Object Detection and Recognition

---- Realtime facial component detection and smile recognition

**Group 11: LIU Jinyu / LING Liyang / ZHANG Xichen / GAO Han**

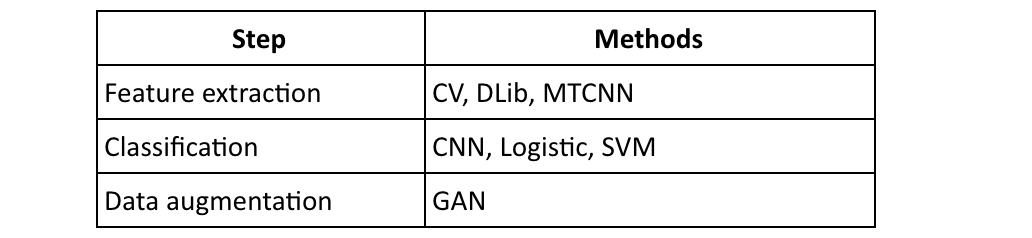
# Introduction

Nowadays, deep learning is widely used in objective detection and recognition. To be more specific, this project selected to practice facial component detection and smile recognition with available training capacity based on a series of machine learning methods. The final result is finding smiles in Real-time camera. The follows will introduce how features are extracted from face images and the classification of smile face and not smile face.

**Data set**

The Data set is from GENKI 4K [1], containing 4000 face images with label.

**Methods**

****

We tried to get involved the whole process from the detection of face, localize important features from face, until smile classification using machine learning methods. Such as Logistic classification, CNN, SVM and GAN.

**Preprocessing**

To achieve finding smiles, we conducted two approaches with different training set.

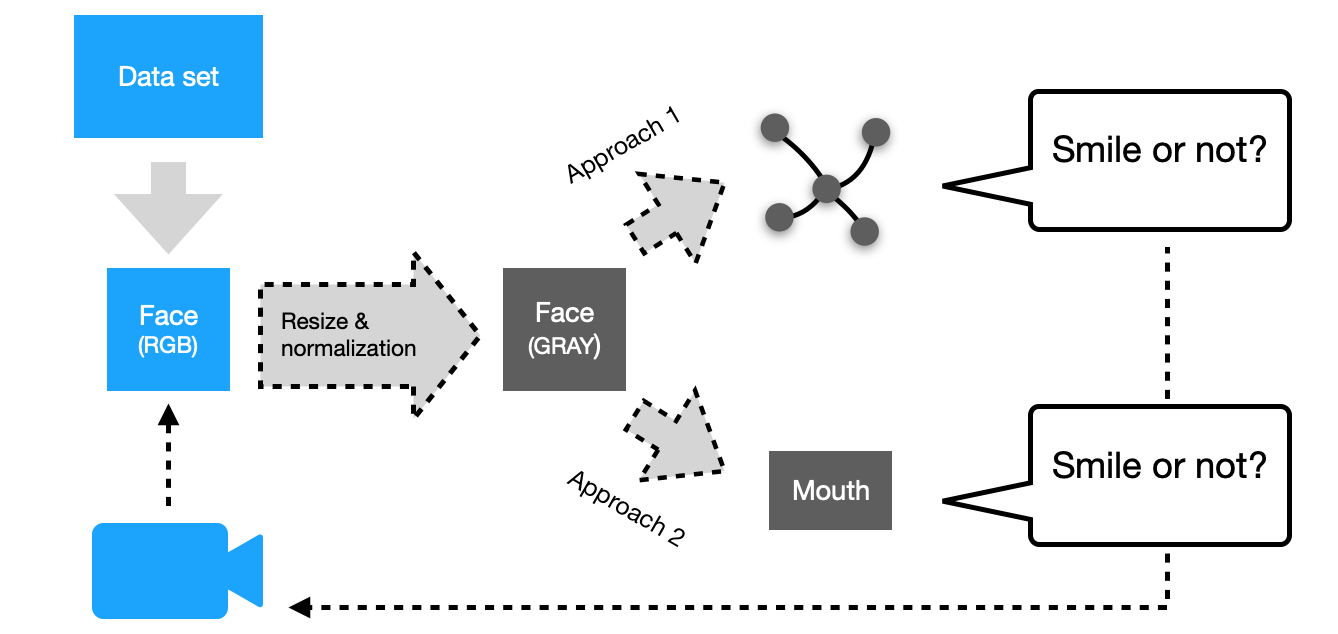
****

Figure 1

In the very beginning, we localized face from original image by OpenCV [2], then resized the face image after grayscale and normalized the pixel value.

In the next stage, the first approach was going to extract some facial landmarks from face, such as 5 points and 68 points [3]. Then use the points’ coordination as training set.

The second solution try to cut the mouth directly from face as mouth contains the most notable features in identifying smile face, such as the teeth and mouth shape. Then used the mouth data as training set.

# Implementation

In this chapter, we will explain the two approaches respectively.

## Facial Landmark Based Implementation

Basically, it’s known that the characters on people's face can be signed with several facial landmarks, the first plan aims at finding out all those character points and based on them to construct their connection with smile.

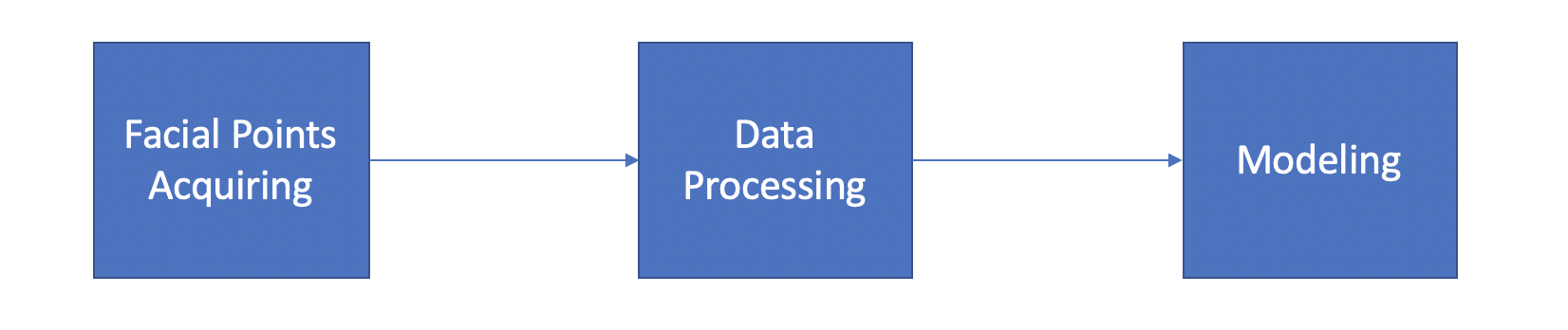


Figure 2

Over all, there are three main steps. the first is to Acquire the facial points we needed, the next to do data processing to extract useful information and try to find out the best suitable model for making classification.

### Facial landmarks acquiring

Figure 3

Because it's too hard for us to start with acquiring facial points by ourselves, for the first step, we ask for help from a third-party package Dlib [4]. Through its pre-trained model, we can get the coordinates of 68 facial landmarks directly from the face images.

Here some pictures have been tagged as examples. it's very clear that the character points described eyebrows, eyes, nose, mouth and face contour. The tagger seems have pretty good performance.

### Data Processing

Figure 4

The next step is on how to use these data.

At beginning we just flatten all coordinates from 2 to 1 dimension and got 136 values as features. but while using those features as the training dataset and applying trained model onto our smile recognition program, the final performance is not good enough and highly lagged. So some more processing is necessary.

In addition, we analyzed the correlation between facial points and smile. From the heat map, coordinates of mouth show the strong contribution to smile prediction. So, we can reduce points from 68 to 32 excluding nose and face contour points.

What’s more, to make the features have the same standard, coordinates values are normalized with the size of face image and form our final training dataset.

Here gives expected result.

Through proper modelling, we can use the points of eyes and mouth to make smile classification.

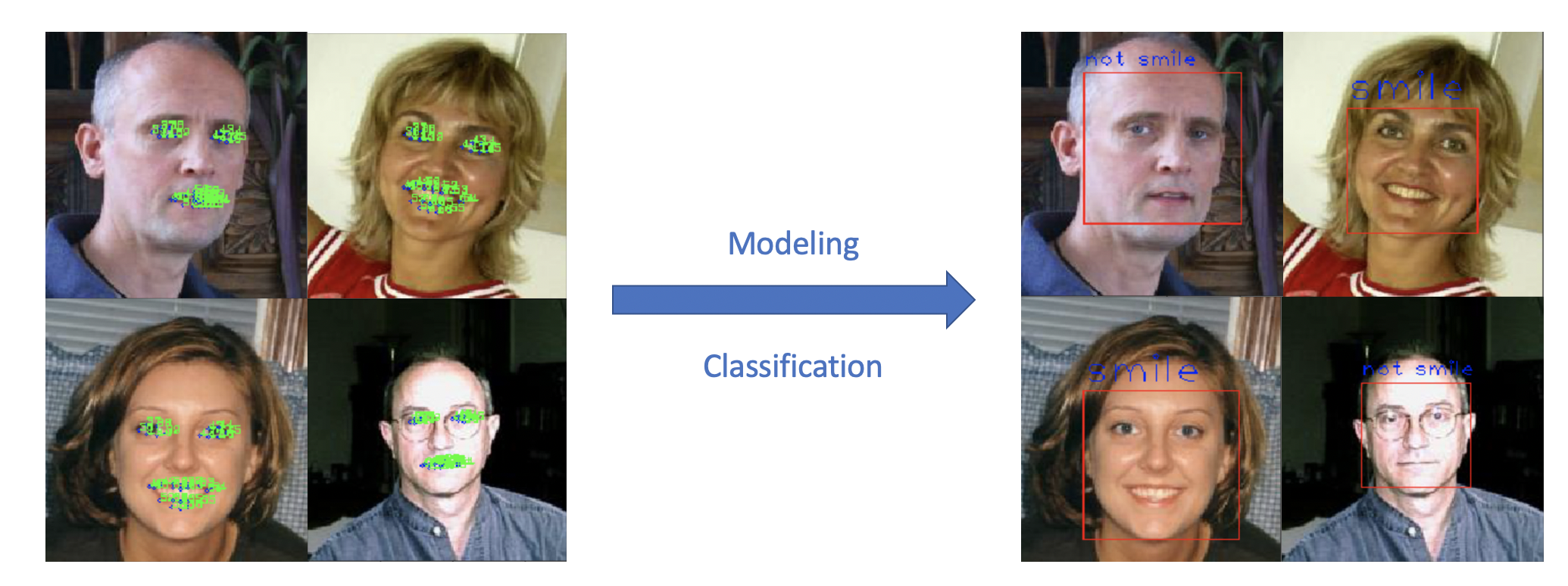
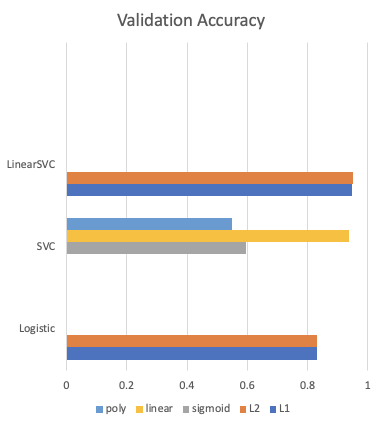


Figure 5

### Modeling

Basically, all these training data are values of coordinates which are separated from facial points, and the ratio of sample number and feature number is not big enough, based on them, initially linear related models are chose to make classification.

For logistic regression, it’s easy to understand and apply, we want to use it as baseline. For SVM, we want to exert its strength on dealing data with high sample-feature ratio and compare the performance on different kernel functions. After testing, the accuracy just proves our assumption, as long as linear kernel function is used, the model would have better accuracy. For the penalty strategy, L2 could be a bit superior to L1.

In conclusion, among these models, we pick up Linear SVC with L2 penalty as the model of approach A. But the problem of delay on smile recognition problem still remain to be solved, so here we come plan B.

Figure 6

## Mouth Based Implementation

The basic idea of this approach is based on an experience that when people smiling, the mouth is the most obvious indicator for the action. Hence, the process of this approach, as the graph shown, separated into 4 procedures: Mouth location, cutting pictures, passing through a standard normalization process, and finally modelling with multiple machine learning methods.

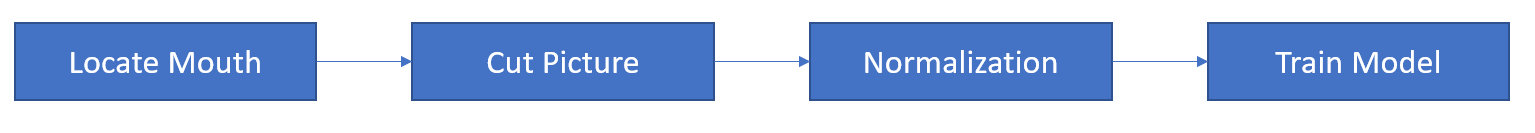


Figure 7

### Mouth Location

However, our dataset does not contain the tag for mouth positions. As a result, our priority is to figure out how to locate the mouth. In this approach we use MTCNN [6] to do the tagging, which can find the position of 5-facial landmarks from human face. Just as the pictures in the left shows, the eyes, nose, and two mouth corners are found in their exact positions. Although we can just use the pre-trained model directly, we still choose to train a CNN model ourselves to detect the mouth position.

In order to achieve the mouth location, we need to do more processing of the source images. First, we use pre-trained model MTCNN to tag the face images to get five points (which is a vector contains 10 values). And then, to make the training process faster, we gray-scaled images and reduce it size to a quarter of the original size. Finally, we normalize both input & output values respectively. After the procedures above, we finally get our training data. And ready to build a model.

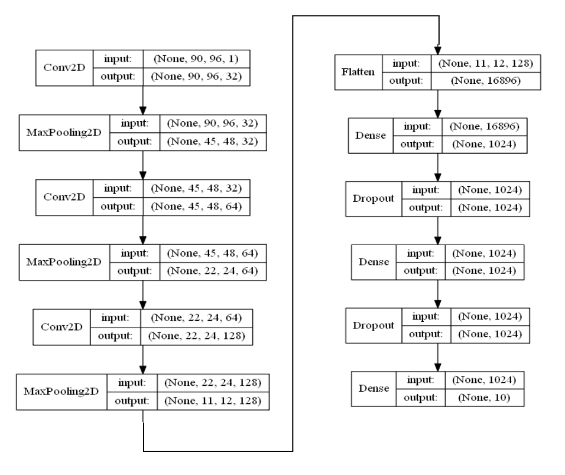


Figure 8 (1) & (2)

The model we construct is shown in the left. It contains 3 convolutional layers and 3 fully connected layers in order, with several max pooling layers, flatten layers in the middle. We use dropout in the dense layer to avoid overfitting. And all unit use ReLu as activation function.

**Mouth Location Training & Evaluation**

The loss in training dataset and validation dataset are shown below, as we can see after 60 epochs, the validation loss became stable. And the loss is almost same as the training loss which means the model learning effectively.

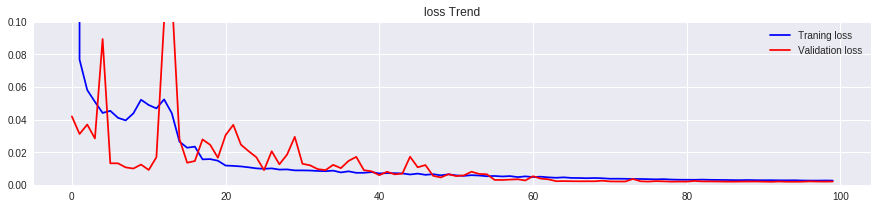


Figure 9

The images in the right are predicted values. As we can see the predicted position of 5 points are roughly right, even with different orientation of the faces. With the five points ready, we apply some simple formula based on No.2, 3, 4 points to calculate the position of mouth. Hereby we use right up corner location and height and weight to square a box of mouth area.

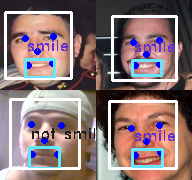
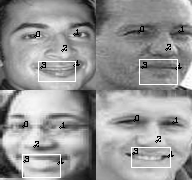


Figure 10 (1) & (2)

Now we can move on to the smile recognition part.

### Smile Recognition

In this part, we have applied multiple machine learning techniques to make prediction, such as logistic regression, CNN and SVM. The pictures in up right are the demonstrations, with three well classified smiley face and one neutral face. We will briefly introduce each method and make evaluation in the final.

**Logistic Regression & SVM / Linear SVM**

Logistic Regression is one of the most basic methods for binary classification. Traditionally, considering the high dimensionality, classify a picture is difficult for this model. But in this case, with a fine-processed feature, despite its simplicity, we believe this model could still have an acceptable performance and providing us a baseline for this project. While considering SVM is effective in high dimensional spaces and has a good performance in MNIST dataset. Therefore, we have applied both method in the project.

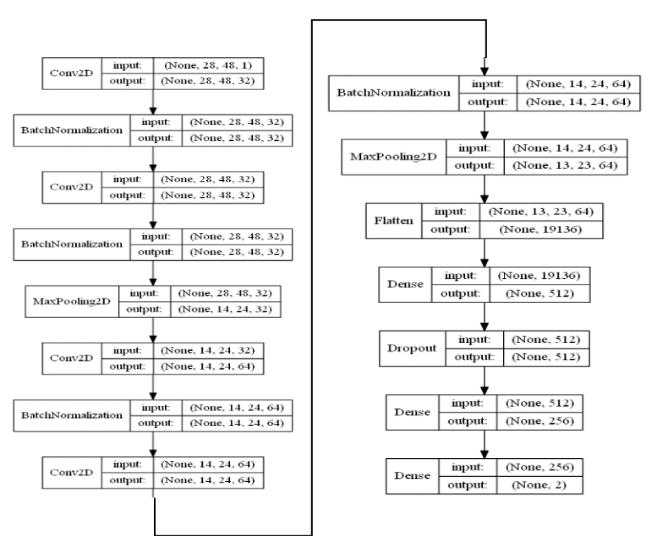


Figure 11

**CNN**

Different from the other two models, convolution neural network has been proven that have an outstanding performance in image processing area in practice. Thus, we hope it can hand over a satisfying result. The model structure is shown in the below, it contains 4 convolutional layers and 2 fully connected layers. Excepting the output layer using sigmoid as activation function, all others using ReLu.

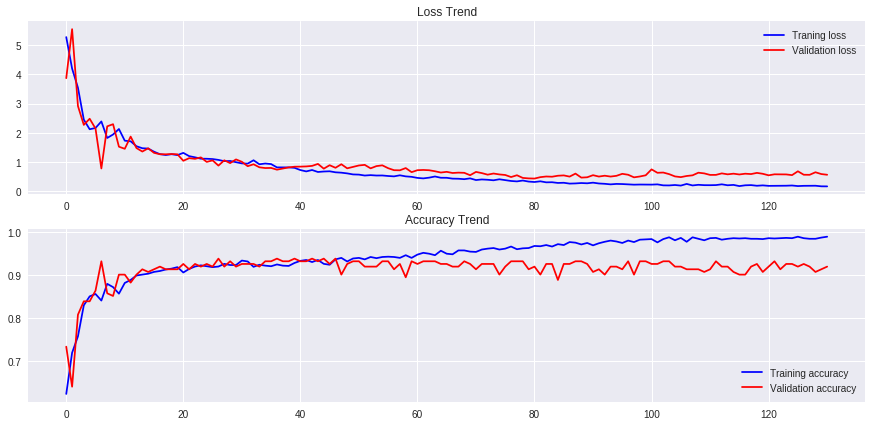
**Training**

Figure 12

The performance is shown here, as we totally trained 130 epochs, the final accuracy keeping fluctuating in the range around 90 to 94 percent. The validation accuracy reaches the top around 40 epochs, and start drop gradually with quite vary, this is a sign of slightly overfitting.

### Data Augmentation

**CGAN**

Generative adversarial networks (GANs) are models capable of mapping noise vectors into realistic samples from a data distribution.

In GAN, there is no control over modes of the data to be generated. The conditional GAN changes that by adding the label y as an additional parameter to the generator and hopes that the corresponding images are generated. These labels may give a significant head start to GAN for what to look for. Another possibility is that our visual system is biased and more sensitive to these labels. Hence, the generated images are perceived to be better.

The two models are as following. In each iteration, discriminator and generator are trained separately, and improve each other in the process. Based on the existing dataset, the CGAN [6] can generate more images for the training of classification.

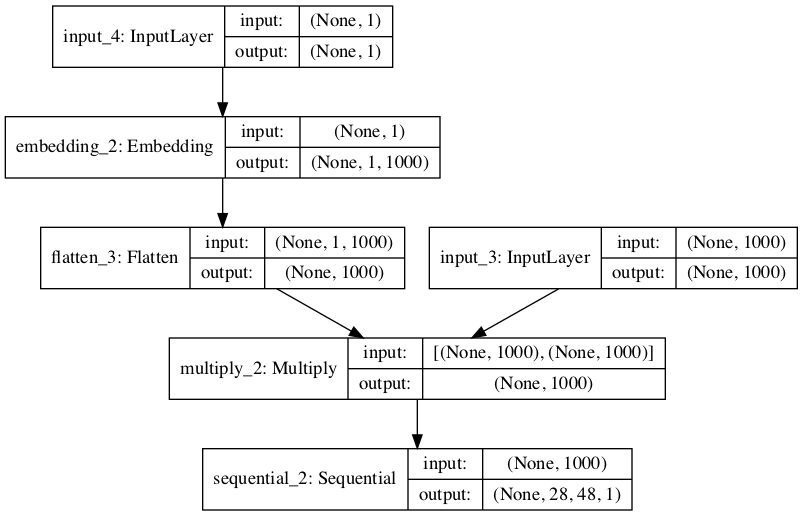
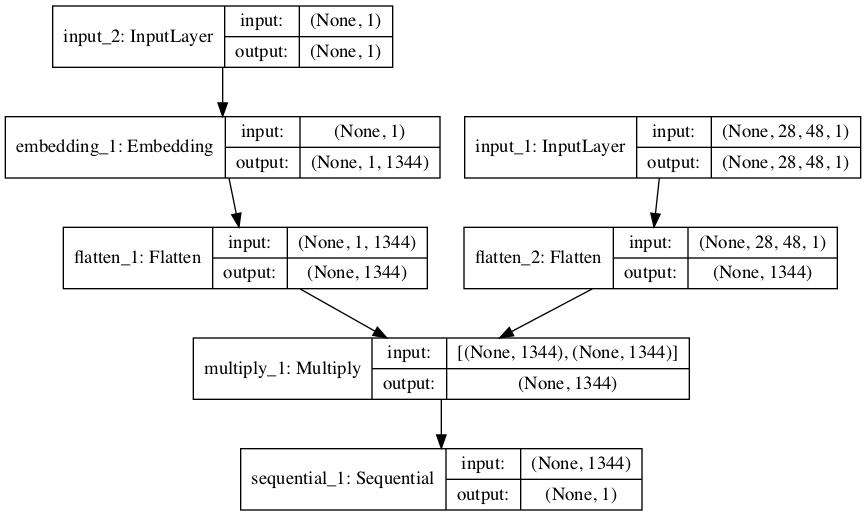


Figure 13. Generator(left) & Discriminator(right)

# Evaluation & Performance

Based on two plans of extracting features, this project is going to evaluate the performance of corresponding models. The first one is accuracy, which is a general criterion for examining performance of different classification methods. Secondly, in real-time application, efficiency will seriously influence user experience. High latency might not catch face motion on time. Then for the operation of system, stability is indispensable. With low stability, slight change might lead to significant change of prediction results.

## Accuracy

### Plan A

For plan A, the project extract 68 face landmarks as features to train models. Here are 6 corresponding models. They are linear regression with L1 regularization, linear regression with L2 regularization, SVM with sigmoid kernel function, SVM with linear kernel function, SVM with polynomial kernel function and linear method with SVM. The picture below shows that the combination with linear method and SVM can get highest accuracy. And from the picture of ROC curve in linear SVM model, its accuracy can meet requirement.

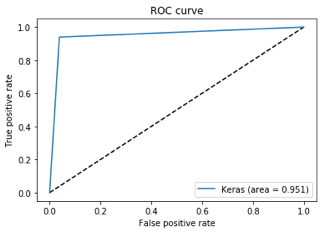
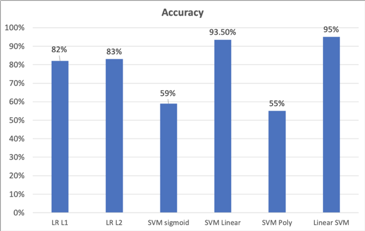


Figure 14

### Plan B

For plan B, the project uses MTCNN to extract mouth from faces on picture. So features just come from pixels in mouth area. Here three corresponding models are CNN, Linear SVM and Logistic regression. From the picture below obviously, we can see that linear SVM in plan A outperforms models in plan B. That is because plan A collect features not only in mouth but other parts of face, which can better detect smile without opening mouth. Below is the picture of ROC curve in CNN model. It has better shape than linear SVM.

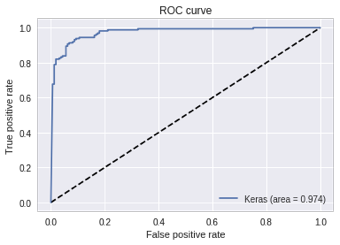
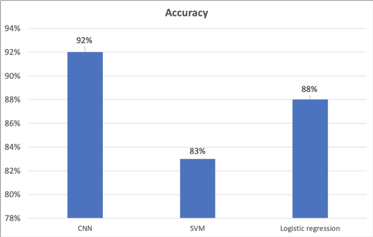


Figure 15

### Actual test

For the most normal case when a person smiles with frontal face, CNN and linear SVM from two plans can detect smile accurately. For a person having frontal face smiling with closed mouth, both models can also achieve.

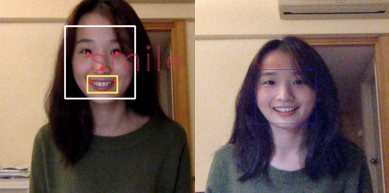
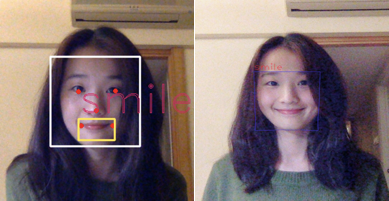
 

Figure 16 Frontal smile face with closed mouth (close) Figure 17.Frontal smile face with open mouth (close)

For a person sitting remotely, if she smiles with closed mouth, linear SVM can catch the tiny change, while CNN misses it. It proves that SVM has more accuracy in closed mouth smile detection. For the case of looking up smile face with open mouth, both models don’t catch it. CNN made wrong prediction while SVM didn’t discover face at all.

Figure 18 Frontal smile face with closed mouth (remote) Figure 19 Looking up smile face with open mouth

## Efficiency

Here the project examines the efficiency of CNN method and Linear SVC method in two angles.

* In real-time show we gradually increase the number of people emerging in the screen to test if the method can detect emoji separately.
* In real-time show, we shake our heads with large volume and different angles to test if these methods have severe latency.

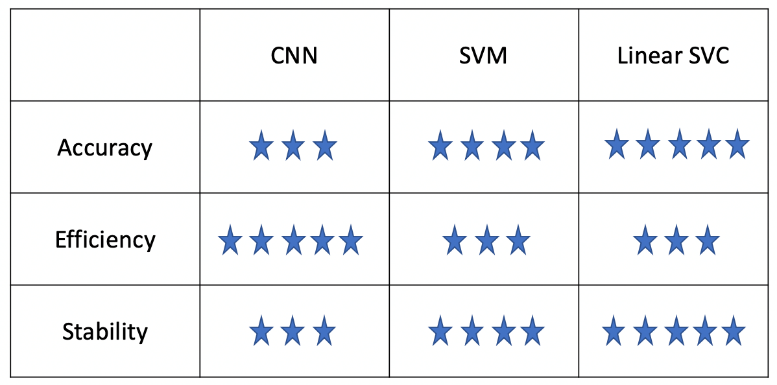
From the video of plan A, the implementation of linear SVM has serious latency, which will significantly influence actual application. From the video of plan B, the implementation of CNN has no obvious latency, which means it can run efficiently.

## Stability

Different from the way to evaluate the efficiency, here the project examines the stability of CNN method and Linear SVC method with adjusting the angle of face slightly. It means that we want to know if tiny fluctuation of input data will change the final prediction results.

From the video of plan A, the prediction result will not change due to tiny fluctuation of head position, which means it is stable. From the video of plan B, the person left shows fluctuated results, which means it has lower stability.

## Summary of Performance



From the chart there is a conclusion that the combination of linear methods and SVM has the highest accuracy. Meanwhile, due to its extraction of whole 68 face landmarks, it has low efficiency. With MTCNN to detect face from original picture, CNN can meet the requirement of actual application. But it might be easily influenced by tiny fluctuation of head position, while SVM and linear SVM don’t have.

## Potential Optimization

Based on work above, there are some possible optimization in future work.

**Robustness**

The detection for non-frontal face emoji still need to improve. Here more advanced face detection algorithm is a possible way.

**Accuracy**

For CNN, detection for closed mouth smile is poor, while SVM is good but restricted by high latency caused by multiple landmark points processing. The combination of them might be useful.

**Efficiency**

To make SVM faster, one possible way is to further reduce number of landmark points used to choose ones that are more relevant to smile emoji.

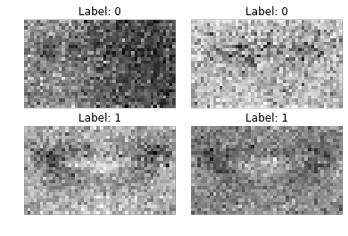


Figure 20

**Data Augmentation: GAN**

Traditional data augmentation methods require knowledge of the underlying task to perform well.

Here the project utilizes GAN to generate more fake images of mouth based on the distribution of existing dataset. Here we can train a generative model that synthesizes new images and their corresponding segmentation masks from random noise.

The preliminary results are listed here. Although the picture is blurry, it can also be distinguished that with label 0 is closed mouth with label 1 means open mouth.

## 

# References and Acknowledgements

**References:**

1. GENKI-4K, <http://mplab.ucsd.edu/wordpress/?page_id=398>
2. OpenCV cascade, https://docs.opencv.org/master/db/d28/tutorial\_cascade\_classifier.html
3. OpenCV + Dlib 68 landmarks example, <https://my.oschina.net/wujux/blog/1622781>
4. Dlib, <http://dlib.net/>
5. Conditional Generative Adversarial Nets, <https://arxiv.org/abs/1411.1784>
6. K. Zhang, Z. Zhang, Z. Li, Senior Member, IEEE, and Y. Qiao, Senior Member, IEEE, "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks," in IEEE Signal Processing Letters, 2016, pp. 1499 - 1503

**Acknowledgements:**

Presentation URL: <https://www.youtube.com/watch?v=ht7uIPfbH3Y&feature=youtu.be>

**Task Assignment:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Task** | **Collaboration** |
| LING Liyang | Plan A implementation | 1. Discussion of topics and solutions 2. Preliminary attempt of several solutions 3. Application framework 4. Production of PPT & video & report materials |
| ZHANG Xichen | Plan B implementation |
| LIU Jinyu | GAN implementation & video editing |
| GAO Han | Evaluation & MTCNN implementation |