Positive	68.00%	29.33%	2.67%	68.00%	28.00%	0.00%	4.00%
Senti-Attend	Negative	19.33%	37.33%	43.34%	12.00%	36.00%	44.00%	8.00%

sentiment is neutral. From inspection of the captions, it seems that in some cases Senti-Attend just did not generate any sentiment (e.g. "a man is walking with a horse on the beach" — all three annotators rightly judged this as neutral) but in others cases the sentiment was weak or not perceived (e.g. "a great man is flying a kite on the beach" — intended to be positive, this received one vote for each sentiment). In aggregate, the injected sentiment is the appropriate one by a large margin, with the model falling back to objective descriptions in most other cases.

6.5 Summary

In this work, we have proposed the Senti-Attend model, an image captioning architecture trained end-to-end with novel mechanisms for incorporating embedded sentiment and spatial attention. The model learns both high-level sentiment embeddings, which conditions our caption generator in general, and word-level ones, which influence the word prediction process. Implementing the mechanism with one-hot sentiment representation instead of sentiment embedding results in less effective captions, showing the usefulness of our embedding approach. The Senti-Attend model significantly outperforms state-of-the-art work in this domain using all standard image captioning evaluation metrics. The linguistic analysis demonstrates that the improved performance is due at least in part to selecting suitable and varied adjectives and adjective-noun pairs in the generated captions, focusing on the objects that humans choose to describe with sentiment-infused terms. In addition, human evaluations of sentiment broadly agree that the sentiment is appropriate.

In comparison with our proposed approach in Chapter 5, Senti-Attend achieves higher overall metrics in generating sentiment-bearing captions, reflecting a better preservation of relationship between image and caption content (**RQ 4**, discussed in Chapter 1). It does this by applying an attention mechanism and an embedding approach to effectively learn both semantic and sentiment-bearing information in an end-to-end fashion. The results show that injecting stylistic information in terms of embedding vectors is effective at each time step, as the input of the caption generator, as well as for predicting the next generated word.

Part III

Image Captioning of Adversarial Images

7

Pick-Object-Attack: Type-Specific Adversarial Attack for Object Detection

Many recent studies have shown that deep neural models are vulnerable to adversarial samples: images with imperceptible perturbations, for example, can fool image classifiers. In this chapter, we generate adversarial examples for object detection, which entails detecting bounding boxes around multiple objects present in the image and classifying them at the same time, making it a harder task than against image classification. We specifically aim to attack the widely used Faster R-CNN by changing the predicted label for a particular object in an image: where prior work has targeted one specific object (a stop sign), we generalise to arbitrary objects, with the key challenge being the need to change the labels of *all* bounding boxes for all instances of that object type. To do so, we propose a novel method, named Pick-Object-Attack. Pick-Object-Attack successfully adds perturbations only to bounding boxes for the targeted object, preserving the labels of other detected objects in the image. In terms of perceptibility, the perturbations induced by the method are very small. Furthermore, for the first time, we examine the effect of adversarial attacks on object detection in terms of a downstream task, image captioning; we show that where a method that can modify all object types leads to very obvious changes in captions, the changes from our constrained attack are much less apparent.[1]

7.1 Introduction

Deep learning systems have achieved remarkable success for several computer vision tasks. However, adversarial attacks have brought into question the robustness of such systems. Goodfellow *et al.* [40] and Szegedy *et al.* [41] presented early attacks against image classifiers, using gradient-based techniques to construct inputs with the ability to fool deep learning systems. Since then adversarial attacks have been extensively studied for image classification, including being shown to be transferable across different image classifiers [42]. These attacks are usually categorised into two types (i) Targeted and (ii) Non-targeted. In a targeted attack, the goal is to modify the input so as to make the deep learning system predict a specific class, whereas in a non-targeted attack, the input is modified so as to cause the prediction of any incorrect class.

A more challenging task is to construct adversarial examples that will fool an object detection system, with each image containing multiple objects and multiple proposals for each object; Xie *et al.* [58] provide an analysis of this complexity. Chen *et al.* [59] motivate this task with the example of object detection by an autonomous vehicle to recognise a stop sign and the risks involved in an adversarial attack in that context.

These two works tackle the issue of adversarial attacks against object detection and are the most relevant to our work. Xie *et al.* [58] propose a non-targeted attack where the predictions

[1] The content of this chapter is based on the following paper:

Omid Mohamad Nezami, Akshay Chaturvedi, Mark Dras, Utpal Garain (2020). Pick-Object-Attack: Type-Specific Adversarial Attack for Object Detection. *arXiv preprint arXiv:2006.03184*. (The first two authors contributed equally to this work.)

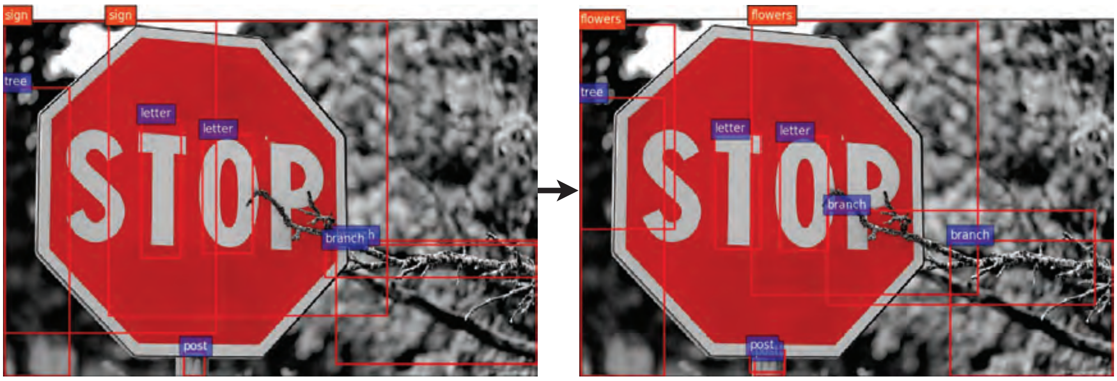


Figure 7.1: Example of our adversarial attack. Pick-Object-Attack adds imperceptible perturbations to the first image (on the left) resulting in the second image (on the right). It succeeds in changing the predicted class of the targeted object from “sign” to “flowers” (shown in orange) while other predicted classes (shown in blue) are unchanged.

of all objects are changed simultaneously. Chen *et al.* [59] propose an attack against the object detector to misclassify only stop sign images; the attack method deliberately adds perceptible noise to the images.

In this chapter, the proposed Pick-Object-Attack aims to change the label of a particular object while keeping the labels of other detected objects unchanged. In this sense, it is a generalisation of Chen *et al.* [59], where there may be a particular object that the attacker wants to be misclassified. More generally, it is often a goal of adversarial attacks to be imperceptible to observers; attacking just a single object, with the small number of bounding boxes involved, minimises the changes to the image relative to modifying all the objects as in Xie *et al.* [58]. Moreover, changes to the image — even if imperceptible to humans — could be perceptible via downstream tasks. For instance, object detection plays a crucial role in the state-of-the-art visual question answering (VQA) and image captioning systems [8]. Changing the entire image may lead to dramatically different answers or captions, which are easily perceptible, and hence alert the user indirectly. Figure 7.1 shows an example of our proposed attack, where only the label of a particular object is changed from “sign” to “flowers” whereas other objects are detected correctly. This is because the perturbation is only added to the bounding boxes with the predicted label “sign”.

In this chapter, we propose both targeted and non-targeted versions of Pick-Object-Attack against Faster R-CNN [60], a widely used and high-performing object detector. Chen *et* *al.* [59] and Xie *et al.* [58] used a version of Faster R-CNN trained on the COCO dataset [160]. In this work, we use Faster R-CNN trained on the Visual Genome dataset [33] which includes a larger set of classes in comparison with the COCO dataset. Bottom-up features obtained from this version of Faster R-CNN are employed in state-of-the-art VQA and image captioning systems [8]. These systems use the bottom-up and top-down attention, explained in §2.3.2, to attend to the bounding boxes in order to generate a caption (or an answer). In terms of image captioning, we study the relation between the adversarially perturbed objects in images and the extent of the changes in the generated captions.

7.2 Related Work

In this section, we give a brief overview on adversarial attack for different vision tasks and discuss related work on adversarial attack against Faster R-CNN in detail.

7.2.1 Adversarial Attack

The generation of adversarial samples was first investigated in the context of deep learning by Szegedy *et al.* [41], who used a gradient-based optimization to arbitrarily manipulate the input sample of a deep neural network for image classification. This manipulation usually aims to find similar samples with differences that are imperceptible to human observers, in order to change the predicted class. Later works [40, 53–57] have led to better methods for generating adversarial samples, using different proposed attack mechanisms, to mislead different classification models. In addition to classification, adversarial samples have also been crafted for other tasks such as image captioning [175]. They studied earlier image captioning models [1, 7] which use features from image classifiers. Here, two types of adversarial examples, targeted keyword and targeted caption, are created using an optimization-based method. The examples can induce image captioning systems to generate pre-defined keywords or captions.

Moreover, adversarial samples have been developed for physical world scenarios where, for example, the printed versions of the samples are used to attack deep learning classification models [176–178]. These approaches mostly target single object images to fool a classifier.

However, in the physical world, we usually face multiple objects in an image. Under such a condition, an attack would be required to fool an object detector, which detects the bounding boxes of objects in addition to classifying them. Eykholt *et al.* [177] discussed that misleading an object detector, such as YOLO [179] and Faster R-CNN [60], is more difficult than misleading an image classifier.

In this chapter, we attack Faster R-CNN, which is a widely used and high-performing system for object detection. We focus on both targeted and non-targeted attacks to mislead Faster R-CNN. Although the possibility of a black-box attack, i.e., no access to the parameters of the model to be attacked, has been investigated in the literature [180], we assume that our attack method has access to the parameters (white-box attack).

7.2.2 Adversarial Attack against Faster R-CNN

Faster R-CNN consists of two stages, a region proposal network (RPN) for detecting the bounding boxes of objects, and a classifier for classifying the boxes [60]. Let $\boldsymbol{I}_{org}$ be an input image with a number (N) of detected bounding boxes, $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N\}$ where $\boldsymbol{h}_i$ is represented by four coordinates. Although the RPN can generate a dynamic number of bounding boxes from the image, an upper bound is usually set on the number of bounding boxes ranked by their confidence levels. The confidence level of each bounding box is calculated using the objectness score and non-maximum suppression (NMS). The RPN predicts an objectness score indicating the probability of an object being present inside the box and the NMS threshold reduces the number of detected boxes. The output of Faster R-CNN will be the classification for the detected boxes, $\{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_N\}$ where $\boldsymbol{g}_i$ indicates the predicted class for $\boldsymbol{h}_i$. $\boldsymbol{g}_i$ is a K-length vector consisting of the predicted probability for the K classes.

Chen *et al.* [59] proposed both *targeted* and *non-targeted* attacks on Faster R-CNN, in the white-box setting, but only for stop sign images. They selected stop signs due to security-related issues in the real world, e.g. self-driving cars. They added *perceptible* perturbations to make their adversarial samples robust after printing. Very recently, Huang *et al.* [181] targeted stop signs, but by adding *perceptible* perturbations around the border of the signs.

In contrast, we target a random set of different objects for both *targeted* and *non-targeted* attacks. We add *imperceptible* perturbations to fool Faster R-CNN.

Xie *et al.* [58] proposed a *non-targeted* attack on Faster R-CNN in the white-box setting. They added *imperceptible* perturbations to *all pixels* in the input image to change the classes for all detected objects. Here, for the adversarial image, the RPN usually generates a different set of bounding boxes, with different scales. The bounding boxes change because adding the perturbations can change their confidence levels. In this work, they change the upper bound of detected boxes from 300 to 3000 to ensure that the transfer of classification error among nearby boxes. In contrast, in our Pick-Object-Attack, we do not increase the upper bound of number of boxes and only add *imperceptible* perturbations to the boxes corresponding to a targeted object to change its predicted class. We do not change the pixel values of other boxes. Unlike Xie *et al.* [58], we study both *targeted* and *non-targeted* attacks.

7.3 Method

7.3.1 Faster R-CNN Model

We evaluate our attack method against Faster R-CNN with ResNet-101, pre-trained on the ImageNet dataset [67], then trained on the object and attribute instances of the Visual Genome dataset [33]. The model leads to the state-of-the-art on different tasks like image captioning and visual question answering [8] in addition to generating a high object detection performance. Previous works [58, 59] studied attacking Faster R-CNN trained on the COCO dataset [160] having only 80 object classes. In comparison, the Visual Genome dataset has 1600 object classes. It includes 3.8M object instances while the COCO dataset includes 1.5M object instances. It also contains 2.8M attributes and 2.3M relationships.

7.3.2 Pick-Object-Attack

Let $\boldsymbol{I}_{org}$ denote the original image. Let N be the number of bounding boxes and K be the number of classes. An object detector can be mathematically expressed as a function $f : \boldsymbol{I} \longrightarrow (\boldsymbol{g}, \boldsymbol{h})$ where $\boldsymbol{g} \in \mathbb{R}^{N \times K}$ denotes the probability distribution for N bounding boxes, and $\boldsymbol{h} \in \mathbb{R}^{N\times 4}$ denotes the predicted coordinates $(x_i, y_i, x_{i+1}, y_{i+1})$ of the bounding boxes. Let o_{pick} denote the selected object to attack and $\boldsymbol{a} \subseteq \{1, 2, .., N\}$ denote the indexes of the boxes with predicted class o_{pick} for the image $\boldsymbol{I}_{org}$. Faster R-CNN rescales the input image so that the shortest size is 600 pixels. Given the original image $\boldsymbol{I}_{org}$ of shape s, our proposed attack generates an adversarial image $\boldsymbol{I}_{adv}$ of the same shape. For an image $\boldsymbol{I}$, we denote the rescaled image (with the shortest side being 600) by $\boldsymbol{I}'$.

Mask Detection As mentioned before, our proposed attack aims to change the label of a targeted object o_{pick}. To do so, we need to update the input image and add perturbations to the region or the bounding boxes including the targeted object by optimizing the attack. Thus, for each attack, we first prepare a binary mask denoted by $\boldsymbol{M}$ which has a same shape as $\boldsymbol{I}_{org}$. $\boldsymbol{M}$ is 1 for bounding boxes with predicted label o_{pick} and is 0 otherwise. Then, for a loss function L and an image $\boldsymbol{I}'$ (obtained by rescaling image $\boldsymbol{I}$), we obtain $\boldsymbol{\nabla}'_{I'}\boldsymbol{L}$ during the *backward pass* given by

$$\boldsymbol{\nabla}'_{I'}\boldsymbol{L} = r\boldsymbol{\nabla}_{I'}\boldsymbol{L} \tag{7.1}$$

where r is the learning rate and $\boldsymbol{\nabla}_{I'}\boldsymbol{L}$ is the gradient of loss L for image $\boldsymbol{I}'$. We resize the gradient $\boldsymbol{\nabla}'_{I'}\boldsymbol{L}$ and apply the mask $\boldsymbol{M}$ according to the following equation

$$\boldsymbol{\nabla}_{I}\boldsymbol{L} = \boldsymbol{M} \odot \mathrm{rescale}(\boldsymbol{\nabla}'_{I'}\boldsymbol{L}, s) \tag{7.2}$$

where $\odot$ denotes bitwise multiplication. For the proposed attack, we use the final obtained gradient $\boldsymbol{\nabla}_{I}\boldsymbol{L}$ for updating image $\boldsymbol{I}$. Here, rescaling means converting the calculated gradient into the original size of the input image so that we can apply the mask. We explain the loss functions for both the non-targeted and targeted attacks below.

Non-Targeted Attack Our goal in the non-targeted attack is to generate an image $\boldsymbol{I}_{adv}$ so that none of the detected boxes have the predicted class o_{pick}. To achieve this, we use the following loss function, L given by

$$L = -\sum_{\mathrm{a}\in\boldsymbol{a}} \log(g_{\mathrm{a},\mathrm{o}_{pick}}) \tag{7.3}$$

Algorithm 2: Non-Tar-Confident/ Non-Tar-Frequent

```
Input:$I_{org}, r, max_{iter}, o_{pick}$
Output:$I_{adv}$
Get mask $M$ from $I_{org}$ and $o_{pick}$
$success \leftarrow$ False
$I \leftarrow I_{org}$
for $j \leftarrow 1$ to $max_{iter}$ do
    Compute $a$ for image $I$
    if $a = \varnothing$ then
        $success \leftarrow$ True
        break
    Compute loss $L$ using $equation$ 7.3
    Compute $\nabla_I L$ using $equation$ 7.2
    $I \leftarrow I + \nabla_I L$
    Truncate image $I$ in the range $[0, 255]$
end for
$I_{adv} \leftarrow I$
return $I_{adv}$
```

where $g_{i,j}$ denotes the predicted probability of the j^{th} class for the i^{th} box. The proposed attack modifies the image I, via gradient-ascent, using the gradient, $\nabla_I L$. We have two variants: one attacks the most confident object (o_{pick} is the most confident object) called Non-Tar-Confident and another one attacks the most frequent object (o_{pick} is the most frequent object) called Non-Tar-Frequent. These are the most challenging setups: choosing a low-confidence or less-frequent object would make it easier to induce a misclassification. These attacks run for max_{iter} iterations for a fixed r, and the attack is considered unsuccessful if we fail to achieve the goal. Algorithm 2 summarizes our non-targeted attack.

Targeted Attack Our goal in the targeted attack is to generate an image I_{adv} so that none of the detected boxes have the predicted class o_{pick} and some of the boxes have the predicted

Algorithm 3: Tar-Confident/ Tar-Frequent

```
Input:$\boldsymbol{I}_{org}, r, max_{iter}, \mathrm{o}_{pick}, \mathrm{k}$
Output:$\boldsymbol{I}_{adv}$
Get mask $\boldsymbol{M}$ from $\boldsymbol{I}_{org}$ and $\mathrm{o}_{pick}$
$success \leftarrow$ False
$\boldsymbol{I} \leftarrow \boldsymbol{I}_{org}$
for $j \leftarrow 1$ to $max_{iter}$ do
    Compute $\boldsymbol{a}$ for image $\boldsymbol{I}$
    if $\boldsymbol{a} = \varnothing$ then
        $\boldsymbol{a} = \arg\max_{\mathrm{u}} g_{\mathrm{u,k}}$ where u are set of
        predicted boxes with positive IoU with mask $\boldsymbol{M}$
    if $\mathrm{k} = \arg\max_{c} g_{\mathrm{a},c}$ for any $\mathrm{a} \in \boldsymbol{a}$ and $\mathrm{o}_{pick} \neq \arg\max_{c} g_{\mathrm{a},c}$ for all $\mathrm{a} \in \boldsymbol{a}$ then
        $success \leftarrow$ True
        break
    Compute loss $L$ using $equation$ 7.4
    Compute $\nabla_{\boldsymbol{I}}\boldsymbol{L}$ using $equation$ 7.2
    $\boldsymbol{I} \leftarrow \boldsymbol{I} - \nabla'_{\boldsymbol{I}}\boldsymbol{L}$
    Truncate image $\boldsymbol{I}$ in the range $[0, 255]$
end for
$\boldsymbol{I}_{adv} \leftarrow \boldsymbol{I}$
return $\boldsymbol{I}_{adv}$
```

class k. Here, k denotes the targeted class for the selected object o_{pick}. To achieve this, we use the following loss function, L given by

$$L = -\sum_{\mathrm{a} \in \boldsymbol{a}} \log(g_{\mathrm{a,k}}) \tag{7.4}$$

where $g_{i,j}$ denotes the predicted probability of the j^{th} class for the i^{th} box. The proposed attack modifies the image $\boldsymbol{I}$, via gradient-descent, using the gradient, $\nabla_{\boldsymbol{I}}\boldsymbol{L}$. Similar to the non-targeted attack, we have two variants: Tar-Confident and Tar-Frequent. These attacks run for max_{iter} iterations for a fixed r, and the attack is considered unsuccessful if our goal is not achieved. Algorithm 3 summarizes our targeted attack. During the attack, if there are no boxes with label o_{pick}, we set $\boldsymbol{a}$ to be the box having the maximum probability of k among all the boxes having a positive Intersection over Union (IoU) with the mask $\boldsymbol{M}$.

7.4 Evaluation Setup

7.4.1 Intrinsic Evaluation

For intrinsic evaluation, we study the success of the attacks against Faster R-CNN, and the magnitude of changes to the images caused by the attacks. In this section, we discuss metrics used to measure the effectiveness of the proposed attacks and the implementation details where the values of the hyperparameters are specified.

Metrics In the following paragraphs, we describe our evaluation metrics for intrinsic evaluation such as success rate, perceptibility, ACAC, ACTC, SSIM and mAP.

Success Rate We use success rate defined as the percentage of attacks that successfully generate adversarial examples. This is a common metric for evaluating adversarial attacks (higher means better performance).

Perceptibility To quantify the perceptibility of change in image, we follow previous work [41, 55, 58] in calculating a score δ for an adversarial perturbation given by

$$\delta_i = \frac{\left\|\boldsymbol{I}_{i,adv} - \boldsymbol{I}_{i,org}\right\|_2}{\sum \boldsymbol{M}_i} \tag{7.5}$$

where $\boldsymbol{I}_{i,adv}$ is the i^{th} adversarial image, $\boldsymbol{I}_{i,org}$ is the i^{th} original image, and $\boldsymbol{M}_i$ is the mask of the i^{th} image in pixels. We normalize the ℓ_2 norm of the image difference by the size of the mask, as our proposed attack adds noise only inside the mask and the size of the mask varies across images (lower means better performance).

ACAC and ACTC We adapt these measures for object detectors from attacks against classifiers [182]. For the non-targeted attacks, Average Confidence of True Class (ACTC) is calculated for object class o_{pick} for all predicted boxes with positive IoU with the mask. This is a performance metric measuring the success of the attack methods to escape from o_{pick} (lower means better performance). For the targeted attacks, Average Confidence of Adversarial Class (ACAC) is calculated for object class k for all predicted boxes with label k. This shows the confidence of the attack methods to generate k (higher means better performance).

The Structural SIMilarity (SSIM) We calculate the Structural SIMilarity (SSIM) to measure the similarity between the original image and the adversarial example since it is a metric which correlates well with human perception. The definition of SSIM($\boldsymbol{I}_{org}, \boldsymbol{I}_{adv}$) between a single original image $\boldsymbol{I}_{org}$ and an adversarial sample $\boldsymbol{I}_{adv}$ is given in [183]. We calculate the mean SSIM (MSSIM) (in the next sections, we call this SSIM for the sake of simplicity) across all pairs of original and adversarial images (higher means better performance):

$$MSSIM = \frac{1}{n}\sum_{i=1}^{n} \text{SSIM}(\boldsymbol{I}_{org,n}, \boldsymbol{I}_{adv,n}) \tag{7.6}$$

mAP Mean average precision (mAP) is calculated for objects outside the mask $\boldsymbol{M}$. The high value of mAP signifies that other objects outside the mask were detected correctly. mAP is calculated using original prediction as ground truth (higher means better performance).

Implementation Details We test our proposed attack on a set of 1000 images randomly selected from the validation set of the MSCOCO dataset [160]. For the targeted attacks, we run attacks for 10 randomly chosen objects (k) per image resulting in 10k samples. We fix the learning rate in equation 7.1 (r) to 10k and set the maximum of iterations (max_{iter}) to 60.

7.4.2 Extrinsic Evaluation

We are also interested in seeing how detectable the adversarial changes are in a downstream task: perturbations might be difficult for a human to detect in an image but can be very obvious from distortions in the downstream task. Image captioning is our downstream task:

captions that are completely unlike the original ones could make manipulation obvious. With respect to the goals of this thesis, exploring different kinds of manipulations allows us to investigate the relationship between these kinds of changes and the effects on the captions.

We use the image captioner of Anderson *et al.* [8], which uses an attention mechanism to attend to the bounding boxes obtained using Faster R-CNN to generate the caption, and gives the state-of-the art results. We investigate how much our Pick-Object-Attack changes captions compared to an object detection attack that modifies the entire image, like that of Xie *et al.* [58]. We note here that our goal differs from image captioning attacks like that of Chen *et al.* [175]. Their goal is to force the captioner to generate specific terms, whereas we just use the image captioner to measure downstream perceptibility of object detection attacks.

Metrics The standard image captioning metrics (including BLEU [164], METEOR [166], CIDEr [100], ROUGE [165] and SPICE [167]): these are used to compare generated captions with human-produced reference captions, and higher scores indicate greater overlap with these reference captions. We will use these slightly differently. Here, we are interested in the overlap of the caption for the adversarial image and *the caption for the original image*, used as the reference caption. A higher score means that the two captions are more similar, i.e. the caption for the adversarial image is less distorted. In addition, we calculate the percentage of cases for which the proposed attacks can remove the keyword corresponding to o$_{pick}$ from the adversarial caption when the keyword is present in the original caption (KWR).

Implementation Details As a comparison to our Pick-Object-Attack, we design a non-targeted attack against *all* objects based on Xie *et al.* [58]. We choose a fixed label for all the boxes and do gradient descent until none of the original objects are detected (the detail is described in Algorithm 4). We name this attack Non-Tar-All. We use the same learning rate as our previous attacks for a fair comparison and increase max_{iter} to 120. Since attacking all objects is a difficult task, we obtained a low success rate for Non-Tar-All (targeted attack against all objects is not feasible). We generate captions using three non-targeted attacks: Non-Tar-All, Non-Tar-Frequent, Non-Tar-Confident and two targeted attacks: Tar-Frequent, Tar-Confident. To do so, we use 100 successful adversarial examples for the non-targeted

Algorithm 4: Non-Tar-All

```
Input:$\boldsymbol{I}_{org}, r, max_{iter}$
Output:$\boldsymbol{I}_{adv}$
$\boldsymbol{c}_{org} \leftarrow$ set of predicted classes for $\boldsymbol{I}_{org}$
Randomly select class z $\notin \boldsymbol{c}_{org}$
$success \leftarrow$ False
$\boldsymbol{I} \leftarrow \boldsymbol{I}_{org}$
for $j \leftarrow 1$ to $max_{iter}$ do
    if $\arg\max_{c} g_{\mathrm{b,c}} \notin \boldsymbol{c}_{org}$ for all boxes b then
        $success \leftarrow$ True
        break
    $L \leftarrow -\sum_{\mathrm{b}} log(g_{\mathrm{b,z}})$
    $\boldsymbol{\nabla}_{I}\boldsymbol{L} \leftarrow$ rescale($\boldsymbol{\nabla}'_{I'}\boldsymbol{L}, \boldsymbol{s}$)
    $\boldsymbol{I} \leftarrow \boldsymbol{I} - \boldsymbol{\nabla}_{I}\boldsymbol{L}$
    Truncate image $\boldsymbol{I}$ in the range $[0, 255]$
end for
$\boldsymbol{I}_{adv} \leftarrow \boldsymbol{I}$
return $\boldsymbol{I}_{adv}$
```

attacks for a shared set having 100 images. We use 1000 successful adversarial examples for the targeted attacks for the shared set (10 per image).

7.5 Results

7.5.1 Intrinsic Evaluation

Quantitative Results Table 7.1 shows the success rate, ACAC and ACTC for our variants of the Pick-Object-Attack. Generally, the non-targeted attacks are more successful compared to the targeted ones. Since we only need to induce a misclassification for the non-targeted attacks, we can achieve a better success rate. Tar-Confident has the lowest success rate. For Tar-Confident, out of 2303 unsuccessful attacks, 1750 attacks are unsuccessful since Tar-Confident cannot find any bounding box with a positive IoU with the mask. This never happens for Tar-Frequent since the mask is larger for the most frequent object in comparison with the most confident one in the image. Out of the cases where there is a bounding box with positive IoU with the mask for Tar-Confident, the success rate is 93.30%. Non-Tar-Confident generates the highest success rate since it does not face this condition. ACAC and ACTC show that the attack approaches can generate high confidence for adversarial and low confidence for original classes. Similar to Xie *et al.* [58], we randomly permute the perturbations generated by the proposed attacks for the adversarial images. This leads to near zero success rates for all attacks showing that the spatial structure of the perturbations plays a major role in fooling Faster R-CNN rather than the magnitude of the perturbations.

Table 7.1 also shows the mAP metric for our proposed attacks. The proposed attacks add perturbations only inside the mask with the purpose of preserving the labels of the bounding boxes outside the mask. However, this perturbation may lead to a different set of bounding boxes by the region proposal network (RPN). These bounding boxes are more likely to have a positive IoU with the mask. Here, mAP shows the impact of perturbation on the bounding boxes outside the mask. As shown in Table 7.1, the proposed attacks mostly do not change the bounding boxes since they generate high mAP values. These results demonstrate that there are two factors impacting on the mAP: the amount of perturbations and the size of the mask. Our targeted attacks add more perturbations to images to fool Faster R-CNN to detect targeted classes and they have lower values for the mAP in comparison with the non-targeted attacks.

Table 7.1: Success Rate (SR), ACAC, ACTC and mAP for different proposed attacks.

APPROACHES	SR	ACAC	ACTC	mAP
Tar-Frequent	89.90%	26.55%	_	86.09%
Tar-Confident	76.97%	24.53%	_	91.97%
Non-Tar-Frequent	95.30%	_	1.25%	94.20%
Non-Tar-Confident	98.40%	_	2.59%	95.47%

Table 7.2: Mean and standard deviation, for the successful cases, of δ and SSIM between original and adversarial images.

APPROACHES	δ		SSIM
	MEAN	STD. DEV.	
Tar-Frequent	1.53×10^{-3}	1.41×10^{-3}	98.53%
Tar-Confident	1.06×10^{-2}	2.82×10^{-2}	98.73%
Non-Tar-Frequent	6.62×10^{-4}	8.97×10^{-4}	99.22%
Non-Tar-Confident	6.65×10^{-3}	1.58×10^{-2}	99.32%

Table 7.3: Success Rate for our proposed attacks after resizing the adversarial images with different scales: 0.6, 0.8, 1.2 and 1.4.

APPROACHES	*Scale*			
	0.6	0.8	1.2	1.4
Tar-Frequent	16.75%	44.70%	72.35%	78.30%
Tar-Confident	11.30%	38.08%	65.72%	74.32%
Non-Tar-Frequent	2.31%	9.76%	26.76%	34.63%
Non-Tar-Confident	14.23%	26.42%	42.78%	52.34%

The attacks against the most frequent objects (Tar-Frequent and Non-Tar-Frequent) also generate lower mAP than the most confident objects (Tar-Confident and Non-Tar-Confident) since the size of the mask for the frequent objects is larger than the confident objects.

As shown in Table 7.2, SSIM is high for all attack approaches. This shows that the approaches are successful in adding imperceptible perturbations to images (the added perturbations are shown in Figure A.9 and Figure A.10 in Appendix A). Table 7.2 also shows the mean and standard deviation, for *successful* cases, of δ. Tar-Confident generates the highest δ. Similarly, Non-Tar-Confident has more δ in comparison with Non-Tar-Frequent. This means that attacking the most confident object is harder than attacking the most frequent object in the image, even though there are typically more instances of the most frequent object. In fact,

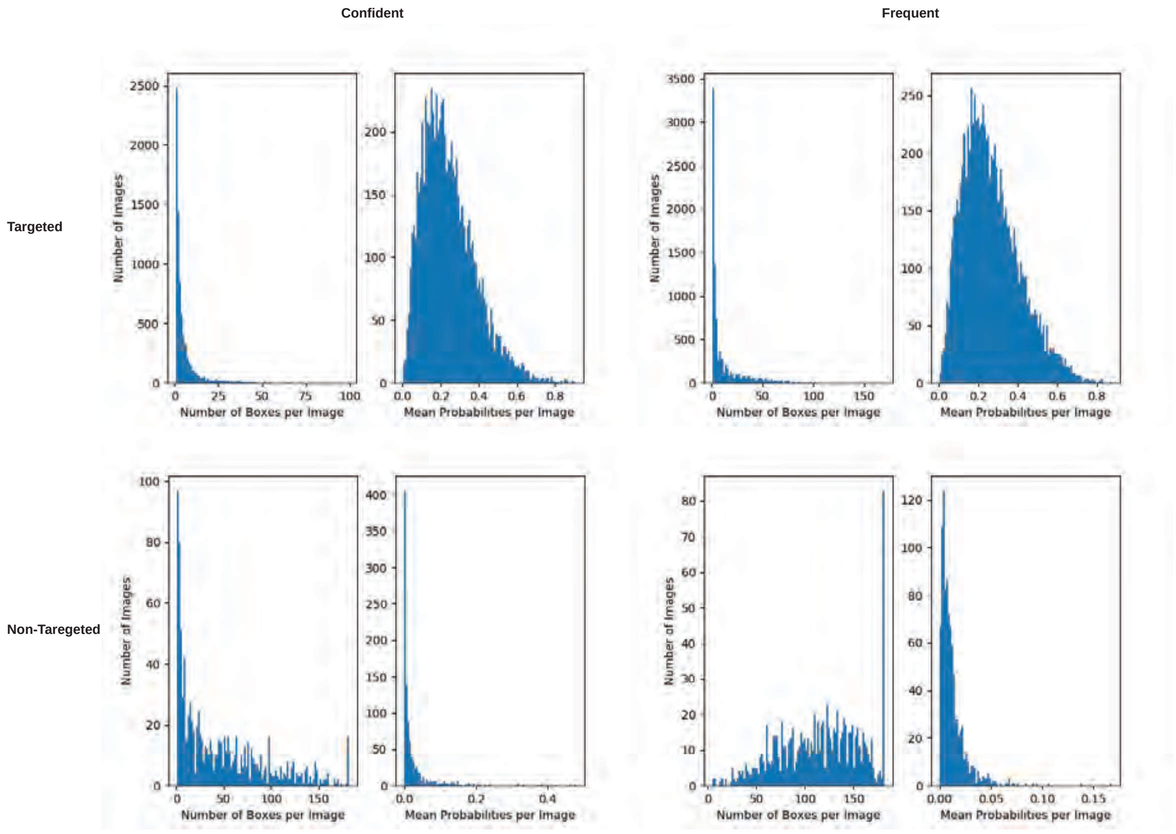


Figure 7.2: The histograms of number of boxes and mean probabilities for the proposed attacks.

from Table 7.2, we can see that Non-Tar-Confident requires more noise than Tar-Frequent.

Table 7.3 shows the robustness of adversarial images generated using the proposed attacks against resizing with different scales. The targeted attacks are more robust in comparison with the non-targeted attacks since they add more perturbations to images to generate particular classes. These results show that the adversarial images are more robust for bigger scales in comparison with smaller scales.

Figure 7.2 shows the histograms of number of boxes and mean probabilities. The first row includes the histogram of the number of boxes, having the predicted label as the targeted class (k), with a positive IoU with the mask. It also includes the histogram of the mean probabilities of the targeted class for the boxes in the targeted attacks. The second row includes the histogram of the number of boxes with a positive IoU with the mask. It also includes the histogram of mean probabilities of the original class (o_{pick}) for the boxes in the non-targeted

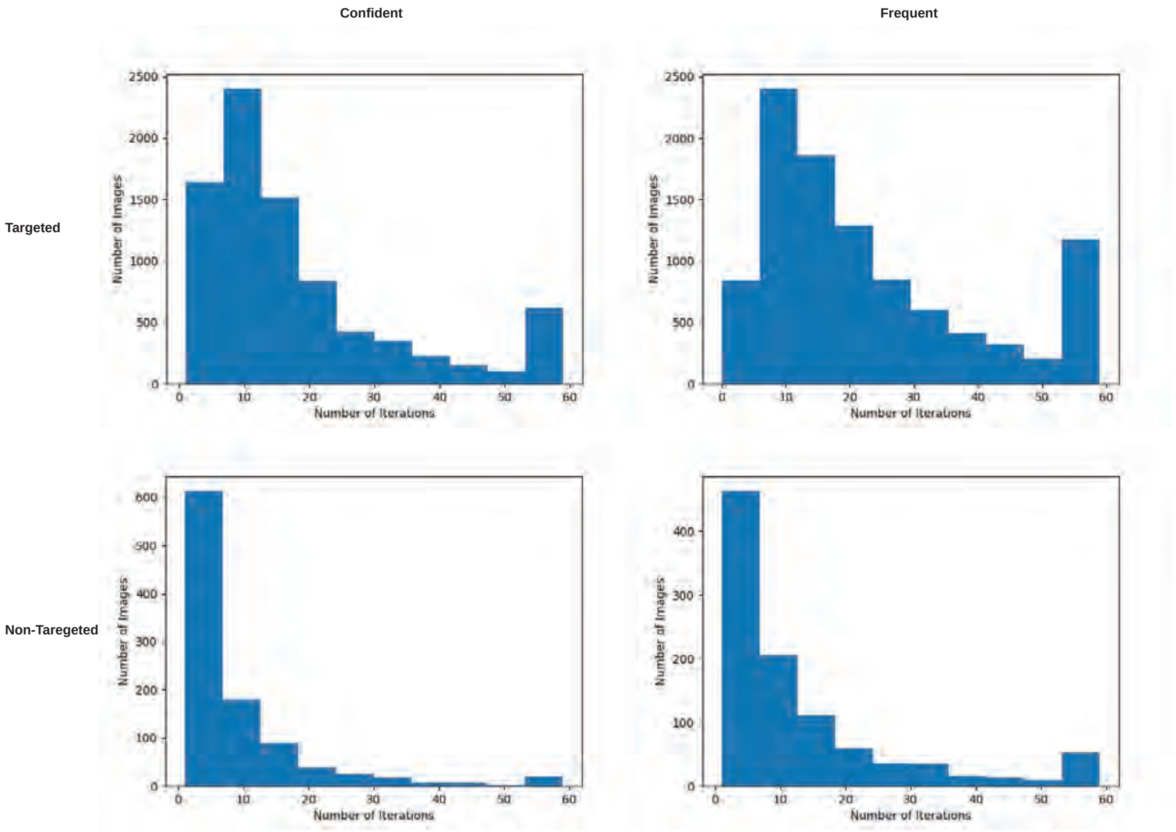


Figure 7.3: The histogram of number of iterations for the proposed attacks.

attacks. This shows that the number of boxes for the attacks against the frequent object is more than the attacks against the confident object. The mean probability of the targeted class for both Tar-Confident and Tar-Frequent are almost similar; however, the mean probability of the original class for Non-Tar-Confident is more than Non-Tar-Frequent.

Figure 7.3 shows the histogram of number of iterations. The first row shows the histogram of the number of iterations for the targeted attacks and the second row for the non-targeted attacks. The maximum number of iterations is 60. If an attack takes 60 iterations, this indicates an unsuccessful attack (we do not show the unsuccessful attacks for Tar-Confident when there is no bounding box having a positive IoU with the mask). The histograms show that attacking the most frequent object requires more iterations in comparison with attacking the most confident object. This is because attacking the most frequent object requires changing the label of more boxes in the image. As expected, the targeted attacks take more iterations than the non-targeted ones.

Qualitative Results Consider the pair of images in the upper row of Figure 7.4. For generating the adversarial image on the right, we targeted “cat” for “sheep” in this example. The outputs of Faster R-CNN (the labels of bounding boxes) show that “sheep” is changed to “cat”. Similarly, in the lower row, the targeted attack approach successfully changes all instances of “bird” to “sign” as shown in the labels of bounding boxes.

7.5.2 Extrinsic Evaluation

Quantitative Results Table 7.4 shows the image captioning metrics for different attack approaches. Since Non-Tar-All changes the whole image, it generates the lowest values for the metrics. The differences are quite dramatic: BLEU-1 is much smaller for Non-Tar-All than for any variant of Pick-Object-Attack; BLEU-3 is zero for Non-Tar-All which shows that there are zero overlaps of trigrams between perturbed and original captions.

Comparing our Pick-Object-Attack variants, Tar-Frequent and Non-Tar-Frequent change more regions in the image because they attack the most frequent object. Thus, they generate lower values in comparison with Tar-Confident and Non-Tar-Confident, respectively. The targeted attacks have lower values in comparison with the non-targeted ones since they add more perturbations to images to generate particular classes. From these results, it is evident that fewer changes in the image lead to fewer changes in the corresponding captions.

In terms of keyword removal (KWR), Tar-Confident and Non-Tar-Confident have higher values in comparison with Tar-Frequent and Non-Tar-Frequent since they add more perturbations to change the label of the most confident object in the image. Tar-Frequent and

Table 7.4: Image captioning metrics and KWR (in %) for different attacks (B-N is BLEU-N).

APPROACHES	B-1	B-2	B-3	B-4	CIDEr	METEOR	ROUGE-L	SPICE	KWR
Non-Tar-All	23.15	6.91	0.00	0.00	5.59	8.19	22.70	0.86	_
Tar-Frequent	44.77	31.82	24.45	19.58	179.26	20.67	44.03	22.98	72.43
Tar-Confident	57.73	47.57	40.92	35.81	331.74	30.30	57.28	40.19	80.17
Non-Tar-Frequent	63.28	53.23	46.27	40.62	389.55	33.06	62.91	48.16	54.00
Non-Tar-Confident	70.39	62.59	56.98	52.48	495.17	38.03	69.39	57.18	76.00

Table 7.5: Mean and standard deviation of ℓ_2-norm of the difference image normalised by the image size, and SSIM between original and adversarial images.

APPROACHES	ℓ_2-norm		SSIM
	MEAN	STD. DEV.	
Non-Tar-All	1.40×10^{-3}	3.97×10^{-4}	98.23%
Tar-Frequent	1.22×10^{-3}	4.86×10^{-4}	98.16%
Tar-Confident	1.20×10^{-3}	4.61×10^{-4}	98.42%
Non-Tar-Frequent	4.55×10^{-4}	2.24×10^{-4}	99.12%
Non-Tar-Confident	4.08×10^{-4}	2.50×10^{-4}	99.23%

Tar-Confident have higher values than their non-targeted versions since they aim to generate a particular class (since o_{pick} is not fixed for Non-Tar-All, we do not provide KWR for this approach).

To study perceptibility of attack, we calculate mean, standard deviation of ℓ_2-norm of the difference image and SSIM between the adversarial images, used for the extrinsic evaluation, and the original images. Since Non-Tar-All modifies the whole image, to compare across attacks, we normalize the ℓ_2-norm of the difference image by the image size for all attacks (we include ℓ_2-norm normalised by mask size, as per Equation 7.5, for direct comparison with Table 7.2 in Table 7.6). As shown in Table 7.5, all methods generate perturbations with low perceptibility. The non-targeted variants of Pick-Object-Attack are less detectable than the targeted ones; Non-Tar-All is more similar to the targeted variants of Pick-Object-Attack, although the perturbations are still small. The perceptibility of Non-Tar-All by these standard

Table 7.6: Mean and standard deviation of δ for adversarial images used in the extrinsic evaluation.

APPROACHES	δ	
	MEAN	STD. DEV.
Tar-Frequent	1.49×10^{-3}	8.40×10^{-4}
Tar-Confident	8.37×10^{-3}	1.85×10^{-2}
Non-Tar-Frequent	5.52×10^{-4}	3.35×10^{-4}
Non-Tar-Confident	4.18×10^{-3}	8.39×10^{-3}

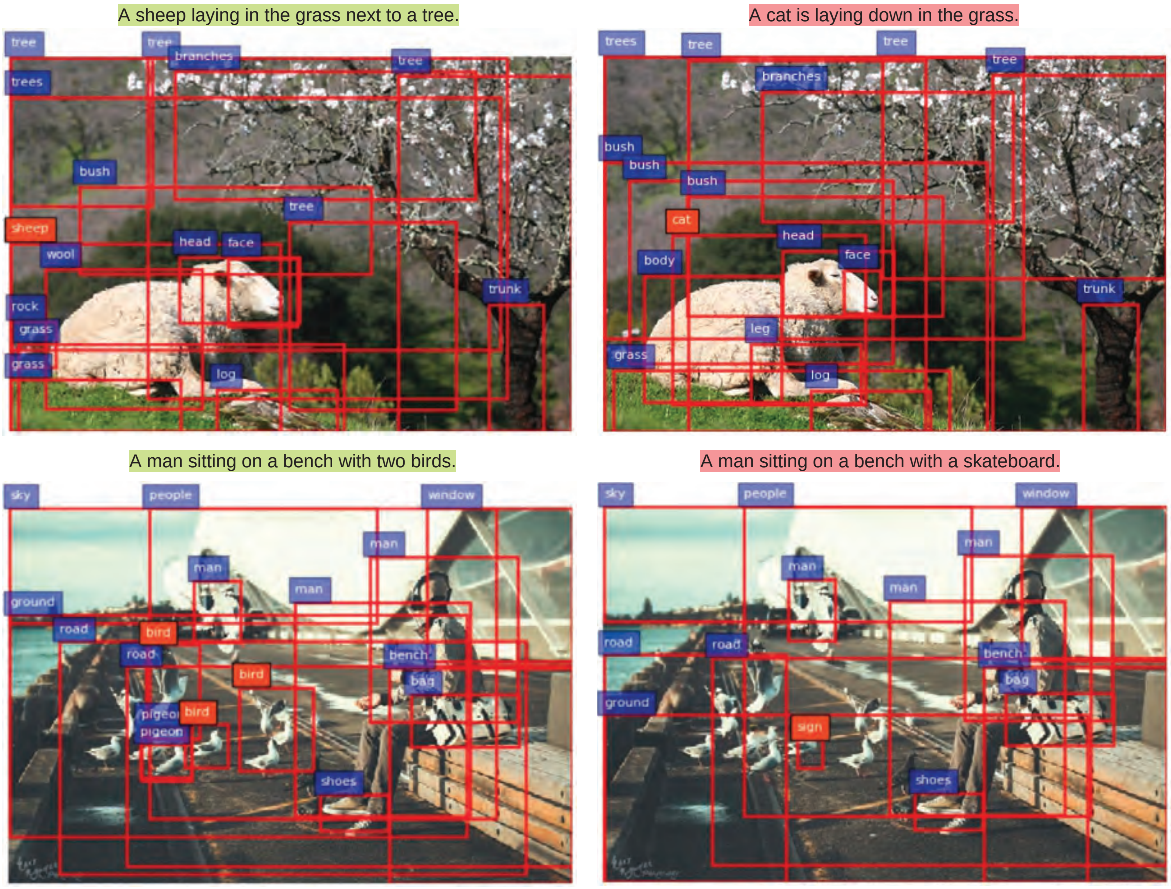


Figure 7.4: The first column includes the original images and the second column includes the adversarial images with their corresponding generated captions. The bounding boxes and the labels on these images are the outputs of Faster R-CNN.

metrics, however, contrasts strongly with the effects on the downstream image captioning task that we describe above, suggesting that the evaluation of how detectable adversarial perturbations are should extend beyond the standard perceptibility metrics.

Qualitative Results Figure 7.4 shows two examples fed into the captioning model (the attention weights of the model for these examples are visualized in Figure A.11 and Figure A.12 in Appendix A). The original image in the first row leads to the caption of “a sheep laying in the grass next to a tree”. As discussed in §7.5.1, a targeted attack changes “sheep” to “cat”; the caption is correspondingly changed to “a cat is laying down in the grass”. This means that our attack against Faster R-CNN can indirectly attack the captioning model to generate a different caption with our targeted class (“cat”). This is also true for the image in

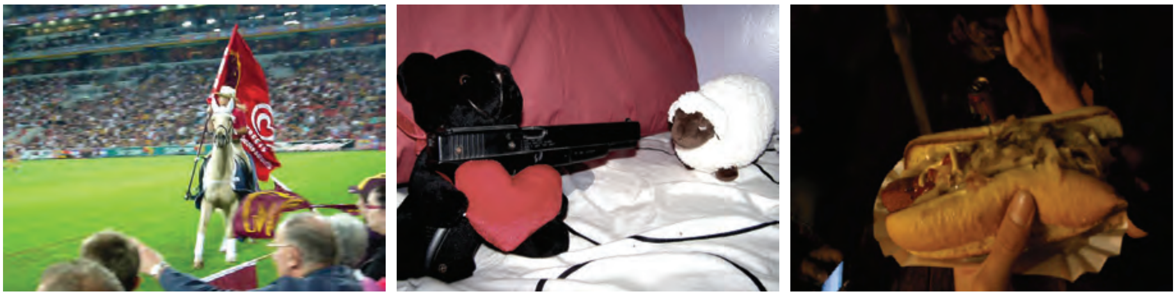


Figure 7.5: Original images corresponding to the sample generated captions by Non-Tar-All.

the second row for generating a different caption. The original caption for the image is "a man sitting on a bench with two birds". As noted in §7.5.1, the attack approach successfully changes "bird" to "sign"; the caption for the adversarial example here is "a man sitting on a bench with a skateboard" which is different from the original one. Although the attack model leads to a new caption, the caption does not include our targeted class ("sign"); "skateboard" is chosen because it is strongly favoured by the language model.

As indicated by Table 7.4, Non-Tar-All changes captions much more dramatically, e.g. "A man riding a horse in front of a crowd" becomes "A bunch of food on a grill with meat being dogs", "Two stuffed teddy bears sitting on a bed" becomes "A blender that is sitting in the water" and "A person holding a hot dog on a bun" becomes "A close up view of an airplane with a knife" for the images in Figure 7.5 from left to right, respectively. Table 7.7 shows more example captions generated for adversarial images using different variants of Pick-Object-Attack and Non-Tar-All with their original captions.

7.6 Summary

We have proposed Pick-Object-Attack, a type-specific adversarial attack for Faster R-CNN, the widely used and high-performing object detector that is used in a state-of-the-art image captioning system. The proposed approach attacks a specific object in an image and aims to preserve the labels of other detected objects in the image. We study both targeted and non-targeted attacks. For each one, we have two variants: attacking the most frequent and the

Table 7.7: Examples generated captions of adversarial images using different attacks with their original captions.

APPROACHES	ORIGINAL CAPTIONS	ADVERSARIAL CAPTIONS
Non-Tar-All	Two stuffed teddy bears sitting on a bed. A man riding a horse in front of a crowd. A person holding a hot dog on a bun.	A blender that is sitting in the water. A bunch of food on a grill with meat being dogs. A close up view of an airplane with a knife.
Tar-Frequent	A donut and a donut sitting on a table. A man jumping a skateboard on a skateboard. Two birds are flying over a building in a city.	A plate with a doughnut and a donut on it. A man jumping through the air with a skateboard. Two birds sitting on a boat in the water.
Tar-Confident	A man riding a horse in front of a crowd. A black and white photo of a city street with cars. A living room with a table and a table.	A person riding a horse in front of a dog. A tower with a clock on top of it. A man taking a picture in a bathroom mirror.
Non-Tar-Frequent	Two cats sitting in a bath tub sink. A black and white photo of a city street with cars. A living room with a table and a table.	A black and white dog is standing in a boat. A black and white photo of a city street with cars. A living room with a couch and a table.
Non-Tar-Confident	A group of people walking around a parking meter. A television and a television in a room. A vase with white flowers on a desk.	A man is holding a parking meter on a pole. A living room with a couch and a chair. A vase with white flowers on a desk.

most confident object in the image. Amongst them, the lowest success rate is obtained by the Tar-Confident because this approach sometimes fails to find bounding boxes within the mask. The results show that attacking the most confident object requires more noise than the most frequent object. The proposed attacks achieve high mAP values for bounding boxes outside the mask which shows that they preserve the labels of other detected objects. In addition to standard perceptibility metrics, we carried out an extrinsic evaluation to study the impact of the adversarial images on the state-of-the-art image captioning system. We compared the captions generated by different variants of Pick-Object-Attack with a baseline attack adapted from [58] that modifies the entire image. The results show that although all models produce perturbations with low perceptibility, the baseline attack produces dramatically distorted captions, in contrast with Pick-Object-Attack, suggesting firstly that extrinsic evaluation on downstream tasks would be a useful complement to standard perceptibility measures; and secondly that the size of perturbations (as measured by perceptibility metrics) are not accurate predictors of effects on generated captions.

Thus, in response to **RQ 5** discussed in Chapter 1, different versions of our proposed attack, which target changing the label of a particular object in an image, make small changes in the generated captions produced by the state-of-the-art image captioning system [8]. However, changing all detected objects in visual content leads to an entirely different image caption. Overall, it suggests that the relationship between changes to the visual source and the resulting generated caption is complex, and needs more work to be fully understood.

8

Conclusions and Future Work

To address the research questions discussed in Chapter 1, we presented several image captioning models controlled using facial expression features, as visually-grounded information, and style-bearing content, as non-grounded information. In the last part of the thesis, we proposed an adversarial attack against Faster R-CNN and analysed its impact on the generated captions by a state-of-the-art captioning system using Faster R-CNN. In this chapter, our key findings and future work are highlighted and discussed.

8.1 Answers and Key Findings

In Chapter 3, we proposed a rich face representation model for engagement recognition using deep learning. To train the model, we collected a new dataset including images of students annotated with engaged and disengaged labels. The model initialized with the weights of our facial expression recognition (FER) model, which generates the state-of-the-art results, to address this research question:

RQ 1. *How can a facial expression recognition model be trained to generate representative and transferable features for other tasks?*

The engagement recognition model generates more effective results using different evaluation metrics in comparison with a comprehensive set of baseline models including the model without the initial weights of the FER model.

In Chapter 4, we used the FER model to extract emotional features for image captioning. We proposed two different kinds of image captioning models called Face-Cap and Face-Attend incorporating the features to address this research question:

RQ 2. *Given the existing image captioning datasets, can incorporating the recognized emotions from facial expression analyses produce better image captions?*

Face-Cap uses the one-hot encoding of facial expression features generated by the FER model. Face-Attend uses an attention mechanism to attend to the convolutional features extracted by the FER model. To train our models, we used a subset of the Flickr 30K image caption dataset [4]. We compared different variants of the models with a comprehensive set of baseline models using different qualitative and quantitative analyses. The results show that both Face-Cap and Face-Attend achieved more effective results in comparison with the baseline models without the FER features. Face-Attend generates more effective results compared to Face-Cap by applying an attention mechanism to attend to the fine-grained features of the FER model. We showed that applying the FER features leads to better image captions.

In Chapter 5, we proposed a novel image captioning model called Attend-GAN to generate style-bearing image captions. The model is designed to address this research question:

RQ 3. *What kind of image captioning model can better generate captions with diverse stylistic patterns?*

Attend-GAN includes two main parts to incorporate style: an attention mechanism to preserve the correlation between an image and a caption along with an adversarial training mechanism to generate more diverse stylistic patterns. Attend-GAN performs better than previous systems on the SentiCap dataset [2]. The results show that Attend-GAN generates image captions with stylistic adjectives and adjective noun pairs which are highly diverse.

In Chapter 6, we proposed a novel image captioning model with style, named Senti-Attend, trained in an end-to-end fashion to address this research question:

RQ 4. *How can an image captioning model be trained in an end-to-end fashion to generate stylistic captions which are still faithful to visual content?*

Senti-Attend is trained on the combination of factual and stylistic captions with an extra input to specify the targeted style. It has an attention mechanism to link visual attention with image captions. It also embeds the targeted style into two embeddings: a high-level embedding to capture the overall style of the generated caption and a word-level embedding to capture the style of each generated word. The results show that Senti-Attend generates the state-of-the-art performances on the SentiCap dataset and leads to style-bearing image captions having strong semantic correlations with visual content.

In Chapter 7, we proposed a novel adversarial attack for Faster R-CNN which named Pick-Object-Attack aiming to change the label of an arbitrary object while preserving the labels of other detected objects in the image. We examined the impact of the attack on the state-of-the-art image captioning model [8] to address this research question:

RQ 5. *How is an adversarial attack against object detection in an image possible, such that it changes the label of a particular object, and what impact does that have on the captions generated by a state-of-the-art image captioning model?*

The results showed that Pick-Object-Attack achieves high success rates over 1000 randomly selected images from the validation set of the MSCOCO dataset [160]. We implemented both targeted and non-targeted attacks by adding imperceptible perturbations to images. Moreover, we compared different variants of Pick-Object-Attack and a baseline model, changing the entire image, in terms of their impacts on the generated captions. The results show that Pick-Object-Attack leads to less distorted captions; other attack that changes the source image to the same extent as measured by standard perceptibility metrics, however, leads to dramatically distorted captions. The question of the precise relationship between changes to the source and impact on the caption is still an open question.

8.2 Future Work

The thesis contains different aspects incorporating extra visually-grounded information such as facial expression features and non-grounded information such as style-bearing content to propose novel controlled image captioning models. It also contains our proposed adversarial attack and shows the attack's impact on the generated captions by a state-of-the-art image captioning model. Using these aspects, there are many potential directions as future work.

In this section, we note some of these ideas:

- **Extending Emotional Content in Image Captioning Models** There is other recent work that explore other aspects of emotional content in images; we note specifically the dataset of You *et al.* [12]. In future work, we are interested in exploring this broader emotional content of images, which is reflected in the NRC Emotion Lexicon [169] we used in our linguistic analysis of captions. The labels in the dataset are provided using the emotional reactions of people in facing a wide variety of images (Figure 8.1). We are specifically interested in exploring whether and how emotional properties of such images can be captured in a representation, and how such a representation can be applied in automatic image captioning.

- **Further Research with Style-Bearing Image Captioning** Future work consists of developing an approach to distinguish between stylized and factual parts of the generated caption like Mathews *et al.* [49]. They proposed an approach including two components: one has the role of generating semantic terms using images and another one has the role of turning these terms into a stylistic sentence using a large dataset of unaligned stylistic text without images. We are especially interested in developing an effective semantic term generator using Faster R-CNN, as our object detection model, to find a correlation between bounding boxes in images and their corresponding semantic terms in the captions. We aim to explore how this can help preserving the semantic aspect

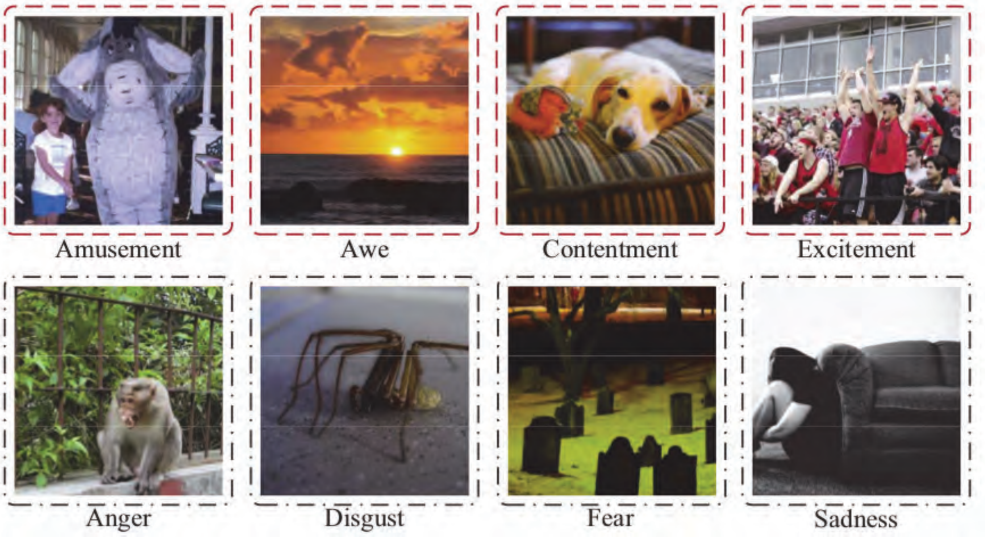


Figure 8.1: Images with positive (the top row) and negative (the bottom row) emotion categories from You *et al.* [12]

of the generated caption while adding the stylistic components to it using the large unaligned dataset.

- **Developing Our Adversarial Attack for Other Downstream Tasks and Purposes** As future work, we plan to explore the impact of our attack against other downstream tasks such as visual question answering (VQA). We also aim to study the more challenging task of attacking attributes as well as objects detected by Faster R-CNN, simultaneously. This might be a difficult scenario since it is relatively straightforward for an object detector to learn a set of attributes corresponding to a specific object. As an example, the object "tree" is most likely to have an attribute "green" hence changing "green tree" to "blue tree" is challenging for the adversary. This will also deepen our understanding of the precise relationship between changes to the source image and the generated caption.

8.3 Summary

The thesis presents novel image captioning models controlling over the caption generation process by incorporating extra visually-grounded and non-grounded information. It also presents an adversarial attack against the object detection task and shows the impact of this attack on the generated captions by a state-of-the-art image captioning model. However, many interesting research questions are remained to be addressed. We hope the thesis as a PhD project serves for reference purposes to control over image captioning systems and examine the robustness of object detectors generating object-based features for image captioning.

A

Appendix

References

[1] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. *Show and Tell: A Neural Image Caption Generator*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164 (IEEE, 2015). xvii, xviii, 2, 3, 19, 21, 22, 26, 28, 62, 88, 128

[2] A. P. Mathews, L. Xie, and X. He. *SentiCap: Generating Image Descriptions with Sentiments*. In *AAAI Conference on Artificial Intelligence*, pp. 3574–3580 (2016). xvii, 3, 4, 5, 8, 9, 28, 29, 31, 63, 88, 89, 90, 95, 97, 103, 106, 108, 112, 113, 120, 151

[3] O. Mohamad Nezami, M. Dras, S. Wan, C. Paris, and L. Hamey. *Towards Generating Stylized Image Captions via Adversarial Training*. In *Pacific Rim International Conference on Artificial Intelligence*, pp. 270–284 (Springer, 2019). xvii, 4, 113

[4] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. *From Image Descriptions to Visual Denotations: New Similarity Metrics for Semantic Inference over Event Descriptions*. Transactions of the Association for Computational Linguistics **2**, 67 (2014). xvii, xx, 2, 5, 7, 12, 63, 65, 150

[5] M. D. Zeiler and R. Fergus. *Visualizing and Understanding Convolutional Networks*. In *European Conference on Computer Vision*, pp. 818–833 (Springer, 2014). xviii, 17, 18, 19

[6] J. Chen and D. Wang. *Long Short-Term Memory for Speaker Generalization in Supervised Speech Separation*. The Journal of the Acoustical Society of America **141**(6), 4705 (2017). xviii, 20

[7] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. In *International Conference on Machine Learning*, pp. 2048–2057 (2015). xviii, 2, 3, 7, 19, 21, 22, 23, 24, 25, 36, 62, 64, 68, 77, 88, 106, 128

[8] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. *Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6077–6086 (2018). xviii, 2, 3, 6, 7, 10, 11, 13, 23, 26, 27, 62, 64, 73, 77, 88, 90, 106, 127, 128, 130, 136, 147, 151

[9] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, and D.-H. Lee. *Challenges in Representation Learning: A Report on Three Machine Learning Contests*. In *International Conference on Neural Information Processing*, pp. 117–124 (Springer, 2013). xviii, 34, 35, 46, 65, 66

[10] K. Simonyan and A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition* (2014). xviii, 19, 46, 47, 52, 54, 66, 68, 108

[11] M. J. Jacobson, C. E. Taylor, and D. Richards. *Computational Scientific Inquiry with Virtual Worlds and Agent-Based Models: New Ways of Doing Science to Learn Science*. Interactive Learning Environments **24**(8), 2080 (2016). xviii, 6, 42, 48

[12] Q. You, J. Luo, H. Jin, and J. Yang. *Building a Large Scale Dataset for Image Emotion Recognition: The Fine Print and The Benchmark*. In *AAAI Conference on Artificial Intelligence*, pp. 308–314 (2016). xxii, 152, 153

[13] B. Woolf, W. Burleson, I. Arroyo, T. Dragon, D. Cooper, and R. Picard. *Affect-Aware Tutors: Recognising and Responding to Student Affect*. International Journal of Learning Technology **4**(3-4), 129 (2009). xxv, 50

[14] S. Aslan, S. E. Mete, E. Okur, E. Oktay, N. Alyuz, U. E. Genc, D. Stanhill, and A. A. Esme. *Human Expert Labeling Process (HELP): Towards a Reliable Higher-Order*

User State Labeling Process and Tool to Assess Student Engagement. Educational Technology pp. 53–59 (2017). xxv, 50

[15] Y. Bengio, A. Courville, and P. Vincent. *Representation Learning: A Review and New Perspectives*. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(8), 1798 (2013). 1, 16

[16] Y. LeCun, Y. Bengio, and G. Hinton. *Deep Learning*. Nature **521**(7553), 436 (2015).

[17] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning* (MIT press, 2016). 1, 16

[18] M. Hodosh, P. Young, and J. Hockenmaier. *Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics*. Journal of Artificial Intelligence Research **47**, 853 (2013). 2, 21

[19] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. *Image Captioning with Semantic Attention*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4651–4659 (IEEE, 2016). 3, 22, 23, 25, 26, 62

[20] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. *Self-Critical Sequence Training for Image Captioning*. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, p. 3 (2017). 2, 3, 23, 25, 37, 62, 106

[21] M. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga. *A Comprehensive Survey of Deep Learning for Image Captioning*. ACM Computing Surveys (CSUR) **51**(6), 118 (2019). 2, 21

[22] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, and B. Plank. *Automatic Description Generation from Images: A Survey of Models, Datasets, and Evaluation Measures*. Journal of Artificial Intelligence Research **55**, 409 (2016). 2, 21

[23] Y. Li, T. Yao, T. Mei, H. Chao, and Y. Rui. *Share-and-Chat: Achieving Human-Level Video Commenting by Search and Multi-View Embedding*. In *Proceedings of the 24th*

ACM International Conference on Multimedia, pp. 928–937 (ACM, 2016). 2, 3, 21, 28, 88

[24] I. Sutskever, O. Vinyals, and Q. V. Le. *Sequence to Sequence Learning with Neural Networks*. In *Advances in Neural Information Processing Systems*, pp. 3104–3112 (2014). 2, 19, 22

[25] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. *Microsoft COCO Captions: Data Collection and Evaluation Server*. arXiv Preprint arXiv:1504.00325 (2015). 2, 111

[26] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. *Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models*. In *IEEE international conference on computer vision*, pp. 2641–2649 (2015). 2

[27] L. Anne Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell. *Deep Compositional Captioning: Describing Novel Object Categories without Paired Training Data*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–10 (2016). 2, 3, 29

[28] S. Venugopalan, L. Anne Hendricks, M. Rohrbach, R. Mooney, T. Darrell, and K. Saenko. *Captioning Images with Diverse Objects*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5753–5761 (2017). 2

[29] J. Lu, J. Yang, D. Batra, and D. Parikh. *Neural Baby Talk*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7219–7228 (2018). 3

[30] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng. *StyleNet: Generating Attractive Visual Captions with Styles*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3137–3146 (2017). 3, 8, 9, 28, 29, 30, 31, 63, 88, 89, 103, 106

[31] T. Chen, Z. Zhang, Q. You, C. Fang, Z. Wang, H. Jin, and J. Luo. *"Factual"or"Emotional": Stylized Image Captioning with Adaptive Learning and Attention*.

In *European Conference on Computer Vision (ECCV)*, pp. 519–535 (2018). 3, 8, 9, 28, 29, 31, 63, 88, 90, 97, 103, 106, 112, 113

[32] B. Huber, D. McDuff, C. Brockett, M. Galley, and B. Dolan. *Emotional Dialogue Generation using Image-Grounded Language Models*. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 277 (ACM, 2018). 3, 28

[33] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, and D. A. Shamma. *Visual Genome: Connecting Language and Vision using Crowdsourced Dense Image Annotations*. International Journal of Computer Vision **123**(1), 32 (2017). 3, 26, 128, 130

[34] L. Melas-Kyriazi, A. Rush, and G. Han. *Training for Diversity in Image Paragraph Captioning*. In *2018 Conference on Empirical Methods in Natural Language Processing*, pp. 757–761 (2018). 3

[35] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing. *Toward Controlled Generation of Text*. In *International Conference on Machine Learning*, pp. 1587–1596 (2017). 3, 7, 32, 63, 64, 107

[36] C. Lisetti. *Affective Computing* (1998). 5, 62

[37] M. Scheutz, P. Schermerhorn, J. Kramer, and C. Middendorff. *The Utility of Affect Expression in Natural Language Interactions in Joint Human-Robot Tasks*. In *ACM SIGCHI/SIGART Human-Robot Interaction: Proceeding of the First ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, vol. 2, pp. 226–233 (2006).

[38] T. Jay and K. Janschewitz. *Filling the Emotion Gap in Linguistic Theory: Commentary on Potts' Expressive Dimension*. Theoretical Linguistics **33**(2), 215 (2007).

[39] Jonczyk and R. Jończyk. *Affect-Language Interactions in Native and Non-Native English Speakers* (Springer, 2016). 5

[40] I. J. Goodfellow, J. Shlens, and C. Szegedy. *Explaining and Harnessing Adversarial Examples*. Stat **1050**, 20 (2015). 5, 9, 126, 128

[41] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. *Intriguing Properties of Neural Networks*. In *International Conference on Learning Representations* (2014). 9, 126, 128, 134

[42] Y. Liu, X. Chen, C. Liu, and D. Song. *Delving into Transferable Adversarial Examples and Black-box Attacks*. In *Proceedings of 5th International Conference on Learning Representations* (2017). 5, 9, 126

[43] A. Kamath, A. Biswas, and V. Balasubramanian. *A Crowdsourced Approach to Student Engagement Recognition in E-Learning Environments*. In *Winter Conference on Applications of Computer Vision*, pp. 1–9 (IEEE, 2016). 6, 42, 43, 45, 53

[44] N. Alyuz, E. Okur, E. Oktay, U. Genc, S. Aslan, S. E. Mete, B. Arnrich, and A. A. Esme. *Semi-Supervised Model Personalization for Improved Detection of Learner's Emotional Engagement*. In *International Conference on Multimodal Interaction*, pp. 100–107 (ACM, 2016). 6, 42, 43, 45

[45] H. Monkaresi, N. Bosch, R. A. Calvo, and S. K. D'Mello. *Automated Detection of Engagement using Video-Based Estimation of Facial Expressions and Heart Rate*. IEEE Transactions on Affective Computing **8**(1), 15 (2017). 6, 42, 43, 45, 56

[46] N. Bosch. *Detecting Student Engagement: Human Versus Machine*. In *International Conference on User Modeling, Adaptation, and Personalization*, pp. 317–320 (ACM, 2016). 7, 44

[47] H. O'Brien. *Theoretical Perspectives on User Engagement*. In *Why Engagement Matters*, pp. 1–26 (Springer, 2016). 7, 42, 44

[48] Q. You, H. Jin, and J. Luo. *Image Captioning at Will: A Versatile Scheme for Effectively Injecting Sentiments into Image Descriptions*. arXiv Preprint arXiv:1801.10121 (2018). 7, 30, 63, 64, 69, 70, 77, 107, 112, 113

[49] A. Mathews, L. Xie, and X. He. *SemStyle: Learning to Generate Stylised Image Captions using Unaligned Text*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8591–8600 (2018). 8, 9, 28, 31, 88, 90, 106, 152

[50] J. W. Pennebaker and L. A. King. *Linguistic Styles: Language Use as an Individual Difference*. Journal of Personality and Social Psychology **77**(6), 1296 (1999). 8, 28

[51] H. Zhou, M. Huang, T. Zhang, X. Zhu, and B. Liu. *Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory*. In *Thirty-Second AAAI Conference on Artificial Intelligence* (2018). 9, 32, 114

[52] S. Ghosh, M. Chollet, E. Laksana, L.-P. Morency, and S. Scherer. *Affect-LM: A Neural Language Model for Customizable Affective Text Generation*. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 634–642 (2017). 9, 32, 114

[53] N. Carlini and D. Wagner. *Towards Evaluating the Robustness of Neural Networks*. In *IEEE Symposium on Security and Privacy*, pp. 39–57 (IEEE, 2017). 9, 128

[54] K. Eykholt and A. Prakash. *Designing Adversarially Resilient Classifiers using Resilient Feature Engineering*. arXiv Preprint arXiv:1812.06626 (2018).

[55] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. *Deepfool: A Simple and Accurate Method to Fool Deep Neural Networks*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582 (2016). 134

[56] A. Nguyen, J. Yosinski, and J. Clune. *Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 427–436 (2015).

[57] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. *The Limitations of Deep Learning in Adversarial Settings*. In *IEEE European Symposium on Security and Privacy*, pp. 372–387 (IEEE, 2016). 9, 128

[58] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille. *Adversarial Examples for Semantic Segmentation and Object Detection*. In *IEEE International Conference on Computer Vision*, pp. 1369–1378 (2017). 9, 13, 126, 127, 128, 130, 134, 136, 138, 146

[59] S.-T. Chen, C. Cornelius, J. Martin, and D. H. P. Chau. *ShapeShifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector*. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 52–68 (Springer, 2018). 9, 126, 127, 128, 129, 130

[60] S. Ren, K. He, R. Girshick, and J. Sun. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. In *Advances in Neural Information Processing Systems*, pp. 91–99 (2015). 10, 26, 127, 129

[61] Y. Bengio, D.-H. Lee, J. Bornschein, T. Mesnard, and Z. Lin. *Towards Biologically Plausible Deep Learning*. arXiv Preprint arXiv:1502.04156 (2015). 16

[62] J. Schmidhuber. *Deep Learning in Neural Networks: An Overview*. Neural Networks **61**, 85 (2015). 16, 18

[63] G. E. Hinton. *Learning Distributed Representations of Concepts*. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, vol. 1, p. 12 (Amherst, MA, 1986). 16

[64] A. Krizhevsky, I. Sutskever, and G. E. Hinton. *ImageNet Classification with Deep Convolutional Neural Networks*. In *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012). 16, 17, 53

[65] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. *Distributed Representations of Words and Phrases and Their Compositionality*. In *Advances in Neural Information Processing Systems*, pp. 3111–3119 (2013). 16, 19

[66] G. E. Dahl, D. Yu, L. Deng, and A. Acero. *Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition*. IEEE Transactions on audio, speech, and language processing **20**(1), 30 (2011). 16

[67] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. *ImageNet: A Large-Scale Hierarchical Image Database*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255 (2009). 16, 97, 130

[68] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. *Backpropagation Applied to Handwritten Zip Code Recognition*. Neural Computation **1**(4), 541 (1989). 17

[69] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. *Going Deeper with Convolutions*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9 (2015). 19

[70] K. He, X. Zhang, S. Ren, and J. Sun. *Deep Residual Learning for Image Recognition*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016). 19, 97, 108

[71] A. Graves. *Generating Sequences with Recurrent Neural Networks*. arXiv Preprint arXiv:1308.0850 (2013). 19

[72] C. D. Manning. *Computational Linguistics and Deep Learning*. Computational Linguistics **41**(4), 701 (2015).

[73] G. Goth. *Deep or Shallow, NLP is Breaking out*. Communications of the ACM **59**(3), 13 (2016).

[74] Y. Goldberg. *A Primer on Neural Network Models for Natural Language Processing*. Journal of Artificial Intelligence Research **57**, 345 (2016). 19

[75] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. arXiv Preprint arXiv:1406.1078 (2014). 19

[76] S. Hochreiter and J. Schmidhuber. *Long Short-Term Memory*. Neural Computation **9**(8), 1735 (1997). 19

[77] F. A. Gers, J. Schmidhuber, and F. Cummins. *Learning to Forget: Continual Prediction with LSTM*. Neural Computation **12**(10), 2451 (2000). 19

[78] D. Bahdanau, K. Cho, and Y. Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. arXiv Preprint arXiv:1409.0473 (2014). 20, 23

[79] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. *Attention Is All You Need*. In *Advances in Neural Information Processing Systems*, pp. 5998–6008 (2017). 20, 23

[80] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. *Every Picture Tells a Story: Generating Sentences from Images*. In *European Conference on Computer Vision*, pp. 15–29 (Springer, 2010). 21, 22

[81] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi. *Composing Simple Image Descriptions using Web-Scale N-Grams*. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pp. 220–228 (Association for Computational Linguistics, 2011). 21

[82] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. *Baby Talk: Understanding and Generating Simple Image Descriptions*. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(12), 2891 (2013). 21, 22

[83] V. Ordonez, G. Kulkarni, and T. L. Berg. *Im2Text: Describing Images using 1 Million Captioned Photographs*. In *Advances in Neural Information Processing Systems*, pp. 1143–1151 (2011). 21

[84] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. *Grounded Compositional Semantics for Finding and Describing Images with Sentences*. Transactions of the Association for Computational Linguistics **2**, 207 (2014).

[85] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik. *Improving Image-Sentence Embeddings using Large Weakly Annotated Photo Collections*. In *European Conference on Computer Vision*, pp. 529–545 (Springer, 2014). 21

[86] R. Kiros, R. Salakhutdinov, and R. S. Zemel. *Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models*. arXiv Preprint arXiv:1411.2539 (2014). 21

[87] X. Chen and C. Lawrence Zitnick. *Mind's Eye: A Recurrent Visual Representation for Image Caption Generation*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2422–2431 (IEEE, 2015). 22

[88] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. *Long-Term Recurrent Convolutional Networks for Visual Recognition and Description*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2625–2634 (IEEE, 2015).

[89] J. Johnson, A. Karpathy, and L. Fei-Fei. *DenseCap: Fully Convolutional Localization Networks for Dense Captioning*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4565–4574 (2016). 26, 62

[90] A. Karpathy and L. Fei-Fei. *Deep Visual-Semantic Alignments for Generating Image Descriptions*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3128–3137 (IEEE, 2015).

[91] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. *Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN)*. arXiv Preprint arXiv:1412.6632 (2014). 22

[92] D. Elliott and F. Keller. *Image Description using Visual Dependency Representations*. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1292–1302 (2013). 22

[93] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi. *Collective Generation of Natural Image Descriptions*. In *ACL*, pp. 359–368 (Association for Computational Linguistics, 2012). 22

[94] C. Koch and S. Ullman. *Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry*. In *Matters of Intelligence*, pp. 115–141 (Springer, 1987). 23

[95] M. Corbetta and G. L. Shulman. *Control of Goal-Directed and Stimulus-Driven Attention in the Brain*. Nature Reviews Neuroscience **3**(3), 201 (2002).

[96] M. W. Spratling and M. H. Johnson. *A Feedback Model of Visual Attention*. Journal of Cognitive Neuroscience **16**(2), 219 (2004).

[97] R. A. Rensink. *The Dynamic Representation of Scenes*. Visual Cognition **7**(1-3), 17 (2000). 23

[98] V. Mnih, N. Heess, and A. Graves. *Recurrent Models of Visual Attention*. In *Advances in Neural Information Processing Systems*, pp. 2204–2212 (2014). 23

[99] J. Ba, V. Mnih, and K. Kavukcuoglu. *Multiple Object Recognition with Visual Attention*. arXiv Preprint arXiv:1412.7755 (2014). 23

[100] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. *CIDEr: Consensus-Based Image Description Evaluation*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4566–4575 (IEEE, 2015). 25, 74, 96, 112, 136

[101] R. J. Williams. *Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning*. Machine Learning **8**(3-4), 229 (1992). 25

[102] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*, vol. 135 (MIT press Cambridge, 1998). 25

[103] Y. Yu, H. Ko, J. Choi, and G. Kim. *End-to-End Concept Word Detection for Video Captioning, Retrieval, and Question Answering*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3261–3269 (IEEE, 2017). 25

[104] M. Jaderberg, K. Simonyan, and A. Zisserman. *Spatial Transformer Networks*. In *Advances in Neural Information Processing Systems*, pp. 2017–2025 (2015). 26

[105] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein. *ImageNet Large Scale Visual Recognition Challenge*. International Journal of Computer Vision **115**(3), 211 (2015). 26, 68

[106] J. Jin, K. Fu, R. Cui, F. Sha, and C. Zhang. *Aligning Where to See and What to Tell: Image Caption with Region-Based Attention and Scene Factorization*. arXiv Preprint arXiv:1506.06272 (2015). 26

[107] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. *Selective Search for Object Recognition*. International Journal of Computer Vision **104**(2), 154 (2013). 26

[108] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser. *Multi-Task Sequence to Sequence Learning*. In *International Conference on Learning Representations* (2016). 29

[109] B. Pang and L. Lee. *Opinion Mining and Sentiment Analysis*. Found. Trends Inf. Retr. **2**(1-2), 1 (2008). URL http://dx.doi.org/10.1561/1500000011. 29, 63

[110] A. Radford, R. Jozefowicz, and I. Sutskever. *Learning to Generate Reviews and Discovering Sentiment*. arXiv Preprint arXiv:1704.01444 (2017). 30

[111] E. Reiter and R. Dale. *Building Natural Language Generation Systems* (Cambridge University Press, 2000). 31

[112] A. Gatt and E. Krahmer. *Survey of the State of the Art in Natural Language Generation: Core Tasks, Applications and Evaluation*. Journal of Artificial Intelligence Research **61**, 65 (2018). 31

[113] D. P. Kingma and M. Welling. *Auto-Encoding Variational Bayes*. Stat **1050**, 10 (2014). 32

[114] B. Fasel and J. Luettin. *Automatic Facial Expression Analysis: A Survey*. Pattern recognition **36**(1), 259 (2003). 33, 63

[115] P. Ekman. *Darwin and Facial Expression: A Century of Research in Review* (Ishk, 2006). 33

[116] T. M. Field, R. Woodson, R. Greenberg, and D. Cohen. *Discrimination and Imitation of Facial Expression by Neonates*. Science **218**(4568), 179 (1982). 33

[117] T. Kanade, J. F. Cohn, and Y. Tian. *Comprehensive Database for Facial Expression Analysis*. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pp. 46–53 (IEEE, 2000).

[118] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. *A 3D Facial Expression Database for Facial Behavior Research*. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pp. 211–216 (IEEE, 2006).

[119] A. J. Fridlund. *Human Facial Expression: An Evolutionary View* (Academic Press, 2014).

[120] E. Sariyanidi, H. Gunes, and A. Cavallaro. *Automatic Analysis of Facial Affect: A Survey of Registration, Representation, and Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(6), 1113 (2015). 33, 63

[121] Y.-I. Tian, T. Kanade, and J. F. Cohn. *Recognizing Action Units for Facial Expression Analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(2), 97 (2001). 33

[122] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. *A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions*. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(1), 39 (2008). 33

[123] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, and A. M. Dobaie. *Facial Expression Recognition via Learning Deep Sparse Autoencoders*. Neurocomputing **273**, 643 (2018). 33

[124] Y. Tang. *Deep Learning using Linear Support Vector Machines*. In *International Conference on Machine Learning* (2013). 34, 46

[125] S. E. Kahou, X. Bouthillier, P. Lamblin, C. Gulcehre, V. Michalski, K. Konda, S. Jean, P. Froumenty, Y. Dauphin, and N. Boulanger-Lewandowski. *Emonets: Multimodal Deep Learning Approaches for Emotion Recognition in Video*. Journal on Multimodal User Interfaces **10**(2), 99 (2016). 34

[126] S. E. Kahou, C. Pal, X. Bouthillier, P. Froumenty, Ç. Gülçehre, R. Memisevic, P. Vincent, A. Courville, Y. Bengio, and R. C. Ferrari. *Combining Modality Specific Deep Neural Networks for Emotion Recognition in Video*. In *International Conference on Multimodal Interaction*, pp. 543–550 (ACM, 2013). 34

[127] Z. Yu and C. Zhang. *Image Based Static Facial Expression Recognition with Multiple Deep Network Learning*. In *International Conference on Multimodal Interaction*, pp. 435–442 (ACM, 2015). 34, 46, 65

[128] B.-K. Kim, S.-Y. Dong, J. Roh, G. Kim, and S.-Y. Lee. *Fusing Aligned and Non-Aligned Face Information for Automatic Affect Recognition in the Wild: A Deep Learning Approach*. In *CVPR Workshops*, pp. 48–57 (IEEE, 2016). 34, 46, 65

[129] Z. Zhang, P. Luo, C.-C. Loy, and X. Tang. *Learning Social Relation Traits from Face Images*. In *International Conference on Computer Vision*, pp. 3631–3639 (2015). 35

[130] C. Pramerdorfer and M. Kampel. *Facial Expression Recognition using Convolutional Neural Networks: State of the Art*. arXiv Preprint arXiv:1612.02903 (2016). 35, 46, 47, 65, 66

[131] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. *Generative Adversarial Nets*. In *Advances in Neural Information Processing Systems*, pp. 2672–2680 (2014). 35, 90, 94

[132] A. Radford, L. Metz, and S. Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. arXiv Preprint arXiv:1511.06434 (2015). 35

[133] L. Yu, W. Zhang, J. Wang, and Y. Yu. *SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient*. In *AAAI Conference on Artificial Intelligence*, pp. 2852–2858 (2017). 35, 36, 90, 94

[134] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. *Image-to-Image Translation with Conditional Adversarial Networks*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134 (2017).

[135] K. Wang and X. Wan. *SentiGAN: Generating Sentimental Texts via Mixture Adversarial Networks*. In *International Joint Conferences on Artificial Intelligence*, pp. 4446–4452 (2018). 35, 37

[136] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. *InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets*. In *Advances in Neural Information Processing Systems*, pp. 2172–2180 (2016). 35

[137] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, and M. Lanctot. *Mastering the Game of Go with Deep Neural Networks and Tree Search*. Nature **529**(7587), 484 (2016). 36

[138] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola. *Style Transfer from Non-Parallel Text by Cross-Alignment*. In *Advances in Neural Information Processing Systems*, pp. 6830–6841 (2017). 36

[139] A. M. Lamb, A. G. A. P. Goyal, Y. Zhang, S. Zhang, A. C. Courville, and Y. Bengio. *Professor Forcing: A New Algorithm for Training Recurrent Networks*. In *Advances in Neural Information Processing Systems*, pp. 4601–4609 (2016). 36

[140] X. Liang, Z. Hu, H. Zhang, C. Gan, and E. P. Xing. *Recurrent Topic-Transition GAN for Visual Paragraph Generation*. arXiv Preprint arXiv:1703.07022 (2017). 36, 37

[141] M. Arjovsky, S. Chintala, and L. Bottou. *Wasserstein Generative Adversarial Networks*. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 214–223 (2017). 36, 94

[142] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville, and Y. Bengio. *An Actor-Critic Algorithm for Sequence Prediction*. arXiv Preprint arXiv:1607.07086 (2016). 37

[143] R. Luo, B. Price, S. Cohen, and G. Shakhnarovich. *Discriminability Objective for Training Descriptive Captions*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6964–6974 (2018). 37

[144] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. *Sequence Level Training with Recurrent Neural Networks*. arXiv Preprint arXiv:1511.06732 (2015). 37

[145] R. Shetty, M. Rohrbach, L. Anne Hendricks, M. Fritz, and B. Schiele. *Speaking the Same Language: Matching Machine to Human Captions by Adversarial Training*. In *IEEE International Conference on Computer Vision*, pp. 4135–4144 (2017). 38

[146] A. Kapoor, S. Mota, and R. W. Picard. *Towards a Learning Companion that Recognizes Affect*. In *AAAI Fall symposium*, pp. 2–4 (2001). 43

[147] J. Whitehill, Z. Serpell, Y.-C. Lin, A. Foster, and J. R. Movellan. *The Faces of Engagement: Automatic Recognition of Student Engagement from Facial Expressions*. IEEE Transactions on Affective Computing **5**(1), 86 (2014). 43, 44, 45, 52, 56

[148] N. Bosch, S. D'Mello, R. Baker, J. Ocumpaugh, V. Shute, M. Ventura, L. Wang, and W. Zhao. *Automatic Detection of Learning-Centered Affective States in the Wild*. In *ACM International Conference on Intelligent User Interfaces*, pp. 379–388 (ACM, 2015). 43, 45, 56

[149] J. Grafsgaard, J. B. Wiggins, K. E. Boyer, E. N. Wiebe, and J. Lester. *Automatically Recognizing Facial Expression: Predicting Engagement and Frustration*. In *Educational Data Mining 2013* (2013). 44

[150] N. Bosch, S. K. D'mello, J. Ocumpaugh, R. S. Baker, and V. Shute. *Using Video to Automatically Detect Learner Affect in Computer-Enabled Classrooms*. ACM Transactions on Interactive Intelligent Systems **6**(2), 17 (2016). 45

[151] A. D'Cunha, A. Gupta, K. Awasthi, and V. Balasubramanian. *Daisee: Towards User Engagement Recognition in the Wild*. arXiv Preprint arXiv:1609.01885 (2016). 45

[152] D. E. King. *Dlib-ML: A Machine Learning Toolkit*. Journal of Machine Learning Research **10**(Jul), 1755 (2009). 49, 65

[153] N. Dalal and B. Triggs. *Histograms of Oriented Gradients for Human Detection*. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893 (IEEE, 2005). 53

[154] V. Nair and G. E. Hinton. *Rectified Linear Units Improve Restricted Boltzmann Machines*. In *International Conference on Machine Learning*, pp. 807–814 (2010). 53, 54

[155] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. The Journal of Machine Learning Research **15**(1), 1929 (2014). 53, 54

[156] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua. *SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6298–6306 (IEEE, 2017). 62, 97

[157] J. Lu, C. Xiong, D. Parikh, and R. Socher. *Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning*. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 6, p. 2 (2017).

[158] Y. Tian, X. Wang, J. Wu, R. Wang, and B. Yang. *Multi-Scale Hierarchical Residual Network for Dense Captioning*. Journal of Artificial Intelligence Research **64**, 181 (2019). 62

[159] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. *The Extended Cohn-Kanade Dataset (CK+): A Complete Dataset for Action Unit and Emotion-Specified Expression*. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pp. 94–101 (IEEE, 2010). 64

[160] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. *Microsoft COCO: Common Objects in Context*. In *European Conference on Computer Vision*, pp. 740–755 (Springer, 2014). 66, 88, 95, 128, 130, 135, 151

[161] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. *Analysis of representations for domain adaptation*. In *Advances in neural information processing systems*, pp. 137–144 (2007). 66

[162] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. *Domain adaptation via transfer component analysis*. IEEE Transactions on Neural Networks **22**(2), 199 (2010). 66

[163] S. Ioffe and C. Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. In *International Conference on Machine Learning*, pp. 448–456 (2015). 66, 94

[164] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. *BLEU: A Method for Automatic Evaluation of Machine Translation*. In *Annual Meeting of the Association for Computational Linguistics*, pp. 311–318 (Association for Computational Linguistics, 2002). 74, 96, 112, 136

[165] C.-Y. Lin. *Rouge: A Package for Automatic Evaluation of Summaries*. Text Summarization Branches Out (2004). 74, 96, 112, 136

[166] M. Denkowski and A. Lavie. *Meteor Universal: Language Specific Translation Evaluation for any Target Language*. In *Conference on Machine Translation*, pp. 376–380 (2014). 74, 96, 112, 136

[167] P. Anderson, B. Fernando, M. Johnson, and S. Gould. *SPICE: Semantic Propositional Image Caption Evaluation*. In *European Conference on Computer Vision*, pp. 382–398 (Springer, 2016). 74, 78, 96, 98, 112, 114, 136

[168] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network*. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 173–180 (Association for Computational Linguistics, 2003). 75, 96

[169] S. M. Mohammad and P. D. Turney. *Crowdsourcing a Word–Emotion Association Lexicon*. Computational Intelligence **29**(3), 436 (2013). 75, 152

[170] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. arXiv Preprint arXiv:1412.6980 (2014). 77, 97, 114

[171] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht. *The Marginal Value of Adaptive Gradient Methods in Machine Learning*. In *Advances in Neural Information Processing Systems*, pp. 4151–4161 (2017). 78

[172] T. Tieleman and G. Hinton. *Lecture 6.5-RMSprop: Divide the Gradient by a Running Average of Its Recent Magnitude*. COURSERA: Neural Networks for Machine Learning **4**(2), 26 (2012). 97

[173] H. Zang and X. Wan. *Towards Automatic Generation of Product Reviews from Aspect-Sentiment Scores*. In *Proceedings of the 10th International Conference on Natural Language Generation*, pp. 168–177 (Association for Computational Linguistics, Santiago de Compostela, Spain, 2017). URL https://www.aclweb.org/anthology/W17-3526. 106

[174] H. Gong, S. Bhat, L. Wu, J. Xiong, and W.-m. Hwu. *Reinforcement Learning Based Text Style Transfer without Parallel Training Corpus*. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3168–3180 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019). URL https://www.aclweb.org/anthology/N19-1320. 106

[175] H. Chen, H. Zhang, P.-Y. Chen, J. Yi, and C.-J. Hsieh. *Attacking Visual Language Grounding with Adversarial Examples: A Case Study on Neural Image Captioning*. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 2587–2597 (2018). 128, 136

[176] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. *Synthesizing Robust Adversarial Examples*. In *International Conference on Machine Learning*, pp. 284–293 (2018). 128

[177] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. *Robust Physical-World Attacks on Deep Learning Visual Classification*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1625–1634 (2018). 129

[178] A. Kurakin, I. J. Goodfellow, and S. Bengio. *Adversarial Examples in the Physical World*. In *Artificial Intelligence Safety and Security*, pp. 99–112 (Chapman and Hall/CRC, 2018). 128

[179] J. Redmon and A. Farhadi. *Yolo9000: better, faster, stronger*. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271 (2017). 129

[180] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. *Practical Black-Box Attacks Against Machine Learning*. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 506–519 (ACM, 2017). 129

[181] Y. Huang, A. W.-K. Kong, and K.-Y. Lam. *Attacking Object Detectors without Changing the Target Object*. In *Pacific Rim International Conference on Artificial Intelligence*, pp. 3–15 (Springer, 2019). 129

[182] X. Ling, S. Ji, J. Zou, J. Wang, C. Wu, B. Li, and T. Wang. *DEEPSEC: A Uniform Platform for Security Analysis of Deep Learning Model*. In *IEEE Symposium on Security and Privacy* (2019). 135

[183] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. *Image Quality Assessment: From Error Visibility to Structural Similarity*. IEEE Transactions on Image Processing **13**(4), 600 (2004). 135