WORKPLACE STRESS LEVEL DETECTION

Wang Hongtao, Yang Yizhou

Institute of Systems Science, National University of Singapore, Singapore 119615

ABSTRACT

The stress in our society is so high which causes many suicide in workplace, based on this background we aim to develop a system which can evaluate stress level immediately. Our system use DNN detect face and feed the cropped face region into model to differentiate expression, then reason stress level from collected time series expression using Hmm model which can help manager and HR react timely in workplace.

Index Terms— Face Detection, Emotion Recognition, HMM, Stress

1. INTRODUCTION

1.1. background

Nowadays, with the rapid growth of the economy, in our society people are facing all kinds of pressure like family pressure or economic pressure because of the highly competitive and peer evaluations, and more and more companies put a lot of pressure on their employees.

High levels of pressure will make people feel stressed, angry, upset or even depressed, and all these negative emotions can influence employees' physical and mental health, also these will affect the work efficiency, which is a situation that employers and employees are unwilling to see.

So we decide to create a system which can detect real-time emotions for staff, then evaluate the pressure by these motions. The input is a video (from monitors or webcam) and the output is a chart which indicates the pressure level for each staff.

1.2. Objectives

The purpose of our system is detecting the emotion of employees and using emotion data to reason about employees' stress level. Finally we can generate a chart to reflect the pressure level of staff for managers and HR.

For managers, based on the chart, they can evaluate the pressure by comparing the assigned work and roughly conclude whether the causes of pressure are from workloads or other aspects. Managers can decide to reduce or add work tasks to employees.

For the HR department, based on this chart, they can find some potentially stressed employees in a more timely manner, and send them timely care before things get worse.

2. LITERATURE REVIEW

Our system can be divided into 3 main sections: face detection, emotion recognition, reasoning stress level using HMM.

2.1. Face decetion

For face detection, we find many approaches can fulfill the requirement, such as OpenCV Haar-cascade, OpenCV DNN, Dlib HoG, Dlib CNN... We try different approaches and compare these approaches in different situations.

2.1.1. OpenCV Haar-cascade

Advantages 1) It can work almost in real time on the CPU; 2) Simple architecture; 3) It can detect faces of different proportions.

Disadvantages 1) There will be a lot of predictions of non-human faces as human faces; 2) Not suitable for non-frontal face images; 3) Not anti-occlusion.

2.1.2. OpenCV DNN

Advantages 1) The most accurate of these four methods; 2) It can run in real time on the CPU; 3) It is suitable for different face orientations: up, down, left, right, side, etc. 4) It can work even under severe occlusion; 5) It can detect faces of various scales.

2.1.3. Dlib HoG

Advantages 1) The fastest method on the CPU; 2) Suitable for frontal and slightly non-frontal faces; 3) The model is small compared to the other three; 4) It can still work under small occlusions.

Disadvantages 1) Small faces cannot be detected, because the minimum face size of the training data is 80×80, but users can train the detector with smaller face data; 2) The bounding box usually excludes part of the forehead and even the chin

Part; 3) Cannot work well under severe occlusion; 4) Not suitable for side and extreme non-frontal, such as looking down or looking up.

2.1.4. Dlib CNN

Advantages 1) Suitable for different face orientations; 2) Robust to occlusion; 3) Working very fast on GPU; 4) Very simple training process.

Disadvantages 1) CPU speed is very slow; 2) Cannot detect small faces, because the minimum face size of its training data is 80×80, but users can train the detector by themselves with smaller size face data; 3) Face surround The frame is even smaller than the DLib HoG face detector.

2.1.5. Conclusion

Finally compared with these approaches, we decide to use OpenCV DNN to detect face, because compared with Haar-cascade, it can detect not limit for frontal face, and is effective for occlusion.

2.2. Emotion Recognition

Rank	Model	Accuracy	Extra Training Data	Paper	Code	Result	Year	Tags
1	FER-VT	90.04	✓	Facial expression recognition with grid-wise attention and visual transformer			2021	
2	LResNet50E-IR	89.257	✓	Exploring Emotion Features and Fusion Strategies for Audio-Video Emotion Recognition			2020	
3	RAN (VGG-16)	89.16	✓	Region Attention Networks for Pose and Occlusion Robust Facial Expression Recognition			2019	
4	SENet Teacher	88.8	✓	Emotion Recognition in Speech using Cross-Modal Transfer in the Wild			2018	

Fig. 1. Ranking Result of Facial Expression Recognition on FERPlus

Initially, we set up our model using FER+ dataset to predict emotion, but we found low accuracy around 0.65, can not reach a good result, so we found from existing approach, we can see from Figure 1, the top ranking accuracy result of Facial Expression Recognition on FERPlus, and we learn from the structure of network to train our model, finally get a better accuracy for emotion recognition.

3. DATASET

The face recognition part uses an off-the-shelf model, so instead of presenting its dataset here, this section focuses on the dataset used for expression recognition.

3.1. FER2013plus[1]

We are using the FER2013plus dataset, which is a re-labelled version of the FER2013[2] dataset by Microsoft.

We begin with the FER2013 dataset. "The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).train.csv contains two columns, "emotion" and "pixels". The "emotion" column contains a numeric code ranging from 0 to 6, inclusive, for the emotion that is present in the image. The "pixels" column contains a string surrounded in quotes for each image. The contents of this string a space-separated pixel values in row major order. test.csv contains only the "pixels" column and your task is to predict the emotion column.The training set consists of 28,709 examples. The public test set used for the leaderboard consists of 3,589 examples. The final test set, which was used to determine the winner of the competition, consists of another 3,589 examples.This dataset was prepared by Pierre-Luc Carrier and Aaron Courville, as part of an ongoing research project. They have graciously provided the workshop organizers with a preliminary version of their dataset to use for this contest."[2]

However, some of the images in this dataset were mislabelled and had many unrelated animated cartoon images, so Microsoft re-labelled it in 2016 and named it FER2013 plus.

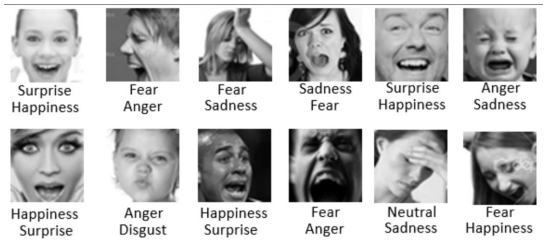


Fig. 2. Example of FER+

"The FER+ annotations provide a set of new labels for the standard Emotion FER dataset. In FER+, each image has been labeled by 10 crowd-sourced taggers, which provide better quality ground truth for still image emotion than the original FER labels. Having 10 taggers for each image enables researchers to estimate an emotion probability distribution per face. "[1]

3.2. Pre-processing

Before we can use the fer2013 plus dataset, we need to download the re-labelled tables, and the original fer2013 dataset.Then the fer2013 dataset was reclassified based on the annotated information. In the FER2013 plus dataset, there are a total of eight categories for expressions, which are 'neutral', 'happiness', 'surprise', 'sadness', 'anger', 'disgust', 'fear', 'contempt'.

3.3. Training & Testing

For the training process, we used 28558 images as the training set, 2579 images as the validation set and 3573 images as the test set.

4. PROPOSED SYSTEM

4.1. Flow chart

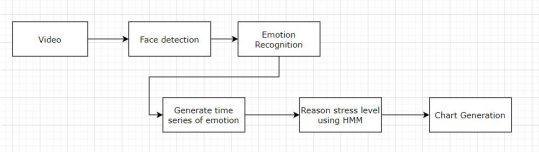


Fig. 3. Flow Chart

From Figure 3 We can see our structure of system. Our system is divided into 5 parts: Face detection, Emotion recognition, Generate time series of emotion for each person, Reason stress level using HMM model, Chart Generation. We use one video which have four people as our test video to test each function.

4.2. Face detection

The face detection is locate the face and label the face. As mentioned from Part2.literature review we test several approaches to fulfill face detection function, and we compare the advantages and disadvantages, found OpenCV DNN performs better than other approaches. Initially we used Haar-cascade, some frames we can not detect face because of obstacles and not frontal face. Finally we use DNN to do face detection.

4.3. Emotion recognition and generate time-series emotion

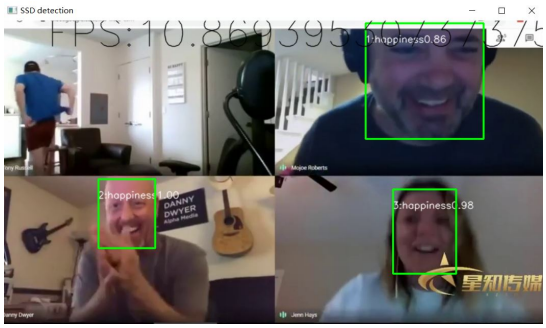


Fig. 4. Emotion Recognition in our test video

The emotion recognition is recognize person expression. The input is cropped face region in face detection part, and feed into our emotion recognition model to differentiate expressions.

We use FERPlus (Face Expression Recognition Plus dataset) as our dataset to train the model. The FER+ dataset is an extension of the original FER dataset, where the images have been re-labelled into one of 8 emotion types: neutral, happiness, surprise, sadness, anger, disgust, fear, and contempt.

We also compare different structure of existing model and learn from them, then train our own model to recognize emotion. The final result is labelling face emotion, like Figure 4. From figure we can see in our test video, we label each person number and their emotion separately, and record them then use Hmm model to reason their stress.

4.4. Reason stress level using Hmm and generate chart

Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process — call it — with unobservable ("hidden") states. As part of the definition, HMM requires that there be an observable process whose outcomes are "influenced" by the outcomes of in a known way. We use HMM model to reason stress level for each person.

After that we can make managers and HR to analysis the chart to make sure each staff mental and physical healthy.

For managers, based on the chart, they can evaluate the pressure by comparing the assigned work and roughly conclude whether the causes of pressure are from workloads or other aspects. Managers can decide to reduce or add work tasks to employees.

For the HR department, based on this chart, they can find some potentially stressed employees in a more timely manner, and send them timely care before things get worse.

5. EXPERIMENTAL RESULTS

5.1. Face detection

At first we intended to use the haar cascade classifier in opencv to detect faces, but in practice, we found that the haar cascade classifier was too ineffective for occlusion, so we replaced it with a SSD to detect faces.

5.1.1. Haar cascade classifier

In Figure 5, we can see that the faces in the top left and right corners are not recognised because the face in the top left corner is wearing a hat and the face in the top right corner has its head obscured.

This situation is actually very common in real life, so we intended to implement a different approach with a higher accuracy rate. After searching for information, we decided to use an SSD-based face recognition model.

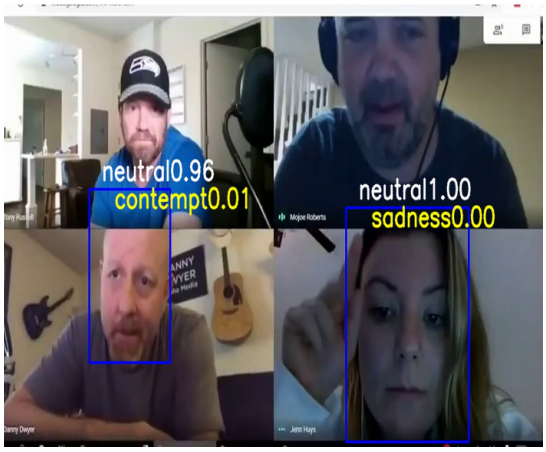


Fig. 5. Face detection by Haar cascade classifier

5.1.2. SSD-based face detection model

We used a trained SSD model as a face detection model. The model uses res10 as the backbone, the name called "res10_300x300_ssd_iter_140000_fp16.caffemodel".

Using this model, our face results are much more correct.As shown in Figure 6, the model detects faces with occlusion, located in the top left and top right corners respectively.

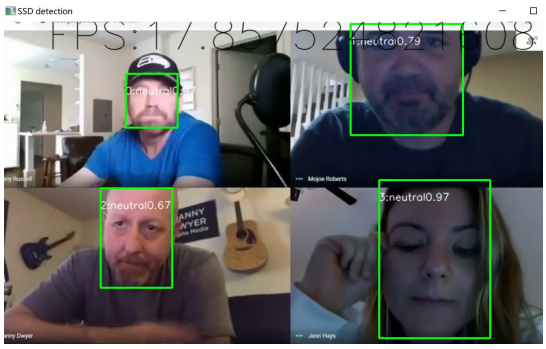


Fig. 6. Face detection by SSD model

5.2. Expression detection

Once we have the position of the face, we can put the face region into our own trained expression detection model to recognize expressions.

5.2.1. CNN Model

As shown in Figure 7, we have used VGG-13 as the backbone

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 64, 64, 64)        640
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 64, 64, 64)        36928
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 32, 32, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 32, 32, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 128)       73856
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 32, 32, 128)       147584
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 128)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 16, 16, 128)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 16, 16, 256)       295168
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 16, 16, 256)       590080
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 16, 16, 256)       590080
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 256)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 8, 8, 256)         0
_________________________________________________________________
flatten (Flatten)            (None, 16384)             0
_________________________________________________________________
dense (Dense)                (None, 1024)              16778240
_________________________________________________________________
dropout_3 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dropout_4 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 8200
=================================================================
Total params: 19,570,376
Trainable params: 19,570,376
Non-trainable params: 0
```

Fig. 7. model layers

5.2.2. Initialization & Training

In fact, it took us close to a week to train the model. From a complete non-convergence at the beginning to the final result, we tried many parameters, and unfortunately we did not record our wrong attempts, so the parameters provided here in this chapter are the ones that were arrived at after much trial and error. The method of initialising the weights is also important in the fine-tuning process and we have tried "He_normal", "glorot_uniform" and other methods to initialise the weights. Based on the convergence performance and results, we finally chose "glorot_uniform" as the initial weighting function.

Figure 8 shows some of the parameters that we use for initialisation.

```
def vgg13_model(n_classes):
    model = Sequential()
    model.add(Conv2D(64, (3, 3), strides=(1, 1), input_shape=(64, 64, 1), padding='same', activation='relu',
                     kernel_initializer='glorot_uniform'))
    model.add(Conv2D(64, (3, 3), strides=(1, 1), padding='same', activation='relu', kernel_initializer='glorot_uniform'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    model.add(Conv2D(128, (3, 3), strides=(1, 1), padding='same', activation='relu', kernel_initializer='glorot_uniform'))
    model.add(Conv2D(128, (3, 3), strides=(1, 1), padding='same', activation='relu', kernel_initializer='glorot_uniform'))
```

Fig. 8. The parameters in initialization

We also tried many combinations of parameters for the training part, as shown in Figure 9, and finally we decided to use "AdaGrad" for the optimizer, with the batch size set to 64, and train 100 epochs.

```
model.compile(
        optimizer="sgd",
        metrics=['accuracy'],    #评价指标
        loss='categorical_crossentropy')   #计算损失---分类交叉熵函数，  binary_crossentropy（二分类）
```

Fig. 9. The parameters for training

5.2.3. *Results*

As shown in Figure 10, the model starts overfitting at 30 epochs, so we take the model at 25 epochs, and the accuracy is 80.1% at this time. In our previous tests we actually found

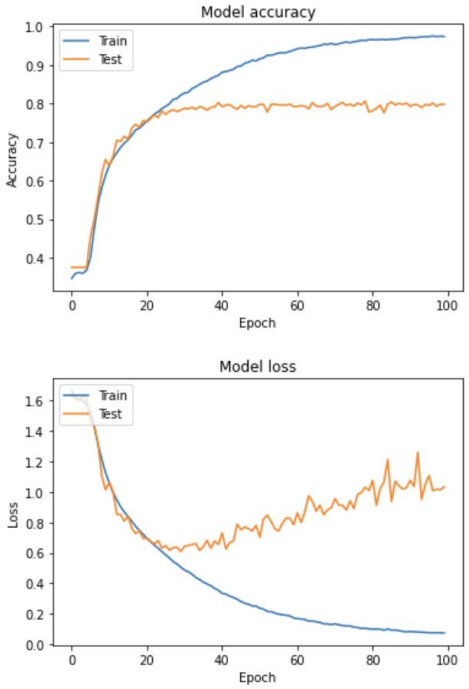


Fig. 10. Training history

overfitting, more so than in Figure 11. We therefore adopted the following approach.

1. Add dropout layers.

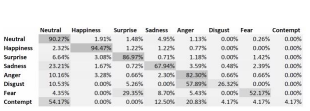
2. Reduce the depth of the CNN. We removed the number of layers in VGG-13 where the number of kernel is 512. In total, three convolution layers have been reduced.

As shown in Figure 13, we use the same backbone net, the same label fusion method as Microsoft, their model accuracy is 83.8%, our accuracy is 80.1%, I think this result is acceptable.

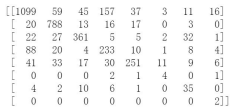
Table 1. The performance comparison.

Model	Proposed approach
Microsoft VGG-13	0.8389
Model proposed in this report	0.8015

Figure 11 shows the confusion matrix for the two models.Analysing this confusion matrix, we can see that the accuracy is fair, but there are some similar expressions that cause false predictions between them, such as 'Anger' and 'Disgust', 'Neutral' and 'Contempt', 'Surprise' and 'Fear', 'Neutral' and 'Sadness'.



Microsoft VGG-13



Model proposed in this report

Fig. 11. Confusion matrix

Most of the expressions that were predicted incorrectly were among the more stressful expressions in our classification, so this will need to be recorded specifically in subsequent predictions.

5.3. HMM

The Figure 12 show the observation matrix, and the Figure 13 show the state transform matrix we can observe each person emotion (Neutral, Happy, Surprise, Sadness, Anger, Disgust, Fear, Contempt) from the video, and the state is stress level, in our system we divide into five level: HH(very happy), H(happy), N(neutral), LS(low stress), HS(high stress).

	Neutral	Happy	Surprise	Sadness	Anger	Disgust	Fear	Contempt
HH	0.1	0.3	0.6	0.0	0.0	0.0	0.0	0.0
LH	0.2	0.6	0.2	0.0	0.0	0.0	0.0	0.0
N	0.4	0.15	0.05	0.15	0.05	0.05	0.05	0.1
LS	0.1	0.0	0.0	0.3	0.1	0.1	0.1	0.3
HS	0.1	0.0	0.0	0.15	0.2	0.2	0.2	0.15

Fig. 12. Observation Matrix

	HH	LH	N	LS	HS
HH	0.2	0.4	0.2	0.1	0.1
LH	0.2	0.4	0.2	0.1	0.1
N	0.1	0.2	0.4	0.2	0.2
LS	0.1	0.1	0.2	0.4	0.2
HS	0.1	0.1	0.2	0.4	0.2

Fig. 13. Transition Matrix

When we test our system we run 15 seconds video to collect each person emotion. In our test video we can detect 4 person, we separately label each person 0,1,2,3. we know for emotion dict = 0: "neutral", 1: "happiness", 2: "surprise", 3: "sadness", 4: "anger", 5: "disgust", 6: "fear", 7: "contempt", stress level = 0: "HH", 1: "LH", 2: "N", 3: "LS", 4: "HS" We assume emotion to stress = 0: 2, 1: 3, 2: 4, 3: 1, 4: 0, 5: 1, 6: 0, 7: 1, so we create observation matrix and transition matrix like above figure, then we can reason each person stress level

for 15 seconds video. We can see stress level result for each person from Figure 14 and then we generate a chart collect these stress level for each person to clearly show the stress, like Figure 15.

Fig. 14. Stress Level for each person

6. CONCLUSIONS AND FUTURE WORK

As shown in Figure 15, we end up with a report on the change in an individual's stress level over time.The horizontal coordinate is time and the vertical coordinate represents the sequence of pressures over time

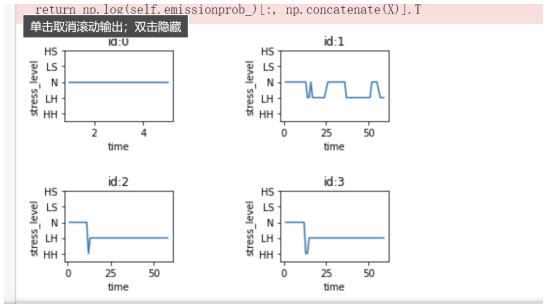


Fig. 15. Report of this System

With this report, the supervisor, or the HR department, can make a comprehensive assessment of the person's stress level and make interventions.

6.1. Future work

We have also identified a number of problems that need to be addressed in future work.

1. Expression recognition

Although the overall expression recognition accuracy is relatively high, some expressions expressing stress are less well recognised and pose difficulties in real-time detection.

As the confusion matrix in figure 11 shows, this model has difficulty distinguishing between some similar expressions. Even asking humans to judge a person's instantaneous expressions, such as 'Surprise' and 'Anger', is difficult.

In our application, we took a video of a time period and predicted all the expressions in that time period and then chose the expressions that appeared the most to represent that time period. Although this method improves the performance of the recognition, it does not essentially solve the problem.

In the future it may be possible to incorporate a variety of judgement criteria, such as voice, the reaction of those around you, atmosphere and other factors to predict expressions in combination.

Fig. 16. What is this expression? "Fear", "Anger" or "Surprise"?

2. From expression to emotion

In terms of our everyday experience, it can be difficult to spot a person's emotions, especially in public. People are very good at hiding their emotions in public. So deducing the person's emotions just by their expressions is actually very difficult. In the future, if we want to increase the accuracy rate, we may seek the help of some psychologists so that we can look for the characteristics of human emotions, not only expressions but also possibly actions, tone of voice, reactions, etc.

6.2. Conclusions

The system proposed in this paper only provides an idea of how to move from expression recognition to emotion recognition and, in practical terms, is still far from our goal.

Because the perceived expression of emotions is a very subjective thing for humans. Even some people may not have the ability to do this, which is also known as emotional intelligence. Quantifying emotion is very difficult, but not impossible to achieve, and the method offered in this paper is one way to quantify emotion through expressions.

It is believed that someday in the future machines will be able to communicate with humans emotionally and master how to accurately recognise emotion in faces.

7. CONTRIBUTIONS

Full Name	Roles
Wang Hongtao	UI design and development Backend function Project report writing
Yang Yizhou	Model training Application Coding Report Writing

8. REFERENCES

[1] Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, and Zhengyou Zhang, "Training deep networks for facial expression recognition with crowd-sourced label distribution," in *ACM International Conference on Multimodal Interaction (ICMI)*, 2016.

[2] Aaron Courville Pierre-Luc Carrier, "fer2013," .