KU LEUVEN

Image-text Alignment for Artworks

Feiyang Tang

Thesis submitted for the degree of Master of Science in Artificial Intelligence, option Big Data Analytics

Thesis supervisor:
Prof. dr. ir. Marie-Francine Moens

Assessor:
Prof. dr. ir. Tinne Tuytelaars

Mentor:
Ir. Shurong Sheng

Academic year 2019 – 2020

Preface

I would first like to thank my promoter Prof. Marie-Francine Moens for giving me this precious opportunity to conduct this intriguing research under the LIIR laboratory in KU Leuven. Many thanks to my daily advisor Shurong Sheng. The door to her office was always open whenever I ran into a trouble spot or had a question about my research or writing. She consistently allowed this thesis to be my work but steered me in the right direction whenever she thought I needed it.

I would also like to acknowledge Prof. Tinne Tuytelaars as the assessor of this thesis, and I am truly gratefully indebted to her for her very valuable comments on this thesis.

Finally, I must express my very profound gratitude to my mum and to my friends Lieven, Jinjiu, Nanxu and Shuxian for always making me feel at home here in Belgium, providing me with unfailing support and continuous encouragement throughout this year and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. To rest of my family and friends in New Zealand and China, thanks for your comfort calls and endless support. Thank you all.

Feiyang Tang

Contents

Abstract

With the rapid development of the Internet in the past decades, everything around us started to convert to digital gradually. The recent pandemic of COVID-19 made everyone realise the importance and convenience of going digital. For example, while we are staying at home, we can enjoy the masterpieces without leaving the house by merely browsing websites of popular art exhibitions and museums. However, there are many challenges in the area, such as the uncompleted online data and the inadequate online user experience in the museum. To provide users with a more enjoyable experience, we need to effectively annotate artworks to describe artwork image or sub-image with its textual attributes automatically. Therefore, this thesis targets two tasks: coarse-grained level cross modal retrieval and fine-grained cross modal retrieval. Recently, there is an interest in finding all potential alignments between the image area and the word at the same time using cross attention, thereby calculating the similarity of the text. This cross attention based work has also been proved to excel and surpass other techniques in the task of image-text alignment for natural images. We have adopted this attention-based approach to conduct these two tasks on two benchmark datasets and achieve acceptable results. Our presented framework supports annotating artworks in an automatic mode and provides descriptions between image/sub-image and textual attributes in either a coarse-grained or fine-grained level.

List of Figures and Tables

List of Figures

List of Tables

List of Abbreviations

Abbreviations

ILSVRC	ImageNet Large Scale Visual Recognition Challenge
FDDB	Face Detection Data Set and Benchmark
LFW	London Fashion Week Data
RCNN	Region Convolutional Neural Network
SS	Seletive Search algorithm
RPN	Region Proposal Network
YOLO	You Only Look Once algorithm
SOTA	State of the Art (the best)
CNN	Convolutional Neural Networks
VGG	Very Deep Convolutional Networks for Large-Scale Visual Recognition
SCAN	Stacked Cross Attention
ResNet	Residual Neural Network
GRU	Gated Recurrent Unit

Chapter 1

Introduction

1.1 Cross Modal Retrieval

As a part of the millions of internet users who often surf on the internet, we always "Google": enter the keywords to be searched to retrieve our desired text information, which is to retrieve text with text in this case. Sometimes we also upload images on Google to find similar ones, which refers to using an image to retrieve images. However, considering the situation when we use textual information to retrieve images on Google, at this time the type of information we enter and the type of information obtained is different, known as "cross modal".

Cross modal retrieval can be understood as finding the relationship between different modal samples and using a particular modal sample to search for other modal samples with approximate semantics. For example: use the image to retrieve the corresponding text, or use the text to retrieve the desired image. Of course, modals are not limited to images and text, such as voice, physiological signals, and video can be used as components of cross modal retrieval.

The goal of cross modal retrieval is to calculate the similarity between different modal data. For a given query sample, retrieve different modal data related to the query sample. The key challenge lies in the inconsistent representation of different modalities, making it difficult to directly measure the similarity, that is, the "semantic gap" problem. There are two mainstream cross-modal retrieval methods: common space learning and cross-modal similarity calculation.

The common space learning method enables a cross modal similarity to be directly calculated in this space by learning a unified common space for data in different modalities, including methodologies such as classic statistical analysis, deep learning, graph based reduction, metric, ranking, dictionary learning, cross modal hashing, and so on.

Unlike learning from common spaces, cross modal similarity calculation does not learn the common space, but directly calculates the cross modal similarity, for example, using the nearest neighbour algorithm.

At present, the types of multimedia in cross modal retrieval are usually image, text, audio, video and sometimes 3D graphics. Most cross modal retrieval models

are currently focused on the retrieval between image and text, as well as a small numbers concentrate on audio and video. On the one hand, almost no databases are containing all types of modalities; on the other hand, there are "semantic gaps" in the various forms of different types of modalities.

The "semantic gap" problem is a core challenge faced by cross-modal retrieval: data in different modalities have different feature representations, and their similarity is difficult to measure directly. In order to solve the above predicament, an intuitive method is to combine the representations across modalities, that is, to map data in different modalities from their independent representation spaces into a third-party, shared common space so that the similarities among them can be measured. Recently, with the great development and broad application of deep learning, using a common space for different representations based on deep learning has become a popular topic in related research.

1.2 Motivation

Image-text alignment is a fundamental research topic in the inter-field of computer vision and natural language processing. This alignment can be used to two downstream tasks: image annotation and image search. The topic of image-text alignment and cross modal retrieval has been widely discussed in the field of natural images. However, image-text alignment in the cultural heritage domain has not been a lot exploit yet. It can save the intense labour from annotating the artworks for online digital artwork archives if we can automatically describe an artwork image or sub-image with its textual attributes. Furthermore, this topic can help to boost the multi-modal question answering performance in the cultural heritage domain by providing fine-grained image-text correspondence information [30]. Therefore, it is interesting to explore the methods that can figure out the artwork image or sub-image and text correspondence.

While a large number of papers discussed aligning image-text and coarse-grained modal information retrieval, the fragment level image-text alignment problem has not been as widely dealt within the multi-modal question-answering research domain. Coarse-grained modal information retrieval can retrieve information between images and sentences. However, it sometimes does not work well on some artwork datasets, which generally contain some fine-grained patterns and objects in one artwork image, therefore decrease the effectiveness of the retrieval model. We provide demonstrations on the difference between these two levels of cross modal retrieval task in the following section.

1.3 Dataset

The datasets involved in this thesis research are from Sheng et al.'s work [31] which were originally collected from the following online sources: the Brooklyn Museum [21], the Metropolitan Museum [22], and the British Museum [20]. Based on these sources, there are two artwork datasets that we used here: the ancient Egyptian art

image dataset and the ancient Chinese art image dataset. According to [31], the two datasets from Sheng et al.'s work were "*collected based on the geographical location of the origin of the artworks because caption words may differ much depending on the cultural background of the location*". More details of these two datasets are displayed below in Table 1.1.

Dataset	Num. of Artworks	Aver. Length	Num. of Tokens
Egyptian	16,146	9	10,694
Chinese	6,847	10	4,721

Table 1.1: Statistics of Our Datasets [31]

The datasets have 22,993 ancient artwork images. Images are stored at varying dpi and the compressed `jpeg` image file size ranges between 20 to 300 KB. As stated in [31]: "*the paragraph-level descriptions were split into multiple sentences and a maximum of five sentences are retained for each artwork to reduce data imbalance. Duplicate images were removed by [31] in the datasets based on their hash code. Tokens occurring less than two times were removed by the authors from the training vocabulary.*" The datasets were separated into partitions of 80%, 10%, and 10% for respectively training, validation, and test uses.

Here we focus on ancient Egyptian and Chinese artworks; they consist of 16,146 images from Egyptian domain and 6,847 images in Chinese. Figure 1.1 and Figure 1.2 show three examples of artworks from the Egyptian and Chinese collection with textual attributes.

1.3.1 Examples of Cross Modal Retrieval under Different Levels

In this section, we look at two simple examples of ancient Egyptian artworks and how textual description can be retrieved to match its image under two coarse-grained and fine-grained levels.

Figure 1.3 shows a porcelain cup with green pattern on it. Using a coarse-grained multi-modal retrieval model, we can retrieve the textual description sentences "*Porcelain stem cup with polychrome overglaze enamels*", which is sufficient for recognising this artwork. However, this retrieved result does not contain more detailed information such as "*there are two men in independent scenes contemplating the moon in a landscape setting with inverted plantain leaves around the stem and a further figure inside the cup in a red double ring medallion which is much worn*".

In the real-world scenarios, there is much more likely for us to encounter an artwork showing in Figure 1.4. Our traditional coarse-grained multi-modal retrieval model retrieves "*a red and a white pot*" for this artwork but it is not detailed and did not cover sufficient information in the artwork image. Therefore, this motivated us to propose a fine-grained multi-modal retrieval model which can focus on the fragment level image/sentence retrieval. The description on the image was retrieved by our fine-grained multi-modal retrieval model, which has significantly more detailed information.

- female figure
- jar with boat design in a nearby case
- boat design figure with upraised arms
- human noses
- source of the breath
- paint on the male
- dark patch
- white paint
- human trait
- high-status individuals
- three figurines
- white skirts

- fragment of the feet
- base of a statue
- new development
- religious practices
- divine image
- time nonroyal individuals
- hermopolis as the location of the temple
- realistic rendering of each toe
- closer examination of the sculptor
- break in this fragment
- attention to the realistic rendering
- arch of the foot

- statue of ity-sen
- ancient egyptian sculptors
- youthful bodies in a limited numbe
- subject with the left common standing pose
- two legs
- unobstructed view
- most hieroglyphs
- two-dimensional counterparts
- three-dimensional hieroglyphs

Figure 1.1: Examples of Artworks of Egyptian Artwork Dataset

- bronze alter vase in the shape of a zun
- decorated with figures and flowers
- with detatchable handles in the shape of animals

- bottle
- dragon and cloud scrolls
- made of blue underglaze porcelain

- green dragons among clouds and waves on yellow ground
- vase
- made enamelled ceramic, porcelain, engraved

Figure 1.2: Examples of Artworks of Chinese Artwork Dataset

1.4 Research Questions

In this thesis, we study image-text alignment for artworks in coarse and fine-grained levels. Fine-grained image-text alignment refers to the fragment level cross modal re-

Figure 1.3: Ancient Egyptian Artwork Example (coarse-grained)

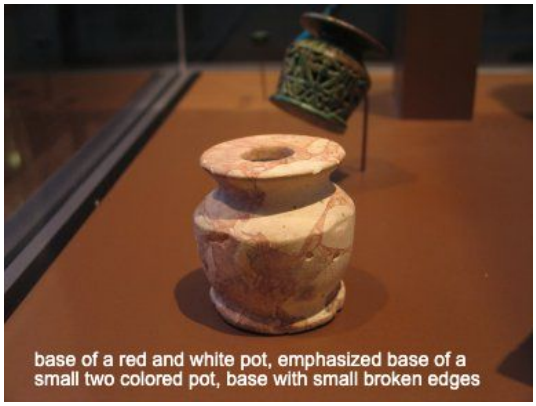


Figure 1.4: Ancient Egyptian Artwork Example (fine-grained)

trieval. The prior sections list the background and motivations, the specific research questions that we look at are:

- Is the image-text alignment model experimented on natural images effective for artwork datasets?
- Can coarse-grained cross modal retrieval model be adapted to fine-grained retrieval and how?

1.4.1 Cross Modal Retrieval Framework

As mentioned above, our primary research task here is to achieve cross modal retrieval (i.e. between image and text) for artworks. Figure 1.5 illustrates a brief working framework for the tasks in a fine-grained level.

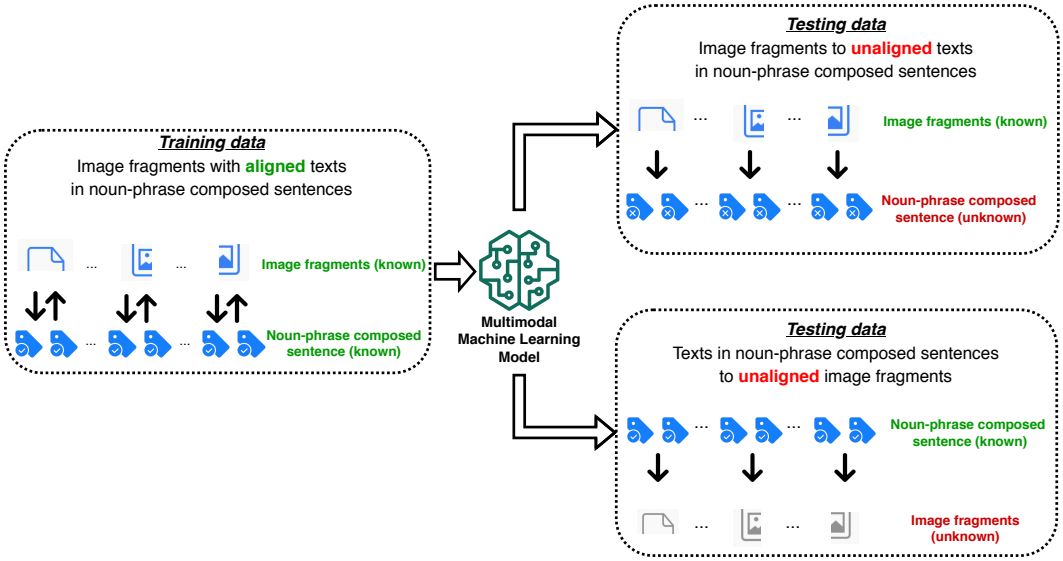


Figure 1.5: Cross Modal Retrieval Process (fine-grained)

We train our multi-modal machine learning model on the known image features and corresponding textual attributes; this training process helps us learn the potential relationships between image and text. This trained model will be used to retrieve textual attributes from known image features and vice versa. We believe this may help with automating the artwork annotation process and significantly save labours on manually annotating artwork information.

1.5 Contributions

The main contributions made by this thesis are:

1. We review work from several disciplines which may be of relevance to the present subject of inquiry and provide commentary on how the findings from these disciplines may be useful. (Section 2)

2. We adopted SCAN [15] as the coarse-grained cross modal retrieval model, analysed its structure and applied it on our proposed Egyptian and Chinese artworks datasets to achieve the image-text alignment. By employing this model, we are able to have a coarse-grained alignment between artworks image and textual attributes. This lays the foundation our novel task of fine-grained cross modal retrieval for artwork items. (Section 3)

3. By focusing on fragment level image features and textual attributes instead of feeding the whole images and sentences, we are able to perform cross modal retrieval in a more fine-grained level. This allows the future multi-modal retrieval tasks on artworks to achieve more detailed results. (Section 4)

1.6 Structure of Thesis

This thesis is structured into the following chapters:

- *Chapter 1 Introduction*
  We provide the reader with a relevant background to understand this thesis.

- *Chapter 2 Related Works*
  We introduce relevant research in image recognition, deep learning, object detection, natural language processing and image-text alignment. In particular, we detail seminal research and review the overall state of the current research. We also review the difference in the works pertaining to the traditional visual-semantic alignment technique versus the more recent cross attention image-text alignment framework.

- *Chapter 3 Coarse-grained Cross Modal Retrieval*
  We introduce our coarse-grained cross-modal retrieval modal - SCAN, discuss how its components interact with each other and explain how SCAN uses cross attention to improve image-text alignment. We also show the preliminary result running SCAN on our ancient Egyptian and Chinese artwork datasets.

- *Chapter 4 Fine-grained Cross Modal Retrieval*
  We proposed our novel fine-grained cross modal retrieval model, which now focus more on the fragment level image-text alignment. We then perform several experiments on evaluating the effectiveness of our image generation from text and vice versa by the recall. We also point out the direction of possible future improvements by discussing several recent related publications.

- *Chapter 5 Conclusion*
  We conclude the work and add some final reflections and remarks.

Chapter 2

Related Works

Image-text alignment is a fundamental research topic in the inter-field of computer vision and natural language processing. There are many approaches proposed to associate images with textual attributes or vice versa. However, the fielded applications on bidirectional image sentence mapping appear to be relatively few, especially for multi-modal question answering.

It has been suggested that this is due to the intense labour has been paid on annotating artworks for online digital artwork archives, automated image or sub-image with its textual attributes description could significantly improve the payoff. In this chapter we will survey and summarise the literature of image-text alignment and some proposed applications on multi-modal question answering.

The structure of this chapter is as follows. Section 2.1 discusses the history and some basic knowledge of image recognition. Section 2.2 discusses the preliminaries on deep learning and some mainstream object detection techniques. Section 2.3 explains the definition of image-text alignment and related proposed solutions on that task. Section 2.4 summarises this section.

2.1 Image Recognition

The history of research on image recognition stems from the 1960s when Marvin Minsky, also known as "the father of Artificial Intelligence" asked his student Gerald Sussman to "connect a camera to a computer and do something with it" [13]. But with minimal resource, this topic did not get enough attention at first.

2.1.1 Early Researches on Image Recognition

After entering 1970s, the advent of modern electronic computers gives computers a chance to try to answer what they see through images. Researchers first tried to learn from the same way human look at things. It was generally believed that humans could see and understand things because they could observe things in 3-D with two eyes, which now seems rather absurd. Therefore, researchers believed that for a computer to understand the image it sees, it must first recover the three-

dimensional structure of the object from its two-dimensional image. This is the so-called "three-dimensional reconstruction" method.

Another inspiration is that it was believed that people could recognise an object, for instance: an apple because people already have a priori knowledge: "Apples are red, round, and smooth". If a machine was also established with such a knowledge base, then it could match the images with its knowledge base, and potentially comprehend what it sees corresponding to what it already knew. This is the so-called "a priori knowledge base" method. However, this method can only extract very few basic features, which is not very practical.

By the 1990s, image processing hardware technology had made huge progress. Meanwhile, researchers began to design different algorithms: introduction of statistical methods and local feature descriptors, which led to more significant development of computer vision technology and started to be widely used in the industries. In the "a priori knowledge base" method, the shape, colour, surface texture, and other characteristics of objects are affected by the viewing angle and the observation environment, and they will change under different angles, different lights, and different occlusions. To solve that dilemma, the proposed new method judges things through identification of local features, and establish a local feature index on objects, which can be more accurately matched even if the perspective or observation environment changes.

After entering the 21^{st} century, computer vision develops rapidly thanks to the massive data brought by the rise of the Internet, the advent of digital cameras, and the widespread application of machine learning methods. In the past, many rule-based processing methods have been replaced by machine learning: machines automatically summarise the characteristics of objects from massive data then identify and classify. Many applications are emerging at this stage, including camera face identification, security face recognition and license plate recognition, etc. The accumulation of data has also produced many evaluation data-sets, such as official face recognition and face comparison recognition platforms: *FDDB* and *LFW*. One of the most famous ones is *ImageNet* [6], which contains 14 million labelled images divided into tens of thousands of categories.

2.1.2 Image Recognition with Neural Networks

After 2010, with the power of deep learning, computer vision technology has experienced explosive growth and industrialisation. With the adoption of neural networks, image recognition tasks can be achieved much more efficient and accurate. Figure 2.1. below gives us a straightforward illustration of how neural networks help with image recognition.

Next, we introduce deep neural networks and how neural networks can be adopted to solve object detection tasks, which is the core part of our project.

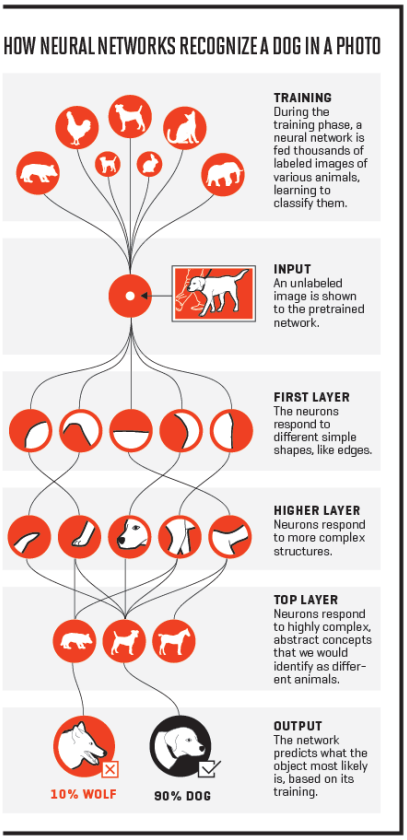


Figure 2.1: How Neural Networks Performs an Image Recognition Task [26]

2.2 Deep Learning and Object Detection

Through deep neural networks, the accuracy of various types of visual recognition tasks has been dramatically improved. In the well-known computer vision competition ILSVR, the error rate of thousands of object recognition was as high as 25.8% in 2011. After the introduction of deep learning in 2012, the error rates in the following four years reached 16.4%, 11.7%, and 6.7%, 3.7%, with significant breakthroughs. Now, face recognition can even achieve a false positive rate of less than one over a million.

Now we know that deep learning has several advantages on image processing and always surpasses other mainstream techniques. But what is deep learning? The following paragraphs give a brief idea of deep learning and how it works.

2.2.1 Deep Learning

In real life, human beings can often solve many problems by intuition. For example, when a human sees Figure 2.2. below, he or she can immediately know that there is a cat and a dog in the picture.

Figure 2.2: Image of a Cat and a Dog

This may feel a natural task for human, but think about how do human know that there are a cat and a dog, but not two cats or two dogs instead? As human can differentiate these two from the picture at a glance, let's try to describe the appearance of cats and dogs. Taking the above picture as an example, we can describe the morphological characteristics of cats as follows: it has a round head, wide cheeks, wide ear roots, deep auricles, and rounded parts at the tip. For the dog in the picture above, its head is flat and wide; the ears are small and thin, the tips of the ears are slightly rounded, the dark apricot eyes, and short hair. Words "broad cheeks" and "round ear tips" often apply to cats and dogs, while the length of short hair and round ears cannot be quantified with a specific number. When we try to adopt a more specific description like "dark apricot eyes", a new problem arises: not all breeds of dogs have such characteristics, but we can still recognise them quickly at a glance.

In short, it is challenging to distinguish cats and dogs accurately with a few words or sentences, but this problem is often solved quickly and accurately by intuition. To solve these seemingly intuitive problems on computers, it is difficult to describe them with specific language or mathematical rules. This brought the invention of deep learning.

Deep learning is to let computers simulate human cognitive processes and learn from experience (also as known as "intuition"). Make computers understand tasks using a hierarchical concept system like human while each concept is defined by some relatively more straightforward concepts: building simpler concepts to learn complex concepts. The word "deep" means more layers of learning system.

Deep learning has a long and rich history. With the increasing amount of available training data and the continuous improvement of computer software and hardware, the scale of deep learning models has also increased to solve increasingly complex application problems, and the accuracy has continued to increase.

2.2.2 Image Understanding

How to retrieve information out from images that can be understood by a computer has always been the most discussed field of computer vision. Deep learning model have become a popular research area due to its powerful representation capabilities, coupled with the accumulation of data and advances in computing power.

But how to understand an image? There are three main levels according to the needs of subsequent tasks: classification, detection and segmentation. Figure 2.3. below provides an example how image understanding tasks can be performed [25].

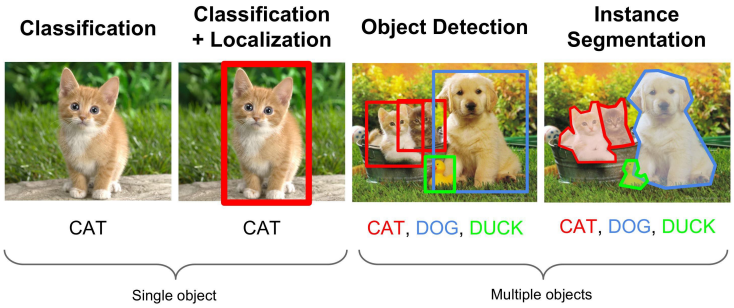


Figure 2.3: Three Steps of Image Understanding [25]

Classification

Classification is the first task, is to structure an image into information of a certain category then describe it by a predetermined category (string) or instance ID. Classification task is the simplest and most basic image understanding task, and it is also the first task for deep learning models to achieve breakthroughs and achieve large-scale applications. In the application field, face recognition and scene recognition can be classified as classification tasks.

Detection

The classification task provides a description of the content of an entire image, while the detection process concentrates on a specific targeting object, which requires the categorical and position information of the target to be obtained together. Comparing to the classification task, detection task illustrates an idea of both foreground and background of the image. To perform detection task, we need to distinguish the desired targets from the general background and identify its description, i.e. type and location of this target. Therefore, the output of this detection model should be a list, each item of this list provides the type and location of this detected target, which is commonly represented by the coordinates of a rectangular detection frame.

Segmentation

Segmentation includes semantic segmentation and instance segmentation. The former is an extension of the previous background separation and requires the separation of image parts with different semantics, while the latter is an extension of the detection task and requires the outline of the target which is finer than the detection frame. Segmentation is a pixel-level description of an image. It provides meaning behind each pixel type, such as the segmentation of roads and non-roads in driver-less driving technology.

2.2.3 Object Detection

After familiarised the steps of image understanding, in this section we focus on the second task: object detection, which is also the main task of our proposed work. The following paragraph will provide several mainstream models on solving object detection tasks.

CNN

First we start with introducing a series of commonly used models called convolutional neural networks (CNN).

Convolution Process The convolution process is based on a small matrix, that is, a convolution kernel. The pixel matrix of each layer mentioned above is continuously scanned in steps, the scanned number is multiplied by the number of the corresponding position of the convolution kernel, and then the sum is calculated. Each time you scan, you get a value, and after all the scans, a new matrix is generated. The following Figure 2.4 illustrates an example of convolution process in a CNN model:

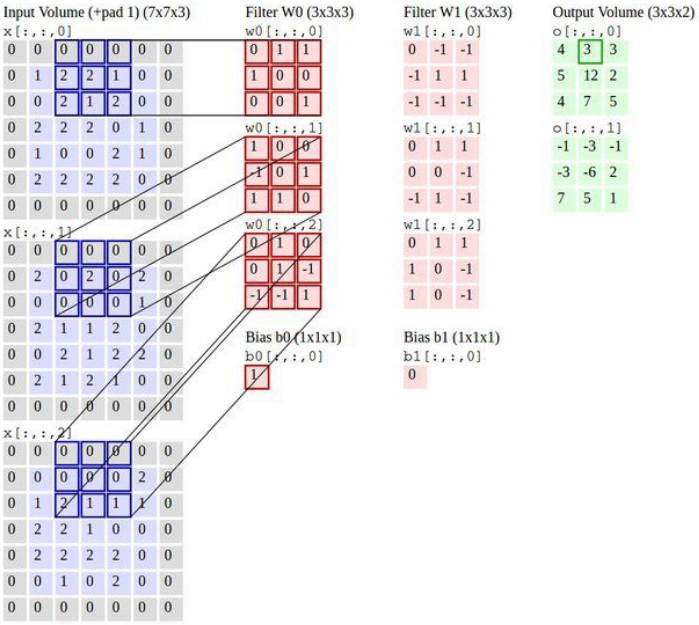


Figure 2.4: Example of Convolution Process [24]

How to determine the number of convolution layers? Normally we take a small matrix of $(3, 3)$. Each value in the convolution kernel is the neuron parameter (i.e. weight) that we need to find (i.e. train). At the beginning, there will be an initial value randomly. When training the network, the network will pass after These parameter values are continuously updated to the propagation until the best

parameter value is found. But how do we know which is the "best", this is evaluated by a loss function.

The step size of the convolution kernel refers to the movement of the convolution kernel by several grids at a time, with horizontal and vertical directions.

The convolution operation is equivalent to feature extraction, and the convolution kernel is equivalent to a filter to extract the features we need.

Padding After the convolution operation, the dimensions become smaller, and the resulting matrix is smaller than the original matrix, which is difficult to calculate and hard to perform convolution, thus, we need padding.

Before each convolution operation, we need to wrap a layer of 0 outside the original matrix. It is possible to only fill in horizontally, or only vertically, or 0 on all sides, so that after convolution, the size of output image will be consistent with input image. Figure 2.5 below illustrates an example of padding zeros to an image.

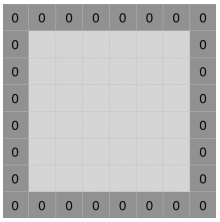


Figure 2.5: Example of Zero-padding Added to Image [5]

Pooling After the convolution operation, we extracted a lot of feature information. Adjacent areas have similar feature information and can be replaced with each other. If all these feature information are retained, there will be information redundancy, which increases the computational difficulty. At this time, pooling is equivalent to a dimension reducing operation.

Pooling happens in a small matrix area. The maximum or average value of the area will be used to replace the area. The size of the small matrix can be set when the network is built. This small matrix scans from the upper left corner to the lower right corner to perform pooling.

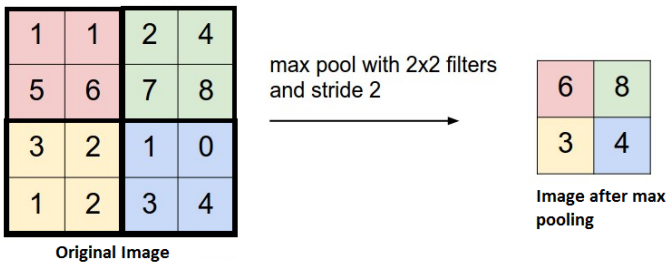


Figure 2.6: Example of Max-pooling [5]

Fully Connected Layer For layers $n-1$ and n, any node in layer $n-1$ is connected to all nodes in layer n. That is, when each node of the n-th layer performs calculations, the input of the activation function is the weight of all nodes of the $n-1$ layer. The middle layer like below is fully connected.

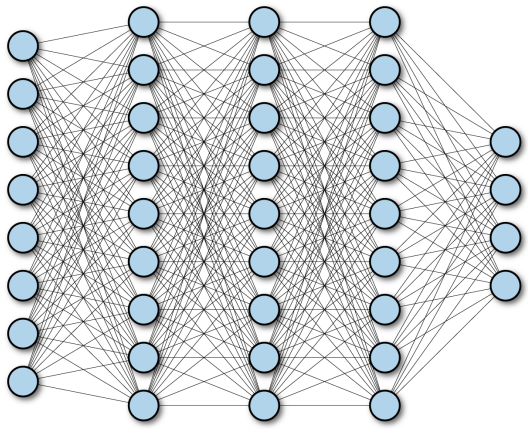


Figure 2.7: Example of Fully Connected Layer

VGG16

`VGG16` [33] is a deep network model developed by the computer vision team of Oxford University and researchers at Google DeepMind in 2014. The network has a total of 16 training parameters. The `VGG16` network won the second place in the ILSVRC 2014 competition classification project and the first place in the positioning project, which proved its asset and made it a very commonly used model in the field of CNN.

Configuration `VGG` has a relatively simple structure, and its generalisation performance of migrating to other image also achieves well. `VGG` is still often used to extract image features.

Conforming to the different sizes of the convolution kernel and numbers of convolution layers in a VGG model, there are six different `ConvNet` configurations: `A`, `A-LRN`, `B`, `C`, `D`, and `E`, where `D` and `E` are more commonly discussed, named `VGG16` and `VGG19` respectively. Figure 2.8 displays the six different structural configurations of VGG. In Figure 2.8, each column corresponds to a structural configuration. For example, section `D` in the figure indicates the structure adopted by `VGG16`.

`VGG16` contains the following components:

- 13 convolutional layers (`conv3-XXX`)
- 3 fully connected layers(`FC-XXXX`)
- 5 max-pooling layers (`maxpool`)

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 2.8: ConvNet Configuration of VGG [33]

Among these different layers, the convolutional layer and the fully connected layer have weight coefficients, so they are often named weighted layers. The total number of weighted layers is $13 + 3 = 16$, which represents the digit "16" in the name of `VGG16` model. (The pooling layer does not involve weights, which means pooling layers were excluded for counting).

Structure and Features The outstanding feature of `VGG16` is simplicity, which is reflected in:

- All the convolutional layers use the identical convolution kernel parameters.
- `conv3-XXX` represents convolution layers, where `conv3` indicates that the size of the convolution kernel is 3, that is, the width and height are both 3. And 3×3 is very small kernel size, combined with other parameters: `stride = 1`, `padding = same`, so that each convolutional layer (tensor) can maintain the same width and high. `XXX` represents the number of channels of the convolutional layer.
- The pooling layer uses the same pooling kernel parameters.

- `VGG16` is constructed with several stacked convolutional layers and pooling layers, which makes it fairly easy to build a deep network structure (considering that back to 2014, sixteen layers were considered very deep).

Based on the above explanations, `VGG16` becomes popular and surpasses many other models because of its small filters and deeper network architecture.

Figure 2.9 illustrates the overall structure of `VGG16`. From left to right, a coloured picture is the input to the network. The white box is the convolution layer, the red is the pooling, the blue is the fully connected layer, and the brown box is the prediction layer. The role of the prediction layer is to convert the information output from the fully connected layer into the corresponding class probability, and play a classification role.

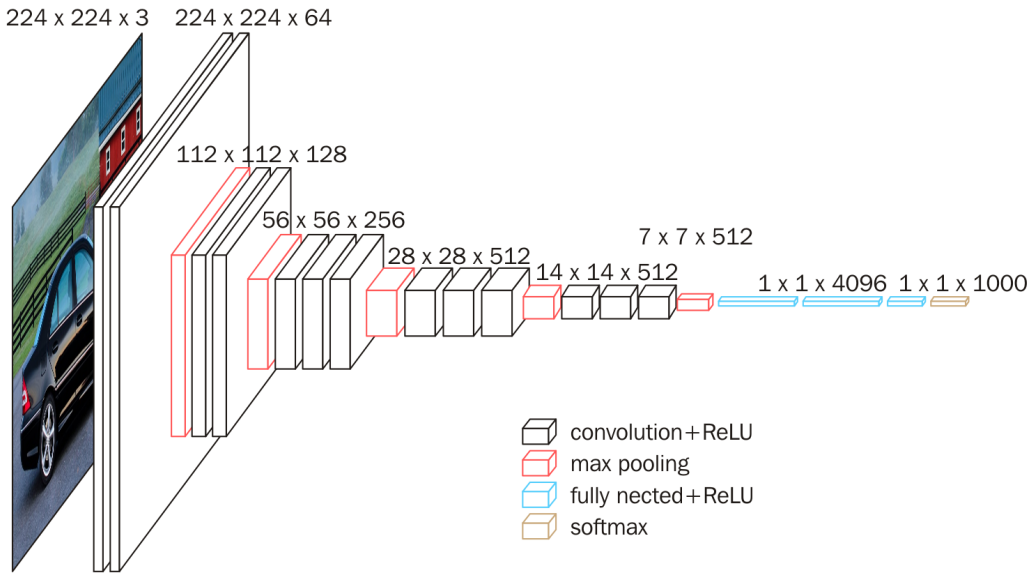


Figure 2.9: General Structure of VGG16 Model [33]

Block Structure The convolutional layer and pooling layer of `VGG16` are divided into different blocks on the right side of Figure 2.9. These blocks are labelled `Block1` to `block5` from front to back. Each block includes a few convolutional layers and a pooling layer. For example: `Block4` contains:

- 3 convolutional layers: `conv3-512`
- 1 pooling layer: `maxpool`

And in the same block, the number of channels of the convolutional layer is the same, for example:

- `block2` contains 2 convolutional layers, each of which is represented by `conv3-128`, that is, the convolution kernel is: $3 \times 3 \times 3$, the number of channels is 128

- `block3` contains 3 convolution layers, each convolution layer is represented by `conv3-256`, that is, the convolution kernel is: $3 \times 3 \times 3$, the number of channels is 256

The structure of `VGG16` divided by blocks is given below in Figure 2.10.

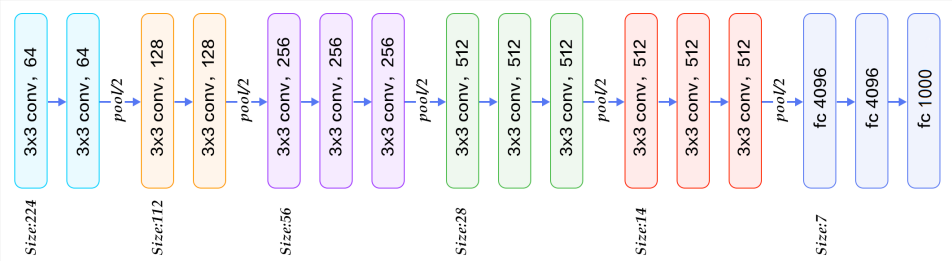


Figure 2.10: Structure of VGG16 Model by Blocks [33]

The input image of `VGG` is $224 \times 224 \times 3$:

- The number of channels doubles, from 64 to 128 in order, and then to 256, until 512 remains the same and no longer doubles
- Height and width change halved from $224 \rightarrow 112 \rightarrow 56 \rightarrow 28 \rightarrow 14 \rightarrow 7$

2-stage Model

The 2-stage model is named for its 2-stage processing of pictures, also known as the region-based method. Here we choose the R-CNN series work as a representative of 2-stage model, which is also the model we adopted for this project.

R-CNN

Traditional computer vision methods often use well-designed manual features, such as SIFT and HOG to describe images, while deep learning methods advocate the acquisition of features. From the experience of image classification tasks, the effects obtained by the CNN network automatically acquired features has exceeded the characteristics of manual design. Girshick et al. [9] applied convolutional networks in local areas to give convolutional networks ability to learn high-quality features.

R-CNN achieves object detection in two stages. The first is to propose a few regions of interest which may have potential objects, that is, the local cropping of the image, named the step of "Region Proposal" [9]. The paper [9] used selective search algorithm to run the SOTA performing classification network, i.e. AlexNet in many cases, and then obtained the types of objects in each region. A basic structure of R-CNN is illustrated below as Figure 2.11.

The proposal of R-CNN has two significant contributions: first, CNN can be used for region-based localisation and segmentation of objects; second, when the number of supervised training samples is scarce, pre-trained models on additional data are able to obtain excellent results after fine-tuning. The first contribution motivated

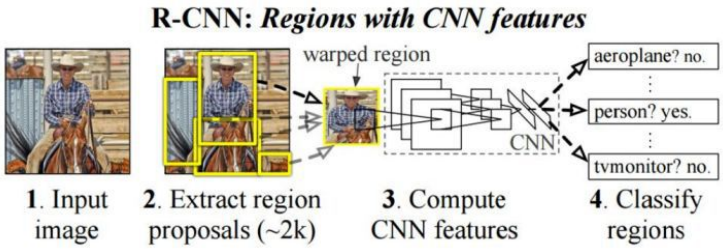


Figure 2.11: Structure of R-CNN [9]

almost all 2-stage methods, and the second contribution used the model trained in the classification task (Imagenet [6]) as the base network. The fine-tuning method for detecting problems has also been used widely in the subsequent works.

The idea of R-CNN is straightforward, the original detection task is transformed into a classification task based on the region, which is a test of the deep learning method on the detection task. There are also many issues with the model itself, for example there are three different models needs to be trained: proposal, classification, regression and also performance drawbacks triggered by redundant calculations. But generally R-CNN can be considered as the pioneer in the field.

Fast R-CNN

In 2015 Girshick [8] pointed out the reason why R-CNN is time-consuming is that CNN is performed separately on each Proposal. Without sharing calculations, it is proposed that after the basic network is run on the entire picture, it is introduced into the R-CNN sub-network, sharing the large Partially calculated, hence the name "Fast".

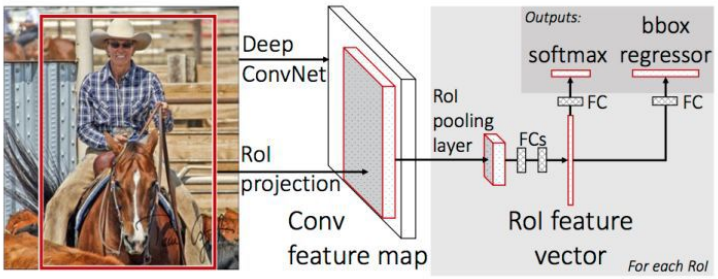


Figure 2.12: Structure of Fast R-CNN [8]

Figure 2.12 above shows the architecture of Fast R-CNN. A feature map is obtained from the feature extractor, at the same time, a Selective Search algorithm is run on the original image and the RoI i.e. Region of Interest, a coordinate group that can be mixed with region proposal is mapped to the feature map. Then RoI Pooling is performed for each RoI, this operation obtains feature vectors of equal length, then sorts the obtained feature vectors into positive and negative samples

when maintains a certain ratio of positive and negative samples. Next fast R-CNN batches them into the parallel R-CNN sub-network, performs classification and regression at the same time and unify both losses.

This structure of Fast R-CNN is exactly the prototype of the meta-structure adopted by the mainstream two-stage method for detection tasks. According to the original paper, "fast R-CNN [8] unifies proposal, feature extractor, object classification and localisation in a whole structure, and improves the efficiency of feature utilisation through shared convolution calculations", which acts as the biggest contribution.

Faster R-CNN

Faster R-CNN [28] lays the foundation of the two-stage method. Its proposed novel RPN network replaces the classic selective search algorithm to achieve the detection task end-to-end based on a neural network. Generally speaking, Faster R-CNN is a combination of RPN plus a Fast R-CNN, sharing most of the characteristics of convolution calculation with RCNN significantly reduced the calculation task brought by RPN, which means "Faster R-CNN is able to run at 5fps on a single GPU and reaches SOTA in terms of accuracy" [28].

The main contribution of Faster R-CNN [28] is the creation of Regional Proposal Networks to replace the previous SS algorithm. RPN network models the task of Proposal to the problem of binary classification: whether it is an object. Figure 2.13 below illustrate the basic structure of Faster R-CNN [28].

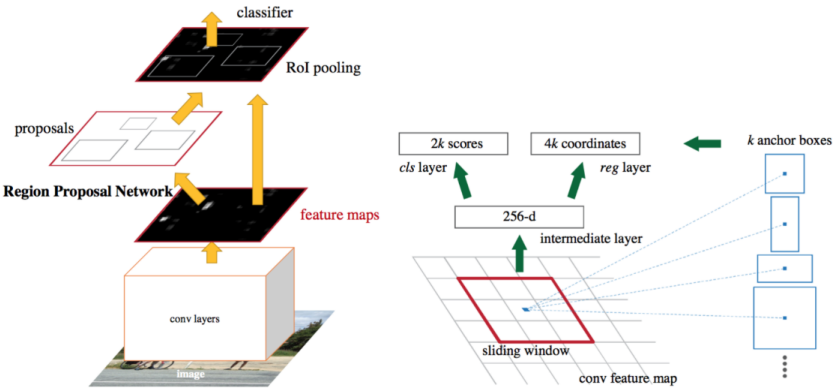


Figure 2.13: Structure of Faster R-CNN [28]

The first step is to generate anchor boxes with various size and aspect ratio on a sliding window (as shown in the right part of Figure 2.13), give IoU a predetermined threshold, label anchor box as positive or negative corresponding to the Ground Truth. Thus, the input samples to RPN is organised into anchor box (coordinates) of each anchor box and whether this anchor box has an object or not (binary label). RPN network maps each sample to four coordinate values and a probability value which corresponds the probability of there is an object in the anchor box, values for

the four coordinates define the position of the object. Finally, combine the loss of binary and coordinate regression, as the target of training RPN network.

Faster R-CNN completed the "deep" part of the inspection tasks using RPN network. The idea of using a sliding window to generate anchor box is also widely adopted such as in YOLO v2, etc.

2.3 Image-text Alignment

Image-text matching and visual question answering (VQA) are the frontiers of image and text multi modal fusion. The former needs to map images and texts to the same semantic space, and then judge their similarity by distance; the latter needs to find suitable answers in all candidate sets.

2.3.1 Convolutional Neural Network Architectures for Matching Natural Language Sentences

This paper [14] is a common method for text processing used in Image-Text Matching tasks, which is equivalent to a classic work of natural language processing.

The core idea is to apply the convolution operation to text. The problem to be solved in this article is the matching between Text.

Text Convolutional

First of all, we explain the convolution operation of the text. After the discrete text information: embedding is done, the continuous features of the text will be obtained, that is, a word can be represented by a vector. At this time, it is very similar to process an image. A word is equivalent to one pixel in an image, and a sentence is composed of several such words. In a specific implementation, the feature dimension of Embedding can be equivalent to the number of channels in the image, so that the words are connected end to end, and an image with a length of $1 \times N$ (the number of text words) is finally formed. At this time, a convolutional network of text can be implemented.

Images can be resized to the same size by linear transformation methods, but text cannot, so the convolution of the text generally uses a Gated Convolutional Layer, that is, for sentences of insufficient length, it is forced to zero, as shown below in Figure 2.14:

Another point to note is that Maxpooling will be selected as the pooling method for general text so that local features can be maximised accordingly.

Architecture

This paper proposed two network structures. One is the matching post-fusion technology, that is, to obtain their vector representation through CNN in two texts to be matched, and then perform stitching and add several layers of full connections to predict the final match.

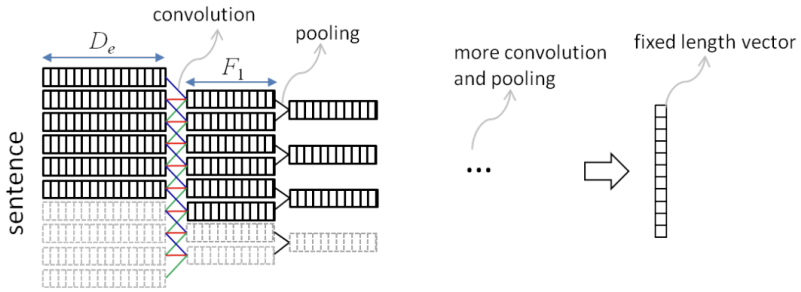


Figure 2.14: Structure of Convolutional Sentence Model [14]

The network structure of Arc-I is shown in the Figure 2.15:

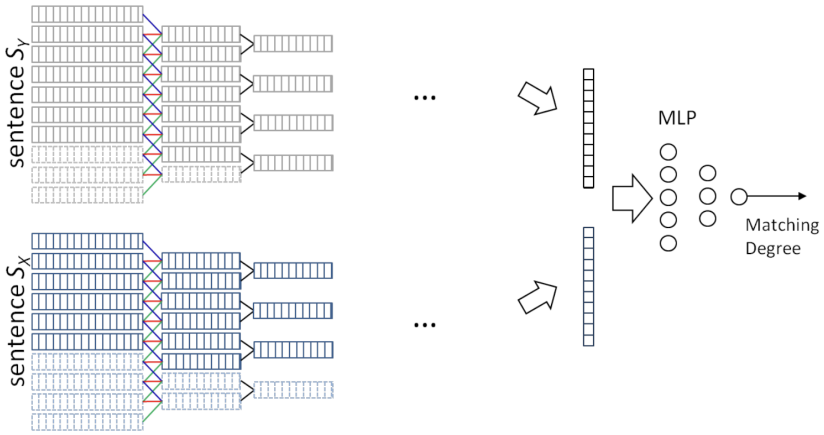


Figure 2.15: Arc-I for matching two sentences [14]

It can be seen from the figure that its feature processing method of encoding belongs to the post-fusion of features, and is very naive. In such a network model, the author did not consider the correlation between different sentences and the order of words, that is, the model mapped the two sentences into a semantic space involuntarily. In order to solve this problem, the author developed a second method:

The network structure of Arc-II is shown in the Figure 2.16:

This model combined all the word positions of the two sentences, and each position was expressed by the following formula:

$$\hat{\mathbf{z}}_{i,j}^{(0)} = \left[\mathbf{x}_{i:i+k_1-1}^{\top}, \quad \mathbf{y}_{j:j+k_1-1}^{\top}\right]^{\top}$$

x and y respectively represent the sentence to be matched. According to the above formula, the two sentences will form a square `FM1`. Each position of `FM1` is composed of $2k_1$ vectors. That is, this is actually a 4-dimensional structure `[#Sentence * #Sentence * 6 * #Embd_size]`. A one-dimensional convolution using the latter two dimensions will form a three-dimensional feature map of `layer-2`.

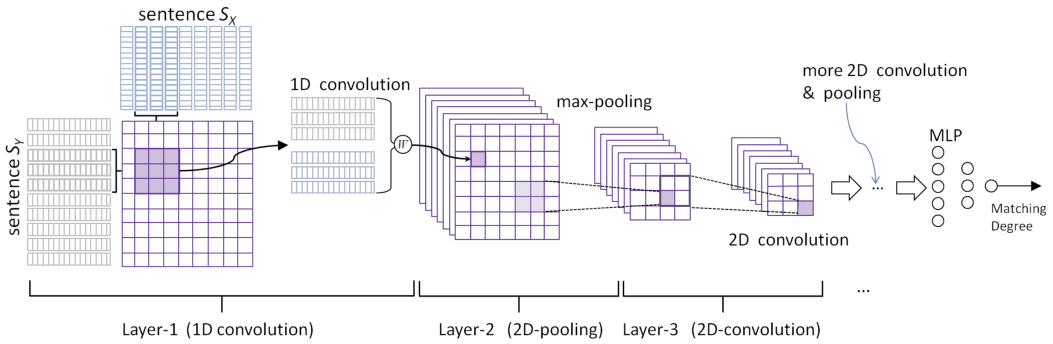


Figure 2.16: Arc-II of convolutional matching model [14]

Then the network structure behind is the common model in the image.

Summary

- The convolutional network processes text data and introduces location information into the convolutional network to form a 3d feature map. Gated Convolution, to ensure that the gradient will not be affected when the sentence is not long enough.
- MaxPooling guarantees maximum response of local features.

2.3.2 m-CNNs

This work [19] needs to solve the problem of image and text matching. There is not much innovation in image features, that is, features extracted by CNN networks. Different from his previous work, it fuses image features with text features and directly inputs them to the convolutional network for matching. m-CNNs also pays attention to text information with different granularities.

The way m-CNNs deals with features is by directly concatenating the features.

Word-level Matching CNN

The finest-grained matching problem. The network structure is as follows:

The calculation formula is as follows:

$$\vec{\nu}^{i}_{(0)} \overset{\text{def}}{=} \nu^{i}_{wd} \left\| \nu^{i+1}_{wd} \right\| \cdots \left\| \nu^{i+k_{rp}-1}_{wd} \right\| \nu_{im}$$

Phase-level Matching CNN

Word granularity matching, the network structure is as follows:

The calculation formula is as follows:

$$\vec{\nu}^{i}_{ph} \overset{\text{def}}{=} \nu^{i}_{ph} \left\| \nu^{i+1}_{ph} \right\| \cdots \left\| \nu^{i+k_{rp}-1}_{ph} \right\| \nu_{im}$$

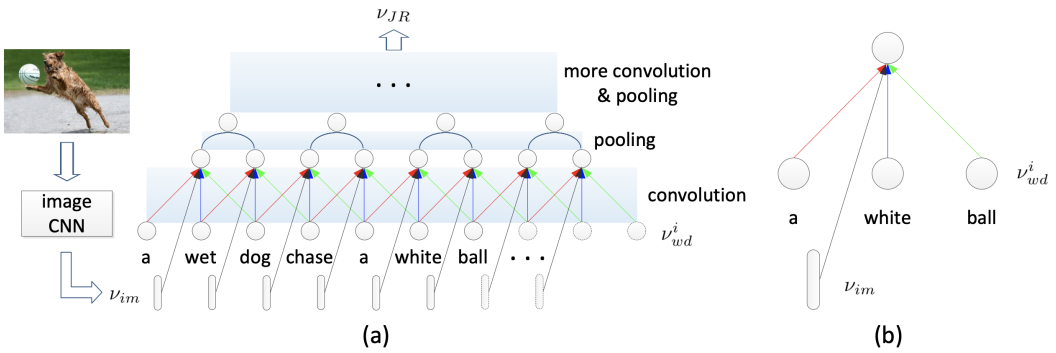


Figure 2.17: The word-level matching CNN. [19]

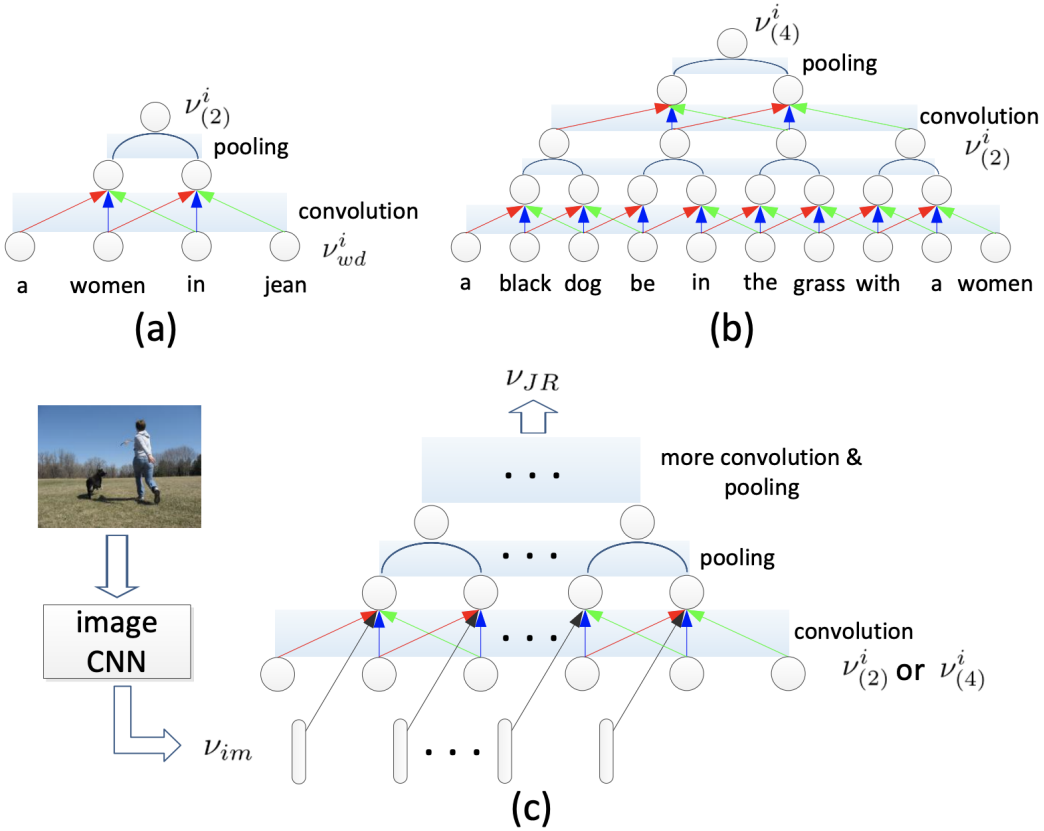


Figure 2.18: The phrase-level matching CNN and composed phrases. [19]

Phrase-level matching CNN is similar to word, except that the position is different. The author uses two different granularity phrases in the text, one is two words and one is four words.

A very useful concept in this is the receptive field, which means that a neuron is calculated from a few words. This concept is the same as the concept of the

receptive field in an image.

Sentence-level Matching CNN

For sentence level matching, the network structure is as follows:

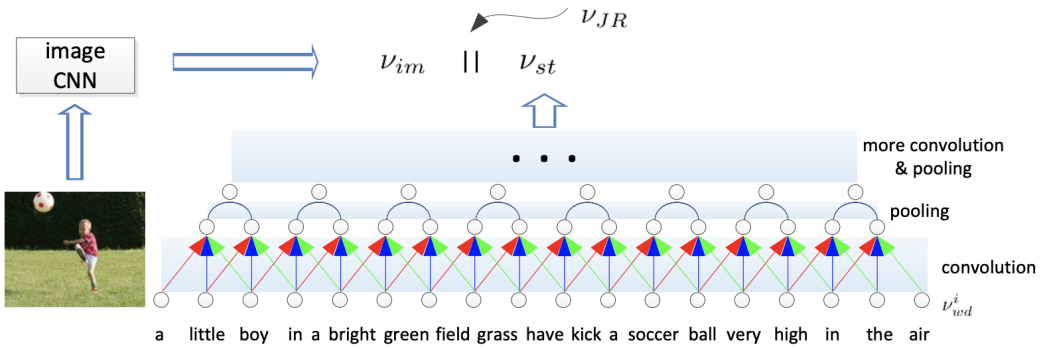


Figure 2.19: The sentence-level matching CNN and composed phrases. [19]

Sentence level matching used the last feature for stitching.

From the result, it can be seen that the final ensemble effect is the best, but the performance of the single model is about average.

2.3.3 Dual-Path Convolutional Image-Text Embedding

The problem to be solved in this work [40] is also the problem of image and text matching. Unlike the previous two works, it does not use front fusion (generally, the effect of front fusion is better than post fusion). But some interesting ideas were introduced.

Architecture

The merit of this network structure is the introduction of residual in text's encoder, which solves the problem of matching images and texts after fusion.

Loss function

Instance loss was widely discussed in the loss function. The so-called instance loss is the task of doing graphic and text matching. Each pair is used as an instance, and then a `softmax` classifier is used to learn the loss function.

The idea is to adapt the algorithm to the data, that is, to convert the matching problem into a classification problem.

This kind of transformation risk is quite large because our query is diverse; we can never exhaust all query and image pairs, so the basic idea is to achieve it on a multitasking basis.

Before explain multitask loss, we firstly focus on the ranking loss.

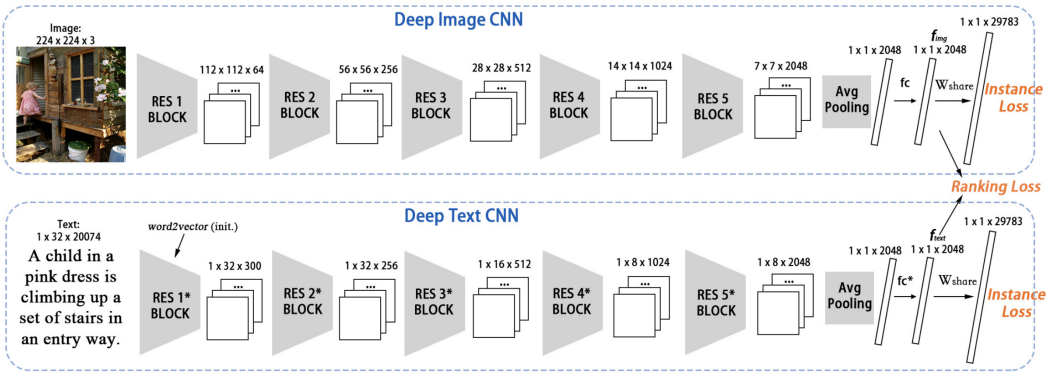


Figure 2.20: Architecture of 2 Convolutional Neural Networks [40]

The general definition of ranking loss is as follows:

$$l\left(x_n, y_{pos}, y_{neg}\right) = \max\left(0, \mu - S\left(x_n, y_{pos}\right) + S\left(x_n, y_{neg}\right)\right)$$

S is the similarity calculation function, and μ is the minimum margin in which the similarity needs to be guaranteed. Such ranking loss only considers the loss of x as the matching object, but does not consider y. Since x and y are a pair, a two-way loss is required.

Based on this problem, the author proposes the following Ranking loss loss function:

$$L_{rank} = \max\left(0, \alpha - D\left(I_a, T_a\right) + D\left(I_a, T_n\right)\right) + \max\left(0, \alpha - D\left(T_a, I_a\right) + D\left(T_a, I_n\right)\right)$$

The front part is Ranking Loss of Image, and the back part is Ranking Loss of Text.

So the final author's loss consists of a total of 3 parts, namely Ranking loss, visual Instance loss, and text Instance loss.

$$L = \lambda_1 L_{rank} + \lambda_2 L_{visual} + \lambda_3 L_{textual}$$

where $\lambda_1, \lambda_2, \lambda_3$ are predefined weights for different losses.

Some Interesting Tricks

This work contains many interesting tricks to achieve better results which we will point out as follows.

1. Embedding models initialised with `word2vec` are better than random initialisation.

2. Sentence jittering, the model randomly adds a certain amount of zero padding to the beginning and end of the sentence.

3. Instance loss finally shares a same weight parameter to ensure that the information of two different modalities is mapped to the same space.

Two stage training is also the highlight of this article:

- Stage-I. At this stage, text residual network is firstly trained. Then it fixes the ranking loss and image in CNN models, so that the CNN of text can be trained. The reason of doing this is the network of text was randomly initialised. To add ranking loss because the image and text are definitely not in the same semantic space at the beginning, we are worried that this loss will be brought into the network of text. Fixing the image CNN because we do not want the gradient of the image and the gradient of the text to interfere with each other.

- Stage-II, when text's CNN training converges, adding ranking loss and image CNN to the model, and training end-to-end together, the final result is the best.

2.3.4 DANs

The main research contribution of DAN [23] is to introduce attention mechanism into neural network model. The starting point is relatively obvious: the ultimate problem of image and text matching is the matching problem between the entire text and the entire Image, however, this problem is more challenging to solve, so a basic solution is to split the tasks. Text is composed of different words, while image is composed of different regions. If we can match the words of text to the regions of image, this problem will become simpler.

The basic idea is to use the attention mechanism to match text words with image regions in the network automatically. The author cites two types of attention mechanisms: visual attention and text attention.

Both types of attention used the previous state and determine the "position" of attention for the next state.

Visual Attention

The formula for visual attention is:

$$\mathbf{h}_{\mathbf{v},n}^{(k)} = \tanh\left(\mathbf{W}_{\mathbf{v}}^{(k)}\mathbf{v}_n\right) \odot \tanh\left(\mathbf{W}_{\mathbf{v},\mathbf{m}}^{(k)}\mathbf{m}_{\mathbf{v}}^{(k-1)}\right)$$

$$\alpha_{\mathbf{v},n}^{(k)} = \text{softmax}\left(\mathbf{W}_{\mathbf{v},\mathbf{h}}^{(k)}\mathbf{h}_{\mathbf{v},n}^{(k)}\right)$$

$$\mathbf{v}^{(k)} = \tanh\left(\mathbf{P}^{(k)}\sum_{n=1}^{N}\alpha_{\mathbf{v},n}^{(k)}\mathbf{v}_n\right)$$

All $\mathbf{W}$ in the formula are parameters that the network needs to learn, $\mathbf{h}$ is the hidden state, and $\mathbf{m}$ is the memory vector.

Textual Attention

The formula for textual attention is:

$$\mathbf{h}_{\mathbf{u},t}^{(k)} = \tanh\left(\mathbf{W}_{\mathbf{u}}^{(k)}\mathbf{u}_t\right) \odot \tanh\left(\mathbf{W}_{\mathbf{u},\mathbf{m}}^{(k)}\mathbf{m}_{\mathbf{u}}^{(k-1)}\right)$$

$$\alpha_{\mathbf{u},t}^{(k)} = \text{softmax}\left(\mathbf{W}_{\mathbf{u},\mathbf{h}}^{(k)}\mathbf{h}_{\mathbf{u},t}^{(k)}\right)$$

$$\mathbf{u}^{(k)} = \sum_t \alpha_{\mathbf{u},t}^{(k)}\mathbf{u}_t$$

The step size $\mathbf{K}$ of the two Attentions is a super parameter, and the author proves that $\mathbf{K} = 2$ is the best in experiments.

Visual and Textual Representation

The visual features use the features of the second layer of Resnet or VGG, and the text features use the features of bidirectional RNN (LSTM). The visual features are shown in the following Figure **??**:

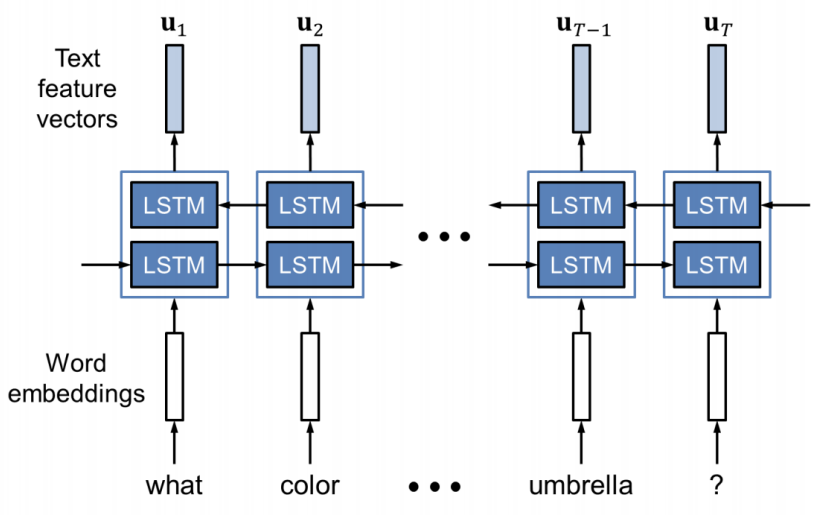


Figure 2.21: Bidirectional LSTMs for text encoding [23]

VQA and Image-Text Matching

This research solved two different problems which both used the previous attention mechanism. However, the methods of applying attention are different.

Visual Question and Answer In the VQA dataset, all the answers are one single word, so in essence, this problem is a classification problem, that is, we need to know which one of the answer sets the question is in the end.

The network structure diagram is displayed as follows in Figure 2.22:

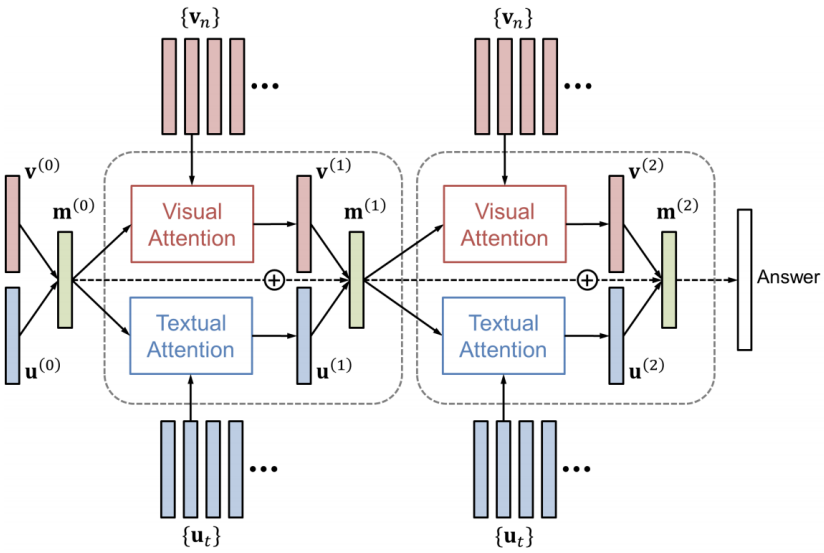


Figure 2.22: r-DAN in case of $\mathbf{K} = 2$ [23]

As can be seen from the figure, it uses the last memory vector for classification. The memory vector calculation formula is as follows:

$$\mathbf{m}^{(k)} = \mathbf{m}^{(k-1)} + \mathbf{v}^{(k)} \odot \mathbf{u}^{(k)}$$

The initialisation of different parameters is as follows:

$$\mathbf{m}^{(0)} = \mathbf{v}^{(0)} \odot \mathbf{u}^{(0)}$$

where

$$\mathbf{v}^{(0)} = \tanh\left(\mathbf{P}^{(0)} \frac{1}{N} \sum_n \mathbf{v}_n\right)$$

$$\mathbf{u}^{(0)} = \frac{1}{T} \sum_t \mathbf{u}_t$$

Because it is an image question answering task, it is necessary to fuse the text features with the image features at each Attention step, and finally output them. According to the last fused features, a `softmax` classifier is sufficient.

Image-Text Matching The biggest difference between the image and text matching problem and VQA is solving a ranking problem, so we needs to compare the distance between the two features, so we cannot share the same memory vector.

Corresponding image and text have their own memory vector, their calculation formula is as follows:

$$\mathbf{m}_{\mathbf{v}}^{(k)} = \mathbf{m}_{\mathbf{v}}^{(k-1)} + \mathbf{v}^{(k)}$$
$$\mathbf{m}_{\mathbf{u}}^{(k)} = \mathbf{m}_{\mathbf{u}}^{(k-1)} + \mathbf{u}^{(k)}$$

The network structure is shown as follows in Figure **??**:

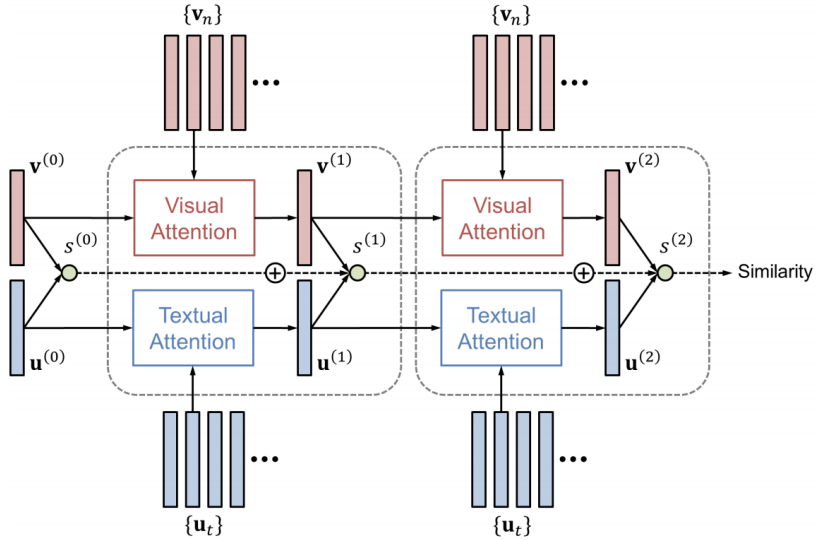


Figure 2.23: m-DAN in case of $\mathbf{K} = 2$ [23]

In the final experiment, the author used the same Ranking Loss as in the previous work. One difference is that each step of Attention will generate a matching vector. What is done here is to add all S.

2.3.5 Summary of Image-Text Alignment Related Works

Throughout these papers, all models are dealing with graphic and text fusion, that is, multi-modal fusion. One of his most basic starting points is that the encoder models for text and images must be good enough, which is the first one, convolution in the article (it seems that Recurrent neural network can also be used).

With an excellent encoder, the next thing to do is the fusion of features. Currently, there are two ways to fuse features, one is pre-fusion, and the other is post-fusion. Pre-fusion inputs image information and text information to a network for further encoder, and finally uses the task-related network; post-fusion is to directly concate the features from the image text encoder and then input to the task-related network. Generally speaking, pre-converged networks are better than post-converged.

Graphic matching is a matching problem of all sentences and all images. It may be relatively tricky to solve this problem directly, so sometimes it is necessary to split these data into components.

The other is that there are many tricks in the model training. Sometimes or most of the time, it is not because our idea is not good enough, but because we have insufficient experience in training the model, so we tell us that we must be patient in training the model. The road is right, you can keep going, and it will have good results.

2.4 Conclusion of Literature

There are many research focusing on the field of object detection and image-text alignment. We reviewed the research problem and studies their proposed solutions and methodologies.

In object detection, the major problem is to extract featured object out from images. Convolutional neural network (CNN) [24] was firstly proposed and became the basis of most of the preceding improved models in the field. There are classic deep learning models like VGG [33] and RCNN [9], followed by more recently improved ones like Fast RCNN [8] and Faster RCNN [28]. These deep learning models significantly helped solve the task of object detection and they can achieve promising results.

For image-text alignment, the primary focus is how to map both image and text to the same semantic space. There are CNN models which was able to apply the convolution operation to text such as m-CNN. Recently, models like DAN focus on applying attention mechanism on neural networks to align different image regions to different text tokens, which greatly improved the quality of image-text alignment.

Chapter 3

Coarse-grained Cross Modal Retrieval

In Chapter 2, we discussed several models proposed to solve the image-text alignment task. However, they all have drawbacks in terms of different aspects. In this chapter we explain the **S**tacked **C**ross **A**ttentio**n** Model (SCAN) [15] and applied it on performing coarse-grained cross-modal retrieval task. Followed by discussing how it excel the image-text alignment task and why we choose to investigate it to solve the problem later in the task of coarse-grained cross modal retrieval for artworks.

The structure of this chapter is as follows. Section 3.1 gives an introduction and motivation of SCAN. Section 3.2 explains the structure and methodologies used in SCAN, also how all its components iterates with each other. Section 3.3 briefly discusses the strengths of the adopted model for coarse-grained cross-modal retrieval. Section 3.4 illustrates the preliminary experimental results on our artwork datasets and the achievement of SCAN. Section 3.5 summarises this chapter.

3.1 Introduction

There are several models proposed recently to solve the task of cross modal retrieval, and many have achieved excellent accomplishments. Lee, et al. [15] uses the proposed **S**tacked **C**ross **A**ttentio**n** (SCAN) to find all potential alignments between the image area and the words, thereby calculating the similarity between the graphics and the text. Existing methods perform fixed-step attention inference so that only a limited semantic alignment can be found at a time, and SCAN can find all possible semantic alignments at the same time. Since the number of semantic alignments varies among different images and sentences, the corresponding relationship inferred by the Stacked Cross Attention method is more comprehensive, thereby making the image-text alignment more interpretable.

Also, this method used some of the currently available optimisation methods, such as the use of hard-negative, triplet ranking loss, etc. SCAN is proved to be very effective in image-text alignment task, thus, in this chapter, we employ SCAN on our coarse-grained cross modal retrieval task.

We explain the structure of SCAN model in the next sections.

3.2 Image Feature Extraction

Before we perform any cross modal retrieval and alignment between image and text, we need to firstly extract features from images. There are many models we have discussed before in Section 2, here we use one of the most widely used model called faster R-CNN to both extract the top-k image regions from an image and represent the image regions of interest as a continuous vector as in the work done by Anderson, et al. [2]. It proposes a top-down and bottom-up attention model method, which is applied to the related issues of visual scene understanding and prominent question answering system.

The article mentions the use of faster R-CNN to employ a bottom-up attention model - that we adopted. Different from the traditional faster R-CNN, [2] uses "*not only the object detector but also the attribute classifier for each region of interest, so that a binary description of the object (attribute, object) can be obtained*". Bottom-up attention model allows the overlap of interest frames through the preset threshold, which can understand the image content more effectively.

3.2.1 Why Adopt Faster R-CNN?

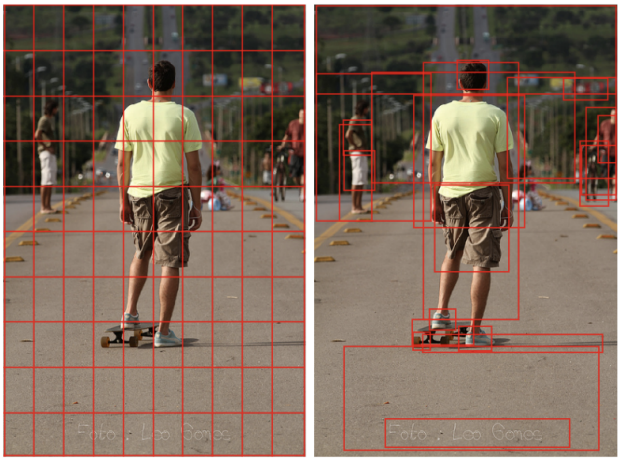


Figure 3.1: Faster R-CNN with attention comparing to CNN [2]

It can be seen from the Figure 3.1 that using CNN demands more characteristics than R-CNN, and many of them are usually worthless. The object detection method of R-CNN first "*captures the interest region for an image, then applies an object detector to each interest region, so that the image category can be accurately obtained; the CNN method expects the input of an entire image and is used for broad sample classification, which are often complicated and computationally intensive*" [2]. Besides, faster R-CNN improves on previous generations of R-CNN methods and earns

the ability to recognise almost all objects with only one input, which significantly improves the processing efficiency.

3.2.2 Bottom-Up Attention Model

From Figure 3.2 below, it can be seen that the difference from the prior works is that the set of thresholds allows overlapping of interest frames, which can more effectively understand the image content. In this model, according to [2], "*not only the object detector but also the attribute classifier is used for each region of interest, so that a binary description of the object (attribute, object) can be obtained*". This description is proved more suitable for real-world applications.

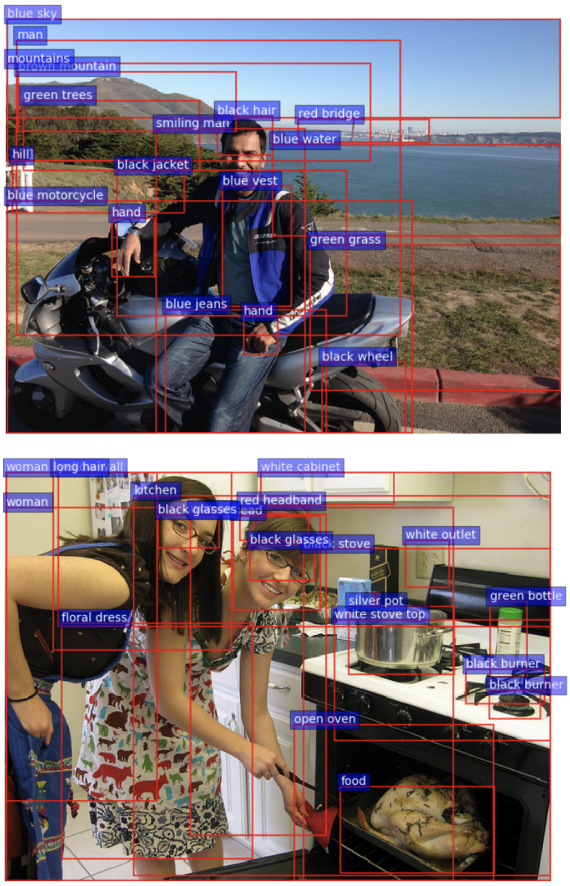


Figure 3.2: What bottom-up attention model captures? [2]

3.3 SCAN Model

In this section, we discuss SCAN model with a brief introduction on its structure, followed by in-depth explanations on each specific stages - what is happening behind

and why?

3.3.1 Brief Structure

1. Use bottom-up attention mechanism [2] to detect the image area and extract the features of the image area;
2. Map the words in the sentence and their sentence context to feature vectors;
3. Stacked Cross Attention is used to deduce the similarity of images and text by aligning image regions and word features;
4. The loss function of SCAN focuses on the hardest negative image-text pairs in each batch (that is, the most unmatched image-text pairs).

Next, we are going to explain each stage in detail.

3.3.2 Image-Text Matching

The process is shown below in Figure 3.3.

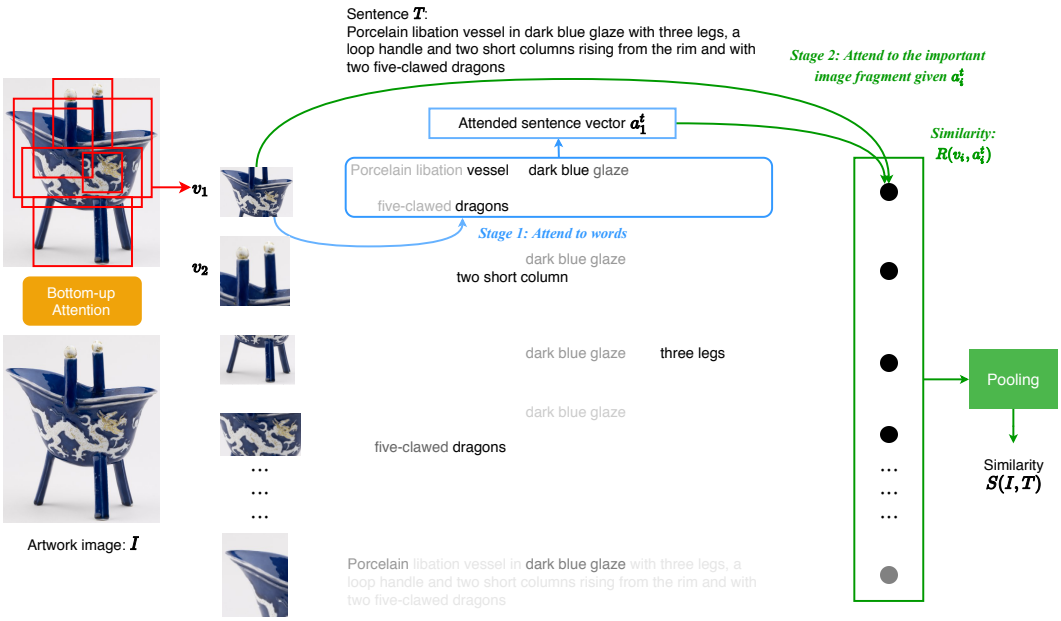


Figure 3.3: Image-Text Stacked Cross Attention

First, use bottom-up attention [2] to extract multiple proposals into features for the image, then map to the same dimensions as the sentence features, and use bi-direction GRU to extract features for the sentence.

Stage 1: calculate the attention representation a_i^t of all words for each region i, and add them together to obtain the sentence representation a_i^t, the formula is as follows:

$$a_i^t = \sum_{j=1}^{n} \alpha_{ij} e_j$$

where

$$\alpha_{ij} = \frac{\exp\left(\lambda_1 \bar{s}_{ij}\right)}{\sum_{j=1}^{n} \exp\left(\lambda_1 \bar{s}_{ij}\right)}$$

Stage 2: calculate the cosine similarity of the i-th region and the obtained a_i^t.

$$R\left(v_i, a_i^t\right) = \frac{v_i^T a_i^t}{\|v_i\| \, \|a_i^t\|}$$

Finally, i areas are superimposed together to get the similarity between image and text, using LogSumExp pooling (LSE), i.e.

$$S_{LSE}(I,T) = \log\left(\sum_{i=1}^{k} \exp\left(\lambda_2 R\left(v_i, a_i^t\right)\right)\right)^{(1/\lambda_2)}$$

Alternatively, we can summarise $R\left(v_i, a_i^t\right)$ with average pooling (AVG), i.e.

$$S_{AVG}(I,T) = \frac{\sum_{i=1}^{k} R\left(v_i, a_i^t\right)}{k}$$

3.3.3 Text-Image Matching

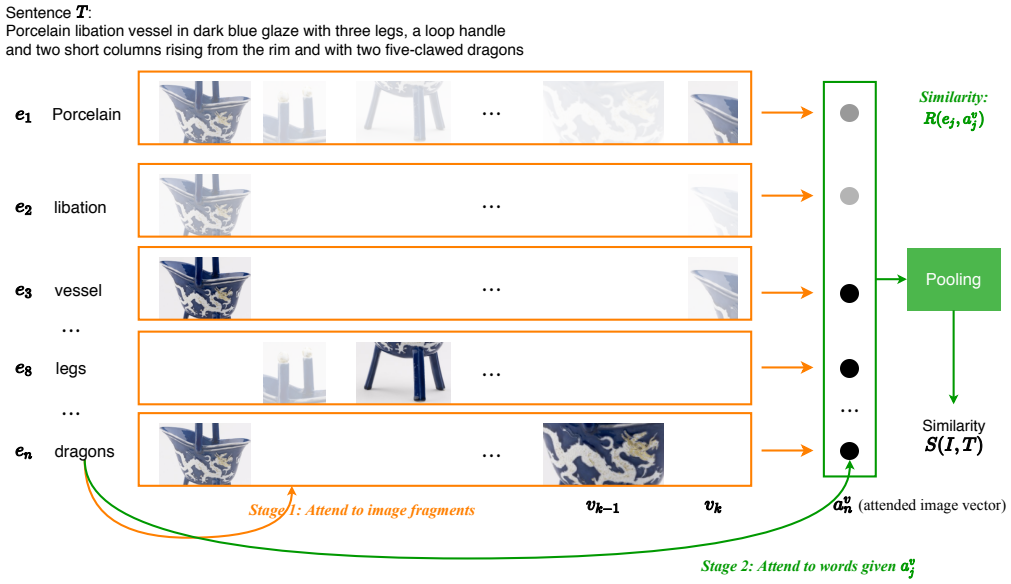


Figure 3.4: Text-Image Stacked Cross Attention

The overall steps correspond exactly to the above, except that each word is used to calculate the similarity with the attention of a picture, which is not repeated here. The process is illustrated in Figure 3.4.

3.3.4 Image and Text Feature Representation

Image

Bottom-up attention technique [2], which is a method of target detection, is obtained based on faster-RCNN. Faster R-CNN first obtains the regions of interest from the input image, then applies an object detector for each region of interest, so that these image features can be captured accurately.

Here the flow for image feature representation is: faster-RCNN, Residual NN (Resnet)101 ⇒ 2,048 dimensional features ⇒ fully-connected layer transform to h-dimensional ⇒ get image feature set.

Text

A RNN (recurrent neural networks) is used. Here the flow for text feature representation is: word ⇒ one-hot vector showing an index of the word in vocab ⇒ embed to 300-dimensional vector ⇒ bidirectional GRU map to h dimensions word feature.

3.3.5 Target Alignment

Target alignment is essentially the setting of the loss function. In this case, SCAN employs a hinge-based triplet ranking loss with margin α [15].

$$l(I,T) = \sum_{\hat{T}} [\alpha - S(I,T) + S(I,\hat{T})]_+ + \sum_{\hat{I}} [\alpha - S(I,T) + S(\hat{I},T)]_+$$

- where $[x]_+ \equiv \max(x, 0)$ and S represents the similarity score calculation function (i.e. S_{LSE})
- The first sum calculates all negative sentences retrieval result $\hat{T}$ given an image I input and the second sum takes over all negative images retrieval result $\hat{I}$ given a sentence T input.
- If an image I and a sentence T are closer to one another in the joint common space than any negative pairs, the hinge loss is zero by the margin α,.

Here SCAN [15] only consider the hard negatives in a mini-batch of stochastic gradient descent instead of summing over all the negative samples to make the calculations more computational efficient.

- for a pair (I, T), the formula is shown below. e.g. the hard negative of one image is the image that has the highest similarity with the text besides this original pair. (vice versa for text)

- Now the hardest negatives are given by

$$\hat{I}_h = \text{argmax}_{m \neq I} S(m, T)$$

and

$$\hat{T}_h = \text{argmax}_{d \neq T} S(I, d)$$

- The final loss is:

$$l_{hard}(I, T) = \left[\alpha - S(I, T) + S\left(I, \hat{T}_h\right)\right]_+ + \left[\alpha - S(I, T) + S\left(\hat{I}_h, T\right)\right]_+$$

3.4 Preliminary Results

Here we perform SCAN on two artwork datasets introduced in Chapter 1: one ancient Egyptian artworks and one Chinese artworks. For ancient Egyptian artworks dataset, it has 14,353 images in the training set, 1,793 images in the testing/validation set. The other Chinese artworks dataset, there are 6,086 images in the training set, 761 images in the testing and validation set each.

For experiment settings, we tested on a Ubuntu machine with Intel Xeon Processor E5-1620 (10M Cache, 3.60 GHz) CPU and a GeForce GTX TITAN X GPU. The specific parameters settings are listed below:

Settings for image representation

- We used faster R-CNN model and ResNet-101 model pre-trained by *Anderson et al.* on `Visual Genomes` dataset, performs detection of interesting regions as bottom-up attention to extract features from images.
- To perform retrieval on a coarse-grained level, we use whole images as input. Instead of capturing multiple ($k > 1$) Region of Interests (ROIs) for each image, here we set $k = 1$ which captures all full images after average pooling and extracted 2,048-dimensional features vector.
- We used L2 normalisation (Euclidean distance) into 1,024 joint common spaces (same for GRU), these will be used as image feature vectors.

Settings for text representation

- We obtained 300 dimensional word embedding as input to GRU then use embedding matrix to map it into 1,024 joint embedding spaces.

3.4.1 Evaluation Metrics

Recall is one of the most commonly used metrics in the field of information retrieval. Here in this research project, we evaluate the performance of image annotation (image query) and image retrieval (sentence query) by the recall at K (R@K). The same as many other works in the field, "R@K is defined as the fraction of queries for which the correct item is retrieved in the closest K points to the query" [15].

3.4.2 Results

The following Table 3.1 illustrates the results of running SCAN model on our ancient Egyptian and Chinese art alignment datasets.

	Image Annotation			Image Retrieval		
Method	R@1	R@5	R@10	R@1	R@5	R@10
	Ancient Chinese Art Alignment Dataset					
SCAN i-t AVG	15.3	38.5	49.9	14.1	37.6	50.2
	Ancient Egyptian Art Alignment Dataset					
SCAN i-t AVG	3.3	20.4	36.1	8.0	22.9	33.8

Table 3.1: Result of SCAN on Artwork Datasets

The results are beyond satisfactory as shown; however, as in this stage, we only used the features obtained using bottom-up attention [2] to train and test, the result may be misleading. As bottom-up attention model was trained on natural images, which means the features we obtained for training and testing set may be irrelevant to those contained in artworks. Noted using an entire artwork image and a sentence caption to feed SCAN may also lose detailed information in artworks, which will be further discussed in Chapter 4.

3.4.3 Examples

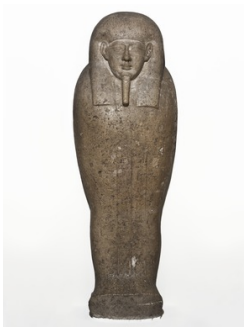
Below we display a few examples from our coarse-grained cross modal retrieval model: one for each dataset under sentence retrieval and image retrieval.

Sentence Retrieval

Figure 3.5 illustrates two examples which obtained sentences from full image queries. There are several characteristics were successfully obtained from the left Egyptian anthropoid statue and the right Chinese vase including the shape and colour but not in a more detailed level. That is, for example, our textual retrieval results captured "*human-headed*", "*white glassy porcelain vase*" and even a little bit detail: "*colourful patterns on the body*", however, more details need to be furnished such as "*long thin beard*" and "*flowering begonia, iris, and butterfly*". These cannot be accomplished well under the current settings we used which is coarse-grained based on image and sentence level.

Image Retrieval

Similar results appeared in image retrieval examples shown as Figure 3.6. The model was able to pick out major shapes and overall structure but detailed information cannot be well aligned between full images and sentences thus cause inaccurate retrieval on a more fine-grained level.



Top-5 Ranked Sentences:
human-headed ruler after mummification process, Egyptian kingdom anthropoid statue from the New Kingdom, statue in a shape of human, dark brown coloured, god imsety with human-headed, development of a new kind of mummy in human figurines

Ground Truth Sentences:
lord of the city of hardai appeared from the New Kingdom, became common in the Late Period, lord anthropoid wearing a head gear and a long thin beard made with brown stone, step toward permanent protection in the afterlife

Top-5 Ranked Sentences:
porcelain bottle with pear-shaped body with coloured decorations, fine white glassy porcelain delicately painted in mixed enamels in 'Gu Yue' style, white vase with colourful patterns on the body, white vase decorated with green dragons flying above clouds, white porcelain bottle with a green and red body

Ground Truth Sentences:
milky white bottle-shaped glass vase with depressed globular body decorated in mixed enamels with rock, flowering begonia and iris, and a butterfly in flight also poetical inscription on neck in black and two seals in red enamels.

Figure 3.5: Sentence Retrieval Example for Given Image Queries (coarse-grained)

Query:
inscription on this large light brown pharmaceutical jar, it was made for the physician's tomb with horizontal wavy pattern, but this is less likely, since Harkhebi's name lacks the epithet "justified," which was usually appended to the name of the deceased

Query:
Flower pot made of blue underglaze porcelain with rice grain ornament

Noted: shown top-5 ranked images, ranking from left to right

Figure 3.6: Image Retrieval Example for Given Sentence Queries (coarse-grained)

There is a significant issue in under the current model: too many irrelevant textual attributes in our training data - especially our Egyptian dataset. These adjuncts, preposition and conjunctions are often noises comparing to the noun phrases, which generally contain essential information of artwork descriptions. For instance, in the process of text-image stacked cross attention, we first attend words to image

fragments and then attend to words given attended image vector. These produced attended image vectors may lose some degree of accuracy when these noisy words took part in the attention steps. Therefore, some subtle, detailed features may not be correctly captured by SCAN while mixing with noises. To solve this drawback, we make some changes to SCAN in the next Section.

3.5 Conclusion

Automated image-text mutual annotation would help to transform traditional library artwork collection to digital. In this chapter, we employed a well-known cross modal retrieval model: Stacked Attention and evaluated our Egyptian and Chinese artwork datasets on it. Considering the unique representation of artworks and the rich information they usually contain, in the next chapter, we modify this current model to perform the cross modal retrieval task at a fine-grained level.

Chapter 4

Fine-grained Cross Modal Retrieval

In Chapter 3, we explained in detail on the structure and principle behind our coarse-grained cross modal retrieval model. As we discussed, there are aspects that we can focus on to achieve cross modal retrieval on another level: fine-grained. In this chapter, we present our model for the novel task of fine-grained cross modal retrieval.

The structure of this chapter is as follows. Section 4.1 gives a brief introduction on our plans for modifying the previous model into a fragment level. Section 4.2 explains the processed artwork datasets and why the pre-processing is necessary for our model. Section 4.3 proposes our fine-grained cross modal retrieval model on a fragment level instead of focusing on whole images or sentences. Section 4.4 shows the results of our presented model running on our annotated artworks datasets. Section 4.5 points out some existing shortcomings of our methodology and potential fields to be focused on in the future. Section 4.6 concludes this section.

4.1 Fragment Retrieval

In this chapter, we are trying to solve the cross modal retrieval task at a fine-grained level, which means we shall not limit our retrieval on the image or sentence level but a fragment level. We applied some changes in the previous model in Chapter 3:

- Instead of attending between image fragments with each token in sentence captions, here we extract noun phrases from sentence captions to perform mutual attention with image fragments in order to find correspondence between the two modalities on fragment level.
- During the testing process, we replace the original annotations of artworks but adopt manually handcrafted annotations as ground truth, which shall give us a more accurate experiment result.

Next, we present our datasets used in this chapter.

4.2 Dataset Preparation

As we mentioned in Chapter 1, the same datasets used in this Chapter and Chapter 3 are from the image caption paper by Sheng et al. [31]. For each artwork, two files were available: a high-resolution `jpeg` image file and an accompanying `xml` and `json` file including the metadata.

Python scripts (see Appendix A) were written to parse the `xml` and `json` files. Instead of using the raw captions, here the textual attributes we use are already processed into phrases. The technique here used is from Handler et al. [10], it extracts noun phrases from captions in raw `xml` and `json` files which potentially helped us improve the accuracy as in most cases noun phrases contain more relevant information of artworks.

As one single artwork can often have more than one annotation existed in its corresponding `xml` file, it is essential to extract all related annotations out and also combine them into one record for training and testing purpose, which saves computational power and simplifies the model input. We generate a combined `.txt` file for each artwork image which contains all related textual attribution then pass it to the noun phrase extraction process in order to achieve the retrieval in a fragment level; results are saved in `json` format. Details are covered in Appendix A.

Our training tasks are based on these noun phrases and extracted image fragments. The following section demonstrates the architecture of our fine-grained cross modal retrieval model.

4.3 Overall Architecture

The architecture of the fine-grained cross modal retrieval model is depicted in Figure 4.1. This model takes full artwork images and full sentence captions as input. The full artwork image input is a (224, 224) sized high-resolution colour picture and the full sentence caption comes from our dataset which contains several descriptions of artworks.

The model has two pipelines processing image and text input from start, for image input, we first detect salient regions as bottom-up attention [2] to extract features from images using faster R-CNN [28] and ResNet-101 [12] (mentioned in Section 2.2.3). These obtained image fragments and their representations will be passed into a fully-connected layer, in order to transform them into a joint embedding space with the same dimension of text (i.e. the dimension of GRU explanation in next paragraph).

For full sentence captions as text input, as we mentioned in Section 4.2, we do not process the whole sentence directly but first, extract noun phrases out to achieve a finer-grained level. These noun phrases features will be passed into a bidirectional GRU to map them into the same dimension joint embedding space as image features.

After we learnt image and text representation in a common space, we can use SCAN to attend image to text and attend text to image, in order to get better alignment in between. After computed mutual attention between image and text,

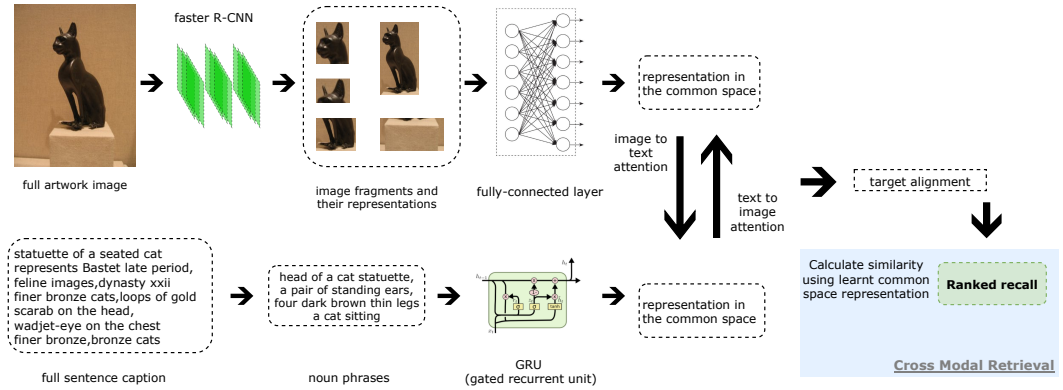


Figure 4.1: Fine-grained Cross Modal Retrieval Model Architecture

we start our follow-up part, which is cross modal retrieval. We use our learnt common space representation to calculate similarity scores between image and text - in this case, cosine similarity; meanwhile, calculate ranked precision and recall for testing.

4.4 Experiments

For experiment settings, we used the same settings and environment as mentioned in Chapter 3 : Ubuntu machine with Intel Xeon Processor E5-1620 (10M Cache, 3.60 GHz) CPU and a GeForce GTX TITAN X GPU.

Settings for image representation

- To save intense labour on retraining model for feature extraction task, we kept the weights pre-trained by Anderson et al. on `Visual Genomes` dataset like in Chapter 3 for faster R-CNN model and ResNet-101 model and performed detection of interesting regions as bottom-up attention to extract features from images.

- We captured $k = 36$ Region of Interests (ROIs) for each image after average pooling and extracted 2,048-dimensional features vector.

- We used L2 normalisation (Euclidean distance) into 1,024 joint embedding spaces (same for GRU), these will be used as image feature vectors.

Settings for text representation

- We obtained 300-dimensional word embedding as input to GRU then use embedding matrix to map it into 1,024 joint embedding spaces.

4.4.1 Ground Truth

For this specific testing task, as we mentioned before, we used our manually annotated ground truth datasets. Each artwork has an updated `xml` file which contains handcrafted image features and textural attributes. Here we show an example of ground truth annotation in Figure 4.2.

```
<object>
<name>head of a cat statuette carved in brown,
with a pair of standing ears</name>
<pose>Unspecified</pose>
<truncated>0</truncated>
<difficult>0</difficult>
<bndbox>
<xmin>84</xmin>
<ymin>40</ymin>
<xmax>172</xmax>
<ymax>127</ymax>
</bndbox>
</object>
<object>
<name>Four dark brown coloured thin shaped legs of
a cat sitting in a relaxed posture</name>
<pose>Unspecified</pose>
<truncated>0</truncated>
<difficult>0</difficult>
<bndbox>
<xmin>99</xmin>
<ymin>196</ymin>
<xmax>201</xmax>
<ymax>300</ymax>
</bndbox>
</object>
```

Figure 4.2: Sample Artwork with Ground Truth Features

From Figure 4.2 we are looking at an Egyptian artwork with a cat sculpture. For all object that exist in the artwork, those are, "*head of a cat statuette carved in brown, with a pair of standing ears*" and "*four dark brown coloured thin shaped legs of a cat sitting in a relaxed posture*", each of them has a corresponding detailed location, pose, and truncated information attached.

4.4.2 Results

Table 4.1 present the quantitative results on Chinese and Egyptian artwork datasets where all formulations of our proposed method outperform recent approaches in all measures. Here we denote t-i as Text-Image formulation by text and i-t as Image-Text formulation, AVG as average pooling and LSE as LogSumExp pooling. Like most of the SCAN based models, we tested four different methods by different combinations of formulations and pooling methods. In addition, to check the necessity of bidirectional GRU, we also include a test under one-directional GRU with i-t AVG.

i-t usually surpasses t-i on both pooling methods, and it is evident that using bidirectional GRU improves image annotation R@1 by 2.9 in Chinese artworks and 1.1 in Egyptian artworks, 1.6 and 1.2 for image search. The best result of the model are 12.8 on image annotation (R@1) and 9.3 on image retrieval (R@5) relatively on Chinese artwork dataset; 6.4 on image annotation (R@1) and 4.7 on image retrieval

	Image Annotation			Image Retrieval		
	Chinese Artwork Alignment Dataset					
Method	R@1	R@5	R@10	R@1	R@5	R@10
t-i LSE	8.7	14.1	20.3	5.7	9.2	15.6
t-i AVG	7.9	14.4	20.6	6.2	9.6	16.2
i-t LSE	**12.8**	**20.7**	28.7	**9.3**	15.7	22.4
i-t AVG	12.6	20.3	**28.9**	9.1	**15.9**	**23.2**
One-directional GRU + i-t AVG	15.5	18.5	24.8	7.5	14.7	19.8
	Egyptian Artwork Alignment Dataset					
Method	R@1	R@5	R@10	R@1	R@5	R@10
t-i LSE	4.7	9.2	17.5	1.9	8.6	14.9
t-i AVG	5.1	9.4	17.9	2.2	8.9	15.2
i-t LSE	5.8	11.5	20.9	**4.7**	11.8	18.6
i-t AVG	**6.4**	**11.6**	**21.2**	4.4	**12.1**	**18.7**
One-directional GRU + i-t AVG	5.3	10.8	20.1	3.5	10.1	16.8

Table 4.1: Result of Fragmented SCAN on Chinese/Egyptian Artwork Dataset

(R@1) relatively on Egyptian artwork dataset. In terms of R@10, our model was able to obtain a 28.9 recall for image annotation task on Chinese dataset which is gratifying and promising considering the strict and detailed ground truth testing set we used.

We noticed that the recalls for sentence and image retrieval on both Chinese and Egyptian artwork datasets are not satisfactorily high. To better understand and evaluate our model, we demonstrate several successful and unsuccessful cross modal retrieval examples in the next section for comparative analysis.

4.4.3 Cross Modal Retrieval Examples

Here we illustrate four typically successful and unsuccessful cross modal retrieval samples of our model: one for each dataset under sentence retrieval and image retrieval.

Successful Examples

Figure 4.3 shows two good text retrieval examples from given image fragment queries. The bottom two Egyptian image fragments achieved excellent retrieval results, the significant features such as colour and shape: "*grey human face*", "*light brown cat head*" were accurately captured. Although our Egyptian dataset has much more noisy textual information, after obtained noun phrases as input, we noticed that the result was much more detailed; detailed features like "*big eyes*", "*broken nose*" and "*standing ears*" were successfully identified.

Comparing to those personification figures in Egyptian artworks, our Chinese artwork dataset has more abstract presentations, and some tiny details can be ex-

Top-5 Ranked Phrases:
black and white drawings of mountains and trees, a set of small houses beside trees, houses drawn in black outline, a set of black inked inscriptions of trees and mountains, multi coloured drawings of red houses and black trees

Ground Truth Phrases:
drawings of mountains, trees and houses painted on an album leaf in monochromatic colours

Top-5 Ranked Phrases:
blue motif of flowers and leaves, central flower and rock motif, blue, underglaze blue decorated bottom, motif with leaves in blue underglaze, round blue and white motif

Ground Truth Phrases:
a central symmetrical flower motif with an intricate structure in blue and white

Top-5 Ranked Phrases:
grey human face with big eyes and lips, Egyptian male face with broken nose, male face with broken nose on a dark grey stone, grey coloured human face

Ground Truth Phrases:
dark grey coloured face with a broken nose tip, two large eyes, and a wrinkled forehead

Top-5 Ranked Phrases:
light brown cat head with standing ears, head of a cat with eyes and ears, dark yellow coloured rough textured cat head, cat head made with light brown stone

Ground Truth Phrases:
discoloured head of a cat coffin with pigmented brown finish

Figure 4.3: Successful Example of Text Phrase Retrieval for Given Image Fragment Queries (fine-grained)

tremely subtle and hard to distinguish from the background. For instance, the top fragment from a traditional Chinese drawing has plentiful features such as those red houses surrounded by trees. Our model was able to pick up almost all of them in a detailed descriptive manner: "*small houses beside trees*", "*black inked inscriptions*". However, when it comes to sporadically appeared features such as the pattern in the second Chinese artwork fragment, subtle details such as "*symmetrical flower*" may be lost although the model still extracted major features like "*central flower*" and colours.

Next, Figure 4.4 illustrates example image fragment retrieval results from given noun phrase queries. The results from Chinese artwork dataset were impressive as the model successfully distinguished very subtle features from image fragments such as "*six-character reign mark in double ring*" and "*among clouds*". The bottom two

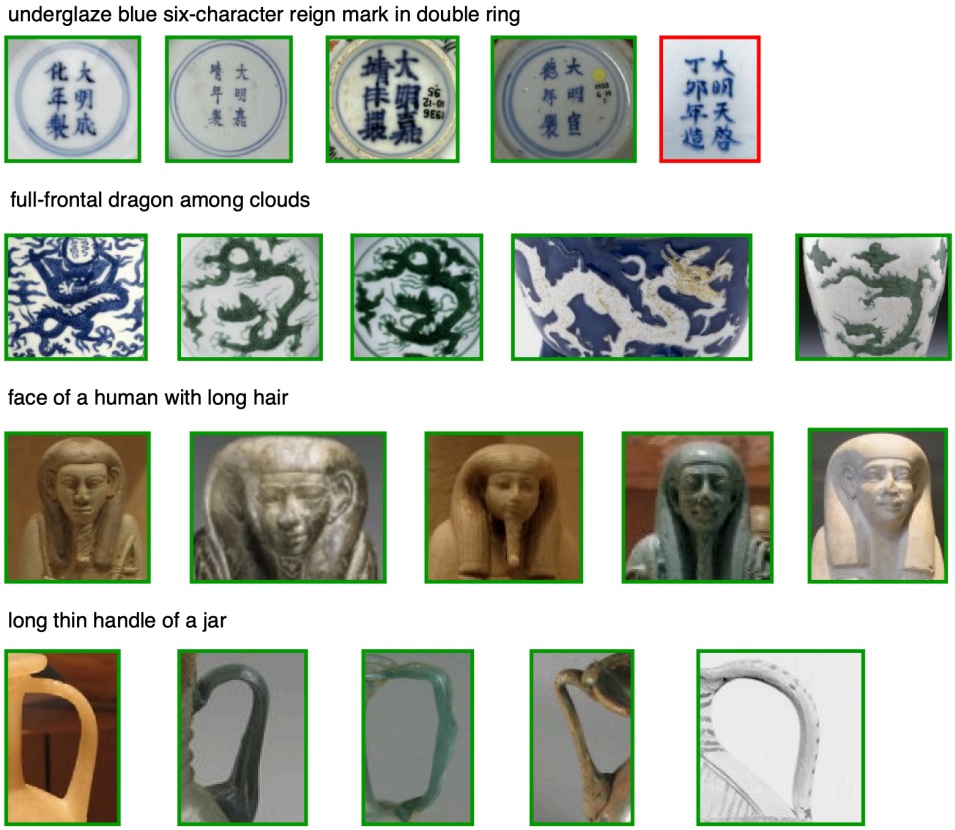


Figure 4.4: Successful Example of Image Fragment Retrieval for Given Text Phrase Queries (fine-grained)

Egyptian noun phrases are fairly simpler than the Chinese ones, significant features such as "*long hair*" and "*long think handle*" were accurately captured.

Unsuccessful Examples

Despite those brilliant fine-grained cross modal retrieval cases we have shown above, here Figure 4.5 displays two unsuccessful examples noun phrase retrieval results from image fragment queries. Our model was not able to identify the top Egyptian pig amulet head. It has a tricky shape; its rare and unique light-blue colour also made it more difficult for the model to retrieve the correct textual descriptions; our model identified it as a hippopotamus amulet instead. The similar situation happened in the bottom Chinese artwork fragment; European motifs rarely appear in Chinese artworks which misled our model to recognise it as a more widely known figure "*qilin*". Our training dataset does not have enough data on this intricate pattern, which raised the difficulty of retrieval task.

Finally, Figure 4.6 demonstrates another four unsuccessful example of image fragment retrieval results from noun phrase queries. Similar to the noun phrase retrieval case we reviewed above, the two samples here also have considerably spo-

Top-5 Ranked Phrases:
head of a kingdom white dog amulet, typical head of a hippopotamus amulet, late middle kingdom osiride head, head of Egypt by the first millennium god symbol

Ground Truth Phrases:
light blue head of a pig with a long nose sniffing the base

Top-5 Ranked Phrases:
a 'qilin' in the centre with a chrysanthemum spray around, dragon among clouds, intricate blue motif with dragons, blue and white motif of several birds among clouds

Ground Truth Phrases:
blue rare European motifs

Figure 4.5: Unsuccessful Example of Text Phrase Retrieval for Given Image Fragment Queries (fine-grained)

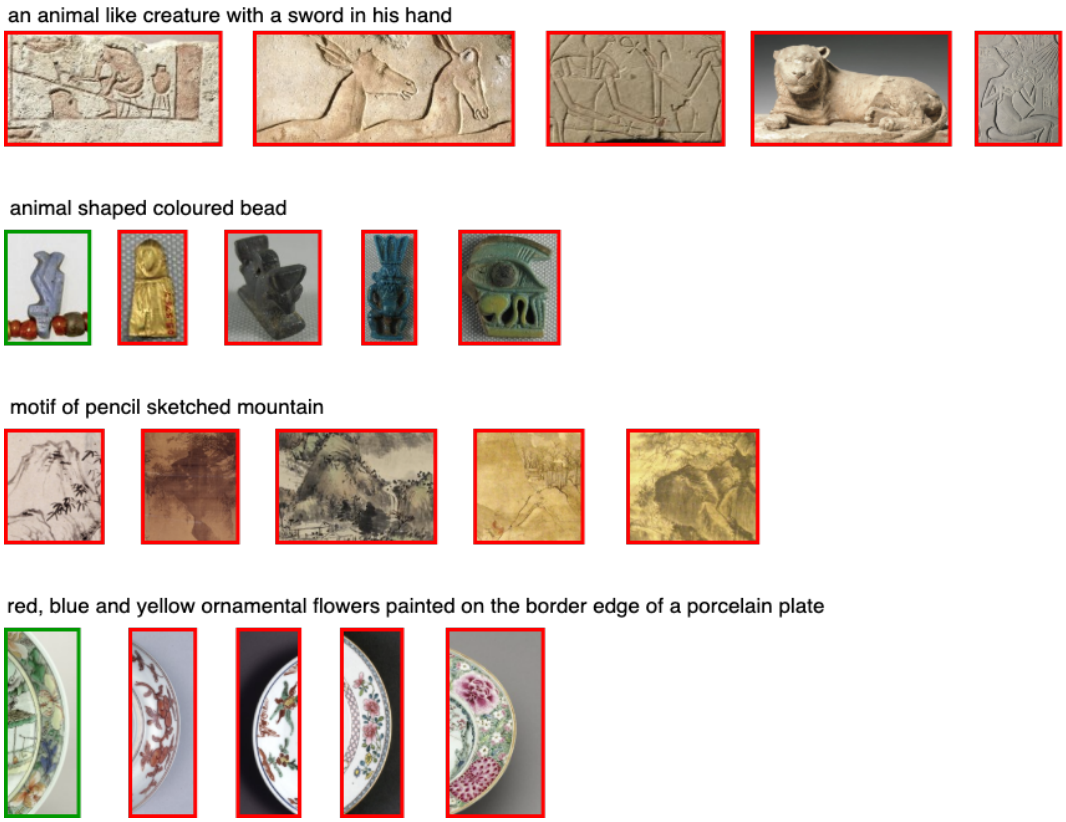


Figure 4.6: Unsuccessful of Image Fragment Retrieval for Given Text Phrase Queries (fine-grained)

radic features. The model usually can pick up the features such as shape and colour, also repeatedly appeared local characteristics, however, for occasional and distinct local features such as "*a sword in hand*", "*animal shaped bead*" and "*pencil sketched mountain*", the model may get baffled, which lead to inaccurate and inadequate results. For example, the third query requires the painting to be pencil sketched which challenged our model besides the fact that it successfully captured "*motif of mountain*" but lost on the "*pencil sketched*". Rare features are proved to be more challenging for our model to retrieve.

4.4.4 Discussion

Generally speaking, Chinese artworks surpassed Egyptian artworks on cross modal retrieval tasks - more detailed descriptive noun phrases provided in Chinese artworks. There are still a large amount of irrelevant noisy textual attributes in Egyptian artwork dataset even after noun phrase extraction, which significantly impedes the process of distinguishing features. Subsequently, adequate but not excellent recalls are expected. There are three possible causes:

- Our training dataset cannot guarantee a balance of features. For example, there are only 6,000 training images for Chinese artworks, and many of them contain diverse shapes and details. Insufficient training features may influence the ability of the model; therefore, cause inaccurate retrieval results.
- As we adopted new ground truth annotations acting testing purpose, considering these features are mostly handcrafted and being almost 100% accurate, potentially increased the chance of wrong retrieval results which were retrieved based on original raw training phrases obtained from the museum. If we were able to also manually annotate artworks in the training dataset to retrain the model, we suppose the retrieval result shall improve.
- One of the crucial requirements for image-text alignment is accurate and proper feature extraction. However, in our case, we used the bottom-up attention mechanism [2], which was designed and trained on real-world images (`VisualGenome`). We all know that artworks usually have widely different representations and requires unique and specific treatment on feature extraction.

Nevertheless, we did not have extra time to redesign our feature learning algorithm to make it more adaptable to artwork representations. To improve our model for better fine-grained cross modal retrieval results, in the next section, we dig into some recent research on the aspect of image-text alignment.

4.5 Future Directions

In this section, we point out some cutting-edge researches on the field of image-text alignment that are related to our methodology as future directions for further

improvements. We are going to discuss in two main directions: the way to align image and text and the way to extract image features (representations).

4.5.1 Similarity or Fusion?

Here we investigated recent research in the field of VQA, the problems of VQA and image-text matching have a lot in common. For example, both accept image and text features as input and then encode. If image-text alignment task is treated as a binary classification problem, the only difference is that the output of VQA task is of multiple classes.

Generally speaking, the processing of image-text matching problems can be divided into "similarity" and "classification". "Similarity" is to use the traditional methods such as cosine similarity or dot product to calculate the similarity between image and text in the same embedding space to judge whether it matches or not. The representative methods are SCAN [15] and VSRN [16]. The "classification" method is to use the neural network to fit a function that is better than the cosine similarity, to determine whether the input image and text feature are a match or not. This input comes from two modes and the output The neural network design for a particular result is generally called "fusion", which is more common in VQA, and the representative method is MTFN [38].

Now let us discuss the pros and cons of traditional similarity calculation and fusion by starting from analysing this problem from a theoretical perspective.

First, we look at a general bilinear fusion method proposed in MUTAN [3]:

$$\text{score} = \sigma\left(\left(x_{\text{img}} \odot x_{\text{txt}}\right)\mathbf{W}\right), x \in \mathcal{R}^{1\times d}, \mathbf{W} \in \mathcal{R}^{d\times 1}$$

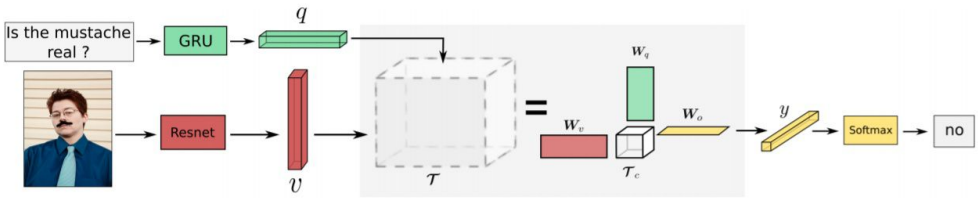


Figure 2: MUTAN fusion scheme for global Visual QA. The prediction is modeled as a bilinear interaction between visual and linguistic features, parametrized by the tensor $\mathcal{T}$. In MUTAN, we factorise the tensor $\mathcal{T}$ using a Tucker decomposition, resulting in an architecture with three intra-modal matrices $\boldsymbol{W}_q$, $\boldsymbol{W}_v$ and $\boldsymbol{W}_o$, and a smaller tensor $\mathcal{T}_c$. The complexity of $\mathcal{T}_c$ is controlled *via* a structured sparsity constraint on the slice matrices of the tensor.

Figure 4.7: Bilinear Fusion in MUTAN [3]

Where x represents extracted feature and $\mathbf{W}$ is a learn-able matrix. It is merely to reduce the dimensions after taking the product of the two features and then output the matching probability through the `sigmoid` function. This is also the primary method used in MTFN [38], which is referred to as bilinear fusion in the following.

Here we use the simplest and most traditional similarity calculation method: dot product here to compare with bilinear fusion:

$$\text{score} = \sigma\left(x_{\text{img}} x_{txt}^{\text{T}}\right)$$

Observing the above two formulas, we can find that bilinear fusion is not much different from the dot product. The dot product is to multiply the corresponding position elements of the two vectors and sum all the elements, while the bilinear fusion is first getting dot product of the corresponding position elements, then get the product of $\mathbf{W}$. This is equivalent to the weighted sum of the elements, that is, when $\mathbf{W}$ is an all-one matrix, bilinear fusion degenerates into a dot product.

From this point of view, bilinear fusion can be understood as weighting different positions when doing dot product when $\mathbf{W}$ is a weight matrix. Weighting helps, but there are two problems:

- Optimisation. Can we guarantee a $\mathbf{W}$ better than the all-one matrix?
- Design issues. The dimension of the weight matrix is $d \times 1$, that is, in the inference, for the element product of any pair of graphic features, the weight of each element position is identical, which is unreasonable.

In many experiments, it has not been shown that fusion must perform better than dot product, and dot product is obviously more straightforward and faster than fusion.

Early Fusion

There are many improvements to fusion this year, one of the most widely discussed is early fusion. "Early" means that comparing to the bilinear fusion mentioned above, it put the multi-modal features into the neural network earlier while there is only one linear layer in bilinear fusion which locates at the end of the entire model. B2T2 [1] compared early fusion and late fusion on the Visual Commonsense Reasoning (VCR) dataset [39], and concluded that early fusion usually works better. However, it can be seen from Figure 4.8 shown below that this comparison can be somewhat unfair, but at least early fusion added more detection information.

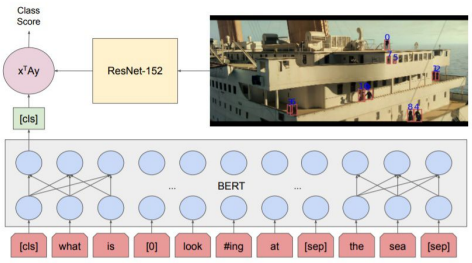


Figure 2: Dual Encoder architecture with late fusion. The model extracts a single visual feature vector from the entire image. Bounding boxes are ignored.

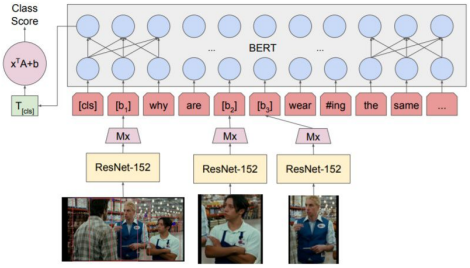


Figure 3: B2T2 architecture with early fusion. Bounding boxes are inserted where they are mentioned in the text and at the end of the input, as described in Sec. 4.

Figure 4.8: Late Fusion (left) & Early Fusion (right) in B2T2 [1]

It should be noted that early fusion is mentioned in many papers, but the structural designs were slightly different, such as RAMEN [32].

Obviously, early fusion has contributed a lot to this aspect, and its performance on VQA is also relatively comparable, but we have not seen enough reliable evidence that early fusion completely exceeds late fusion represented by bilinear fusion. However, at least some aspects have shaken the dominance of bilinear, and these early fusion methods are slightly coarse and have the potential to be better designed. We can expect some future works to break this paradigm.

Back to image-text matching, if the idea of RAMEN [32] early fusion is introduced into the research, considerable progress may be made.

4.5.2 Better Representation Learning?

It seems that the most critical factor affecting the alignment task is the characteristics/feature learnt by the network, and the calculation of similarity is not as crucial to a certain extent. VSRN [16] also proved this point by introducing many components to learn features, and then used the most straightforward dot product to achieve the current State-Of-The-Art performance.

Representation learning embodies its importance in earlier work. The bottom-up and top-down [2] attention mechanism used by our previous model has indirectly accomplished many image-text alignment pieces of research. Afterwards, VQA, image-text matching, and image caption all started to use [2] to extract features, which improved SCAN [15] by a significant amount and shows that a better representation learning model is crucial.

2019 has been a year of the rapid development of representation learning. The leader-board of VCR [39] has many popular large-scale pre-trained models such as ViBERT [17], VideoBERT [34], UNITE [4] and MoCo [11]. Considering the special category of image data we focus on - artworks, here we put more attention on those specialised in artwork feature extraction. Recently, some works [36, 18, 37, 35, 29] have proposed new methodologies for artwork feature extraction and contributed a lot in the field.

Ma el al. [18] introduced a new methodology for artwork representation in 2017: MTMR (multitask and multi-range). MTMR extracts from the fisher vector based on scale-invariant feature transformation (SIFT) from the painting:

- Local, regional and global features
- Multiclass area coding structure
- Multitask learning framework

Figure 4.9 illustrates the overall architecture of the MTMR representation framework. The above, middle and bottom represent different levels of feature extraction using SIFT-based fisher vector: local, regional and global, respectively. Inside each level, there are multiclass area coding and multitask learning structures. These obtained features in different levels will be finally passed to the random forest model as an ensemble mechanism to retrieve a final result.

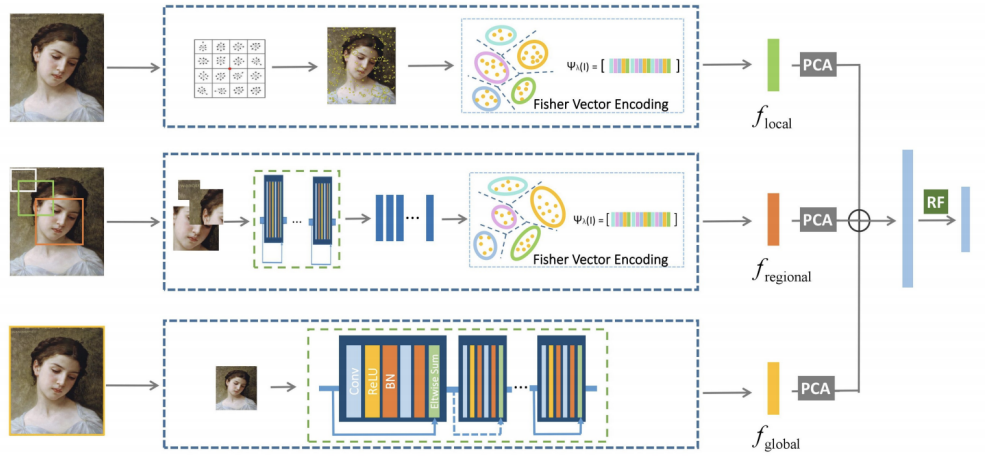


Figure 4.9: Overview of MTMR Representation Framework [18]

Figure 4.10 shows the detailed structure of how multitask learning works in the MTMR framework. The right grey box contains several different tasks which are split to process in parallel. There is a residual network inside the green box, which consists of fully connected layers and convolutional layers for artwork image feature extraction.

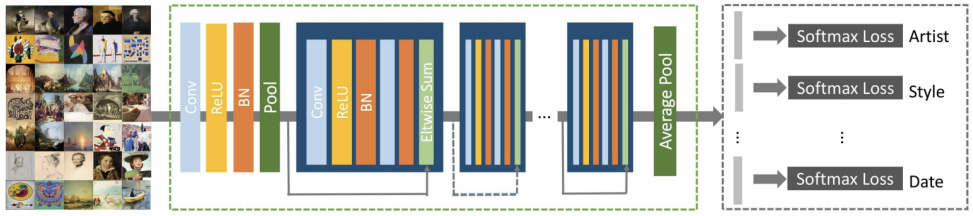


Figure 4.10: MTMR's Multitask Learning Structure [18]

Besides the MTMR framework, in 2019, Shen et al. [29] tried to find almost repetitive patterns from a large number of artworks.

Due to the differences in artistic media (oil paintings, pastel paintings, sketches, etc.) and the inherent deviations in the reproduction process, this goal is more difficult to mine than standard examples. The critical technology is to use self-supervised learning to fine-tune standard depth features on specific art collections to adapt them to this task. Correctly, use spatial consistency between adjacent feature matches as a supervised fine-tuning signal. The adjusted features enable more accurate matching (not affected by differences in style) and can be used with standard pattern discovery methods based on geometric verification to identify repeating patterns in the dataset.

Figure 4.11 demonstrates the feature learning process proposed. First candidate

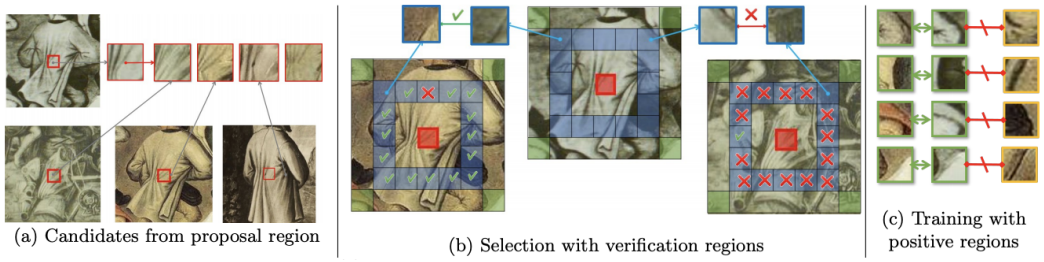


Figure 4.11: Feature Learning Strategy [29]

correspondences need to be obtained, which come from matching the proposal regions in red boxes with the original complete dataset. Next, these correspondences are checked by comparing features from validation regions (in blue). Finally, only positive results from the previous step will be extracted and kept.

The method is evaluated on multiple different datasets and showed significant qualitative findings. In terms of quantitative evaluation, the researchers marked 273 approximately repeating details in the dataset of 1,587 works of art by Lao Jan Bruegel and his studio. In addition to works of art, the researchers also showed improvements in the positioning of the algorithm on the `Oxford5K` photo dataset [27] and historical photo positioning on the LTLL (Large Time Lags Location) dataset [7].

We believe that a multitasking framework focusing on different levels of features like MTMR [18] along with a fine-grained feature matching strategies mentioned in Shen et al.'s research [29] will significantly help improve the fine-grained cross modal retrieval model.

4.6 Conclusion

In this chapter, we first presented a fine-grained cross modal retrieval model based on SCAN. It enables image-text retrieval on a fragment level which potentially eased the artwork annotation task. Next, we conducted several experiments under different settings to test the performance of our model on Egyptian and Chinese datasets by comparing their recall rates. To better illustrate and explain the results, we displayed several successful and unsuccessful retrieval examples from our experiments, which shows our model has the ability to pick commonly shared features out accurately but sometimes gets stuck on rare and unique features. Lack of enough artwork training examples is also suspected as a cause of adequate recalls. We finally pointed out future directions on the image-text alignment task by investigating recent research works and proposed two significant aspects that could benefit our model.

Chapter 5

Conclusion

In this thesis, we first comprehensively reviewed the mainstream proposed image-text alignment frameworks for cross modal retrieval task and investigated SCAN over artwork datasets. Then, we highlighted the deficiencies of the current methods on artworks. Against the shortage of current methods, we proposed a fine-grained cross modal retrieval model which can perform the task on a fragment level. This model adopted the idea of bottom-up attention [2] to capture image regions, stacked across attention [15] for image-text alignment and noun-phrase extraction for more fragmented textual retrieval along with a manually handcrafted ground truth testing set for better evaluations. Experiments demonstrated that this modal could obtain some subtle features from artworks and does not lose significant features in the meantime.

In this final chapter, we summarise our findings and results presented in the thesis and discuss future directions for our research.

5.1 Achievements

The following list highlights the major achievements of this project:

- We employed SCAN to achieve coarse-grained cross modal retrieval on artwork data, helped accelerate the artwork annotation task.
- We modified our coarse-grained cross modal retrieval model to fragment level; this enables the cross modal retrieval task at a more fine-grained level focusing on some subtle characteristics in artworks.
- Through our evaluation, we showed that our design is feasible and performed the cross modal retrieval on a fragment level.

5.2 Limitations

This thesis has presented algorithms and approaches that contribute to solving problems in cross model retrieval for the artworks. Despite the contributions, there are

some limitations to the proposed algorithms and approaches.

The first limitation is related to our image representation learning. By adopting research from a real-world dataset training based such as bottom-up attention, it is highly likely to miss peculiar patterns in artworks. It has been proved that feature learning plays a crucial role in the field of image-text alignment; a novel method targeting artwork feature learning may significantly help the task.

The second limitation is related to our dataset settings. We noticed some unbalance in our dataset that some specific types of artwork do not frequently appear as others, which may affect the training and testing result. As we know that a good computer vision model often requires training on a massive amount of data such as ImageNet [6], a better-structured dataset may help with this research.

5.3 Future Works

There are several future directions for the work this thesis presents. We summarise the discussion of future works for each of the specific topics mentioned in Chapter 4 and propose future directions we wish to pursue.

5.3.1 Image-text Matching

There are already much research has been devoted to this area. Some used simple similarity calculations with attention mechanism; some introduced fusion structure such as bilinear fusion. However, this newly proposed method "early fusion" may be an interesting algorithm to employ for the task of image-text matching.

5.3.2 Representation Learning

Considering the significant differences between the representations of real-world image and artworks, it worths our efforts to develop a novel representation learning mechanism targeting artworks. We believe this should significantly improve the accuracy of cross modal retrieval on artworks.

Appendix A

Implementation Details

This appendix provides some `Python` source code for the implementation mentioned in Section 4.

A.1 Project Structure

In this section, we go through the structure of my implementation using structured tables. This gives the audience and future researchers a brief idea on how the process looks like.

Source Code File	Description
`egyptian_convert.py`	Contains `xml` parser and `json` parser. This generates caption for each picture, extract features base on the corresponding caption and also generate features for each image by using parsed `xml` file
`preprocess.ipynb`	Combines the features extracted from botton-up attention (Egyptian)
`chinese_artworks_convert.ipynb`	Combines the features extracted from botton-up attention (Chinese)

Table A.1: Python Source Code Files for Preprocessing

Source Code File	Description
`data.py`	Class `PrecompDataset` is where we changed the path of extracted phrases and also where to change process methods.
`model.py`	Provides SCAN model based on VSE++.
`train.py`	Provides training process using settings in `model.py`.

Table A.2: Python Source Code Files for Training

Source Code File	Description
`evaluation.py`	Provides testing process using the fragment level annotations and ground truth.
`demo.ipynb`	Provides a demo for testing process.

Table A.3: Python Source Code Files for Testing

A.1.1 Obtain Image Features

We used bottom-up attention [2] to extract features from our Egyptian and Chinese artwork images. This methodolgy used a faster R-CNN and a ResNet101 as core architecture. We obtained a pre-trained model from its Github open repository which was trained on `VisualGenome` dataset consisting a large amount of real-world images. A sample image feature extraction command for Egyptian training images is displayed below:

```
$ python bottom-up-features/extract_features.py
--image_dir artworks/train --out_dir artworks/features
--cfg bottom-up-features/cfgs/faster_rcnn_resnet101.yml
--model bottom-up-features/models/bottomup_pretrained_10_100.pth
```

We save these obtained image features under `/features` directory. These image features of Egyptian and Chinese artworks are available in numpy array format, which can be used for training directly in the future.

A.1.2 Obtain Textual Features

According to captions and corresponding image labels, we can extract the corresponding image names from `xml` and `json` files which contains textual attributes. This extraction process can be done using `egyptian_convert.py`, `python` script then generates captions for each image, make sure that each has a corresponding `.txt` file created. These produced `.txt` files are saved in `/phrase` directory.

To achieve the retrieval on a fragment level, we then extract noun phrases from these `.txt` files. We use `/preprocess.ipynb` to obtain `vocab.json`. The API adopted here was proposed by Handler et al. [10] and available on Github as well.

A.1.3 Training and Testing

For training process, we can simply modify the `PrecompDataset` class in `data.py` to change related processing methods. A sample training command for Chinese training images is displayed below (with i-t average pooling formulation):

```
python train.py --data_name chinese_artworks
--logger_name $RUN_PATH/chinese_artworks_scan/log
--model_name $RUN_PATH/chinese_artworks_scan/log
--max_violation --bi_gru --img_dim 2048
--agg_func=Mean --cross_attn=i2t --lambda_softmax=4
```

For testing process, we load our saved model then run `evaluation.py` script. A sample evaluation command is shown below:

```
from vocab import Vocabulary
import evaluation
evaluation.evalrank("$RUN_PATH/chinese_artworks_scan/model_best.pth.
    tar", data_path="$DATA_PATH", split="test")
```

A.2 Python Snippets

In this section, we briefly display the preprocessing part of our implementation which is also crucial to the entire project as an efficient and accuracy extraction from `xml` files can make the training and testing process easier and more smooth.

Here we focus on two specific files, one is `egyptian_convert.py` which is able to parse `xml` and `json` files. Another one is `preprocess.ipynb` which combines features and vocabularies for each distinct artwork.

A.2.1 Texual Attribute Parser

Here the code snippet of `egyptian_convert.py` is displayed below. Similarly, `chinese_artwork_convert.py` works for Chinese artwork dataset.

```
import xml.etree.ElementTree as ET
import os
import numpy as np
from tqdm import tqdm
import json

# this module used to generate caption for each picture and extract
    features base on the corresponding caption

# parse the xml and get the text for relevant element

def get_text(path):
    tree = ET.parse(path)
    root = tree.getroot()
    file_name = root[1].text
    res = []
    for object in tree.findall("object"):
        res.append(object[0].text)
    return file_name,res

# generate features for each image by using parsed xml file

def xml_parser(path,features_path,save_path):
    list_dir = os.listdir(path)
    features = []
    not_list = []
    for image in tqdm(list_dir):
        file_path = os.path.join(features_path, image.split('.')[0] +
    '.npy')
```

```
28          feature = np.load(file_path, allow_pickle=True).tolist()["
    features"]
29          print(feature)
30          feature = np.array(feature)
31          features.append(feature[:10, :])
32      features = np.stack(features)
33      np.save(save_path, features)
34
35  def json_parser(path):
36
37      # load the json files extract the sentence and corresponding
    picture name.
38
39      with open(path,"r") as file:
40          data = json.load(file)
41          for image in data["images"]:
42              try:
43                  file_name = image["filename"]+".txt"
44                  sentence = image["sentences"]
45                  sentence = sentence[0]["raw"]
46              except:
47                  continue
48              # save the sentence in relevant files
49              with open(os.path.join("../data/phrase_train", file_name),
    "w") as f:
50                  f.write(sentence)
```

A.2.2 Feature Combination

Here the code snippet of `preprocess.ipynb` is displayed below.

```
1  import torchvision
2  import torch
3  import json
4  import cv2
5  import os.path as osp
6  from PIL import Image
7  from tqdm import tqdm_notebook
8  import numpy as np
9  import nltk
10 from collections import Counter
11
12 # combine the features extracted from botton-up attention
13
14 content = json.load(open(osp.join(dataset_name, 'caption_data.json')))
15
16 images = {
17     'train': [],
18     'test': [],
19     'val': []
20 }
21
22 for image in content['images']:
23     filepath = image['filepath']
24     if len(image['sentences']) == 0:
```

```
25          continue
26      images[filepath].append(image)
27
28  for phase in images.keys():
29      features = []
30      for image in tqdm_notebook(images[phase]):
31          file_path = osp.join(dataset_name, 'features', image['filename
        '].split('.')[0] + '.npy')
32          print(file_path)
33          feature = np.load(file_path)
34          features.append(feature[:10, :])
35      features = np.stack(features)
36      np.save(osp.join(dataset_name, phase), features)
37
38  from vocab import Vocabulary, serialize_vocab
39  from collections import Counter
40  import phrasemachine
41
42  threshold = 4
43  counter = {}
44  for phase in images.keys():
45      captions = [x['sentences'][0]['raw'] for x in images[phase]]
46      for caption in captions:
47          temp = list(phrasemachine.get_phrases(caption)["counts"].keys
        ())
48          for key in temp:
49              if key not in counter.keys():
50                  counter[key] = 1
51              else:
52                  counter[key] += 1
53  # Discard if the occurrence of the word is less than min_word_cnt.
54  words = [word for word, cnt in counter.items() if cnt >= threshold]
55
56  # create a vocab wrapper and add some special tokens
57  vocab = Vocabulary()
58  vocab.add_word('<pad>')
59  vocab.add_word('<start>')
60  vocab.add_word('<end>')
61  vocab.add_word('<unk>')
62
63  # Add words to the vocabulary.
64  for i, word in enumerate(words):
65      vocab.add_word(word)
66  serialize_vocab(vocab, osp.join(dataset_name, 'vocab.json'))
67
68  with open(osp.join(dataset_name, 'data.json'), 'w') as f:
69      json.dump(images, f)
```

Bibliography

[1] C. Alberti, J. Ling, M. Collins, and D. Reitter. Fusion of detected objects in text for visual question answering, 2019.

[2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering, 2017.

[3] H. Ben-younes, R. Cadene, M. Cord, and N. Thome. Mutan: Multimodal tucker fusion for visual question answering, 2017.

[4] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. Uniter: Universal image-text representation learning, 2019.

[5] Chm. Deep learning cage match: Max pooling vs convolutions, Sep 2018.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[7] B. Fernando, T. Tommasi, and T. Tuytelaars. Location recognition over large time lags. *Computer Vision and Image Understanding*, 139:21–28, 2015.

[8] R. Girshick. Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1440–1448, Washington, DC, USA, 2015. IEEE Computer Society.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 580–587, Washington, DC, USA, 2014. IEEE Computer Society.

[10] A. Handler, M. Denny, H. Wallach, and B. O'Connor. Bag of what? simple noun phrase extraction for text analysis. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 114–124, Austin, Texas, Nov. 2016. Association for Computational Linguistics.

[11] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning, 2019.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[13] L. Hill. Urban legends in computer vision.

[14] B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional neural network architectures for matching natural language sentences, 2015.

[15] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He. Stacked cross attention for image-text matching, 2018.

[16] K. Li, Y. Zhang, K. Li, Y. Li, and Y. Fu. Visual semantic reasoning for image-text matching, 2019.

[17] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, 2019.

[18] D. Ma, F. Gao, Y. Bai, Y. Lou, S. Wang, T. Huang, and L. yu Duan. From part to whole: Who is behind the painting? *Proceedings of the 25th ACM international conference on Multimedia*, 2017.

[19] L. Ma, Z. Lu, L. Shang, and H. Li. Multimodal convolutional neural networks for matching image and sentence, 2015.

[20] T. B. Museum. The british museum collection, 2020.

[21] T. B. Museum. The brooklyn museum collection, 2020.

[22] T. M. Museum. The metropolitan museum of art collection, 2020.

[23] H. Nam, J.-W. Ha, and J. Kim. Dual attention networks for multimodal reasoning and matching, 2016.

[24] S. Orhan and Y. Bastanlar. Detecting photos with leopards using convolutional neural networks. 10 2016.

[25] A. Ouaknine. Review of deep learning algorithms for object detection, Feb 2018.

[26] R. Parloff. From 2016: Why deep learning is suddenly changing your life, Jun 2019.

[27] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, June 2017.

[29] X. Shen, A. A. Efros, and M. Aubry. Discovering visual patterns in art collections with spatially-consistent feature learning, 2019.

[30] S. Sheng, K. Laenen, and M.-F. Moens. Can image captioning help passage retrieval in multimodal question answering? In L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, and D. Hiemstra, editors, *Advances in Information Retrieval*, pages 94–101, Cham, 2019. Springer International Publishing.

[31] S. Sheng and M.-F. Moens. Generating captions for images of ancient artworks. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM 19, page 24782486, New York, NY, USA, 2019. Association for Computing Machinery.

[32] R. Shrestha, K. Kafle, and C. Kanan. Answer them all! toward universal visual question answering models, 2019.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[34] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid. Videobert: A joint model for video and language representation learning, 2019.

[35] W. R. Tan, C. S. Chan, H. Aguirre, and K. Tanaka. Artgan: Artwork synthesis with conditional categorical gans, 2017.

[36] M. Tomei, L. Baraldi, M. Cornia, and R. Cucchiara. What was monet seeing while painting? translating artworks to photo-realistic images. In L. Leal-Taixé and S. Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 601–616, Cham, 2019. Springer International Publishing.

[37] M. Tomei, M. Cornia, L. Baraldi, and R. Cucchiara. Art2real: Unfolding the reality of artworks via semantically-aware image-to-image translation, 2018.

[38] T. Wang, X. Xu, Y. Yang, A. Hanjalic, H. T. Shen, and J. Song. Matching images and text with multi-modal tensor fusion and re-ranking. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM 19, page 1220, New York, NY, USA, 2019. Association for Computing Machinery.

[39] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi. From recognition to cognition: Visual commonsense reasoning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[40] Z. Zheng, L. Zheng, M. Garrett, Y. Yang, and Y.-D. Shen. Dual-path convolutional image-text embedding with instance loss, 2017.