COMP4026 Group Project Report

I. Project Tittle

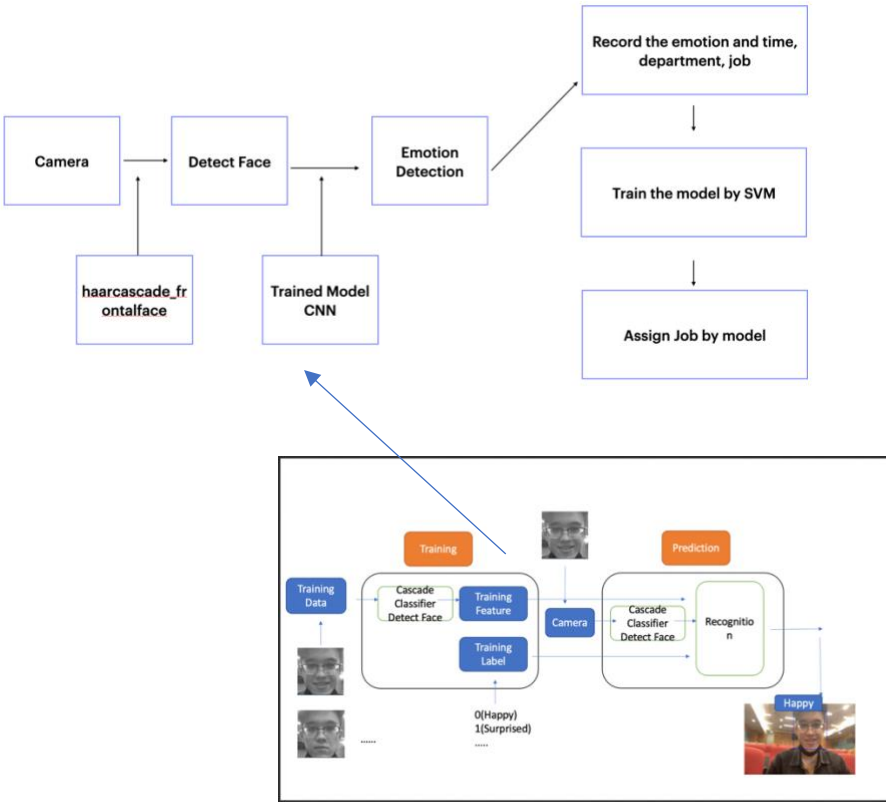
Job Assignment System by Real-time Emotion Detection

II. Project Background

Emotion is the essential role of facial expression and it could provide a lot of information for communication. Emotion recognition is a active research field in the area of computer vision and pattern recognition while it could provide many important functions like monitoring the status of the driver, supervising the candidates during the exam. The individual facial emotion recognition system has been developed powerfully while the multiple facial emotion recognition system and real-time emotion recognition system needed to be improved especially the accuracy of various emotions. In this project, I prepare to develop the system to detect the emotion in the image, video and real-time scenario.

In the company, job assignment is always a problem for the supervisor. How to assign the job to improve the overall efficiency and how to assign the job to specific staff in specific time period of specific department. So I develop the job assignment system to improve the overall working efficiency by real time emotion detection

III. System Overview



The systems are divided into two sub-systems. One subsystem is to train the 2D CNN model and realize the real-time emotion recognition after face detection for each frame by cascade classifier.

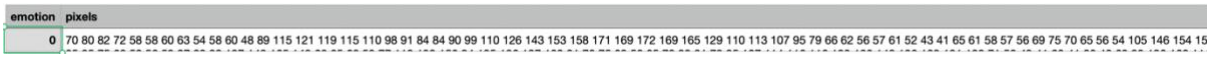
And another sub-system is to utilize the trained model to record the real-time emotion and the corresponding job, timestamp and department to train the SVM model and utilize it.

IV. Data Source

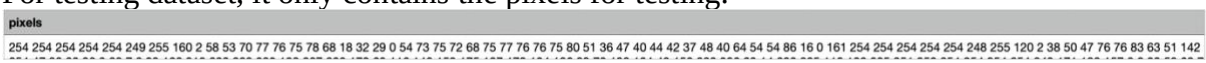
Kaggle Facial Expression Recognition Dataset (https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data)

The data consists of 28,709 training examples and 3,589 test examples which are all the 48 x 48 grayscale images of faces.

For training dataset, the emotion column stands for the emotion class while totally 7 classes included in this dataset (0-> Angry, 1-> Disgust, 2-> Fearful, 3-> Happy, 4->Sad, 5->Surprised, 6-> Neutral). The pixels column stands for image pixels.



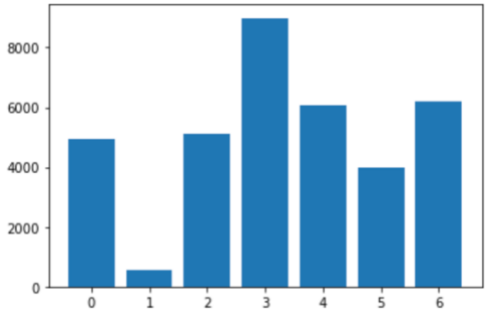
For testing dataset, it only contains the pixels for testing.



The below figure shows the data distribution for each class.

The class 0 has 4953 samples, class 1 has 547 samples, class 2 has 5121 samples, class 3 has 8989 samples, class 4 has 6077 samples, class 5 has 4002 samples and class 6 has 6198 samples.

It is obvious that the imbalanced data issue has happened, especially for the class 1. And I will conduct the several experiments to handle the issue



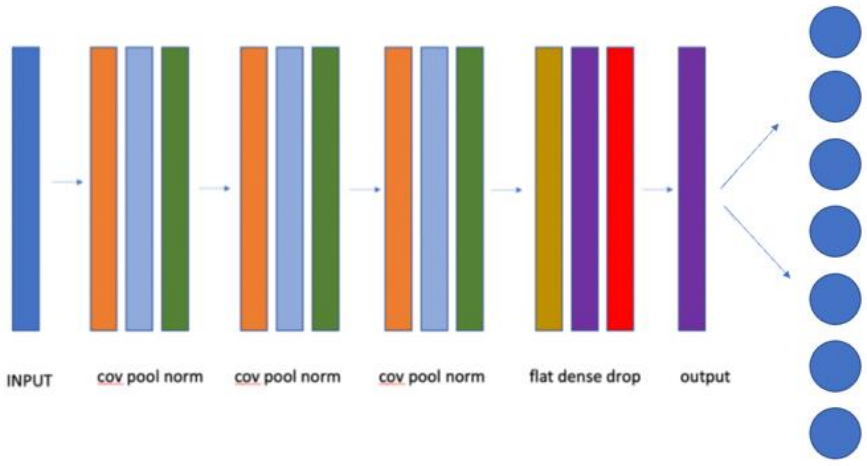
V. Methodology or Algorithm Introduction

a. OpenCV Cascade Classifier

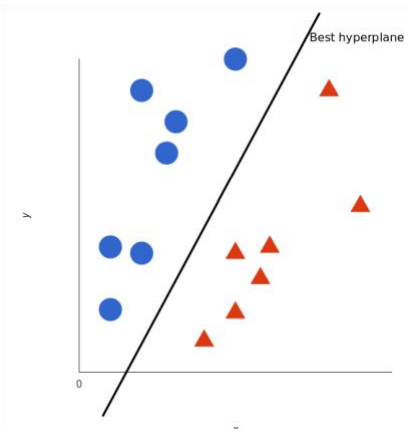
Detect the human face in the image and video which is applied in the lab course

b. 2D CNN

I design the CNN as the below figure with 3 sets of combinations of convolution layer, maxpooling layers and normalization layer. The only difference of them is that the filter amounts of convolutional layer. The filters are 32, 64, 128 and kernel size is 3. For maxpooling layer, the pool size is 2. Then, conduct the batchnormlization layer and drop out layer with drop rate equal to 0.4 to reduce the effect of overfitting. Next, apply flatten layer to flatten the value and one dense layer with 1024 neurons is adopted followed by the output layer while the first part is to do the feature extraction and the second part is to do the classification. And when undertaking the training, adjusting the number of various layers and different parameters to conduct the expirements.



c. Support Vector Machine



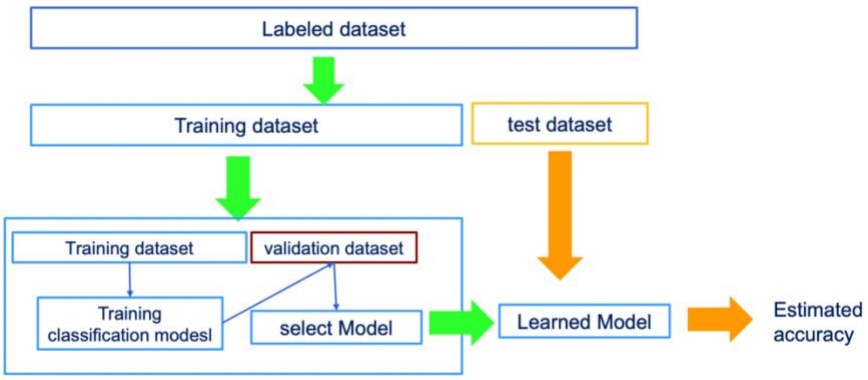
Apply the SVM to predict the emotion based on the given job, time and department.

When the real time emotion detection system is running, the emotion , job , timestamp and department will be recorded. The job, timestamp and department are three features and emotion is the corresponding label. Feed the data into SVM model and train it. Thus, when the supervisor would like to assign the job to someone, he will put the job information, timestamp and department into the pretrained SVM model and get the specific emotion as the output to decide whether this job is suitable for this staff at the timestamp of the department.

Currently, I do not have the real data about job, timestamp and department. Thus I randomly offer the value in a specific range to train the model.

VI. Performance

Divide the dataset into training dataset, validation dataset and testing dataset as the ration 0.64 : 0.16 : 0.2. Conduct the training as the below figure.



a. Experiment I (Original Dataset with imbalanced issue)

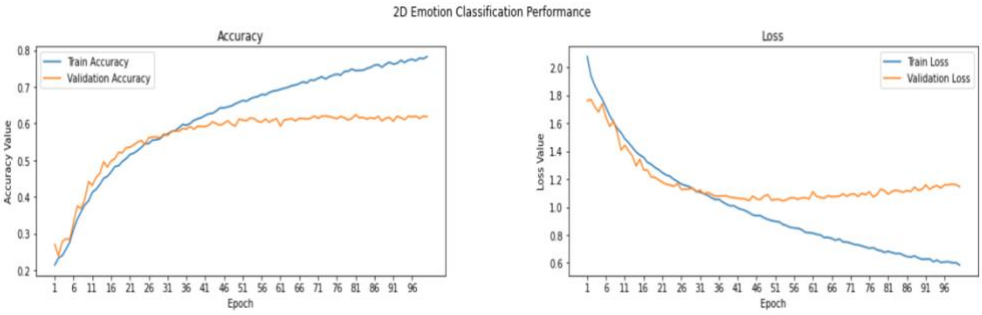
i. Training Parameters

Environment : Colab & GPU
Library : Tensorflow Keras
Batch size: 32
Epochs : 100
Learning rate : 0.0001
Loss function: categorical_crossentropy

ii. Training Loss and accuracy

The below figure is the loss, accuracy for training data and validation data.

The validation accuracy is up to 0.62 and lowest validation loss is 1.18

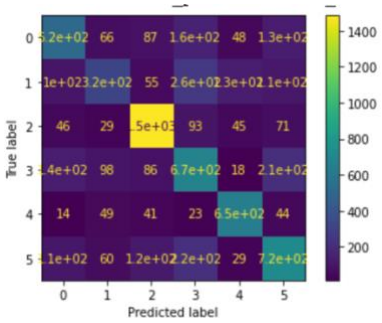


iii. Evaluation Performance

The below figures are evaluation metrics and confusions matrix

The overall accuracy is 0.62
Based on macro average, the precision is 0.60, recall is 0.60 and F1_score is 0.59.
Based on weighted average, the precision is 0.61, recall is 0.62 and F1_score is 0.61

	precision	recall	f1-score	support
0	0.56	0.52	0.54	1013
2	0.52	0.33	0.40	988
3	0.79	0.84	0.81	1768
4	0.47	0.55	0.51	1224
5	0.70	0.79	0.74	818
6	0.56	0.57	0.56	1257
accuracy			0.62	7068
macro avg	0.60	0.60	0.59	7068
weighted avg	0.61	0.62	0.61	7068



b. Experiment II (Add the class weight for training)

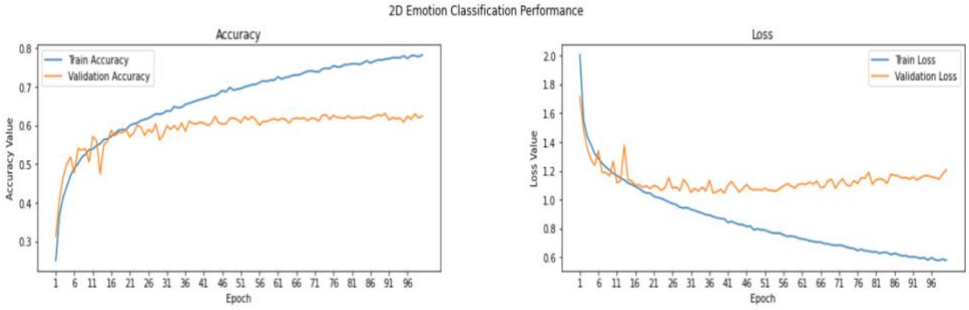
When conducting training by tensorflow keras, we can add the class weight for each class in model fit function to handle imbalanced data issue.

Define the class weight as class_weight = {0:1,1:5,2:1,3:0.8,4:1,5:1.2,6:1},

i. Training Performance

The below figure is the loss, accuracy for training data and validation data.

The validation accuracy is up to 0.62 and lowest validation loss is 1.12



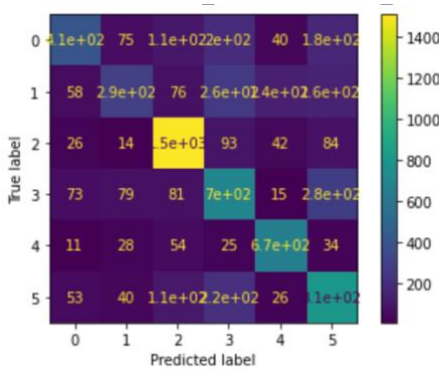
ii. Evaluation Performance

The below figures are evaluation metrics and confusions matrix

The overall accuracy is 0.62
Based on macro average, the precision is 0.61, recall is 0.60 and F1_score is 0.59.
Based on weighted average, the precision is 0.62, recall is 0.62 and F1_score is 0.61

According to the recall, the performance has been improved

	precision	recall	f1-score	support
0	0.65	0.40	0.50	1013
2	0.55	0.30	0.39	988
3	0.78	0.85	0.81	1768
4	0.47	0.57	0.51	1224
5	0.71	0.81	0.76	818
6	0.52	0.65	0.58	1257
accuracy			0.62	7068
macro avg	0.61	0.60	0.59	7068
weighted avg	0.62	0.62	0.61	7068



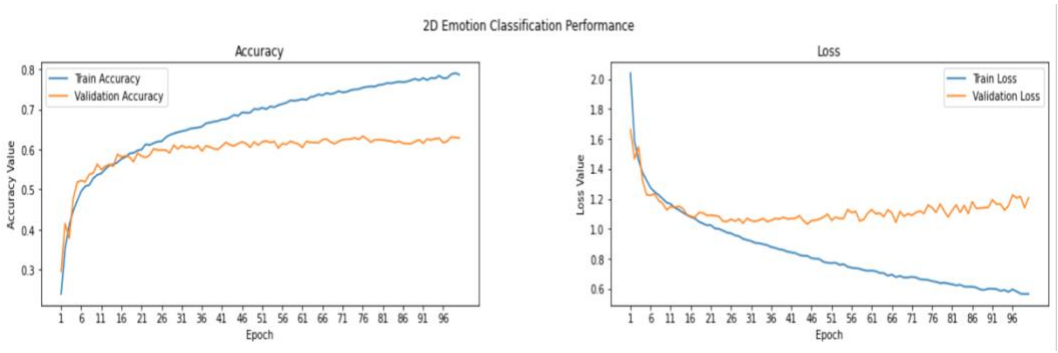
c. Experiment III (Add the class weight for training)

Define the class weight as class_weight = {0:1,1:10,2:1,3:0.8,4:1,5:1.2,6:1},

i. Training Performance

The below figure is the loss, accuracy for training data and validation data.

The validation accuracy is up to 0.63 and lowest validation loss is 1.10



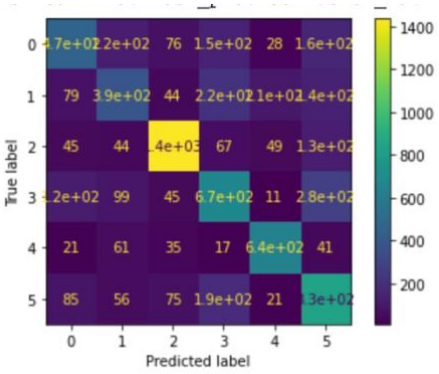
ii. Evaluation Performance

The below figures are evaluation metrics and confusions matrix

The overall accuracy is 0.63
Based on macro average, the precision is 0.63, recall is 0.62 and F1_score is 0.62.
Based on weighted average, the precision is 0.63, recall is 0.63 and F1_score is 0.63

According to the accuracy, recall, precision and F1_score, the performance has been improved compared to the previous two expiremnets.

	precision	recall	f1-score	support
0	0.57	0.46	0.51	1013
2	0.50	0.40	0.44	988
3	0.84	0.81	0.83	1768
4	0.51	0.55	0.53	1224
5	0.75	0.79	0.77	818
6	0.52	0.66	0.58	1257
accuracy			0.63	7068
macro avg	0.62	0.61	0.61	7068
weighted avg	0.63	0.63	0.63	7068



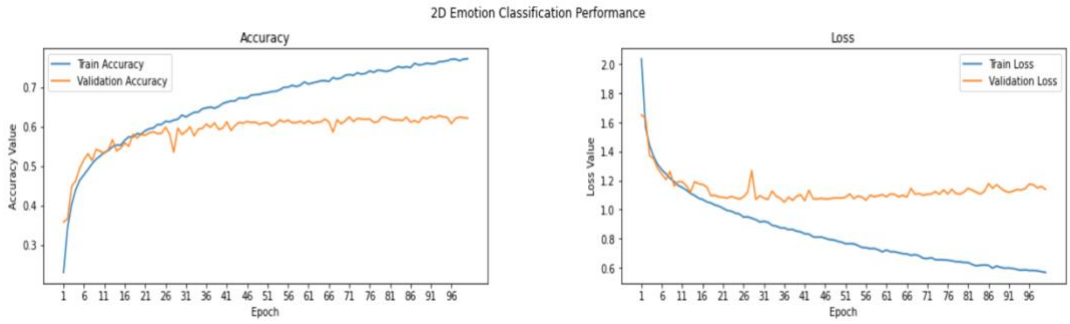
d. Experiment IV (Add the class weight for training)

Define the class weight as class_weight={0:0.9,1:12,2:1,3:0.7,4:0.9,5:2,6:0.9},

i. Training Performance

The below figure is the loss, accuracy for training data and validation data.

The validation accuracy is up to 0.62 and lowest validation loss is 1.10



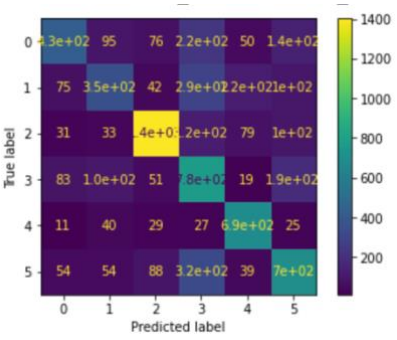
ii. Evaluation Performance

The below figures are evaluation metrics and confusions matrix

The overall accuracy is 0.61
Based on macro average, the precision is 0.61, recall is 0.60 and F1_score is 0.60.
Based on weighted average, the precision is 0.62, recall is 0.61 and F1_score is 0.61

According to the accuracy, recall, precision and F1_score, the performance has been reduced compared to the last two experiments.

	precision	recall	f1-score	support
0	0.63	0.42	0.51	1013
2	0.52	0.36	0.42	988
3	0.83	0.79	0.81	1768
4	0.44	0.63	0.52	1224
5	0.69	0.84	0.76	818
6	0.56	0.56	0.56	1257
accuracy			0.61	7068
macro avg	0.61	0.60	0.60	7068
weighted avg	0.62	0.61	0.61	7068



VII. Conclusion and Discussion

For the performance of several experiments, the optimal model is to utilize the class weight as class_weight = {0:1,1:10,2:1,3:0.8,4:1,5:1.2,6:1} with highest accuracy, recall, precision and F1_score. But compare to other experiments, the performance does nor improve too much. The reason is that the class 1 data size is too smaller compared to other datasets. And increase the weight cannot influence too much on the result.

For the further work, one way is to construct the more effective 2D CNN to achieve performance and another way is to handle the imbalanced data in a better appraoch

VIII. References

a. https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/overview
b. https://techxplore.com/news/2015-11-meh-ugh-facial-emotion-pic.html
c. https://github.com/neha01/Realtime-Emotion-Detection
d. https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html