UNIVERSITY OF LIVERPOOL

Visual Attention Mechanism in Deep Learning and Its Applications

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Shiyang Yan 201159131

November 2018

Dedication

This thesis is wholeheartedly dedicated to my beloved parents, who have been my source of inspiration and gave me strength when I thought of giving up, who continually provide their spiritual, emotional, and financial support.

To my relatives, friends, and colleagues who shared their words of advice and encouragement to finish the PhD study.

Abstract

Recently, in computer vision, a branch of machine learning, called deep learning, has attracted high attention due to its superior performance in various computer vision tasks such as image classification, object detection, semantic segmentation, action recognition and image description generation. Deep learning aims at discovering multiple levels of distributed representations, which have been validated to be discriminatively powerful in many tasks. Visual attention is an ability of the vision system to selectively focus on the salient and relevant features in a visual scene. The core objective of visual attention is to achieve the least possible amount of visual information to be processed to solve the complex high-level tasks, e.g., object recognition, which can lead the whole vision process to become effective. The visual attention is not a new topic which has been addressed in the conventional computer vision algorithms for many years. The development and deployment of visual attention in deep learning algorithms are of vital importance since the visual attention mechanism matches well with the human visual system and also shows an improving effect in many real-world applications. This thesis is on the visual attention in deep learning, starting from the recent progress in visual attention mechanism, followed by several contributions on the visual attention mechanism targeting at diverse applications in computer vision, which include the action recognition from still images, action recognition from videos and image description generation.

Firstly, the soft attention mechanism, which was initially proposed to combine with Recurrent Neural Networks (RNNs), especially the Long Short-term Memories (LSTMs), was applied in image description generation. In this thesis, instead, as one contribution to the visual attention mechanism, the soft attention mechanism is proposed to directly plug into the convolutional neural networks for the task of action recognition from still images. Specifically, a multi-branch attention network is proposed to capture the object that the human is intereating with and the scene in which the action is performing. The soft attention mechanism applying in this task plays a significant role in capturing multi-type contextual information during recognition. Also, the proposed model can be applied in two experimental settings: with and without the bounding box of the person. The experimental results show that the proposed networks achieved state-of-the-art performance on several benchmark datasets.

For the action recognition from videos, our contribution is twofold: firstly, the hard attention mechanism, which selects a single part of features during recognition, is essentially a discrete unit in a neural network. This hard attention mechanism shows superior capacity in discriminating the critical information/features for the task of action recognition from videos, but is often with high variance during training, as it employs the REINFORCE algorithm as its gradient estimator. Hence, this brought another critical research question, i.e., the gradient estimation of the discrete unit in a neural network. In this thesis, a Gumbel-softmax gradient estimator is applied to achieve this goal, with much lower variance and more stable training. Secondly, to learn a hierarchical and multi-scale structure for the multi-layer RNN model, we embed discrete gates to control the information between each layer of the RNNs. To make the model differentiable, instead of using the REINFORCE-like algorithm, we propose to use Gumbel-sigmoid to estimate the gradient of these discrete gates.

For the task of image captioning, there are two main contributions in this thesis: primarily, the visual attention mechanism can not only be used to reason on the global image features but also plays a vital role in the selection of relevant features from the fine-grained objects appear in the image. To form a more comprehensive image representation, as a contribution to the encoder network for image captioning, a new hierarchical attention network is proposed to fuse the global image and local object features through the construction of a hierarchical attention structure, to better the visual representation for the image captioning. Secondly, to solve an inherent problem called exposure-biased issue of the RNN-based language decoder commonly used in image captioning, instead of only relying on the supervised training scheme, an adversarial training-based policy gradient optimisation algorithm is proposed to train the networks for image captioning, with improved results on the evaluation metrics.

In conclusion, comprehensive research has been carried out for the visual attention mechanism in deep learning and its applications, which include action recognition and image description generation. Related research topics have also been discussed, for example, the gradient estimation of the discrete units and the solution to the exposure-biased issue in the RNN-based language decoder. For the action recognition and image captioning, this thesis presents several contributions which proved to be effective in improving existing methods.

Acknowledgements

I would like to express my sincere gratitude to Dr. Bailing Zhang, my primary supervisor, who provided me an opportunity for a research study and constantly guides me in the area of machine learning and computer vision. I want to express my deep thanks to Dr. Wenjin Lu and Prof. Jeremy S. Smith, my co-supervisors, for their valuable help and suggestions for my PhD study. I want to express my gratitude to Prof. Jeremy S. Smith, for the valuable help provided for the published papers, also, for the guidance and help when I was in Liverpool. It has been a valuable journey of the PhD studies in which I have learnt not only the research methodologies but also great perseverance during the research process.

I want to express my thanks to my advisors, Dr. Andrew Abel and Dr. Waleed AI-Nuaimy, who helped to evaluate my PhD studies and provide valuable suggestions for the research process.

I also want to thank all the co-authors of the published research, who offered advises, helps, and comments to my research, which are of great help during the PhD study.

I want to extend the thanks to the lab-mates, for their help and suggestions, especially, Chao Yan, Rongqiang Qian, Yizhang Xia, Fangyu Wu and Muhammad Samer.

I want to thank many friends of the Computer Science Department.

I am also thankful for the help from my old friends, former colleagues and tutors.

Finally, I offer my gratefully thanks to my family for their encouragement and support all the time.

Contents

List of Figures

List of Tables

List of Acronyms

AP Average Precision.

ATT Attention Models.

BoVW Bag of Visual Words.

CHAM Convolutional Hierarchical Attention Model.

CNNs Convolutional Neural Networks.

COCO Common Objects in Context.

Conv-Attention Convolutional Attention Model.

DCC Deep Compositional Captioning.

DNN Deep Neural Network.

DPM Deformable Part Model.

FC-Attention Fully Connected Attention Model.

GANs Generative adversarial networks.

GRU Gated Recurrent Unit.

HAN Hierarchical Attention Networks.

HC Histogram-based Contrast.

HICO Humans Interacting with Common Objects.

HM-AN Hierarchical Multi-scale Attention Networks.

HM-RNN Hierarchical Multi-scale Recurrent Neural Network.

ILSVRC-2012 ImageNet Large Scale Visual Recognition Challenge-2012.

LSTMs Long Short-term Memories.

MaliGAN maximum-likelihood augmented discrete GAN.

mAP mean Average Precision.

MLE Maximum Likelihood Estimation.

MLP Multi-layer Perceptron.

NLP Natural Language Processing.

RC Region-based Contrast.

ReLU Rectified Linear Unit.

RL Reinforcement Learning.

RNNs Recurrent Neural Networks.

RoI pooling Region-of-Interest Pooling.

SGD Stochatic Gradient Descend.

SPP Spatial Pyramid Pooling.

VQA visual question answering.

Chapter 1

Introduction

1.1 Overview

Machine learning has powered many aspects of modern society: from conventional industry to current internet business like web search engine, social networks, and content filtering. It is continuing to increase its impact on modern life. To name a few, the functionalities of machine learning include recognising objects in images, translating one language to another, match news items, recommending news based on user's interests and of course, select the relevant results in search engine.

Recently, one of the branches of machine learning family called deep learning has shown dominant performance in tasks mentioned previously and becomes increasingly important in machine learning and artificial intelligence. Conventional machine learning techniques were limited in their ability to process natural data in their raw form. For decades, constructing pattern recognition system required careful engineering and considerable domain expertise to design a feature extractor that transformed the raw data into a suitable internal representation or feature vector from which the learning subsystem, often a classifier or predictor, could classify or predict patterns in the input. These hand-crafted features, if not appropriately designed, could severely deteriorate the system performance. On the other hand, representation learning, which deep learning belongs to [2], is a set of learning methods which can be fed with only raw data and automatically discover the internal representation of the data during the process of learning.

[5] has given an empirical view of what the representation learning means, taking the example of one of the most popular models in deep learning called Convolutional Neural Networks (CNNs) [6]. In [5], the authors visualize each of the layers in the trained CNN to find what each layer represents. Interestingly, an image, for example, comes in the form of an array of raw pixels, and the learned features in the first layer of representation usually represent the presence of edges at particular orientations and locations in the image. The second layer typically detects motifs by spotting specific arrangements of edges, regardless of small variations in the edge positions. The third layer may assemble motifs into larger combinations that correspond to parts of familiar objects, and subsequent layers would detect objects as combinations of these parts. The representation of the CNN becomes more abstract in the higher layers than the lower layers. The CNN is only an example of representation learning. Not just the CNNs model but also other models like RNNs show excellent performance in various machine learning tasks. The RNNs are especially good in the sequence-to-sequence problem, which is very common in many real-world applications. For instance, machine translation, image captioning, action recognition and some problems associated with video or language are all sequence-based recognition tasks. The RNNs try to model the sequence evolution of the features by using recurrent connections, which proved to be effective in modeling the sequential dependencies.

Meanwhile, it has been found in the literature that humans do not focus their attention on an entire scene at first glance [7]. Instead, they retrieve parts of the scene or objects sequentially to find the relevant information. The visual attention mechanism had long been the research topic in neural science, computer vision, and machine learning. Most of the conventional computer vision algorithms applied visual attention mechanism only based on the low-level raw features to find the saliency. With the rapid development in representation learning and deep learning, more research learns the internal representations automatically during the training process. This technology also empowers the visual attention model to automatically retrieve relevant information for the specific task instead of solely relying on static low-level image features. Attention-based models have been shown to achieve promising results on several challenging tasks such as neural machine translation [8], image captioning [4] and action recognition [9].

The visual attention models are mainly categorised into bottom-up models and top-down models [10]. The bottom-up attention models are mainly driven by the low-level features of the visual scene. The goal of bottom-up attention is to find the salient points, which stands out from its surrounding and attracts our attention at first glance. Most of the traditional bottom-up attention models rely on hand-crafted low-level image features such as colour and intensity to produce saliency map. Most of the recently applied and effective visual attention mechanism in deep learning field belongs to the family of the top-down attention. The top-down attention is learnt in the training process and mainly driven by the discriminative training. It tries to learn the crucial features which are useful for the task at hand. This basic idea of grasping the crucial features introduces the main research topic of this thesis, which drives us to research into the mechanism of visual attention and its application in many real-world tasks.

In this thesis, following the basic idea of the visual attention, we employed, extended and improved the current visual attention mechanism in several computer vision tasks, which include the action recognition from still images, the action recognition from videos and the image description generation. The action recognition from still images is a human-related image recognition problem, the action recognition from videos are video-based recognition task, and the image description generation is an image understanding problem. In this thesis, the three important applications of computer vision can be realised with the aid of the visual attention mechanism to improve the final performance in challenging dataset. For action recognition from still images, the contextual information associated with the human is what the attention mechanism try to capture; for action recognition from videos, the crucial information in the spatial and temporal domain is what the attention mechanism focuses; for image description generation, the attention mechanism is to align the corresponding object features with the generated word automatically, which is of vital importance this kind of language-related problem.

Corresponding to the three topics, we carried out three pieces of research, all based on the visual attention mechanism:

- Many of the previously visual attention mechanism is associated with the RNNs, which tries to allocate attention region by considering the temporal dependencies. In our research [11], the visual attention mechanism is shown to be powerful in feedforward networks. By applying the multi-branch attention networks in the CNN model, action recognition from still images can be successfully realised.

- For action recognition from videos, the visual attention mechanism is extended to convolutional LSTM model in another research [12] to capture the spatial information of the CNN model, to better recognise action categories in videos. The hard attention mechanism is normally achieved by using reinforcement learning-based gradient estimator in previous research, in this thesis, instead, a novel hard attention mechanism, using Gumbel-softmax as the gradient estimator and applying in action recognition from videos is proposed in the third research [13]. This kind of gradient estimator is also used to form the discrete gates in learning hierarchical and multi-scale RNN structure, which can reason on the temporal structure of the input sequence of video frames.

- Finally, for image description generation, a hierarchical attention mechanism, which can reason on the global image features and also the fine-grained local object features, is proposed to form an advanced visual representation of the image. In this research, another contribution is using a reinforcement learning-based adversarial training algorithm to optimise the image description generator, to alleviate the exposure biased problem in the RNN-based language model.

1.2 Motivations and Challenges

1.2.1 Motivations

When human process a visual scene, they do not acquire the information of the entire scene at once and also do not treat the whole scene in equal, instead, they process the scene from the most salient or important objects first, and then adjust the attention to the relevant cues for the recognition task at hand. Take an easy example for illustration, given a scene of a dish full of potatoes and some beef, human might first recognise that the scene is about food since there is a dish which is the most important object in the scene; at a second glance, human might find the potatoes, and subsequently, they find the beef. At last, they acquire enough information to recognise the visual scene. To conclude, unlike the common practice in the machine in which the image is stored in numbers in equal importance, human pay attention differently to objects step by step to recognise a visual scene.

Inspired by this phenomenon, our main motivations for this thesis are to discover and research the implementation of the visual attention mechanism in artificial intelligence and computer vision, and subsequently test the feasibility of the proposed attention mechanism for diverse and challenging real-world applications in computer vision, which include the action recognition from still image, the action recognition from video and the automatic generation of image description.

These three tasks are all challenging and are often associated with medium or high level semantics of visual recognition. Action categories are mid-level semantics which can be used to bridge the low-level image features and high-level visual understanding. To understand an image or a video, the action of the target person is with high importance and also one of the most difficult part of the recognition. Image description generation is closely related with image understanding as natural language description can be considered as a human's way of understanding of the visual scene. Hence, another motivation of this thesis is to research into the internal mechanism of a computer vision system in discovering the semantics of a visual representation, and finally, trying to bridge the gap between the visual world and semantic world.

1.2.2 Challenges

- Action recognition is a challenging task if only the still image is provided. The action category of a person is often associated with the temporal behavior, which, is missed in a still image. Hence, a recognition model has to fully consider the contextual information for the action recognition from a still image. The challenge for this task is how to employ the visual attention mechanism to discover the contextual information in an image.

- As discussed previously, action recognition from the video is often associated with the temporal behavior of the person. However, the temporal evolution of the visual scene is essentially dynamic, which is much different from the image-based recognition. The challenge lies in the modeling of these dynamics and also the application of the attention mechanism in raising the performance of the temporal recognition problem.

- The automatic generation of image descriptions in natural language is a difficult task since it is a cross-discipline problem combined with both computer vision and Natural Language Processing (NLP). The image description in natural language can be considered as a high structural output. The modeling of the structural information with the visual attention mechanism is a challenging task, which tries to achieve a high-level machine intelligence since many images cannot be easily described even by a human.

1.3 Thesis Contributions

1. A comprehensive analysis of the visual attention mechanism in computer vision is presented in this thesis. Two important applications of computer vision and deep learning called action recognition and image description generation are discussed. Especially, in both of the two applications, the system performance is improved by using the visual attention mechanism.

2. For the task of action recognition in still images, a multi-branch attention network is proposed to capture the contextual information to improve the discriminating capability of the model for the task. It is worthy to mention that this is one of the early attempts to implement a visual attention mechanism in a CNN model. (Chapter 3)

3. For the task of action recognition in videos, two types of visual attention mechanism, including the soft attention and hard attention, are proposed and combined with a novel hierarchical multi-scale RNN model. The final performance validates that the hierarchical multi-scale RNN can capture the long-term dependency and the attention mechanism demonstrates a powerful modeling capacity in grasping the key information. (Chapter 4)

4. A novel hierarchical attention mechanism and a policy gradient optimisation technique blending with the adversarial training framework, are proposed for the task of image captioning. The hierarchical attention mechanism can reason on both the global image features and local object features while the policy gradient optimisation can compensate the exposure bias problem in the RNN-based language model. The novel architecture demonstrates good system performance. (Chapter 5)

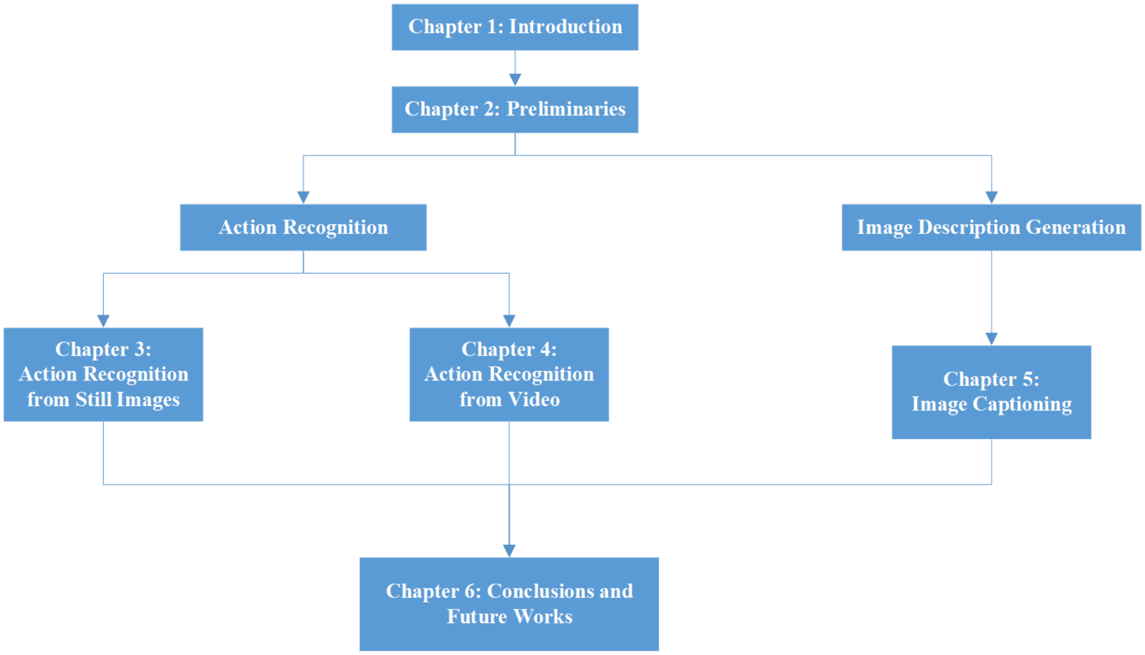

Chapter 1: Introduction
Chapter 2: Preliminaries
Action Recognition
Image Description Generation
Chapter 3: Action Recognition from Still Images
Chapter 4: Action Recognition from Video
Chapter 5: Image Captioning
Chapter 6: Conclusions and Future Works

Figure 1.1: The structure of this thesis.

1.4 Thesis Structure

A diagram of the thesis structure is shown in Fig. 1.1. A general introduction of the topic and background is provided in the Introduction, followed by the preliminaries of deep learning and the visual attention mechanism applied in this thesis. Subsequently, two applications, namely, action recognition and image description generation, are introduced with different types of visual attention mechanism. Specifically, the topic of action recognition includes action recognition from still images and from videos which are all powered by the application of the proposed visual attention mechanism. The task of image description generation is also carried out into research with visual attention mechanism. Lastly, a conclusion of this thesis and future works are introduced in the last chapter.

1.5 Publications

1.5.1 Periodical Papers

1. **Shiyang Yan**, Fangyu Wu, Jeremy S Smith, Wenjin Lu, Bailing Zhang: Image Captioning Based on Hierarchical Attention Mechanism and Policy Gradient Optimization. IEEE Transactions on Multimedia. (Under Review)

2. **Shiyang Yan**, Jeremy S Smith, Wenjin Lu, Bailing Zhang: Abnormal Event Detection from Videos using Two Stream Recurrent Variational Autoencoder. IEEE Transactions on Cognitive and Developmental Systems. (Under Revision)

3. **Shiyang Yan**, Jeremy S Smith, Bailing Zhang: Action Recognition from Still Images Based on Deep VLAD Spatial Pyramids. Signal Processing Image Communication, 54 (2017): 118-129.

4. **Shiyang Yan**, Jeremy S Smith, Yizhang Xia, Wenjin Lu, Bailing Zhang: Multi-Scale Convolutional Neural Networks for Hand Detection. Applied Computational Intelligence and Soft Computing, 2017 (2017).

5. **Shiyang Yan**, Jeremy S Smith, Wenjin Lu, Bailing Zhang: Multi-branch Attention Networks for Action Recognition in Still Images. IEEE Transactions on Cognitive and Developmental Systems, 2017 (2017).

6. **Shiyang Yan**, Jeremy S. Smith, Wenjin Lu, Bailing Zhang: Hierarchical Mult-scale Attention Networks for Action Recognition. Signal Processing Image Communication, 61 (2018): 73-84.

1.5.2 Conference Papers

1. **Shiyang Yan**, Fangyu Wu, Jeremy S. Smith, Wenjin Lu, Bailing Zhang: Image Captioning using Adversarial Networks and Reinforcement Learning. 2018 International Conference on Pattern Recognition, (ICPR 2018), Beijing, 2018.

2. **Shiyang Yan**, Jeremy S. Smith, Wenjin Lu, Bailing Zhang: CHAM: action recognition using convolutional hierarchical attention model. 2017 International Conference on Image Processing. 2017 IEEE International Conference on Image Processing (ICIP 2017), Beijing, 2017, pp. 3958-3962.

3. **Shiyang Yan**, Jeremy S. Smith, Bailing Zhang: Attributes and Action Recognition Based on Convolutional Neural Networks and Spatial Pyramid VLAD Encoding. Asian Conference on Computer Vision (ACCV 2016), Taipei, 2016, pp. 500-514. Springer, Cham.

4. **Shiyang Yan**, Yuxuan Teng, Jeremy S. Smith, Bailing Zhang: Driver behavior recognition based on deep convolutional neural networks. Natural Computation, 2016 12th International Conference on Fuzzy Systems and Knowledge Discovery (ICNC-FSKD 2016), Changsha, 2016, pp. 636-641.

5. **Shiyang Yan**, Yudi An, Jeremy S. Smith, Bailing Zhang: Action detection in office scene based on deep convolutional neural networks. 2016 International Conference on Machine Learning and Cybernetics (ICMLC 2016), Jeju Island, 2016, vol. 1, pp. 233-238.

Chapter 2

Preliminaries of Deep Learning and Visual Attention Mechanism

2.1 Preliminaries of Deep Learning

Deep learning algorithms are subsets of the machine learning algorithms, which aim at discovering multiple levels of distributed representations. Recently, various deep learning algorithms have been proposed to solve traditional artificial intelligence problems. This chapter aims to introduce the preliminaries of deep learning algorithms which are related to the research topic of the thesis, followed by the introduction and review of the visual attention mechanism.

2.1.1 Logistic Regression

Logistic Regression is a classical learning algorithm and also a fundamental part of the neural network model [14]. Logistic Regression introduces the Logistic function into the Linear Regression model.

The distribution function of the Logistic distribution is defined as:

$$P(x;\mu,s) = \frac{1}{e^{(-(x-\mu)/s)}} \tag{2.1}$$

The Logistic Regression model implements the following conditional probabilistic distribution:

$$\begin{aligned} P(y=1|x) &= \frac{e^{(w\cdot x+b)}}{1+e^{(w\cdot x+b)}} \\ P(y=0|x) &= \frac{1}{1+e^{(w\cdot x+b)}} \end{aligned} \tag{2.2}$$

where x is the input, y is the output, w and b are the parameters, w is the weight vector, b is the bias and $w \cdot x$ is the dot product of w and x.

The equation 2.2 can get the conditional probabilities of the output to be 1 and 0 given the input samples.

The odds of an event is the ratio of the probability of happening of this event to the probability of not happening. If the probability of an event happening is P, then the odds of this event is $\frac{p}{1-p}$, also, the log odds of the event is $logit(P) = \log \frac{p}{1-p}$.

For the Logistic Regression model, the log odds of the event is hence $\log \frac{P(y=1|x)}{1-P(y=1|x)} = w \cdot x$, which indicates that the log odds of the Logistic Regression is a linear function of the input x.

2.1.2 Basic Neural Network Model

From the viewpoint of a neural network, the Logistic Regression can be interpreted as one layer neural network, with a Sigmoid (Logistic) activation function as the non-linear mapping function, which is shown in Fig. 2.1.

If the multi-layer mapping is embedded in this system, it can form a neural network learning model, or more specific, a feedforward neural network [15] [16]. The feedforward networks, or Multi-layer Perceptron (MLP), are vital in deep learning models. A feedfor-

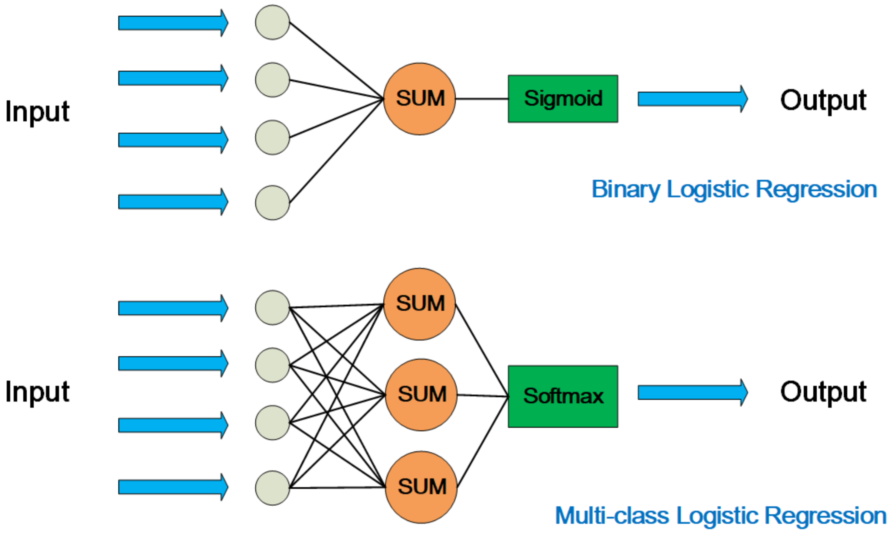


Figure 2.1: The neural network interpretation of Logistic Regression.

ward network aims to approximate some non-linear functions. The feedforward networks are of extreme importance to machine learning practitioners.

They form the basis of many critical applications. For example, the convolutional network used for object recognition from images is a specific kind of feedforward network. Feedforward networks are a conceptual stepping stone on the path to recurrent networks, which power many natural language applications. Feedforward neural networks are called networks because they are typically represented by composing together many different functions. The model is a directed acyclic graph which describes how the functions are composed together.

A general structure of the feedforward network is shown in Fig. 2.2. The neural network has several layers in which each of them performs a matrix operation and a non-linear mapping. The neural network model is proved to be universal approximators [17], which can approximate any measurable functions to any desired degree of accuracy. Also, there are no theoretical constraints for the success of feedforward networks.

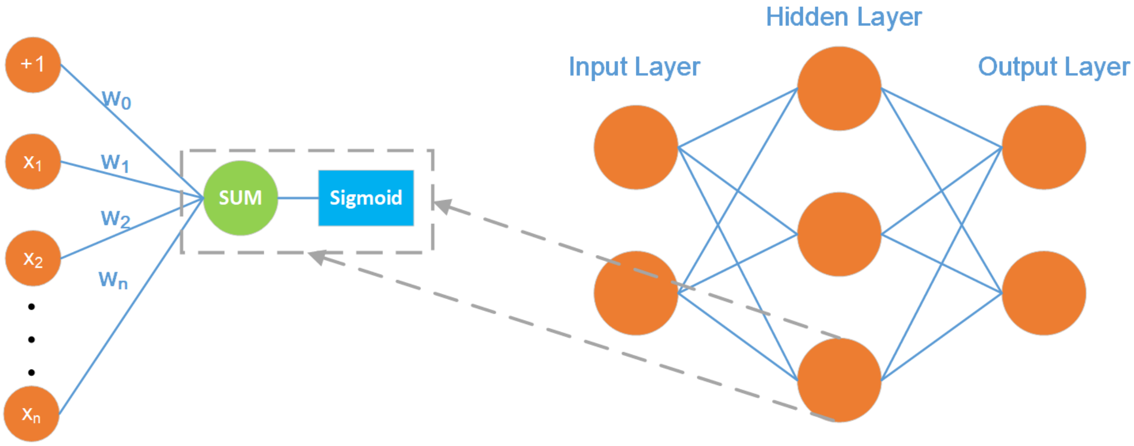


Figure 2.2: The structure of a feed forward neural network.

2.1.3 Convolutional Neural Networks

CNNs [6] are a particular type of neural network for processing data that has a known grid-like topology. Examples include natural language or speech data, which can be considered of as a 1D grid taking samples at regular intervals, and visual data, which can be considered of as a 3D grid of pixels. Convolutional networks have been very successful in many real-world applications. The name 'convolutional neural network' indicates that the network employs a mathematical operation called 'convolution'. Convolution is a special kind of linear operation. Convolutional neural networks are neural networks that use convolution in place of general matrices multiplication in layers of them.

A typical CNN for hand-written digits recognition is shown in Fig. 2.3. In addition to the convolution operation in the CNN, another layer called 'Pooling (Subsampling)' is also an important operation in the CNN, which will be discussed later.

Convolution Operation

The convolution operation on a continuous function is defined in Equation 2.3, it can be interpreted as using a kernel function $w(a)$, to calculate a weighted average of function

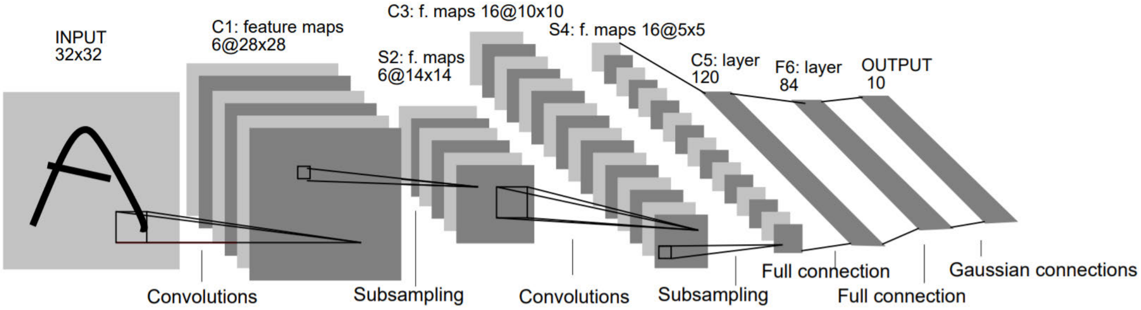


Figure 2.3: A typical CNN for hand-written digits recognition [1].

$x(a)$ and $w(a)$.

$$s(t) = \int x(a)w(t-a)da \tag{2.3}$$

From the Latin 'convolvere', 'to convolve' means to roll together. For mathematical purposes, convolution is the integral measuring how much two functions overlap as one passes over the other. Think of convolution as a way of mixing two functions by multiplying them.

The convolutional operation can also be defined as asterisk, in Equation 2.4.

$$s(t) = (x * w)(t) \tag{2.4}$$

In the case of CNN terminology, the first argument, i.e., the function x, is the input and the second argument, i.e., the function w, is referred to as the kernel. The output is often referred to as feature map.

In most cases, the data used are sampled not in every instant, but at a certain interval, in other words, these data are discredited. The time index t, consequently, then takes on only integer values. The discrete convolution is defined in Equation 2.5.

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{2.5}$$

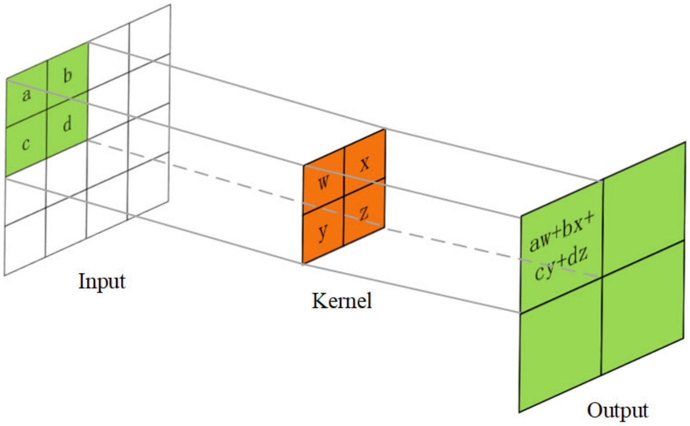


Figure 2.4: An illustration of the convolutional operation in a CNN.

In machine learning applications, the input is usually a multidimensional array of data, and the kernel is usually a multidimensional array of parameters that are adopted by the learning algorithm. These multidimensional arrays will be referred as tensors.

In practice, the infinite summation can be implemented as a summation over a finite number of array elements, if the tensors are considered as zero everywhere except where the data is stored in the multidimensional arrays.

Also, convolutions can be used over more than one axis at a time. For instance, if a two-dimensional image I is taken as our input, a two-dimensional kernel K is utilised. Then, the two-dimensional convolution can be defined in Equation 2.7.

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I(m,n)K(i-m,j-n) \tag{2.6}$$

Since the convolution operation is commutative, alternatively, Equation 2.7 can also be written as:

$$S(i,j) = (K * I)(i,j) = \sum_{m}\sum_{n} I(i-m,j-n)K(m,n) \tag{2.7}$$

A commonly used effective operation process of convolution in a CNN is described by

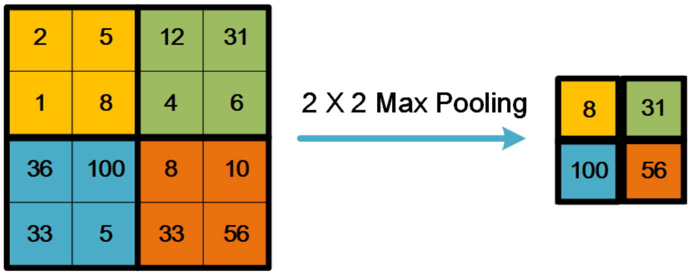


Figure 2.5: An illustration of the max pooling operation in a CNN.

Fig. 2.3.

Pooling Operation

A typical layer of a CNN consists of three steps. In the first step, the layer performs several convolutions in parallel to produce a set of linear activations. In the second step, each linear activation is run through a nonlinear activation function, such as the Rectified Linear Unit (ReLU) function [18]. This step is sometimes called the detector stage. In the third stage, a pooling function to used to modify the output of the layer.

This section aims to give a general introduction to pooling. A pooling operation replaces the output of the neural network at a specific location with a summary statistic of the nearby outputs. The most commonly used pooling in CNN is max-pooling. Pooling helps to make the representation approximately invariant to small translations of the input. Invariance to the translation means that if the input is translated by a small amount, the values of most of the pooled outputs do not change, which increases the robustness of the neural recognition network. The application of pooling can be seen as adding an infinitely strong prior that the function that the layer learns must be invariant to small translations. When this assumption is correct, it can significantly improve the statistical efficiency of the network.

The max-pooling operation is shown in Fig. 2.5. In each colour-indicated grid, the max-pooling selects the maximum value to replace the data in the original grid, and form a new tensor as an output of the max-pooling layer.

Stochastic Pooling A disadvantage of max-pooling is that it is sensitive to overfit the training set, making it hard to generalize well during testing [19]. To solve this problem, Zeiler et al. [20] proposed a stochastic pooling approach which replaces the conventional deterministic pooling operations with a stochastic procedure, by randomly picking the activation within each pooling region according to a multinomial distribution. This stochastic nature is helpful to prevent the overfitting problem.

Spatial Pyramid Pooling (SPP) and Region-of-Interest Pooling (RoI pooling) The CNN model requires a fixed-sized input image. This restriction may bring problems for images of arbitrary sizes, especially in the CNN-based object detection schemes. To eliminate this limitation, He et al. [21] replaced the last pooling layer with a SPP, for object recognition. The SPP can extract fixed-length features from arbitrary images (or region candidates), and can be applied in a CNN structure for arbitrary tasks, to improve the performance of the CNN model.

Subsequently, Girshick [22] proposed a simplified SPP layer for object recognition, called RoI pooling. This pooling layer is simpler and also enables the CNN model to handle arbitrary-sized input images. More importantly, the RoI Pooling layer enables the parameter sharing in the computation-intensive convolutional layers [22]. This research is extremely important in object detection. Most subsequent research [23] [24] [25], for various tasks, employed the RoI pooling layer to deal with input images.

Spatial Transformers Due to the typically small spatial support for max-pooling, the spatial invariance is only realised over a deep hierarchy of max-pooling and convolutions, and the intermediate features in a CNN model are not actually invariant to large transformations of the input data [26] [27]. To mitigate this issue, Jaderberg et al. [28] proposed an important model, the spatial transformer networks, which explicitly allows the spatial transformation of data in the network. The spatial transformers result in arbitrary CNN models which learn invariance to translation, scale, rotation and more generic warping. Also, the spatial transformer can be interpreted as an attention mechanism, but is more flexible and can be trained purely with backpropagation without reinforcement learning techniques.

Capsule Networks Geoffrey Hinton pointed out many drawbacks of the max-pooling operation such as the side effect of 'coarse coding' [29]. To address this issue, Sabour et al. [30] proposed the 'Capsule Networks' in which a dynamic routing scheme is proposed between the capsules to replace the max-pooling. This type of 'routing-by-agreement' is more effective than the primitive form of routing in max-pooling, which allows neurons in one layer to ignore all but the most active feature detector in a local pool. This research is considered as an recent breakthrough in the deep learning area [31].

Activation Function

Activation in a neural network provides non-linear mappings that take the inputs and do some mathematical operations on them. Many such activation functions exist and are discussed as follows:

Sigmoid (Logistic) This non-linearity takes an input a real-valued function and outputs value in the range of 0 and 1. It is similar to the Logistic Regression and has been widely applied in neural networks for a long time. However, it suffers from saturating and vanishing gradient problem. The Equation 4.18 defines the Sigmoid function.

$$Sigmoid(x) = \frac{1}{1 + e^{x}} \tag{2.8}$$

Tanh As shown in Equation 2.9, it is clear that Tanh can be considered as a scaled up version of a sigmoid, outputting values in the range of -1 and 1. The problem of saturating gradients also exists with this function. The Tanh function is widely applied in Recurrent Neural Networks (RNNs).

$$Tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = 2Sigmoid(2x) - 1 \tag{2.9}$$

ReLU ReLU is a linear activation function which has a threshold at zero as shown in Equation 2.10. The convergence of gradient descent has been proved to be accelerated by applying ReLU [18].

$$ReLU(x) = max(0, x) \tag{2.10}$$

Training Strategies

The deep architecture and large number of parameters in a CNN model bring high performance and also another problem, the overfitting problem during training. Several regularization techniques that try to compensate this issue are introduced in this section.

Drop-out Drop-out is proposed by Geoffrey Hinton [32] and discussed in detail in [33]. During each training batch, the algorithm will randomly drop certain amount of the feature detectors to enhance the generalization ability of the model, and prevent overfitting. Subsequently, Warde-Farley [34] analysed the feasibility of the drop-outs and pointed out that drop-out is an effective ensemble learning method.

Pre-training and Fine-tuning One of the purposes of pre-training for deep learning practitioners is preventing overfitting. It is associated with data augmentation and transfer-learning. Pre-training means initialising the CNN model with a set of pre-trained parameters rather than randomly-initialised ones. Also, the deep neural networks are highly non-linear function. The backpropagation algorithm might lead the neural networks to local minima. Pre-training can provide a good start point for the initialisation of the parameters of the deep neural networks. It is a very popular practice in deep learning area, due to the advantages that it can accelerate the learning process and improve the generalisation capability. Erhan et al. [35] conducted an extensive research on why the pre-training steps help in raising the system performance. Deep learning researchers employ well-known CNN architecture pre-trained on ImageNet [36] dataset and fine-tune the model for the task at hand.

Common CNN Architectures

In this section, some of the commonly used CNN architectures in computer vision are presented.

LeNet This CNN architecture was one of the pioneering research in CNNs by LeCun et al. [37]. In this research, the hand-written digits were recognised by a CNN. It finds application in reading zip codes, digits, and so on. The lack of high-level computing machines at that time restricted the large-scale application of CNNs.

AlexNet This architecture developed by Alex Krizhevsky, Ilya Sutskever and Geoff Hinton [18] is credited as the first work in CNNs to popularise it in the field of computer vision. The network was similar to LeNet, but instead of alternating convolution layers and pooling layers, AlexNet had all the convolutional layers stacked together. Also, they proved the feasibility of ReLU function in training large-scale CNN. Moreover, compared to LeNet, this network is much bigger and deeper. AlexNet was able to win the ImageNet Large Scale Visual Recognition Challenge-2012 (ILSVRC-2012) [36]) competitions achieving top-1 and top-5 error rates on test dataset.

GoogleNet This CNN architecture from Szegedy et al. [38] from Google won the ILSVRC 2014 competition. They proposed a new architecture called Inception (v1) that gives more utilisation of the computing resources in the network. GoogleNet is a particular incarnation that has twenty-two layers of Inception module but with less parameter compared to AlexNet. Later, many improvements had been made on Inception-v1, with the principle being the introduction of batch normalisation which led to Inception-v2 by Ioffe et al. [39]. More refinements were added to this version, and the architecture was referred to as Inception-v3 [40]. Also, the Inception network is continuing to be developed [41].

VGG-Net A famous structure, developed by Karen Simonyan and Andrew Zisserman [42], called VGG-Net, has been adopted by many types of research for various computer vision tasks. The authors of [42] have done a thorough analysis of the depth factor in a CNN, keeping all other parameters fixed. This trial could have led to a vast number of parameters in the network, but it was efficiently controlled by using tiny 3x3 convolution filters in all layers. The VGG-Net was the runner-up in ILSVRC 2014 contest.

Residual-Net A severe problem, preventing the CNN to be deeper, is the vanishing gradient problem [43]. He et al. developed a CNN framework by utilising a residual connection between layers, can reduce the vanishing gradient effect on the training of a very deep network [43]. A primary drawback of this framework is that it is much expensive to evaluate due to the significant number of parameters. However, the number of parameters can be reduced to an extent by removing the first Fully-Connected layer (most of the parameters are in this layer in a CNN), without any effect on the final performance.

2.1.4 RNNs

As previously discussed, the vanilla neural networks and CNNs are feedforward neural networks, which lacks the capability of processing sequential or structural data. RNNs are a family of neural networks for processing sequential data. In contrast to the CNN which is specialised for processing a grid of values such as an image, an RNN is specialised for a sequence of values $x^{(1)}, x^{(2)}, ..., x^{(t)}$.

To go from multi-layer neural networks to recurrent neural networks, it is interesting to

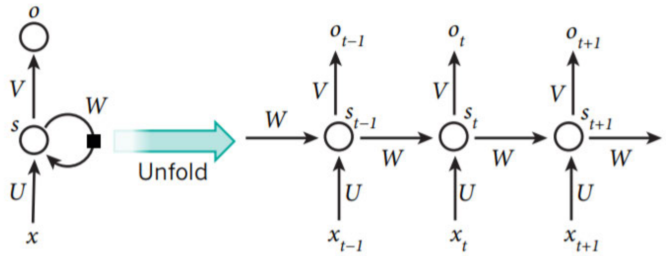


Figure 2.6: The unfolding of the computational graph of a RNN [2].

discuss one of the early ideas found in machine learning and statistical models of the 1980s: sharing parameters across different parts of a model. Parameter sharing makes it possible to extend the neural networks to examples of different forms and generalise across them. If the neural networks are not able to share parameters among different time steps of a sequential data, they cannot generalise among the data. Sharing parameters are critical if there is a relationship in data between different time steps. A traditional fully connected feedforward network would have separate parameters for each input feature, so it has to learn all of the rules of the data separately at each time step. By comparison, a RNN shares the same weights across several time steps.

To see the secrets of the RNNs, the concept of a computational graph is firstly introduced. A computational graph is a way to formalise the structure of a set of computations, such as those involved in taking inputs and parameters to outputs and final loss function. In this section, the idea of unfolding a recurrent computation into a computational graph that has a repetitive structure, typically corresponding to a chain of computational units, is firstly introduced. Unfolding this graph results in the sharing of the parameters across a neural network structure.

LSTMs

The most effective sequence models used in practical applications are called gated RNNs in which the most widely used neural networks are LSTMs. The vanilla RNNs suffer from the vanishing gradient problem [44]. Roughly speaking, the error gradients would vanish exponentially quickly with the size of the time lag between important events, which makes training very difficult. Hence, to mitigate this problem, Hochreiter and Schmidhuber [45] proposed a gated RNN called LSTMs.

An LSTM consists of an input gate (i_t), an output gate (o_t), a forget gate (f_t), cell memory (c_t) and hidden state (h_t), which follow the following updating rules:

$$\begin{aligned}
&i_t = tanh(W_{xi} * x_t + W_{hi} * h_{t-1} + b_i)\\
&f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + b_f)\\
&o_t = tanh(W_{xo} * x_t + W_{ho} * h_{t-1} + b_o)\\
&g_t = \sigma(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c)\\
&c_t = f_t \cdot c_{t-1} + i_t \cdot g_t\\
&h_t = o_t \cdot \phi(c_t)
\end{aligned} \tag{2.11}$$

where $*$ indicates the matrices multiplication, $\cdot$ indicates elementwisely multiplication, W_* means the weight matrices and b_* are the bias vectors.

As can be seen from the Equation 2.11, the RNN dynamically updates the three gates, cell memory and hidden state in every time step. The architecture of the LSTMs can be seen in Fig. 2.7.

A criticism of the LSTM model is that it is mainly ad-hoc and that it has a substantial number of components whose purpose is not apparent. Consequently, it is not clear why the LSTM is an optimal architecture, it is possible that better architectures exist [46].

Gated Recurrent Unit (GRU)

A GRU is proposed by Cho et al. [47] for the task of sequence-to-sequence modelling. Similar to the LSTM, the GRU has gating units that modulate the flow of information inside the unit, however, without having a separate memory cells.

A GRU is defined by Equation 2.12. Fig. 2.8 demonstrates the architecture of the

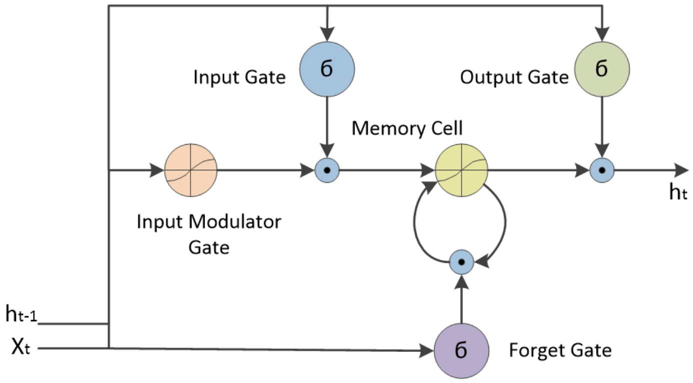


Figure 2.7: The architecture of the LSTMs.

GRU.

$$\begin{aligned} r_t &= \sigma(W_x r * x_t + W_h r * h_{t-1} + b_r) \\ z_t &= \sigma(W_x z * x_t + W_h z * h_{t-1} + b_z) \\ \tilde{h}_t &= tanh(W_x h * x_t + W_h h(r_t \cdot h_{t-1}) + b_h) \\ h_t &= (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t \end{aligned} \tag{2.12}$$

The GRU is an alternative to the LSTM which is similarly difficult to justify. [46] provided an empirical analysis on the performance of the LSTM and GRU, indicating that the GRU outperformed the LSTM on nearly all tasks except language modelling with the naive initialization. [48] however, indicated that the LSTM and GRU generates different results, depending on the tasks.

2.1.5 GANs

The Theory of the GANs

GANs are an example of the generative model. GANs was first proposed by [3] in 2014. It is proposed initially to generate realistic images given a random signal. The fundamental idea of GANs is to set up a game between two players. One of them is the generator, which

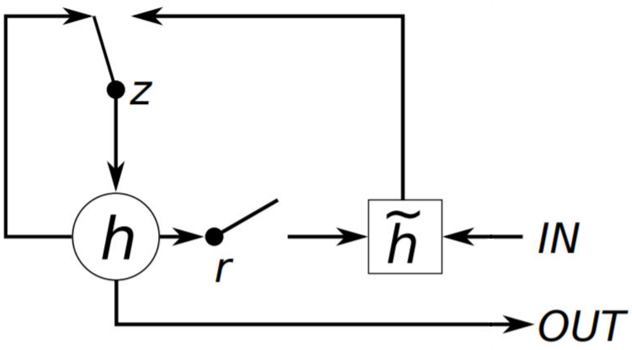


Figure 2.8: The architecture of the GRU.

creates samples that are intended to come from the same distribution as the training data. The other player is the discriminator which tries to differentiate the generated samples from real samples. The discriminator learns using traditional supervised learning algorithms, discriminating inputs into two categories (whether from generated or real samples). The generator is trained to deceive the discriminator. This is an adversarial game in which the generator tries to generate samples more like the real ones while the discriminator is trained to better discriminate between the generated and real samples. The generator must learn to make samples that are indistinguishable from the genuine samples to make the game successful, and hence, the generator network can learn to generate samples that are drawn from the same distribution as the training data.

In the original GANs, the adversarial framework applied when the models are both MLP [3]. In fact, CNNs can also be used in this framework [49], also RNNs [50]. To learn the generator's distribution p_g over data x, the GANs define a prior on input noise variables $p_z(Z)$, then represent a mapping to data space as $G(z; \theta_g)$, where G is the generator which is represented by a differentiable function such as neural networks. The discriminator is another neural network, $D(x; \theta_d)$ which outputs a single value, representing whether the samples are generated or real. Then the discriminator D is trained to maximise the probability that the correct labels are assigned to the training samples and generated

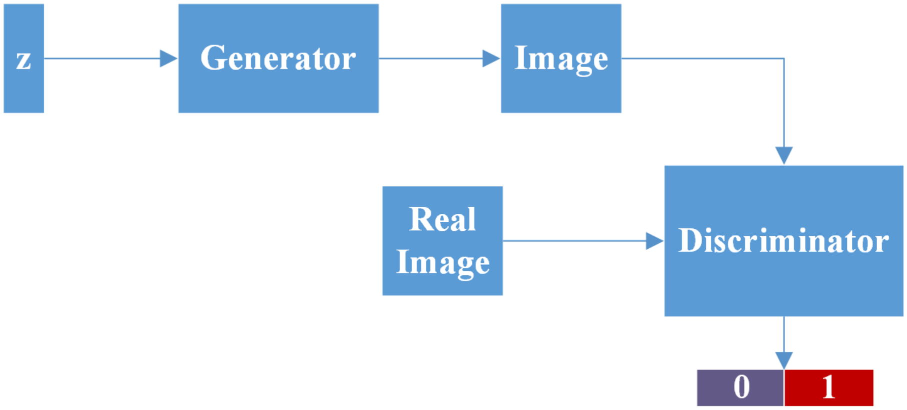


Figure 2.9: The structure of a typical GANs model [3].

samples. The generator, G, is trained simultaneously to minimize $\log(1 - D(G(z)))$. In summary, the D and the G play a two-player minimax game as described in Equation 2.13. The structure of a typical GANs model is shown in Fig. 2.9.

$$\underset{G}{min}\,\underset{D}{max}\,V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{2.13}$$

The GANs framework is not restricted in image generation, in fact, it can be applied in many tasks. For instance, language generation is an essential task in natural language processing and also has significant practical value. [50] propose the SeqGAN for language generation. As explained in Equation 2.13, the generator and discriminator are trained simultaneously, which means that the gradient can be back propagated from the discriminator to the generator, since image generation is a continuous process. However, language generation is a discontinuous, often token by token. To directly apply GANs on the task of language generation is infeasible. To tackle this difficulty, [50] propose to use reinforcement technique in which the probability of the generated samples to be real is considered as a reward value for the generator. Hence, with the aid of reinforcement learning algorithms, the SeqGAN can be trained, with improving results over conventional supervised learning.

The GANs in Discrete Settings

Since in Chapter 5, the GANs in language problem is what need to be dealt with, some of the research in this area is firstly reviewed. Despite the successes in capturing continuous distributions such as image generation, the application of the GANs to discrete settings, such as natural language processing tasks, is restricted. The reason is that the difficulty of backpropagation through discrete random variables combined with the inherent instability of the GANs training objective, prevent the conventional GANs to be feasible in natural language processing tasks.

To address this issue, [51] proposed maximum-likelihood augmented discrete GAN (MaliGAN) for discrete settings by replacing the initial GANs objective with a low-variance objective using the discriminators output that follows the log-likelihood. [52] proposed a boundary-seeking GANs by using the estimated difference measured from the discriminator to compute importance weights for generated samples, providing a policy gradient optimisation solution for training the generator. Instead of the policy gradient optimisation, [53] used a Gumbel-softmax gradient estimator to deal with the discreteness problem to train the GANs.

SeqGAN proposed in [50] is another attempt to solve this issue. The SeqGAN use three-steps training strategy by using the policy gradient algorithm to backpropagate the reward signal, obtained from the discriminator, to the generator. This training strategy was also used in [54] for visual paragraph generation, but with a Wasserstein-GANs objective [55], which proves to be more stable for the GANs training. In Chapter 5, the idea of the SeqGAN is used, but with a discriminator to evaluate the coherence and consistency between the multi-modal information, for image captioning.

2.2 Visual Attention Mechanism

When processing a complex visual scene, human beings do not tend to look at the visual scene in its entirety at once. Instead, they focus on a subset of the visual content to faster the visual analysis process. Inspired by this phenomenon, visual attention mechanism becomes a hot research topic in computer vision, neuroscience and deep learning field. It is widely used in object segmentation, object recognition, image caption generation, human action recognition and visual question answering (VQA). In the last few years, deep learning has been growing rapidly. Many CNNs and RNNs have achieved much better performance in various computer vision and natural language processing tasks, compared to previous traditional methods. Recent progress in deep learning witnessed a close relation between deep learning and visual attention mechanism. The visual attention models are mainly categorised into bottom-up models and top-down models [10].

2.2.1 Bottom-up Visual Attention

The bottom-up attention models are mainly driven by the low-level features of the visual scene. The goal of bottom-up attention is to find the salient points, which stand out from its surrounding and attracts our attention at first glance. Most of the traditional bottom-up attention models rely on hand-crafted low-level image features such as colour and intensity to produce saliency map. Histogram-based Contrast (HC) and Region-based Contrast (RC) algorithm [56] are typical bottom-up attention methods which generate saliency map by evaluating global contrast differences and spatially weighted coherence scores. The bottom-up attention model [57] was implemented with Faster R-CNN [23], while spatial regions are represented as bounding boxes, providing a significant improvement on VQA tasks.

A related research topic, saliency detection, is also driven by the low-level visual features. Most of the saliency detection methods [58] [59] [60] use low-level image features such as contrast, edge, intensity, which can be considered as fixed and bottom-up approach.

One of the drawbacks of the bottom-up attention is that it cannot automatically learn the task-specific attention features since it is purely based on the low-level visual features. Zhou et al. [61] applied global average pooling [62] to discriminate salient CNN features for the target object category. It is a kind of task-relevant approach but still not flexible enough.

2.2.2 Top-down Visual Attention

Most of the recently applied and effective visual attention mechanism in deep learning field belongs to the family of top-down attention. Different attention models have been proposed and applied in object recognition and machine translation. Mnih et al. [63] proposed an attention mechanism to represent static images, videos or as an agent that interacts with a dynamic visual environment. Also, Ba et al. [64] presented an attention-based model to recognise multiple objects in images. The two models mentioned previously are all related to RNNs and with the aid of a reinforcement learning strategy.

Bahdanau et al. [8] proposed a novel attention model for neural machine translation without the prerequisite of reinforcement learning, which can be trained end-to-end by the back propagation method. It is called a soft attention model. Later, a comprehensive study for hard attention bound with reinforcement learning and soft attention for the task of image captioning was published by Xu et al. [4]. Followed up researches include action recognition with soft attention proposed by Sharma et al. [65] and video description generation [66].

The top-down visual attention, according to the related works mentioned previously, can be further categorised into two classes: hard attention and soft attention. The soft attention model, usually ‘softly’ assign differentiated weights on different image regions or locations, can be directly trained using backpropagation algorithm; The hard attention model, however, can be considered as a discrete unit, which performs hard decision on which part of the image features to be utilised. Backpropagation algorithm can not directly train it since its discreteness. Hence, it often needs some gradient approximation tricks to make the training process working.

As an overview of the published works, soft attention models were mainly realized with the leverage of RNNs for handling sequences or time-domain information. To directly process static image, in Chapter 3, it is much desirable to implement soft attention models in the general CNN frameworks. Teh et al. [67] applied the soft attention mechanism in CNNs for weakly supervised object localisation and achieved excellent results in the PASCAL VOC detection challenge [68]. They emphasised the relative importance on candidate proposals to automatically select target regions with only region-level considered.

The typical soft attention and hard attention mechanism are first introduced, which are initially proposed by [4] for image captioning task, then our proposed CNN-based soft attention mechanism (detailed explanations and its applications can be seen in Chapter 4) and a kind of hard attention mechanism (detailed explanations and its applications can be seen in Chapter 4).

In Chapter 5, to unify the region-level and image-level attention framework, a hierarchical soft attention model is proposed by using a two-layer LSTMs network to reason on both of the image features and region candidates. This is our contribution to the current soft attention model, which can be applied in image captioning, action recognition and many other related tasks.

Soft Attention Mechanism

This soft attention mechanism is proposed in [4] for image captioning. Specifically, the model comprises an encoder and a decoder. They use a convolutional neural network pretrained on the ImageNet dataset [69] in order to extract a set of convolutional features. These features, denoted as $a = \{a_1, ..., a_L\}$, correspond to certain portions of the 2-D image.

The LSTMs network, proposed initially by Hochreiter and Schmidhuber in [45], is applied as the language decoder.

$$\begin{aligned}
i_t &= \sigma(W_{xi} * z_t + W_{hi} * h_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * z_t + W_{hf} * h_{t-1} + b_f) \\
o_t &= \sigma(W_{xo} * z_t + W_{ho} * h_{t-1} + b_o) \\
g_t &= \sigma(W_{xc} * z_t + W_{hc} * h_{t-1} + b_c) \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot g_t \\
h_t &= o_t \cdot \phi(c_t)
\end{aligned} \tag{2.14}$$

In Equation 2.14, i_t, f_t, o_t, c_t and h_t are the input gate, forget gate, output gate, cell memory and hidden state of an LSTM network, respectively. g_t and h_t are the input and the output of the LSTM model. z_t is the context vector, which can be processed by the soft attention mechanism and can capture visual information associated with a certain input location. The soft attention mechanism has to automatically allocate adaptive weights for the image locations to facilitate the task at hand.

$$e_{ti} = f_{att}(a_i, h_{t-1}) \tag{2.15}$$

where $a_i \in \{a_1, ..., a_L\}$. Equation 2.15 actually maps the image features from each location, along with information from the hidden state, into an adaptive weight, which indicates the importance of each image location for the recognition.

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{L} exp(e_{tk})} \tag{2.16}$$

Then, Equation 2.16 normalises the adaptive weights into a probability value in the range of 0 and 1 using the Softmax function. Once these weights (summed to 1) are computed, the weights vector α_t element-wisely multiplied with image feature vector a and summed up to the context vector z_t, which can be expressed as in Equation 2.17, which is the expectation of weighted features maps.

$$z_t = \sum_{i=1}^{L} \alpha_{t,i} a_i \tag{2.17}$$

Then the context vector z_t is forwarded to the LSTM network to generate captions, as described in Equation 2.14. This soft attention mechanism can adaptively select the relevant visual parts of the given image features and thus facilitate the recognition.

Hard Attention Mechanism

The hard attention was also proposed in [4]. Their hard attention was realised with the aid of a REINFORCE-like algorithm. In this section, this kind of hard attention mechanism is also introduced.

The location variable l_t indicates where the model decides to focus attention on the t^{th} step of a language inference. $l_{t,i}$ is an indicator of a one-hot representation which can be set to 1 if the i^{th} location contains a relevant feature.

Specifically, a hard attentive location of $\{\alpha_i\}$ is assigned:

$$p(l_{i,t} = 1 | l_{j<t,a}) = argmax(\alpha_{t,i}) \quad = argmax \left(\frac{exp(W_i h_{t-1})}{\sum_{j=1}^{K \times K} exp(W_j h_{t-1})} \right) \tag{2.18}$$

where a represents the input image features.

An objective function L_l can be defined that is a variational lower bound on the marginal log-likelihood log $p(y|a)$ of observing the action label y given image features a. Hence, L_l can be represented as:

$$L_l = \sum_l p(l|a) \log\ p(y|l,a) \leq \log\ \sum_l p(l|a) p(y|l,a) = \log p(y|a) \tag{2.19}$$

$$\frac{\partial L_l}{\partial W} = \sum_l p(l|a) [\frac{\partial \log\ p(y|l,a)}{\partial W} + \log\ p(y|l,a) \frac{\partial \log\ p(l|a)}{\partial W}] \tag{2.20}$$

Ideally, the gradients of Equation 2.20 is what need to be computed. However, it is not feasible to compute the gradient of expectation in Equation 2.20. Hence, a Monte Carlo approximation technique is applied to estimate the gradient of the operation of expectation.

Therefore, the derivatives of the objective function with respect to the network parameters can be expressed as:

$$\frac{\partial L_l}{\partial W} = \frac{1}{N}\sum_{n=1}^{N}[\frac{\partial \log\ p(y|\tilde{l}_n, a)}{\partial W} + \log\ p(y|\tilde{l}_n, a)\frac{\partial \log\ p(\tilde{l}_n|a)}{\partial W}] \tag{2.21}$$

where $\tilde{l}$ is obtained based on the argmax operation as in Equation 2.18.

Similar with the approaches in [4], a variance reduction technique is used. With the k^{th} mini-batch, the moving average baseline is estimated as an accumulation of the previous log-likelihoods with exponential decay:

$$b_k = 0.9 \times b_{k-1} + 0.1 \times \log\ p(y|\tilde{l}_k, a) \tag{2.22}$$

The learning rule for this hard attention mechanism is defined as follows:

$$\frac{\partial L_l}{\partial W} \approx \frac{1}{N}\sum_{n=1}^{N}[\frac{\partial \log\ p(y|\tilde{l}_n, a)}{\partial W} + \lambda(\log\ p(y|\tilde{l}_n, a) - b)\frac{\partial \log\ p(\tilde{l}_n|a)}{\partial W}] \tag{2.23}$$

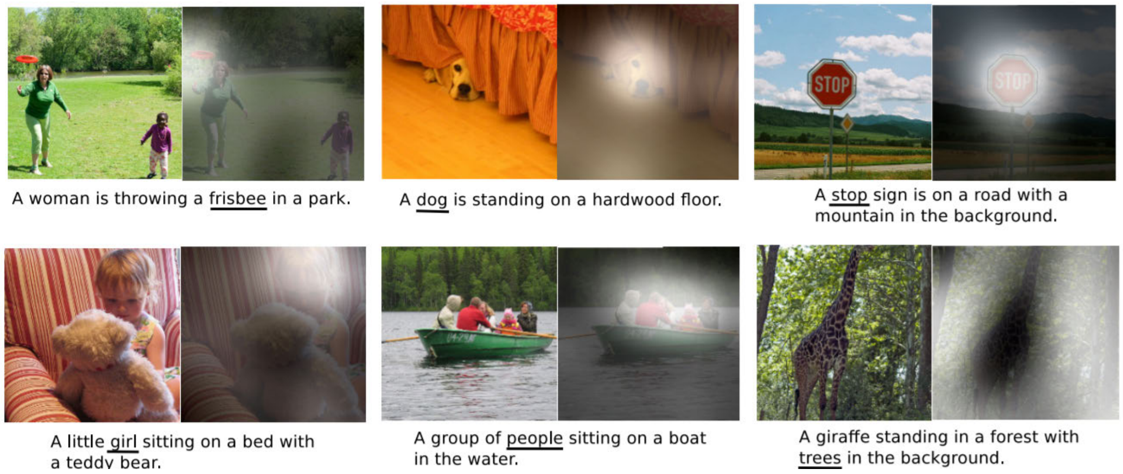
where λ is a pre-defined parameter.

As pointed out in Ba et al. [64], Mnih et al. [63] and Xu et al. [4], this is a formulation which is equivalent to the REINFORCE learning rule [70]. For convenience, it is abbreviated as REINFORCE-Hard Attention in the following sections.

2.2.3 Recent Development of the Attention Mechanism

The Development of the Attention Mechanism in Computer Vision

[63] is an early attempt to propose an attention mechanism for deep learning algorithm in computer vision. Subsequently, the attention mechanism in computer vision and the attention mechanism in natural language processing are intertwined and promote each



A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Figure 2.10: The attention mechanism in [4]

other. The image captioning model proposed in [4] can be seen as a milestone in visual attention mechanism, in which both a soft attention and a hard attention model are proposed and analysed to improve the performance and efficiency in neural image captioning. Fig. 2.10 shows the attention mechanism in [4]. In this figure, the brighter area indicates the attentive regions, which normally correspond to the generated words.

In image captioning, there are many subsequent research using the visual attention mechanism. For instance, [71] proposed to incorporate semantic attention by using attributes of the image. [72] introduced novel channel-wise attention which allocates weights on each channel of the convolutional feature maps for image captioning, with improved results. [73] used an object localizer for the bottom-up attention, and the soft attention model described previously as the top-down attention model for image captioning, improving the existing results by a large margin. A hierarchical attention mechanism is implemented for image captioning, which are described in Chapter 5.

Theoretically, the attention mechanism is not restricted to the image captioning task. For object detection, [74] train a class-specific object localization model using a reinforcement learning algorithm and utilize the model for a detection task by evaluating all the regions, which can be considered as a type of attention mechanism. [75] proposed an attention networks for object localisation. [76] used a visual attention mechanism on the region candidates to localise objects in a weakly-supervised setting. [77] incorporated global and local contexts by using attention model to the region-based CNN (Fast RCNN [22]) for object detection, with improved results. Specially, [78] proposed a residual attention networks, which can be easily generalized to hundreds of layers, for image classification.

Many other computer vision problems benefit from the visual attention mechanism. For person re-identification, the attention mechanism also demonstrates improved effect. For instance, [79] proposed an end-to-end comparative Attention Networks for person re-identification. [80] proposed a CNN-based attention model which is specially designed for the person re-identification in a triplet architecture. [81] formulated an idea of jointly learning multi-granularity attention selection and feature representation for optimising person re-identification. [82] learn context-aware sequence features and proposed a dual attention mechanism for sequence comparison.

Moreover, for action recognition, a multi-branch attention model is proposed for action recognition from still images, which are discussed in Chapter 3. [65] used a soft attention mechanism with RNN networks for action recognition from videos. [83] applied a spatial and a temporal attention-aware pooling operation for action recognition. [84] proposed a spatial-temporal attention mechanism for action recognition in skeleton data. [85] introduced an attention pooling operation for action recognition. The soft attention and the hard attention mechanism are combined with a novel hierarchical multi-scale RNN for action recognition from videos [13].

The Rise of the Attention Mechanism in Natural Language Processing

The attention mechanism has long been the research topic of machine learning researchers. With the rapid development in deep learning, the attention mechanism in deep learning is firstly developed in the area of natural language processing in [8], which developed a soft alignment model called soft attention mechanism for neural machine translation. Before the introducing of attention mechanism in machine translation, translation relies on reading a complete sentence and compress all information into a fixed-length vector, a sentence with hundreds of words represented by a fixed-length vector will inevitably lead to information loss, inadequate translation. The attention mechanism partially fixes this problem. It allows the machine to look over all the information of the original sentence, then generates a proper word according to its context.

Subsequently, many research followed the idea to improve the attention mechanism in natural language processing. Among them, [86] proposed the self-attention mechanism for neural machine translation and achieved the state-of-the-art results on the translation task. According to [86], an attention function can be described as mapping a Query and a set of Key-Value pairs to an Output, where the Query, Keys, Values, and Output are all vectors. Specifically, there are often three steps to establish the attention mechanism:

- Calculate the similarities between the Query and the Key. There are normally three ways to obtain the similarity
  - Dot Product between the Query and the Key.
  - Cosine Similarity between the Query and the Key.
  - Using a neural network to estimate the similarities between the Query and the Key.
- Normalize the similarity scores, often using Softmax function.
- Based on the attention weights which are the normalised similarity scores, calculate the weighted sum of the Value.

In discovering the text structure, the self-attention mechanism considers the Query, the Keys and the Values are the same things. Subsequently, numerous researches [87] [88] [88] [89] applied the self-attention mechanism in natural language processing due to its advantages in computing efficiency, the long-term dependency and the parallel computing capacity [86].

Chapter 3

Contextual Action Recognition from Still Images using Multi-branch Attention Networks

3.1 Introduction

Action recognition is one of the central issues in computer vision as actions often serve as the key instrument for the semantic description of an image containing humans. Actions are also directly linked to mid-level concepts for high-level tasks such as image captioning. Despite the tremendous progress made, there still exist many obstacles, particularly the description of the variations in human pose, the objects a person interacts with, and the scene where the action takes place. There are two pathways to study action recognition, namely video-based and still image-based. Among the two, video-based action recognition has been relatively well investigated [90] [91]. Still image based action recognition, however, has been studied less. The lack of motion information is arguably one of the major obstacles for still image based action recognition.

In the recent years, many methods have been proposed to tackle action recognition

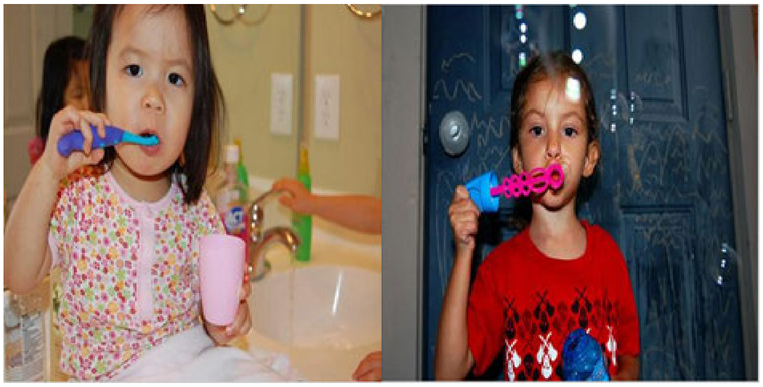


Figure 3.1: Example of similar pose leading to different actions.

problems. Among them, human-object interactions have been studied as one of the important instruments toward the recognition of object-related actions [92] [93]. As the human pose often plays a fundamental role in action recognition, another interesting approach is to find solutions for human pose estimation [94]. However, that approach is limited by the fact that similar poses can be associated with different actions. This is well illustrated in Fig. 3.1. The two children in the figure have similar poses. However, one is brushing her teeth while the other is blowing bubbles. The problem can be alleviated by either the introduction of contextual information, which is one of the main subjects of this chapter, or an appropriate combination of pose and human-object interaction as proposed in [95], in which a conditional random field is applied to jointly model the pose and objects a person is interacting with. Other approaches for still image based action recognition include the part-based model, with the Deformable Part Model (DPM) [96] as the most influential one. The Poselets model [97] further developed DPM, which employs key points to build an ensemble model of human body parts, achieving improved performance in some vision tasks.

Intuitively, the solutions to human action classification hinge on the acquisition of local and global contextual information. To be more specific, local information associated with discriminative parts or objects provides detailed contextual features which would be essential to action recognition. Object-related actions are associated with particular objects, which often provide critical hints for recognition. Additionally, the global contextual information about the configuration of surrounding scenes is also instrumental. To summarise, the comprehensive description of action comprises the articulation of body parts, the objects a person interacts with and the scene in which the action is performed. The action types in sports can well illustrate this. For example, for the action of ‘playing football’, the poses of players, the football itself and the football pitch are all strong evidence for this action category.

Specifically, to fully consider the contextual information when recognising actions, two types of contextual information are introduced: the scene-level context and region-level context, corresponding to the global and the local context respectively. The scene-level context is to consider the surrounding scene while the region-level context is to exploit the key body parts or objects a person is interacting with. The scene-level context is coarse-grained, and the region-level context is more fine-grained. In practice, given an image, the scene often means the background and region-level context are around the target person. Hence, these two kinds of context can be dealt with at the same time.

The relationship between contextual cues and visual attention has long been recognised [98]. Human perception is characterised by an important mechanism of focusing attention selectively on different parts of a scene. In NLP, the attention model has also been extensively studied, with applications including sequence to sequence training in machine translation [8], with the aid of two types of attention model, namely, hard attention and soft attention. Soft attention is deterministic and can be trained using backpropagation [4], which has also been extended and applied to the image captioning task [4]. Sharma et al. [65] used pooled convolutional descriptors with soft attention based models for video-based action recognition and achieved good results. However, the above works on attention-based networks are all implemented with RNNs. It would be interesting to investigate the applicability of attention mechanism in the general CNNs frameworks to

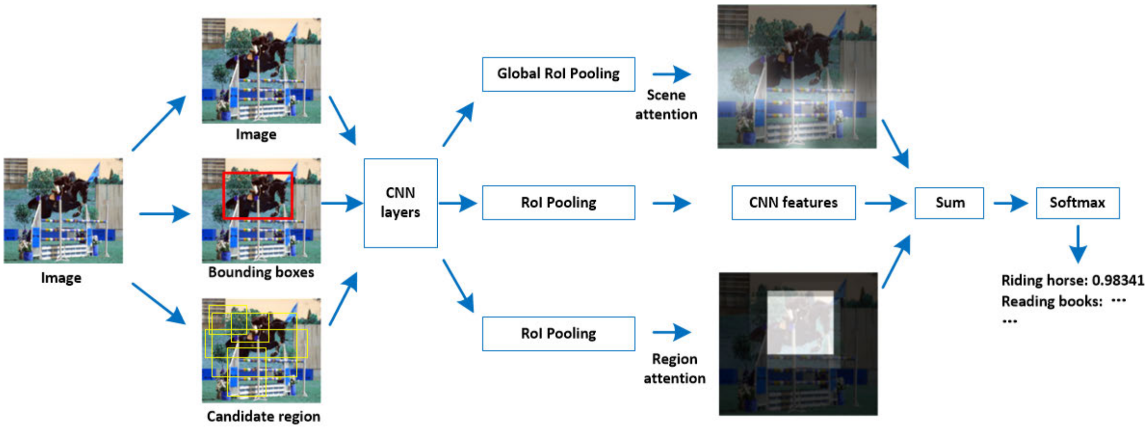


Figure 3.2: System diagram of our proposed Multi-branch Attention Networks.

which static images are the subjects to process. Though the spatial transformer networks proposed in [28] can be considered as an approach to realise soft attention in general CNN framework, our motivation is different from theirs as our model operates on both the region-level and scene level. To the best of our knowledge, this is the first research to incorporate soft attention mechanism into CNNs for action recognition from still images. For convenience, the proposed scheme is formed as Multi-branch Attention Networks. The CNN model with multi-branch attention mechanism can be trained in an end-to-end way, which can be illustrated by the system diagram shown in Fig. 3.2.

3.2 Related Works

In this section, some recent research on action recognition and attention models are reviewed and the relevance to our research is discussed.

3.2.1 Action Recognition

Video-based action recognition has been well studied. The recently published papers [99] provide a good literature review. Still image-based action recognition can be roughly categorised into three groups. The first group makes use of the human body information [94] [100]. Normally the bounding box of the human is used to indicate the location of the person. For instance, Thurau et al. [94] exploit human poses by learning a pose primitive for action recognition. There are also approaches making use of information from human body parts to aid the action recognition. Maji et al. [101] developed a body pose representation approach by learning and forming Poselets which are patches learned from body parts. Gkioxari et al. [100] concentrated on human body parts within a CNN model and developed a part-based approach by leveraging convolutional features, with the effectiveness demonstrated using several publicly available datasets.

The second group use the human-object interaction to discover the action categories by modelling the human-object pair and its interactions. For example, Yao et al. [95] modelled a person's body parts and objects by a conditional random field to recognise actions from still images. Yao et al. [93] developed Grouplet to recognise human-object interactions by encoding appearance, shape and spatial relations of multiple image patches. Desai et al. [102] formulated the problem of action recognition as a latent structure labelling problem and developed a unified, discriminative model for human object interaction. Recently, deep CNNs have also been employed for action recognition. For instance, Gkioxari et al. [24] proposed an interesting method by automatically selecting the most informative regions (usually the objects) around the person bounding box and achieved promising results on several datasets.

The third group have recourse to the scene context information. The background in an image can be taken as the context or scene of an executed action. For example, Delaitre et al. [103] studied the efficiency of different strategies based on the Bag of Visual Words (BoVW) approaches. It was found that the information extracted from the background does help to boost the performance of the recognition task. Similarly, Gupta et al. [104] encoded the scene for action image analysis and achieved excellent results.

As a contrast to the previously published approaches, both the objects a person is interacting with and the scene are considered as contextual information and are modeled them explicitly to form a unified, effective model. This is achieved with the aid of a soft attention mechanism embedded into a CNN model.

3.2.2 Attention Model

One important property of human perception is that humans do not tend to process a whole scene in its entirety at once. Instead, humans pay attention selectively on parts of the visual scene to acquire information where it is needed [63]. Different attention models have been proposed and applied in object recognition and machine translation. Mnih et al. [63] proposed an attention mechanism to represent static images, videos or as an agent that interacts with a dynamic visual environment. Also, Ba et al. [64] presented an attention-based model to recognise multiple objects in images. The two models mentioned above are all related with RNNs and with the aid of a reinforcement learning strategy.

Bahdanau et al. [8] proposed a novel attention model for neural machine translation without the prerequisite of reinforcement learning, which can be trained end-to-end by the backpropagation method. It is called a soft attention model. Later, a comprehensive study for hard attention bound with reinforcement learning and soft attention for the task of image captioning was published by Xu et al. [4]. Followed up researches include action recognition with soft attention proposed by Sharma et al. [65] and video description generation [66].

A related research topic, saliency detection, is also motivated by human perceptions. However, most of the saliency detection methods [58] [59] [60] used low-level image features, e.g., contrast, edge, intensity, which can be considered as fixed and bottom-up approach in contrast with the top-down approach of attention mechanism. Usually these methods cannot capture the task-specific information. Zhou et al. [61] applied global average pooling [62] to discriminate salient CNN features for the target object category. It is a kind of task-relevant approach. However, it is still not flexible enough to operate on the region-level as the soft attention does in this chapter. The region-level context, which is a more fine-grained feature, can be captured by region-level attention easily. In short, the attention mechanism is a more recent and flexible approach, which can learn relevant features for the specific task and plays a significant role in various vision tasks.

3.3 Approach

In this section, the proposed approach of Multi-branch Attention Networks for action recognition is introduced. The augmented CNN system contains three branches: target person region classification, scene-level attention and region-level attention.

Our model was built on the VGG16 [42], which is a very useful CNN structure for large-scale image analysis. According to [42], the VGG16 network has five convolutional blocks, each with three convolutional layers. The structure of convolutional layers is unchanged, with attention networks cascading these convolutional layers.

3.3.1 Classification of target person region

Usually the benchmark action recognition datasets provide the bounding box of the target person, e.g., the PASCAL VOC 2012 Action Challenge [68] and Stanford 40 Action Dataset [105]. As the model is fully supervised, this branch of CNN model is designed to perform the classical recognition of person regions. The RoI pooling developed by Girshick [106] is applied to pool different size regions into fixed size feature maps to facilitate the following classification. This branch is built based on Fast RCNN [106], only with some minor modifications. Specifically, the foreground with an overlap more than 0.5 with the target person region are selected, and the foreground over background ratio are set as 1, which indicates the framework is used for classification instead of detection. This can also be considered as a kind of data augmentation because the model samples on candidate regions instead of limiting the samples only on target person region.

3.3.2 Region-level Attention

To exploit the fine-grained properties of a given image, the second branch for the CNN is designed to explicitly capture more informative regions regarding the action performed. A similar strategy of selecting regions is used, as in the R*CNN [24]. In the R*CNN, a set of regions called secondary region is selected based on the overlap ratio with the bounding box of the target region. In this research, a ratio threshold is also set to select regions for this branch. Intuitively, the regions that overlap with a specific ratio usually indicate the parts of a person or objects a person is interacting with. The regions far away from a person will be ignored based on the overlap smaller than the threshold. As a result, more related regions will be selected for further processing at the first step.

Subsequently, selected regions are aggregated with RoI pooling resulting in the fixed size feature maps. In this branch, the fully connected features, instead of convolution features, is used, because there is a certain number of regions to process. Feature from fully connected layers have a lower dimension and hence can largely reduce the computational burden. All the extracted features are forwarded to a successive layer to generate a score map. If there are n regions each with d dimension, then the n feature maps can be shaped to one feature map with a dimension of n×d. Then the score map S is with a dimension of n×1. In practice, this is a fully connected layer which can be easily implemented. The soft attention model requires a region location softmax to generate the attention map which is expressed as follows:

$$A_i = \frac{exp(S_i)}{\sum_{i=1}^{n} exp(S_i)} \tag{3.1}$$

where A_i is the element of the attention map for the ith region. To allocate weights on regions, this attention map is elementwise multiplied with the features F:

$$\widetilde{F} = A \odot F \tag{3.2}$$

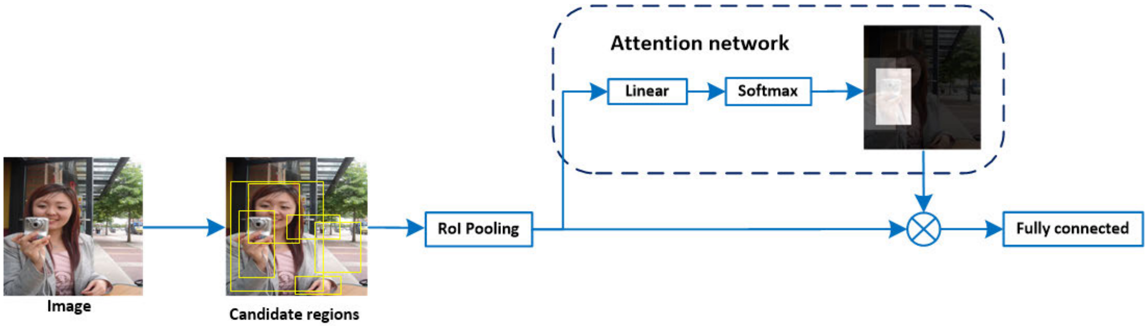


Figure 3.3: Illustration of region Attention.

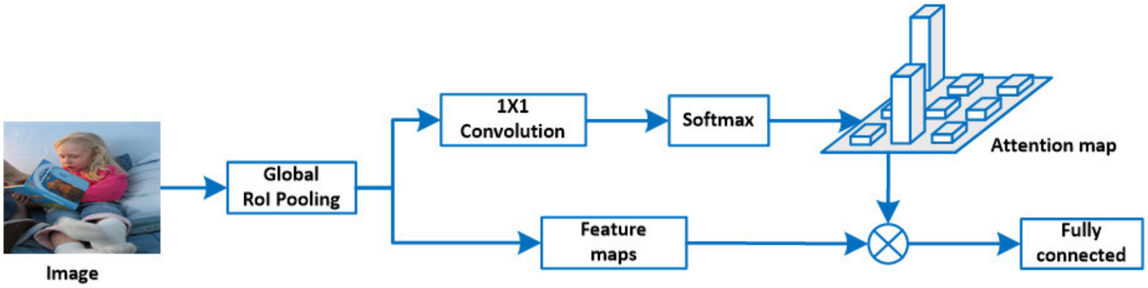


Figure 3.4: Illustration of scene Attention.

where $\widetilde{F}$ is the attentive feature map. To obtain a final feature representation of the regions, all the weighted features are summed up into one representation:

$$E = \sum_{i=1}^{n} \widetilde{F_i} \tag{3.3}$$

The feature representation of all weighted regions E can be used by the fully connected layer to obtain classification scores. More details of the block diagram of region attention branch are illustrated by Fig. 3.3.

3.3.3 Scene-level Attention

This branch of the CNN model is to consider the scene-level context of an action category. As previously explained, scene or background information often plays a supportive role in action recognition. However, indiscriminative extraction of all of the background would be counterproductive as some subregions of the scene may not be relevant to the action of interest. To solve this problem, the attention model is applied to select the most informative locations within the background discriminatively. Hence, the soft attention over the CNN features of the scene or background is used as one branch to aid the action recognition.

As a scene means typically the entire image, the original convolutional features is firstly pooled into a fixed size feature map by a new pooling layer: global RoI pooling, which divides the entire image or feature map into several grids and then performs max pooling inside each grid. The obtained feature map will have the same size regardless of the original image size. More formally, an image with arbitrary size can be pooled into a feature map F with size w×h×d, in which w, h and d are the width, height and a channel size of the feature map, respectively.

The pooled feature map is then convolved by a 1×1 convolution layer and the output channel of this convolution layer is also 1. A score map Z of w×h×1 can be consequently obtained. Following the practice of soft attention in [4], the score map is further processed by a location softmax which is defined as follows:

$$A_{ij} = \frac{exp(Z_{ij})}{\sum_{i=1}^{w}\sum_{j=1}^{h} exp(Z_{ij})} \tag{3.4}$$

where A_{ij} is the element of the attention map at position (i,j). Then the attention map A is elementwise multiplied with the feature map F which can be expressed as follows:

$$\widetilde{F} = A \odot F \tag{3.5}$$

where $\widetilde{F}$ is the attentive feature map. To obtain the final feature representation of the scene, the attentive feature map is summed over positions which can be described by:

$$E = \sum_{i=1}^{w}\sum_{j=1}^{h} \widetilde{F}_{ij} \tag{3.6}$$

The feature E is subsequently forwarded to the fully connected layers to obtain the classification scores. Fig. 3.4 further explains the scene-level attention mechanism implemented in the Multi-branch Attention Networks.

3.3.4 Networks Architecture

Table 3.1: Network Configuration

<table>
<tr><td colspan="3">Inputs (Images, candidate regions and labels)</td></tr>
<tr><td colspan="3">Convolution Blocks (Conv1-Conv5) derived from VGG16 [42].</td></tr>
<tr><td>Scene-level attention</td><td>Target person region</td><td>Region-level attention</td></tr>
<tr><td>Global RoI Pooling
(Pooled size: 7×7)</td><td>RoI Pooling
(Pooled size: 7×7)</td><td>RoI Pooling
(Pooled size: 7×7)</td></tr>
<tr><td>1×1 Convolution
(Channel number: 1)</td><td>FC1
(Dimension: 4096)</td><td>Region-FC1
(Dimension: 4096)</td></tr>
<tr><td>Softmax
(Over location)</td><td>FC2
(Dimension: 4096)</td><td>Linear
(Dimension: 1)</td></tr>
<tr><td>Elementwise Product</td><td rowspan="6">Score</td><td>Softmax
(Over input regions)</td></tr>
<tr><td>Sum
(Over Location)</td><td>Elementwise Product</td></tr>
<tr><td>Scene-FC1
(Dimension: 512)</td><td>Sum
(Over Input regions)</td></tr>
<tr><td rowspan="3">Scene-core</td><td>Region-FC2
(Dimension: 4096)</td></tr>
<tr><td>Region-FC3
(Dimension: 4096)</td></tr>
<tr><td>Region-score</td></tr>
<tr><td colspan="3">Sum Scores
(Dimension: Number of Categories)</td></tr>
<tr><td colspan="3">Softmax</td></tr>
<tr><td colspan="3">Cross Entropy Loss</td></tr>
</table>

The details of the CNN architecture are given in Table 3.1. The convolutional blocks are derived from the VGG16 model [42] which includes 5 blocks. The more detailed explanation can be referred to [42].

There are a total of 3 branches following the convolutional blocks starting from a Global

RoI Pooling layer followed by two RoI Pooling layers. The Global RoI Pooling compresses the entire image into 7×7 feature map, which is used for scene-level attention networks. It starts from a 1×1 convolution layer with a channel number of 1. A location softmax layer is connected to generate the attention map. The feature map and attention map are subsequently processed simultaneously and fused into a weighted sum of the features from each location. The following fully connected layer is named 'Scene-FC1' of size 512. The 'Scene-score' can be obtained based on the outputs of the fully connected layer.

The first RoI Pooling (the middle column in Table 3.1) pools the region of the target person into a fixed size feature to perform classical CNN recognition. The second RoI Pooling (the right column in Table 3.1) pools candidate regions generated with selective search algorithm into fixed size feature maps. These feature maps are then forwarded to the fully connected layers 'Region-FC1' to generate feature maps with a dimension of 4096. The region softmax transfers outputs from a linear layer into an attention map over regions. The attention map is elementwise multiplied with the features and summed into a whole feature representation before another two fully connected layer. The 'Region-score', 'Score' and 'Scene-score' are summed and activated by the Softmax layer with Cross Entropy Loss for the training.

3.3.5 Training Strategy

The common pre-training plus fine-tuning practice of applying CNN model is applied for this model. Specifically, the pre-trained VGG16 model on ImageNet [107] was fine-tuned for the task at hand.

The two branches of the attention mechanism can be considered as subsets of parameters towards image features, which are to be found by overall optimisation for the action classification task. Such paramters on the optimisation task, make the direct application of Stochatic Gradient Descend (SGD) very challenging. Our intuition is to borrow the idea from alternating optimisation [108].

More formally, the full parameter set of the CNN model can be considered as $X = \{X_1, X_2\}$, where X_1 corresponds to the parameters from the branch of the target person region classification and region-level attention and X_2 indicates the parameters from the branch of scene-level attention. The task is to optimise the CNN model which is a function of these parameters: $F = F(X)$. Alternating optimisation is an iterative procedure to minimise all the variables by alternating restricted minimisations over the individual subsets of variables X, in this case, X_1 and X_2 [108]. Specifically, a two-step training strategy is proposed for our networks: the target person region recognition and region-level attention are trained jointly at first. This is equal to optimise over the subset of X_1. Then the scene-level attention is added to the network while keeping the weights from the convolutional blocks, the target person region classification and the region-level attention unchanged. This means the optimisation over subset X_2 is performed subsequently. In the first-step training, the maximum iterations were set as 40,000. Once trained, the model was added with the scene-level attention branch and further trained with other 25,000 iterations.

As indicated in [106], training all the convolutional layers of VGG16 model would be unnecessary. Instead, the first two convolutional blocks are kept unchanged and trained other layers during the first-step training.

During training, 50 candidate boxes were randomly selected based on a threshold of overlap ratio for the training of region attention as 50 boxes can reach a balance of training efficiency and generalisation capability. 500 candidate boxes were selected for region-level attention network when testing as 500 boxes can cover most of the important regions. Further increasing this number may introduce noise and also slow the testing process.

As shown in Table 3.2, for the proposed model, the training takes about 1s per iteration and testing takes about 0.2s per image. Compared with the Fast RCNN (single branch), the training time increased because of the additional branches. However, the differences of testing time are minute, with 0.16s for Fast RCNN and 0.2s for our model, respectively. Actually, when testing on PASCAL VOC 2012, both Fast RCNN and our model take about

Table 3.2: Efficiency Analysis of the proposed model on a PC embedded with a TITAN X GPU

Model	Fast RCNN	Ours
Training	0.70s (per iteration)	1.00s (per iteration)
Testing	0.16s (per image)	0.20s (per image)

Table 3.3: The AP results on PASCAL VOC validation set

Approach	jumping	phoning	playing instrument	reading	riding bike	riding horse	running	taking photo	using computer	walking	Mean AP (%)
Image classification (VGG16 model)	78.9	64.0	91.5	71.6	88.6	92.6	83.2	71.1	89.7	53.9	78.5
Fast RCNN (single branch, no regression) [106]	82.4	69.9	90.7	72.1	93.5	97.0	84.1	82.7	87.6	65.6	82.6
Fast RCNN (single branch, with regression) [106]	87.4	70.2	91.2	75.0	**95.4**	**97.8**	85.7	81.6	85.9	72.4	84.3
Two branch (no regression, with threshold)	86.3	76.6	90.8	79.6	93.6	97.0	85.6	84.4	92.5	67.4	85.4
Multi-branch attention (no regression, with threshold)	87.8	77.0	92.3	81.4	94.4	96.5	86.2	82.8	92.2	71.3	86.2
Two branch (with regression, no threshold)	85.8	73.2	90.0	81.8	93.3	96.3	85.0	78.2	90.7	70.3	84.5
Multi-branch attention (with regression, no threshold)	85.6	72.7	91.4	81.3	93.4	96.6	84.8	79.1	90.4	70.8	84.6
Two branch (with regression, with threshold)	87.8	77.1	92.5	**81.4**	94.3	96.5	**86.3**	83.3	92.2	71.1	86.3
Multi-branch attention (with regression, with threshold)	**87.8**	**78.4**	**93.7**	81.1	95.0	97.1	86.0	**85.5**	**93.1**	**73.4**	**87.1**

5-7 minutes to finish, which indicates that the efficiency does not seriously deteriorate though there are two more branches in the model compared with Fast RCNN.

3.4 Experiments

The Multi-branch Attention Networks were implemented based on the Caffe platform [109]. The training was conducted with stochastic gradient descent (SGD) with a batch size of 32. All the experiments were conducted with a Nividia Titan X GPU installed in a PC running the Ubuntu 14.04 operating system.

3.4.1 Experimental Setting 1 (with the bounding box of the target person)

PASCAL VOC 2012 Action Dataset

The PASCAL VOC Action dataset serves as one of the PASCAL VOC 2012 challenges [68], which consists of 10 different actions, jumping, phoning, playing an instrument, reading,

Table 3.4: Comparison of each of the three branches and their random combinations on PASCAL VOC validation set.

Approach	jumping	phoning	playing instrument	reading	riding bike	riding horse	running	taking photo	using computer	walking	Mean AP (%)
Fast RCNN alone (the first branch)	87.4	70.2	91.2	75.0	95.4	97.8	85.7	81.6	85.9	72.4	**84.3**
Region-level attention alone (the second branch)	80.7	70.0	88.8	79.7	89.6	94.4	81.3	75.4	88.8	66.3	81.5
Scene-level attention alone (the third branch)	66.3	67.0	82.5	66.9	77.9	84.4	71.4	62.5	85.2	46.5	71.0
The first and second branch	87.8	77.1	92.5	81.4	94.3	96.5	86.3	83.3	92.2	71.1	**86.3**
The first and third branch	83.2	70.0	90.3	72.7	89.5	92.6	82.0	74.4	89.7	65.3	81.0
The second and third branch	83.9	78.1	93.8	80.9	93.6	95.4	84.9	82.7	93.0	69.9	85.6
Multi-branch attention	87.8	78.4	93.7	81.1	95.0	97.1	86.0	85.5	93.1	73.4	**87.1**

Table 3.5: The AP results on PASCAL VOC test set

Approach	CNN layers	jumping	phoning	playing instrument	reading	riding bike	riding horse	running	taking photo	using computer	walking	Mean AP (%)
Oquab et al. [110]	8	74.8	46.0	75.6	45.3	93.5	95.0	86.5	49.3	66.7	69.5	70.2
Hoai [111]	8	82.3	52.9	84.3	53.6	95.6	96.1	89.7	60.4	76.0	72.9	76.3
Action Part [100]	16	84.7	67.8	91.0	66.6	96.6	97.2	90.2	76.0	83.4	71.6	82.6
Simonyan et al. (VGG16 model) [42]	16&19	89.3	71.3	94.7	71.3	**97.1**	98.2	90.2	73.3	88.5	66.4	84.0
R*CNN [24]	16	91.5	84.4	**93.6**	83.2	96.9	98.4	**93.8**	**85.9**	**92.6**	**81.8**	**90.2**
Multi-branch attention (ours)	16	**92.7**	**86.0**	93.2	**83.7**	96.6	**98.8**	93.5	85.3	91.8	80.1	**90.2** *

* The official results: `http://host.robots.ox.ac.uk:8080/leaderboard/displaylb_noeq.php?challengeid=11&compid=10`

riding bike, riding horse, running, taking photo, using computer, walking as well as examples of people not performing some of these actions, which are labeled as other. The target person boxes containing the people are provided both at training and testing time. During testing, for every sample, the probabilities for all actions are calculated and the Average Precision (AP) are computed.

The challenge organisers require participators to make use of the validation set for parameter optimisation and the test set to report performance [68]. Hence, the performance on the validation set are first measured and then the results of the test set are submitted to the evaluation server. When evaluating the validation set, the training set was used only for training. Both the training set and the validation set were applied for training when submitting results for the test set and evaluating performance.

The comparative experiments were conducted to optimise parameters and confirm the effectiveness of the proposed model. Table 3.3 provides the AP results on the validation set. From the table, the following observations can be obtained:

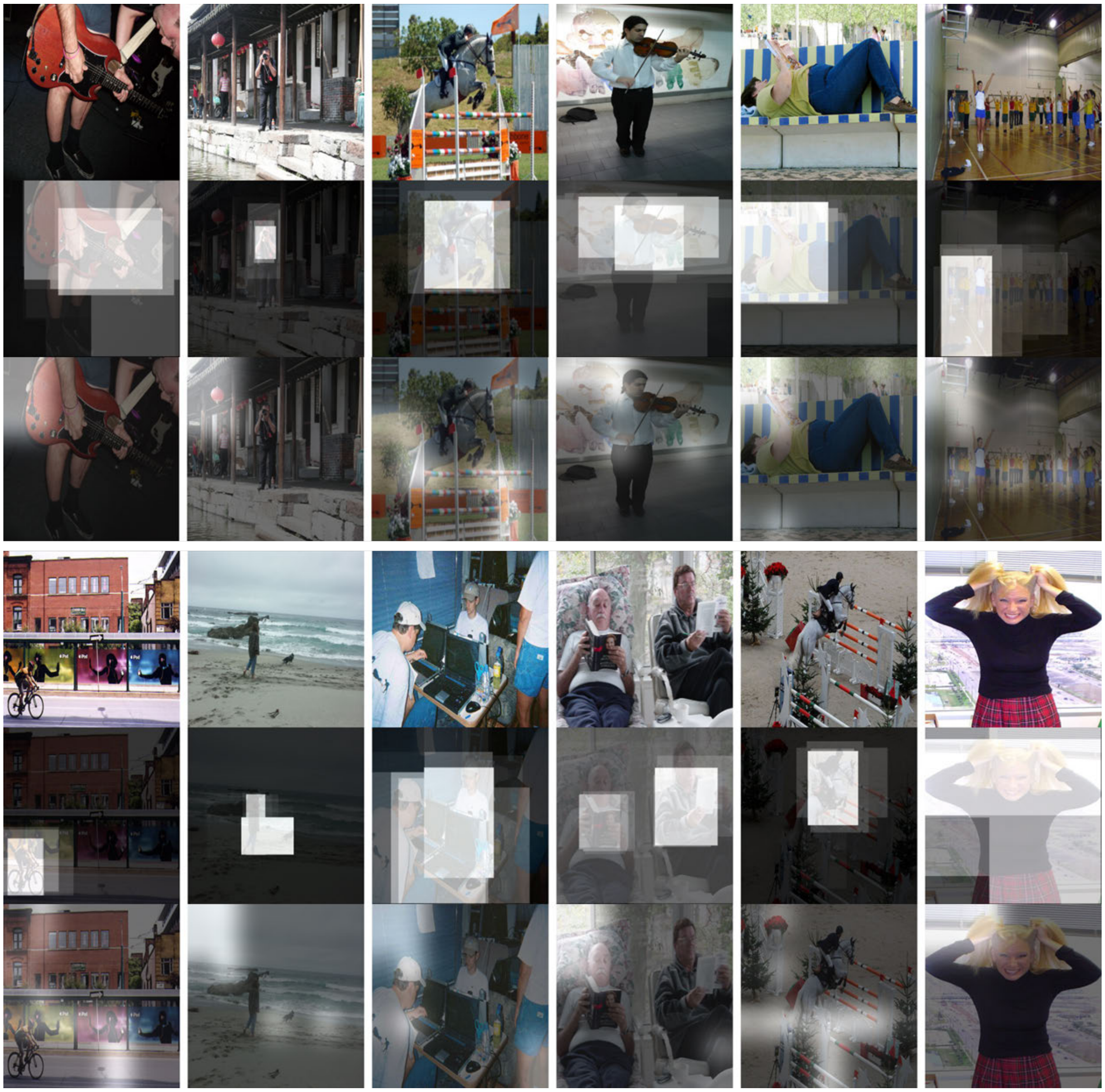


Figure 3.5: Visualization of region attention and scene attention on the PASCAL VOC test set.

Baseline approach The Fast RCNN [106] was set as the baseline approach because it is generally acknowledged as a better object detection model with much-improved performance than RCNN [112]. However, Fast RCNN is not limited to detection and can also be applied in action recognition from still images [24], with some modifications. Specifically, the foreground over background ratio in Fast RCNN is set as 1 during training, which indicates the model does not need to discriminate foreground from a background as in the detection scheme.

For Fast RCNN without bounding box regression, 82.6% AP was reported. Adding bounding box regression can boost the AP performance to 84.3% which testifies again that multi-task training with bounding box regression can boost the performance reported earlier [106].

The proposed methods Experiments were conducted for the two-step training strategy as explained above. The AP performance from the first-step model which uses a two-branch network (target person region and region-level attention) is reported first. Borrowing the benefit of bounding box regression in Fast RCNN, a regression layer is added in the first-step training. It turns out that our model achieves better AP results than Fast RCNN, with 85.4% AP without bounding box regression and 86.3% AP when adding the bounding box regression layer. Also, the threshold for selecting candidate boxes plays a significant role in promoting performance. Specifically, boxes overlap more than 0.1 and less than 0.7 with bounding boxes of the target person were selected for the branch of region attention. The obvious improvement of AP performance when adding threshold is indicated in Table 3.3. This is reasonable because only bounding boxes that overlap with the person in a range can exploit useful context information such as the objects the person interacts with. After this, the second-step training was conducted to train the Multi-branch Attention Model. As the weights from the branch of target person bounding box classification are kept as constants, there is no need to add bounding box regression in the second-step training. In summary, our proposed Multi-branch Attention Networks produce the best mean AP value (87.1%) among all the experimental settings which validate the effectiveness of this model.

To further evaluate our method and compare it with other newly proposed approaches, the experimental results on the test set were generated and submitted to the PASCAL VOC evaluation server for the final evaluation. The training strategy explained in Section 3.3.5 is used, considering both the training set and validation set of PASCAL VOC 2012 as the training set. This is a reasonable deployment as the challenge organisers allow the validation set to be used in training when reporting results on the test set. Once trained with alternating optimisation of 40,000 and 25,000 iterations as the first step and second step, respectively, the model was directly used for testing. Also, a current leading method such as R*CNN [24], which will be discussed later, used a similar strategy. Hence, it is also a fair deployment. Table 3.5 shows the AP results of the proposed approach and other competing methods. Oquab et al. [110] trained an 8-layers network on the box of the target person to perform action recognition. Hoai et al. [111] used an 8-layers CNN model to extract features from fully-connected layers from regions at multiple locations and scales inside the image and accumulate their scores for prediction, which is more comprehensive than only training on the box from target person. The results of this method are also better than Qquab et al. [110]. Simonyan et al. [42] combined the VGG16 and VGG19 network and re-trained classifier such as SVMs using fully connected features from the target person region and entire image.

The current top-ranked method on the PASCAL VOC 2012 Action dataset is R*CNN [24] which was trained on the target person region with a secondary box. The secondary box was selected using the multi-instance learning method during training and testing. Specifically, R*CNN applied the max operation on scores generated by secondary boxes and combined them with the target person region for recognition. Our methods achieved the same mean AP results with R*CNN, with a 90.2% mean AP value on the testing set.

A visualisation of the attention model is provided in Fig. 3.5. The original image, region-level attention and scene-level attention are plotted in separate three rows. The brighter a place of an image is, the more important it is for recognition. The region-level attention generates important bounding boxes while scene-level attention captures attentive regions as indicated by the figure. It is interesting to discover that normally the two attention models generate different regions which imply that they are complementary. Note that all the example images are randomly selected.

Analysis of each of the three branches Table 3.4 presents the AP results of each of the three branches and their random combinations. In single branch settings, the branch of general image features (Fast RCNN branch) yields the best results, which shows that the person regions play a fundamental role in recognition. Besides this branch, the most important branch is the second branch (region-level attention), which discover the fine-grained contextual information. From the table, it is evident that the third branch alone (scene-level branch) cannot provide excellent results. However, as discussed previously, when fused together with the other two branches, satisfactory results can be obtained, which indicates that there are little correlations between scene-level attention branch and the other two branches. This is also what is intended to accomplish by the alternating optimisation initially, which is to guarantee that the scene-level attention is to capture the complementary information of the other two branches. In the random combinations of two branches, the first and second branch together generates best results. This phenomenon shows consistency with the performance of a single branch as the first and second branch are the two most important parts of the networks.

Stanford 40 Dataset

The proposed method was also evaluated on the Stanford 40 dataset [105] which is a larger database containing 40 different types of daily human actions. It has 9352 images in total. The number of images for each class ranges from 180 to 300. The dataset provides the

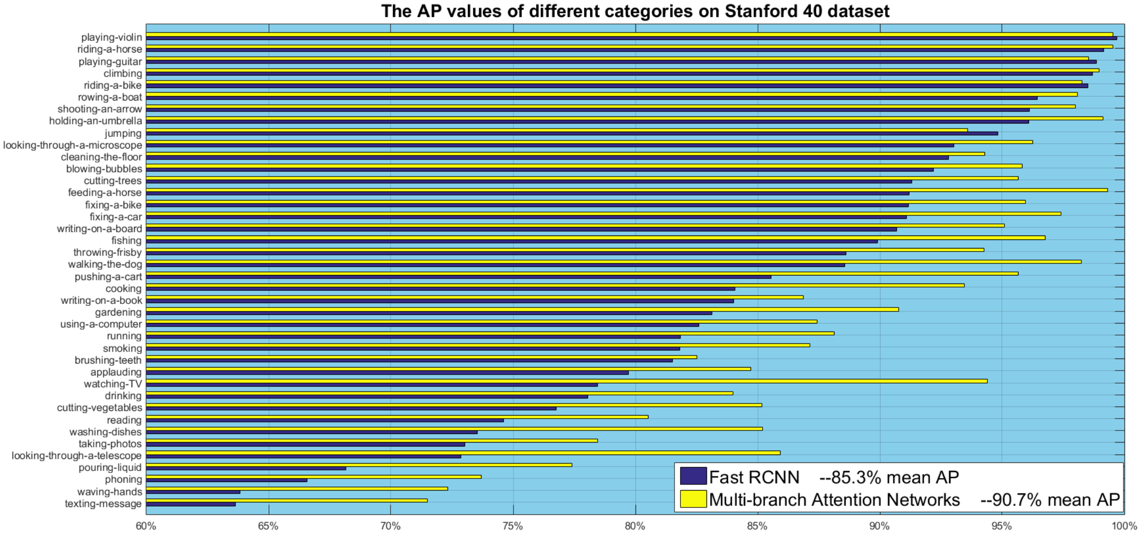


Figure 3.6: The AP results for different categories on the Stanford 40 dataset.

Table 3.6: The AP results on the Stanford 40 dataset and comparison with previous results.

Method	Mean AP(%)
Object bank [113]	32.5
LLC [114]	35.2
EPM [115]	40.7
DeepCAMP [116]	52.6
Khan et al. [117]	75.4
Semantic parts [118]	80.6
VLAD spatial pyramids [11]	88.5
R*CNN [24]	**90.9**
Fast RCNN alone [106]	85.3
Region-level attention alone	81.0
Scene-level attention alone	72.1
Two branch (ours)	90.6
Multi-branch Attention Networks (ours)	90.7

training and testing splits for each class, namely 100 images of each class for training and the rest for testing.

Fig. 3.6 shows the bar chart of the AP values over the 40 action categories from the

Fast RCNN (the baseline) and our Multi-branch Attention Networks. It is apparent from the figure that our approach outperforms the baseline by a large margin, with a 90.7% mean AP compared with 85.3% mean AP of baseline approach.

Table 3.6 shows a comparison of our methods with alternative approaches. The model with two branches (from first-step training) shows good results. The improvement of the mean AP result by adding the branch of scene-level attention is obvious. With the Multi-branch Attention Networks, the mean AP result is improved to 90.7%. To conclude, a comparable results with the current state-of-the-art method (R*CNN) on the Stanford 40 dataset are achieved with a 5.4% higher mean AP than Fast RCNN which is the baseline method.

3.4.2 Experimental Setting 2 (without the bounding box of the target person)

The bounding boxes of target persons are crucial during training as they provide the fundamental feature for the person to be recognised. However, they are often hard to obtain during real-world applications as the manual annotation for the bounding boxes is somewhat time-consuming and painful. Also, the requirements of inputting bounding boxes severely discourage further applications of the topic. Hence, in this section, it is shown that when the annotations of the bounding boxes of the target person are not provided, the proposed model can also perform well in the task of action recognition.

As the bounding box of the target person during training and testing is not utilised, the model architecture is modified to facilitate the recognition. From results of experimental settings 1, if lacking the general CNN features of the target person, the most important branch is the region-level attention. Hence, in order to make the networks effective and simple, two branches in the networks are set for this experimental settings:

- Image classification Branch: The entire image is forwarded to a Global RoI pooling layer and perform general image classification. This is a fundamental branch which

Table 3.7: The AP results on PASCAL VOC validation set (experimental setting 2)

Approach	jumping	phoning	playing instrument	reading	riding bike	riding horse	running	taking photo	using computer	walking	Mean AP (%)
Image classification (VGG16 model)	78.9	64.0	91.5	71.6	88.6	92.6	83.2	71.1	89.7	53.9	78.5
Ours	78.4	72.1	91.4	75.4	88.9	93.7	84.3	70.2	90.3	55.5	80.0

Table 3.8: The AP results on PASCAL VOC test set (experimental setting 2)

Approach	jumping	phoning	playing instrument	reading	riding bike	riding horse	running	taking photo	using computer	walking	Mean AP (%)
Simonyan et al. (Image classification) [42]	-	-	-	-	-	-	-	-	-	-	79.2
Zhang et al. (Minimum annotations) [119]	86.7	72.2	**94.0**	71.3	**95.4**	97.6	88.5	72.4	88.8	65.3	83.2
Ours	**87.2**	**81.5**	89.9	**78.8**	94.4	**94.9**	**90.0**	**73.8**	**90.0**	65.3	**84.5**

also provides a baseline of our two branch model.

- Region-level attention Branch: The region attention branch is to retrieve relevant regions during recognition automatically, this is similar to the region attention branch explained previously. The only difference here is that the bounding box selection is omitted as the region of the target person is not provided.

During training, the two branches are trained jointly with 40,000 iterations under the Caffe platform. The AP results on PASCAL VOC 2012 action dataset are then reported.

PASCAL VOC 2012 Action Dataset

As shown in Table 3.7, the model achieved 80.0% mean AP performance on the PASCAL VOC validation dataset while the general image classification only achieved 78.5% mean AP result. To further validate the proposed methods, the AP results from PASCAL VOC evaluation server are then reported. As shown in Table 3.8, the proposed model got 84.5% mean AP, which are the state-of-the-art results among the methods without training bounding boxes. This can be attribute to our region-level attention branch, which serves as a model which can automatically retrieve not only the contextual information but also the person region, in experimental setting 2.

Table 3.9: The AP results on the Stanford 40 dataset with experimental settings 2.

Method	Mean AP(%)
Image classification (VGG16 model)	81.4
Zhang et al. (Minimum annotations) [119]	82.6
Ours	85.2

Stanford 40 Dataset

Table 3.9 provides the performance on Stanford 40 action dataset on the experimental setting 2. The proposed method achieves 85.2% mean AP results on the 40 action categories of the dataset, which is competitive with the mean AP results (85.3%, see Table 3.6) of Fast RCNN (with training bounding boxes). Also, our method leads the scheme in [119], which is a recently proposed action recognition method without the training bounding boxes.

HICO dataset

The PASCAL VOC action dataset and stanford 40 dataset can be considered as medium-sized datasets. To further test the generalisation capability of the proposed approach on a big dataset, experiments were also conducted on Humans Interacting with Common Objects (HICO) dataset [120]. This dataset is currently the largest one for action recognition, which consists of 50,000 images labelled to 600 human-object interaction categories. It is also related to Common Objects in Context (COCO) dataset [121] as each category in the HICO dataset is composed of a verb-object pair, with objects belonging to the 80 object categories from MS COCO. However, the HICO dataset does not provide human bounding boxes for a pre-defined action category. Hence, it is only suitable for the experimental setting 2 in this chapter.

Different from PASCAL VOC action dataset and stanford 40 dataset in which the action categories are exclusive, more than one human-object interaction category is labelled for a

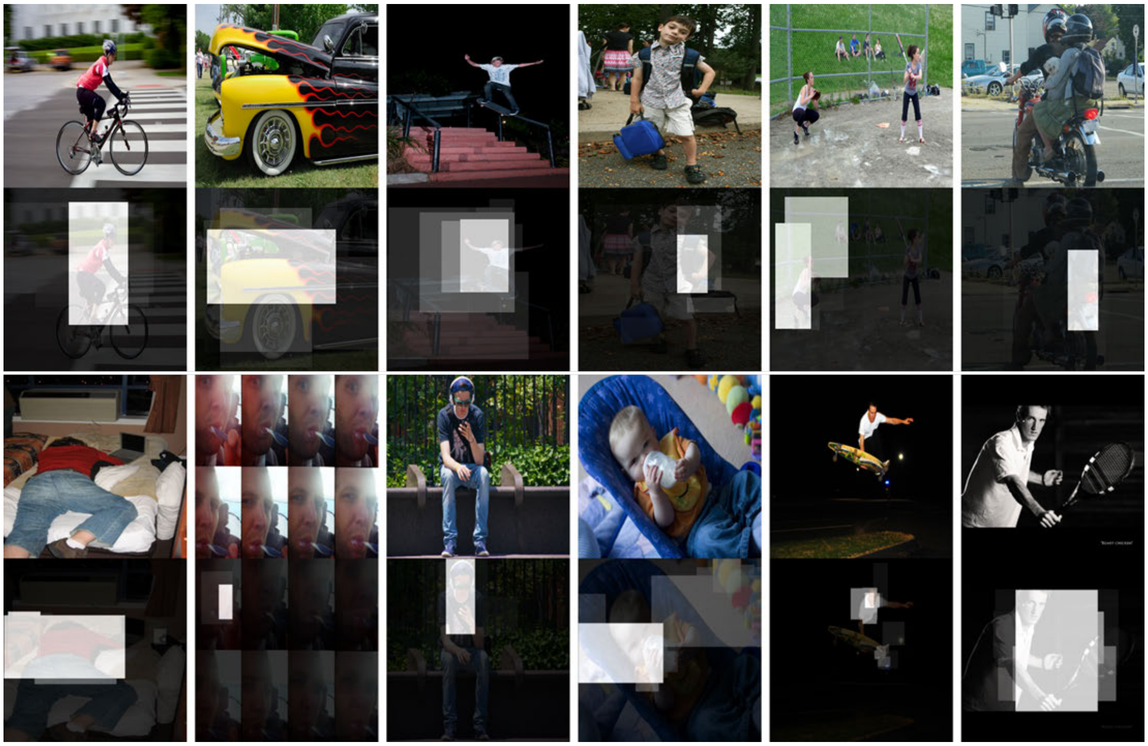


Figure 3.7: The learnt region-attention map of HICO dataset in the experimental setting 2.

Table 3.10: The mean AP results on the HICO dataset with experimental settings 2.

Method	Mean AP(%)
AlexNet+SVM [120]	19.4
VGG16, Image classification [122]	29.4
VGG16, R*CNN [122]	28.5
VGG16, Scene-RCNN [122]	29.0
RoI and Scene fusion [122]	**33.6**
Ours	32.8

single instance. These action categories can be considered as mid-level features, in contrast with those as high-level actions in PASCAL VOC and Stanford 40 dataset. Hence, we treat each of the human-object interaction categories as a binary classification problem and use Sigmoid as the activation function instead of Softmax. As the dataset is larger, we train them with 60,000 iterations in Caffe platform and report the mean AP results of our approach.

Table 3.10 demonstrates the mean AP results of our approach and comparison with other methods. Specifically, the baseline approach reported in [120] applied an AlexNet and SVM classifier for recognition, with only 19.4% mean AP. [122] reported results of several methods. They first applied VGG16 for general image classification approach, achieved 29.4% mean AP. For R*CNN approach, they used a pre-trained Faster RCNN object detector to detect human bounding boxes. With these bounding boxes, they then trained R*CNN and Scene-RCNN as in [24]. However, the mean AP results of R*CNN and Scene-RCNN is even worse than general image classification, the possible reason, as explained in [122], is that R*CNN try to find a single box using multi-instance learning, which is not able to cover all 600 action categories. This is not a problem in the proposed method because the region-level attention is fully exploited and 500 boxes are sampled to facilitate the recognition. As shown in Table 3.10, the proposed approach achieved competitive results with the one proposed in [122] but are simpler and more efficient. A visualisation of learnt attention region is shown in Fig. 3.7.

Table 3.11: P-value for the Obtained Results in the Experiments

	Dataset	Upper-tailed p-value
Experimental Setting 1	Pascal VOC Validation Set	0.0009
	Pascal VOC Test Set	0.0052
	Stanford 40 Dataset	0.0
Experimental Setting 2	Pascal VOC Validation Set	0.0260
	Stanford 40 Dataset	0.0
	HICO Dataset	0.0310

3.4.3 Testing the Statistical Significance of Experimental Results

For a more comprehensive evaluation of the proposed model, in addition to the mean AP evaluation protocol, the practice in [123] [124] [125] is applied to test the statistical significance of our experimental result through Fisher-Pitman permutation tests. Specifically, the evaluation protocol of [126] is applied to calculate the upper-tailed p-value of the AP results from the baseline (image classification using VGG16) and the proposed model. To test if a null hypothesis can be rejected, p-value calculated using permutation tests is a suitable evaluation protocol [127].

A result has statistical significance when it has a low probability of occurring given the null hypothesis [128]. Specifically, the null hypothesis is set as that the proposed model does not bring an improvement on the performance. Then the permutation tests were performed on all the datasets used for both of the Experimental Setting 1 and Experimental Setting 2. The results can be seen in Table 3.11. As indicated by the results, the upper-tailed p-values from the listed datasets are close to 0. Also, all the upper-tailed p-values are smaller than 0.05, which [127], indicates that the null hypothesis can be rejected with statistical significance. This validates the research hypothesis that the proposed model can improve the performance.

3.5 Discussion

The proposed method has demonstrated impressive performances for action recognition. This can be attributed to the following aspects:

- When modeling the contextual information, we proposed to discriminate between two categories of context, the scene-level context and the region-level context, and model them in a complementary way. This is illustrated in Fig. 5.3, where the attentive parts in region-level attention and scene-level attention are captured differently.

- The two-step training strategy can better optimize the model. The first-step is to jointly train two branches (target person box classification and region-level attention network). Intuitively, the region-level context has a close relationship with the features extracted from the target person area. Hence, training them jointly can be beneficial. When the weights of the firstly trained branches are fixed, in order to optimize the model, a newly added layer is able to capture complementary information of the first two branches. Hence, this training strategy can guarantee that the newly added branch, namely scene-level attention network, can capture complementary information. This is also clarified by the direct increase in AP results when accomplishing the second-step training as revealed in Table 3.3 and Table 3.6.

- As we systematically analyzed the performance of each of the three branches in Table 3.4, we came to the conclusion that when training bounding box of target person is missing, the most important branch is the region-level attention. Based on this realization, a two branch model was designed for experimental setting 2, which also achieved satisfactory results.

- The R*CNN [24] already show good results on the experimental setting 1, with competitive results on several benchmark dataset compared with our methods, however, in experimental setting 2, the R*CNN are not as good as the proposed methods. The major reason is that our model tries to capture comprehensive contextual information whilst the R*CNN only uses the multi-instance learning to grasp the maximum contributor to the recognition.

3.6 Conclusion

This chapter proposed a novel CNN model abbreviated as Multi-branch Attention Networks for action recognition in still images. This model incorporates a soft attention mechanism into a CNN model to explicitly exploit scene-level context and region-level context. The two context branches and target person region classifications are integrated for the final prediction. A two-step training strategy was proposed based on alternating optimisation. Comprehensive experiments have been conducted for comparisons on both experimental settings with and without the bounding boxes of the target person, with results on the PASCAL VOC action dataset, the Stanford 40 dataset and HICO dataset verifying the advantages of the proposed model. The proposed methods can be easily extended to achieve better performance by using more advanced CNN model like Residual-Net [129].

Chapter 4

Action Recognition from Video Sequences based on Visual Attention Mechanism

4.1 Introduction

The task of action recognition from video sequences is the main focus of this chapter. Two pieces of research are presented in Section 4.2 and Section 4.3, respectively. In Section 4.2, a convolutional hierarchical attention model is proposed by utilizing convolutional LSTMs with attention and a self-defined hierarchical architecture for action recognition. This piece of research show good results but the self-defined hierarchical architecture is not flexible enough in dealing with long video sequence. Hence, in the second piece of research, which corresponds to Section 4.3, a hierarchical multi-scale RNNs is applied with two kinds of attention mechanism: the soft attention and hard attention, for action recognition. The hierarchical structure in this section is automatically learnt from data, and show better performance than previous research.

4.2 Action Recognition Using Convolutional Hierarchical Attention Model

4.2.1 Introduction

Action recognition in the video has been a popular yet challenging task which has received significant attention from the computer vision society [130] [90]. The potential applications of action recognition include video retrieval (i.e., YouTube videos), intelligent surveillance and interactive systems. Compared with action recognition from still images, the temporal dynamics provides an important clue to recognise human actions in videos.

Among the proposed models to capture the spatial-temporal transition in videos, RNNs are the preferred candidate due to the special internal memory being able to process arbitrary sequences of inputs. An RNN is a class of artificial neural network where connections between the units form a directed cycle, and the internal state created from the network allows it to exhibit dynamic temporal behaviour. Much research was conducted on RNNs in the 80s [131] [132] for time-series modelling, however, this was hampered for a long period by the difficulties of training, particularly the vanishing gradient problem [44]. Roughly speaking, the error gradients would vanish exponentially quickly with the size of the time lag between important events, which makes training very difficult. To mitigate this problem, a class of models with a long-range learning capability, called LSTMs, was introduced by Hochreiter, et al [45]. LSTM consists of memory blocks, with each block containing self-connected memory units to learn when to forget previous hidden states and when to update hidden states given new information. It has been verified that complex temporal sequences can be learnt by LSTM [133].

LSTM has a close relationship with attention models in vision research and NLP. Human perception is characterised by an vital mechanism of focusing attention selectively on different parts of a scene which has long been an essential subject in the vision community. An attention model can be built using LSTM on top of image features to decide when the model should focus on certain parts of the image sequentially. In NLP, the attention model was proposed for the sequence to sequence training in machine translation [8], where two types of attention model have been studied, hard attention and soft attention. Soft attention is deterministic and can be trained using back-propagation [4]. Soft attention was then extended to the image captioning task [4] since image captioning can be mostly considered as an image to language translation. Sharma, et al.[65] used pooled convolutional descriptors with soft attention based models for action recognition and achieved good results. Continuing the previous research, the soft attention model in the action recognition context is investigated, and several improvements are proposed. Usually the LSTM is built on fully connected layers in which all the state-to-state transitions are matrix multiplication. This structure does not take spatial information into account. Xingjian, et al.[134] proposed convolutional LSTM in which all the transitions are convolutional operations. Following [134], the soft attention model is improved by using the convolutional LSTM.

In real-world applications, action is usually composed of a set of sub-actions. For instance, jump shooting basketball often consists of three sub-actions- jumping, shooting and landing. This is a typical hierarchical structure regarding motion dynamics. In other words, actions are composed of multiple granularities. A straightforward way to model the layered action would be a hierarchical structure. Following [135] in which a Hierarchical Attention Networks (HAN) was proposed, HAN was applied with a convolutional LSTM to recognise multiple granularities of layered action categories. The proposed model can be termed CHAM which means Convolutional Hierarchical Attention Model (CHAM).

The main contributions can be summarised as follows:

(1) As deep features from CNNs preserve the spatial information, the soft attention model is improved by introducing convolutional operations inside the LSTM cell and attention map generation process to capture the spatial layout.

(2) To explicitly capture layered motion dependencies of video streams, a hierarchical two layer LSTM model is built for action recognition.

(3) The proposed model is tested on three widely applied datasets, the UCF sports dataset [136], the Olympic dataset [137] and the HMDB51 dataset [138] with improved results on other published work.

4.2.2 Soft attention Model for Video Action Recognition

Convolutional Soft Attention Model

LSTM was proposed by Hochreiter, et al. [45] in 1997 and have subsequently been refined. LSTM can avoid the gradient vanishing problem and implements long-term memory by incorporating memory units that allow the network to learn when to forget previous hidden states and when to update hidden states. The input, forget, and output gates are composed of a sigmoid activation layer and matrix multiplication to define how much information flow should be passed to the next time-step. All the parameters in the gates can be learnt in the training process.

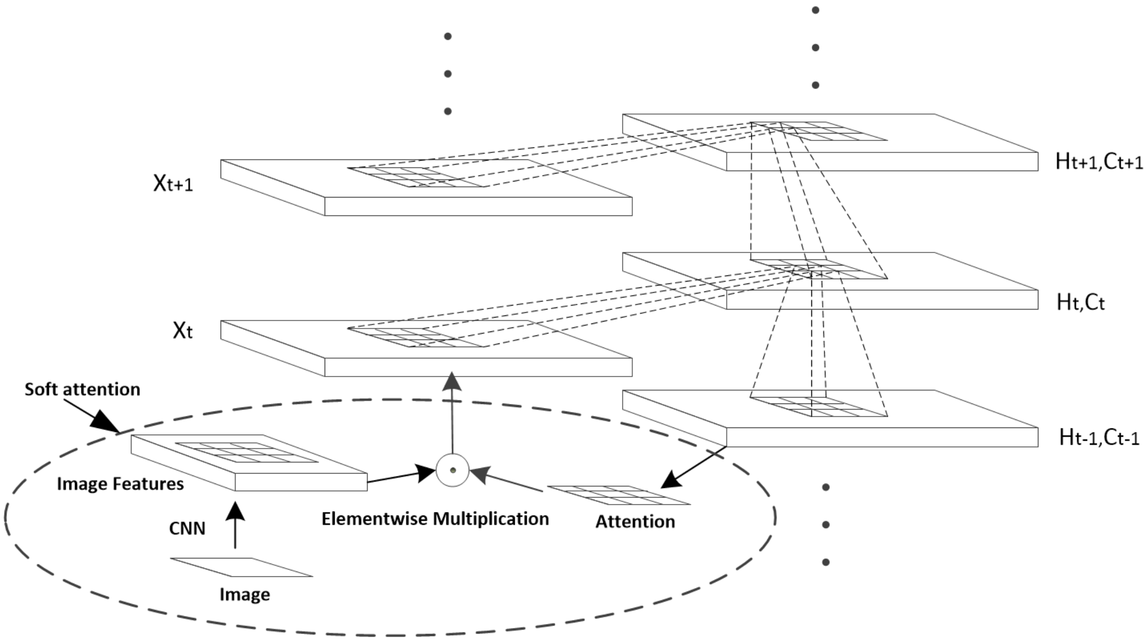


Figure 4.1: The convolutional soft attention mechanism.

Following the idea of [134], the state-to-state transitions in LSTM are replaced with convolutional operations which are illustrated in Fig. 4.1. In Fig. 4.1, the dashed lines indicate the convolution operations, all the input-to-state and state-to-state transitions are replaced with convolutions. Moreover, the attention map is derived from the hidden layer of the LSTM also using convolutional operations. The attention map will be elementwise multiplied with image features to select the most informative regions to focus on.

Our soft attention model is built upon deep CNN features. The features were extracted from the last convolutional layer from a CNN model trained on the ILSVRC-2012 [36] database. The last convolutional features would have the shape of $K \times K \times D$. The features are considered as K^2 number of D feature vectors in which each of the feature vectors represents overlapping receptive fields in the input image, and our soft attention model choose to focus on different regions in each time step.

Letting $\sigma(x) = (1 + e^{-x})^{-1}$ be the sigmoid non-linear activation function and $\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = 2\sigma(2x) - 1$ be the tangent non-linear activation function, the convolutional LSTM model with soft attention follows these updating rules:

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + b_i) \tag{4.1}$$

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + b_f) \tag{4.2}$$

$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + b_o) \tag{4.3}$$

$$g_t = \sigma(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \tag{4.4}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t \tag{4.5}$$

$$h_t = o_t \cdot \phi(c_t) \tag{4.6}$$

Here, i_t, f_t, o_t are the input, forget and output gates of the LSTM model, respectively. They are calculated according to Equations 4.1 - 4.3. c_t is the cell memory while h_t is the hidden state of the LSTM model. A $*$ indicates the convolution operation. $W_\sim, b_\sim$ are convolutional weights and bias, respectively. The multiplication operations are all elementwise multiplication. x_t is the input to the LSTM model at each time step. It can capture the attention information given image features and the hidden state of LSTM from the last time step. Assuming F_t is the frame level image features which are $K \times K \times D$ dimension, x_t, the attention map on image features, can be computed as follows:

$$x_t = l_t^{ij} \cdot F_t \tag{4.7}$$

$$l_t^{ij} = SOFTMAX(W_z * \phi(W_{ha} * h_{t-1} + W_{xa} * x_t + b_a)) \tag{4.8}$$

l_t^{ij} indicates the attention value of each region which is dependent on the hidden state of the last time step and the input image features of this time step. i, j means the horizontal and vertical position of the attention map, respectively. This is achieved by simple weighting of the image features with attention values to preserve the spatial information instead of getting the expectation of image features as in [4]. This is essentially a type of amplification of the 'attention' location of features for the classification at hand. In practice, the hidden state of the last time step and input features are convolved by maps W_{ha} and W_{xa} respectively before passing to a softmax activation layer as in Equation 4.8. The softmax values can be considered as the importance of each region in the image features for the model to pay attention.

Finally, the model applied the cross-entropy loss for action classification.

$$LOSS = -\sum_{t=1}^{T} \sum_{i=1}^{C} y_{t,i} \log(\hat{y}_{t,i}) \tag{4.9}$$

where y_t is the label vector, $\hat{y}_t$ is the classification probabilities at time step t. T is the number of time steps and C is the number of action categories.

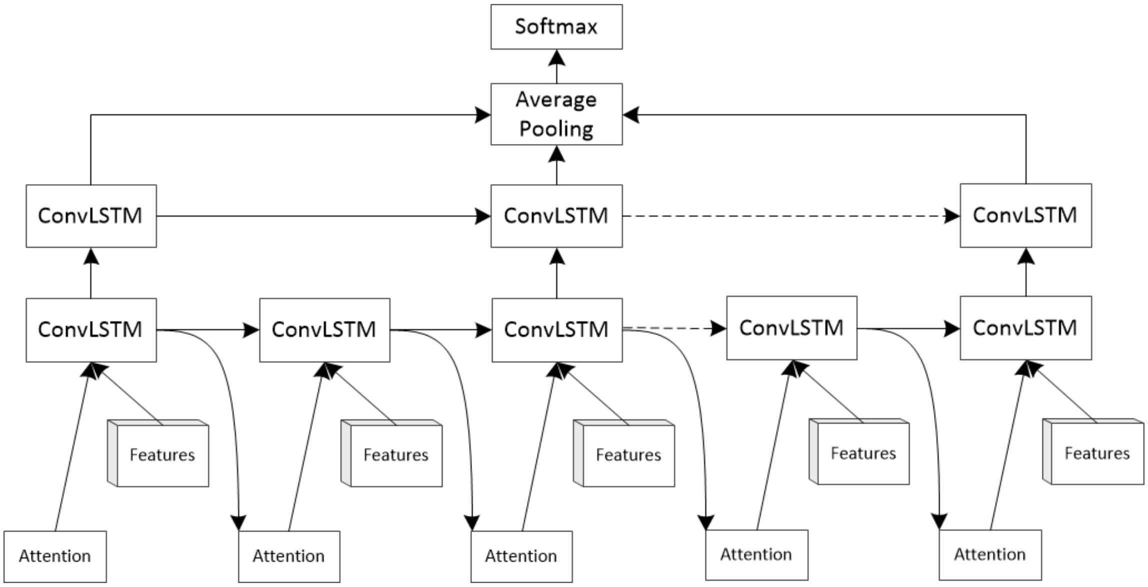


Figure 4.2: The system architecture of the hierarchical model.

Hierarchical Architecture

As previously introduced, the hierarchical architecture of our CHAM is to capture layered motion dependencies. Fig. 4.2 illustrates the system structure of our hierarchical model. The first layer is the attention layer and is also able to reason on the more fine-grained properties of the temporal dependency. The second layer directly connects with first layer but skips several steps in order to catch the coarse granularity of the motion information. Then the output features of the first layer and second layer are concatenated before forwarding to the fully connected layers and an average pooling layer. Then a softmax classifier is connected to generate the results.

4.2.3 Experiments

Datasets Introduction

The approach was evaluated on three datasets, namely the UCF sports [136], the Olympic sports [137] and the more difficult HMDB51 [138]. The UCF sports dataset contains actions collected from various sports on broadcast channels such as ESPN and the BBC. This dataset consists of 150 videos and with 10 different action categories present. The Olympic sports dataset was collected from YouTube sequences [137] and contains 16 different sports categories with 50 sequences per class. The full name of HMDB51 is Human Motion Database and it provides three train-test splits each consisting of 5100 videos. These clips are labelled with 51 action categories. The training set for each split has 3570 videos and the test set has 1530 videos.

For the UCF sports dataset, the dataset are manually divided into a training and a testing set. 75% are used for training, and 25% for testing. The frame-level accuracy are then reported based on the testing dataset.

For the Olympic sports dataset, the original training-testing split is used with 649 clips for training and 134 clips for testing. Following [137], the AP of each category is evaluated on this dataset.

When evaluating our methods on HMDB51, the original training-testing split is applied and the accuracy of each split is tested. As [65] has the results of the conventional soft attention scheme, only the performance of our methodologies are tested.

Implementation Details

Firstly, the frame-level CNN features are extracted using MatConvNet [139] based on Residual-152 Networks[129] trained on the ILSVRC-2012 [36] dataset. The images were resized to 224×224, hence the dimension of each frame-level features is 7×7×2048.

Then CHAM was built using the Theano [140] platform. A convolutional kernel size of 3×3 is used for state-to-state transition in LSTM and a 1×1 convolutional kernel is used for attention map generation to capture spatial information of the CNN features. When the kernel size is 3×3, to ensure the states of LSTM in different time step have the same number of columns and rows as inputs, padding is needed before the convolution operation starts. All these convolutional kernels have 512 channels. A dropout is also applied on the

Table 4.1: Accuracy on UCF sports

Methods	Accuracy
FC-Attention [65]	70%
Conv-Attention(Ours)	72%
CHAM(Ours)	**74%**

output before being fed to the final softmax classifier with a ratio of 0.5.

Also, to carry out comparative studies, a Convolutional Attention Model (Conv-Attention) using only one layer of the convolutional LSTM was built. The Fully Connected Attention Model (FC-Attention) based soft attention [65] was also implemented as a baseline approach. The matrix dimension of state-to-state transition in the fully connected LSTM is set as 512. The soft attention mechanism followed the settings in [65]. All the experiments were conducted using an NVIDIA TITAN X.

For the network training, a mini-batch size of 64 samples is applied at each iteration. For each video clip, the FC-Attention and Conv-Attention networks randomly selected 30 frames for training while CHAM selected 60 frames for training with a second LSTM layer skip every 2 time steps. The backpropagation algorithm are applied through time and an Adam optimiser [141] with a learning rate of 0.0001 is utilized to train the networks. The learning rate was changed to 0.00001 after 10,000 iterations.

Results and Discussion

The results of the UCF sports dataset can be seen in Table 4.8. The Conv-Attention which apply convolutional LSTM for soft attention achieves 72% accuracy on the UCF sports dataset while FC-attention has 70% accuracy. CHAM has the highest accuracy of 74% which indicates that the hierarchical architecture can further improve on the system performance.

The AP value of our methods are then recorded on the Olympics sports dataset as shown in Table 4.9. The Conv-Attention method has a mean AP value of 75.5% which is

Table 4.2: AP on Olympics sports

Class	Vault	Triple Jump	Tennis serve	Spring board	Snatch
FC-Attention [65]	97.0%	88.4%	52.3%	60.0%	23.2%
Conv-Attention (Ours)	97.0%	94.0%	49.8%	66.4%	26.1 %
CHAM (Ours)	97.0%	98.9%	49.5%	69.2%	47.8%
Shot put	Pole vault	Platform 10m	Long jump	Javelin Throw	High jump
67.4%	69.8%	84.1%	100.0%	89.6%	84.4%
60.0%	100.0 %	86.0%	98.0%	87.9%	80.0%
79.8%	60.8%	89.7%	100%	95.0%	78.7%
Hammer throw	Discus throw	Clean and jerk	Bowling	Basketball layup	mAP
38.0%	100.0%	76.0%	60.0%	89.8%	73.7%
36.6%	97.8%	100.0%	46.8%	81.2%	75.5%
37.9%	97.0%	84.8%	46.7%	89.1%	**76.4%**

Table 4.3: Accuracy on HMDB51

Methods	Accuracy
FC-Attention [65]	41.3%
Conv-Attention (Ours)	42.2%
CHAM (Ours)	**43.4%**

Table 4.4: Comparison with related methods on HMDB51

Methods	Accuracy	Spatial Image Only	Fine-tuning
Softmax Rgression [65]	33.5%	Yes	No
Spatial Convolutional Net [90]	40.5%	Yes	Yes
Trajectory-based modeling [142]	40.7%	No	No
Average pooled LSTM [65]	40.5%	Yes	No
FC-Attention [65]	41.3%	Yes	No
ConvALSTM [143]	43.3%	Yes	Yes
CHAM (Ours)	**43.4%**	Yes	No

higher than the FC-attention performance (73.7%). Similarly, the improvement brought by the hierarchical architecture is also validated on this dataset, with a 76.4% mean AP value achieved by the proposed CHAM model. The hierarchical model is especially good at long-term action categories, for instance, 'Snatch' and 'Javelin Throw' on which the CHAM method leads the other approaches by a large margin.

The results on the HMDB51 dataset can be seen in Table 4.11. Similar observations can be made: the Conv-Attention has a higher accuracy value of 42.2%, and the CHAM

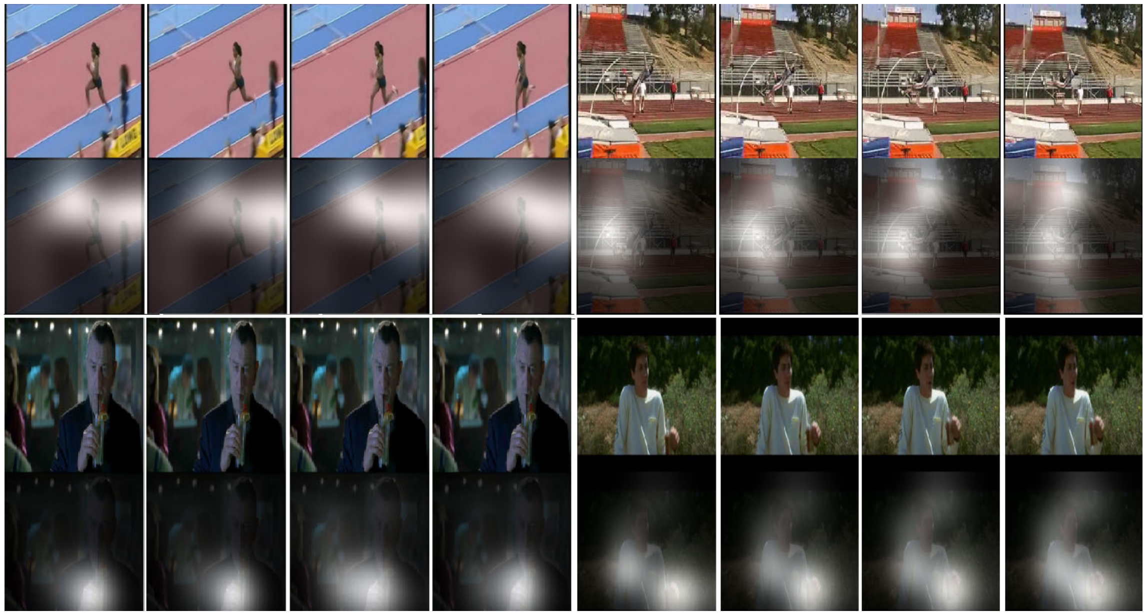


Figure 4.3: Visualization of the attention mechanism.

added another 1.2% gain to the final result, which is 43.4%.

Table 4.12 shows the comparison results on the HMDB51 dataset. From the table, the following observations can be made:

(1) Our CHAM method outperformed most of the previous methods which are only based on spatial image features.

(2) Even though our CNN model was not fine-tuned, the results remain competitive compared with many approaches which had applied fine-tuning.

(3) The proposed model shows good potential to achieve better results. Future work can be undertaken by fine-tuning the CNN model on a specific dataset.

Fig.4.3 provides some examples of visualisation of the learned attention region; the regions of a person are brighter which means they are the attention region learned automatically.

4.2.4 Conclusion

In this chapter, a novel model: CHAM, is proposed. This is achieved by applying convolutional LSTM, a novel RNN model, for the implementation of a soft attention mechanism and a hierarchical system architecture for action recognition. The convolutional LSTM can catch the spatial layout of the CNN features while the hierarchical system architecture can fuse information on the temporal dependencies from multiple granularities of the dataset. Finally, the CHAM method was tested on three widely used datasets, the UCF sports dataset, the Olympic sports dataset and the HMDB51 dataset, with improved results.

4.3 Hierarchical Multi-scale Attention Networks for Action Recognition

4.3.1 Introduction

Action recognition in videos is a fundamental task in computer vision. Recently, with the rapid development of deep learning, and in particular, deep CNNs, a number of models [18] [42] [38] [43] have been proposed for image recognition. However, for video-based action recognition, a model should accept inputs with variable length and generate the corresponding outputs. This special requirement makes the conventional CNN model that caters for a one-versus-all classification unsuitable.

For decades RNNs have been applied to sequence-to-sequence applications, often with good results. However, a significant limitation of the vanilla RNN models, which strictly integrate state information over time, is the vanishing gradient effect [44]: the ability to back propagate an error signal through a long-range temporal interval becomes increasingly impossible in practice. To mitigate this problem, a class of models with a long-range dependencies learning capability, called LSTMs, was introduced by Hochreiter and Schmidhuber [45]. Specifically, LSTM consists of memory cells, with each cell containing units to learn when to forget previous hidden states and when to update hidden states with new information.

Much sequential data often have a complex temporal structure which requires both hierarchical and multi-scale information to be modelled properly. In language modelling, a long sentence is often composed of many phrases which further can be decomposed into words. Meanwhile, in action recognition, an action category can be described by many sub-actions. For instance, ‘long jump’ contains ‘running’, ‘jumping’ and ‘landing’. As stated in [144], a promising approach to model such hierarchical representation is the multi-scale RNN. One popular approach of implementing multi-scale RNNs is to treat the hierarchical timescales as pre-defined parameters. For example, Wang et al. [135] implemented a multi-scale architecture by building multiple layers LSTM in which higher layers skip several time steps. In their paper, the skipped number of time steps is the parameter to be pre-defined. However, it is often impractical to pre-define such timescales without learning, which also leads to a poor generalisation capability. Chung et al. [144] proposed a novel RNN structure, Hierarchical Multi-scale Recurrent Neural Network (HM-RNN), to automatically learn time boundaries from data. These temporal boundaries are similar to rules described by discrete variables inside RNN cells. Normally, it is difficult to implement training algorithms for discrete variables. Popular approaches include unbiased estimator with the aid of REINFORCE [145]. In this chapter, the HM-RNN is re-implemented by applying the recently proposed Gumbel-sigmoid function [146] [147] to realise the training of stochastic neurons due to its efficiency [148].

In the general RNN framework for sequence-to-sequence problems, the input information is treated uniformly without discrimination on the different parts. This will result in the fixed length of intermediate features and hence subsequent sub-optimal system performance. The practice is in sharp contrast to the way humans accomplish sequence processing tasks. Humans tend to selectively concentrate on the part of the information and at the same time ignores other perceivable information. The mechanism of selectively focusing on relevant contents in the representation is called attention. The attention based

RNN model in machine learning was successfully applied in NLP, and more specifically, in neural translation [8]. For many visual recognition tasks, different portions of an image or segments of a video have different importance, which should be selectively weighted with attention. Xu et al. [4] systematically analysed stochastic hard attention and deterministic soft attention models and applied them in image captioning tasks, with improved results compared with other RNN-like algorithms. The hard attention mechanism requires a stochastic neuron which is hard to train using the conventional backpropagation algorithm. They applied REINFORCE [145] as an estimator to implement hard attention to image captioning.

The REINFORCE is an unbiased gradient estimator for stochastic units. However, it is very complex to implement and often has high gradient variance during training [148]. In this chapter, the applicability of Gumbel-softmax [146] [147] is studied in hard attention because Gumbel-softmax is an efficient way to estimate discrete units during the training of neural networks. To mitigate the problem of temperature sensitivity in Gumbel-softmax, an adaptive temperature scheme is applied [148] in which the temperature value is also learnt from the data. The experimental results verify that the adaptive temperature is a convenient way to avoid manual searching for the parameter. Additionally, the deterministic soft attention [4] [65] and stochastic hard attention implemented by REINFORCE-like algorithms [63] [64] [4] are also tested in action recognition. Combined with HM-RNN and the two types of attention models, the proposed Hierarchical Multi-scale Attention Networks (HM-AN) are systematically evaluated for action recognition in videos, with improved results.

Our main contributions can be summarised as follows:

- A HM-AN by implementing HM-RNN with Gumbel-sigmoid is proposed to realise the discrete boundary detectors.
- Four methods of realising an attention mechanism, are proposed for action recognition in videos, with improved results over many baselines.

- By incorporating Gumbel-softmax and Gumbel-sigmoid into HM-RNN, the stochastic neurons in the networks are made to be end-to-end trainable by error backpropagation.
- For the hard attention model based on Gumbel-softmax, an adaptive temperature for the Gumbel-softmax is proposed, which generates much-improved results over a constant temperature value.
- Through visualisation of the learnt attention regions, the boundary detectors of HM-AN and the adaptive temperature values, insights for further research can be provided.

4.3.2 Related Works

Hierarchical RNNs

The modelling of hierarchical temporal information has long been an important topic in many research areas. The most notable model is LSTM proposed by Hochreiter and Schmidhuber [45]. LSTM employs the multi-scale updating concept, where the hidden units' update can be controlled by gating such as input gates or forget gates. This mechanism enables the LSTM to deal with long-term dependencies in the temporal domain. Despite this advantage, the maximum time steps are limited to within a few hundred because of the leaky integration which makes the memory for long-term gradually diluted [144]. The maximum time steps in video processing are several dozen frames which makes the application of LSTM in video recognition very challenging.

To alleviate this problem, many researchers tried to build a hierarchical structure explicitly, for instance, HAN proposed in [135], which is implemented by skipping several time steps in the higher layers of the stacked multi-layer LSTMs. However, the number of time steps to be skipped is a pre-defined parameter. How to choose these parameters and why to choose a certain number are unclear.

More recent models like clockwork RNN [149] partitioned the hidden states of an RNN into several modules with different timescales assigned to them. The clockwork RNN is more computationally efficient than the standard RNN as the hidden states are updated only at the assigned time steps. However, finding the suitable timescales is challenging which makes the model less applicable.

To mitigate the problem, Chung et al. [144] proposed the HM-RNN. The HM-RNN can learn the temporal boundaries from data, which allows the RNN model to build a hierarchical structure and enables long-term dependencies automatically. However, the temporal boundaries are stochastic discrete variables which are very hard to train using the standard backpropagation algorithm.

A popular approach to train the discrete neurons is the REINFORCE-like [70] algorithms. This is an unbiased estimator but often with high gradient variance [144]. The original HM-RNN applied a straight-through estimator [145] because of its efficiency and simplicity in implementation. Instead, in this chapter, the more recent Gumbel-sigmoid [146] [147] are applied to estimate the stochastic neurons. This is much more efficient than other approaches and achieved state-of-the-art performance among many other gradient estimators [146].

Attention Mechanism

One important property of human perception is that humans do not tend to process a whole scene, in its entirety, at once. Instead, humans pay attention selectively on parts of the visual scene to acquire information where it is needed [63]. Different attention models have been proposed and applied in object recognition and machine translation. Mnih et al. [63] proposed an attention mechanism to represent static images, videos or as an agent that interacts with a dynamic visual environment. Also, Ba et al. [64] presented an attention-based model to recognise multiple objects in images. These two models are all with the aid of REINFORCE-like algorithms.

The soft attention model was proposed for the machine translation problem in NLP [8], and Xu et al. [4] extended it to image caption generation as the task is analogous to ‘translating’ an image into a sentence. Specifically, they built a stochastic hard attention model with the aid of REINFORCE and a deterministic soft attention model. The two attention mechanisms were applied to the image captioning task, with excellent results. Subsequently, Sharma et al. [65] built a similar model with soft attention applied to action recognition from videos.

There are some subsequent works on the attention mechanism. For instance, in [66], the attention model is utilised for video description generation by softly weighting the visual features extracted from the frames in each video. Li et al. [143] combined a convolutional LSTM [134] with the soft attention mechanism for video action recognition and detection. Teh et al. [76] extended the soft attention into CNN networks for weakly supervised object detection.

One important reason for applying soft attention instead of the hard version is that the stochastic hard attention mechanism is difficult to train. Although the REINFORCE-like algorithms [70] are unbiased estimators to train stochastic units, their gradients have high variants. To solve this problem, recently, Jang et al. [146] proposed a novel categorical reparameterization technique using the Gumbel-softmax distribution. The Gumbel-softmax is a superior estimator for categorical discrete units [146]. It has been proved to be efficient and has high-performance [146].

Action Recognition

Action recognition has received significant attention recently. Most approaches focused on the design of novel features, such as trajectory-based features [130], CNN based features [150] [90] [151]. For example, [152] built a simple representation to explicitly model the motion relationships, with outstanding results based on popular classifiers like Support Vector Machine (SVM) on several benchmark datasets.

Some researches built model to better exploit these powerful features by the operation of fusion. For instance, [153] proposed a regularised Deep Neural Network (DNN) to fuse the CNN features, the trajectory features and the audio features for action categorization, with promising results. [90] [151] fused CNN features and motion features for better recognized action categories in video.

RNNs have been popular for speech recognition [154], image caption generation [4], and video description generation [66]. There have also been efforts made for the application of LSTM RNNs in action recognition. For instance, [133] proposed an end-to-end training system using CNN and RNN deep both in space and time to recognise activities in video. [155] also explicitly models the video as an ordered sequence of frames using LSTM. Most of the previous work treat image features extracted from CNNs as static inputs to an RNN to generate action labels at each frame. The attention mechanism can discriminate the relevant features from these static inputs and can improve the system performance. Moreover, the interpretation of CNN features will be much easier if the attention mechanism can be applied to action recognition because the attention mechanism automatically focuses on specific regions to facilitate the classification.

In this chapter, the HM-RNN are re-implemented to capture the hierarchical structure of temporal information from video frames. By incorporating the HM-RNN with both stochastic hard attention and deterministic soft attention, the long-term dependencies of video frames can be captured.

Research related to ours also includes the attention model proposed by Xu et al. [4] and [156]. [4] first applied both stochastic hard attention and deterministic soft attention mechanisms for spatial locations of images for image captioning. [156] instead used weighting on image patches to implement region-level attention. In this chapter, similar to [4], both stochastic hard attention and deterministic soft attention are studied. However, when implementing hard attention, [4] borrowed the idea of REINFORCE while the more recent Gumbel-softmax is proposed to estimate discrete neurons in the attention mechanism.

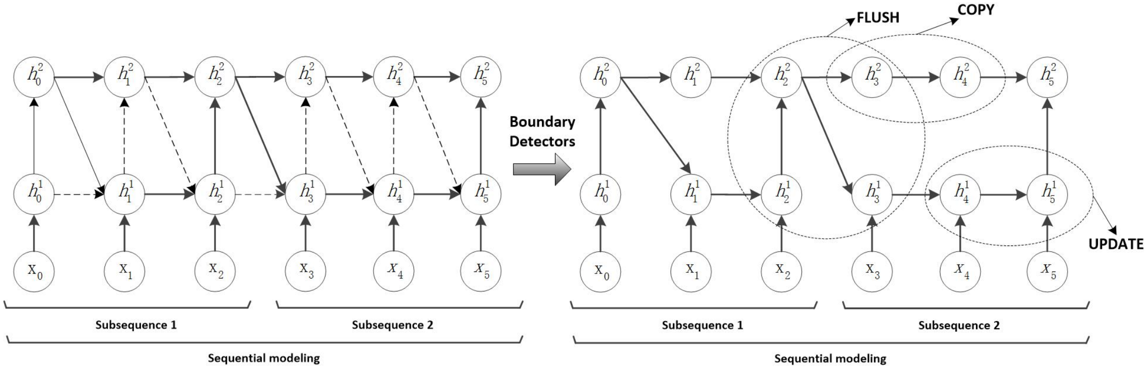


Figure 4.4: Network Structure of the HM-AN.

4.3.3 The proposed methods

In this section, the HM-RNN structure proposed in [144] are first re-visited, then the proposed HM-AN networks are introduced, with details of Gumbel-softmax and Gumbel-sigmoid to estimate the stochastic discrete neurons in the networks.

HM-RNN

HM-RNN was proposed in [144] to better capture the hierarchical multi-scale temporal structure in sequence modelling. HM-RNN defines three operations depending on the boundary detectors: UPDATE, COPY and FLUSH. The selection of these operations is determined by the boundary states z_t^{l-1} and z_{t-1}^l, where l and t represent the current layer and time step, respectively:

$$\begin{aligned} &UPDATE, && z_{t-1}^l = 0 \text{ and } z_t^{l-1} = 1; \\ &COPY, && z_{t-1}^l = 0 \text{ and } z_t^{l-1} = 0; \\ &FLUSH, && z_{t-1}^l = 1. \end{aligned} \tag{4.10}$$

The updating rules for the operation UPDATE, COPY and FLUSH are defined as follows:

$$c_t^l = \begin{cases} f_t^l \odot c_{t-1}^l + i_t^l \odot g_t^l, & UPDATE \\ c_{t-1}^l, & COPY \\ i_t^l \odot g_t^l, & FLUSH \end{cases} \tag{4.11}$$

The updating rules for hidden states are also determined by the pre-defined operations:

$$h_t^l = \begin{cases} h_{t-1}^l, & COPY \\ o_t^l \odot c_t^l, & otherwise \end{cases} \tag{4.12}$$

The (i, f, o) indicate the input, forget and output gate, respectively. g is called the 'cell proposal' vector. One of the advantages of HM-RNN is that the updating operation (UPDATE) is only executed at specific time steps instead of all the time, which significantly reduces the computation cost.

The COPY operation copies the cell memory and hidden state from the previous time step to the current time step in the upper layers until the end of a subsequence, as shown in Fig. 4.4. Hence, the upper layer can capture coarser temporal information. Also, the boundaries of subsequence are learnt from the data which is a big improvement over other related models. To start a new subsequence, the FLUSH operation needs to be executed. The FLUSH operation firstly forces the summarised information from the lower layers to be merged with the upper layers, then re-initialise the cell memories for the next subsequence.

In summary, the COPY and UPDATE operations enable the upper and lower layers to capture information on different time scales, thus realising a multi-scale and hierarchical structure for a single subsequence. The FLUSH operation can summarise the information from the last subsequence and forward them to the next subsequence, which guarantees the connection and coherence between parts within a long sequence.

The values of gates (i, f, o, g) and the boundary detector z are obtained by:

$$\begin{pmatrix} i_t^l \\ f_t^l \\ o_t^l \\ g_t^l \\ z_t^l \end{pmatrix} = \begin{pmatrix} sigm \\ sigm \\ sigm \\ tanh \\ hardsigm \end{pmatrix} f_{slice} \begin{pmatrix} s_t^{recurrent(l)} + \\ s_t^{top-down(l)} + \\ s_t^{bottom-up(l)} + \\ b_l \end{pmatrix} \tag{4.13}$$

where

$$s_t^{recurrent(l)} = U_l^l h_{t-1}^l \tag{4.14}$$

$$s_t^{top-down(l)} = U_{l+1}^l (z_{t-1}^l \odot h_{t-1}^{l+1}) \tag{4.15}$$

$$s_t^{bottom-up(l)} = W_{l-1}^l (z_t^{l-1} \odot h_t^{l+1}) \tag{4.16}$$

and the hardsigm is estimated using the Gumbel-sigmoid which will be explained later. In the equation, the U_l and W_l are the weight matrices, and b_l is the bias matrix.

HM-AN

A RNN-based framework can tackle the sequential problems inherent in action recognition and image captioning in computer vision. As previously explained, HM-RNN can learn the hierarchical temporal structure from data and enable long-term dependencies. This inspired our proposal for the HM-AN model.

As attention has been proved very useful in action recognition [65], in HM-AN, to capture the implicit relationships between the inputs and outputs in sequence to sequence problems, both hard and soft attention mechanisms are applied to explicitly learn the critical and relevant image features regarding the specific outputs. A more detailed explanation is as follows.

Estimation of Boundary Detectors In the proposed HM-AN, the boundary detectors z_t are estimated with Gumbel-sigmoid, which is derived directly from the Gumbel-softmax proposed in [146] and [147].

The Gumbel-softmax replaces the argmax in the Gumbel-Max Trick [157] [158] with the following Softmax function:

$$y_i = \frac{exp(\log(\pi_i + g_i)/\tau)}{\sum_{j=1}^{k} exp(\log(\pi_j + g_j)/\tau)} \tag{4.17}$$

where $g_1, ..., g_k$ are *i.i.d.* sampled from the distribution Gumbel (0,1), and τ is the temperature parameter. k indicates the dimension of the generated Softmax vector.

To derive the Gumbel-sigmoid, the Sigmoid function can be re-written as a Softmax of two variables: π_i and 0.

$$\begin{aligned} sigm(\pi_i) &= \frac{1}{(1 + exp(-\pi_i))} = \frac{1}{(1 + exp(0 - \pi_i))} \\ &= \frac{1}{1 + exp(0)/exp(\pi_i)} = \frac{exp(\pi_i)}{(exp(\pi_i) + exp(0))} \end{aligned} \tag{4.18}$$

Hence, the Gumbel-sigmoid can be written as:

$$y_i = \frac{exp(\log(\pi_i + g_i/\tau)}{exp(\log(\pi_i + g_i)/\tau) + exp(\log(g')/\tau)} \tag{4.19}$$

where g_i and g' are independently sampled from the distribution Gumbel (0,1).

To obtain a discrete value, values of $z_t = \widetilde{y_i}$ are set as:

$$\widetilde{y_i} = \begin{cases} 1 & y_i \geq 0.5 \\ 0 & otherwise \end{cases} \tag{4.20}$$

In our experiments, all the boundary detectors z_t are estimated using the Gumbel-sigmoid with a constant temperature of 0.3.

Deterministic Soft Attention To implement soft attention over image regions for the action recognition task, a similar strategy is applied to the soft attention mechanism in

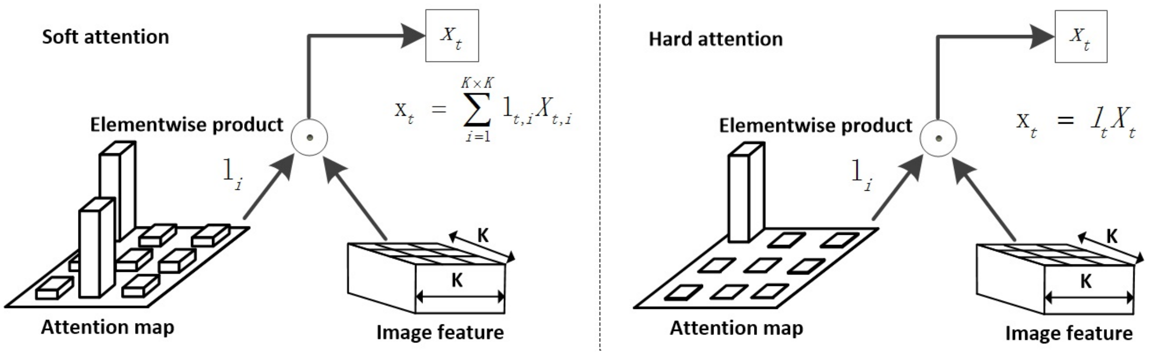


Figure 4.5: The soft attention and hard attention mechanism.

[65] and [4].

Specifically, the model predicts a Softmax over K×K image locations. The location Softmax is defined as:

$$l_{t,i} = \frac{exp(W_i h_{t-1})}{\sum_{j=1}^{K\times K} exp(W_j h_{t-1})} \qquad i = 1, ..., K^2 \tag{4.21}$$

where i means the ith location corresponding to the specific regions in the original image.

This Softmax can be considered as the probability with which the model learns the specific regions in the image, which is important for the task at hand. Once these probabilities are obtained, the model computes the expected values over image features at different regions:

$$x_t = \sum_{i=1}^{K^2} l_{t,i} X_{t,i} \tag{4.22}$$

Where x_t is considered as inputs of the HM-AN networks. In our HM-AN implementations, the hidden states used to determine the region softmax is defined for the first layer, i.e., h_{t-1}^1. The upper layers will automatically learn the abstract information of input features as previously explained. The soft attention mechanism can be visualised in the left side of Fig. 4.5.

Stochastic Hard Attention

REINFORCE-like algorithm Stochastic hard attention was proposed in [4]. Their hard attention was realised with the aid of a REINFORCE-like algorithm. In this section, this kind of hard attention mechanism is also introduced.

The location variable l_t indicates where the model decides to focus attention on the t^{th} frame of a video. $l_{t,i}$ is an indicator of a one-hot representation which can be set to 1 if the i^{th} location contains a relevant feature.

Specifically, a hard attentive location of $\{\alpha_i\}$ is defined:

$$p(l_{i,t} = 1 | l_{j<t,a}) = argmax(\alpha_{t,i}) \quad = argmax\left(\frac{exp(W_i h_{t-1})}{\sum_{j=1}^{K\times K} exp(W_j h_{t-1})}\right) \tag{4.23}$$

where a represents the input image features.

An objective function L_l can be defined as a variational lower bound on the marginal log-likelihood log $p(y|a)$ of observing the action label y given image features a. Hence, L_l can be represented as:

$$L_l = \sum_l p(l|a) \log\ p(y|l,a) \quad \leq \log\ \sum_l p(l|a)p(y|l,a) = \log p(y|a) \tag{4.24}$$

$$\frac{\partial L_l}{\partial W} = \sum_l p(l|a)[\frac{\partial \log\ p(y|l,a)}{\partial W} + \log\ p(y|l,a)\frac{\partial \log\ p(l|a)}{\partial W}] \tag{4.25}$$

Ideally, the gradients of Equation 4.25 is what need to be computed. However, it is not feasible to compute the gradient of expectation in Equation 4.25. Hence, a Monte Carlo approximation technique is applied to estimate the gradient of the operation of expectation.

Therefore, the derivatives of the objective function with respect to the network parameters can be expressed as:

$$\frac{\partial L_l}{\partial W} = \frac{1}{N}\sum_{n=1}^{N}[\frac{\partial \log\ p(y|\tilde{l}_n, a)}{\partial W} + \log\ p(y|\tilde{l}_n, a)\frac{\partial \log\ p(\tilde{l}_n|a)}{\partial W}] \tag{4.26}$$

where $\tilde{l}$ is obtained based on the argmax operation as in Equation 4.23.

Similar to the approaches in [4], a variance reduction technique is used. With the k^{th} mini-batch, the moving average baseline is estimated as an accumulation of the previous log-likelihoods with exponential decay:

$$b_k = 0.9 \times b_{k-1} + 0.1 \times \log\ p(y|\tilde{l}_k, a) \tag{4.27}$$

The learning rule for this hard attention mechanism is defined as follows:

$$\frac{\partial L_l}{\partial W} \approx \frac{1}{N}\sum_{n=1}^{N}[\frac{\partial \log\ p(y|\tilde{l}_n, a)}{\partial W} + \lambda(\log\ p(y|\tilde{l}_n, a) - b)\frac{\partial \log\ p(\tilde{l}_n|a)}{\partial W}] \tag{4.28}$$

where λ is a pre-defined parameter.

As pointed out in Ba et al. [64], Mnih et al. [63] and Xu et al. [4], this is a formulation which is equivalent to the REINFORCE learning rule [70]. For convenience, it is abbreviated as REINFORCE-Hard Attention in the following sections.

Gumbel Softmax In the hard attention mechanism, the model selects one crucial region instead of taking the expectation. Hence, it is a discrete stochastic unit which cannot be trained using backpropagation. [4] applied the REINFORCE to estimate the gradient of the stochastic neuron. Although REINFORCE is an unbiased estimator, the variance of the gradient is large and the algorithm is complex to implement. To solve these problems, the Gumbel-softmax is proposed to apply to estimate the gradient of the discrete units in our model. Gumbel-softmax is better than REINFORCE and much easier to implement [146].

The Softmax can be replaced with Gumbel-softmax in Equation 4.21 and remove the process of taking expectation to realise the hard attention.

$$l_{t,i} = \frac{exp(\log(W_i h_{t-1} + g_i)/\tau)}{\sum_{j=1}^{K\times K} exp(\log(W_j h_{t-1} + g_j)/\tau)} \qquad i = 1...K^2 \tag{4.29}$$

The Gumbel-softmax will choose a single location indicating the most important image region for the task. However, the search space for the temperature parameter is too large to be manually selected. The temperature is a sensitive parameter as explained in [146]. Hence in this chapter, an adaptive temperature is applied as in [148]. The adaptive temperature determines the value depending on the current hidden states. In other words, instead of being treated as a pre-defined parameter, the value of temperature is learnt from the data. Specifically, the following mechanism is used to determine the temperature:

$$\tau = \frac{1}{Softplus(W_{temp} h_t^1 + b_{temp}) + 1} \tag{4.30}$$

Where h_t^1 is the hidden state of the first layer of our HM-AN. Equation 4.30 generates a scalar for the temperature. In the equation, adding one can enable the temperature to fall within the scope of 0 and 1. The hard attention mechanism can be seen on the right-hand side of Fig. 4.5.

Application of HM-AN in Action Recognition

The proposed HM-AN can be directly applied in video action recognition. In video action recognition, the dynamics exist in the inputs, i.e., the given video frames. With the attention mechanism embedded in RNN, the critical features of each frame can be discovered and discriminated in order to facilitate recognition.

For action recognition, the HM-AN applies the cross-entropy loss for recognition.

$$LOSS = -\sum_{t=1}^{T}\sum_{i=1}^{C} y_{t,i} \log(\hat{y}_{t,i}) \tag{4.31}$$

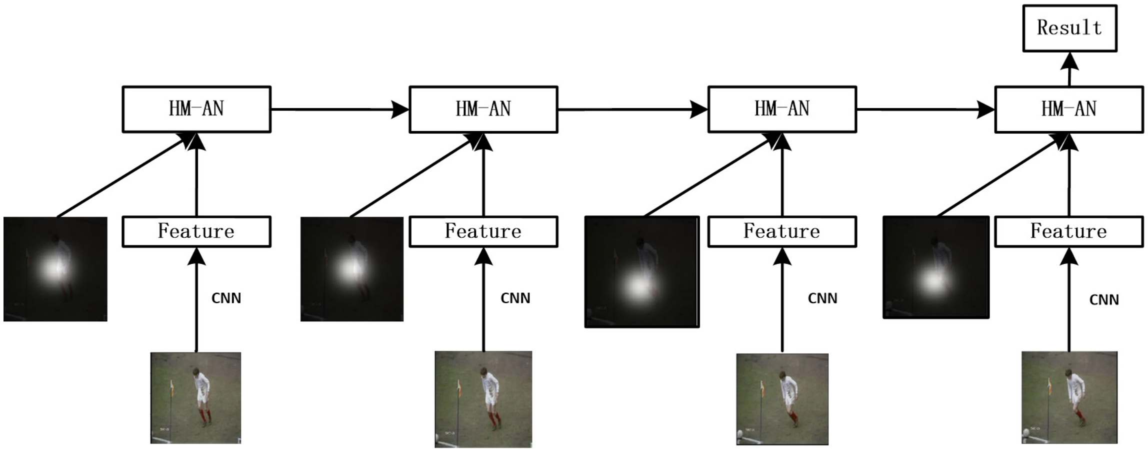


Figure 4.6: Action recognition with HM-AN.

where y_t is the label vector, $\hat{y}_t$ is the classification probabilities at time step t. T is the number of time steps and C is the number of action categories. The system architecture of action recognition using HM-AN is shown in Fig. 4.6.

4.3.4 Experiments

In this section, the implementation details are firstly explained then the experimental results on action recognition are reported.

Implementation Details

The HM-AN are implemented using the Theano platform [159] and all the experiments are conducted on a server embedded with a Titan X GPU. In the experiments, HM-AN is a three layer stacked RNN. The outputs are concatenated by hidden states from three layers and forwarded to a softmax layer.

In addition to the baseline approach (LSTM networks), four versions of HM-AN were implemented for comparison:

- Softmax regression. This is to perform a general image classification task based on spatial features.

- LSTM with soft attention (Baseline). The baseline approach is set as a one layer LSTM networks with the soft attention mechanism.
- Deterministic soft attention in HM-AN (Soft Attention). This is to determine how soft attention mechanism performs with the HM-AN.
- Stochastic hard attention with reinforcement learning in HM-AN (REINFORCE-Hard Attention). This type of hard attention mechanism is described in Section 4.3.3.
- Stochastic hard attention with a 0.3 temperature for Gumbel-softmax in HM-AN (Constant-Gumbel-Hard Attention). A constant temperature is applied in Gumbel-softmax to accomplish the proposed hard attention model.
- Stochastic hard attention with adaptive temperature for Gumbel-softmax in HM-AN (Adaptive-Gumbel-Hard Attention). The temperature is set as a function of the hidden states of RNN.

For the experiments, with the help of the MatConvNet platform [139], the frame-level CNN features are first extracted from the last convolutional layer (res5cx) based on Residue-152 Networks [43] trained on the ILSVRC-2012 [36] dataset. The images were resized to 224×224. Hence the dimension of each frame-level features is 7×7×2048. For the network training, a mini-batch size of 64 samples is applied at each iteration. For each video sequence, the baseline approach randomly selected a sequence of 30 frames for training while the proposed approaches selected a sequence of 60 frames for training in order to verify the proposed HM-AN's capability to capture long-term dependencies. The optimal length for LSTM with attention is 30 and increasing the number will seriously deteriorate the performance. In order to determine the optimal length of sequence feeding into the networks, several trials are performed as described in Section 4.3.4, determining that the

Table 4.5: Networks Structure Configuration.

Input to HM-AN		Size of Inner Units of HM-AN	
Inputs	$7 \times 7 \times 2048$	Hidden Unit Size	2048
Output Layers		Cell Memory Size	2048
1st Layer Outputs	2048	Gate Size (i, f, o, g)	2048
2nd Layer Outputs	2048	Boundary Detector Size	2048
3rd Layer Outputs	2048	Training Parameters	
Concatenation Layer	6144	Dropout	0.5
Fully connected Layer 1	1024	Learning Rate	0.00001
Fully connected Layer 2	Class Categories	Video Sequence Length	60

Table 4.6: Number of Iterations and Epoches for Convergence on Different Datasets.

Dataset	Iterations	Epoches
UCF Sports	400	2
Olympic Sports	2500	2
HMDB51	10000	2

optimal length for the HM-AN is 60. The backpropagation algorithm are applied through time and Adam optimiser [141] with a learning rate of 0.0001 is used train the networks. The learning rate was changed to 0.00001 after 10,000 iterations. At test time, the class predictions for each time step are computed and then those predictions are averaged over 60 frames. Table 4.5 provides a detailed description of the network configuration. Table 4.6 shows the number of iterations and epochs needed for convergence on different datasets.

Experimental Results and Analysis

Datasets The proposed approach are tested on three widely used datasets, namely UCF Sports [136], the Olympic Sports datasets [137] and the more difficult Human Motion Database (HMDB51) dataset [138]. Fig. 4.7 provides some examples of the three datasets used in this chapter. The UCF Sports dataset contains a set of actions collected from various sports which are typically featured on broadcast channels such as ESPN or BBC. This dataset consists of 150 videos with a resolution of 720 × 480 and contains 10 different

(a) UCF Sports dataset

(b) Olympic Sports dataset

(c) HMDB51 dataset

Figure 4.7: Some examples from the datasets used in this chapter.

action categories. The Olympic Sports dataset was collected from YouTube sequences [137] and contains 16 different sports categories with 50 videos per class. Hence, there are a total of 800 videos in this dataset. The HMDB51 dataset is a more difficult dataset which provides three train-test splits each consisting of 5100 videos. These sequences are labelled with 51 action categories. The training set for each split has 3570 videos, and the test set has 1530 videos.

For the UCF Sports dataset, as there is a lack of training-testing split for evaluation, the dataset are manually divided into training and testing sets. 75 per cent are selected for training and the remaining 25 per cent are left for testing. The classification accuracy are then reported on the testing dataset.

As for Olympic Sports dataset, the original training-testing split is used with the 649

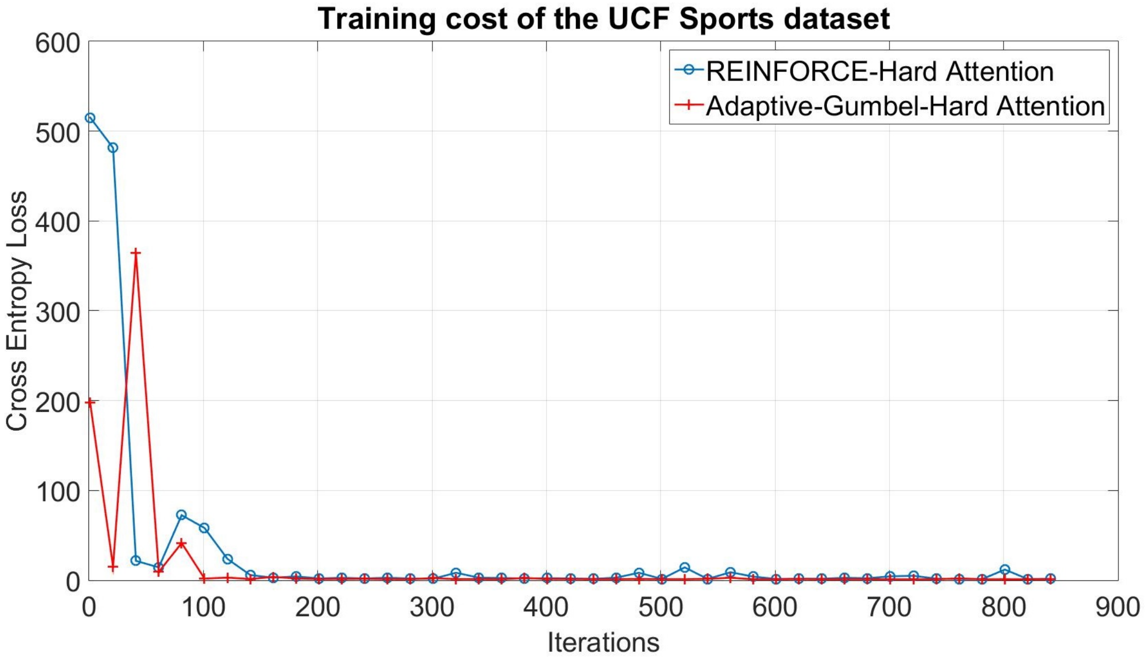


Figure 4.8: Training cost of the UCF Sports dataset.

sequences for training and 134 sequences for testing provided in the dataset. Following the practice in [137], the AP are evaluated for each category on this dataset.

When evaluating the proposed method on HMDB51, the original training-testing split is used and the classification accuracy on the testing set is reported.

Results

UCF Sports dataset The performance of the LSTM with soft attention proposed in [65] are first tested on the UCF Sports dataset and obtained 70.0% accuracy. All the experimental settings were the same as those in [65]. Then the proposed four approaches mentioned previously are evaluated. As described in [65], the optimal sequence length is 30 frames.

One of the expectations of using HM-AN is to enable long-term dependencies. In order

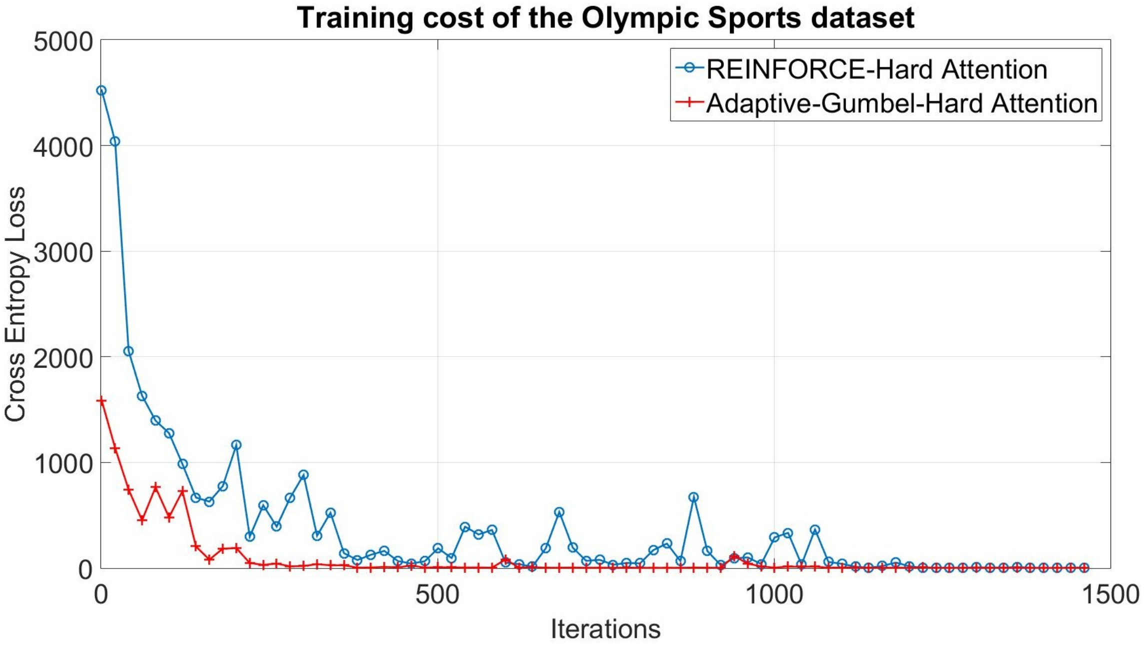


Figure 4.9: Training cost of the Olympic Sports dataset.

to find the optimal length for HM-AN, certain experiments are performed. As shown in Table 4.7, the optimal length of the video sequence is 60 frames. Increasing or decreasing the length would cause a drop in the overall result accuracy.

HM-AN with the stochastic hard attention, which is realised with REINFORCE-like algorithm improves the results to 82.0%. HM-AN with soft attention is similar to the REINFORCE-Hard Attention, with an accuracy of 81.1%. The hard attention mechanism realised by Gumbel-softmax with adaptive temperature achieves 82.0% accuracy, similar to our REINFORCE-Hard Attention model. However, the Constant-Gumbel-Hard Attention which uses Gumbel-softmax with constant temperature value of 0.3 only yields 76.0% accuracy, which indicates the significant role of adaptive temperature in maintaining the system performance. Fig. 4.8 shows the curves of training cost cross-entropy for the Adaptive-Gumbel-Hard Attention approach and REINFORCE-Hard Attention approach, respectively. It can be seen from the figure that the REINFORCE-Hard Attention con-

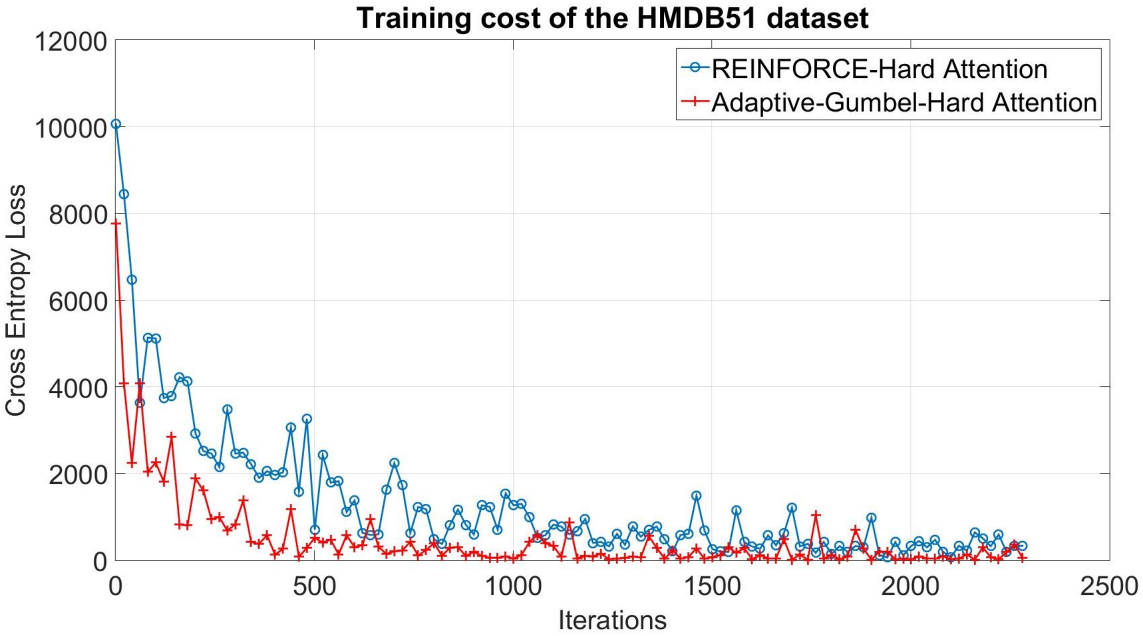


Figure 4.10: Training cost of the HMDB51 dataset.

Table 4.7: Accuracy on UCF Sports using Adaptive-Gumbel-Hard Attention with different sequence lengths.

Sequence Length	Accuracy
30 frames	70.0%
40 frames	74.0%
50 frames	78.0%
60 frames	**82.0%**
70 frames	80.1%

verges marginally slower than the approach of Adaptive-Gumbel-Hard Attention.

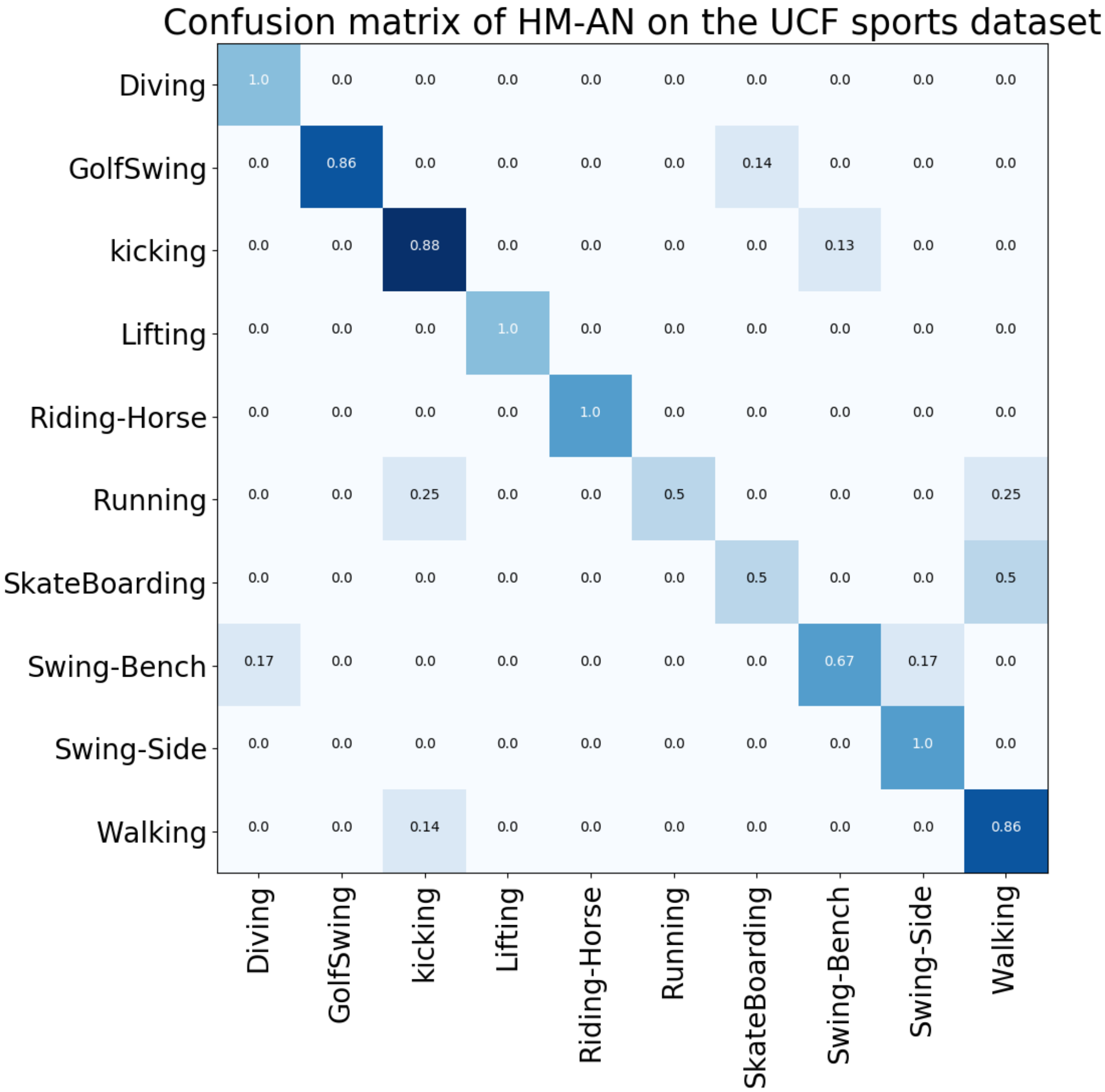
As shown in Table 4.8, the proposed model is compared with the methods proposed in [160] in which a convolutional LSTM attention network with hierarchical architecture was used for action recognition. The hierarchical architecture in [160] was pre-defined while the proposed model can learn the hierarchy from the data. The improvements demonstrated by our methods are evident as shown in Table 4.8.

Table 4.8: Accuracy on UCF Sports

Methods	Accuracy
Softmax Regression (Residue-152 Features)	66.0%
Baseline (Residue-152 Features)	70.0%
Conv-Attention [160] (Residue-152 Features)	72.0%
CHAM [160] (Residue-152 Features)	74.0%
Soft Attention (Residue-152 Features)(Ours)	81.1%
REINFORCE-Hard Attention (Residue-152 Features)(Ours)	**82.0%**
Constant-Gumbel-Hard Attention(Residue-152 Features) (Ours)	76.0%
Adaptive-Gumbel-Hard Attention (Residue-152 Features)(Ours)	**82.0%**

Olympic Sports dataset The Olympic Sports dataset is of medium size. Results from this dataset are shown in Table 4.9. The mean Average Precision (mAP) result of baseline approach is 73.7%. Our method HM-AN with Soft attention achieves 82.4% mAP. However, unlike the UCF Sports dataset, the mAP result of REINFORCE-Hard Attention is 77.1%, which is lower than the approach of Soft Attention. The Constant-Gumbel-Hard Attention, which is implemented by Gumbel-softmax with a constant temperature of 0.3, obtains a mAP value of 82.3%. By making the temperature value of Gumbel-softmax adaptive, the proposed model achieves 82.7% mAP, the highest among all our experimental results. Again, our proposed methods show superior performance compared to the hand-designed hierarchical model in [160].

HMDB51 dataset HMDB51 is a more difficult and larger dataset. First of all, the accuracy of softmax regression is tested, based on Residue-152 networks, with 38.2% accuracy, which improved this approach based on GoogleNet features by 4.7%. This is consistent with previous findings where the Residue-152 networks reported 23.0% top 1 error on ImageNet dataset [36], which is 11.2% percent less than the GoogleNet results (34.2%) [161] [43]. However, all the subsequent experiments are all performed using features from Residue-152 features, which verify that the performance gain is from the proposed model instead of the advanced image features. The performance of the baseline approach



	Diving	GolfSwing	kicking	Lifting	Riding-Horse	Running	SkateBoarding	Swing-Bench	Swing-Side	Walking
Diving	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GolfSwing	0.0	0.86	0.0	0.0	0.0	0.0	0.14	0.0	0.0	0.0
kicking	0.0	0.0	0.88	0.0	0.0	0.0	0.0	0.13	0.0	0.0
Lifting	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
Riding-Horse	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Running	0.0	0.0	0.25	0.0	0.0	0.5	0.0	0.0	0.0	0.25
SkateBoarding	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.5
Swing-Bench	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.67	0.17	0.0
Swing-Side	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
Walking	0.0	0.0	0.14	0.0	0.0	0.0	0.0	0.0	0.0	0.86

Figure 4.11: Confusion Matrix of HM-AN with Adaptive-Gumbel-Hard Attention on the UCF Sports dataset.

is shown in Table 4.11, with 40.8% accuracy. The three-layer LSTMs with soft attention based on GoogleNet features was reported in [65], with 41.3% accuracy. To make the comparison fair, the three layer LSTMs with soft attention are also tested on Residue-152

Table 4.9: AP on Olympics Sports

Class	Vault	Triple Jump	Tennis serve	Spring board	Snatch
Softmax Regression (Residue-152 Features)	97.7%	100.0%	42.8%	58.4%	31.7%
Baseline (Residue-152 Features)	97.0%	88.4%	52.3%	60.0%	23.2%
Conv-Attention (Residue-152 Features) [160]	97.0%	94.0%	49.8%	66.4%	26.1%
CHAM (Residue-152 Features) [160]	97.0%	98.9%	49.5%	69.2%	47.8%
Soft Attention (Residue-152 Features)(Ours)	99.0%	100.0%	60.7%	64.2%	38.6%
REINFORCE-Hard Attention (Residue-152 Features) (Ours)	100.0%	95.0%	50.8%	56.3%	28.6%
Constant-Gumbel-Hard Attention (Residue-152 Features) (Ours)	97.0 %	99.0%	62.6 %	58.7%	40.3%
Adaptive-Gumbel-Hard Attention (Residue-152 Features) (Ours)	98.1%	98.9%	62.1%	64.3%	45.4%

Class	Shot put	Pole vault	Platform 10m	Long jump	Javelin Throw	High jump
Softmax Regression (Residue-152 Features)	61.5%	88.8%	85.6%	96.6%	95.0%	79.7%
Baseline (Residue-152 Features)	67.4%	69.8%	84.1%	100.0%	89.6%	84.4%
Conv-Attention (Residue-152 Features) [160]	60.0%	100%	86.0%	98.0%	87.9%	80.0%
CHAM (Residue-152 Features) [160]	79.8%	60.8%	89.7%	100.0%	95.0%	78.7%
Soft Attention (Residue-152 Features)(Ours)	77.2%	85.4%	91.5%	98.9%	97.0	77.2%
REINFORCE-Hard Attention (Residue-152 Features) (Ours)	90.6%	100.0%	86.7%	100.0%	89.7%	77.5%
Constant-Gumbel-Hard Attention (Residue-152 Features) (Ours)	87.8%	100.0%	93.1%	100.0%	93.2%	82.8%
Adaptive-Gumbel-Hard Attention (Residue-152 Features) (Ours)	84.1%	100.0%	94.8%	100.0%	95.3%	86.2%

Class	Hammer throw	Discus throw	Clean and jerk	Bowling	Basketball layup	mAP
Softmax Regression (Residue-152 Features)	32.9%	84.2%	78.0%	41.5%	89.3%	72.7%
Baseline (Residue-152 Features)	38.0%	100.0%	76.0%	60.0%	89.8%	73.7%
Conv-Attention (Residue-152 Features) [160]	36.6%	97.8%	100.0%	46.8%	81.2%	75.5%
CHAM (Residue-152 Features) [160]	37.9%	97.0%	84.8%	46.7%	89.1%	76.4%
Soft Attention (Residue-152 Features)(Ours)	44.1%	94.2%	83.8%	63.9%	89.2%	77.1%
REINFORCE-Hard Attention (Residue-152 Features) (Ours)	52.9%	95.8%	92.4%	69.4%	98.1%	82.4%
Constant-Gumbel-Hard Attention (Residue-152 Features) (Ours)	54.7%	95.8%	91.3%	60.5%	100.0%	82.3%
Adaptive-Gumbel-Hard Attention (Residue-152 Features) (Ours)	53.8%	95.8%	84.9%	62.5%	97.0%	**82.7%**

features. However, a pronounced improvement on the final result can not be obtained, with 42.4% accuracy (1.1% gains over the result from [65]). Our HM-AN model with soft attention improves the accuracy to 43.8%. Then the REINFORCE-Hard Attention approach is applied to this dataset. The resulting accuracy turns out to be lower than the HM-AN with soft attention. Moreover, the model with REINFORCE-like algorithm converges slower than the Gumbel-softmax with adaptive temperature, also with more oscillations on the training cost, which is shown in Fig. 4.10. With a constant temperature value of 0.3 for hard attention, the model achieves 44.0% accuracy. Again, the improvement by adding adaptive temperature is obvious, with 44.2% accuracy on the HMDB51 dataset. The accuracy results are further summarized in Table 4.11.

The performance of the proposed HM-AN are also compared with some published models related to the proposed approach. The proposed approach shares similarity with the spatial convolutional net from the two-stream scheme [90]. The difference is that the two-stream approach performs fine-tuning on the CNN model, with improved accuracy of

40.5%. Recent research on the two-stream approach [151] reported better results, with 47.1% accuracy. However, the evaluation of the two-stream method is based on each video while our evaluation is based on 60 frame sequences. The sequence-based accuracy usually is lower than the video-based accuracy as described in [162]. The video-based approaches are only provided for reference since the evaluation of them is different from sequence-based approaches.

For sequence-based approaches, the methods not from the RNN family but only with the spatial image, show poor performance as illustrated in Table 4.12. Specifically, the softmax regression approach [65] directly uses extracted image features of each frame and performs softmax regression on them, with 33.5% accuracy. The softmax regression approach based on image features from Residue-152 networks improves the accuracy to 38.2%. [65] reported that the LSTM without attention achieves 40.5% accuracy [65]. When adding the soft attention mechanism, improved accuracy of 41.3% can be obtained. The Conv-Attention [160] and ConvALSTM [143] both use convolutional LSTM with attention. The differences are that Conv-Attention extracts features from Residue-152 Networks [43] without fine-tuning while ConvALSTM extracts image features from a fine-tuned VGG16 model. The ConvALSTM leads Conv-Attention by a small margin, with 43.3% accuracy. As explained previously, CHAM [160] has a hand-designed hierarchical architecture, which is in contrast with ours in which the temporal hierarchy is formed through training. Our best setting (Adaptive-Gumbel-Hard Attention) reports the highest accuracy (44.2%) among methods from the RNN family and leads the CHAM results (43.4%) by 0.8 per cent. In sequence-based approaches, the one that outperforms ours is the Long-term temporal convolutions [162], with 52.6% accuracy. This method has a 3D-convolution architecture and is trained directly on the specific dataset, which is very different from our approach.

Analysis and Visualization Four approaches (Soft Attention, REINFORCE-Hard Attention, Constant-Gumbel-Hard Attention and Adaptive-Gumbel-Hard Attention) were tested on three different datasets: UCF Sports dataset, the Olympic Sports dataset and

Table 4.10: Accuracy of Softmax Regression on HMDB51 based on Different Features

Image Features	Accuracy
GoogleNet	33.5%
Residue-152 Network	38.2%

Table 4.11: Accuracy on HMDB51

Methods	Accuracy
Softmax Regression (Residue-152 Features)	38.2%
Baseline (Residue-152 Features)	40.8%
Three LSTM Layers with Attention (Residue-152 Features)	42.4%
Soft Attention (Residue-152 Features)(Ours)	43.8%
REINFORCE-Hard Attention (Residue-152 Features)(Ours)	41.5%
Constant-Gumbel-Hard Attention (Residue-152 Features)(Ours)	44.0%
Adaptive-Gumbel-Hard Attention (Residue-152 Features)(Ours)	**44.2%**

Table 4.12: Comparison with related methods on HMDB51

Methods	Accuracy	Spatial Image Only	Fine-tuning of CNN model
Video Accuracy			
Spatial Convolutional Net (8 Layers CNN model) [90]	40.5%	Yes	Yes
Spatial Convolutional Net (VGG 16) [151]	47.1%	Yes	Yes
Composite LSTM Model [163]	44.0%	Yes	No
Trajectory-based modeling [142]	40.7%	No	No
Deep 3D CNN [164]	**51.9%**	Yes	Yes
Sequence Accuracy			
ConvALSTM (VGG16 model) [143]	43.3%	Yes	Yes
Long-term temporal convolutions [162]	**52.6%**	Yes	Yes
Softmax Regression (GoogleNet Features) [65]	33.5%	Yes	No
Average pooled LSTM [65] (GoogleNet Features)	40.5%	Yes	No
Three LSTM Layers with Attention (GoogleNet Features) [65]	41.3%	Yes	No
Three LSTM Layers with Attention (Residue-152 Features)	42.4%	Yes	No
Conv-Attention (Residue-152 Features) [160]	42.2%	Yes	No
CHAM (Residue-152 Features) [160]	43.4%	Yes	No
Adaptive-Gumbel-Hard Attention (Residue-152 Features) (Ours)	**44.2%**	Yes	No

the HMDB51 dataset. On the UCF Sports dataset, the REINFORCE-Hard Attention and Adaptive-Gumbel-Hard Attention generate satisfactory results and show better performance than the soft attention and Constant-Gumbel-Hard Attention. This indicates

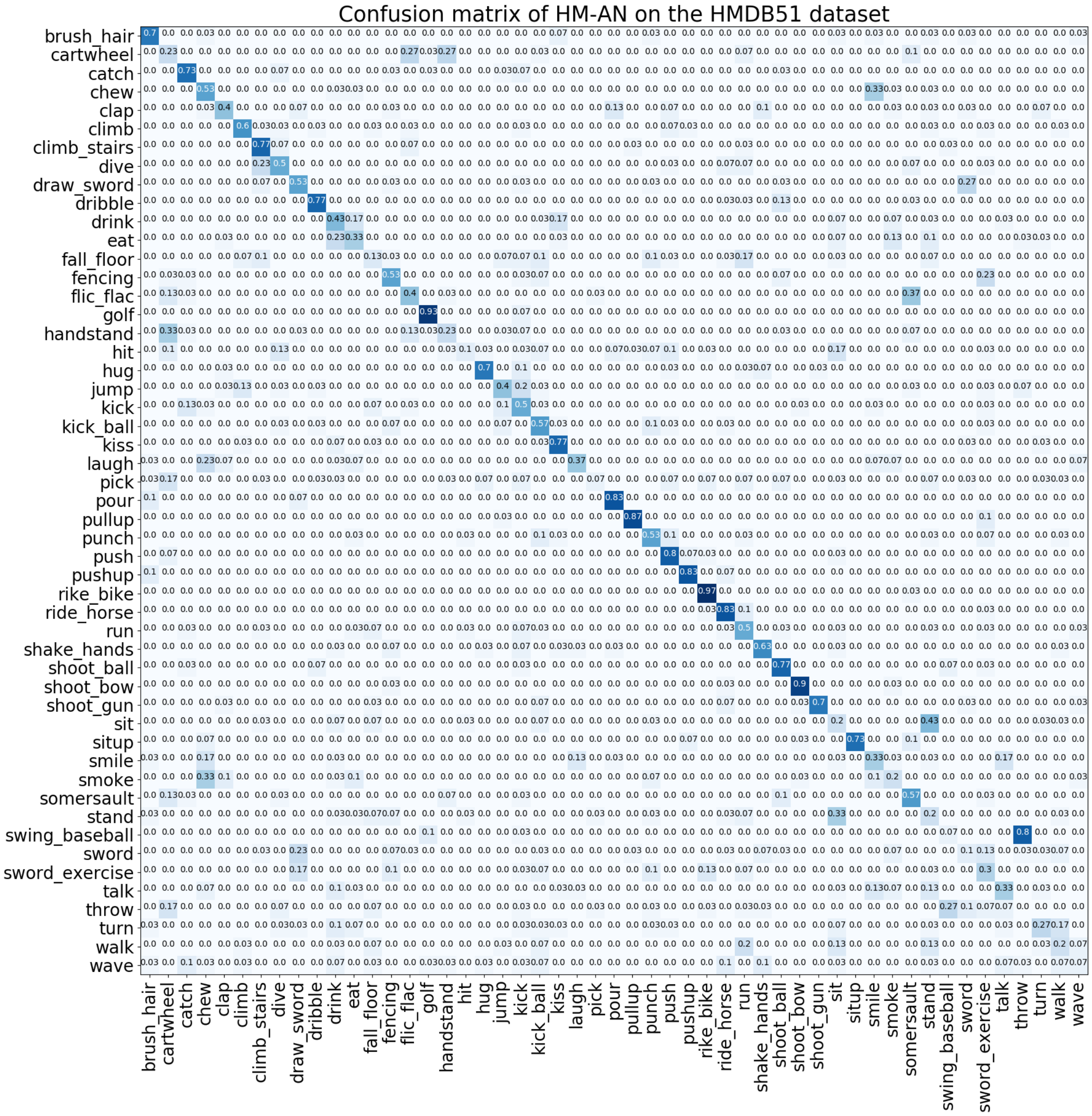


Figure 4.12: Confusion Matrix of HM-AN Adaptive-Gumbel-Hard Attention on the HMDB51 dataset.

that the adaptive temperature is an efficient method to improve performance in the implementation of Gumbel-softmax based hard attention.

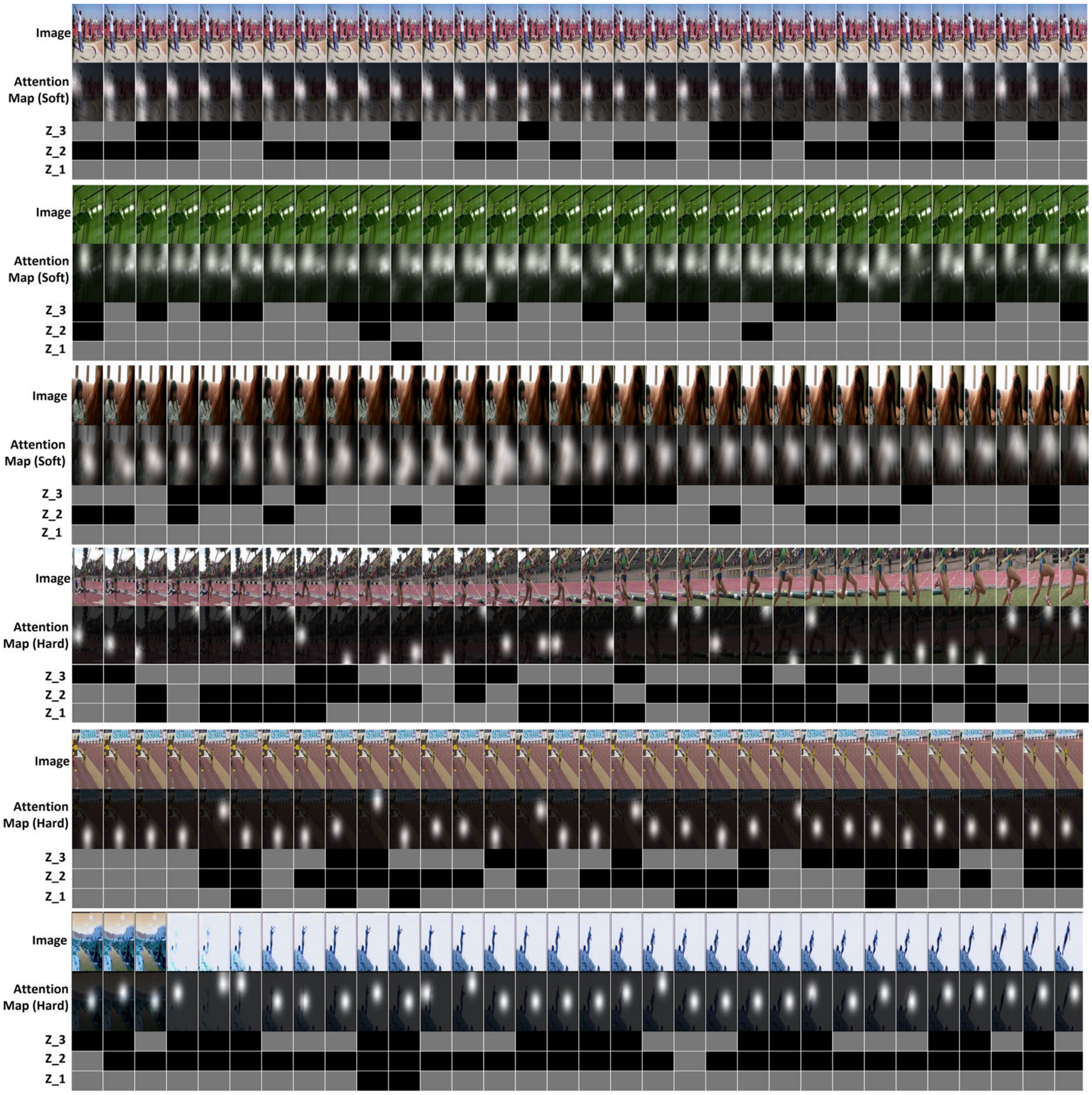


Figure 4.13: Visualization of attention maps and detected boundaries for action recognition.

On both of the Olympic Sports dataset and HMDB51 dataset, the best approach is the Adaptive-Gumbel-Hard Attention while the REINFORCE-Hard Attention is even worse than the soft attention mechanism. On the bigger datasets, the advantages of Gumbel-

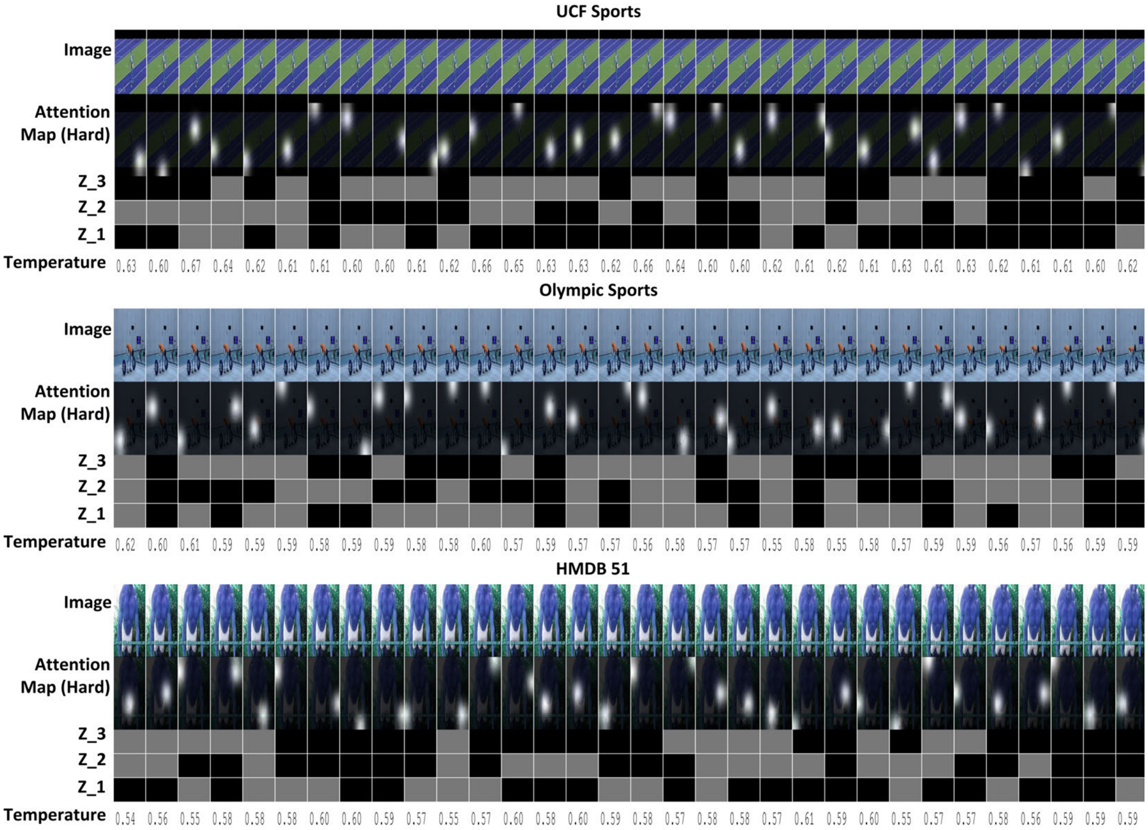


Figure 4.14: Visualization of temperature values with attention maps and detected boundaries for action recognition.

softmax include small gradient variance and simplicity, which are obvious compared with the REINFORCE-like algorithms. This shows that Gumbel-softmax generalises well on large and complex datasets. This is reflected not only by the result accuracy but also by the training cost curves in Fig. 4.9 and Fig. 4.10. This conclusion is also consistent with the findings in other recent research [148] which also applied both REINFORCE-like algorithms and Gumbel-softmax as estimators for stochastic neurons.

The visualization of attention maps and boundary detectors learnt by the HM-AN is shown in Fig. 4.13. In the attention maps, the brighter an area is, the more important it is for the recognition. The soft attention captures multi-regions while the hard attention selects only one important region. As can be seen from the figure, in different time steps, the attention regions are different which means the model is able to select region to facilitate the recognition through time automatically. The z_1, z_2 and z_3 in the figure indicate the boundary detectors in the first layer, the second layer and the third layer, respectively. In the figure, for the boundary detectors, the black regions indicate there exists a boundary in the time-domain whilst the grey regions show the UPDATE operation can be performed. The HM-AN can capture the multi-scale properties in the time-domain as different layers show different boundaries.

From the reported results, it is found that on all three datasets, the Constant-Gumbel-Hard Attention approach is worse than the approach of Adaptive-Gumbel-Hard Attention. This is because which temperature parameter is optimal for the dataset is not a prior knowledge. To provide a better understanding of the network, Fig. 4.14 shown how the adaptive temperature change along with the test samples on three datasets. The figure shown that the adaptive temperature is about 0.6, which is very different from the pre-defined 0.3 temperature in Constant-Gumbel-Hard Attention.

On the UCF Sports dataset, the Constant-Gumbel-Hard Attention is significantly worse than other approaches, including the REINFORCE-Hard Attention, with only 76.0% accuracy. As shown in Fig. 4.14, the temperature from the UCF Sports dataset is slightly higher than the other two datasets, which means the 0.3 pre-defined temperature parameter is not an appropriate option. Also, the approach of Adaptive-Gumbel-Hard Attention makes the networks converge much quicker as shown in Fig. 4.8, Fig. 4.9 and Fig. 4.10, which also explains the higher accuracy results of this method.

4.3.5 Conclusion

In this chapter, a novel RNN model, HM-AN, was proposed, which improves HM-RNN with attention mechanism for visual tasks. Specifically, the boundary detectors in HM-AN are implemented by the recently proposed Gumbel-sigmoid. Two versions of the attention mechanism were implemented and tested. Our work is the first attempt to implement hard attention in vision tasks with the aid of Gumbel-softmax instead of REINFORCE algorithm. To solve the problem of a sensitive parameter of the softmax temperature, the adaptive temperature methods were applied to improve the system performance. To validate the effectiveness of HM-AN, experiments were conducted on action recognition from videos. Through experimenting, it is proved that HM-AN is more effective than LSTMs with attention. The attention regions of both hard and soft attention and boundaries detected in the networks provide visualisation for the insights of what the networks have learnt. Theoretically, our model can be built based on various features, e.g., Dense Trajectories, to improve the performance. However, unlike many previous state-of-the-art methods like the two-stream approach [90], the emphasis in this chapter is to prove the superiority of the model itself compared with other RNN-like models given same features. Hence, only the deep spatial features is applied. Note the model can be easily extended to two-stream approach by using the optical flow features. Note that the RNN-based model is more time-consuming than the CNN-based model in action recognition. However, our work can facilitate further research on the hierarchical RNNs and its applications to computer vision tasks.

Chapter 5

Image Captioning based on Visual Attention Mechanism

5.1 Introduction

This chapter focuses on the application of the visual attention mechanism in the task of image description generation (image captioning). Section 5.2 proposes an attention model with adversarial training for image captioning. It uses a simple soft attention model for image captioning and an reinforcement learning-based adversarial training scheme to optimise the whole network. Section 5.3 improves the research in Section 5.2 in two aspects: a hierarchical attention networks to capture more fine-grained object features is proposed. Also, in the adversarial training scheme, a novel discriminator to measure the coherence and consistency between the content of the image and the generated language is proposed, with much improved effect on the final performance.

5.2 Image Captioning using Attention Mechanism and Adversarial Training

5.2.1 Introduction

Image captioning, i.e., automatically describing the content of an image, is a fundamental problem in machine learning which connects computer vision and natural language processing. It tries to mimic the human ability to process huge amounts of salient visual information into descriptive language, which is one of the primary goals of artificial intelligence.

In recent years, remarkable progresses have been made towards naturalistic image description generation [165] [166] [167] [71], owing to the development of deep learning [2]. In these works, inspired by the success of the sequence-to-sequence model of neural machine translation [8] [168], most of them represented the image as a single feature vector from the top layer of pre-trained CNNs and cascaded RNNs to generate text. Subsequent research [167] introduced the attention mechanism on image locations to discriminate between important and relevant image features to facilitate image captioning.

However, most of the previously proposed models trained the RNN using Maximum Likelihood Estimation (MLE) to generate image descriptions. As argued in [169], the MLE approaches suffer from the so-called exposure bias in the inference stage: the model generates a sequence iteratively and predicts the next token based on the previously predicted ones that may never be observed in the training data. In image captioning, the MLE also suffers from a problem that the generated captions do not correlate well with a human assessment of quality.

Instead of only relying on the MLE, an alternative scheme is under the framework of the GANs [3]. GAN was first proposed to generate realistic images. GAN learns generative models without explicitly defining a loss function from the target distribution. Instead, GAN introduces a discriminator network which tries to differentiate real samples from generated samples. The whole network is trained using this adversarial training strategy. One can subsequently build a discriminator to judge how realistic the samples generated by the caption generator are. The caption generator is similar to the generator in conditional GAN [170], which is conditioned on the image features.

There is an inherent problem in GAN when dealing with language problems. Language, unlike images, is essentially a discrete problem. Directly providing these discrete tokens as inputs to the discriminator does not allow the gradients to backpropagate through them since they are discontinuous. One solution is to implement a Reinforcement Learning (RL) [171] framework to estimate the gradients of the discontinuous units. However, the RL framework, when dealing with sequence generation, has the problem of lacking the intermediate reward, as discussed in [50]. The reward signal can only be obtained when the whole sequence is generated. This is not suitable since what is wanted is the long-term reward of each intermediately generated token, which is to optimise the whole sequence better.

To tackle the issues mentioned above, the framework of GAN is applied for image captioning. In the proposed scheme, the discriminator not only considers the similarity between the generated captions and the reference captions but also the consistencies between the captions and image features. Through evaluation of the discriminator, the networks can better compensate for the issue where some unrealistic captions might be generated using MLE. Also, to deal with the discreteness of language, the image captioning generator is considered as an agent of RL. The feedback from the discriminator are considered as the rewards for the generator. To update the parameters of image captioning generator in this framework, the generator is considered as a stochastic parameterised policy. The policy network is trained using Policy Gradient [172], which naturally solves the differential difficulties in conventional GAN. Also, to solve the problem of the lack of intermediate rewards, the idea from the famous “AlphaGo” program [173] is applied in which a Monte Carlo roll-out strategy is used to sample the expected long-term reward for an intermediate move. If the sequence token generation is considered as the action to be taken in RL, a similar Monte Carlo roll-out strategy can be applied to obtain the intermediate rewards. [50] has successfully applied the Monte Carlo roll-out in sequence generation. In this chapter, a similar sampling method is used to deal with intermediate rewards during the process of caption generation.

During implementation, the caption generator is built based on the "show, attend and tell" model [167]. The feature processing and soft attention mechanism are adopted the same in [167]. Then the the image captioning model is considered as the generator, and another RNN network is used as a discriminator, to automatically evaluate how realistic the generated captions are. The outputs from the discriminator are considered as the rewards in the RL framework. The entire networks are trained using the Policy Gradient algorithm. The proposed model was evaluated on the COCO dataset [121], with improved results over the model based on MLE.

Our contributions can be summarised as follows:

- GAN and RL is proposed to be applied to train a neural model for the image captioning task.
- A Monte Carlo roll-out strategy is applied to obtain intermediate rewards for RL in the sequence generation scenario.
- Experiments prove the effectiveness of adversarial training and RL in the task of image captioning.

5.2.2 The Proposed Method

The proposed scheme is based on GANs, in which a generator and a discriminator are trained using the minimax game in an adversarial way. On the one hand, the generator tries to generate realistic samples to fool the discriminator into believing they are real ones. On the other hand, the discriminator is trained to identify the differences between the generated samples and real ones.

In the proposed scheme, the image captioning generator is considered as the generator in a GAN framework, which tries to generate naturalistic image descriptions. A discriminator is built to judge whether the generated sequence is realistic. In the vanilla GAN, the gradient from the discriminator can be backpropagated directly to the generator, which makes the whole network trainable. However, due to the discrete problem of language, this is not achievable using vanilla GAN. Hence, the model is considered in the framework of RL and a Policy Gradient is applied to estimate the gradients of the generator. In the following subsections, the generator, the discriminator, the Policy Gradient algorithm and the training algorithm will be explained, respectively. The system diagram can be seen in Fig. 5.1.

Image Captioning Generator

The image caption generator is based on the model in [167]. Specifically, the model consists of an encoder and a decoder. A convolutional neural network (Residual Net [43]) pre-trained on the ImageNet dataset [69] is used in order to extract a set of convolutional features. These features, denoted as $a = \{a_1, ..., a_L\}$, correspond to certain portions of the 2-D image. The convolutional features are extracted instead of fully connected ones in order to build a soft attention mechanism to discriminate the visual location of the given image.

The LSTMs network, initially proposed by Hochreiter and Schmidhuber in [45], is applied as the language decoder because of its superior performance in natural language processing.

$$\begin{aligned}
i_t &= \sigma(W_{xi} * z_t + W_{hi} * h_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * z_t + W_{hf} * h_{t-1} + b_f) \\
o_t &= \sigma(W_{xo} * z_t + W_{ho} * h_{t-1} + b_o) \\
g_t &= \sigma(W_{xc} * z_t + W_{hc} * h_{t-1} + b_c) \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot g_t \\
h_t &= o_t \cdot \phi(c_t)
\end{aligned} \tag{5.1}$$

In Equation 5.11, i_t, f_t, o_t, c_t and h_t are the input, forget, output, cell memory and hidden state of an LSTM network, respectively. z_t is the context vector, which can be processed by a soft attention mechanism and can capture visual information associated with certain input locations. The soft attention mechanism has to automatically allocate adaptive weights, on image locations, to facilitate the task at hand.

$$e_{ti} = f_{att}(a_i, h_{t-1}) \tag{5.2}$$

Equation 5.12 maps the image features from each location, along with information from the hidden state, into an adaptive weight, which indicates the importance of each image location for recognition.

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{L} exp(e_{tk})} \tag{5.3}$$

Then, Equation 5.13 normalises the adaptive weights into a probability value in the range of 0 to 1 using the softmax function. Once these weights (sum to 1) are computed, the weights vector α_t is elementwisely multiplied with the image feature vector a and are summed up to generate the context vector z_t, which can be expressed as in Equation 5.14.

$$z_t = \sum_{i=1}^{L} \alpha_{t,i} a_i \tag{5.4}$$

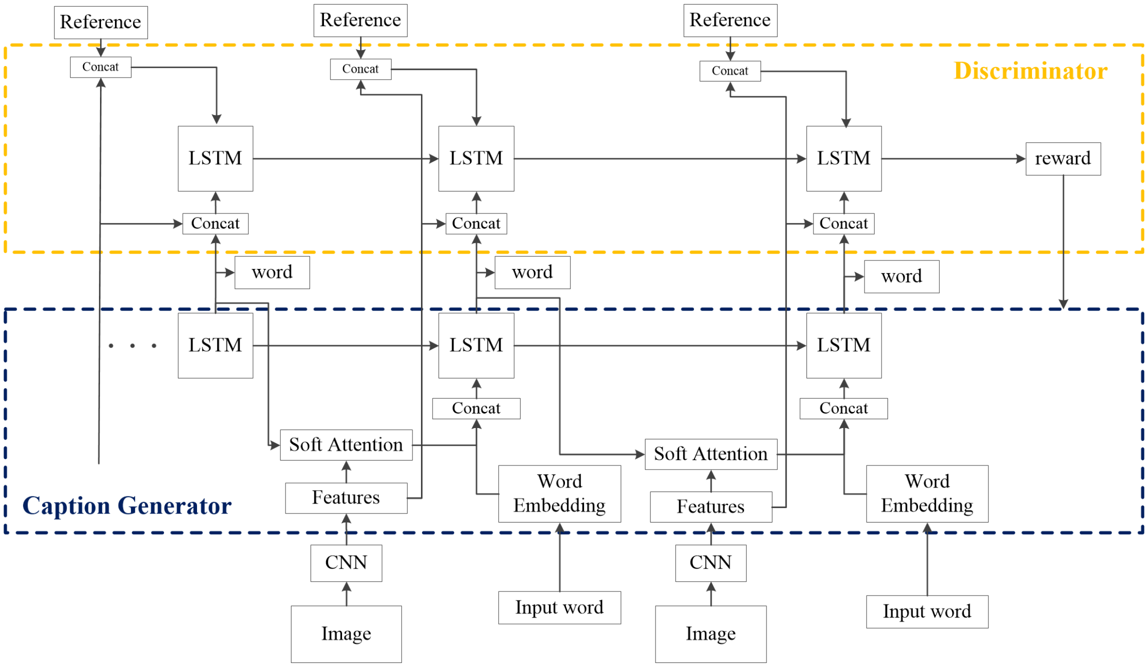


Figure 5.1: System Diagram of the Proposed Model.

Then the context vector z_t is forwarded to the LSTM network to generate captions, as described in Equation 5.11. This soft attention mechanism can adaptively select relevant visual parts of the given image features and thus facilitate recognition.

Discriminator

The generated sequences and the reference sequences are simultaneously given to the discriminator. Before being forwarded to the discriminator, both of the embedding matrices of the generated sequences and the reference sequences are concatenated with the image features, which can be seen in Fig. 5.1. This operation is to consider the coherence between certain captions (sequences) and the corresponding image features, which can make the generated captions more realistic and naturalistic. The reference sequences are labelled as true while the generated sequences are labelled as false during the training of the discriminator. The model is also an LSTM network with softmax cross entropy loss. Hence, the discriminator outputs the probabilities of a sample being true. These probabilities, are then considered as the reward signal in the RL framework, to be utilised by the Policy Gradient algorithm for updating the parameters of the image caption generator.

Optimization via Policy Gradient

Following [172], the objective of the policy network $G_\theta(y_t|y_{1:t-1})$ (the image caption generator), is to generate a sequence from the start state s_0 to maximize its expected long-term reward as described in Equation 5.21:

$$J(\theta) = E[R_T|s_0, \theta] = \sum_{y_1 \in Y} G_\theta(y_1|s_0) \cdot Q_{D_\theta}^{G_\theta}(s_0, y_1) \tag{5.5}$$

where R_T is the reward for a complete sequence. $Q_{D_\theta}^{G_\theta}(s, y)$ is the action-value function of a language sequence, which is defined as the expected accumulative reward starting from state s, taking action a, and then following policy G_θ.

The action-value function is estimated using the REINFORCE algorithm [70] and considers the probability of being real, generated by the discriminator, as a reward, which can be defined as in Equation 5.22.

$$Q_{D_\theta}^{G_\theta}(a = y_T, s = Y_{1:T-1}) = D_\theta(Y_{1:T}) \tag{5.6}$$

As can be seen in Equation 5.22, the discriminator only provides a reward for a complete sequence. One should not only care about the reward for complete tokens but also the long-term reward for the future time-steps since the long-term reward is what is actually needed. Similar to the game of Go [173] in which the agent sometimes gives up immediate interest but cares about the final victory, a similar Monte Carlo roll-out strategy is applied for an intermediate state, i.e., an unfinished sequence. An N-time Monte Carlo search is represented as in Equation 5.23.

$$\begin{aligned} &Y^1_{1:T}, ..., Y^N_{1:T} = MC^{G_\theta}(Y_{1:t}; N) \\ &MC =\sim Multinomial(logits) \end{aligned} \tag{5.7}$$

where $Y^n_{1:T}$ is the generated sequence tokens and $Y^n_{t+1:T}$ is Monte Carlo sampled based on a roll-out policy, which, in this case, is set the same as the image caption generator. *logits* is the output of the LSTM decoder. MC is defined as a sampling procedure from Multinomial distribution.

If there is no intermediate reward, the Monte Carlo roll-out strategy can sample the possible future tokens N times and average these rewards to achieve the goal of reward estimation, which is described in Equation 5.24.

$$Q^{G_\theta}_{D_\theta}(a = y_t, s = Y_{1:t-1}) = \\ \begin{cases} \frac{1}{N}\sum_{n=1}^{N} D_\theta(Y^n_{1:T}), Y^n_{1:T} \in MC^{G_\theta}(Y_{1:t}; N), & for\ t < T \\ D_\theta(Y_{1:T}), & for\ t = T \end{cases} \tag{5.8}$$

The Monte Carlo roll-out strategy can be better visualised in Fig. 5.8.

Once the reward value from the discriminator is obtained, it is ready to update the generator. One can use the Policy Gradient theorem from [172] and write the gradient of the objective function (reward signal) as in Equation 5.26.

$$\bigtriangledown_\theta J(\theta) = \sum_{t=1}^{T} E_{Y_{1:t-1}\sim G_\theta}[\sum_{y_t \in Y} \bigtriangledown G_\theta(y_t|Y_{1:t-1}) \cdot Q^{G_\theta}_{D_\theta}(Y_{1:t-1}, y_t)] \tag{5.9}$$

Since the expectation can be approximated by sampling, the parameters of the image caption generator can be updated using Equation 5.27.

$$\theta \leftarrow \theta + \alpha_h \bigtriangledown_\theta J(\theta) \tag{5.10}$$

In practice, advanced gradient algorithms such as RMSprop [174] and Adam [141] are used

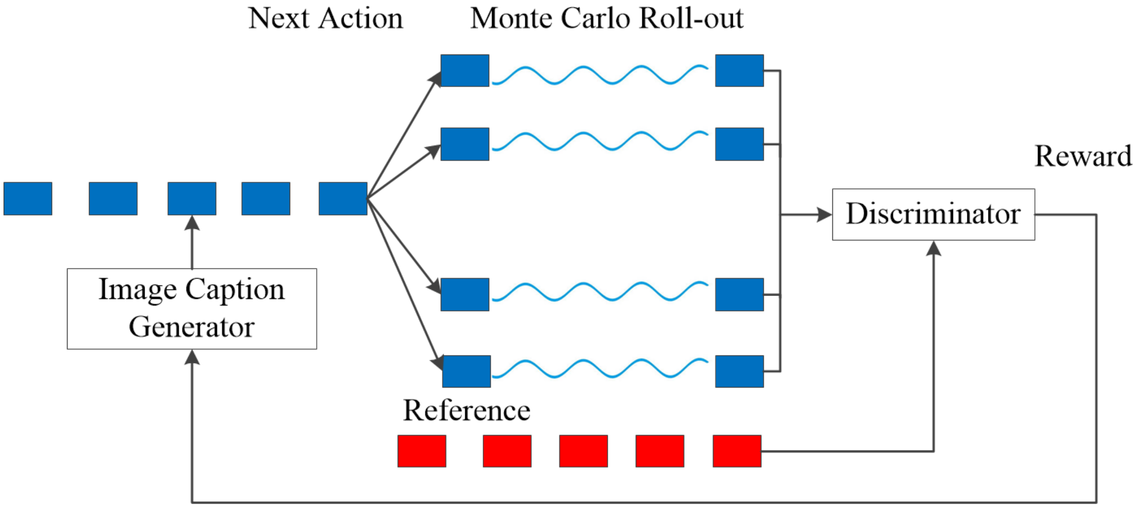


Figure 5.2: Monte Carlo roll-out.

in training the caption generator.

Adversarial Training

The image caption generator and discriminator are adversarially trained in the GAN framework [3]. In GAN [170], the discriminator can pass the gradient directly to the generator. Due to the discreteness of sequence generation, RL is applied to estimate the gradient for the generator in our model.

Specifically, the training strategy can be described in Algorithm 2. the image caption generator is firstly pre-trained using MLE. In practice, this is equivalent to the cross-entropy loss [175]. Hence, the pre-training step is set as the same with [167]. The trained model is used to generate some captions which are set as fake samples, which, along with the reference captions, are fed to the discriminator for training. Similarly, the discriminator is also pre-trained for certain steps. The next step is the adversarial training step, in which the image caption generator and discriminator have trained alternatively until the convergence of the networks.

Algorithm 1 Image Caption Generation by Adversarial Training and Reinforcement Learning

```
Require: Image Caption Generator $G_\theta$; Discriminator $D_\theta$.
  Pre-training $G_\theta$ using MLE by some epoches.
  Generating negative samples using pre-trained $G_\theta$ to train $D_\theta$.
  Pre-training $D_\theta$ by 2500 iterations.
  repeat
    for update generator for 1 step do
      Generate a sequence $Y_{1:T} = (y_1, .., y_T)$.
      for $t = 1$ to $T$ do
        Compute the intermediate reward $Q(t)$ by Monte Carlo roll-out.
      end for
      Update the parameters $\theta$ using Policy Gradient.
    end for
    for update discriminator for 1 step do
      Training discriminator $D_\theta$ using reference sequence (True) and generated sequence
      (Fake) using current generator.
    end for
  until Convergence
```

5.2.3 Experimental Results

Experimental protocol

The experiments were conducted using the COCO dataset [121]. To be consistent with [167], the COCO 2014 released version was used, which includes 123,000 images. The "Karpathy" splits [165] are used. The standard evaluation protocol contains BLEU [176] and METEOR [177].

At training time, the maximum length of the input sequence is set to 20. During the testing phase, the maximum length of the generated symbols is set to 30.

Implementation Details

The raw images are resized to 224 × 224 pixels. Then the deep convolutional features (from the layer "res5c") are extracted using a pre-trained Residual-152 network [43] under the Caffe platform [109] because of its high efficiency in extracting features. The features

Table 5.1: Comparison of image captioning results on the COCO dataset with different image encoders

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Soft attention with MLE (VGG-19)	65.7	44.7	30.5	21.1	21.6
Soft attention with GAN and RL (VGG-19)	66.7	45.4	31.0	21.4	21.5
Soft attention with MLE (Residual Net)	70.0	50.3	35.4	25.1	23.6
Soft attention with GAN and RL (Residual Net)	**71.6**	**51.8**	**37.1**	**26.5**	**24.3**

Table 5.2: Experimental validation of the improvement by using Monte Carlo roll-out

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Soft attention with GAN and RL without Monte Carlo roll-out (VGG-19)	66.0	45.0	30.4	21.1	21.3
Soft attention with GAN and RL with Monte Carlo roll-out (VGG-19)	**66.7**	**45.4**	**31.0**	**21.4**	**21.5**
Soft attention with GAN and RL without Monte Carlo roll-out (Residual Net)	71.2	50.9	36.8	26.2	24.0
Soft attention with GAN and RL with Monte Carlo roll-out (Residual Net)	**71.6**	**51.8**	**37.1**	**26.5**	**24.3**

from the first fully connected layer of the VGG16 [42] network are also extracted, to make an experimental comparison on different image encoders. The "show, attend and tell" model are re-implemented on the Tensorflow platform [178]. The adversarial networks and Monte Carlo roll-out are also implemented under the same platform.

The batch size is set as 64 and learning rate to 0.0001 for both the MLE pre-training and Adversarial training. The number of Monte Carlo roll-outs is set as 20. During sampling, the maximum log-likelihoods that the network outputs are used. Although other techniques, like beam search, are proven to be better than maximum log-likelihoods, what needs to be analysed is the improvement of the model itself instead of other greedy techniques. Hence, the maximum log-likelihoods sampling are used in both the MLE training and adversarial training.

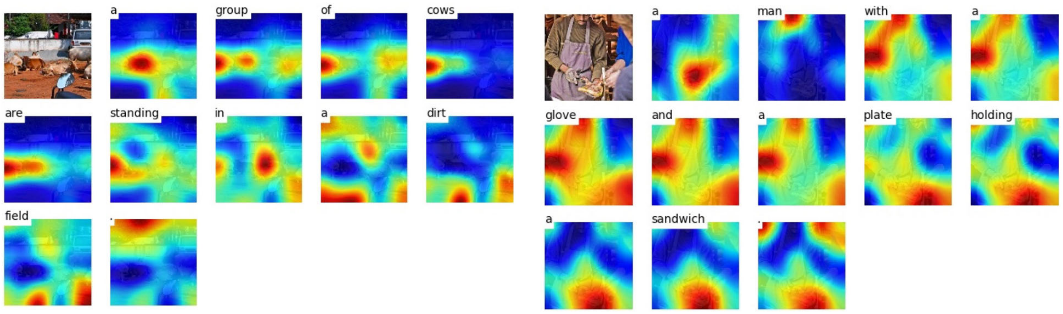
Results

Quantitative Evaluation

- Following [167], the generated captions are evaluated using the metrics of BLEU (1-4) and Meteor and performed certain ablation studies in different settings.

Table 5.3: Comparison of image captioning results on the COCO dataset with previous methods

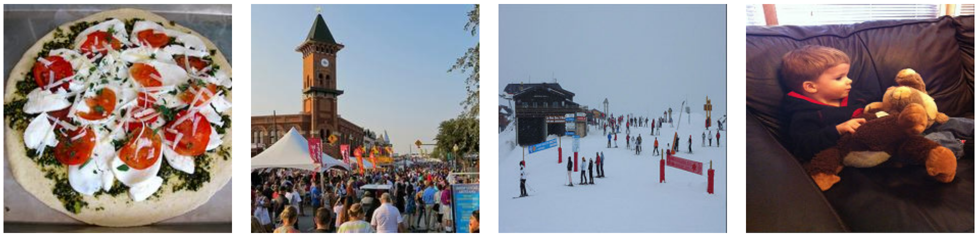
Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
CMU/MS Research [179]	-	-	-	-	20.4
MS Research [180]	-	-	-	-	20.7
LRCN [133]	58.7	39.0	25.0	16.5	-
BRNN [165]	64.2	45.1	30.4	20.3	-
Google NIC	66.6	46.1	32.9	24.6	-
Log Bilinear [167]	70.8	48.9	34.4	24.3	20.0
LSTM with Soft attention [167]	70.7	49.2	34.4	24.3	23.9
LSTM with hard attention [167]	**71.8**	50.4	35.7	25.0	23.0
RL with G-GAN [181]	-	-	30.5	29.7	22.4
RL with Embedding Reward [182]	71.3	**53.9**	**40.3**	**30.4**	**25.1**
Soft attention with GAN and RL (VGG-19)	66.7	45.4	31.0	21.4	21.5
Soft attention with GAN and RL (Residual Net)	71.6	51.8	37.1	26.5	24.3



(a) (b)

Figure 5.3: Visualization of attention maps.

- In addition to using the Residual Net as an image encoder, the VGG-19 [42] is also utilised as the image encoder to see the critical role of advanced image features in the image captioning task. The results can be seen in Table 5.1. The advanced image features from Residual Net bring a significant gain on the overall performance of caption generation. Take the results using MLE for example, for the metric of BLEU (1-4); the average raise is 4.7, which is a pronounced increase for the image captioning task.



(a) Caption generated by MLE:
A pizza with tomatoes and onions on it.
Caption generated by our model:
A pizza with cheese and vegetables on a plate.

(b) Caption generated by MLE:
A crowd of people standing in front of a building.
Caption generated by our model:
A crowd of people standing around a large clock tower.

(c) Caption generated by MLE:
A group of people standing on top of a snow covered slope.
Caption generated by our model:
A group of people are skiing on a snowy hill.

(d) Caption generated by MLE:
A small child is playing a video game.
Caption generated by our model:
A small child sitting on a couch holding a stuffed animal.

Figure 5.4: Visualization of generated languages.

- Given the same image features, the proposed method using GAN and RL leads the MLE method in most of the evaluation metrics, under the same image features and the same generator model, which proves the effectiveness of the adversarial training and policy gradient technique, which is shown in Table 5.1.

- To study the effectiveness of Monte Carlo roll-out, a model without a Monte Carlo roll-out strategy, i.e., the reward can only be obtained after the full captions are generated, is first tested. The results of this model are compared with the model using the Monte Carlo roll-out strategy, which can be seen in Table 5.2. As the results reveal, scores from all the evaluation metrics increase by adding an intermediate reward using Monte Carlo roll-out.

- As described in Table 5.3, the proposed method outperforms many related approaches including the attention models [167], which validates the improved effect brought by adversarial training and RL. RL with G-GAN [181] applies conditional GAN and policy gradient to generate image descriptions. Although their results on the evaluation metrics are not improved, they prove that the generated captions are more diverse and natural. Embedding Reward [182] applies a policy network to generate captions and a value network to evaluate the reward. Additionally, they also apply an advanced inference method called lookahead inference and beam search during testing. Competitive results are also achieved on this dataset.

Qualitative Evaluation

- The visualisation of the attention maps learnt can be seen in Fig. 5.3. In different time steps, the model adaptively selects relevant parts for the generated word. In the figure, a red region means these parts are selected while a blue region indicates unimportant parts.
- Some examples of generated captions are also randomly selected for both the MLE model and our model, which are described in Fig. 5.12. In the figure, the generated captions from our model are more accurate and realistic since our discriminator can measure the coherence between captions and image contents.

5.2.4 Conclusion

This research focused on the image captioning task, which is a fundamental problem in artificial intelligence. To address the inherent exposure bias problem of MLE training in sequence problems, an adversarial training method was applied. To estimate the gradients of the network, the feedback from the discriminator was treated as the reward signal in the RL framework. In RL, a long-term reward for each action is needed. In sequence generation, however, the reward can only be obtained when the sequence is generated. To tackle this issue, a Monte Carlo roll-out sampling method was applied to estimate the intermediate reward for each time step. The whole network was trained using the proposed three-step training strategy, which includes pre-training the regenerator, pre-training the discriminator, and adversarial training. Experimental results prove the improved effects of the proposed method. Also, visualisation shows the generated captions from the proposed model are more accurate than the ones from MLE training.

5.3 Image Captioning based on Attention Mechanism and Reinforcement Learning

5.3.1 Introduction

Naturalistic description of an image is one of the primary goals of computer vision, which has received much attention in the field of artificial intelligence recently. It is a high-level task and much more complicated than some fundamental recognition tasks, e.g., image classification [18] [42] [43] [183], image retrieval [184] [185] [186], object detection and recognition [112] [22] [23] [187]. This requires the system to comprehensively understand the content of an image and bridge the gap between the image and the natural language. Automatically generating image descriptions is useful in multimedia retrieval, and image understanding.

Some pioneering research has been carried out in generating image descriptions [188] [180]. However, as pointed out in [165], most of these models often rely on hard-coded visual concepts and sentence templates, which limits their generalisation capability. Recently, with the rapid development of deep learning in image recognition and natural language processing, the current trend of image captioning approaches [189] is to follow the encoder-decoder framework, which shares the similarity with that in neural machine translation [47]. Most of these approaches represented the image as a single feature vector from the top layer of pre-trained CNNs and cascaded RNNs to generate languages.

The tasks like image captioning and machine translation can be considered as a structured output problem where the task is to map the input to an output that possesses its structure, as stated in [190]. An inherent challenge in these tasks is the structure of the output is closely related to the structure of the input. Hence, a critical problem in these tasks is alignment [190]. Take neural machine translation, for example, [8] trained a neural model to align the output to the input for machine translation softly. Subsequent research [4] applied the visual attention model to address this problem in image captioning, with much improvement. The visual attention mechanism is to dynamically select the relevant receptive fields in the CNN features to facilitate the image description generation, which, in other words, is to align the output words to spatial regions of the source image. In this chapter, the visual attention mechanism is also employed for image captioning.

Nevertheless, natural language often consists of very meticulous descriptions, which correspond to the fine-grained objects of an image. As pointed out by [191], there are certain limitations of the most existing neural model-based schemes due to the mere use of the global feature representation in the image level. Some of the fine-grained objects might not be recognised by only relying on the global image features. In this chapter, a scheme to use a pre-trained image detection model, i.e., Faster RCNN [23], to retrieve the fine-grained image features from the top detected objects, is proposed. These fine-grained object features can provide complementary information for the global image representation, which will be proved in the experiments. Regarding the model structure, the object features are also processed by a visual attention mechanism, and are added to the original model to form a hierarchical feature representation and hence it can generate more accurate descriptions.

In addition to the improvement of the image feature representation, the current language model, which is widely used in neural machine translation and image captioning, is also considered to be improved. An issue with most of the previous language model is the training framework, namely, the RNN using MLE to generate image descriptions. As pointed out in [169], the MLE approaches suffer from the so-called exposure bias in the inference stage: the model generates a sequence iteratively and predicts the next token based on the previously predicted ones that may never be observed in the training data. In image description generation, the MLE also suffers from a problem that the generated languages do not correlate well with a human assessment of quality [181].

Instead of only relying on the MLE, an alternative scheme is the GANs [3]. GAN was first proposed to generate realistic images. The GAN learns generative models without explicitly defining a loss function from the target distribution. Instead, GAN introduces a discriminator network which tries to differentiate real samples from generated samples. The whole network is trained using an adversarial training strategy. One can subsequently build a discriminator to judge how realistic are the samples generated by the description generator. The role of the caption generator, in this model, is similar to that of the generator in the conditional GAN [170], which is conditioned on the image features.

However, language generation is a discrete process. Directly providing the discrete samples as inputs to the discriminator does not allow the gradients to be backpropagated through them. The RL [171] framework provides a solution to estimate the gradients of the discontinuous units. The RL framework, when dealing with sequence generation, has the problem of lacking the intermediate reward, as discussed in [50]. The reward value can only be obtained when the whole sequence is generated. This is not suitable as what is wanted is the long-term reward of each intermediately generated token, so the whole sequence better optimised.

In the proposed scheme, the discriminator takes into account not only the differences between the generated captions and the reference captions but also the consistencies between captions and image features. Through the evaluation of the discriminator, the networks can better compensate for some unrealistic captions which might be generated under the MLE training. However, to deal with the discreteness of language, the image captioning generator is considered as an agent of RL. The feedback from the discriminator are considered as the rewards for the generator. To update the parameters of the image description generator in this framework, the generator is considered as a stochastic parameterised policy. The policy network is trained using Policy Gradient [172], which naturally solve the differential difficulties in conventional GAN. Also, to solve the problem of lacking intermediate rewards, a similar idea with the famous “AlphaGo” program [173] is used. A Monte Carlo roll-out strategy is applied to sample the expected long-term reward for an intermediate move. If the token generation is considered as the action to be taken in RL, a similar Monte Carlo roll-out strategy can be applied to obtain the intermediate rewards. [50] has successfully applied the Monte Carlo roll-out in sequence generation. In this chapter, a similar sampling method is used to deal with intermediate rewards during the process of caption generation.

To summarise, our contribution in this chapter is threefold:

- A hierarchical attention mechanism is proposed to reason on the global features and the local object features for image captioning.
- The policy gradient algorithm combined with the GAN is proposed for the training and optimisation of the language model, with improvements over the MLE training scheme.
- Through extensive experiments, the proposed algorithm is validated, and comparable results with current state-of-the-art methods are achieved on the COCO dataset.

5.3.2 Related Work

Deep Model-based Image Captioning

Promoted by the recent success of deep learning network in image recognition tasks and machine translation, the research on generating image description or image captioning has made remarkable progress [192] [165] [180] [193] [189] [133]. As mentioned above, most of the previously proposed approaches consider the image description generation as a translation process, mainly by borrowing the idea of the encoder-decoder framework [194] from neural machine translation [47]. Generally, this paradigm considers a deep CNN model as the image encoder, which maps the image into a static feature representation, and an RNN as a decoder to decode this static representation to an image description.

The whole framework is trained using supervised learning under MLE. The generated description should be grammatically correct and match the content of the image.

Specifically, Karpathy et al. [165] proposed an alignment model through a multi-modal embedding layer. This model can align parts of description with the corresponding regions of the image, which attracts significant attention. Jia et al. [193] proposed a variation of LSTM, called gLSTM, for the image captioning task to mainly tackle the problem of losing track of the image content. This model includes the semantic information along with the whole image as inputs to generate captions. Donahue et al. [133] applied both of the convolutional layers and recurrent layers to form a Long-term Recurrent Convolutional Network (LRCN) for visual recognition and description.

Bahdanau et al. [8] pointed out that a potential problem with this approach is that the model should compress all the necessary information of a source sentence into a fixed-length representation. This may make it difficult for the neural network to cope with long sentences. The static feature representation in the encoder-decoder framework, for both of machine translation and image captioning, cannot automatically retrieve relevant information from the source and thus at last influence the final performance. In neural machine translation, Bahdanau et al. [8] proposed a kind of soft attention mechanism for machine translation, which enables the decoder to focus on the relevant parts of the source sentence automatically. In computer vision, the attention mechanism has long been the focus of much research [8] [63] [64] since human perception does not tend to process a whole scene in its entirety at once but applies some mechanisms to focus on the information needed selectively. A comprehensive study for hard attention bound with reinforcement learning and soft attention for the task of image captioning was published by Xu et al. [4].

Yao et al. [195] tackled the video captioning task through capturing global temporal structures among video frames with a temporal attention mechanism, which makes the model dynamically focus on the keyframes that are more relevant with the predicted word. Attention Models (ATT) developed by You et al. [71] first extracted semantic concept proposals and fused them with RNNs into hidden states and outputs. This method used K-NN, multi-label ranking to extract semantic concepts or attributes and fused these concepts into one vector using an attention mechanism. Similarly, Yao et al. [196] embedded attributes with image features into an RNN with various methods to boost the image captioning performance. Recently, Chen et al. [72] proposed to combine the spatial attention and the channel-wise attention mechanism for image captioning, with improved results. Alternatively, Li et al. [191] proposed a global-local attention mechanism to include local features extracted from the top detected objects from a pre-trained object detector. Inspired by [191], the local features from top detected objects are also included. However, a hierarchical model is built in this research while they treated local and global features equivalently.

Policy Gradient Optimization for Image Captioning

Another approach to boost the performance of language tasks is to compensate for the so-called exposure bias problem in RNN-based MLE learning. As pointed out in [197], RNNs are trained by MLE, which essentially minimised the KL-divergence between the distribution of target sequences and the distribution defined by the model. This KL-divergence objective tends to favour a model that overestimates its smoothness, which can lead to unrealistic samples [198].

In order to tackle the problems and generate more realistic image descriptions, some researches directly use evaluation metrics such as BLEU [176], METEOR [199] and ROUGE [200] as the reward signal and build the model under the RL framework. For instance, Ranzato et al. [201] are the first research using the policy gradient algorithm in an RNN-based sequence model, in which a REINFORCE-based approach was used to calculate the sentence-level reward and a Monte-Carlo technique was employed for training. Liu et al. [202] studied several linear combinations of the evaluation metrics and proposed to use a linear combination of SPICE [203] and CIDEr [204] as the reward signal and apply a policy gradient algorithm to optimise the model, with improved results. This research used a Monte-Carlo roll-out strategy to obtain the intermediate reward during the process of description generation. More recently, Bahdanau et al. [205], instead of sentence-level reward in training, applied the token-level reward in temporal difference training for sequence generation.

As discussed previously, the GAN [3] estimates a difference measure using a binary classifier, called a discriminator, to discriminate between the target samples and generated samples. GANs rely on back-propagating these difference estimates through the generated samples to train the generator to minimise these differences. Hence, the whole network in GAN is trained in an adversarial way. The GAN was originally proposed to generate naturalist images [3] [170] [206] [55]. Directly applying a GAN for the language problem is impossible since sequences are composed of discrete elements in many application areas such as machine translation and image captioning.

A possible solution to tackle the discreteness problem of language is to use the Gumbel-Softmax approximation [146] [207]. For instance, Shetty et al. [208] use a GAN to generate more realistic and accurate image descriptions with the aid of Gumbel-Softmax to deal with the discontinuousness issue in language processing. Another more general solution is to borrow an idea from the RL framework, in which the feedback from the discriminator is considered as the reward for the language generator. Dai et al. [181] built a model based on conditional GAN to generate different and naturalistic image descriptions and paragraphs, which utilises a policy gradient for optimisation. Yu et al. [50] proposed a model called SeqGAN, which unified the GAN framework and RL learning problem, this has recently received much attention [53] [209]. They propose a three steps training strategy, which includes the pre-training the generator, pre-training the discriminator and the final adversarial training. In this chapter, inspired by the SeqGAN, a discriminator to applied judge the fitness of the generated image descriptions concerning the image content and apply the policy gradient optimisation technique [172] to train the model. Unlike the original SeqGAN, our discriminator not only cares about the differences between the target

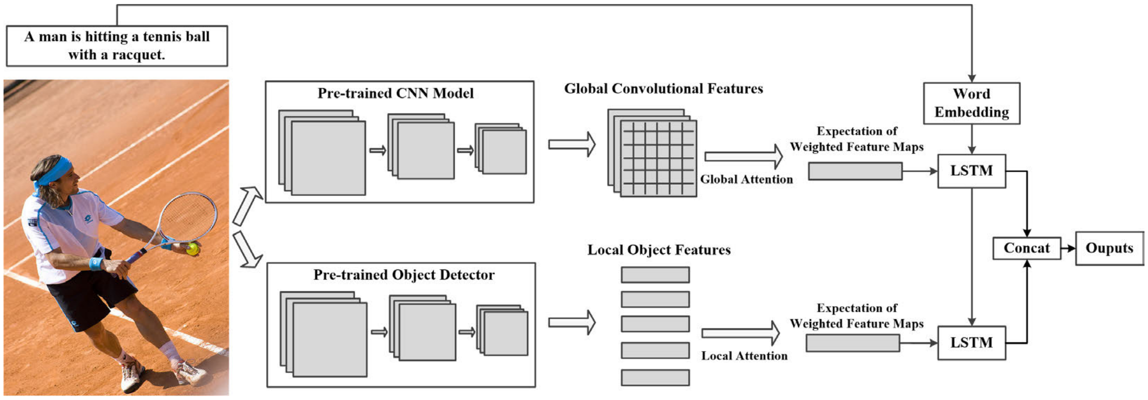

A man is hitting a tennis ball with a racquet.
Pre-trained CNN Model
Global Convolutional Features
Word Embedding
Expectation of Weighted Feature Maps
Global Attention
LSTM
Concat
Ouputs
Local Object Features
Pre-trained Object Detector
Expectation of Weighted Feature Maps
Local Attention
LSTM

Figure 5.5: The hierarchical attention model structure.

language and model-generated language but also considers the coherence of the language with the image content.

5.3.3 Approach

In this section, the proposed method is described based on two parts: the hierarchical attention mechanism and the policy gradient optimisation algorithm.

Hierarchical Attention Mechanism

The hierarchical attention mechanism consists of two parts: a spatial attention mechanism which corresponds to global CNN features and a local attention mechanism which corresponds to object features.

The spatial attention mechanism is based on the model in [4]. Specifically, the model comprises an encoder and a decoder. A convolutional neural network pre-trained on the ImageNet dataset [69] is used to extract a set of convolutional features. These features, denoted as $a = \{a_1, ..., a_L\}$, correspond to certain portions of the 2-D image.

The Long Short-term Memory (LSTM) network, initially proposed by Hochreiter and Schmidhuber in [45], is applied as the language decoder because of its superior performance in natural language processing.

$$\begin{aligned}
i_t &= \sigma(W_{xi} * z_t + W_{hi} * h_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * z_t + W_{hf} * h_{t-1} + b_f) \\
o_t &= \sigma(W_{xo} * z_t + W_{ho} * h_{t-1} + b_o) \\
g_t &= \sigma(W_{xc} * z_t + W_{hc} * h_{t-1} + b_c) \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot g_t \\
h_t &= o_t \cdot \phi(c_t)
\end{aligned} \tag{5.11}$$

In Equation 5.11, i_t, f_t, o_t, c_t and h_t are the input gate, forget gate, output gate, cell memory and hidden state of an LSTM network, respectively. g_t and h_t are the input and the output of the LSTM model. z_t is the context vector, which can be processed by the soft attention mechanism and can capture visual information associated with a certain input location. The soft attention mechanism has to automatically allocate adaptive weights for the image locations to facilitate the task at hand.

$$e_{ti} = f_{att}(a_i, h_{t-1}) \tag{5.12}$$

where $a_i \in \{a_1, ..., a_L\}$. Equation 5.12 actually maps the image features from each location, along with information from the hidden state, into an adaptive weight, which indicates the importance of each image location for the recognition.

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{L} exp(e_{tk})} \tag{5.13}$$

Then, Equation 5.13 normalises the adaptive weights into a probability value in the range of 0 and 1 using the Softmax function. Once these weights (summed to 1) are computed, these weights vector α_t are element-wisely multiplied with image feature vector a and sum them to the context vector z_t, which can be expressed as in Equation 5.14. This can be seen as the expectation of weighted features maps.

$$z_t = \sum_{i=1}^{L} \alpha_{t,i} a_i \tag{5.14}$$

Then the context vector z_t is forwarded to the LSTM network to generate captions, as described in Equation 5.11. This soft attention mechanism can adaptively select the relevant visual parts of the given image features and thus facilitate the recognition.

The local attention mechanism is formulated using object features and another LSTM model. A pre-trained object detector is used to retrieve the top N detected object features, which are denoted as $d = \{d_1, ..., d_N\}$. Another LSTM model with soft attention is applied to allocate adaptive weights to each of these features.

$$e_{ti}^d = f_{att}^d(d_i, h_{t-1}^d) \tag{5.15}$$

where h^d indicates the hidden state of the LSTM model for the local attention mechanism.

$$\alpha_{ti}^d = \frac{exp(e_{ti}^d)}{\sum_{k=1}^{L} exp(e_{tk}^d)} \tag{5.16}$$

Similarly, Equation 5.16 normalizes the adaptive weights for local features to a probability value with the Softmax function.

$$z_t^d = Concat(\sum_{i=1}^{N} \alpha_{t,i}^d d_i, h_{t-1}) \tag{5.17}$$

Equation 5.17 demonstrates that the context vector for local attention model catching information from both the local features and the global attention mechanism, where $Concat$ indicates the concatenation operation of the features. This context vector is then forwarded to a second LSTM model as described by Equation 5.18.

$$
\begin{aligned}
i_t^d &= \sigma(W_{xi}^d * z_t^d + W_{hi}^d * h_{t-1}^d + b_i^d) \\
f_t^d &= \sigma(W_{xf}^d * z_t^d + W_{hf}^d * h_{t-1}^d + b_f^d) \\
o_t^d &= \sigma(W_{xo}^d * z_t^d + W_{ho}^d * h_{t-1}^d + b_o^d) \\
g_t^d &= \sigma(W_{xc}^d * z_t^d + W_{hc}^d * h_{t-1}^d + b_c^d) \\
c_t^d &= f_t^d \cdot c_{t-1}^d + i_t^d \cdot g_t^d \\
h_t^d &= o_t^d \cdot \phi(c_t^d)
\end{aligned}
\tag{5.18}
$$

The two LSTM models, denoted as $LSTM^G$ for the global features and $LSTM^L$ for the local features are jointly trained to map the hierarchical feature representation with language. $LSTM^L$ is at a higher level, which can be used to decode the hidden states for the final outputs. However, the gradient vanishing problem cannot be avoided if only the hidden states from $LSTM^L$ is used to decode information. Inspired by [43] in which a shortcut in network connections is applied to solve the gradient vanishing problem, the hidden states from $LSTM^G$ and $LSTM^L$ are concatenated to be decoded and mapped to language vectors, which can be seen in Equation 5.19.

$$
\begin{aligned}
&h_t^{output} = Concat(h_t, h_t^d) \\
&logits = W_p h_t^{output} \\
&P(s_t|I, s_0, s_1, s_2, ..., s_{t-1}) = Softmax(logits)
\end{aligned}
\tag{5.19}
$$

In MLE training, if the length of a sentence is T, the loss function can be formulated as in Equation 5.20, which is the sum of the log likelihood of each word.

$$
Loss = \sum_{i=0}^{T} \log(p(s_t|I, s_0, s_1, s_2, ..., s_i)) \tag{5.20}
$$

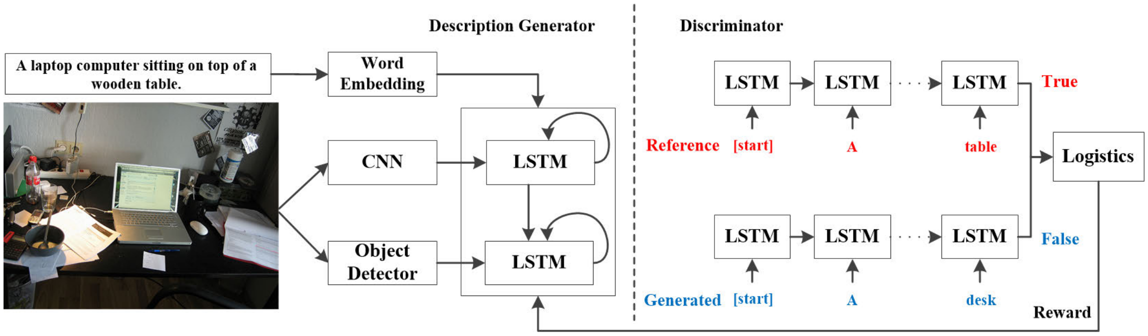


Figure 5.6: Policy Gradient optimization with a discriminator to evaluate the similarity between the generated sentence and the reference sentence.

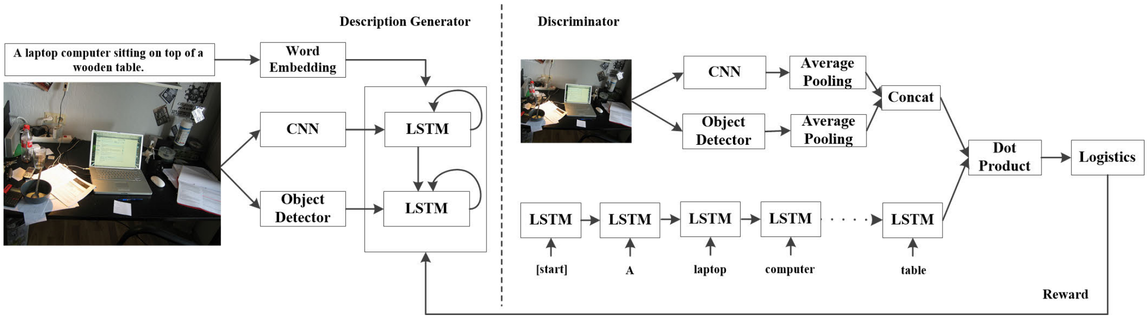


Figure 5.7: Policy Gradient optimization with a discriminator to evaluate the coherence between the generated sentence and the image contents.

Policy Gradient Optimization

In addition to only using the MLE to train the image caption generator, to alleviate the previously discussed exposure bias problem in RNN-based MLE training as discussed previously, a policy gradient optimisation algorithm is also applied in the RL framework to increase the quality of the generated descriptions.

Both of the generated descriptions and the reference descriptions are inputs of the discriminator. The level of coherence of the descriptions and image content is calculated by the dot product, which is forwarded to the discriminator, as described in Fig. 5.7. This operation is to consider the coherence between certain captions (sequences) and corresponding image features, which can make the generated captions more realistic and naturalistic. The reference sequences are labelled as true while the generated sequences are labeled as false during the training of the discriminator. The model is also an LSTM network with Softmax Cross Entropy loss. Hence, the discriminator outputs the probabilities of a sample being true. These probabilities, are then considered as the reward signal in the RL framework, to be utilised in the Policy Gradient algorithm for updating the parameters of the image caption generator.

Following [172], the objective of the policy network $G_\theta(y_t|y_{1:t-1})$ (the image caption generator), is to generate a sequence from the start state S_0 to maximize its expected long-term reward as described by Equation 5.21:

$$J(\theta) = E[R_T|s_0, \theta] = \sum_{y_1 \in Y} G_\theta(y_1|s_0) \cdot Q_{D_\theta}^{G_\theta}(s_0, y_1) \tag{5.21}$$

where R_T is the reward for a complete sequence. $Q_{D_\theta}^{G_\theta}(s, y)$ is the action-value function of a language sequence, which is defined as the expected accumulative reward starting from state s, taking a certain action, and then following policy G_θ.

The action-value function is estimated using the REINFORCE algorithm [70] and considers the probability of being real generated by the discriminator as a reward, which can be defined as in Equation 5.22.

$$Q_{D_\theta}^{G_\theta}(a = y_T, s = Y_{1:T-1}) = D_\theta(Y_{1:T}) \tag{5.22}$$

As can be seen in Equation 5.22, the discriminator only provides a reward for a complete sequence. One should not only care about the reward for a complete tokens but also the long-term reward for the future time-steps since the long-term reward is what is actually needed. Similar to the game of Go [173] in which the agent sometimes give up an immediate interest but cares about the final victory, a similar Monte Carlo roll-out strategy is applied for an intermediate state, i.e., an unfinished sequence. An N-time Monte Carlo search is represented in Equation 5.23.

$$\begin{aligned}
&Y^1_{t+1:T}, ..., Y^n_{t+1:T}, ..., Y^N_{t+1:T} = MC^{G_\theta}(Y_{1:t}; N) \\
&MC =\sim Multinomial(logits)
\end{aligned} \tag{5.23}$$

Where $Y_{1:t}$ is the generated sequence tokens and $Y^n_{t+1:T}$ is the Monte Carlo sampling based on a roll-out policy, which, in our case, is set as the same as the image caption generator for convenience. In reality, any policy can be applied to perform the roll-out operation. *logits* is the output of the LSTM decoder. MC is defined as a sampling procedure from a Multinomial distribution.

If there is no intermediate reward, the Monte Carlo roll-out strategy can sample the future possible tokens N times and average these rewards to achieve the goal of reward estimation, which is described in Equation 5.24.

$$\begin{aligned}
&Q^{G_\theta}_{D_\theta}(a = y_t, s = Y_{1:t-1}) = \\
&\frac{1}{N}\sum_{n=1}^{N} D_\theta(Y^n_{1:T}), Y^n_{1:T} \in MC^{G_\theta}(Y_{1:t}; N), \quad for\ t < T \\
&D_\theta(Y_{1:T}), \quad for\ t = T
\end{aligned} \tag{5.24}$$

The Monte Carlo roll-out strategy can be better visualised in Fig. 5.8.

Once the reward value from the discriminator is obtained, it is ready to update the generator. The goal is to maximise the average reward starting from the initial state as defined in Equation 5.25.

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} V_\theta(s_0|X_i, Y_i) \tag{5.25}$$

where N is the number of samples used for training. The Policy Gradient theorem from [172] can be used and the gradient of the objective function (reward signal) can

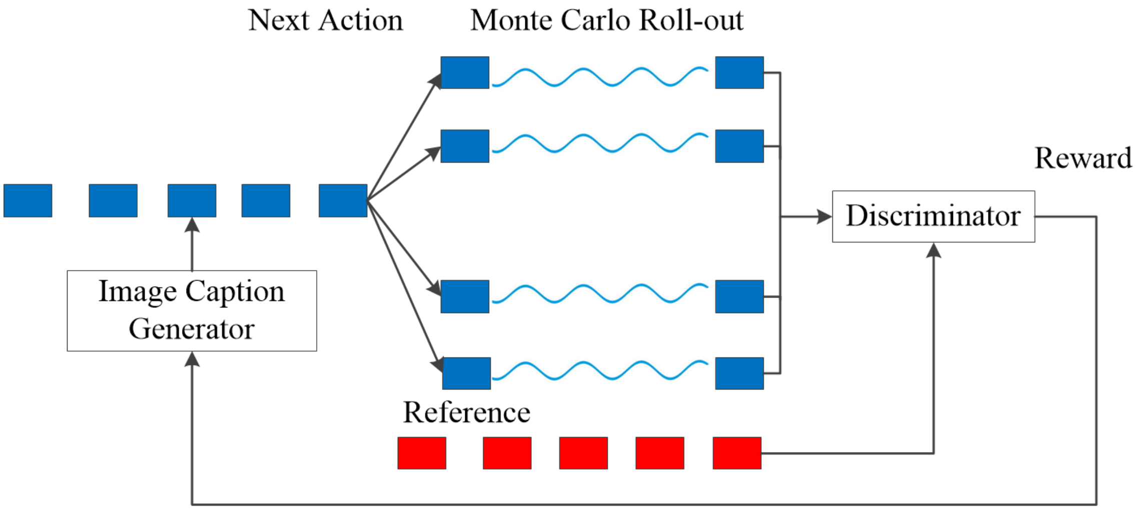


Figure 5.8: Monte Carlo roll-out.

be defined in Equation 5.26.

$$\triangledown_\theta J(\theta) = E_{Y_{1:t-1} \sim G_\theta}[\sum_{y_t \in Y} \triangledown G_\theta(y_t|Y_{1:t-1}) \cdot Q_{D_\theta}^{G_\theta}(Y_{1:t-1}, y_t)] \tag{5.26}$$

Since the expectation can be approximated by sampling, the parameters of the image caption generator can be updated using Equation 5.27.

$$\theta \leftarrow \theta + \alpha_h \triangledown_\theta J(\theta) \tag{5.27}$$

In practice, advanced gradient algorithms such as RMSprop [174] and Adam [141] can be used in training the caption generator.

The image caption generator and discriminator are adversarially trained in the framework of GAN [3]. In GAN [170], the discriminator can pass the gradient directly to the generator. Due to the discreteness of the sequence generation, RL is applied to estimate the gradient of the generator in our model.

Specifically, the training strategy is described in Algorithm 2. the image caption generator is initially pre-trained using MLE. In practice, this is equivalent to the Cross-Entropy loss [175]. Hence, the pre-training step can be set the same as in [4]. The trained model is used to generate some captions which are set as fake samples, which, along with the reference captions, are fed into the discriminator for training. Similarly, the discriminator is also pre-trained for specific steps. The next steps are the adversarial training steps, in which the image caption generator and discriminator have trained alternatively until convergence of the networks.

In addition to the sentence comparison scheme introduced previously and shown in Fig. 5.6, a scheme is also employed to evaluate the coherence between the generated captions and the image content. Specifically, both of the global features and local object features are processed by average pooling in order to obtain fixed-size feature representation, denoted as V_i. The captions, similar to the sentence comparison scheme, are also encoded into a fixed-size vector, using an LSTM model, denoted as V_w. The two vectors V_i and V_w are then dot produced and forwarded to logistic function to obtain the reward for RL training, which can be seen in Fig. 5.7.

5.3.4 Experimental Validation

Dataset Introduction

The experiments were conducted using the COCO dataset [121]. To be consistent with the previous researches, the COCO 2014 released version is used, which includes 123,000 images. The dataset contains 82,783 images in the training set, 40,504 images in the validation set and 40,775 images in the test set. As the ground-truth for the MSCOCO test set is not available, the validation set is further split into a validation subset for model selection and a test subset for social experiments. This is the "Karpathy" split [165]. It utilises the whole 82,783 training set images for training and selects 5,000 images for validation and 5,000 images for testing from the official validation set. The standard evaluation protocol contains BLEU [176], METEOR [199], CIDEr [204] and ROUGE-

Algorithm 2 Image Caption Generation by Adversarial Training and Reinforcement Learning

```
Require: Image Caption Generator $G_\theta$; Discriminator $D_\theta$.
  Pre-training $G_\theta$ using MLE by some epoches.
  Generating negative samples using pre-trained $G_\theta$ to train $D_\theta$.
  Pre-training $D_\theta$ by some steps.
  repeat
    for update-generator for 1 step do
      Generate a sequence $Y_{1:T} = (y_1, .., y_T)$.
      for $t = 1$ to $T$ do
        Compute the intermediate reward $Q(t)$ by Monte Carlo roll-out.
      end for
      Update the parameters $\theta$ using Policy Gradient.
    end for
    for update-discriminator for 1 step or 5 steps do
      Training discriminator $D_\theta$ using reference sequence (True) and generated sequence
      (Fake) using current generator.
    end for
  until Convergence
```

L [200].

BLEU is the most popular metric for the performance evaluation in machine translation. The metric is only based on the n-gram statistics. The BLEU-1, BLEU-2, BLEU-3 and BLEU-4 measure the performance of the 1, 2, 3, 4-gram, respectively. METEOR is based on the harmonic mean of unigram precision and recall and seeks correlation at the corpus level. CIDEr can be used to evaluate the generated sentences with human consensus. ROUGE-L measures the common maximum-length subsequence for the target sentence and the generated sentence.

Implementation Details

For all the images in the COCO dataset, the global convolutional features is obtained (from the layer “res5c”) using a pre-trained Residual-152 network [43] on the platform of Caffe [109], with a dimensionality of 49×2048. The local object features are also extracted using a Faster RCNN [23] object detection network pre-trained on the COCO dataset. Specifically, the top K detected object features are obtained from the layer of “FC6” layer of the VGG16 model [42] used in Faster RCNN, with the dimensionality of $K \times 4096$. The hierarchical attention mechanism and policy gradient optimisation are built on the TensorFlow platform [178].

Training the Faster RCNN on the MSCOCO dataset In order to obtain better local object features, the Faster RCNN model is trained on COCO object detection dataset. The model is first pre-trained on the ILSVRC-2012 object detection dataset [69]. The COCO object detection dataset shares the same images with the image caption task. Consequently, the same splits with the image caption dataset are kept for training. The training process on the COCO dataset is almost the same as the pre-training on ImageNet. The initial learning rate is set to 0.001. The momentum of the stochastic gradient descent is set to 0.9, and the weight decay is set to 0.0005.

Language Pre-processing To pre-process the language, the special symbols such as ‘.’, ‘,’, ‘(’, ‘)’ and ‘-’ are replaced with blank spaces while ‘&’ is replaced with ‘and’. Since the maximum length of the descriptions is set as 20 words, the caption references from the original dataset which are longer than 20 are deleted. For the vocabulary establishment, following the open-source code of [165], words that occur more than 5 times in the vocabulary are included. The symbol ‘NULL’ is mapped to 0, ‘START’ to 1 and ‘END’ to 2.

Training Details of the Model The network was first pre-trained using MLE for ten epochs. During training, the size of the hidden states of the two LSTM models is set as 512. The same size of hidden states of [191] is chosen as they achieved satisfactory performance with this size of the hidden states. The batch size is set as 32 and the learning rate as 0.001, and the Adam algorithm [141] is used to train the network. Subsequently, the discriminator is trained for 2500 steps, followed by an adversarial training scheme, in which the caption generator and discriminator have trained alternatively until convergence. During the pre-training steps of the discriminator and the policy gradient-based adversarial training as described previously, the Adam algorithm is also applied. The learning rate for these steps are set as 0.0001. Following the open-source code of [165], at training time, the maximum length of the input sequence is set to 20 words. During the testing time, alternatively, the maximum length of a generated symbols is set as 30 words. During the training of the proposed model, a trainable word embedding layer is added from Google's TensorFlow platform [178]. All the experiments are conducted on a server embedded with NVIDIA TITAN X GPU and installed with the Ubuntu 14.04 operating system.

Results

Quantitative Evaluation In this section, a comprehensive quantitative evaluation is conducted using different experimental settings on the COCO dataset.

Comparison between the global attention, the local attention and the hierarchical attention model The results are first obtained using only the global attention model, which is similar to the soft attention model in [4]. Since advanced CNN features from the Residual-152 model are used, the results of BLEU, METEOR, CIDEr and ROUGE-L are all satisfactory and are listed in Table 5.4. Then only the local attention model using the detected object features from a Faster RCNN detector is tested, with results which are much lower than those for the global attention model as listed in Table 5.4. One of the possible reasons is that the Faster RCNN only uses the VGG16 model, which is not as powerful as the Residual-152 network. Another reason is that the local object features, despite their capability to provide complementary information to the global attention model, can sometimes miss many essential features. Finally, our proposed hierarchical attention model is tested under MLE training, which utilises both the global and local attention for image captioning. The results improve the baseline significantly, which can be seen in Table 5.4. Specifically, all of the seven evaluation metrics are improved using our hierarchical attention model

The determination of the number of tops detected objects To determine the best number k for the top detected objects in the local attention model, an ablation study is performed. The 10, 20 and 30 top detected object features are extracted and tested using the hierarchical attention model, respectively. The results can be seen in Table 5.5. With the increase of the number k from 10 to 30, the performance increases accordingly. Although the maximum length of our generated sentences is set as 30, not every word represents an object. Also, intuitively, there are a maximum of 30 objects within an image. Hence, in the following experiments, the 30 top detected object features are used for the local attention model.

The performance of Policy Gradient with reward only from language comparison Next, the reinforcement learning steps can be performed. The discriminator which only compares the similarity between the reference sentence and the generated sentence is tested firstly. Specifically, the model defined in Fig. 5.6 is used. The discriminator is first trained in 2500 steps, which is found to be sufficient for the discriminator to converge. The loss curve of the image caption generator is shown in Fig. 5.9. After 2500 steps pre-training the discriminator, the loss of the image caption generator starts to decline, which validates that the policy gradient starts to work. Then the generator and discriminator are further trained adversarially for another 1 epoch and report the results in Table 5.6. Two different settings are tested in the adversarial training steps. The first setting is to train 1 step for the discriminator, followed by another step for the generator. Another setting is to train the discriminator for 5 steps, followed by 1 step training for the generator. The final results of the two setting are similar, which all slightly improve the MLE training baseline. The reason for the improvement is because the reinforcement

Table 5.4: Comparison of image captioning using different attention mechanism results on the COCO dataset

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	ROUGE-L
Soft Attention [4]	70.7	49.2	34.4	24.3	23.90	-	-
Global Attention	70.121	50.304	35.434	25.111	23.658	84.701	54.308
Local Attention	64.059	42.359	28.089	19.033	20.203	56.898	49.861
Hierarchical Attention	**72.611**	**52.769**	**37.802**	**27.243**	**24.731**	**88.140**	**56.048**

learning solves the exposure bias problem during MLE training. However, this scheme lacks the measurement of the similarity between the generated descriptions and the image contents, which prevents the image caption generator from generating more naturalistic and diverse descriptions.

The performance of Policy Gradient with reward from the measurement of coherence between language and image content To train the image caption generator to generate more naturalistic and diverse descriptions, the model defined in Fig. 5.6 is tested. First the global features are extracted and average pooling is performed, resulting with a feature dimension of 2048. The dot product is used to measure these image features and language embedding features by a discriminator, which can be considered as the reward within the reinforcement learning framework. The experimental results from this model can be seen in Table 5.7. However, the results from all of the seven metrics are even lower than the MLE training baseline. One possible reason, is the measurement of discriminator which only uses the global features, which is not consistent with the hierarchical attention model in the generator side. As can be seen from the Table 5.7, the results from this model are similar to that of global attention model, since the reward signal from the discriminator tends to force the generator to produce sentences that only matches the global features. A model exactly like in the one defined in Fig. 5.7 is built. This model includes both of the global image features and the local object features, and thus guarantees that the discriminator and the generator are utilizing the same information source. The final results can be seen in Table 5.7, which outperform all of other experimental settings.

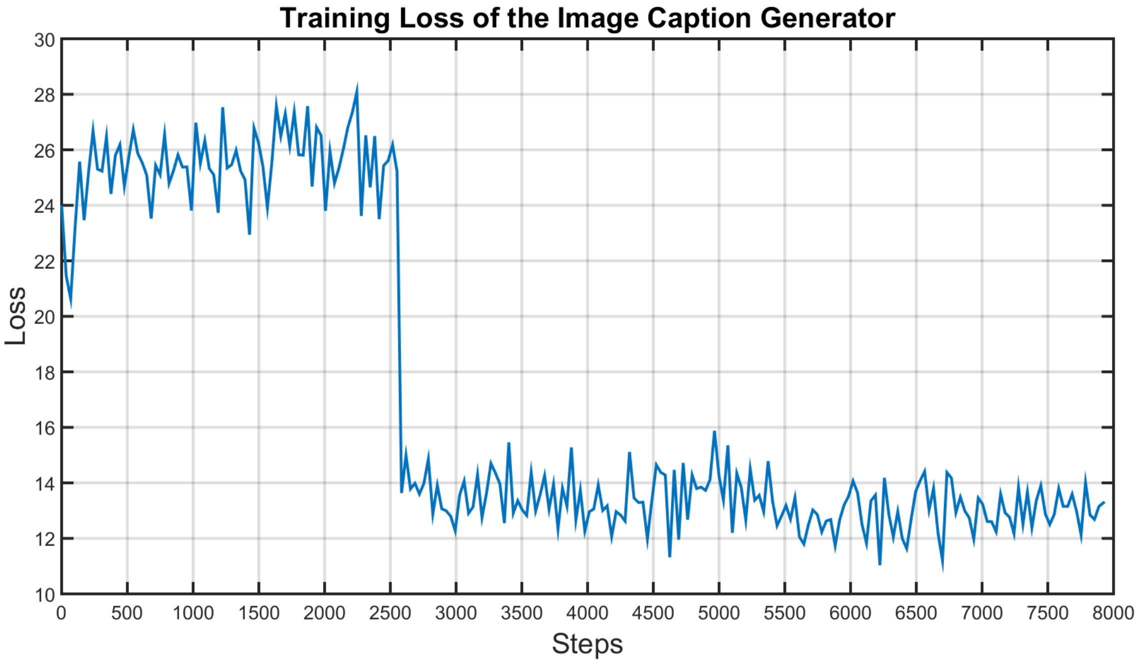


Figure 5.9: The loss curve of the image caption generator during reinforcement learning steps.

Table 5.5: Comparison of image captioning results on the COCO dataset with different numbers of objects

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	ROUGE-L
Hierarchical Attention with 10 Objects for Local Attention	70.601	50.423	36.643	25.389	24.633	87.316	55.241
Hierarchical Attention with 20 Objects for Local Attention	72.159	52.498	37.552	26.918	24.725	**88.639**	55.825
Hierarchical Attention with 30 Objects for Local Attention	**72.611**	**52.769**	**37.802**	**27.243**	**24.731**	88.140	**56.048**

Table 5.6: Comparison of image captioning results on the COCO dataset with different settings for policy gradient (PG) optimization

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	ROUGE-L
MLE training only	**72.611**	52.769	37.802	27.243	24.731	88.140	56.048
PG with 2500 steps for pre-training D followed by 1 D and 1 G step	72.450	**52.845**	**38.141**	27.551	24.543	87.416	**55.876**
PG with 2500 steps for pre-training D followed by 5 D and 1 G step	72.104	52.739	38.122	**27.602**	**24.928**	**89.072**	56.063

Table 5.7: Comparison of image captioning results on the COCO dataset for policy gradient (PG) optimization with discriminator for evaluation of the coherence between language and image content.

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	ROUGE-L
MLE training only	72.611	52.769	37.802	27.243	24.731	88.140	56.048
Global Attention	70.121	50.304	35.434	25.111	23.658	84.701	54.308
PG with similarity of global features (1 D and 1 G step)	72.250	52.290	37.099	26.331	23.815	84.516	55.238
PG with similarity of global features (5 D and 1 G step)	72.234	52.120	36.887	26.065	23.957	84.224	55.244
PG with similarity of global-local features (1 D and 1 G step)	**73.036**	**53.688**	**39.069**	**28.551**	**25.324**	**92.449**	**56.539**

To prove the effectiveness of the proposed method, the final results on the “Karpathy” test split are compared with previously published results, which is shown in Table 5.8. Most of the published results on the “Karpathy” split are shown, which are grouped into three categories. The first category corresponds to various methods without external information and reinforcement learning. The best of them (SCA-CNN-ResNet) is the spatial and channel-wise attention model [72] in which both the spatial and channel-wise attention mechanisms are utilized for image captioning. The methods in the second group use extra information during the training of the model. For instance, Semantic Attention [71] utilizes rich extra data from social media to train the visual attribute predictor. Deep Compositional Captioning (DCC) [211] generates extra data to prove its unique transfer

Table 5.8: Comparison of image captioning results on the COCO dataset with previous methods, where ^{1} indicates external information are used during the training process and ^{2} means that reinforcement learning is applied to optimize the model.

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	ROUGE-L
Google NIC [189]	66.6	46.1	32.9	24.6	-	-	-
m-RNN [192]	67	49	35	25	-	-	-
BRNN [165]	64.2	45.1	30.4	20.3	-	-	-
MSR/CMU [210]	-	-	-	19.0	20.4	-	-
Spatial Attention [4]	71.8	50.4	35.7	25.0	23.0	-	-
gLSTM [193]	67.0	49.1	35.8	26.4	22.7	81.3	-
GLA [191]	56.8	37.2	23.2	14.6	16.6	36.2	41.9
MIXER [201]	-	-	-	29.0	-	-	-
SCA-CNN-ResNet [72]	71.9	**54.8**	**41.1**	**31.1**	25.0	-	-
Semantic Attention^1 [71]	70.9	53.7	40.2	30.4	24.3	-	-
DCC^1 [211]	64.4	-	-	-	21.0	-	-
RL with G-GAN^2 [181]	-	-	30.5	29.7	22.4	79.5	47.5
RL with Embedding Reward^2 [182]	71.3	53.9	40.3	30.4	25.1	**93.7**	52.5
Ours^2	**73.036**	53.688	39.069	28.551	**25.324**	92.449	**56.539**

capability. The third group corresponds to the reinforcement learning technique. RL with G-GAN [181] applies conditional GAN and policy gradient to generate image descriptions. Although their results on the evaluation metrics are not improved, they prove that the generated captions are more diverse and naturalistic. Embedding Reward [182] applies a policy network to generate captions and a value network to evaluate the reward. Additionally, they also apply advanced inference method called lookahead inference and beam search during testing. They achieve the current state-of-the-art results on the "Karpathy" split. Although neither external knowledge nor advanced inference technique (including beam search) are used, similar results are achieved to the current state-of-the-art methods (Embedding Reward [182] and SCA-CNN-ResNet [72]), with state-of-the-art results on three important metrics: BLEU-1, METEOR and ROUGE-L and lead other methods significantly. Note that the Embedding Reward method utilized a ranking loss in the dis-

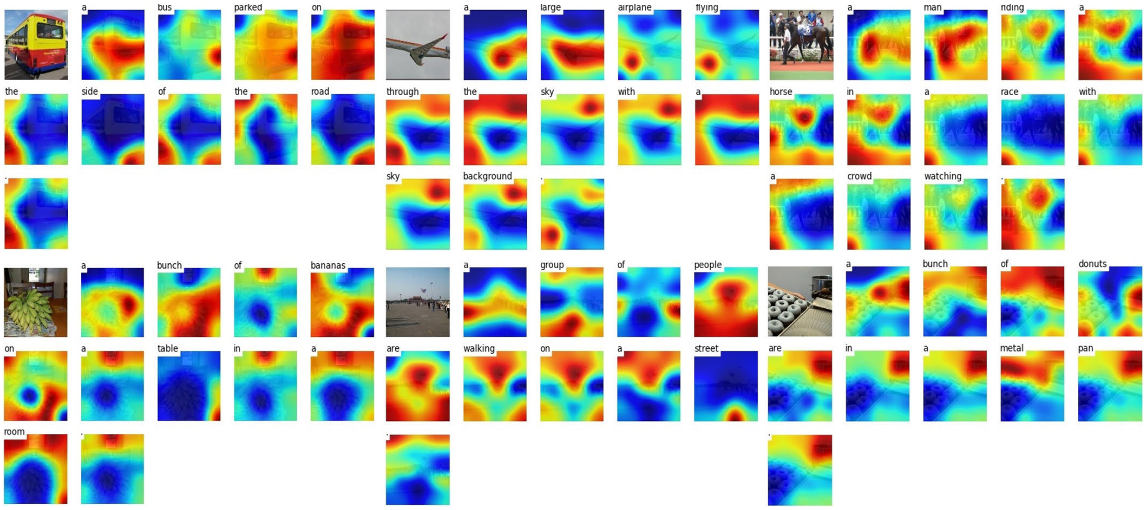


Figure 5.10: Visualization of the global attention maps and generated captions.

criminator to measure how good the generative sentence is, which is a major contribution in the paper [182]. Instead, in our model, we use the simple classification model to measure the coherence and consistency between the generated sentence and the image content, achieving competitive results.

Qualitative Evaluation In addition to the quantitative evaluation using the standard metrics, a qualitative evaluation of the proposed model is performed by visualization. Firstly, some global attention maps corresponding to each generated words as shown in Fig. 5.10. It is obvious in the figure that the attentive regions normally correspond with the semantic meaning of the generated word in each time step. Then some examples are chosen to visualize the local attention weights on the detected objects, which are shown in Fig. 5.11. Only the top 10 detected objects are retrieved because of the limited space, also, the corresponding attentive weights obtained from the local attention mechanism are plotted in the figure. The detector can detect some fine-grained objects, which provide complementary information for the global attention mechanism. At last, some of the generated sentences using different methods are shown. Specifically, the ground-truth sentences,

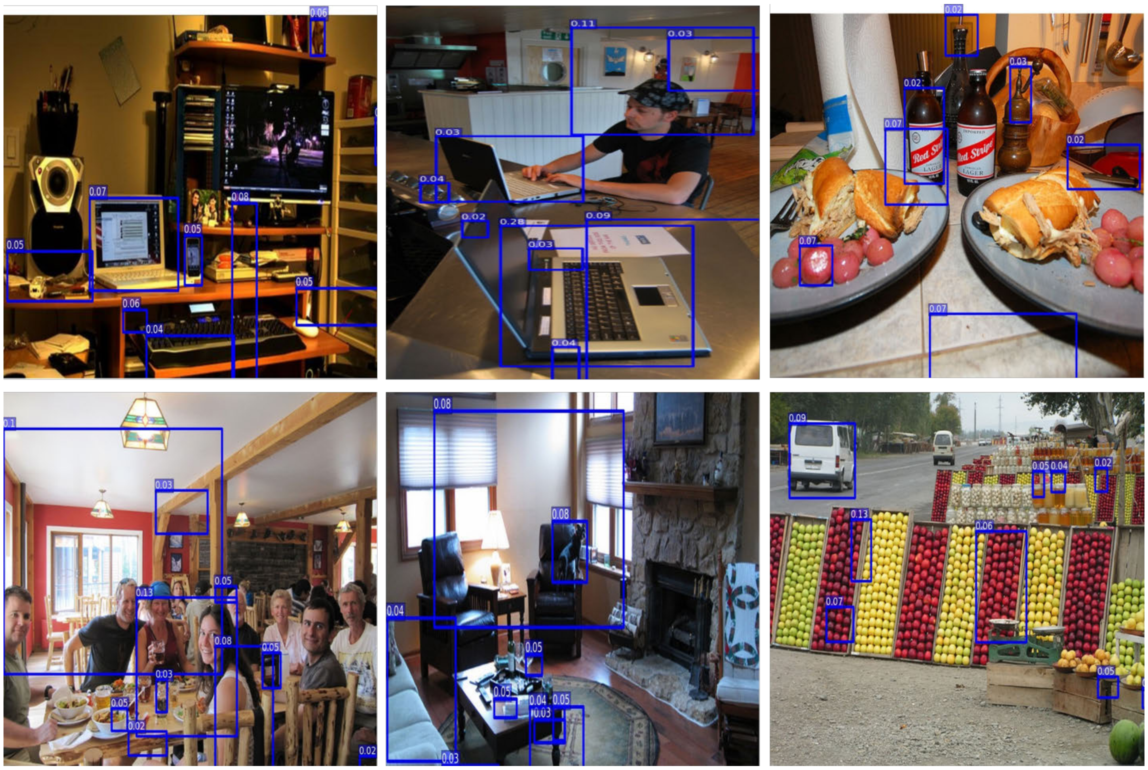
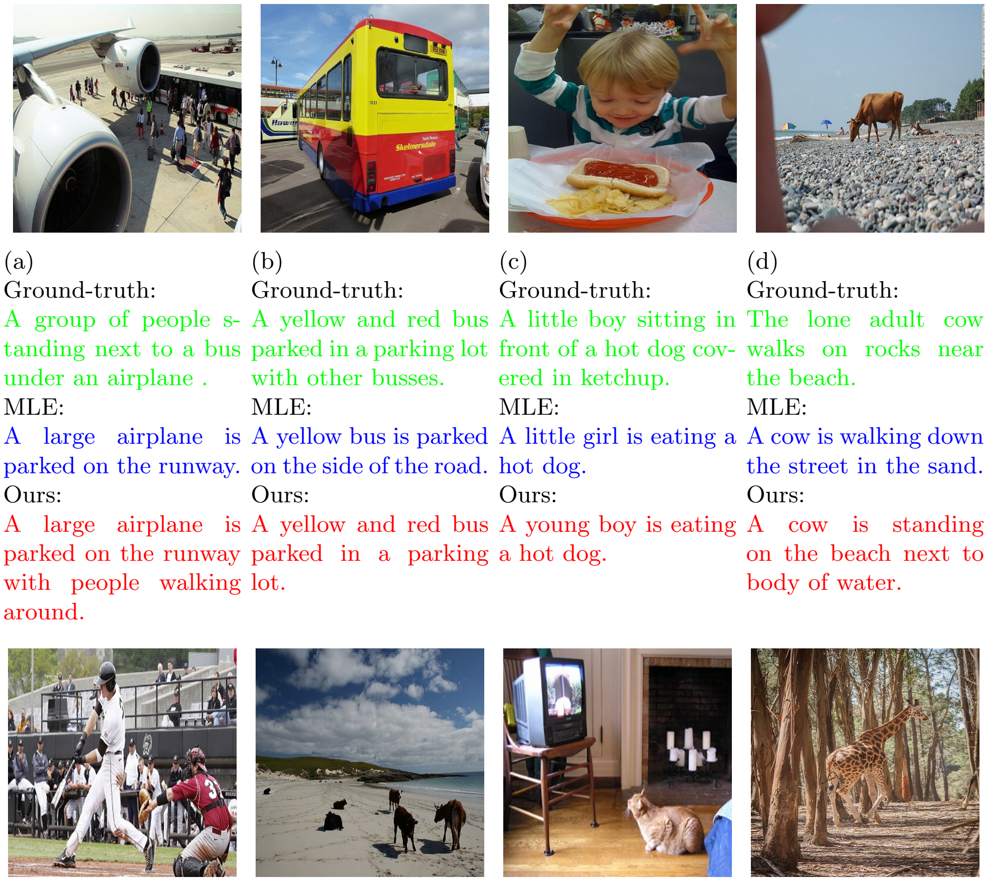


Figure 5.11: Visualization of the attentive weights on the top 10 detected objects.

the descriptions generated by the MLE training-based model and by the proposed model are shown in Fig. 5.12. The text in red are the sentences generated by the proposed model, which are more accurate and naturalist than the MLE-based model, which are shown in blue. Especially, the proposed model show superior performance in finding the fine-grained properties of the image since the RL model automatically measure the coherence of the sentences and the image content. For instance, in Fig. 5.12 (c), the proposed model successfully determines the gender of the person in the image whilst the MLE training-based model gets it wrong.

5.4 Conclusion

This chapter targets the image captioning task, which is a fundamental problem in artificial intelligence. Based on the recent successes of deep learning, especially the CNN feature representation and the LSTM with attention model, the chapter proposes the use of a hierarchical attention mechanism, considering not only the global image features but also detected object features, with improved results. A significant improvement over the current RNN-based MLE training has also been demonstrated. Specifically, a GAN framework with RL optimisation for the image captioning task is proposed to generate more accurate and high-quality captions. The discriminator is to evaluate the coherence and consistency between the generated sentences and image content, thus providing the rewards for optimisation. The whole model follows a three-step training strategy. Experiments analysis confirms the merits of the framework and key contributors to the improved performance. Comparable results with current state-of-the-art methods are achieved using only greedy inference, which proves the effectiveness of the training procedure. Further work will be conducted towards a more robust discriminator and a simple training strategy as the current three-step training method is too complex in practice.



(a)
Ground-truth:
A group of people standing next to a bus under an airplane .
MLE:
A large airplane is parked on the runway.
Ours:
A large airplane is parked on the runway with people walking around.

(b)
Ground-truth:
A yellow and red bus parked in a parking lot with other busses.
MLE:
A yellow bus is parked on the side of the road.
Ours:
A yellow and red bus parked in a parking lot.

(c)
Ground-truth:
A little boy sitting in front of a hot dog covered in ketchup.
MLE:
A little girl is eating a hot dog.
Ours:
A young boy is eating a hot dog.

(d)
Ground-truth:
The lone adult cow walks on rocks near the beach.
MLE:
A cow is walking down the street in the sand.
Ours:
A cow is standing on the beach next to body of water.

(e)
Ground-truth:
A baseball player swinging a baseball bat during a game.
MLE:
A baseball player is preparing to swing at a pitch.
Ours:
A baseball player is swinging a bat at a ball.

(f)
Ground-truth:
Six cows standing and laying on the beach.
MLE:
A group of cows standing on top of a snow covered field.
Ours:
A group of cows standing on top of a sandy beach.

(g)
Ground-truth:
A fat cat in the living room watching the tv.
MLE:
A cat is sitting in a living room with a television.
Ours:
A cat sitting on the floor watching a television.

(h)
Ground-truth:
A giraffe is walking through the forest with tall trees.
MLE:
A giraffe is standing in the woods with trees in the background.
Ours:
A giraffe standing next to a tree in a forest.

Figure 5.12: Visualization of the generated descriptions.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Unlike the standard way of storing visual information in the computer system, human do not focus its attention on the whole scene at its entirety in equal importance. Instead, they sequentially select the essential objects for the task at hand. This visual processing method of human is a very effective mechanism, which, on the one hand, can discriminate the importance and the unimportance, on the other hand, can generally save the processing resource of the human brain. These two aspects are also with vital importance in a machine intelligence system where the computing resources are precious.

In this thesis, motivated to discover the internal mechanism of the attention in computer vision systems, the visual attention mechanism in deep learning are explored and designed for two applications: action recognition and image description generation. The topic of visual attention has long been a hot research topic in computer vision field. However, until recently, the visual attention mechanism in deep learning has attracted high attention and shows promising results in many real-world applications.

In both of the two applications in this thesis, we propose novel models by using various ways of implementing the visual attention, to empower the machine with a more advanced visual processing mechanism, with an improved effect on many standard evaluation metrics in these applications. The proposed methods have good system performance regarding accuracy and are also efficient enough for real-world applications.

A summary of the conclusions of this thesis is given below.

1. In Chapter 1, we first give an overview of the research topics, followed by the motivations and challenges in the research topics and then present the general architecture of the thesis and summarise the main contributions of the thesis.

2. The recent development of machine learning and its applications are introduced, followed by a description of relevant deep learning theories and the recent development of the visual attention mechanism. In Chapter 2, We also presents a comprehensive review of the recent literatures of the related deep learning models.

3. In Chapter 3, a multi-branch attention network is proposed to capture the contextual information to improve the discriminating capability of the neural network for this task. This neural network achieved the state-of-the-art results on several public benchmark datasets on two experimental settings: the location of the target person is available and not available. Especially, the model achieved the 1st place in the PASCAL VOC 2012 action recognition dataset, and the state-of-the-art results on HICO dataset. The proposed model can be further blended with more advanced CNN architecture like Residual-Net [43] to boost the system performance.

4. In Chapter 4, two types of visual attention mechanism, including the soft attention and hard attention, are proposed and combined with a novel hierarchical multi-scale RNN model, for the task of action recognition from videos. The final performance validates that the HM-RNN can capture the long-term dependency and the proposed hard attention mechanism demonstrates a powerful modelling capacity in grasping the critical information. This model can combine with arbitrary video features,

e.g., the dense trajectories. Also, this model can be easily extended to two-stream architecture by building another stream with optical flow features.

5. In Chapter 5, a novel hierarchical attention mechanism and a policy gradient optimisation technique blending with the adversarial training framework, are proposed for the task of image captioning. The hierarchical attention mechanism can reason on both the global image features and local object features while the policy gradient optimisation can compensate the exposure bias problem in the RNN-based language model. The novel architecture demonstrates good system performance, achieved the state-of-the-art results on several important evaluation metrics in the COCO dataset.

6. This thesis comprehensively studies the recent development of visual attention mechanism in computer vision and deep learning. In two application cases: the action recognition and image description generation, the visual attention mechanism has shown powerful modelling capability. Also, several related research topics have been discussed, including the gradient estimation of the discrete unit in neural networks, the long-term dependency in RNNs, the policy gradient optimisation and the adversarial training.

6.2 Future Work

1. **On the improvement of the current visual attention mechanism by using supervision.** The visual attention mechanism in this thesis are all unsupervised attention mechanism, which is learnt during the training process. Future works include the implementation of a supervised or guided visual attention mechanism, which is expected to have improving effect, since the critical information is more accurate with supervision. One of the obstacles of supervised attention mechanism is the lack of the supervision signal. For instance, labelling the important parts of an image in the visual attention mechanism is labour-intensive. One solution is finding a

way to automatically retrieve this supervision signal from other tasks such as object detection.

2. **On the improvement of the long-term dependency of the RNN model in sequence-to-sequence model.** There are still other available approaches in the implementation of an RNN model to enable long-term dependencies, a more formal and theoretical analysis of this type of model is urgently needed, which is included in the future research. The long-term dependency issue in RNN has long been a critical research topic. The HM-RNN used in this thesis is one solution, but with a very complex model structure. Implementing a simple yet effective RNN for long sequence modelling is an urgent need. The discrete unit might still useful in implementing this kind of networks, so are the gradient estimation algorithms. An hierarchical structure for the long sequence, for instance, a paragraph to describe a visual content, is considered to be included in future work.

3. **On the improvement of the Policy Gradient Optimization in sequence-to-sequence model by using Actor Critic.** The Policy Gradient is with high variance during training, a more stable and effective reinforcement learning framework is expected to apply in the RNN-based language model. The future research considers the alternative of the policy gradient method. For instance, the Actor Critic [212] is another type of reinforcement learning algorithm, which trades off the low variance with a biased estimator. In practice, the Actor Critic seems to have a more stable performance than the Policy Gradient-based algorithms [213].

References

[1] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[2] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[4] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2(3):5, 2015.

[5] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[6] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[7] Ronald A Rensink. The dynamic representation of scenes. *Visual cognition*, 7(1-3):17–42, 2000.

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[9] Shiyang Yan, Jeremy S Smith, Wenjin Lu, and Bailing Zhang. Multi-branch attention networks for action recognition in still images. *IEEE Transactions on Cognitive and Developmental Systems*, 2017.

[10] X Liu and M Milanova. Visual attention in deep learning: a review. *Int Rob Auto J*, 4(3):154–155, 2018.

[11] Shiyang Yan, Jeremy S Smith, and Bailing Zhang. Action recognition from still images based on deep vlad spatial pyramids. *Signal Processing: Image Communication*, 54:118–129, 2017.

[12] Shiyang Yan, Jeremy S Smith, Wenjin Lu, and Bailing Zhang. Cham: action recognition using convolutional hierarchical attention model. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 3958–3962. IEEE, 2017.

[13] Shiyang Yan, Jeremy S Smith, Wenjin Lu, and Bailing Zhang. Hierarchical multi-scale attention networks for action recognition. *Signal Processing: Image Communication*, 61:73–84, 2018.

[14] Raymond H Myers and Raymond H Myers. *Classical and modern regression with applications*, volume 2. Duxbury press Belmont, CA, 1990.

[15] Yann LeCun, D Touresky, G Hinton, and T Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pages 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann, 1988.

[16] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1):43–62, 1997.

[17] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] Matthew D Zeiler. *Hierarchical convolutional deep learning in computer vision*. PhD thesis, New York University, 2013.

[20] Matthew D Zeiler and Rob Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European conference on computer vision*, pages 346–361. Springer, 2014.

[22] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[24] Georgia Gkioxari, Ross Girshick, and Jitendra Malik. Contextual action recognition with r* cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1080–1088, 2015.

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[26] Taco S Cohen and Max Welling. Transformation properties of learned visual representations. *arXiv preprint arXiv:1412.7659*, 2014.

[27] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 991–999, 2015.

[28] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[29] Geoffrey E Hinton et al. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA, 1986.

[30] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017.

[31] Edgar Xi, Selina Bing, and Yang Jin. Capsule network performance on complex data. *arXiv preprint arXiv:1712.03480*, 2017.

[32] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[34] David Warde-Farley, Ian J Goodfellow, Aaron Courville, and Yoshua Bengio. An empirical analysis of dropout in piecewise linear networks. *arXiv preprint arXiv:1312.6197*, 2013.

[35] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.

[36] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[37] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[38] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[39] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[40] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[41] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[44] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[45] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[46] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*, pages 2342–2350, 2015.

[47] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, 2014.

[48] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[49] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[50] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858, 2017.

[51] Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. Maximum-likelihood augmented discrete generative adversarial networks. *arXiv preprint arXiv:1702.07983*, 2017.

[52] R Devon Hjelm, Athul Paul Jacob, Tong Che, Adam Trischler, Kyunghyun Cho, and Yoshua Bengio. Boundary-seeking generative adversarial networks. *arXiv preprint arXiv:1702.08431*, 2017.

[53] Matt J Kusner and José Miguel Hernández-Lobato. Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*, 2016.

[54] Xiaodan Liang, Zhiting Hu, Hao Zhang, Chuang Gan, and Eric P Xing. Recurrent topic-transition gan for visual paragraph generation. *arXiv preprint arXiv:1703.07022*, 2017.

[55] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[56] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2015.

[57] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2017.

[58] Radhakrishna Achanta, Francisco Estrada, Patricia Wils, and Sabine Süsstrunk. Salient region detection and segmentation. *Computer vision systems*, pages 66–75, 2008.

[59] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[60] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence*, 33(2):353–367, 2011.

[61] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[62] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *International Conference on Learning Representations*, 2014.

[63] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

[64] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*, 2014.

[65] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119*, 2015.

[66] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Video description generation incorporating spatio-temporal features and a soft-attention mechanism. *arXiv preprint arXiv:1502.08029*, 2015.

[67] Eu Wern Teh, Mrigank Rochan, and Yang Wang. Attention networks for weakly supervised object localization.

[68] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[69] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[70] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[71] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.

[72] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 6298–6306. IEEE, 2017.

[73] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, volume 3, page 6, 2018.

[74] Juan C Caicedo and Svetlana Lazebnik. Active object localization with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2488–2496, 2015.

[75] Kota Hara, Ming-Yu Liu, Oncel Tuzel, and Amir-massoud Farahmand. Attentional network for visual object detection. *arXiv preprint arXiv:1702.01478*, 2017.

[76] E Teh, Mrigank Rochan, and Yang Wang. Attention networks for weakly supervised object localization. BMVC, 2016.

[77] Jianan Li, Yunchao Wei, Xiaodan Liang, Jian Dong, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Attentive contexts for object detection. *IEEE Transactions on Multimedia*, 19(5):944–954, 2017.

[78] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. *arXiv preprint arXiv:1704.06904*, 2017.

[79] Hao Liu, Jiashi Feng, Meibin Qi, Jianguo Jiang, and Shuicheng Yan. End-to-end comparative attention networks for person re-identification. *IEEE Transactions on Image Processing*, 26(7):3492–3506, 2017.

[80] Alireza Rahimpour, Liu Liu, Ali Taalimi, Yang Song, and Hairong Qi. Person re-identification using visual attention. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 4242–4246. IEEE, 2017.

[81] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In *CVPR*, volume 1, page 2, 2018.

[82] Jianlou Si, Honggang Zhang, Chun-Guang Li, Jason Kuen, Xiangfei Kong, Alex C Kot, and Gang Wang. Dual attention matching network for context-aware feature sequence based person re-identification. *arXiv preprint arXiv:1803.09937*, 2018.

[83] Tam V Nguyen, Zheng Song, and Shuicheng Yan. Stap: Spatial-temporal attention-aware pooling for action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(1):77–86, 2015.

[84] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*, volume 1, pages 4263–4270, 2017.

[85] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *Advances in Neural Information Processing Systems*, pages 34–45, 2017.

[86] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[87] Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. Deep semantic role labeling with self-attention. *arXiv preprint arXiv:1712.01586*, 2017.

[88] Patrick Verga, Emma Strubell, and Andrew McCallum. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. *arXiv preprint arXiv:1802.10569*, 2018.

[89] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

[90] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[91] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011.

[92] Gang Yu, Zicheng Liu, and Junsong Yuan. Discriminative orderlet mining for real-time recognition of human-object interaction. In *Asian Conference on Computer Vision*, pages 50–65. Springer, 2014.

[93] Bangpeng Yao and Li Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 9–16. IEEE, 2010.

[94] Christian Thurau and Václav Hlavác. Pose primitive based human action recognition in videos or still images. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[95] Bangpeng Yao and Li Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 17–24. IEEE, 2010.

[96] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 2241–2248. IEEE, 2010.

[97] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1365–1372. IEEE, 2009.

[98] Annabelle Goujon, André Didierjean, and Evelyne Marmeche. Semantic contextual cuing and visual attention. *Journal of Experimental Psychology: Human Perception and Performance*, 35(1):50, 2009.

[99] Guangchun Cheng, Yiwen Wan, Abdullah N Saudagar, Kamesh Namuduri, and Bill P Buckles. Advances in human action recognition: A survey. *arXiv preprint arXiv:1501.05964*, 2015.

[100] Georgia Gkioxari, Ross Girshick, and Jitendra Malik. Actions and attributes from wholes and parts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2470–2478, 2015.

[101] Subhransu Maji, Lubomir Bourdev, and Jitendra Malik. Action recognition from a distributed representation of pose and appearance. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3177–3184. IEEE, 2011.

[102] Chaitanya Desai, Deva Ramanan, and Charless Fowlkes. Discriminative models for static human-object interactions. In *Computer vision and pattern recognition workshops (CVPRW), 2010 IEEE computer society conference on*, pages 9–16. IEEE, 2010.

[103] Vincent Delaitre, Ivan Laptev, and Josef Sivic. Recognizing human actions in still images: a study of bag-of-features and part-based representations. In *BMVC 2010-21st British Machine Vision Conference*, 2010.

[104] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, Oct 2009.

[105] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1331–1338. IEEE, 2011.

[106] R. Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, Dec 2015.

[107] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C.

Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[108] James C Bezdek and Richard J Hathaway. Some notes on alternating optimization. In *AFSS International Conference on Fuzzy Systems*, pages 288–300. Springer, 2002.

[109] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[110] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

[111] Minh Hoai12. Regularized max pooling for image categorization. 2014.

[112] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[113] Li-Jia Li, Hao Su, Li Fei-Fei, and Eric P Xing. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *Advances in neural information processing systems*, pages 1378–1386, 2010.

[114] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010.

[115] Gaurav Sharma, Frédéric Jurie, and Cordelia Schmid. Expanded parts model for human attribute and action recognition in still images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–659, 2013.

[116] Hamed Pirsiavash Luc Van Gool Ali Diba, Ali Mohammad Pazandeh. Deepcamp: Deep convolutional action & attribute mid-level patterns. In *CVPR 2016, IEEE Conference on Computer Vision and Pattern Recognition.*, 2016.

[117] Fahad Shahbaz Khan, Jiaolong Xu, Joost van de Weijer, Andrew D Bagdanov, Rao Muhammad Anwer, and Antonio M Lopez. Recognizing actions through action-specific person detection. *Image Processing, IEEE Transactions on*, 24(11):4422–4432, 2015.

[118] Zhichen Zhao, Huimin Ma, and Xiaozhi Chen. Semantic parts based top-down pyramid for action recognition. *Pattern Recognition Letters*, 84:134–141, 2016.

[119] Y. Zhang, L. Cheng, J. Wu, J. Cai, M. N. Do, and J. Lu. Action recognition in still images with minimum annotation efforts. *IEEE Transactions on Image Processing*, 25(11):5479–5490, Nov 2016.

[120] Yu-Wei Chao, Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng. Hico: A benchmark for recognizing human-object interactions in images. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[121] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[122] Arun Mallya and Svetlana Lazebnik. Learning models for actions and person-object interactions with transfer to question answering. In *European Conference on Computer Vision*, pages 414–428. Springer, 2016.

[123] Guo-ping Ge and Chun-yan Li. Statistics for experimenters. 1978.

[124] Paul R Cohen. *Empirical methods for artificial intelligence*, volume 139. MIT press Cambridge, MA, 1995.

[125] Ronald Aylmer Fisher. *The design of experiments.* Oliver And Boyd; Edinburgh; London, 1937.

[126] Johannes Kaiser et al. An exact and a monte carlo proposal to the fisher–pitman permutation tests for paired replicates and for independent samples. *Stata Journal*, 7(3):402–412, 2007.

[127] Mark D Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 623–632. ACM, 2007.

[128] Rand R Wilcox. *Statistics for the social sciences.* Academic Press, 1996.

[129] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[130] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[131] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[132] Paul J Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1(4):339–356, 1988.

[133] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[134] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[135] Yilin Wang, Suhang Wang, Jiliang Tang, Neil O'Hare, Yi Chang, and Baoxin Li. Hierarchical attention network for action recognition in videos. *arXiv preprint arXiv:1607.06416*, 2016.

[136] Mikel Rodriguez. Spatio-temporal maximum average correlation height templates in action recognition and video summarization. 2010.

[137] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *European conference on computer vision*, pages 392–405. Springer, 2010.

[138] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[139] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM, 2015.

[140] James Bergstra, Frédéric Bastien, Olivier Breuleux, Pascal Lamblin, Razvan Pascanu, Olivier Delalleau, Guillaume Desjardins, David Warde-Farley, Ian Goodfellow,

Arnaud Bergeron, et al. Theano: Deep learning on gpus with python. In *NIPS 2011, BigLearning Workshop, Granada, Spain*. Citeseer, 2011.

[141] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[142] Yu-Gang Jiang, Qi Dai, Xiangyang Xue, Wei Liu, and Chong-Wah Ngo. Trajectory-based modeling of human actions with motion reference points. In *European Conference on Computer Vision*, pages 425–438. Springer, 2012.

[143] Zhenyang Li, Efstratios Gavves, Mihir Jain, and Cees GM Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 2018.

[144] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*, 2016.

[145] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[146] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[147] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *CoRR*, abs/1611.00712, 2016.

[148] Caglar Gulcehre, Sarath Chandar, and Yoshua Bengio. Memory augmented neural networks with wormhole connections. *arXiv preprint arXiv:1701.08718*, 2017.

[149] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork rnn. In *31st International Conference on Machine Learning (ICML)*, 2014.

[150] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[151] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[152] Yu-Gang Jiang, Qi Dai, Wei Liu, Xiangyang Xue, and Chong-Wah Ngo. Human action recognition in unconstrained videos by explicit motion modeling. *IEEE Transactions on Image Processing*, 24(11):3781–3795, 2015.

[153] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[154] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278. IEEE, 2013.

[155] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.

[156] K. Fu, J. Jin, R. Cui, F. Sha, and C. Zhang. Aligning where to see and what to tell: Image captioning with region-based attention and scene-specific contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017.

[157] Emil Julius Gumbel and Julius Lieblein. Statistical theory of extreme values and some practical applications: a series of lectures. 1954.

[158] Chris J Maddison, Daniel Tarlow, and Tom Minka. A* sampling. In *Advances in Neural Information Processing Systems*, pages 3086–3094, 2014.

[159] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012.

[160] Shiyang Yan, Jeremy S. Smith, Wenjin Lu, and Bailing Zhang. Cham: action recognition using convolutional hierarchical attention model. In *Proceedings of the IEEE Conference on Image Processing*, 2017.

[161] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 00:1–9, 2015.

[162] Gul Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[163] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *ICML*, 2015.

[164] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, Dec 2015.

[165] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[166] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663, 2017.

[167] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML15*, pages 2048–2057, 2015.

[168] Kyunghyun Cho, Bart van Merriënboer, Ça?lar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[169] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015.

[170] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[171] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[172] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[173] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[174] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[175] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

[176] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[177] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, volume 29, pages 65–72, 2005.

[178] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[179] Xinlei Chen and C. Lawrence Zitnick. Learning a recurrent visual representation for image caption generation. *CoRR*, abs/1411.5654, 2014.

[180] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1473–1482, 2015.

[181] Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards diverse and natural image descriptions via a conditional gan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2970–2979, 2017.

[182] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. Deep reinforcement learning-based image captioning with embedding reward. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 290–298, 2017.

[183] S. Tang, Y. T. Zheng, Y. Wang, and T. S. Chua. Sparse ensemble learning for concept detection. *IEEE Transactions on Multimedia*, 14(1):43–54, Feb 2012.

[184] C. Kang, S. Xiang, S. Liao, C. Xu, and C. Pan. Learning consistent feature representation for cross-modal multimedia retrieval. *IEEE Transactions on Multimedia*, 17(3):370–381, March 2015.

[185] S. Bu, Z. Liu, J. Han, J. Wu, and R. Ji. Learning high-level feature by deep belief networks for 3-d model retrieval and recognition. *IEEE Transactions on Multimedia*, 16(8):2154–2167, Dec 2014.

[186] P. Liu, J. M. Guo, C. Y. Wu, and D. Cai. Fusion of deep learning and compressed domain features for content-based image retrieval. *IEEE Transactions on Image Processing*, 26(12):5706–5717, Dec 2017.

[187] Sheng Tang, Yu Li, Lixi Deng, and Yongdong Zhang. Object localization based on proposal fusion. *IEEE Transactions on Multimedia*, 19(9):2105–2116, 2017.

[188] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903, 2013.

[189] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[190] Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia*, 17(11):1875–1886, 2015.

[191] L. Li, S. Tang, Y. Zhang, L. Deng, and Q. Tian. Gla: Global-local attention for image description. *IEEE Transactions on Multimedia*, PP(99):1–1, 2017.

[192] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014.

[193] Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. Guiding long-short term memory for image caption generation. *arXiv preprint arXiv:1509.04942*, 2015.

[194] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[195] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515, 2015.

[196] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei. Boosting image captioning with attributes. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4904–4912, Oct 2017.

[197] Anirudh Goyal, Nan Rosemary Ke, Alex Lamb, R Devon Hjelm, Chris Pal, Joelle Pineau, and Yoshua Bengio. Actual: Actor-critic under adversarial learning. *arXiv preprint arXiv:1711.04755*, 2017.

[198] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[199] Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*, pages 65–72, 2005.

[200] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 71–78. Association for Computational Linguistics, 2003.

[201] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

[202] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy. Improved image captioning via policy gradient optimization of spider. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 873–881, Oct 2017.

[203] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

[204] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[205] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. In *International Conference on Learning Representations (ICLR)*, 2017.

[206] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[207] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[208] Rakshith Shetty, Marcus Rohrbach, Lisa Anne Hendricks, Mario Fritz, and Bernt Schiele. Speaking the same language: Matching machine to human captions by adversarial training. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[209] Lijun Wu, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. Adversarial neural machine translation. *arXiv preprint arXiv:1704.06933*, 2017.

[210] X. Chen and C. L. Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2422–2431, June 2015.

[211] L. A. Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell. Deep compositional captioning: Describing novel object categories without paired training data. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–10, June 2016.

[212] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[213] Li Zhang, Flood Sung, Feng Liu, Tao Xiang, Shaogang Gong, Yongxin Yang, and Timothy M Hospedales. Actor-critic sequence training for image captioning. *arXiv preprint arXiv:1706.09601*, 2017.