

Institute of Systems Science, National University of Singapore

MASTER OF TECHNOLOGY

PROJECT REPORT

06.11.2022

—Facial Expressions Recognition System

Team members

- LI YUHENG
- GUO HONGXI
- HUANG CHENXI
- PAULSON PREMSINGH SAMSON DHANSINGH

1 EXECUTIVE SUMMARY	2
2 PROBLEM DESCRIPTION	3
2.1 Project Objective	3
3 Tools/techniques we have used	4
3.1 Data Preprocess	4
3.2 Model Building & Training	4
3.3 Frontend	5
3.4 Backend	5
4 System Design / Models	7
4.1 System Architecture	7
4.2 Web Application	7
4.3 Models	10
4.4 System performance	13
5. Conclusion	15

1 EXECUTIVE SUMMARY

With the rapid development of machine learning and deep neural networks and the popularization of smart devices, face recognition technology is experiencing unprecedented development, and discussions on face recognition technology have never stopped. As an important part of face recognition technology, facial expression recognition has received extensive attention in the fields of human-computer interaction, safety, robot manufacturing, automation, medical treatment, communication, and driving in recent years, and has become a hot spot research topic in academia and industry.

Emotion is a way/manifestation of people expressing their wishes/themselves in daily life, work or study, which means people can communicate with each other without language communication. For example:

- In the service field, service personnel can judge whether the customer is satisfied with the service or interested in the product by recognizing the customer's facial emotion.
- In the class, the teacher can get the students' reaction to the difficulty of the knowledge through the change of students' expressions.
- In the medical industry, it is possible to wear medical equipment with facial expression recognition to determine whether the patient is abnormal so that relevant measures can be taken in time.
- In a criminal investigation, team crimes and experienced criminal suspects often have anti-interrogation training or experience, so facial emotion recognition technology can be used to judge whether a criminal suspect is lying, which is helpful to save interrogation time.

So our team decided to build a facial expression recognition system and had an amazing time working on this project. And we hope the project can be truly used by users. For sure we will continue improving the facial expression recognition system to make it more international and more accurate.

2 PROBLEM DESCRIPTION

In our case, we are thinking that when making commercials, producers often have a headache: how should I know the expected result from the audience? If the audience can have a deeper impression of the brand and product after watching the advertisement? Is the user showing interest in the specific details where we have cleverly designed in the ad? Expression recognition can help ad makers solve this vexing problem.

The producer only needs to acquire viewers' privacy, and invites viewers to watch the commercial ads, and captures facial expressions at specific segments of the ad, and use the expression recognition system to test the viewer's emotions to determine whether the ad meets expectations.

2.1 Project Objective

We hope to build a complete and user-friendly online facial expression recognition(FER) system: for users, they can choose any of the three ways(Upload image/Real-time camera/Upload video) to upload what they want to be analyzed and get the corresponding analysis results via the web application.

It is worth noting that the system is to categorize each face based on the emotion shown in the facial expression into one of seven categories (Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral), but not automatically give us a new facial expression word. And we also consider the performance of the system, the utilized model we aim to achieve an accuracy rate of about 65% and higher.

3 Tools/techniques we have used

Before getting the final facial expression recognition web app, we have to use different kinds of tools and techniques to handle several works such as data preprocess, model building and training, frontend and backend. For us, it was both a practice of skills and a valuable learning experience.

3.1 Data Preprocess

Usually, when talking about facial expression recognition systems, people will firstly think of the famous FER2013 dataset, because FER2013 was designed by Goodfellow et al. as a Kaggle competition to promote researchers to develop better FER systems. The top three teams all used CNNs trained discriminatively with image transformations. The winner, Yichuan Tang, achieved a 71.2% accuracy by using the primal objective of an SVM as the loss function for training and additionally used the L2-SVM loss function. And this was a new development at the time and gave great results on the contest dataset.

And there's quite a lot of different versions of FER2013 displayed on Kaggle, and most of them are very convenient and well-preprocessed, for the data have already been transformed into a csv file with a clear label on every image. Considering the repetitiveness usage of FER2013, our team decided to use a different dataset on Kaggle called MMA Facial Expression Dataset¹. The dataset consists of around 120,000 images and about 3 times more than FER2013, and the dataset is divided into 3 kinds(train, valid and test), and in every kind it has 7 folders which correspond with 7 expressions.

We used python to do the preprocessing work, and import cv2 to read the images. We used numbers 0-6 to represent 7 different expressions and labeled it to every image in a 2d array, and separated it into input and output for all 3 kinds(train, valid and test). And finally we use np.save() to save the numpy matrix for model training in the next step.

3.2 Model Building & Training

We use the sequential model from Keras to build our neural network based on Python, and we use the GPU on Google Colab to train our system, and the building process and

¹ [MMA Facial Expression Dataset](#)

the basic structure of our model will be displayed and discussed in the next part “System Design and Models”.

3.3 Frontend

As for frontend, we use the bootstrap framework to design the style and layout of the webpage. In fact, for some components on the web page, such as the photo container and the result rendering module, we use templates downloaded from the Internet, which is an approach that is both aesthetically pleasing and convenient.

The image page provides the function of recognizing the expressions in the picture. The user can click the button to upload the photo he/she wants to recognize, and then click the "Start analysis "button to get the result. The video interface provides the camera identification function, but it should be noted that the user needs to open the camera permission before using this function. video file page provides the facial expression recognition function in the video, the user can select the video that needs to be uploaded, and click the "Start analysis "button to recognize the emotion of the person in the video.

3.4 Backend

When it comes to the server, we finally decide to deploy the model on the server based on Flask. Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

For this project, we have to build a window or interface for users to upload their files to be recognized, so we find it very efficient and convenient to call the method built in Flask.

Some basic functions in the backend is shown below:

Function	Action
load_emojis()	load emojis and store them to different emotions
gen()	grayscale and resize the image captured by front camera
shotFunc()	call for the front camera and make screenshot
detect()	use GET to get the image and load the model and return the corresponding emotion
upload()	use POST method to upload the image(notice to use secure_filename)
video()	get the "imgname" from shotFunc()

4 System Design / Models

4.1 System Architecture

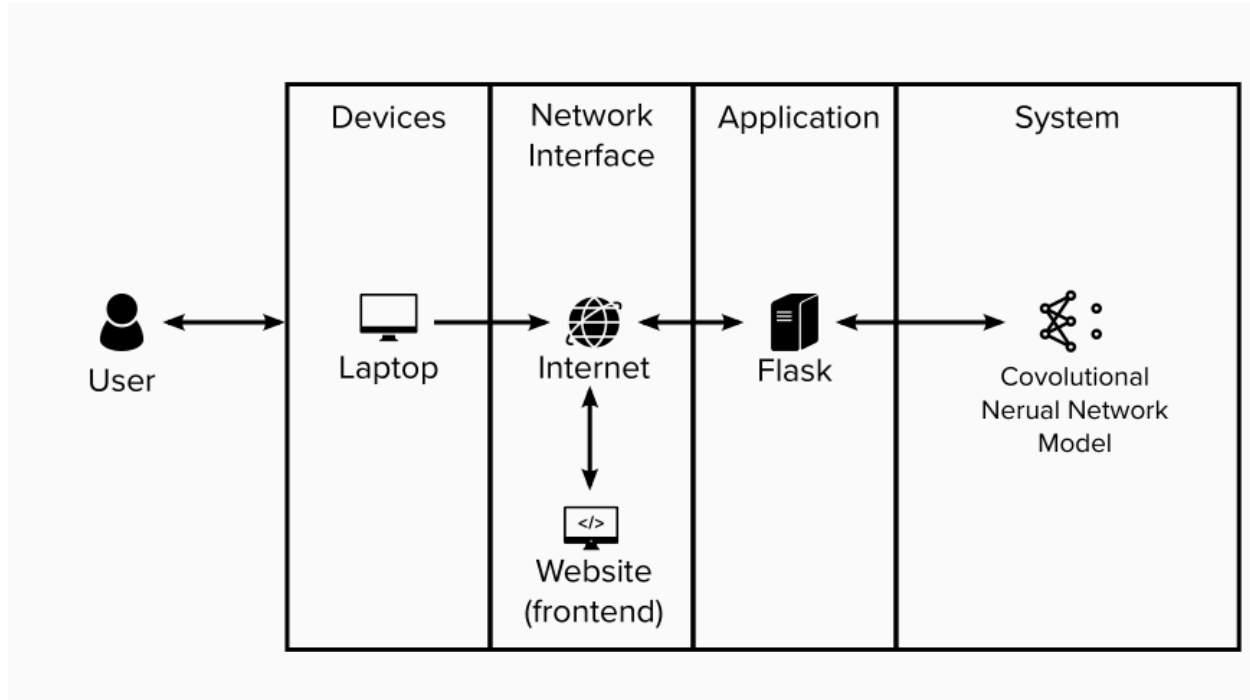


Fig.4.1 - System architecture

This is the basic structure of the whole system: A user accesses our web pages from a personal computer, and uploads the image or uses the front camera to capture the image through our frontend components like buttons. Then the image will be transferred to the backend server, and the server puts the image as an input into our well-trained model. After recognizing the image, the model will give an output as feedback, and similarly, the information will be transferred to the frontend pages through the backend, and finally presented to the user.

4.2 Web Application

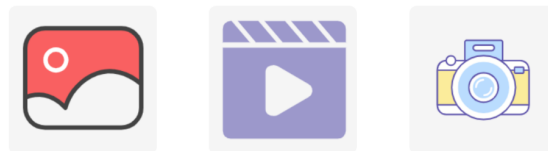
We build a complete and user-friendly online web application: facial expression recognition(FER) system. For users, they can choose any of the three ways(Upload image/Real-time camera/Upload video) to upload what they want to be analyzed and get the corresponding analysis results via the web application:

- Index

😊😐 Facial Expressions Recognition System



You can choose 3 different ways, Please click the picture.



😊😐 @Facial Expressions Recognition System

Fig.4.2 - Index

- Upload image

A user uploads a picture from his/her own computer. The web application performs FER and shows a list of probabilities of emotions that the picture may expose.

😊😐 Facial Expressions Recognition System




Upload graph

Upload video

real-time capture

Upload image



Start analyzing

选择文件 未选择任何文件

Upload image

Recognition Outcome

Angry	0.05%
Disgusted	0.00%
Fearful	0.40%
Happy	0.05%
Sad	99.45%
Surprised	0.01%
Neutral	0.05%

Fig.4.3 - Upload image

- **Real-time camera**

A user uses the front camera from the computers to let the web application get real-time video. The web application displays real-time FER results, including respective prediction confidence for seven different emotions and a corresponding emoji for the identified emotion.

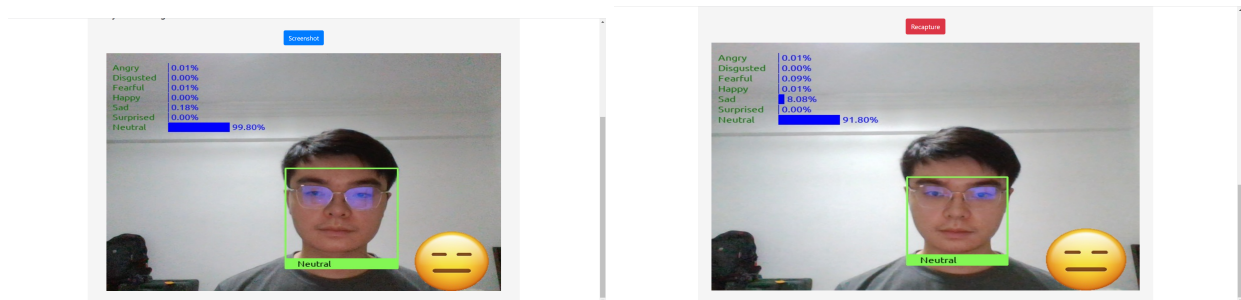


Fig.4.4 - Real-time camera

- **Upload video**

A user uploads a mp4 video file to the web application. The web application performs FER on the video and generates a new video with real-time FER results. The user can play the FER video online or download it.

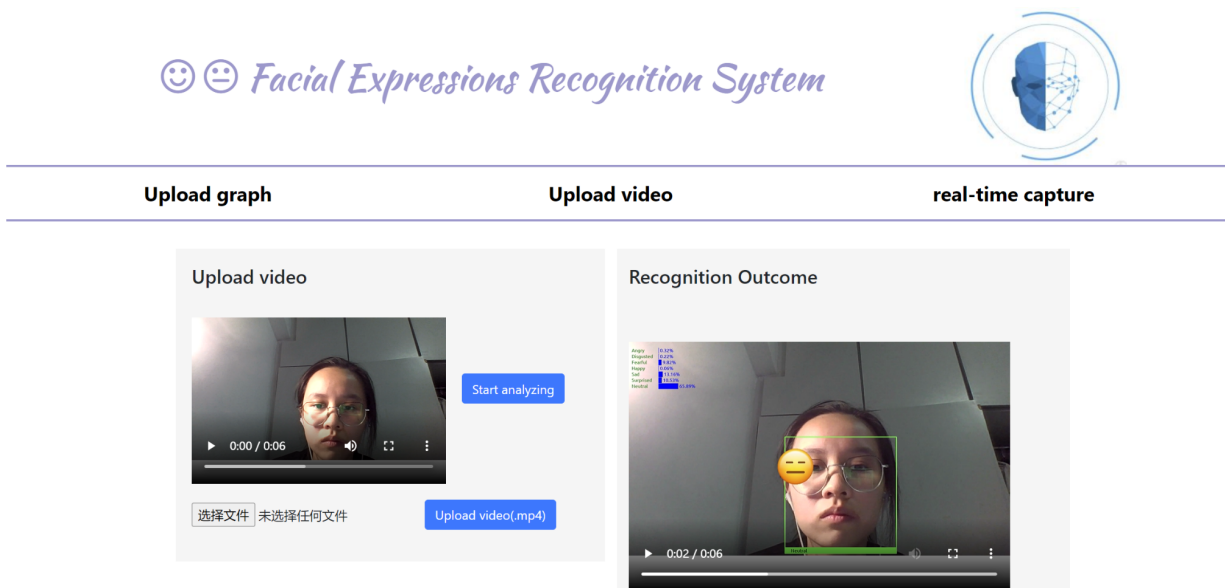


Fig.4.5 - Upload video

4.3 Models

In the process of building a neural network, we encountered many problems like the depth of the model is not deep enough and the dropout rate is a little bit low, and these problems all resulted in the low performance of our model. So after modifying and testing the model, our team built the final version:

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 48, 48, 32)	320
batch_normalization (Batch Normalization)	(None, 48, 48, 32)	128
conv2d_1 (Conv2D)	(None, 48, 48, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 24, 24, 32)	0
conv2d_2 (Conv2D)	(None, 24, 24, 32)	9248
batch_normalization_1 (Batch Normalization)	(None, 24, 24, 32)	128
dropout (Dropout)	(None, 24, 24, 32)	0
conv2d_3 (Conv2D)	(None, 24, 24, 64)	18496
batch_normalization_2 (Batch Normalization)	(None, 24, 24, 64)	256
conv2d_4 (Conv2D)	(None, 24, 24, 64)	36928
batch_normalization_3 (Batch Normalization)	(None, 24, 24, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 64)	0

dropout_1 (Dropout)	(None, 12, 12, 64)	0
conv2d_5 (Conv2D)	(None, 12, 12, 96)	55392
batch_normalization_4 (Batch Normalization)	(None, 12, 12, 96)	384
conv2d_6 (Conv2D)	(None, 12, 12, 96)	83040
batch_normalization_5 (Batch Normalization)	(None, 12, 12, 96)	384
dropout_2 (Dropout)	(None, 12, 12, 96)	0
conv2d_7 (Conv2D)	(None, 12, 12, 128)	110720
batch_normalization_6 (Batch Normalization)	(None, 12, 12, 128)	512
conv2d_8 (Conv2D)	(None, 12, 12, 128)	147584
batch_normalization_7 (Batch Normalization)	(None, 12, 12, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 128)	0
dropout_3 (Dropout)	(None, 6, 6, 128)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 512)	2359808
activation (Activation)	(None, 512)	0
batch_normalization_8 (Batch Normalization)	(None, 512)	2048
dropout_4 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
activation_1 (Activation)	(None, 256)	0

batch_normalization_9 (Batch Normalization)	(None, 256)	1024
dropout_5 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
activation_2 (Activation)	(None, 128)	0
batch_normalization_10 (Batch Normalization)	(None, 128)	512
dropout_6 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 7)	903
activation_3 (Activation)	(None, 7)	0

=====

Total params: 3,002,055
Trainable params: 2,998,983
Non-trainable params: 3,072

Because of the length of the model image, you can check the image of the model in our final zip file.

When it comes to the model training part, We also faced the problem of over-fitting that the test and validation accuracy of the model is much lower than training accuracy. And sometimes the validation loss is erratic: The validation loss first decreased slowly before around the 30th epoch, however it increased during the follow-up training process, and it went even higher in the end compared with the beginning. After searching on Google, Our team adjusted the learning rate which the default value for Adam optimizer is 0.001, and upsized the batch size from 32 to 128 and at last to 512 to avoid over-fitting. And it improved a lot in the end.

And all information about model building and training can be found in our jupyter notebook file.

4.4 System performance

After training the model, we can call the model in our server, and we deploy the built-in face detection module mtcnn/dlib in python to first recognize the human face in the image, and then use our model to detect the facial expression of the image.

And with several tests from our team, we conclude a list of the performance of our model which contains both advantages and areas for improvement:

Advantages:

(1) SYSTEM'S INTELLIGENCE & ROBUSTNESS

This facial expression recognition system is very intelligent because it does not need the user to manually capture the human face in the target image, but can automatically capture the centered human face if the image contains 2 or more human faces. And the recognition process is quick and the outcome is clear to understand and to be analyzed in later steps.

The system is also very robust. There are many fault tolerance settings in the system to improve its robustness. For example, everytime when starting the project, the system will activate the front camera in advance, in case it wastes too much time when we want to call for the real-time recognition. And also the system will prompt the user if the user uploads the unknown format file until the user uploads the file in png, jpg, jpeg and bmp format.

(2) SCALABILITY

There are many new built-in human face detection methods come into our sights via Python, and also if we would have trained a more accurate and better model, it's very convenient for us to update those methods and models in the whole system cause they are written in separate files and clear for us to understand and rewrite.

(3) EASY TO ACCESS

Users don't need to download any software. They only need to enter the correct URL to use this system, which is very convenient.

Disadvantages:

- (1) The system now does not support the upload of bulk files, that means the user can only upload one file at one time and get the recognition output.
- (2) The running time is too long: For the expression recognition function in the video, if the duration of the video exceeds 5 seconds, the running time is very long, which is 1.5 times the usual time.
- (3) In the video part, the size requirements for the video are stricter. It may not be possible to analyze some vertical screen videos.
- (4) Not easy to access: If the user wants to use the system, they have to download the code files and configure the required environment.

5. Conclusion

In this project, we build a complete and user-friendly online facial expression recognition system that can recognize/analyze facial emotions in pictures, videos, and Real-time camera. In terms of data, considering the repeated use of FER2013, we used another dataset on Kaggle called the MMA facial expression dataset which consists of around 120,000 images and about three times more than FER2013. Compared to other datasets, this dataset is divided into three categories: test set, training set, test machine, and validation set. And in every category, it has seven folders that correspond with seven expressions.

In the model building section, we use the sequential model from Keras to build our neural network. During the course of the model building, We encountered some problems leading to poor model performance, like the model was not deep enough, and the dropout rate was a bit low. To solve this problem, we added convolutional layers and adjusted dropout rate value.

In the model training part, we found the problem of overfitting and decided to adjust the learning rate and upsized the batch size from 32 to 512 to avoid over-fitting to solve this problem.

Through this project, the members of our group had a wonderful time working on this project, and definitely picked up useful skills along the way. We have a deeper understanding of the knowledge learned in class and also understand how to apply the neural networks to the project. In addition, we also learned how to work better as a team and how to communicate with each other more effectively.