Sketch Image Recognition Using Deep Features

Yuchao Jiang
School of Computer Science
The University of Adelaide

A thesis submitted for the degree of
Master of Philosophy in the Faculty of Engineering, Computer and Mathematical Sciences
20/11/2016
Contents

List of Figures ix

1 Introduction 1
1.1 Sketch Classification .......................... 1
1.2 Face Sketch Recognition ......................... 3
1.3 Contributions ................................ 5
1.4 Organization of Thesis ......................... 6

2 Background Literature 9
2.1 Sketch Recognition and Deep Neural Networks ............... 9
2.2 Face Sketch Recognition with Photos ...................... 13
2.2.1 Viewed Sketch Based Methods ................... 14
2.2.2 Forensic Sketch Based Methods .................. 15
2.3 Cross-Domain Matching (CDM) ....................... 16
2.3.1 Conventional Metric Learning Methods ............. 16
2.3.2 Deep Networks ................................ 17

3 Sketch Classification 20
3.1 Learning to Classify Sketch Images ....................... 20
3.2 Experiments and Results ........................... 23

4 Face Photo-Sketch Matching with a Semi-Siamese Network 27
4.1 2-Channel Network Based Matching .................. 27
4.2 Semi-Siamese Network Based Matching ................ 30
CONTENTS

4.3 Matching Fusion ........................................ 32
4.4 High Diversity Face Sketch Dataset (HDFS) ............ 33
4.5 Experiments and Results ................................ 37
  4.5.1 Experimental settings ................................ 37
  4.5.2 Evaluation of Individual Approaches .............. 38
  4.5.3 Comparison with Other Approaches ............... 39

5 Discussion and Conclusion ............................... 45

References .................................................. 47
Abstract

Sketch is a special group of images, and the ability to recognize sketches is of great importance for many applications, including the human-computer interaction and childhood education. Previous approaches often pose this as a sub-class of general image classification which can be solved with a conventional pattern recognition method. In this thesis, instead of applying general image classification methods directly to sketches, a model using a new deep neural network that considers more of the unique characteristics of sketches has been developed and studied. Experiment results on the challenging sketch datasets demonstrate the superior performance of this present model in comparison to previous state-of-the-art methods.

In addition to a novel method for sketch image classification, the topic has been further expanded to enable the matching of face sketches to photo images. This is of critical significance, especially in the law enforcement area to identify criminals. Previous work has attempted to address this task by exploring invariant features or looking for a shared subspace. In this paper, an end-to-end method has been proposed whereby the similarity score can be obtained directly when inputting a pair of sketch and photo-face images. In particular, this study investigates matching image pairs with more diversity than previous studies, considering such features as the presence of a beard, or haircut style, etc. The approach taken in this study is one utilizing a CNN based model, which is more robust and applicable to the complexities existing in the real world. The method presented exhibits good performance with both forensic sketch dataset and viewed sketches. Furthermore, in this study, the largest face sketch dataset
has been created which can accommodate, in total, 14750 positive pairs and 985512 negative pairs for evaluation, which will facilitate future research.
Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I also give permission for the digital version of my thesis to be made available on the web, via the University’s digital research repository, the Library catalogue and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signature
Date
Acknowledgements

First I would like to thank Prof. Chunhua Shen, my principal supervisor during my master study years. His dedication to research inspired me a lot, and will definitely influence me throughout my research career in the coming years. Under his supervision, not only have I learnt specific knowledge in the field, but I also know how to think as a mature researcher, including the ability to find unsolved challenges and address them in novel ways.

I would also like to thank staffs and visitors in the Australian Center for Visual Technologies (ACVT). As a non-native English speaker, I made grammar errors in academic manuscripts now and then. I would like to thank my co-supervisor, Prof. Ian Reid, for his time and effort in revising those manuscripts, and Australian Research Council Centre of Excellence for Robotic Vision (ACRV). I also want to thank him and the centre for providing me living cost for a year and a half, and the opportunity to attend summer school and winter school. In a word, I am very lucky to be a member of the ACVT and ACRV.

As a master by research student, I spent most time with PhD students and Postdoctoral researchers. Zhibin Liao, Lingqiao Liu, Mingkui Tan, Guosheng Lin, Yao Li, Dong Gong, Bohan Zhuang, Fayao Liu, Qichang Hu, Lin Wu, Trung Than Pham have helped me a lot on both research and life. Having friends with different culture backgrounds really enriches my life experience.

Finally, thanks go to the excellent support from staffs in the School of Computer Science at the University of Adelaide. Lenka Hill, Thuy Mai, Julie Mayo, Sharyn Liersch and Jo Rogers are administrative staffs of our school. They have offered me enormous support within
the school such as booking flight, which enables me focusing my main
effort on research. Thanks also go to our Head of School, Katrina
Falkner, and our Postgraduate Coordinator Ian Reid. I am really
grateful for what they have done.
List of Figures

1.1 Examples of sketches of the TU-Berlin Benchmark. .................. 2
1.2 Examples of viewed sketch drawn by artists paired with photos from the CUFSF dataset. ................................. 4
1.3 Examples of forensic sketches and their correspondent mugshots. 4

2.1 An example flowchart of traditional sketch classification methods, which includes the hand-craft feature extraction and a classifier to be trained. .................................................. 10
2.2 An example architecture of CNN to be used for classification. Each plane is a feature map. Here the input is a tortoise sketch. In the first layer, the CNN may learn to detect edges from raw pixels, and then use the first feature maps to detect simple shapes in the second layer. The following layers use the shapes to detect higher-level features. Then the last layer classify the the results to classes. 12
2.3 Illustration of metric learning applied to a face sketch-photo recognition task. For simplicity, images are represented as points in 2 dimensions. Red lines indicate false linked pairs and green lines indicate positive linked pairs. We wish to adapt the metric so that positive pairs are close to each other in the specific representation domain. .................................................. 16
2.4 Semantic illustration of the Siamese network architecture that is utilized to learn to compare images. (a) shows the basic Siamese network; and (b) displays the 2-Channel structure. .................. 18
3.1 An example of how large first filter size would influence the performance. The red box refers to the small size filter, and the green box refers to the large size filter. The strokes in the green box have more information than the red box. ........................................ 22
3.2 An example of a neural network with dropout. (a) is the standard network structure, and (b) has some units been dropped. All the connections along with the units are also dropped. ......................... 23
3.3 Random examples of mis-classified sketches. The green labels stand for the benchmarks, and the red labels are the predictions from our network. ................................................... 26

4.1 Illustration of the 2-Channel network based face sketch-photo matching method. The similarity score is the output of this model. ...... 27
4.2 Illustration of the semi-siamese network model for matching. ...... 30
4.3 Structure of the pooling layer. ............................................. 31
4.4 Style templates for synthesising sketches. ............................ 33
4.5 Examples of synthesised sketches and their corresponding original photos. ......................................................... 34
4.6 Examples of photo-sketch pairs that share an identity. .......... 35
4.7 Outline flowchart of the HDFS dataset generation process. Here we use the identity of ‘Daisy Ridley’ as an example. The images in the top square frame are collected from Google Image and filtered artificially considering the identity and the quality. Then those images are randomly split into 2 parts, one part is synthesised into ‘Sketch’ and the other part is kept as ‘Photo’, which form the face dataset. ................................................................. 35
4.8 Performance comparison on both HDFS dataset and CUFSF dataset. .............................. 41
4.9 Ranked verification result photos with a given sketch image. ... 43
4.10 Ranked verification result sketches with a given photo image. Images with red borders are the false recognized .............................. 43
Chapter 1

Introduction

1.1 Sketch Classification

Humans have used sketching to depict our visual world since prehistoric times. Sketching has been considered as one of the most flexible and natural interfaces in the human computer interaction field for some time. Drawing simple sketches can express a great deal of information that can be difficult to explain in text and other forms. It is serves as a communication method with children or illiterate people. With the proliferation and progress moving towards easier and higher level of human-computer interaction, sketching could be a much easier undertaking for many and will certainly continue to have its place in many sorts of applications. A well-designed sketch recognition system will certainly improve the human-computer interaction, thereby proving valuable in many fields, including game design, children’s education, and etc. Example of sketch images are shown in Figure 1.1 [1]. Research on sketches have consequently flourished in recent years, with the development of various applications including sketch classification and recognition [2, 3, 4, 5], sketch based image/3D model retrieval [6, 7], sketch based image recognition [8], and even structure recovery with sketches [9].

Sketch recognition has long been an area of strong interest for scientists in the field of computer vision and graphics. Work on this topic can be traced as far back as the early 1990s [10, 11] Much effort has been devoted to it, but it still poses significant challenges.
Although sketch classification can be regarded as a sub-task of object classification, it retains a degree of uniqueness. Compared with photos, sketches have the nature of: 1) lack of colour and texture information, which may cause distinct categories appear to be similar; 2) the variation of sketches caused by different drawing styles, which would further enlarge the intra-similarities; and 3) and possibly the most high-level and sparse type of visual media that can be understood by humans, which makes the sketch classification problem interesting to discuss, but difficult to solve.

Previous methods for sketch classification mostly rely heavily on the handcraft feature extraction techniques, such as the fisher vector [4], and the geometry properties of sketch shapes [5]. In recent years, deep neural networks (DNN) have significantly improved the performance of object recognition. Furthermore, some work has explored the use of DNN to classify sketches [3, 2] and has achieved state-of-the-art results.

In this study, a new and simple convolutional neural network to classify sketches end-to-end is proposed in which more of the special characteristics of the sketches to improve the recognition accuracy has been used.
1.2 Face Sketch Recognition

Face sketch recognition is of great importance for law enforcement. Automatic retrieval of photos of suspects from police mug-shot databases can quickly help police narrow down potential suspects. However, in most cases, no clear photo image of the suspects is available, and the best substitute information is the descriptions from eye witnesses. Then a sketch drawn by an artist following the recollection of an eyewitness may significantly assist locating the suspect. These sketches are posted in public places and in the media in the hope that some viewers will provide information about the identity of the suspect. This process is slow and tedious and may not lead to apprehension of the suspect [12]. At present this can be made easier by accessing the extensive mugshot database maintained by law enforcement agencies. Automatic searching of a photo database using a sketch drawing is potentially very useful. Not only will it help the police to locate a group of potential suspects, but it may also assist the witness and the artist to modify the sketch drawing of the suspect interactively based on the retrieval of similar images retrieved [13].

Due to the great need of automatic matching sketch with photos, it has long been an important research area, for the reasons indicated above. It can be regarded as a sub-topic of cross-modal image matching and, as such, could be considered the most important and popular area of study. However, matching faces as a subordinate-level or fine-grained visual problem, is also more difficult than basic-level recognition (e.g. cats vs. dogs), even for human beings, let alone the cross-domain issue.

Prior work in the field of computer vision mostly focused on the viewed sketch, drawn by artists while looking at a photo. Examples of viewed sketches paired with photos of the same identity are shown in Figure 1.2 which are from the CUHK Face Sketch FERET Dataset (CUFSF) [14], which are considered the most popular public available dataset used. However, because the photo and the corresponding viewed sketch are sufficiently similar and well aligned, extracting any grayscale descriptor from both is not sufficient to bridge the cross-domain gap, making the problem appear simpler than it really is [15].

In contrast to viewed sketches, forensic sketches, which are drawn based on
eyewitness description, possibly days after the event, are more practical for law enforcement purposes. Figure 1.3 shows examples of forensic sketches and their corresponding mugshots [16]. With such datasets, matching with photos becomes much more difficult and remains relatively unsolved. The main challenge that researchers encounter, is how to reduce the gap between face photos and sketches including:

- **Cross-domain gap between photos and sketches.** For example, as shown in Figure 1.3, the face image is a dense collection of color pixels captured by digital camera, however the sketches are black lines drawings on a flat white background made up by artist.

- **Sketch quality influenced by the sketch generation process.** Many previous studies have argued about this gap [16, 12, 15, 17, 18]. It is mainly affected by the artists drawing skill, and whether there is an inaccurate or incomplete description by the witnesses.

- **Different nature of the sketch and the photo, such as facial features (beards, mustaches, hairstyles, glasses and even age changes between mugshot and the viewed appearance), as well as facial expression and imaging conditions (lighting, camera characteristics, etc.)** [16]. The image pairs in Figure 1.3
are clear examples, in which the person drawn in sketch in the first image pair wears a hat, but does not in the photo, and the person in sketch in the second pair is obviously older than he is in the photo image. The above examples can significantly influence the recognizing performance; however, this parameter has not been investigated rigorously in previous work.

To bridge these gaps, most existing methods either try to build a sub-domain [19] or feature representation [14, 20] that can be shared using both sketches and photos, or by transfer of one of them to the other domain [21, 22] so that they can be compared without the gap. This has been significantly advanced with the development of the convolutional neural network (CNN), which has achieved state-of-the-art performance with numerous applications in computer vision. Sergey and Nikos [23] first propose the idea of using siamese and 2-channel CNN models to compare image patches. However, it was designed for within-domain image pairs.

In this work, the aim is to address the highlighted challenges in the cross-modality matching of high-diversity forensic sketches and photos, and meanwhile invariant to heterogeneity, distortions and robust to mis-alignment. This study reports that a very shallow convolutional neural network (CNN) with a 2-Channel input, outperforms the present state-of-the-art methods. This study also implements an efficient technique for generating a fixed size feature representation which can be imported into any CNNs. Moreover, a new labelled forensic sketch dataset is proposed by synthesising from face photos with Neuralart [24] to learn and evaluate the proposed sketch-photo face matching methodology.

1.3 Contributions

This research study involves two important topics in the computer vision community, \textit{i.e.}, sketch classification and face sketch recognition.

A very simple and end-to-end method for sketch images, which is based on deep learning, is proposed. The main contributions to sketch classification which arise from this study, are as follows.

1. An end-to-end, deep neural network with input of sketch images and output
of class labels is proposed. No further separate feature extraction process nor classifier is required in comparison to previous work in this area.

2. Emphasis is placed on the uniqueness of sketches with regard to real photos. Such characteristics include the sparse and grayscale properties.

A novel approach is proposed for recognizing face sketches with photos to help in the law enforcement area. Although previous methods have demonstrated their ability for matching face sketches and face photos, they are designed for either viewed sketches or forensic sketches derived under ideal conditions. Furthermore, there was no forensic dataset sufficiently large enough for this research. It is proposed that the methods work well on viewed sketches or ideal forensic sketches, but it does not mean that they are able to be extrapolated to predict real situations. In summary, the contributions of this study on face sketch recognition with face photos comprise the following.

1. From this study, an approach is proposed which allows the matching of high-diversity forensic sketches and photos, which are invariant to heterogeneity, distortions and are robust to misalignment. The method is based on the use of the convolutional neural network methodology. The experiments show that this approach achieves a high degree of performance with both the high-diversity dataset and with the previously reported and well-used, viewed sketch dataset.

2. A new labelled forensic sketch dataset is proposed by synthesizing from face photos, a sketch-photo face matching method from which one will be able to further test, learn and evaluate. As far as can be ascertained, this is the largest labelled sketch dataset, and can composite in total, 14750 positive pairs, and 985512 negative pairs for evaluation and validation by further research.

3. An efficient technique for generating a fixed size feature representation to be inputted into CNNs, has also been implemented.

1.4 Organization of Thesis

This thesis is organized as follows:
Chapter 2: Background. This chapter will cover some of the published research on both sketch recognition and the face sketch recognition with photos.

This section will discuss the previous literature on sketch recognition in two categories; one based on the conventional pattern recognition methods, which formed the most significant part of the sketch recognition group, and the other based on the deep learning, which is currently emerging and has achieved significant results. To assist with explaining the architectural design of the convolutional neural networks, which will be used as the core component of this thesis, the basic idea and framework of convolutional networks, will be introduced. There are relatively few labelled sketch datasets, and a brief introduction of the most popular ones in current use, will be reviewed.

For the most recent literature on face sketch recognition, a categorization will be made on the state-of-the-art approaches based on the type of sketches, which comprise viewed sketches and forensic sketches. An outline will also be made of some representative face sketch photo-matching approaches from the considerable amount of literature covering this field. Methods will also be introduced for general cross-domain image matching methods, including the conventional methods which are represented by both the nearest neighbour method, and deep-learning based methods. When analyzing the previous work, a comparison will be undertaken between these published algorithms and the results from the present study.

Chapter 3: Sketch Classification. This chapter will focus on the sketch classification with deep convolutional neural networks. Initially the details of the CNN architecture for sketch classification, particularly as they relate to its comparative benefits and advantages, will be discussed. The proposed network has been fine-tuned from a network that pretrained with the large ImageNet dataset. Furthermore, the modifications made to its structure to fit the application of sketch classification, will be discussed. Then, the process and the parameter setting of the experiments, with the quantified results and results shown in figures, will be discussed.

Chapter 4: Face Photo-Sketch Matching with a Semi-Siamese Network. In this chapter, a deep-learning based 2-channel structure to match face sketch and photo images, which contrasts with conventional methods for match-
ing cross-domain images, will be discussed. This process represents a very simple and straightforward framework, but provides great flexibility as it starts by processing the two image patches jointly, and then outputs a similarity score. A semi-Siamese network based matching method which achieves higher accuracy for matching face sketch and photo images with higher diversity, and which captures higher level semantic information instead of pixel level similarity, will be proposed. Furthermore, an attempt will be made to combine the 2-channel network and the semi-Siamese network together to improve the performance. Finally, to deal with the limitation caused by existing face-sketch datasets, a new labelled sketch dataset, which is likely the largest known, and yet covers the necessary diversities of real-scenario face photo-sketch pairs, will be applied and tested.

In the experimental part, the performance of our method on the face-sketch photo recognition task on both the viewed sketch and forensic sketch datasets, will be demonstrated. It will also be shown that the combination of the 2-channel network and semi-Siamese network yields significantly better results than current, state-of-the-art methodology using different criteria on both viewed sketch datasets and our proposed high diversity sketch dataset. Furthermore, it will be shown that some methods focus more on the pixel level information while others focus more on the high level semantic information.

**Chapter 5: Discussion and Conclusion.** In conclusion, the work performed in this study for both sketch classification and face-sketch recognition, will be summarized about the key findings from the methodology used, and its relative performance as a novel recognition tool.

It will be shown that the novel method developed in this study, solves the face sketch and photo-recognition problem, as well as the sketch classification problem. An analysis will also be undertaken of the methods performance on different datasets (viewed and forensic) as compared with various existing methods, including the 2-channel network and the semi-Siamese network proposed in this thesis and other previous methods. The summary will conclude by outlining the limitations of the method, as well as suggestions for possible follow-up research, including how to make the present method more flexible and automatic.
Chapter 2

Background Literature

2.1 Sketch Recognition and Deep Neural Networks

The majority of prior work on sketch recognition has been aimed at automatically classifying human hand sketches into known categories. They generally follow the conventional image classification paradigm, that is, extracting hand-crafted features from sketch images followed by their input to a classifier. An example of the flowchart of the traditional sketch classification methods is shown in Figure 2.1.

Among those, Eitz et al. [1] developed a bag-of-features sketch representation together with multi-class support vector machines to classify sketches. Li et al. [25] presented a method for the representation and matching of sketches with a star-graph based ensemble, matching strategy by encapsulating holistic structure matching and learned “bag-of-features” models into a single framework. Schneider and Tuytelaars [4] used fisher vector analysis to extract the features and SVM as the classifier to achieve improved results. In contrast to most existing work, which was limited to a particular domain or limited pre-defined classes, Sun et al. [26] targeted the development of a general sketch recognition system to recognize any semantically meaningful object that a child can recognize. They proposed a probabilistic query-adaptive shape topic model for hand-drawn sketches to simulate the generative process of an image and its textual information. This
Figure 2.1: An example flowchart of traditional sketch classification methods, which includes the hand-craft feature extraction and a classifier to be trained.

The approach of deep learning refers to learning from world in terms of a hierarchy of concepts, and avoiding the need for human operators to formally specify all that the computer needs and learning by the computer itself, with many layers of hierarchy of concepts [27]. Deep Learning is about learning multiple levels of representation and abstraction that help to make sense of the data such as images, sound, and text.

Among these, Convolutional Neural Networks (CNN) are a specialized kind of neural network for processing data and have been extremely successful in practical applications. The idea leverages some very valuable and important factors that help improving a machine learning system, which include sparse interactions,
parameter sharing, and equivariant representations, as well as being able to work with inputs of variable size. The characteristic of sparse interactions allows the network to describe complicated interactions efficiently. Parameter sharing allows the use of the same parameters for several functions in a model which reduces the memory requirements in comparison to dense matrix multiplication methodology. Finally, the equivariance to translation makes the CNN model even more powerful by being robust to the shifts and distortions of the input.

CNNs were conceived from the biological standpoint embracing the early work of Hubel and Wiesel on the cat’s visual cortex [28] which contains a complex arrangement of cells. These cells act as local filters over the input space and are well suited to exploit the strong spatially local correlation present in natural images. The CNN structures are mostly designed for the input of images or speech signals which are supposed to be 2D or 3D. A CNN is always comprised of several convolutional layers with subsampling steps, followed by fully connected layers. The operation of convolution is

\[ s(t) = \int x(a)w(t - a)da \]  

(2.1)

and is typically denoted with

\[ s(t) = (x * w)(t) \]  

(2.2)

where the first argument (in this instance, the function \( x \)) is referred to as the input and the second argument (in this instance, the function \( w \)) as the kernel. The output is referred to as the feature map. A CNN automatically learns the values of its filter kernels based on the task undertaken during the training phase. An example of a CNN architecture which is designed for classification, is shown in Figure 2.2.

Except from the convolutional layers, there are some other important layers that will be used in the following sections and chapters in this study. The first is the pooling layer. Pooling layers are always added after convolutional layers, and are a form of non-linear down-sampling. They can reduce the spatial size of the representations to reduce both the size of the parameters and the computation
Figure 2.2: An example architecture of CNN to be used for classification. Each plane is a feature map. Here the input is a tortoise sketch. In the first layer, the CNN may learn to detect edges from raw pixels, and then use the first feature maps to detect simple shapes in the second layer. The following layers use the shapes to detect higher-level features. Then the last layer classify the the results to classes.

requirements of the networks, thereby reducing the possibility of overfitting. Popu-
larly used pooling methods include the max pooling and the average pooling. After several convolutional layers and pooling layers, the fully-connected layers are followed for the higher-level reasoning in the network, in which the neurons have full connections to all the activations in the previous layer. The fully-
connected layers are always used at the end of convolutional neural networks. The loss layers are only used during the training process to compensate for the deviation between the true label and the predicted label. They are normally the last layer in a network. There are various types of loss functions, including the softmax loss, which is often used for classification problems, and the Euclidean loss for real-valued labels.

To achieve greater specificity in the field of sketch classification, Zhang et al. [29] proposed a SketchNet based on deep convolutional neural network for sketch classification. They used a triplet network to identify the latent discriminative structure of sketch images and taking into account the coherent appearance between the sketch and real images of the same category. Seddati et al. [30] proposed a CNN-based method for partial sketch recognition, which could be used for real-time sketch recognition. They used a new multi-task learning architecture to achieve this. Sangkloy et al. [31] trained cross-domain convo-
volutional networks which embed sketches and photographs in a common feature space. Previous work has been limited to classifying only one type of image, either photo or sketch images, due to the limitation of existing sketch dataset size. Sasaki et al. [32] proposed a method to enable the CNN models to classify both types of images by color transforming between photo and sketch images. They achieved this by utilizing color transformed illustration images to augment the training dataset.

A considerable amount of the previous work in this area, as cited in [30, 29], has been based on the TU-Berlin benchmark reported by Eitz et al. [1], which was the first large scale dataset of human sketches with 20,000 unique sketches and 250 object categories. Humans can correctly identify the object category of a sketch at approximately 73%, and this can also be used to generate partial sketches [30]. Schneider and Tuytelaars [4] modified this benchmark to make it more focused on the relevant aspect (ie. what it looks like), rather than the original drawing intention, which results in 160 classes of objects, each one with at least 56 objects. Recently Sangkloy et al. [31] presented a new hand drawn sketch database paired with photos called the Sketchy database. The crowd workers sketched photographic objects sampled from 125 categories and acquired 75,471 sketches of 12,500 objects.

2.2 Face Sketch Recognition with Photos

Automatically matching between face sketches and photos is faced with the problem of how to build the gaps, and it is not feasible to directly apply face-photo recognition algorithms to face-sketch recognition problems. As far as can be established, Uhl and da Vitória Lobo [33] were the first to propose an automatic matching method for face sketches and photos [16]. Studies on matching facial photo and sketch images can be classified based on the type of the sketches used: viewed or forensic. Example images are shown in Figure 1.2 and 1.3, respectively. Viewed sketches are drawn by artists while looking at the photo [12]. Viewed sketches are drawn by artists while looking at the photo [12], and this contrasts with forensic sketches drawn based on the description given by the eye-witness(es), possibly days after the event. Forensic sketches can be misleading
due to errors in witness memory recall that cause inaccuracies in the sketch drawn by a forensic artist [34].

2.2.1 Viewed Sketch Based Methods

Most previous work has relied on viewed sketches, with popular used and publicly available datasets, such as CUFS [21] and CUFSF [14]). Examples of viewed sketches are shown in Figure 1.2. Recently, significant progress has been made and near perfect results on this dataset have been reported.

The first family of approaches focused on transforming the sketches and photos into one common domain, which was done by either synthesizing a face photo to a pseudo-sketch [13, 35, 36, 21], or synthesizing a sketch to a realistic photo [21, 22]. This can reduce the difference between photo and sketch significantly, thus allowing effective matching between the two, with most of the proposed face-photo recognition approaches being straightforward. However, synthesising sketch or photo originally is an unsolved problem. Unsuccessful pseudo-images will add more difficulty to the matching process.

The second family of approaches reduce the modality gap at the feature extraction stage [14, 20] by designing new face descriptors that are more modality invariant, such as the Histogram of Averaged Oriented Gradients (HAOG) [20]. If the inter-modality difference between the extracted features is large, the discriminative power of the classifiers will be reduced [14].

The third family maps images in different modalities to a common linear subspace in which they are highly correlated [19], with methods such as Partial Least Squares (PLS).

The viewed-sketch datasets are generated under ideal conditions; that is, the faces to be studied are in a frontal pose, with normal lighting, exhibit neutral expression, have no occlusions, and the sketches closely resemble the photo to which it is paired. This has the advantage that the cross-domain gap can be analysed as a control variable. However, it is not suitable for real-world scenarios.
2.2.2 Forensic Sketch Based Methods

Matching forensic sketches has become increasingly high-profile, since real-world scenarios only involve forensic sketches, and to achieve a very good performance on viewed sketches does not necessarily mean that this methodology performs well on forensic sketch datasets [37].

Recently, some forensic datasets have been published and corresponding methods proposed. Most of these works are based on descriptors that are invariant to changes in image modalities. In [12], the authors formulated two popular approaches. The first utilized a holistic method that extracts local features computed on uniform patches across the entire face image, from which individual patch similarity scores are combined to calculate an overall sketch to mugshot match score. The second is between individual facial components to compute an overall sketch to mugshot match score. Klare et al. [18] combined feature engineering with a discriminative method. Ouyang et al. [15] also introduced a dataset that combined forensic sketches and caricatures, and a method which generates a representation that combines mid-level and low-level features, that is invariant to both domains.

Instead of investigating new features, Ouyang et al. [17] focused on bridging the modality gap caused by the fading of memory over time and a sketch dataset with memory gap, by introducing a mapping function. Klare and Jain [38], proposed a method termed Prototype Random Subspaces that does not necessitate feature descriptors that are invariant to changes in image modality. The similarity between two images is measured using a kernel function. It is achieved by projecting the local features of images into a linear discriminant subspace (prototypes in each domain), and the similarity of an image is measured against the prototype images from the corresponding modality.

However, all these strategies are designed to deal with situations where a clear and complete sketch can be achieved, and the photos and sketches can be aligned well. However, any particular patch in the sketch image mostly does not correspond to the associated patch in the photo image in real scenarios. In the present study, this issue is resolved by comparing higher level features with a semi-Siamese CNN, which can determine the similarity between the image pairs.
Figure 2.3: Illustration of metric learning applied to a face sketch-photo recognition task. For simplicity, images are represented as points in 2 dimensions. Red lines indicate false linked pairs and green lines indicate positive linked pairs. We wish to adapt the metric so that positive pairs are close to each other in the specific representation domain.

end-to-end, and does not depend on the ability to find a corresponding patch. Furthermore, it is highly robust to any missing or inaccurate details.

Unlike viewed sketches, sophisticated, labelled forensic sketch datasets that are paired with photos, that are both sufficient in quantity and have high diversity in heterogeneity, distortions and mis-alignment, does not exist. The main sketch/photo datasets are [18] with 159 pairs and the IIIT-D dataset [39] with 190 pairs [17]. In this study, a new dataset to compensate and disentangle these deficiencies, and which can composite in total, 14750 positive pairs, and 985512 negative pairs, is proposed.

2.3 Cross-Domain Matching (CDM)

The goal of cross-domain matching (CDM) is to find correlations between two sets of objects in different domains. The main challenge is that the visual content is only similar on the higher scene level, but quite dissimilar on the pixel level.

2.3.1 Conventional Metric Learning Methods

The most popular used method for matching image pairs is simply the nearest neighbour (NN). The aim to match a sketch $x^s$ to a photo database would be
looking for $i^*$ which

$$
i^* = \arg\min_i |f^s(x^s_i) - f^p(x^p_i)|
$$

(2.3)

where $\{x^p_i\}_{i=1}^N$ and $\{x^s_i\}_{i=1}^M$ are the given sets of photo and sketch face images.

The function $f^s(\cdot)$ and $f^p(\cdot)$ can be mapping functions to minimize the difference between distributions of paired instances while keeping unpaired instances apart, which can work in two ways. One is to synthesize images from one modality to the other \cite{13, 35, 36, 21, 21, 22}, so that they can be compared directly. The other one is to embed data (original images or feature representations) of both domain into a new shared latent space \cite{17, 40, 41} and regard each instance into a new shared latent space. Alternatively, $f(x)$ can be a set of transformations to project image pairs into the same feature subspace, such as a mid-level semantic attribute representation \cite{15}, a Fisher Vector representation \cite{42}, or a CNN representation \cite{43, 44, 45, 46}.

And the $|\cdot|$ in Equ.(2.3) indicates some distance metric to measure the distance or similarity between $f^s(x^s_i)$ and $f^p(x^p_i)$, such as L1, L2 \cite{15, 42, 43, 44, 45} and Cosine distance \cite{47}, which can be used for general purposes.

Also some other distance metric can be learnt automatically from data to capture the data of interest. Figure 2.3 illustrates the idea of metric learning with a simple example. Images are taken from the Google Image\footnote{http://www.dailymail.co.uk/news/article-2178040/Picture-perfect-The-amazing-police-artist-sketches-criminals-she’s-seen-look-EXACTLY-like-real-suspects.html}. Mahalanobis distance metric is the most well-studied and successful framework \cite{48}. Cui et al. \cite{49} proposed a Pairwise-constrained Multiple Metric Learning (PMML) for effectively integrating the face region descriptors of all blocks (resp. volumes) from an image.

An advantage of comparing image pairs with NN is that it is more flexible to create a representation, and it is fast and easy to test. However, the biggest drawback is that the $f(x)$ and the metric learning methods are hand-crafted.

### 2.3.2 Deep Networks

In recent literature, deep learning models have been used extensively to solve various machine learning tasks, and achieved the state-of-the-art results. As pio-
neers, Sergey and Nikos [23] first proposed the CNN-based models for comparing image patches, which is able to learn a general similarity function for comparing images patches directly from image data and integrates the tasks of feature extraction and metric learning into a single framework. Figure 2.4 illustrates the basic structures, of which (a) for the Siamese CNN (S-CNN) architectures and (b) for 2-Channel architecture.

The Siamese network is frequently illustrated as two identical networks for two different samples. The structures of CNN1 and CNN2 are always the same, and the parameters can either be shared (Siamese) or unshared (pseudo-Siamese). In 2-channel networks, each input image pair is combined to form a 2-channel image. The decision layers could be a L2 distance [50], or any other functions. The output of the network could be a similarity score, or just \{-1, +1\} denoting non-matching or matching, respectively.

Instead of comparing hand-engineered features with NN, several end-to-end deep Siamese architectures have been developed in recent years with the objective of projecting the images of similar pairs \(i.e.,\) same identity to be closer to each other, while those of dissimilar pairs to be distant from each other.

To be more precise, Hoffer and Ailon [51] proposed the use of S-CNN as a deep-metric learning method for person re-identification. They used two sub-networks which are connected by a cosine layer. Varior et al. [52] also used S-CNN for the same topic. Besides a baseline S-CNN architecture, they also proposed a Gated S-CNN that produces flexible representations for the same image according to the images they are paired with. In addition to person re-identification, Taigman
et al. [53] utilized S-CNN for the face verification, while Tao et al. [54] used this methodology for tracking.

However, the S-CNNs are used for samples from the same domain. Recently, Wang et al. [6] proposed to use S-CNNs to generate similarities for cross-domain issues (sketch and 3D shape). However, they proposed two different CNN models to handle the distinctive intrinsic properties for the two input sources. Wang et al. [55] adopted the Siamese network for product image search. However, the Siamese networks are formulated to extract features, and a cosine function is added to measure the similarity of the input pairs. In this present study, it is shown that the method can be greatly simplified with only one CNN model to solve the cross-domain problems end-to-end.
Chapter 3

Sketch Classification

3.1 Learning to Classify Sketch Images

In this section, a new method to classify sketches is to be proposed. This method is based on CNN networks. The CNN architecture and the key considerations for sketch classification problem compared to conventional classification methods for general photo images, are outlined and detailed. The settings used and the ideas which were undertaken in this study, are shown to be essential for achieving good results on the sketch datasets.

A CNN could have millions of parameters to be trained for a specific problem, and in this case, the sketch classification. If the networks are trained from randomly initialized parameters, this would need corresponding huge number of training images. However, the existing labeled sketch datasets that suitable for this task, have only limited number and categories. To overcome this gap, fine-tuning a pre-trained network with the sketch dataset would be used, which means to initialize the structure and parameters of a network with a pre-trained network, and then continue to train it with the sketch datasets. Most powerful pre-trained networks are trained on the ImageNet dataset [56] which contains 1.2 million images with 1000 categories. The VGG-S model [57] is one of the outstanding networks, which achieved impressive results in challenging established benchmarks on image recognition and object detection problems. Whilst it is not a very deep network, it is deep enough for the sketch classification task. It has 6
convolution layers and 2 fully connected layers.

Although the VGG-S model is powerful, it was designed for ImageNet dataset classification which is for general images. Since sketch as a subclass retains a degree of uniqueness, including the lack of colour and texture information and the variation of sketches, the architecture to improve the performance of classification specially for sketch images, was improved. Details are explained as the followings.

Initially, larger first-layer filters were used. Free-hand sketch, as shown in Figure 1.1, is a very specific subset of the image categories. One of the main difference from general images is the extreme sparsity, which means that a small patch in sketch image is usually composed of one line segment or even just blank, which is not informative enough. Furthermore, the first layer filter is extremely significant since the processing of all the subsequent layers are all based on the output of the first layer. If insufficient and meaningful information is passed on, then all the following process is meaningless. So, in consideration of this special characteristic, the size of the first layer of our network was increased to cover more informational scope. The current trend of research is toward smaller filter size which is around 3 × 3 \[2\] to capture the textured information. And the first layer filter size of VGG-S is 7 × 7. Here we increased it to 11 × 11. An example of how the first filter size would influence the performance, is shown in Figure 3.1.

Another distinguishing characteristic of sketch images compared with general photo images is that the sketches are black-and-white images, which means that the sketch images have only one single channel. However, the original VGG-S are designed for 3-Channel RGB images and the first layer weight matrix of the network is 3-dimensional. One way to deal with this problem is to generate a new 2-dimensional filter with random weights to replace the original first layer, and then train and refine the first layer parameters with more iterations than other layers. While this method is straight forward, as mentioned before, the parameters of the first layer of a CNN is the most sensitive one. A better alternative method would be initialising the parameters of the first layer with a transformation of the pretrained parameters of VGG-S. The parameters of VGG-S are trained based on a large dataset and are therefore meaningful. Following the mechanism that a colored image converting into a grayscale image, the first-layer parameters are converted in the same way from 3 dimensions to 2 dimension.
Figure 3.1: An example of how large first filter size would influence the performance. The red box refers to the small size filter, and the green box refers to the large size filter. The strokes in the green box have more information than the red box.

The parameters are thus still corresponded to the pixels. This will be detailed in Section 4.1.

Furthermore, as one of the most common issues when training a CNN model, overfitting is a serious problem, due to the limitation of the training sketch data size, and the extremely larger number of parameters in our model to be trained. Overfitting means what the network learnt is the result of the training sketch noise, rather than the training sketch themselves. It can seemingly represent the distribution of the training sketch particularly well, but not so for the testing sketch. In most cases, overfitting is a result of a learning model customizing itself too much to the relationship between training data and the labels. Many methods have been proposed to solve this problem. The most frequent method is to terminate the training process as soon as possible when recognizing an overfitting problem on the validation data arises.

Dropout has been shown to be a powerful technique for addressing this problem [58]. The key idea is to randomly drop units (along with their connections) from the neural network during training. An example of the dropout process is shown in Figure 3.2. Dropout is also a technique which approximates the efficient exponential combining of many different neural network architectures and
is therefore able to improve the performance of the network. Therefore, dropout layers were added after the last two convolutional layers in our model. The dropout rate at 0.5 appears to be the optimal for most cases, however, for the sketch classification problem, this was set at 0.55.

Furthermore, softmax regression loss [59] was used for the training process. Since the dataset has 160 classes, to classify the sketches into those classes, the last layer has 160 outputs, of which each represents the possibility of the input belonging to classes 1 to 160. Conclusively, the architecture of our proposed deep neural network is shown in Table 3.1.

### 3.2 Experiments and Results

In this section, the process(es) which was used on our experiments and the classification results, will be detailed.

The designed network was trained and evaluated based on the improved TU-Berlin sketch dataset [4]. There are two popular used labelled sketch datasets for sketch classification tasks. One is the standard TU-Berlin sketch dataset [1], and the other one is the improved TU-Berlin sketch dataset. Although most previous work has been based on the standard dataset which has larger number of sketches,
Table 3.1: 2-Channel Network Architecture

<table>
<thead>
<tr>
<th>Index</th>
<th>Layer</th>
<th>Type</th>
<th>Filter Size</th>
<th>Filter Num</th>
<th>Stride</th>
<th>Pad</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>Input</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>L1</td>
<td>Conv</td>
<td>11×11</td>
<td>96</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>Norm</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>Maxpool</td>
<td>3×3</td>
<td>-</td>
<td>3</td>
<td>0×2×0×2</td>
</tr>
<tr>
<td>5</td>
<td>L2</td>
<td>Conv</td>
<td>5×5</td>
<td>256</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>Maxpool</td>
<td>2×2</td>
<td>-</td>
<td>2</td>
<td>0×1×0×1</td>
</tr>
<tr>
<td>8</td>
<td>L3</td>
<td>Conv</td>
<td>3×3</td>
<td>512</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>L4</td>
<td>Conv</td>
<td>3×3</td>
<td>512</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>L5</td>
<td>Conv</td>
<td>3×3</td>
<td>512</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>-</td>
<td>Maxpool</td>
<td>3×3</td>
<td>-</td>
<td>3</td>
<td>0×1×0×1</td>
</tr>
<tr>
<td>15</td>
<td>L6</td>
<td>fc</td>
<td>6×6</td>
<td>4096</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>-</td>
<td>Dropout</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>L7</td>
<td>fc</td>
<td>1×1</td>
<td>4096</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>-</td>
<td>Dropout</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>L8</td>
<td>fc</td>
<td>1×1</td>
<td>160</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
in this study we chose the improved one, because it is more suitable for the evaluation of sketch classification tasks. As analysed by Schneider et al. [4], the standard one focuses more on how human draw sketches, however, the improved one focuses more on how human understand sketches, which is important for studying sketch classification problem. This improved benchmark excluded all sketches that were misclassified by humans from the standard one, which means that the human performance defines the ground truth. This dataset contains 160 unique object categories, with each of them having at least 56 objects.

Before the sketch images were fed into the model for either training or testing purpose, they were preprocessed to match the pre-trained VGG-S model. They were first rescaled into size $224 \times 224$, then padded with two ‘255’ in the top, bottom, left, and right spatial directions respectively, and finally subtracted the mean of all pixels in the image.

During the implementation, the library MatConvNet [60] was used for both training and testing. This library is based on the Matlab and the toolbox Vlfeat [61]. In each class, 40 sketches are utilized for training, 10 sketches for validation, and the remaining for testing. The optimisation method is mini-batch gradient descent, and the number of instances in one batch is 128. The learning rate of the first layer, the first fully-connected layer, the last fully-connected layer, and the other layers are initialised as 0.005, 0.002, 0.01, 0.001 at the beginning, respectively (momentum=0.9, weight decay=0.0005). Finally, our method achieves 15.46% error rates that outperforms the alternative method, such as the [62] which is based on the same dataset and achieved error rates at 20%.

Some misclassified sketches were chosen and these are shown in Figure 3.3. While some can be found they are not classified into the categories as desired, although they may be meaningful and some may even make sense.
Figure 3.3: Random examples of mis-classified sketches. The green labels stand for the benchmarks, and the red labels are the predictions from our network.
Chapter 4

Face Photo-Sketch Matching with a Semi-Siamese Network

4.1 2-Channel Network Based Matching

Existing face sketch-photo matching approaches generally fall into metric learning with handcraft features. With the development of deep learning and corresponding methods, from the results of the present study, an end-to-end CNN based framework to solve this cross-domain matching problem, is proposed.

Unlike previous face photo-sketch matching methods [14, 20], no direct notion of descriptor exists in this 2-channel architecture, and thus this methodology provides greater flexibility as it starts by processing the two patches jointly. As shown in the top-left part of Fig. 4.1, the 3-dimensional photo image is first trans-
ferred into grayscale, which has only one-dimensional data. Then the sketch and the photo pair are combined together and considered as a 2-channel image, which is directly fed into the first convolutional layer of the network. In this case, the network consists of a series of convolutional, ReLU and pooling layers. The last layer of the network is simply of a fully connected linear decision layer with one output, which is regarded as the similarity score of the input image pair.

The detailed architecture of the network is shown in Table 4.1, which has been modified but still based on the CNN-S [57]. Apart from the first and the last layers, the structure and parameters are taken directly from the original CNN-S as initial weights, and the parameters of all layers are further trained with face sketch-photo image pairs.

In the first convolutional layer of the original CNN-S net, the weight matrix of each filter is 3 dimensional \( \{M^k_{i,j}\}_{k=1,2,3} \), in which each dimension corresponds to one of the RGB channels. However, in our case, the input is a 2-channel ‘image’ and each weight matrix of the first convolutional layer filters should be 2 dimensional. To make full use of the pretrained weights of CNN-S net, we transfer the \( \{M^k_{i,j}\}_{k=1,2,3} \) into \( \{W^k_{i,j}\}_{k=1,2} \) which would be the new weight matrix of the first layer. We first calculate the means of the filter weights through the 3 dimensions, which is computed as:

\[
M_{i,j} = \frac{1}{3} \times \left( \sum_{k=1}^{3} M^k_{i,j} \right)
\]  

(4.1)

Then we duplicate this matrix \( M_{i,j} \) to form a 2 dimensional weight matrix \( \{W^k_{i,j}\}_{k=1,2} \), in which

\[
\begin{align*}
W^1_{i,j} &= M_{i,j} \\
W^2_{i,j} &= M_{i,j}
\end{align*}
\]  

(4.2)

Through this way, all the filters in the first convolutional layers are transferred to accept 2 dimensional ‘image’ input, and meanwhile keep the pretrained information to a great extent.

The last layer of the original network is a fully connected layer, which was used for classification and have 1000 outputs. However, the last layer of our network should have only one output representing for the similarity score of the
Table 4.1: 2-Channel Network Architecture

<table>
<thead>
<tr>
<th>Index</th>
<th>Layer</th>
<th>Type</th>
<th>Filter Size</th>
<th>Filter Num</th>
<th>Stride</th>
<th>Pad</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>Input</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>L1</td>
<td>Conv</td>
<td>7×7</td>
<td>96</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>Norm</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>Maxpool</td>
<td>3×3</td>
<td>-</td>
<td>3</td>
<td>0×2×0×2</td>
</tr>
<tr>
<td>5</td>
<td>L2</td>
<td>Conv</td>
<td>5×5</td>
<td>256</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>Maxpool</td>
<td>2×2</td>
<td>-</td>
<td>2</td>
<td>0×1×0×1</td>
</tr>
<tr>
<td>8</td>
<td>L3</td>
<td>Conv</td>
<td>3×3</td>
<td>512</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>L4</td>
<td>Conv</td>
<td>3×3</td>
<td>512</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>L5</td>
<td>Conv</td>
<td>3×3</td>
<td>512</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>-</td>
<td>Maxpool</td>
<td>3×3</td>
<td>-</td>
<td>3</td>
<td>0×1×0×1</td>
</tr>
<tr>
<td>15</td>
<td>L6</td>
<td>fc</td>
<td>6×6</td>
<td>4096</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>L7</td>
<td>fc</td>
<td>1×1</td>
<td>4096</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>L8</td>
<td>fc</td>
<td>1×1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

input image pair. Both the structure and the pretrained weights of the original network were abandoned, and a new fully connected layer with random initial parameters was constructed.

With this designed structure and initial parameters, the network is further trained with face photo-sketch image pairs. A hinge-based loss term and squared $l_2$-norm regularization was used which leads to the following learning objective function [23]

$$\min_w \frac{\lambda}{2} ||w||_2 + \sum_{i=1}^{N} max(0, 1 - y_i o_i)$$

where $w$ are the neural network weights, $o_i$ is the $i$-th training sample output, and $y_i$ is the corresponding label (with -1 and 1 denoting a non-matching and a matching pair, respectively). Then averaged stochastic gradient descent (ASGD) have been used to train the model.
When the model is trained, it can produce a similarity score of the input face sketch and photo pair. The higher the score, the higher is the probability that the image pair will be the same identity. The idea of this end-to-end model is quite simple and easy to imply. However, it cannot be ignored that such a 2-channel CNN model was originally used for within domain image pair matching, which give rise to the problem that the matching accuracy for viewed sketches is very satisfactory, but it may not be so for the forensic sketch datasets. In the next section, an alternative approach based on a Siamese network is described which may alleviate this problem.

4.2 Semi-Siamese Network Based Matching

To obtain a greater accuracy for matching face sketch and photo images with high diversity, a semi-Siamese CNN model that can capture higher level semantic information instead of pixel level, is proposed. Figure 4.2 shows the flowchart of the proposed method.

As illustrated in [63], deep-learning methods are a representation-learning method with multiple levels of representation, and higher layers of representation amplify aspects of the input that are important both for discrimination and to suppress irrelevant variations. Therefore, delving deeper and extracting higher level information, are crucial for matching images with high diversity.

For the method described in the present study, the input image pair goes through 3 stages to the output of similarity score. In the first stage, the sketch
and photo images are fed into a CNN model (CNN1) separately. The same network structure is then used with the same parameters for the two CNN1s. Since the network trained in [64] resulted in the state-of-the-art result for the face recognition task, use is made of the structure and parameters of the convolutional layers (28 convolutional layers) of this trained model as CNN1. Then, given an input image (sketch or photo), feature maps can be outputted through CNN1. Our input image size is 224×224 for both sources, and each outputs 512 feature maps of size 14×14, which is decided by the property of defined by Parkhi et al. [64].

During the second stage, the feature maps of each image that output from stage one are transferred into a single map of size 64×64 with a pooling and regularization layer, of which the structure is shown in Figure 4.3. The first step is a spatial pyramid pooling [65] combined with regularization. We pool the responses of each filter with bicubic interpolation, which was previously used for upscaling [66]. We use a 7-level pyramid which is \{ 5×5, 4×4, 3×3, 2×2, 1×1, 3×2, 1×3 \}, totally 64 bins. So now there are 512 feature maps each with a vector of size 64. Then a regularization layer across feature maps to compute the over the 512 feature maps was used, which ends up with 64 feature maps, each with a vector of 64. This is then reshaped into one feature map of shape 64×64, and the output map is ready to input into the final stage.
Table 4.2: CNN model structure

<table>
<thead>
<tr>
<th>Index</th>
<th>Layer</th>
<th>Type</th>
<th>Filter Size</th>
<th>Filter Num</th>
<th>Stride</th>
<th>Pad</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>Input</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>L1</td>
<td>Conv</td>
<td>5×5</td>
<td>32</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>Maxpool</td>
<td>3×3</td>
<td>-</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>Dropout</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>L2</td>
<td>Conv</td>
<td>5×5</td>
<td>32</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>Averagepool</td>
<td>3×3</td>
<td>-</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>Dropout</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>L3</td>
<td>Conv</td>
<td>5×5</td>
<td>64</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>-</td>
<td>ReLU</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td>Averagepool</td>
<td>3×3</td>
<td>-</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>L4</td>
<td>Conv</td>
<td>8×8</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The third stage is a 2-channel CNN (CNN2). The architecture of CNN2 is shown in Table 4.2, which is modified from Cifar-Net. Similar to the structure mentioned in Section 4.1, the weight matrix of the first convolutional layer of the original Cifar-Net is transferred from 3 dimensions to 2 dimensions with Equ.4.1 and Equ.4.2. Furthermore, the last layer is reconstructed with only one output and random parameters at the initial stage. The same learning objective function (Eq.4.3) is also used while training. Additionally, two dropouts at layer 4 and layer 8 to combat with overfitting, were added. The input to this part is a 2-channel patch, which is formed by the output of the two pooling layers (Stage 2).

### 4.3 Matching Fusion

After both the 2-channel network based matching and the semi-siamese network based matching, two similarity scores were obtained for each input pair (i.e., $TS$ and $SS$). Each score was mapped into $[0, 1]$ by linear normalization, leading to the normalized scores (i.e., $TS^*$ and $SS^*$). Based on such normalized scores, a
similarity score fusion criterion was defined as:

\[ FS = \lambda TS^* + (1 - \lambda) SS^*, \]

(4.4)

where \( \lambda \) is a trade-off control factor such that \( 0 \leq \lambda \leq 1 \). And in this case, \( \lambda \) is just set as 0.5 to have both similarity score equal contribution. Finally, the fused similarity score \( FS \) was used for similarity prediction.

### 4.4 High Diversity Face Sketch Dataset (HDFS)

A few face sketch datasets for matching with photos have been introduced before. However, the most popular and publicly available datasets from CUHK, the CUFS [21] and CUFSF [14] (examples are shown in Figure 1.2), have been claimed to be far from the realistic conditions [37]; the forensic datasets have limited labelled data, [67] have 190 pairs and [12] has 75 pairs; and others such as proposed in [68], are designed for other applications like landmarks localization and therefore do not have corresponding photo-images. We therefore introduced a new face photo-sketch dataset, significantly larger and more diverse than existing ones.

Photos from the Google Image were collected by querying the identity key-words, and artificially filter with due regard to their identity and quality. The identities are chosen from famous actors and actresses across the world. Each identity has around 30 images. The identity of each image was randomly split into 2 parts with almost equal quantity, of which one part was kept as ‘Photo’ and the other part synthesized into ‘Sketch’. Therefore, there are no intersections of the photo set and the sketch set which share the same original images. The flowchart of this process is shown in Figure 4.7, in which we use the identity
Figure 4.5: Examples of synthesised sketches and their corresponding original photos.

Table 4.3: Examples of the names and amount of photos and the sketches related to the names. Here ‘No. Ph’ stands for number of photos and ‘No. Sk’ stands for number of sketches. These are the identities that have the most number of sketches and photos.

<table>
<thead>
<tr>
<th>Name</th>
<th>No. Ph</th>
<th>No. Sk</th>
<th>Name</th>
<th>No. Ph</th>
<th>No. Sk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Katherine McNamara</td>
<td>30</td>
<td>12</td>
<td>Qi Shu</td>
<td>26</td>
<td>20</td>
</tr>
<tr>
<td>Chloe Grace Moretz</td>
<td>33</td>
<td>10</td>
<td>Ariel Winter</td>
<td>39</td>
<td>8</td>
</tr>
<tr>
<td>Keke Palmer</td>
<td>32</td>
<td>11</td>
<td>Dita Von Teese</td>
<td>5</td>
<td>42</td>
</tr>
<tr>
<td>Melissa Rauch</td>
<td>34</td>
<td>9</td>
<td>Degang Guo</td>
<td>26</td>
<td>21</td>
</tr>
<tr>
<td>Rachel McAdams</td>
<td>31</td>
<td>12</td>
<td>Ian Somerhalder</td>
<td>5</td>
<td>44</td>
</tr>
<tr>
<td>Hanliang Zhong</td>
<td>29</td>
<td>14</td>
<td>Kate Mara</td>
<td>34</td>
<td>15</td>
</tr>
<tr>
<td>Ryan Reynolds</td>
<td>24</td>
<td>20</td>
<td>Dakota Johnson</td>
<td>44</td>
<td>7</td>
</tr>
<tr>
<td>Tom Hiddleston</td>
<td>6</td>
<td>40</td>
<td>Julianne Hough</td>
<td>35</td>
<td>20</td>
</tr>
<tr>
<td>Zhiying Lin</td>
<td>24</td>
<td>22</td>
<td>Wei Tang</td>
<td>36</td>
<td>20</td>
</tr>
</tbody>
</table>

of ‘Daisy Ridley’ as an example. We show in Table 4.3, some examples of the distribution of the identities, with the names of the most prevalent people.

The sketches are synthesized using the ‘Neural Algorithm of Artistic Style’ [24], with which the transfer the original photo images into sketches using convolutional neural networks, was performed.

The idea of the synthesizing process [24] is to separate the representations of content and style in an image, and manipulate both representations independently to produce new, perceptually meaningful images. The content representations are the feature responses in higher layers when input into a Convolutional Neural Network. Furthermore, the style representations are built on top of the filter
Figure 4.6: Examples of photo-sketch pairs that share an identity.

Figure 4.7: Outline flowchart of the HDFS dataset generation process. Here we use the identity of ‘Daisy Ridley’ as an example. The images in the top square frame are collected from Google Image and filtered artificially considering the identity and the quality. Then those images are randomly split into 2 parts, one part is synthesised into ‘Sketch’ and the other part is kept as ‘Photo’, which form the face dataset.
responses in each layer of the network which was originally designed to capture texture information. Here multiple (two) style images to blend multiple artistic styles are used. This study uses the two sketches drawn by artists in Figure 4.4 as style templates. They are both sketches, but drawn by different artists and have different texture. By integrating both styles, we can generate new and more general sketches, and Figure 4.5 shows some examples of synthesized sketches and their corresponding original photos.

Examples of this High Diversity Face Sketch Dataset (HDFS) are shown in Figure 4.6. This new dataset has 64 identities, with 863 sketches and 1159 photos, which has significantly more images than the previous face sketch datasets. It is also more diverse than previous popular face sketch datasets, especially the CUHK, CUFS, and CUFSF datasets. Although the images are not drawn by artists, they have the necessary diversities of real scenario, including (i) different accessories, including glasses, hats, beard, haircut, expression, and makeup, (ii) illumination variance and no alignment patch pairs, and (iii) age changes.

This new dataset will be used to test and train our face recognition models. For the experiments which were undertaken, the identities were divided randomly into a training set $T$ and testing set $V$. If an identity was preset into training set, all the images, including both photos ($P_T$) and sketches ($S_T$), were only to be used during the training process. Meanwhile, both photos ($P_V$) and sketches ($S_V$) of the identities which preset as a testing set, could also only be used for testing. So when testing, no identity of images has been trained before, and we were trying to match totally foreign images.

Furthermore, we generate image pairs $\{(s^i_\alpha, p^j_\beta)\}$ for training and testing, in which $\alpha$ and $\beta$ represent for their identities, and $i$ and $j$ are their indexes in the identities. $s^i_\alpha \in S_T$ and $p^j_\beta \in P_T$ when training and $s^i_\alpha \in S_V$ and $p^j_\beta \in P_V$ when testing. The positive matching ($M$) and negative non-matching ($N$) pairs are defined as

$$M = \{(s^i_\alpha, p^j_\beta)\}, \alpha = \beta, \quad (4.5)$$

$$N = \{(s^i_\alpha, p^j_\beta)\}, \alpha \neq \beta \quad (4.6)$$

and $\forall i \in \alpha$ and $\forall j \in \beta$. 

36
4.5 Experiments and Results

In this section, the performance of our method on face photo-sketch verification task was studied. An evaluation was made of our method, both on our HDFS dataset, as well as the CUHK Face Sketch FERET Dataset (CUFSF). The CUFSF database has 1,1194 people with lighting variations in the set. Each person has a photo and a sketch with shape exaggeration drawn by an artist. Example image pairs of HDFS and CUFSF datasets are shown in Figure 4.6 and Figure 1.2, respectively.

4.5.1 Experimental settings

Preprocessing In this study, no geometric rectification or photometric rectification was applied, and no process to remove illumination variations, which is more practicable and effective in real world applications than previous methods, was taken. There was no preprocessing for any of the CUFSF dataset, which was different from previous studies [14, 69, 37]. For our HDFS dataset, the only preprocessing used in this paper was to crop face from the whole image. For a given image, a Deformable Part Model (DPM) was used to detect the location of the face [43]. Since all the cropped faces are exactly or almost square, they are first resized to 224×224 pixels. Examples are shown in Figure 4.6.

Generating pairs For our HDFS dataset, 50 identities were randomly selected as the training set, and the remaining 14 identities formed the testing set. Followed by previous papers, in CUFSF dataset, 500 persons were in the training set and the other 694 persons were in the test set. To make sure a reasonable proportion of positive and negative pairs was generated, each sketch was paired with all the photos in the same identity as positive pairs. For the negative set, each sketch was randomly paired with 20 photos from different identities as the sketch. For a pair of images \((s, p)\), if \((s_a, p_b)\) are positive examples, we labelled them as \(y = 1\). Otherwise, if \((s_c, p_d)\) are negative, we labelled them as \(y = -1\).

Learning and Optimization Our model is performed MatConvNet [60]. The optimisation method is ASGD, and training is done in mini-batches of size 100.
It’s trained with momentum 0.9, and weight decay 0.0005. During the training of both the 2-channel model (Sec. 4.1) and the CNN2 part of the semi-siamese model (Sec. 4.2), the learning rate for the first and the last convolutional layer is set as 0.02, and the other layers as 0.01. The CNN1 part of the semi-siamese model is not trained any further, and we just make use of the pretrained parameters. The objective function used for training both of the models is Equ.4.3.

Evaluation Criterion  The verification task is assessed on the two datasets. Each sketch is paired with all the photos from the testing set. And the performance is reported with the Receiving Operator Characteristic (ROC) curve, the Verification Rates (VR) and the Rank-1 recognition rate. Specifically, the ROC curve is generated from True Positive Rates (TPR) and False Positive Rate (FPR), which are obtained by predictions of whether the input image pair belongs to the same identity using a number of similarity score thresholds ranging from 0 to 1. The VR at 0.1% FPR and at 1.0% FPR are computed as in [14]. The rank one recognition rate (ROR) for each of the test input image pair of the given dataset are also computed. Here, the ROR is defined as follows [70]:

$$ROR = \frac{n_{si}}{n_s} \times 100\%,$$

(4.7)

where $n_{si}$ denotes the number of pairs successfully predicted as positive or negative, and $n_s$ stands for the overall number of pairs trying to predict.

4.5.2 Evaluation of Individual Approaches

The face photo and sketch matching performance of the proposed approaches based on three different configurations, were also evaluated by: 1) using the 2-channel based model only; 2) using the semi-Siamese model only; and 3) combining the 2-Channel model and the semi-Siamese model together. Table 4.4 and 4.5 shows results with verification rate (VR) scores, respectively. The 2-Channel model and semi-Siamese model display comparable performance on the CUFSF dataset, but there is a dramatic disparity on the HDFS dataset. It shows that the 2-Channel model works very well with the CUFSF dataset, but shows inferior results with the HDFS dataset. The reason is likely to be that the 2-Channel
structure was originally proposed to solve within domain matching problems, and the CUFSF image pairs resemble each other closely, however, the HDFS dataset image pairs are almost totally different in pixel level. This result authenticates that the 2-Channel CNN can perform well on matching visually similar image pairs, but does not on semantic similar image pairs, which share little low-level information. By incorporating the 2-Channel model with the semi-Siamese approach, the matching process combines both low-level and the high-level information, leading to an improved matching accuracy. Therefore, the configuration which was appeared the best to use was the combination of the 2-channel and the semi-Siamese model. Performance evaluation of this configuration is discussed in Section 4.5.3.

The semi-Siamese network (refer to Section 4.2) using different number of convolutional layers in CNN1, was also evaluated. As shown in Table 4.6, we chose the number of layers from \{34, 32, 28\}, which are fully-connect layers (34, 32) and the last convolutional layer (28). Note that the last fully connect layer is our default choice in the experiments of the paper.

### 4.5.3 Comparison with Other Approaches

In the experiments, a comparison was made of the proposed approach with various published methods, including Fisher Vector (FV), CNN feature (CNNfeature) [64], MvDA-VC [69], CITF [14], and CDFL [71]. Figure 4.8 shows the perfor-
Table 4.6: Performance of networks with different number of shared layers

<table>
<thead>
<tr>
<th>Dataset</th>
<th>layer</th>
<th>34</th>
<th>32</th>
<th>28</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUFSF</td>
<td><a href="mailto:TPR@0.01FPR">TPR@0.01FPR</a></td>
<td>0.597</td>
<td>0.763</td>
<td>0.842</td>
</tr>
<tr>
<td></td>
<td><a href="mailto:TPR@0.001FPR">TPR@0.001FPR</a></td>
<td>0.277</td>
<td>0.421</td>
<td>0.522</td>
</tr>
<tr>
<td>HDFS</td>
<td><a href="mailto:TPR@0.01FPR">TPR@0.01FPR</a></td>
<td>0.849</td>
<td>0.712</td>
<td>0.606</td>
</tr>
<tr>
<td></td>
<td><a href="mailto:TPR@0.001FPR">TPR@0.001FPR</a></td>
<td>0.607</td>
<td>0.398</td>
<td>0.279</td>
</tr>
</tbody>
</table>

The performance of each of these methods on both datasets, and are further quantified in Table 4.7, 4.8 and 4.9.

The results of MvDA-VC, CITE, and CDFL are the original results from the authors. Jin et al. [71], already claimed that the CDFL resulted in the best performance compared with CCA, PLS, CDFE, and other 13 state-of-the-art methods within the CUFSF dataset. Therefore, in this study, a comparison was not undertaken with those methods previously mentioned [71] on the CUFSF dataset, but were limited to only the results utilizing the CDFL.

The FV uses the Fisher Vector [72] as the feature and scored with L2 distance. To generate Fisher Vector presentations, a Gaussian Mixture Model with 256 Gaussians was used as the generative model for the patch descriptors. To estimate the parameters of this GMM, sample descriptors were obtained by applying SIFT in the training images, and reducing them from 128 to 80 dimensions using PCA. This process is achieved with Vlfeat [61].

For the CNN feature, the 4096-D output is used for the penultimate layer, as the default features with the network of [64]. Then the similarity of each photo-sketch pair is scored according to the mean Euclidean distance (L2 distance) between the two features. More formally, given the $i$-th CNN features pair, $x_i$ standing for photo feature and $y_i$ for sketch feature, is score for the pair is given by $\frac{1}{n} \sum_{i=1}^{n} ||x_i - y_i||_2^2$. Lower score corresponds to more similar prediction.

The performance of different layer features on both datasets are also shown in Table 4.10. Layer 34 and layer 32 are the second and first fully connected layer, while layer 28 is the last convolutional layer. Layer 28 and 34 work much better on CUFSF and HDFS dataset, respectively. This is reasonable, since the fully-connected layer features represent for high-level semantic information, and thus spacial information is no longer preserved and is still maintained within the con-
Figure 4.8: Performance comparison on both HDFS dataset and CUFSF dataset.

Table 4.7: Comparison of methods on HDFS dataset

<table>
<thead>
<tr>
<th>Methods</th>
<th><a href="mailto:TPR@0.01FPR">TPR@0.01FPR</a></th>
<th><a href="mailto:TPR@0.001FPR">TPR@0.001FPR</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) 2Channel [23]</td>
<td>0.066</td>
<td>0.018</td>
</tr>
<tr>
<td>(2) FV+L2</td>
<td>0.148</td>
<td>0.053</td>
</tr>
<tr>
<td>(3) CNNfeature+L2 [64]</td>
<td>0.234</td>
<td>0.090</td>
</tr>
<tr>
<td>(4) Semi-Siamese network</td>
<td>0.606</td>
<td>0.279</td>
</tr>
<tr>
<td>(5) Semi-Siamese network (fc)</td>
<td>0.849</td>
<td>0.607</td>
</tr>
<tr>
<td>(6) Combination</td>
<td>0.688</td>
<td>0.310</td>
</tr>
</tbody>
</table>

volutional layers. This information is useful for the CUFSF dataset, for which the image pairs are well aligned, however this is not the case for our HDFS dataset.

Furthermore, our proposed method significantly outperforms other approaches on both datasets, and the results show that our method is invariant to heterogeneity, distortions, as well as robust to mis-alignment. Figure 4.9 and 4.10 visualize some top-ranked matched results, which use sketches and photos as the query image, respectively. As shown in these figures, high diversity, such as hats, glasses, beard, or different haircut and makeup, does not influence the performance. The images that are circled with red lines are mismatched ones, which look quite like the query image. The performance for the HDFS dataset can be further boosted if the CNN1 is replaced with the first 34 layers of [64], which is 0.849 True Positive Rate at 0.1% False Positive Rate, and this is shown as ‘Semi-Siamese Network (fc)’.

We also show the performance on CUFSF dataset in Table 4.9. Comparing the
Table 4.8: Comparison of methods on CUFSF dataset with VR

<table>
<thead>
<tr>
<th>Methods</th>
<th><a href="mailto:TPR@0.01FPR">TPR@0.01FPR</a></th>
<th><a href="mailto:TPR@0.001FPR">TPR@0.001FPR</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) 2Channel [23]</td>
<td>0.921</td>
<td>0.713</td>
</tr>
<tr>
<td>(2) FV+L2</td>
<td>0.882</td>
<td>0.692</td>
</tr>
<tr>
<td>(3) CNNfeature+L2 [64]</td>
<td>0.769</td>
<td>0.463</td>
</tr>
<tr>
<td>(4) Semi-Siamese network</td>
<td>0.931</td>
<td>0.668</td>
</tr>
<tr>
<td>(5) Combination</td>
<td>0.998</td>
<td>0.982</td>
</tr>
<tr>
<td>(6) MvDA-VC [69]</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(7) CITE [14]</td>
<td>–</td>
<td>0.987</td>
</tr>
<tr>
<td>(8) CDFL [71]</td>
<td>0.942</td>
<td>0.775</td>
</tr>
</tbody>
</table>

Table 4.9: Comparison of methods on CUFSF dataset with ROR

<table>
<thead>
<tr>
<th>Methods</th>
<th>Photo-Sketch</th>
<th>Sketch-Photo</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) 2Channel [23]</td>
<td>0.818</td>
<td>0.757</td>
</tr>
<tr>
<td>(2) FV+L2</td>
<td>0.804</td>
<td>0.941</td>
</tr>
<tr>
<td>(3) CNNfeature+L2 [64]</td>
<td>0.688</td>
<td>0.805</td>
</tr>
<tr>
<td>(4) Semi-Siamese network</td>
<td>0.771</td>
<td>0.912</td>
</tr>
<tr>
<td>(5) Combination</td>
<td>0.990</td>
<td>0.996</td>
</tr>
<tr>
<td>(6) MvDA-VC [69]</td>
<td>0.563</td>
<td>0.615</td>
</tr>
<tr>
<td>(7) CITE [14]</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(8) CDFL [71]</td>
<td>–</td>
<td>0.813</td>
</tr>
</tbody>
</table>

Table 4.10: Performance of different layer features with L2 distance

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Layer</th>
<th>34</th>
<th>32</th>
<th>28</th>
<th>26</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUFSF</td>
<td><a href="mailto:TPR@0.01FPR">TPR@0.01FPR</a></td>
<td>0.327</td>
<td>0.453</td>
<td><strong>0.769</strong></td>
<td>0.710</td>
</tr>
<tr>
<td></td>
<td><a href="mailto:TPR@0.001FPR">TPR@0.001FPR</a></td>
<td>0.131</td>
<td>0.172</td>
<td><strong>0.463</strong></td>
<td>0.461</td>
</tr>
<tr>
<td>HDFS</td>
<td><a href="mailto:TPR@0.01FPR">TPR@0.01FPR</a></td>
<td><strong>0.633</strong></td>
<td>0.583</td>
<td>0.126</td>
<td>0.084</td>
</tr>
<tr>
<td></td>
<td><a href="mailto:TPR@0.001FPR">TPR@0.001FPR</a></td>
<td><strong>0.307</strong></td>
<td>0.261</td>
<td>0.046</td>
<td>0.033</td>
</tr>
</tbody>
</table>
Figure 4.9: Ranked verification result photos with a given sketch image.

Figure 4.10: Ranked verification result sketches with a given photo image. Images with red borders are the false recognized.
results on both datasets, we find that fisher vector and 2-Channel CNN networks are dependent on the alignment of the input pairs. We also show our performance with Rank-1 recognition rate. The Semi-Siamese Network outperforms previous state-of-the-art methods. The performance can be further boosted by simply average the similarity score of 2Channel and Semi-Siamese Network, which is a significantly improvement over the CITE [14] and CDFL [71]. This further validates the effectiveness of our method on both the forensic dataset and viewed-sketch dataset.
Chapter 5

Discussion and Conclusion

This thesis has presented a detailed study of two important topics in the vision community, sketch image classification and face sketch recognition with face photos. Sketch is a special group of images, which includes general class free-hand sketches, and face sketches for law enforcement. In this thesis, we started with the topic of sketch classification, then this was expanded into face sketch recognition with photos. Both parts are based on deep learning, especially the convolutional neural networks. As most previous work studies these two subjects separately, this thesis presents a pioneering work on studying free-hand sketch classification and face sketch recognition, in a combined manner.

The sketch classification problem has been studied for many years, however in more recent times, research activity has flourished due to the proliferation of human-computer interactions. However, it is significantly more difficult to recognize sketch images in comparison to general image classification, due to the extreme sparsity and inter-class variations. We proposed a new architecture for convolutional neural networks for the sketch-classification problem, in which we seek to make more use of the special features of sketches to improve the recognition accuracy. Through the experiments described in this study, we found that the results are significantly improved over previous methods, even to the extent that false-classified images are more interpretable.

A new method to recognize face sketches with photos to assist with law enforcement issues, has also been proposed, and this methodology generated state-of-the-art results. Previous methods have focused more on the viewed sketch
dataset, which are easier than real scenarios. Furthermore, it was noted that methods that worked almost perfectly on viewed-sketch datasets, were not sufficiently powerful enough for use with forensic-sketch datasets. We also designed a method that not only generated the best performance on the popular-viewed face-sketch dataset, but also worked extremely well on high-diversity, face-sketch datasets. Specifically, instead of designing hand-craft features that attempt to be invariant for both sketches and photos which follows with a classifier, we proposed an end-to-end method such that the similarity score can be generated directly when inputting a pair of sketch and photo-face images. Our final method is a fusion of two deep networks proposed in this thesis. One is a 2-Channel network, and the other is a semi-Siamese network. Furthermore, we also created the largest face-sketch dataset which can composite in total 14750 positive pairs, and 985512 negative pairs for evaluation and verification with further research. Compared with previous approaches, we have empirically shown that the fusion of our two approaches is able to achieve more accurate and robust results of face-sketch recognition with photos.

With the comparison results generated from the two different datasets (viewed sketch and forensic sketch), we found that different methods resulted in significantly different performance, especially depending on how much spacial information was shared by the image pairs. Given the advantages offered by the proposed methodology, it would be interesting to investigate how to best utilize low-level features that could be combined with the high-level semantic information for matching cross-domain image pairs. This may well be achieved by automatically choosing from which layer the Siamese network part should end, and the 2-Channel network part begin.
References


47


REFERENCES

“Conference on Computer Vision and Pattern Recognition, 2016, pp. 5571–5579. 4, 15, 16, 17


REFERENCES


