University of Sheffield

Tracking individuals across multiple scene and shots in TV dramas

Team Delta: Wonwoo Soh, Mingyan Zeng, Peng Xu, Yucheng He

Supervisor: Yoshi Gotoh

in the

Department of Computer Science

July 13, 2020

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Peng Xu, Mingyan Zeng, Yucheng He, Wonwoo Soh

Signature: Peng Xu, Mingyan Zeng, Yucheng He, Wonwoo Soh

Date: May 28, 2020

Abstract

This project researched about tracking people in TV dramas. We have found that there are many differences exist between the characters in TV drama and the characters in surveillance view. The characters' appearance in TV drama changes frequently, so it's better to choose to use facial details to track characters. It's been found that we need to combine detection and comparison approaches to prove that our approaches contain uniqueness compare to existing ones. In this project, HOG and pre-trained R-CNN are used as detection approaches, and Euclidean distance is used as the comparison method. With using these methods, only 10 manually labelled images are required as a reference, then we were able to get about 90% accuracy. Our model can solve some problems of tracking individuals in the image caused by the lack of image information due to poor quality, noise or illumination problems through some preprocess techniques. Due to time constraints, To prove our model can be used in the video, we applied our model in two short video clips. However, the stability and accuracy of our model in more complex videos are still not guaranteed. There are still a lot of methods we have failed to implement. If these methods can be combined in existing models, we think we can get better performance.

COVID-19 Impact Statement

The lockdown imposed because of COVID-19 caused additional challenges for the completion of this project. In the second semester of the project, the university switched to online delivery of all teaching, and university buildings were closed. All project meetings were shifted to email correspondence and video meetings. In addition, our project plan was revised because we could no longer contact to other team members physically to share the ideas and data , it was no longer able to discuss and get feedback from our supervisor face to face and the delays of communication and work progression occurred by team members live in different time zone.

Acknowledgements

First of all, we would like to thank our supervisor, Dr. Yoshi Gotoh, for all the supports and provision of required materials for the project. Although it got harder to contact each other due to the COVID-19 situation, he provided good feedback and supervision to make this project successful. Secondly of all, thanks to Anton Ragni for giving us opportunity to each of us to work with great team members and provides us good feedback and advice for writing this report. Finally, we would like to thanks to everyone who has supported us during the project, with advice and with inspiration.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Tracking

If you are a fan of sci-fi genre or action-thriller movies, you can easily remind the specific scenes when you hear about this paper's topic 'tracking individuals across multiple scenes and shots in the video'. It is recognizing individuals with names by their appearance and tracks them, sometimes even though they change their appearance, across the moving scene. It sounds unrealistic, but the techniques are developed continuously since a few decades ago and in some fields. Not as much as what in the movie does, but in some degree, those techniques are being used.

Before talking about this paper's topic, understanding the primary keyword 'tracking' will be helpful to understand the concept of our article. In European Commission in EUROSUR-2011[7], this documentation describes several concepts in surveillance, and two definitions of them could help us to define what is tracking in this project.

> Identify: to establish the unique identity of the object as a rule without prior knowledge.
> Recognize: to establish that a detected object is a specific predefined unique object.

After thinking about these definitions carefully, the definition of tracking in TV dramas could be explained. In general, it could be a recognition task, or more specifically, it lies between identification and recognition. For example, the project model takes a video as input. When people appear for the first time in the video, the model will identify the person and establish a unique label to this person. Then when this person appears again in the video, the algorithm will recognize the person because the model has identified this person before.

1.2 Motivation and Research Questions

Again, our topic has been researched by many researchers for many years. While we are studying about the subject, we found out that most works are focus on the surveillance, the characteristics of Surveillance videos are pronounced. People appearing in surveillance views often include the entire body, and most of them are in a posture of moving or standing still. Most studies have been used to track people in different cameras. This happens in a short period of time (short time tracking). Hence the appearance and clothing of the people do not change significantly. Although the details of people in views are not quite obvious, the resolution will cause the details of the character to be lost (face organ details, such as eyes, nose, etc., or hairstyles, facial expressions, etc.), but through the process of the entire body of the character, people Tracking in surveillance can achieve a good performance.

However, the characters in TV dramas are very different from the surveillance views. For example, the roles in TV dramas do not include the entire body, and the actions of the characters are not limited to standing still or walking. Due to changes in the plot, environmental factors may also affect tracking performance, such as illumination: overexposure in strong sunlight situation and underexposure in dark condition, or a background that is highly coincident with the colour of the character. The appearance of the character will also have noticeable changes, such as the character's hairstyle, dress and body. Combining these reasons, we have raised a few questions and carried out research on this project around them.

- How to track individuals on TV drama series?
- How effectively the chosen approach performs on tracking individuals in different types of media, such as TV drama series?
- How to evaluate our works on tracking individuals in TV drama series?

More specifically, the ultimate aim of this project is 'tracking individuals in video that containing dynamic scenes'. For more details, we aim to research the topic with few objectives. Throughout the research, we aim to determine how effectively the approach works in dynamic environment. Within that environment, how well the approach could identify the same individual or individuals from different scenes with different characteristics or changed appearances and how many individuals could be identified well with the approach in the scenes with obstructive factors. All these aims can be achieved by objectives of extracting and editing the data set with specific tool, researching the appropriate approaches and adapt them with our thoughts and ideas to the data set and finally, make comparison with our own standards. When we adapt the approaches, not only the existing ones, but also we modify them on our own ways to make the difference, which will ideally make the improvement in various measures.

1.3 Selection of Data

The data set that this project will use is from organization 'TRECVID', conference which sponsored by the National Institute of Standards and Technology and other U.S government agencies, provides sets of data for research purposes[8]. From the conference, they provide different types of videos as data sets.

The data set we got was used in 'TRECVID' in 2013. The title of TV series is 'EastEnders'[9], a British soap opera which has beeen broadcast in BBC One since 1985[10]. There are 244 video files, which is about 300GB and 464 hours running time in total and the video format is MPEG-4/H.264[9]. The truth table is provided in 'TRECVID' website, but it's not used in our project. Since it's almost impossible to watch all the videos and remember which parts to use within 300GB worth videos, we have decided to collect short scenes from the videos so that we can have approximately an one hour video with differently categorized situations. Collecting those scene are done manually by using video editing tool.

1.4 Structure of the Report

This report is divided into several chapters. Chapter 2 presents the initial topics that can be researched about the topic including data analysis. Chapter 3 presents the state of the art of the topic, which shows the literature reviews that are done. Chapter 4 presents the initial studies that our group made with any implementations or demos made. Chapter 5 shows the results and discuss them in detail. Finally, Chapter 6 states the conclusion of this paper.

Chapter 2

Data Analysis

In this chapter, we will show the factors that investigated and considered during the project, mainly focusing on the data analysis. We will show the characteristics of data, how the data is managed, the data selection method and challenges for handling the data.

2.1 Data Analysis

There are several types of videos which the 'TRECVID' provides, and we were able to select 3 video types that could be used for our project[9]. The sample scene, but not exactly from what 'TRECVID' provides, will be shown as an example for each type to show the differences. The first type is the collections of videos from a various news magazine, science news, news reports, documentaries, educational programming, and archival videos. The approximate composition of the scene for the first type is similar to figure 2.1. The second type is the collections of videos from old TV episodes. The approximate composition of the scene for the second type is similar to figure 2.2. Finally, the third type is the collections of videos from airport security cameras and activity detection. The approximate composition of the scene for the second type is similar to Figure 2.3.

Figure 2.1: Example of News magazine, science news, news reports, documentaries, educational programming, and archival video[1]

Within those three video types, we chose to use the TV episodes as our data set to use after comparing them. As mentioned in the aim and objective section, we require the scenes that are very dynamic, which means the variety of situations happen within the scene. As we can see in Figure 2.1, the news and documentaries are very static. Those static scenes are suitable for image recognition as characters in the scene makes very little actions during the scene. Since image recognition is not our main aim of the project, just part of it, it's not preferred

Figure 2.2: TV Episodes[2]

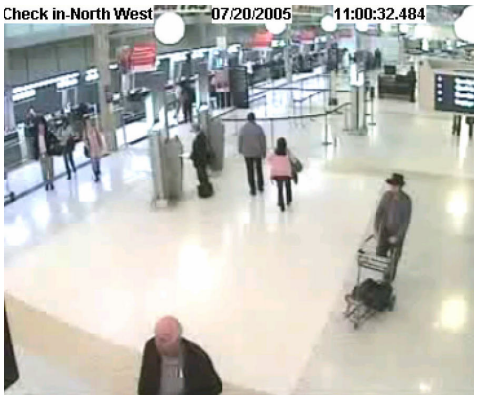

Figure 2.3: Example of Airport Security Cameras and Activity Detection[3]

to use. For the airport CCTV, as we can see from Figure 2.2, the scene is dynamic in some sense since people continuously move around and their behaviours are not identical. Another advantage is the CCTVs are widely spread in the airport, which means tracking certain individual by linking the scenes from different videos could be one of the easy methods we can think of. However, many studies and researches already exist and some of their techniques are actually implemented in some fields. Furthermore, each CCTVs are fixed to record the same place, the degree of angles of scenes are always identical and environmental changes are not easily made. This is far from our requirement. Another consideration is the resolution of the videos. As can be seen from Figure 2.3, the resolution of scenes is not great that we cannot clearly identify the face of each individual. With considering all the factors mentioned above, TV episodes were the best type to fit with our project as they have very dynamic scenes compare to the other two. Within one episode, there exist the environmental changes by moving cameras and viewpoint changes by switching to other cameras in the same scene. Objects move when interacted with the characters and sometimes they become obstacles to recognizing images. Furthermore, in the whole series, the same character shows up in different episodes. Sometimes the appearances or outfits of the character change throughout the series, which gives a challenge whether the tracking techniques can identify the character successfully or not.

2.2 Handling Data

2.2.1 Method

One of the major problems of the project was finding appropriate data. When we look for the data, it was impossible to find the videos which the contents are TV drama episodes and also labelled. Even though we decided to use the data set from 'TRECVID', their data was

also not labelled. There were several methods to label the video data set. The first method is if we use the video itself, use 'video labeler' from 'MATLAB' to label characters in the video. The second method is taking screenshots of characters from the videos and categorize them. Both methods needed to be done manually as there were no tools to do it automatically. We chose the second method since the first method takes too much effort and time compared to the other. We chose 4 main characters, each with 100 images with cropping the images so that nothing else other than the character can be seen from the images. The test set is 200 images with a mix of those 4 main characters and extras so that we can judge our method can find those 4 main characters and distinguish them from others.

The method to create the sample and the test data set follows: First we had to look through all the videos so that we know briefly how the data is formed, how the resolution is and what is the plot and which characters appear and so on. Secondly, we needed to select 4 main characters. The criteria of choosing characters as sample were whether they appear in the show often and whether the character faces different characters in different situations and environments. Then we found most of the scenes which those characters appear and put them together in one video per characters using video edit tool (the tools will be described in Chapter 4). Figure 2.4 shows the screen when edit videos. This allowed us to get the sample data easier since we did not need to look through the whole videos set again if we want to add, replace or find the new scene. Finally, we took 100 screenshots from each sample video sets to create sample data. Some of the images needed to be cropped as they did not contain the character we needed only. For the test data, we again used our edited video sets, find required scenes and took screenshots. The important factor to consider was the test data set should avoid using the same scene that was used in sample data to avoid any bias.

2.2.2 Challenges

There were several challenges while handling the data. The resolution of the video was not the best quality, so we needed to use a video editing tool to look carefully the scenes to crop frame by frame to find the exact scene that we want, otherwise, the image is blurred or easily squashed. Choosing scenes and characters was another challenge for handling the data.

Choosing characters

One of the characteristics of the TV series is there are so many characters, and they all have an their own independent story and sometimes their stories are not collaborated with others. Especially this show 'EastEnders' was broadcast for so long and hundreds of characters exit throughout different time periods. Our data set also mixed different episodes from 2000 to 2005, so not every character appeared in all videos. When we chose an character we thought he/she will appear in many videos, in reality it was hard to acquired sufficient amount of data and therefore we had to discard the previous data and choose different character.

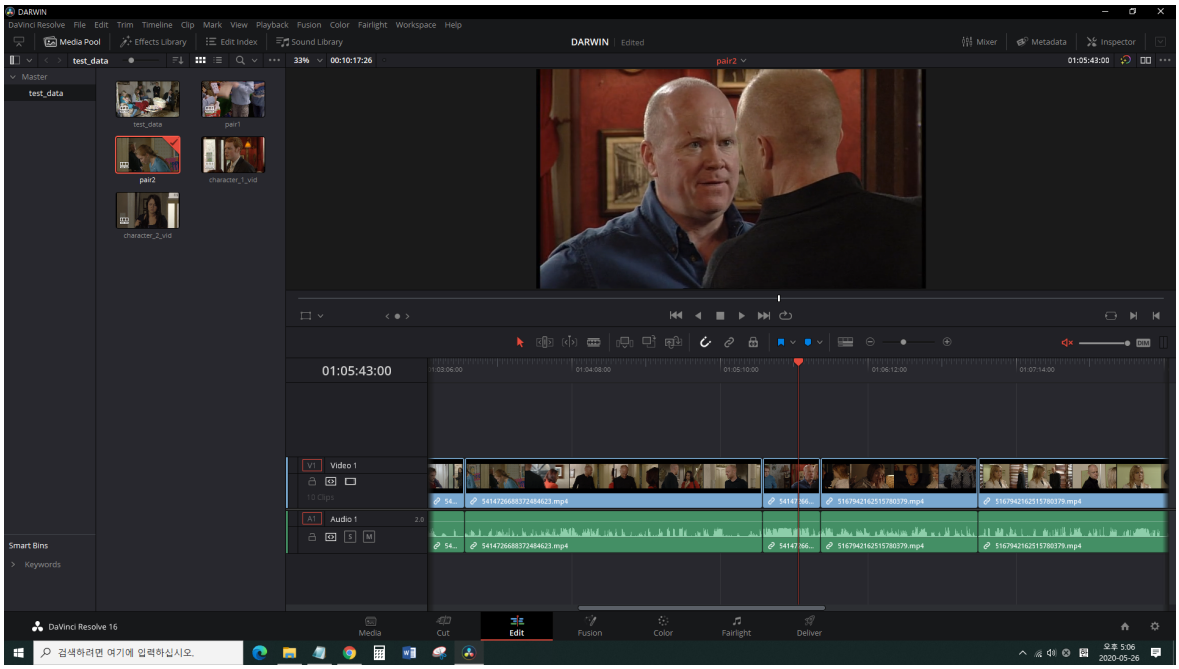

Figure 2.4: Edit data set using video edit tool

Creating sample/test data

As mentioned above, test data is scenes from different videos with a mixed combination of 4 main characters and extras. Figure 2.5 and Figure 2.6 are examples of test data. The main characters are male on the rightmost for Figure 2.5 and ginger-haired male for Figure 2.6. The rest of the characters are defined as extras in our project. If you could notice, they are in the same place, however, the scenes do not show clearly that they are. The only hint to notice it was the scene transition and the exposure of the head of the main character from Figure 2.5 in Figure 2.6 (at the rightmost, between the head of ginger-haired main character and edge of the image). Editor of TV drama series tried to put the characters stories separate and as a result, it was hard for us to find such scenes that they are together.

Sharing data

Another small challenge was sharing of data, due to COVID-19 situation, our members avoided to contact each other physically, therefore the data should be shared through online storage, however, due to unstable internet connection, it was quite challenging to upload the data.

Figure 2.5: One of test data for main character (male on the right) with extra (male of the left)

Figure 2.6: One of test data for main character (male with ginger hair) with extras

Chapter 3

State of the Art

This chapter mainly discusses relevant topics in the field of object detection and re-identification that is useful in this project. First, an evaluation of R-CNNs in the field of object detection is introduced. Second, a brief introduction of the application of R-CNNs in face detection. Third, relevant researches on people re-identification are discussed. Finally, three image preprocessing techniques will be introduced.

3.1 Convolutional Neural Network

3.1.1 R-CNNs

Convolutional neural network(CNN) was heavily used for image classification since Krizhevsky et al[11] show a deeply trained CNN with higher image classification accuracy on the 2012 ImageNet large Scale Visual Recognition Challenge (ILSVRC). However, there no exits of any application of CNN on a field of object detection during the time. SIFT[12] and HOG[13] is the solution for canonical visual recognition task and PASCAL VOC object detection, but the progress has been stalled during 2010-2012. Girshick et al. [14] proposed a Region-based CNN(R-CNN) which has higher object detection performance on PASCAL VOC challenge than HOG-like approaches. R-CNN consists of 2 stages. In the first stage, it uses of selective search method to generate a set of region proposals[15] for the input image, which has been approved on object detection[16]. In the second stage, the set of region proposals was transformed to bounding box and fed to a refined AlexNet[11] and then map to an SVM. The input image is classified by the AlexNet + SVM model and the bounding box of classified images is regressed to reduce the localization errors.

The contribution of the R-CNN is that it transforms CNN on classification tasks to the field of object detection with significant performance improvement on PASCAL VOC challenge. However, the efficiency of the R-CNN is unsatisfied due to (1) every proposals(approximately 2000 for each input image) are forward pass through the CNN in order to extract features, (2) CNN, SVM bounding and box regression is trained separately. To overcome this problem,

Girshick [17] proposes the Fast R-CNN based on his previous research [14]. The approach runs CNN exactly once for each input image. The fixed length feature map is extract from feature map of last convolutional layer and then feeding to the region of interest(ROI) pooling layer. The fast R-CNN jointly train the CNN, SVM and bounding box regression. Softmax is used instead of SVM as the classification layer of the CNN. A linear regression layer is added parallel to the softmax layer to tighten bounding boxes. The performance of Fast-CNN on PASCAL VOC is increased from 53.7%[14] to 66%[17], and the speed is 9 times faster.

The first step of object detection is to generate a set of potential bounding boxes or regions of interest. The proposals were generated by using a fairly slow hand-crafted model such as selective search [18]. The approach consists of two networks, the first, called Region proposal Network (RPN) for generating proposal that share a convolutional layer with the second network which can be any other state-of-art object detection network[11][17] for refining proposals and classifying objects. By sharing the convolutional feature map, only one CNN needs to be trained, the computing cost for generating proposals is nearly cost-free(e.g., 1ms per image).

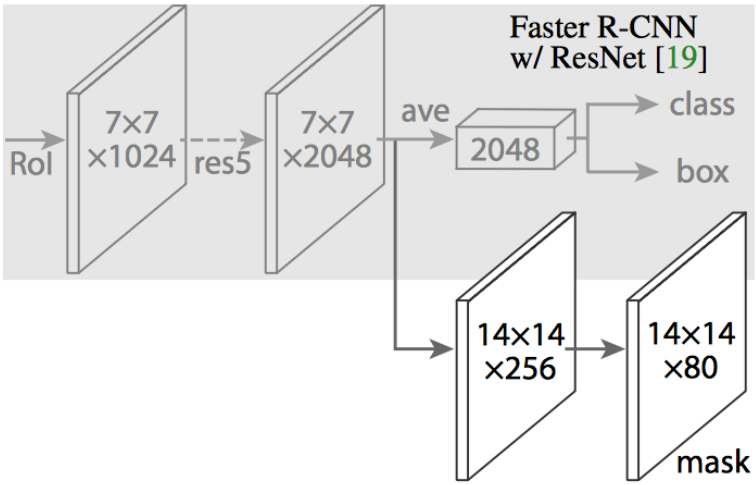

Figure 3.1: Mask RCNN: An FCN is added on the top of ResNet Faster-RCNN

The RPN is a type of fully convolutional network (FCN)[19] and is designed to predict proposals with a wide range of scales and aspect ratios. To deal with those scales and ratios, they introduce anchor in RPN. An anchor is at center of each sliding window location on the shared convolutional feature maps and is scored for predicting how good is the anchor. Following the default setting of [9]. RPN generate 9 anchors with preset scales and aspect ratios. The 9 anchors contain tree scales $(128 \times 128, 256 \times 256, 512 \times 512)$, and each scale contains 3 aspect ratios $(1:1, 1:2, 2:1)$. Therefore, the number of anchors is $9 \times W \times H$, if the size of the shared feature map is $W \times H$. For the generated anchor, the RPN, first determine whether the anchor is the foreground or the background which means to determine

whether has covered the object. The second is to refine the coordinate of the anchor which belong to foreground.

3.1.2 Residual Network

ResNet was first proposed in 2015 by Zhang[20] and won the first place in the ImageNet competition. Afterward, it was popularly used in image detection, segmentation and recognition tasks[18]. The turning point of ResNet is as the depth of the neural network deepens. the accuracy of training set declines. However, we can determine that the degradation problem is not caused by over-fitting, as the accuracy of certain training data will increase, if over-fitting occurs.

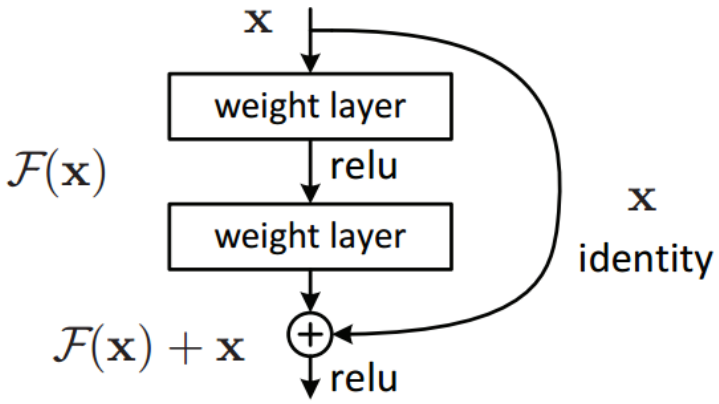

Figure 3.2: Single Residual Unit consists Identity mapping and Residual mapping

ResNet proposed residual unit that allow us to train deeper networks in order to overcome this accuracy drawback problem. The unit consists two new mapping layers. One called Identity mapping, which is the curve on Figure 3.2, and another is the Residual mapping layer which is the rest parts of unit, as shown on Figure 3.2. The residual unit is calculated by formula:

$$y = F(x) + x \tag{3.1}$$

where, x is original input from previous layer, also refer to Identiry mapping. The residual mapping is the diffrence between x and y, therefore, $F(x)$ is the residual. The idea of the residual unit is that if the network is optimum, continue to deepen the network, the residual mapping will be pushed to 0, leaving only identity mapping. In theory, the network has always been in an optimum. Hence the performance of the network will not decrease with depth[21].

ResNet also can be combine with other CNN becomes a better architecture. ResNet has been used in faster-RCNN[18] and Mask-RCNN[22], as illuminate on Figure 3.1. Faster-RCNN share the convolutions layers with RseNet, then both connect to average layer and generate a 2048 diamentions feature map which can be used to image classification or bounding box regression.

3.1.3 R-CNNs Application

So far, we have reviewed the use of CNN features to effectively locate different objects in an image with bounding boxes. [14][17][18]. Kaiming et al.[22] has extended the R-CNN-like approaches to go a step further by carrying out pixel-level segmentation rather than just bounding boxes. Mask R-CNN extends Faster R-CNN by adding FCN [10] branch to output a binary mask for each ROI, which parallel to the existing object detection approaches(e.g., Faster R-CNN). The RoI Pooling layer extracts a small feature map from each RoI, and quantizes the floating-number ROI. The quantization misaligns between ROI and the extracted features which has a negative impact on pixel level mask, while may not have effected the performance of classification. A refined ROI pooling layer is proposed to make the mask work as expected, called: ROI Align. Instead of quantization, bilinear interpolation is used on ROI Align to avoid misalignment.

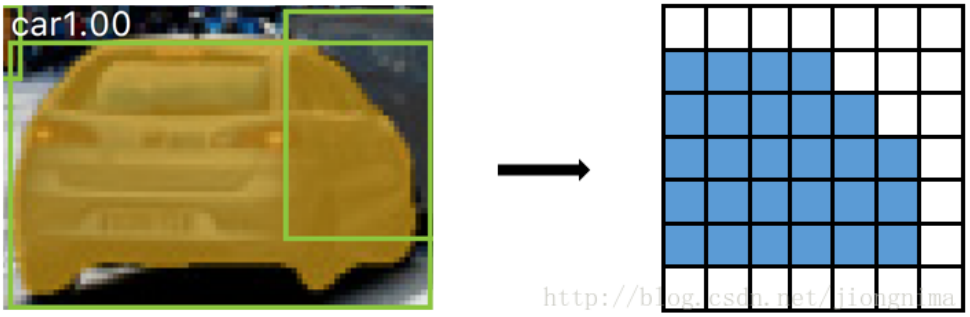

Figure 3.3: The RHS represent the pixel level mask of the car(LHS), the quantization leads a misalignment

A recent study on face detection has demonstrated impressive results by suing R-CNN-likes models. Huaizu and Erik[23] report state-of-the-art result on face detection by directly training Faster-RCNN on the large scale WIDER face dataset[24]. Xudong et al.[25] take the idea a step further by combining feature concatenation, hard negative mining, proper calibration of key parameters to Faster-RCNN. As the consequence, the approach was ranked as one of the best approach on the FDDB benchmark[26]. Coincidentally, Cakirouglu et al.[27] has applied Mask-RCNN to field of face detection. The Mask-RCNN is pre-trained on the face examples collected from PASCAL-VOC, and has been test on the WIDER face dataset.

3.2 Face Re-identification

The people re-identification research has grown rapidly over the past two decades and most of these methods are based on the camera setting, sample set, appearance-base, non-appearance-based, and body model. The figure 3.5 shows the category of people re-identification research approaches. Most studies are based on the surveillance view. The characteristic of the surveillance view is that the people appearing mostly contain the whole body. Although it is impossible to distinguish some small features, such as the use of eyes, ears, nose and other features. People re-identification in surveillance view can still use body characteristics, such as body shape, face shape, hairstyle, and other macro characteristics for re-identification and even make 2d or 3d models to complete people re-identification objective. However, due to the limitations of TV drama itself, the shapes of the characters' bodies are often inconsistent, even only heads of characters appear. Therefore, researching the face re-identification area can better achieve the requirements of this project.

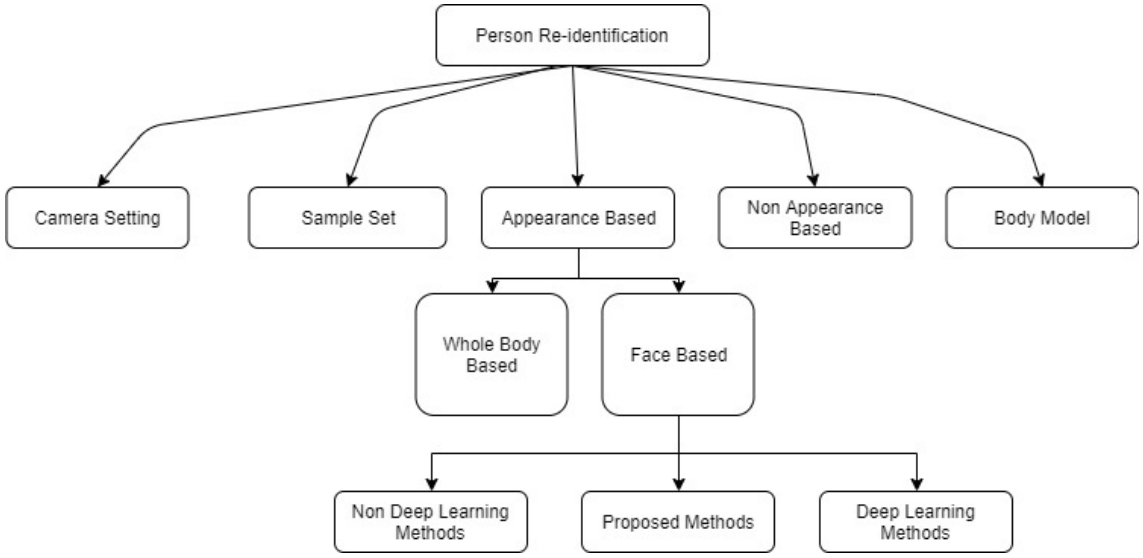

Figure 3.4: People Re-identification

Facial landmarks are the approach for locating facial structures, capable of presenting the eyes, eyebrows, nose, mouth, and jaw of a human face. This technique has been successfully applied in face re-identification with an outstanding performance. Facial landmarks are subset of the Shape prediction problem. This technique requires the face ROI image as an input (such as the bounding box of the face) to locate all key points (face structure) of the corresponding face shape. There are currently two methods of facial landmarks problem that are widely used. One is the Millisecond Face Alignment with an Ensemble of Regression Trees facial landmark detector proposed by Kazemi and Sullivan[5]. This method uses labeled training data, where the label includes specific 2d coordinates for each face structure. By using these training data, a regression face landmark model will be trained, and this model

has excellent performance in predicting facial landmarks. The other method is the supervised descent method proposed by Xiong X and De la Torre F. [28] to obtain a facial landmarks detector. This is a widely used high-performance regression approach. However, this method has two main disadvantages. Jianwen Lou et al.[29] pointed out these disadvantages. One is that this method is highly dependent on the local optimal algorithm. In some cases, this algorithm cannot find the best local optimal value. Another disadvantage is that during the learning process of the algorithm, it may learn in a contradictory descent direction, which will also cause the algorithm to fail to obtain the best performance. Therefore, there are many Facial landmarks detectors derived from SDM.

Deep face feature is another main approach to achieve people re-identification objective. With AlexNet[30] winning first place on ImageNet in 2012, face detection and face recognition objectives uses deep learning approaches have exploded in recent years. Deep face feature approach uses a hierarchical structure method to analyze the shape of input face, illumination conditions, and expression of the face at different levels, which greatly improves the effects of face detection and recognition. Numerous methods explore better effects by combining different network architectures and loss functions. According to the comparison and analysis of Deep face feature methods by Mei Wang and Weihong Deng[31] in 2019, The method proposed by Jingtuo Liu et al.[32] in 2015, combined with CNN-9 neural network architecture and triplet loss function for learning, achieved a performance of 99.77%. Rajeev Ranjan et al[33] published the L2-performance method uses ResNet-101 neural network architecture and L2-Softmax loss function and achieved a score of 99.78%. In 2018, Jiankang Deng et al.[34] made an Arcface architecture used the ResNet-100 neural network architecture and arcface loss function to achieve 99.83%.

3.3 Digital Image Preprocessing

Digital image processing is to use the computer algorithm to perform image processing on the digital image. It allows a wider range of algorithms to be applied to input data digital image processing in order to improve image data (functions) by suppressing unnecessary distortion and/or enhancing some important image functions so that our AI computer vision model can benefit from the improved data. It still plays an important role, even though it is fully researched filed of computer vision for years.

3.3.1 Gaussian Filter

Gaussian filter is a linear smoothing filter, which removes low-frequency digital signal from images. The high-frequency part of the image represents details of the image, the image becomes blurry after Gaussian filter applied, as known as Gaussian blur [35]; To be more

clear,Gaussian distribution function [36] is applied to images to remove low frequency noise.

Gaussian filtering has been widely used in image processing field for many year. Noise is a big issue for preserving edge of a images in computer vision. The application of Gaussian filtering make efforts on generating better quality image for computer vision tasks. The blurry level of Gaussian blur is depends on the parameters of the Gaussian distribution, such as, the stander deviation σ. It is very effective for noise that follows a positive distribution. The one-dimensional Gaussian function is as follows:

$$G(r) = \frac{1}{\sqrt{2\pi\sigma^2}^N} e^{\frac{-r^2}{2\sigma^2}} \tag{3.2}$$

The two-dimensional Gaussian function is as

$$G(u,v) = \frac{1}{2\pi\sigma^2} e^{\frac{-(r^2+v^2)}{2\sigma^2}} [37][38] \tag{3.3}$$

where r, u, v represent the distance from the origin and σ represents the degree of dispersion of the data distribution of normally distributed data. The larger the σ, the more dispersed the data distribution, and the smaller the σ, the more concentrated the data distribution. σ is also known as the shape parameter of the normal distribution. The larger the σ, the flatter the curve. Conversely, the smaller the σ, the thinner the curve.

3.3.2 Median Filter

Median filtering is a typical nonlinear filtering, which can effectively suppress noise based on sorting statistical theory[39]. The basic idea of Median filtering is to replace the pixel with the median of gray value of surrounding pixels. The gray-scale value eliminate isolated noise points by making surrounding pixel value close to the real value. This method can preserve the edge details of the image[40] and blurring caused by linear filters while taking out the impulsive noise and salt and pepper noise[41]. These excellent characteristics are not available in linear filtering. The median filter first generate a filter template, sort the pixel values in the template, and generate a monotonously rising or falling monotonous two-dimensional data sequence. The two-dimensional median filter output is:

$$g(x,y) = medf(x-k, y-1), (k,l \in w) \tag{3.4}$$

where $f(x,y)$and $g(x,y)$ are the original image and the processed image, w is the input two-dimensional template slide on the whole image, as shown on Figure 3.5, usually the size is 3×3 or 5×5 area. It can also be different shapes such as line, circle, cross, circle, etc. Sort out by taking odd data from the two-dimensional template in the image, and replace the data to be processed with the sorted median

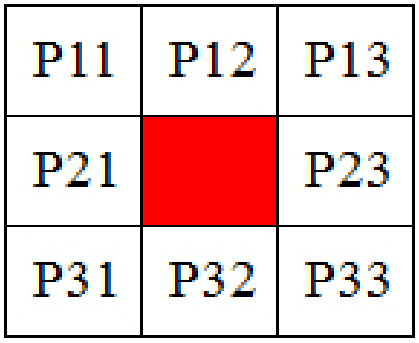

Figure 3.5: Median Blur: the 3 by 3 linear filter slide over image to remove salt and pepper noise

3.3.3 Bilateral Filter

Bilateral filtering is another non-linear filtering, which is a compromise process combining the spatial proximity of the image and the similarity of pixel values, while considering the similarity of space information and gray-scale, to achieve the purpose of edge-preserving denoising, with simple, Non-iterative, local processing characteristics[42]. The reason that the filtering effect of edge-preserving denoising can be achieved is that the filter is composed of two functions: one function is determined by the geometric space distance(Gaussian) and the other is determined by the pixel difference(Gray scale).

The value of the output pixel depends on the weighted combination of the spatial and range values of the neighboring pixels. The formula is as follows:

$$G(i,j) = \frac{\sum_{k,l} f(k,l)w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}[42] \tag{3.5}$$

where function w is the weight coefficient which is the combination of spatial domain and range domain, i, j, k, l represent the coordination of the filter kernel, and The $f(k,l)$ [42] is the pixel values in the point (k,l). The bilateral filter has a Gaussian variance sigma-d than the Gaussian filter, which is based on the spatial distribution of the Gaussian distribution function, so in the vicinity of the edge, the pixels farther away will not affect the pixels on the edge too much, so The pixel values near the edge can be saved, but due to the preservation of high-frequency information, bilateral can only be used to remove low-frequency noise.

Chapter 4

Experiment Design and Implementation

For this project, the characteristics of the characters of people re-identification in TV dramas, are different from the features of the characters of people re-identification in surveillance. When characters appear in TV dramas, most of them appear only with faces. Even if part of the body appears, the part of the body that arises is often not fixed—for example, half, three-quarters of the body or whole body. The actions of people appearing are usually not fixed. For instance, raising hands and even more complex daily movements involve their hands, legs and head. The clothes of the characters in the TV drama also changes frequently according to the change of the plot. All of these points will make it challenging to achieve the re-identification target. But TV dramas provide many close-ups of people's faces with detailed faces features, which makes the face re-identification a more appropriate approach. This project is aiming to track the characters in the TV dramas. The areas of Machine Learning and Deep Learning are developing rapidly, this technology contributes to many areas recently, but it requires numerous data and corresponding labels. TV dramas contain a sufficient amount of data, but the data has no label to train the neural network. Therefore, the objective of this project can only be achieved by combining deep learning, face landmark and face comparison approaches.

4.1 Tools Used for Project

This project planned to use python 3.6[43] as the main language. Python3 provides a rich open-source library of Computer Vision and Deep Learning, which makes implementing related methods and research easier. Other main extension technologies in python version involved are openCV[44], keras[45], dlib[46] and face recognition package[4],etc. To edit the video, we used the freeware version of 'DaVinci Resolve'. Freeware version lacks some features compared to the paid version, but basic features it provides are enough.

4.2 Imaged-based Model Structure

In the field of surveillance or object recognition, the methods that mainly used can be roughly divided into two main categories: image-based and video-based. Intuitively, what we want to solve are video related problems. From this aspect, we should start our project at studying video-based approaches and trying to imply them to solve our problems. However, video-based methods have many drawbacks, which leads to the use of video-based methods is not the best solution. We found that in the existing research, there are not much research used video-based approaches, which makes it difficult for us to get familiar with the operation mechanism and more relevant knowledge of this area in a short time. For the initial stage of our project, video-based approaches are sophisticated. Fewer learning materials will make it difficult for us to get started, and it will be more difficult for our implementation stage in the future.

The image-based approach can provide more convenience for our research. Here we have to emphasize one point, although we are dealing with video-related issues, using an image-based approach is not off-topic. Converting each frame of a video into an image and then processing the images is the primary method of use in most objective recognition researches today. First, there has been quite a lot of research in this area in recent years, so we can quickly become familiar with the relevant knowledge we need and start promptly to deal with our problems. Because the video is converted into images, this allows us to manually select a few relatively easy-to-process images at the beginning of the project as a start. Because we only need to handle a minimal amount of data in the initial stage, once a problem occurs, we can analyze and try to solve the problem more quickly. By treating issues from simple to complex in this way, we can gradually add functions to the basic algorithm we set and finally we will complete all the requirements we initially set.

In recent years, because of the explosive increase in data, deep learning has many excellent performances in dealing with problems in many fields. In the initial stage of the project, we considered using deep learning to solve our problems straightforwardly. However, after careful investigation, we found some issues. Deep learning can be understood as a black box. Deep learning requires a large amount of labelled data as a training set. What the Neural network model can do depends on what kind of data and true labels you use as input. For example, if all images in the dataset are images of people, each image has a bounding box with a corresponding person as a true label. Then after training, this deep learning model can predict the bounding box of people in the new image.

Deep learning is a very straightforward method, but this is not applicable in our data set because our data do not have any true labels. If we need to design a deep neural network that can meet our requirement, then our data set requires not only the bounding box of each face in the picture but also the name of the corresponding face as true labels. It is impossible to achieve in our TV drama data set, so we need to carefully evaluate our task and find a

way to solve it. The first thing we need to do is to understand what we are going to do. In our project, the task can be divided into detection and comparison. All we need to do is to solve these two problems. Detection is a problem that can be solved with deep learning. It does not mean that we need a true label to train our model. We can use a pre-trained model to address the detection problem. For our comparison task, we need to find a stable method that does not require training data or true labels. Therefore, we can describe our solution as face detection, face extraction, and face recognition.

The figure 4.1 shows the structure of the approach of this project. The first target is to find the faces of the characters in the TV drama data set. According to the above section described, the project will choose a pre-trained ResNet[20] to find the ROI (bounding box) of all faces. After using face detection to find the faces of all people, the next step needs to be applied a face extraction method to get the face landmarks of all fronts. The method proposed by Kazemi and Sullivan[5] will be used as the start of the initial study, because the technique is included in the dlib library [46], which makes the initial implementation and analysis of the project manageable. After obtaining landmarks through this method, a global similarity transformation approach will align all landmarks. Finally, the joint Bayesian approach is used to compare and classify the similarity between pictures.

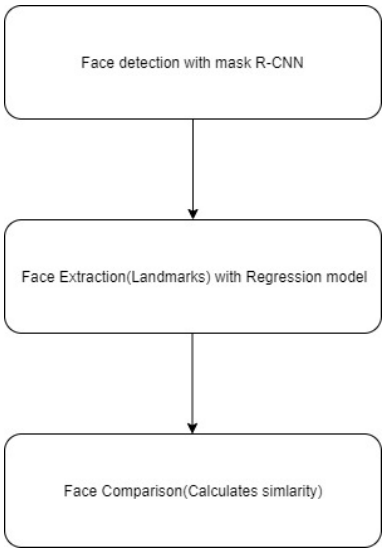

Figure 4.1: simple structure

4.2.1 Face Detection

The structural method involves the model that needs to be trained, however, they are not models with explicit requirements. In this project, we are using two detection models: a Histogram of Oriented Gradients(HOG) and a modified ResNet-34. These models are all

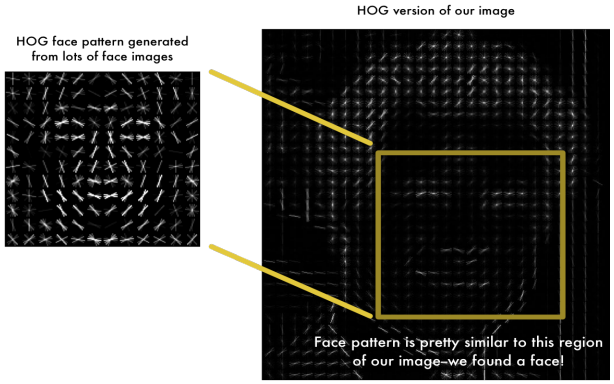

Figure 4.2: The HOG pattern generated from face images, the white arrows represent the darkness/brightness direction of pixels. Credit by Adam[4]

used to complete specific broad objectives(face detection, get landmarks from box-bounded faces). These two models are pre-trained on Labeled Faces in the Wild(LFW) which contains 3 million human face images. The application of these models to our project overcomes the problem of the provided data set are unlabeled.

The principle behind HOG is replacing every single pixel in the image by an arrow. The direction of the arrows is called gradients. It can be calculated by comparing the directly surrounding pixels of the current pixels. The arrows direct to the direction with a darker greyscale. However, saving the gradient for every single pixel are somewhat over detail, we would see the basic pattern of the image at a higher level to find the face location.

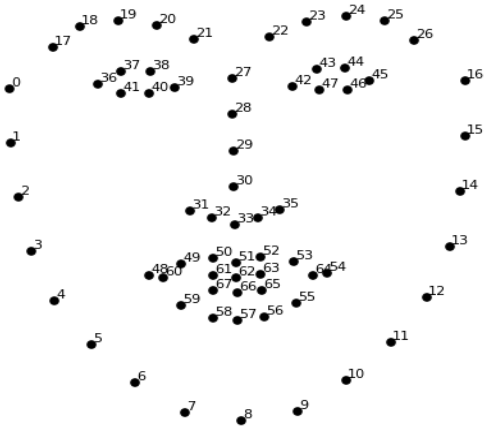

Figure 4.3: The 68 landmarks of every face.[5]

In our project, We divide the image into small squares of 16×16 pixels. In each little square, we calculate gradients in each principal direction. Then we replace the little square with the

arrow with the most direction. The original image now is covering to a simple representation with the basic structure. Finally, the generated image representation is compared to other known HOG pattern that extracted from the LFW database to find the face bounding box.

We review ResNet in detail on Chapter 3.1.2, so we will not repeat it here. However, the number of filters reduced by half to fit our project.

4.2.2 Alignment: Facial Landmark Detection

We extracted the faces from images from the previous step. However, the direction of faces are matters to computation. To solve this problem, we warp each image so that the facial feature(e.g., eyes, lips and nose) are always in the sample place. The facial landmark estimation[5] is used to account this. The principle of the estimation algorithm come up with 68 specific points on every face, as shown in Figure 4.3. Then, a pre-trained algorithm can find theses particular points on any face. Once we find these landmarks, we use them to distort the image and centre the eyes and mouth, etc. By convenience, in our project, we use dlib[46] library to achieve this step.

Figure 4.4: Use of 68 face landmarks to find the facial features

4.2.3 Facial Information Extraction

The critical challenge of face recognition is how does the computer recognize faces apart. The most fundamental solution is comparing the face detected from the previous step with known labelled faces. The problem of this solution is that when the images are in billions, trillion levels, it can not be possible to compare every labelled face cyclically. Hence, we need an approach to extract facial information from each face, such as nose length, mouth size, eye colour, etc. However, facial information is meaningless for the computer. Our solutions are used 128 dimension vector(embedding) to quantify the face. It reduces complex raw

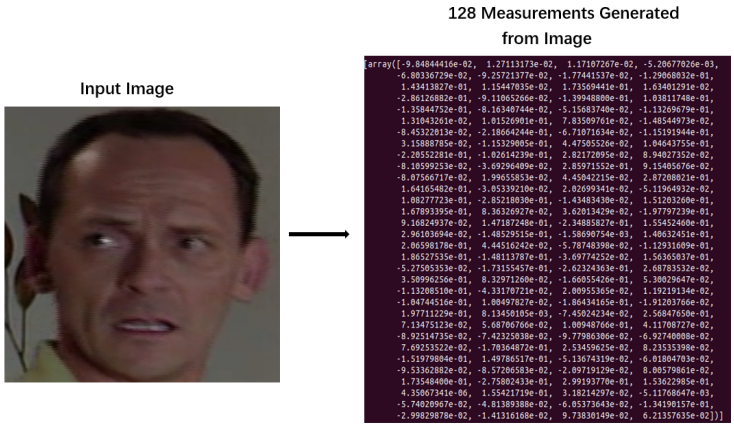

Figure 4.5: Facial extraction via OpenFace[6] method generates a 128-d feature vector per face.

data(images) to list of computer-generated numbers(vectors). We train CNN to generate 128 measurements of each face.

The training process is done using triplets, which processing 3 face images at each time. A single triplet training step takes two example images of the same person and a random face from the dataset, which is different from the other two images. The CNN quantifies and constructs 128-d vector for each face. Then, tweak the weights of the neural network to make sure the 128-d vector of first two faces are close together and away from the third face, as shown on 4.6. The process of facial information embedding by training CNN requires a large amount of data and reliable computation. It takes more than 100 hours of continuous training to get good accuracy on Nvidia 1070. In this case, we directly use released trained networks on OpenFace[6], which perfectly fits our project.

4.2.4 Class Set

All the characters appearing in the input videos are unlabeled, so the project needs a logical structure to describe how to compare and classify the detected characters. We introduce a class set, as shown in Figure 4.7. The class set mainly contains all the classified characters, where different classified characters are stored under a different folder. Every character has a class. It contains a few images of the person. It looks like a different representation approach of training data in deep learning, but it is different from training data in many ways. Each character in the class set only needs a small number of images, and they are not used to train the model. In the comparison task, they are the reference for comparison. Comparing the unknown image with the images in the class set to get the known faces(face in the class set) with the highest similarity to the unknown image.

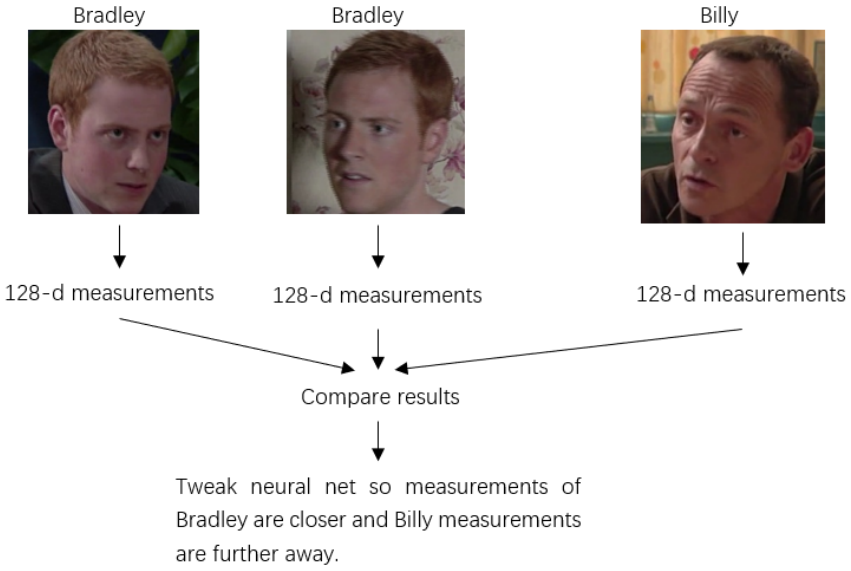

Figure 4.6: The triplet consists of 3 unique face images, two are same person, and third is different.the comparing results tweaks the network, so the same person will have a closer encoded value, and third person further apart

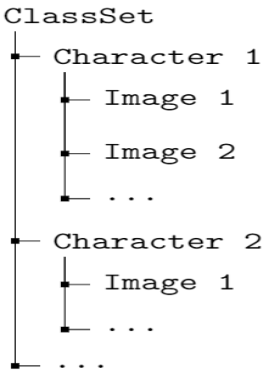

Figure 4.7: Tree view of Class set

4.2.5 Comparing Method

This step is the most straightforward in the entire process. All we have to do is finding the person in the data set that is closest to the 128-d measurement of our test image. In our case, we calculate the Euclidean distance between the 128-d embedding and all face vectors in our data set. If the range is below some threshold, then indicating the faces math. Otherwise, the faces do not match. Furthermore, there existing a more robust comparing method for further investigation, such as K-NN clustering.

4.2.6 Structure Conclusion

We will convert each frame of the video into an image and then process for each image. Each image will be processed by the method described above. Figure 4.8 summarizes this process very well. First, the face detection technology will detect the face in the image and then record the location information related to the bounding box of the located face. Using the location information of the face of the bounding box, use the face extraction method to obtain comparable face information, and compare the information of the face with faces in our class set using the Euclidean Distance method. The detected face will be assigned to the class with the highest similarity scores.

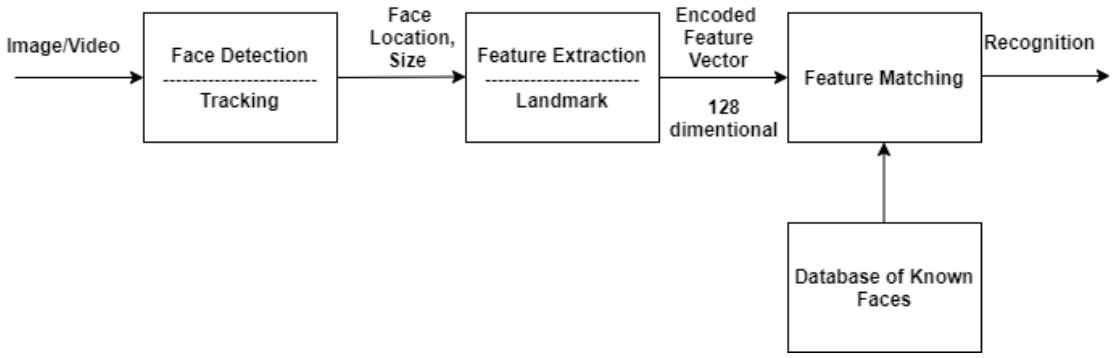

Figure 4.8: Face recognition processing flow

4.3 Image Pre-processing

The devil of face recognition is the noise, especially for our project. The TV drama we used as the data set was first released in 1967. Limited to the photography conditions at the time, the images of the drama were not clear, which constraints the feasibility of face detection. To accounts this, we tried three types of demonizing algorithms. The three algorithms are introduced in Chapter 3.3. Figure 4.9 shows the effect of three different pre-processing techniques on finding the bounding box of a face, which play the most important role in the performance of our model. If we look at the details of the face shown on 4.10, Gaussian filter removes noise from an original face image and produce good quality face images. In the implementation, we first applied these denoising techniques to a small sample set to evaluate better techniques, then applied to the full data set.

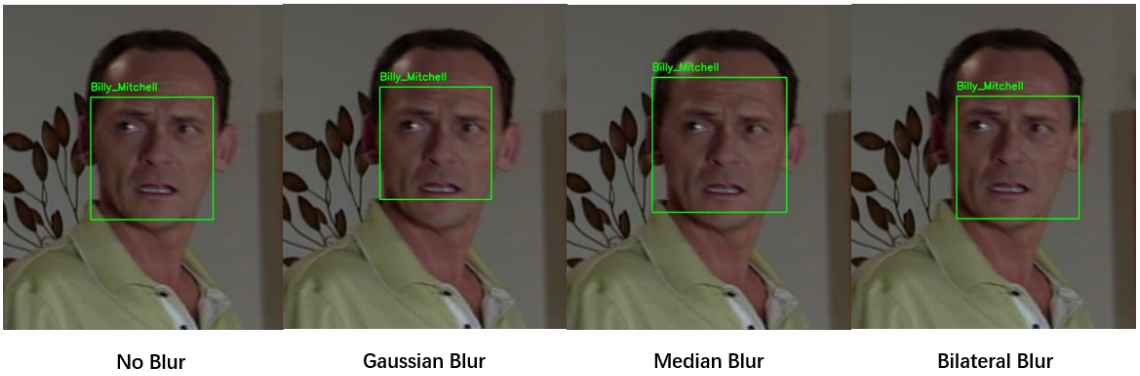

Figure 4.9: the different bounding box of same face, after applying pre-processing techniques

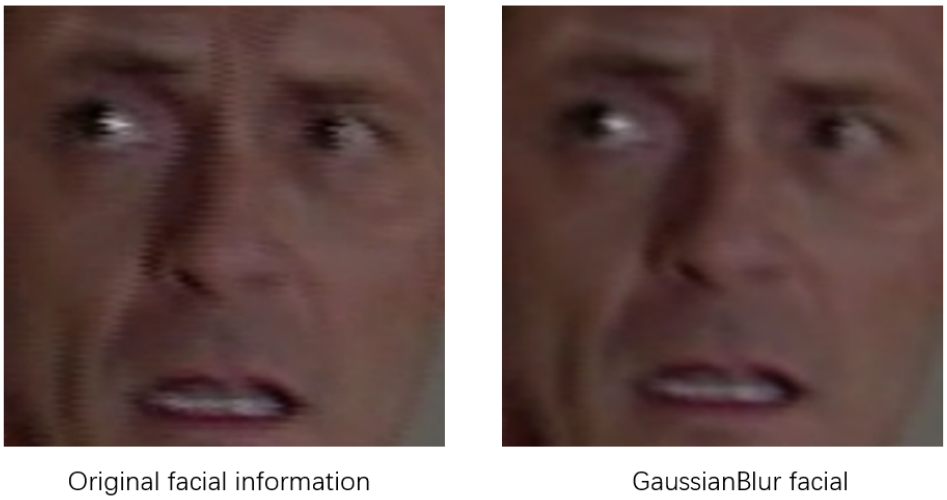

Figure 4.10: The face detail of original image of pre-processed image.

4.4 Video-based Model Hypothesis

In the above sections, All we used for calculation and processing are images, and our final goal is to apply our solution to TV drama videos. After understanding our core technology, we can briefly explain how to solve the problem in the video-based recognition task. Figure 4.8 summarizes this process pretty well. It is very intuitive and easy to understand the processing method. First, convert the video into images, then insert each image into our model in sequence. After obtaining and storing the results, we can process the next frame. We believe that there exists a completely automatic structure. Figure 4.11 shows how this structure works. We introduced the class set in the above section. The class set mainly contains all the classified characters. The set is empty at the beginning. We assume that when a frame as input is imported to the model and a face is detected if the class set is empty, the detected face will be added to the class set as a new class and represented as

person_1. Can deduce the rest from this, the input video is divided into pictures of individual frames, and then faces of characters will be detected one by one. For another situation, we assume that there are person_1, person_2 ... person_n in the class set. If a frame as input is imported and one or more faces are detected, these faces will be successively compared with the classified faces in the class set. By finding the class with the highest score, the detected face will be considered to belong to this class, and the face picture will be added to the person class. If the similarity scores of all person classes are deficient, then it is considered that the detected face has not appeared before, the class set will add a person_n+1 and then put the detected face into it.

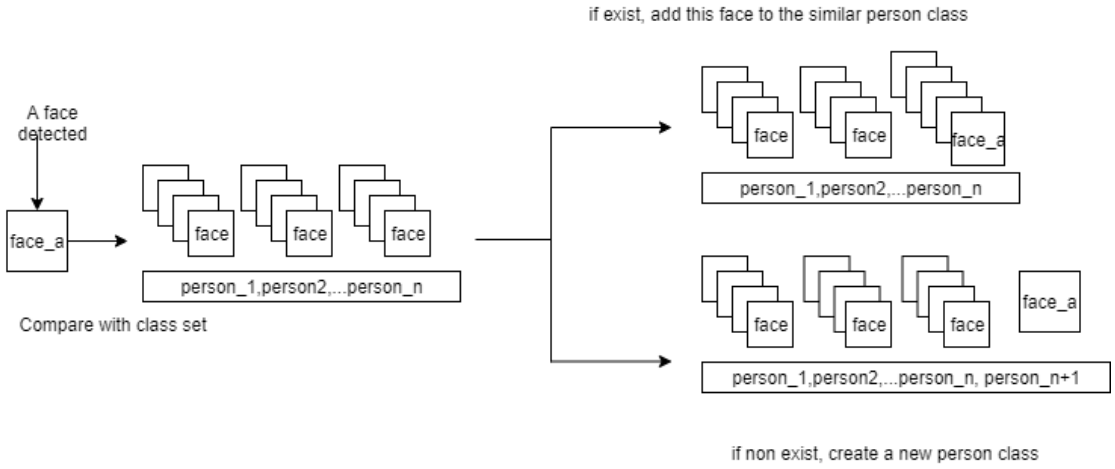

Figure 4.11: Comparison structure

4.5 Evaluation for Detection and Comparison

Among the algorithms involved in this project, the corresponding evaluation methods are roughly divided into two types. One is used to evaluate the model performance of detection, and the other is used to assess the model performance of classification. When defining the test performance method, we encountered many problems. Because this project pays more attention to the performance of the comparison part of the model, which narrows the scope of the test we need to implement. We decided to use a classification method to evaluate our model, which reduced our work a lot. In this project, many outliers are not covered by the general classification project. For example, the model can detect the face but cannot distinguish who it is, and the unknown label is inserted to that face. Or there are people in some pictures, but the model fails to detect their faces. A series of parameters defined by the classification method in many methods can solve the problems we face in the research process, and the performance of the model can be calculated by calculating the required parameters.

Before introducing some definitions required by evaluation, there is a question that needs to be explained first. Although the recognition technology is used in the model to locate the face and calculate the corresponding bounding box, we also counted the number of faces that were not recognized during the experiment. However, in the recognition and classification evaluation methods, true positive(TP), false positive(FP), false negative(FN) and true negative(TN) have different definitions under different inspection standards. We cannot combine the two metrics into one; moreover, as we mentioned earlier, due to the time limitation, we cannot conduct a comprehensive test of the two types of algorithms we used. According to the focus of our work, we will focus on testing model comparison, which is the model performance of classification. Because the data of unrecognizable faces has nothing to do with comparison performance, we will ignore the impact of this data on model performance in the following statistics.

- Recall: ability of a classification model to identify all relevant instances
- Precision: ability of a classification model to return only relevant instances
- F1 score: single metric that combines recall and precision using the harmonic mean

Chapter 5

Results and Discussion

5.1 Results

We present our experimental results in the tables below 5.1 - 5.8. we also briefly explanation of the meaning of some terms in the tables and experimental method we used. In the database of this experiment, we took a total of 4 characters images, and each character has 100 images, a total of 400 images. In these images, only one character is shown in each image. For the quality of the images, we did not choose any images with higher quality or easy to handle on purpose, so many people in the images have only a small part of their faces. In some images, characters are facing away from the camera, and they do not contain any facial information. In some cases, the outline of the character is extremely blurry in part of the images. this issue occurs when the camera moves at high speed and due to the bad quality of the video.

We also looked for some images without lighting conditions, these images are often difficult to distinguish the outline of the character, and some details are lost in the facial information. By adding these poor quality images, we can better evaluate the performance of our algorithm under different conditions. Taking Table 5.1 as an example, it recorded the results of character Bradley Branning in 400 images predicted by the algorithm.

All results are recorded in the form of Accuracy, Precision, Recall and F1-Score. There is a prefix of 10 or 20 on the left of the table. The prefix of 10 means that each character has ten labelled images put into the class set, which means that there are a total of 40 images in the class set. And 20 means that every character in the class set has 20 labelled images, a total of 80 images as a reference for the comparison task. The HOG and R-CNN behind the numbers mean that the model uses the HOG or R-CNN approach for the face detection task of the model to obtain the bounding box. The nopre, Gaussian, etc. represent the images pre-processing method used. The remaining unmarked technologies indicate that the model uses the same method. (such as face extraction) This table only records part of the methods we implemented. Because it takes days to run the model and all tests, there are also some methods we did not have sufficient time to test. However, we did some tests partially and

obtained some results. We will describe this part in detail later.

Bradley Branning	Accuracy	Precision	Recall	F1-score
10-hog-nopre	0.959	0.986	0.973	0.979
10-hog-bilateral	0.947	0.96	0.986	0.973
10-hog-median	0.986	0.986	0.999	0.993
10-hog-gaussian	0.973	0.986	0.986	0.986
20-hog-nopre	0.947	0.96	0.986	0.973
20-hog-bilateral	0.935	0.945	0.993	0.969
20-hog-median	0.973	0.962	0.999	0.987
20-hog-gaussian	0.962	0.96	0.995	0.981

Table 5.1: Bradley Branning with using 10 or 20 samples and HOG for each character in class set

Bradley Branning	Accuracy	Precision	Recall	F1-score
10-rcnn-nopre	0.875	0.919	0.948	0.933
10-rcnn-bilateral	0.838	0.907	0.917	0.912
10-rcnn-median	0.899	0.967	0.927	0.947
10-rcnn-gaussian	0.841	0.891	0.938	0.914
20-rcnn-nopre	0.90	0.957	0.938	0.947
20-rcnn-bilateral	0.856	0.948	0.904	0.926
20-rcnn-median	0.953	0.987	0.913	0.959
20-rcnn-gaussian	0.865	0.944	0.925	0.932

Table 5.2: Bradley Branning with using 10 or 20 samples and RCNN for each character in class set

Billy Mitchell	Accuracy	Precision	Recall	F1-score
10-hog-nopre	0.951	0.999	0.951	0.975
10-hog-bilateral	0.935	0.983	0.951	0.967
10-hog-median	0.918	0.999	0.918	0.957
10-hog-gaussian	0.951	0.999	0.951	0.975
20-hog-nopre	0.943	0.978	0.963	0.968
20-hog-bilateral	0.924	0.965	0.965	0.964
20-hog-median	0.907	0.982	0.931	0.952
20-hog-gaussian	0.938	0.978	0.974	0.976

Table 5.3: Billy Mitchell with using 10 or 20 samples and HOG for each character in class set

Billy Mitchell	Accuracy	Precision	Recall	F1-score
10-rcnn-nopre	0.909	0.999	0.909	0.952
10-rcnn-bilateral	0.850	0.988	0.859	0.919
10-rcnn-median	0.902	0.968	0.929	0.948
10-rcnn-gaussian	0.860	0.989	0.869	0.925
20-rcnn-nopre	0.921	0.999	0.895	0.954
20-rcnn-bilateral	0.867	0.996	0.848	0.924
20-rcnn-median	0.923	0.978	0.916	0.945
20-rcnn-gaussian	0.882	0.998	0.848	0.927

Table 5.4: Billy Mitchell with using 10 or 20 samples and RCNN for each character in class set

Dot Cotton	Accuracy	Precision	Recall	F1-score
10-hog-nopre	0.973	0.999	0.973	0.986
10-hog-bilateral	0.986	0.999	0.986	0.993
10-hog-median	0.986	0.999	0.986	0.993
10-hog-gaussian	0.951	0.999	0.951	0.975
20-hog-nopre	0.961	0.976	0.985	0.991
20-hog-bilateral	0.976	0.976	0.997	0.987
20-hog-median	0.976	0.976	0.997	0.987
20-hog-gaussian	0.939	0.976	0.966	0.981

Table 5.5: Dot Cotton with using 10 or 20 samples and HOG for each character in class set

Dot Cotton	Accuracy	Precision	Recall	F1-score
10-rcnn-nopre	0.883	0.891	0.990	0.938
10-rcnn-bilateral	0.824	0.831	0.990	0.904
10-rcnn-median	0.831	0.838	0.990	0.908
10-rcnn-gaussain	0.836	0.851	0.981	0.911
20-rcnn-nopre	0.896	0.924	0.979	0.951
20-rcnn-bilateral	0.837	0.856	0.981	0.914
20-rcnn-median	0.845	0.861	0.978	0.921
20-rcnn-gasussian	0.852	0.879	0.968	0.923

Table 5.6: Dot Cotton with using 10 or 20 samples and RCNN for each character in class set

5.2 Discussion

By comparing the data in the tables, we found that using ten labelled data as reference, HOG as face detection approach without any pre-process techniques obtained the best performance. In our expectation, the best performance should be obtained by using a pre-trained R-CNN deep learning model combined with some image pre-processing techniques. The results from experiments seem to mean that all the improvement methods we have tried do not affect at all. However, when we analyzed our experimental data in-depth and looked at some of the

Stacey Slater	Accuracy	Precision	Recall	F1-score
10-hog-nopre	0.938	0.999	0.938	0.968
10-hog-bilateral	0.938	0.999	0.938	0.968
10-hog-median	0.938	0.999	0.938	0.968
10-hog-gaussain	0.926	0.999	0.926	0.962
20-hog-nopre	0.926	0.978	0.951	0.961
20-hog-bilateral	0.926	0.978	0.951	0.961
20-hog-median	0.926	0.978	0.954	0.966
20-hog-gaussian	0.914	0.978	0.937	0.951

Table 5.7: Stacey Slater with using 10 or 20 samples and HOG for each character in class set

Stacey Slater	Accuracy	Precision	Recall	F1-score
10-rcnn-nopre	0.881	0.999	0.938	0.968
10-rcnn-bilateral	0.938	0.999	0.938	0.968
10-rcnn-median	0.938	0.999	0.938	0.968
10-rcnn-gaussian	0.926	0.999	0.926	0.962
20-rcnn-nopre	0.893	0.999	0.927	0.973
20-rcnn-bilateral	0.949	0.999	0.927	0.973
20-rcnn-median	0.951	0.999	0.927	0.973
20-rcnn-gaussian	0.935	0.999	0.913	0.968

Table 5.8: Stacey Slater with using 10 or 20 samples and RCNN for each character in class set

results, we found the reason why the experimental data was different from our expectation.

5.2.1 Evaluate Pre-process Approaches

We investigated our Gaussian Blur approach in the beginning. During our research, we found that many reports show that Gaussian can handle the low-frequency noise and chaotic parts of the image very well. When we conducted the first stage experiment, many images with poor image quality were processed by Gaussian Blur, and the image quality was improved significantly. Figure 5.1 shows an image with unobvious contours of the character's edges caused by the high-speed movement of the camera. After Gaussian processing, the quality of the character image has been significantly improved. Therefore, when our experimental results were different from our expectations, we decided to conduct an additional experiment to evaluate the performance of the Gaussian Blur approach.

We used Gaussian for image processing and re-tested the failed photos in 10-HOG-no pre. In the 10-HOG-no pre, there are nine images of people who could not be successfully identified. After Gaussian processing, characters in 3 images can be identified correctly by the model.

We conducted the same test in 10-RCNN-no pre, and 4 of the 13 images that could not successfully identify the characters obtained the correct results after using the Gaussian Blur. Such experimental results show that Gaussian can solve some problems that cannot be solved in images without any pre-processing. Through research, we found that images without any pre-process can also solve some issues that Gaussian processed images cannot address. For example, image X can recognize the correct face by our algorithm that does not perform any pre-process processing. But after image X processed by Gaussian, the algorithm cannot find the face correctly. The number of images where Gaussian cannot find the correct face is higher than the number of additional correct faces, which leads to the Gaussian approach having a worse performance than no pre-process approach.

Figure 5.1: Original image and image with using Gaussian

Gaussian is a representative example. Other pre-process methods can also recognize some faces that could not be recognized by the original approach, but we have no time to test them and record them. We need to point out that we also found a method that can be used to deal with the problem about the loss of facial details caused by different illumination conditions. Figure 5.2 shows that by using the method called Contrast limited adaptive histogram equalization, the originally unclear details becomes clearer. Due to time limitation and the running speed of pre-processing constraints, we do not have enough time to run the algorithm and get complete test results. Therefore, we only tested contrast processing on the pictures in 10-HOG-no pre and 10-RCNN-no pre. In 10-HOG-no pre, contrast equalization can identify 4 out of 9 images and 6 out of 13 in 10-RCNN-no pre.

In many images with poor lighting conditions, contrast equalization can still process these images well and allow the algorithm to make the correct prediction. For the bilateral and median approaches, they preserve edges of the character in the images, and this process can help the model to detect faces easier. After using bilateral to pre-process the images, in 10-HOG-no pre, our model can recognize two more correct faces from 9 unknown faces, and 2 out of 13 for 10-RCNN-no pre. With using Median as pre-process approach, we can get one more correct faces out of 9 unknown faces in 10-HOG-no pre, and 1 out of 13 for 10-RCNN-no pre.

Figure 5.2: Original image and image with using Contrast Limited Adaptive Histogram Equalization

5.2.2 Number of Reference Data

We found that there is no apparent difference between the algorithm with ten labelled images as the reference and the algorithm with 20 labelled images as the reference. We treat this as a good prediction result. Because when we designed this algorithm, we did not want to use too many labelled images. If the model requires too much-labelled data, this will significantly increase our work during the experiment. We want to design a model that does not require too many manual operations so that our algorithm can be applied to other face recognition tasks that without true labels. The performance of 10 labelled images and 20 labelled images is similar, which confirms that our algorithm does not require too many labelled images in the class set.

5.2.3 Evaluate Detection Approaches

There is a huge difference between the results of using HOG and the results using RCNN as the face detection approach, which is very different from our expected results. Because in our expectations, RCNN should be a better face detection algorithm than HOG, and finally we found the reason. We explained what undetected faces were in the previous evaluation section. In the process of face detection, faces in images detected by the RCNN or HOG algorithm failed are undetected faces. However, undetected faces take into account when we evaluate and calculate the performance of HOG and RCNN. RCNN can detect more faces, and that the bounding boxes found by RCNN are more accurate than those found by HOG. We have made additional tables 5.9 5.10. These tables record the number of correct faces found by HOG and RCNN algorithms for each character. They also record the number of undetected faces and unknown faces predicted by HOG and RCNN. RCNN has less undetected faces than HOG. It is a good proof that as a detection approach, RCNN has a better performance than HOG. RCNN can detect more faces and their bounding boxes, but our comparison

approach can't find the correct labels of extra detected faces very well, which leads to the overall performance of RCNN being worse than HOG.

Bradley Branning	PC	PI	UKF	UDF
10-hog-nopre	71	0	2	27
10-hog-bilateral	72	1	0	27
10-hog-median	73	0	0	27
10-hog-gaussian	72	0	1	27
20-hog-nopre	72	0	1	27
20-hog-bilateral	72	1	0	27
20-hog-median	73	0	0	27
20-hog-gaussian	73	0	0	27

Table 5.9: Tabels for section 5.2, PC = predicted correct faces, PI = predicted incorrect faces, UKF = unknown faces, UDF = undetected faces, presents number of faces counted in HOG condition

Bradley Branning	PC	PI	UKF	UDF
10-rcnn-nopre	91	5	0	4
10-rcnn-bilateral	88	8	0	4
10-rcnn-median	89	6	1	4
10-rcnn-gaussian	90	6	0	4
20-rcnn-nopre	90	6	0	4
20-rcnn-bilateral	89	7	0	4
20-rcnn-median	88	8	0	4
20-rcnn-gaussian	90	6	0	4

Table 5.10: Tabels for section 5.2, PC = predicted correct faces, PI = predicted incorrect faces, UKF = unknown faces, UDF = undetected faces, presents number of faces counted in RCNN condition

5.3 Video Results

The above experimental results can only indicate that our model can theoretically handle video after converting video into images. We have not tested the feasibility of our model in any video. However, due to the time limitation, we had to simplify our video testing process. We cropped two videos within 10 seconds, and only one character appeared in each video, and they were all easy to recognize. According to the model we described in the previous section, we will convert each frame of the video into an image and then process each image as input. Our two video clips were converted into about 300 images and subsequently processed them with our model. The following tables 5.11 5.12 show the results of our model with ten labelled samples using HOG and RCNN as the detection methods, and no preprocess and Gaussian Blur respectively as preprocessing approaches. Because the video we captured was concise,

and only one character appeared in front of the camera, the comparison performed pretty well according to the data in the table. In these single scenario tests, we found that both RCNN and Gaussian got a better performance. RCNN has a better performance than HOG is because, in our video scene, the number of faces that HOG and RCNN can detect is almost the same, but the bounding boxes predicted by RCNN are more accurate, which makes the entire model have a better performance. This result also confirms our previous analysis of RCNN is correct. Gaussian also has better performance in video testing. This is because the images cropped from the video are continuous, and there will be a lot of images captured where the camera moves at high speed, and the contours of the characters are blurred in these images. Gaussian can handle the problem of character blur, so it has a better performance. It also confirms that our previous analysis of Gaussian is correct. Although we got better results in the video test, we still have to point out that our experiment can only prove that our model can probably be used in the video set. It does not mean that our model can have outstanding performance in the video set. Because our video scene is straightforward, the characters are accessible for the model to detect, compare and predict the results. We used more complex and variable scenes in the imaging experiment, which led to the unsatisfactory results of our previous tests. So we can't guarantee that our model is still stable in a composite video with no bugs or have the same performance.

Video 1	Accuracy	Precision	Recall	F1-score
10-hog-nopre	0.903	0.951	0.947	0.949
10-hog-gaussian	0.916	0.992	0.923	0.956
10-rcnn-nopre	0.930	0.951	0.977	0.964
10-rcnn-gaussian	0.985	1	0.985	0.992

Table 5.11: Video 1 performance for section 5.3

Video 2	Accuracy	Precision	Recall	F1-score
10-hog-nopre	0.885	1	0.885	0.939
10-hog-gaussian	0.945	1	0.945	0.972
10-rcnn-nopre	0.896	1	0.896	0.945
10-rcnn-gaussian	1	1	1	1

Table 5.12: Video 2 performance for section 5.3

5.4 Future Work

We still have some unfinished tests. For example, in the above section, we emphasized that our model was only evaluated in two simple video sets. The performance and stability of our model in more complex scenes or real TV dramas dataset have not been tested sufficiently. We also have some methods that have been planned or researched that can improve the performance of our model, but we have not yet started to implement them. In the detection

stage, we proposed two methods, HOG and RCNN, and RCNN has better performance. There are more and more deep learning structures with better performance, such as YOLO, faster RCNN, etc. for face detection. It is difficult for us to apply these approaches in a short period, so we abandon it on this project.

For the comparison task, there are many facial extraction methods that we have not had time to give it a try. Because they are more complex and have more requirements for implementation, we have failed to reproduce these methods in a short period, so we also gave up on implementing them. For image preprocessing, although we have confirmed in experiments that they have excellent performance in dealing with specific poor quality images, we still failed to summarize and combine their advantages. We were unable to make a method that combines all the benefits of approaches and automatically gets the best performance in a single model. Similarly, the automatic system mentioned in the above section has not been implemented. We thought these were very interesting. If all these improvements could be achieved, then our model performance could be significantly improved, and it could be easier to use in different datasets.

We currently only focus on face detection. However, there still possibility combine other technique to improve on this project. Different appearance of characters, such as cloth colour, body posture. Moreover, voice wave recognition probably will be an exciting research direction on recognizing characters on a TV drama.

Chapter 6

Conclusion

The main research of this project is how to track people in TV drama. By studying the results of the existing surveillance view, we found that there are some differences between TV drama and surveillance view videos. Because the appearance of characters often changes, and the camera view also changes frequently, many traditional surveillance approaches cannot be applied to our dataset. This led us to finally decide to use facial information as the main direction. In the dataset of this project there are no true labels, which means that we cannot use deep learning to train a model to solve the problem in this project.

By analyzing the requirements of this project, the tracking task is roughly divided into two parts: detection and comparison. We used HOG and a pre-trained RCNN as detection approaches and Euclidean distance in comparison as a comparison method. We also added some pre-process approaches to optimize some poor quality images. In the initial image tests, we failed to get the same test results as we expected. After analysis, we found that due to the complexity of some test images, some improved methods did not get the same results as expected. But after a careful investigation, we can determine that RCNN can get more accurate and a larger number of bounding boxes of faces than what HOG predicts. We also found that Gaussian Blur, Contrast equalization and other methods can also solve the problem of image blur due to the high-speed movement of the camera, and the loss of details of the face caused by poor lighting conditions, etc.

After making a simple video test, we can confirm that the above conclusions are correct. However, the time limitation caused many of the tasks we expected and planned to fail to implement. We need to emphasize that our model can theoretically track faces in the video after a simple video test. But we haven't been able to test complex video or an episode of TV dramas, so the stability and performance of our model in these videos cannot be guaranteed. We also found some better detection and comparison methods, but also due to time limitations, it is difficult for us to implement these in a short time. But we believe that if these improvement approaches can be implemented in the future, our model can get better results.

Bibliography

[1] Alex. (2018) Boston man confesses to killing pedestrain and fleeing scene during tv interview, gets arrested. [Online]. Available: https://grabien.com/file.php?id=400105

[2] (1967) Star trek (the original series) season 1, episode 28, "the city on the edge of forever". [Online]. Available: https://www.flickr.com/photos/29069717@N02/25767008757/in/photostream/

[3] (2013) Trec video retrieval evaluation: Trecvid. [Online]. Available: https://trecvid.nist.gov/

[4] A. Geitgey, "Machine learning is fun! part 4: Modern face recognition with deep learning," Jul 2016. [Online]. Available: https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

[5] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," 06 2014.

[6] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.

[7] E. COMMISSION, "Regulation of the european parliament and of the council; establishing the european border surveillance system (eurosur)," 2011.

[8] (2005) Brisbane airport: The missing cctv footage. [Online]. Available: https://www.expendable.tv/2012/08/brisbane-airport-missing-cctv-footage.html

[9] (2013) Guidelines for trecvid 2013. [Online]. Available: https://www-nlpir.nist.gov/projects/tv2013/index.html#data

[10] (2020) Eastenders. [Online]. Available: https://www.bbc.co.uk/programmes/b006m86d

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *NIPS*, 2012.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, p. 91–110, 2004.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05)*.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[15] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik, "Recognition using regions," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[16] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[17] R. Girshick, "Fast r-cnn," *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *NIPS*, pp. 91–99, 2015.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[20] Zhang, Ren, Sun, and Jian, "Deep residual learning for image recognition," Dec 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[21] V. Fung, "An overview of resnet and its variants," Jul 2017. [Online]. Available: https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035

[22] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[23] H. Jiang and E. Learned-Miller, "Face detection with the faster r-cnn," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017, pp. 650–657.

[24] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5525–5533.

[25] X. Sun, P. Wu, and S. C. Hoi, "Face detection using deep learning: An improved faster rcnn approach," *Neurocomputing*, vol. 299, pp. 42–50, 2018.

[26] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009, 2010.

[27] O. Cakiroglu, C. Ozer, and B. Gunsel, "Design of a deep face detector by mask r-cnn," in *2019 27th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2019, pp. 1–4.

[28] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," 06 2013, pp. 532–539.

[29] C. X. W. Y. e. a. Lou, J., "Multi-subspace supervised descent method for robust face alignment," *Multimedia Tools and Applications*, vol. 78, December 2019.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017. [Online]. Available: http://doi.acm.org/10.1145/3065386

[31] M. Wang and W. Deng., "Deep face recognition: A survey." https://arxiv.org/abs/1804.06655, April 2018.

[32] T. B. Z. W. C. H. Jingtuo Liu, Yafeng Deng, "Targeting ultimate accuracy: Face recognition via deep embedding," https://arxiv.org/abs/1506.07310, June 2015.

[33] R. C. Rajeev Ranjan, Carlos D. Castillo, "L2-constrained softmax loss for discriminative face verification," https://arxiv.org/abs/1703.09507, March 2017.

[34] N. X. S. Z. Jiankang Deng, Jia Guo, "Arcface: Additive angular margin loss for deep face recognition," https://arxiv.org/abs/1703.09507, Jenuary 2018.

[35] E. Tsomko, H. Kim, and E. Izquierdo, "Linear gaussian blur evolution for detection of blurry images," *IET Image Processing*, vol. 4, no. 4, p. 302, 2010.

[36] N. R. Goodman, "Statistical analysis based on a certain multivariate complex gaussian distribution (an introduction)," *The Annals of Mathematical Statistics*, vol. 34, no. 1, p. 152–177, 1963.

[37] R. Haddad and A. Akansu, "A class of fast gaussian binomial filters for speech and image processing," *IEEE Transactions on Signal Processing*, vol. 39, no. 3, p. 723–727, 1991.

[38] M. S. Nixon and A. S. Aguado, "Basic image processing operations," *Feature Extraction and Image Processing*, p. 67–97, 2002.

[39] M. Bottema, "Deterministic properties of analog median filters," *IEEE Transactions on Information Theory*, vol. 37, no. 6, p. 1629–1640, 1991.

[40] E. Arias-Castro and D. L. Donoho, "Does median filtering truly preserve edges better than linear filtering?" *The Annals of Statistics*, vol. 37, no. 3, p. 1172–1206, 2009.

[41] G. R. Arce, "Nonlinear signal processing," 2004.

[42] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*.

[43] P. S. Foundation, "Python," https://www.python.org/, December 2019, access on 19/12/2019.

[44] Itseez, "Open source computer vision library," https://github.com/itseez/opencv, 2015.

[45] F. Chollet, "keras," https://github.com/fchollet/keras, 2015.

[46] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.