Classify Human Face Emotions with Convolutional Neural Network and Decision Tree Algorithm

Jiayu Wang
Research School of Computer Science
The Australian National University
Canberra ACT 2601
u7102306@anu.edu.au

Abstract. In this paper, the research question is to classify the human face emotions by analyzing input images with different methods. The dataset used here is SFEW and it has two versions, one is raw data, and the other is pre-processed with histogram equalization. For the models, I implement three methods including Decision Tree Classifier, LeNet and Customized Convolutional Neural Network. Then I compare the performance between methods and different version of data and find out that the data without any adjustments has better performance for all methods and among the three methods, customized CNN has best performance for the same input data in between about 38.52% and 40.74%. And in the dataset paper, for SFEW, the accuracy is 43.71% for LPQ and 46.28% for PHOG, which is slightly higher than my research, but the performance is still nice compared to the natural possibility of 14.29%.

Keywords: Multi-class Classifications, Convolutional Neural Network, LeNet, Decision Tree, SFEW

1 Introduction

1.1 Background and Motivation

Facial expression is one of the most powerful, natural, and common signals of human expression for emotional state and intention, it is an important aspect for communicating and coordinating interpersonal relationship. According to a research conducted by a psychologist, the information passed through language only occupied 7% of the whole message while the information passed through facial expressions occupied 55% of the whole message during daily life communications of human beings. Thus, facial expression is a form of nonverbal communication, the main ways to express social information between human beings and has great commercial value and social significance. In today's society, the applications for recognizing human face emotions are widely used in robot, medical treatment, driver fatigue monitoring and many other human-computer interaction systems, so people have carried out a lot of research on automatic facial expression analysis, which is vital and meaningful. Moreover, this research mainly focused on classification in real world environment which made the question more challenging.

1.2 Dataset

The dataset used in this research is human face emotions, also named as SFEW. It is a dynamic temporal facial expressions data corpus, which has a close to real world environment, collecting from clips of 37 movies.

SFEW was collected based on the Subtitles for Deaf and Hearing impaired (SDH) and Closed Caption (CC). To be more concise, the researchers searched the expression keywords such as 'smiles', 'cries', 'sobs', 'scared', 'shouts' and 'laughs' from SDH and CC, combined with time stamps that could indicate which part of the movie contained a more meaningful facial emotions that could be extracted, and after that a human observer annotated the collected clips about actors and expressions manually. In paper [1], the above data was preprocessed to become the low-dimensional numerical data by applying principal component analysis (PCA), and the researchers use two kinds of descriptors, PHOG and LPQ for implementation. However, in this research, the used form of dataset is image.

And there are seven emotions in this dataset which are 'angry', 'disgust', 'fear', 'happy', 'neutral', 'sad' and 'surprise', stored in seven files respectively under a total file. The number of images for 'angry', 'fear', 'happy', 'neutral', 'sad' and 'surprise' is 100 and the number of images for 'disgust' is 75, so there are 675 images in total. The images are in RGB form and the size of them is 576*720. Most of the images only contains one face, but there are still some images that contain a main face and half of the other face in different angles and the labels of these images are defined by the main face emotions, sampled in ***Figure 1.***

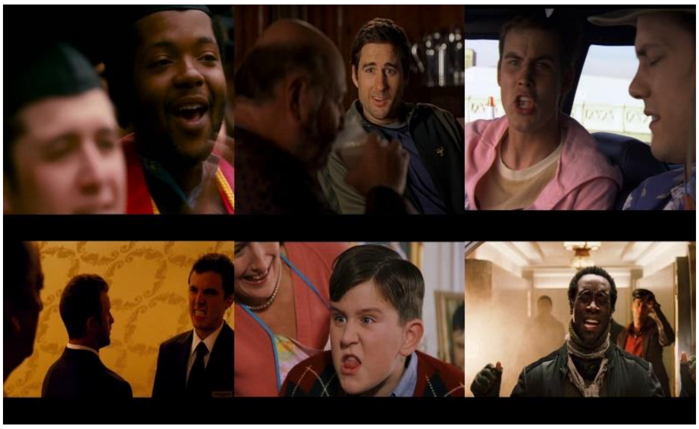


Figure 1. Sample images that have two or more faces showed in class Happy, Angry and Disgust.

1.3 Problem Description and Model Investigation

The main research problem is to classify the images of human face emotions into seven classes. In the first edition to solve the problem, I first used a Neural Network as the model to solve the problem and then found out that although it can classify the emotions, I cannot figure out the inside logics and rules, namely it was not explanatory and was like a black box to human beings, having relatively low interpretability. Then I used Decision Tree algorithm to explore the inner rules and it did not have a good performance though it still had better performance than Neural Network.[2]

To extend, for this research, I also use Decision Tree algorithm, but I change the method Neural Network into Convolutional Neural Network to try to improve the performance. Additionally, I would also like to make inputs in the form of the images with pixels without any dimension reduction and descriptors to compare the performance between the one in the form of LPQ and PHOG.

There are many works about implementing CNN for facial expression recognition. (Liu, Zhang & Pan, 2016) designed a model that consists of several different structured subnets and each subnet is a compact CNN model trained separately. And another research (Shin, Kim & Kwon, 2016) analyzed about baseline CNN architectures and preprocessing methods. Then they found out that among five preprocessing methods (raw, histogram equalization, isotropic smoothing, diffusion-based normalization, difference of Gaussian), a simple three-layer structure consisting of a simple convolutional and a max pooling layer with histogram equalization image input had the best performance.

Thus, for Decision Tree algorithm, I will use the Decision Tree Classifier and adjust the parameters for better performance. And for Convolutional Neural Network, I try two architectures. One is the traditional LeNet, and the other is normal CNN with customized parameters and the detailed structure would be introduced in method section. Additionally, I would use the raw input as well as the histogram equalization for comparison.

2 Method

2.1 Data Pre-process

Since the form of data in dataset SFEW (Static Facial Expressions in the Wild) is image and they are saved in seven different files, there are no specific file for labels. Thus, I use the number zero to six to represent 'angry', 'disgust', 'fear', 'happy', 'neutral', 'sad' and 'surprise' respectively for convenience. I traverse the dictionary which stores the image, read all the images one by one firstly and then append all pixel values to a list every time, meanwhile I also append the label number to the label list. After all the iterations, I can get the all the values for images of one emotion and these need to be done seven times. Next, I split the data into training set and testing set with proportion of 80% and 20% and set random state to 4 to be more stable. All above are the process to get usable raw data.

As mentioned before, I also use histogram equalization to adjust the input for better performance and to implement this, only one step needs to be added. That is, to first separate RGB channels, apply histogram equalization to each channel, and then merge them together lastly. Moreover, for data that need to be the inputs for convolutional neural network, they need to be transformed into tensor dataset to fit pytorch.

2.2 Decision Tree Classifier

The method in technique paper contains logic as “if-else” clause then conclude an output which has pretty much similar idea of a decision tree, so I consider implementing a decision tree classifier to recognize the facial emotions by using ID3 algorithm after comparison. Namely, I use entropy as criterion, and I also set the random state for both dataset and classifier with a fixed number for easier and more reliable comparison.

After training, I print out all the metrics like accuracy, recall or r-square to compare the results and use five-fold cross-validation to get an evenly accurate score. By doing this, the original dataset is divided into five segments and every time the model takes four of them, combining as training dataset and take last segment as the validation dataset. Thus, the model can train five times and get five different accuracies then take the average of them as the final scores. In this way, I can make full use of the dataset and it is very helpful especially when the scale of the dataset is not large. It can also avoid overfitting since sometimes the decision tree classifier can easily be influenced by the noise and take the noise as the important content, leading to overfitting.

2.3 Convolutional Neural Network

To implement this, I first use LeNet. LeNet is the first CNN architecture to apply back propagation to practical applications. And it only contains seven layers, including two convolutional layers, two pooling layers and three fully connected layers. The detailed structure is presented in ***Figure 2.*** Basically, I use the same parameters as traditional LeNet by only changing the shape for inputs. Although LeNet is not the state of the arts these days, but due to the limitations of my hardware sources, the model of this kind of scale would be appropriate for me to use.

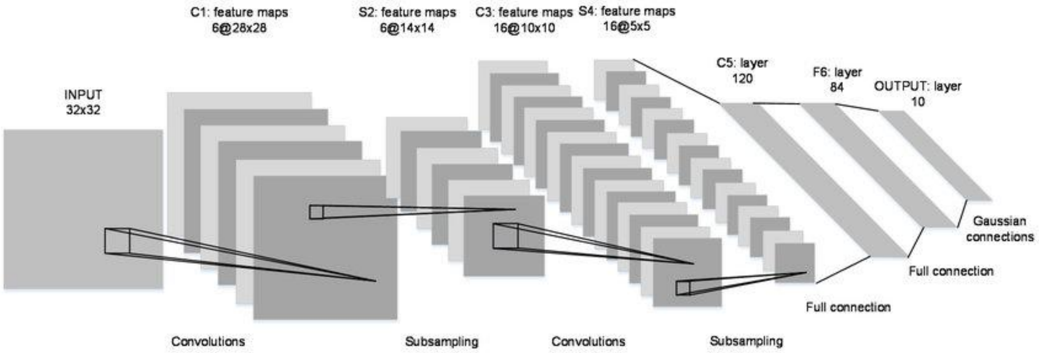


Figure 2. The structure and parameters for traditional LeNet [6]

Moreover, I also customize a CNN for training. This network contains six convolutional layers, three pooling layers and one fully connected layer in total. It first uses two convolutional layers, followed by a pooling layer then uses a batch normalization with a leakyRelu activation function. Then it repeats the above structure twice and end with a fully connected layer. I choose LeakyReLU() as the activation function. The difference between it and the Relu() function is that the former one assigns every negative value with a non-zero slope. And I use function BatchNorm2d() every time before using activation function to assure the performance for network is stable by normalizing the data when the scale of data is large.

The final parameters for customized CNN are determined as follows: The learning rate is 0.01, the momentum is 0.9 and the model was trained 10 epochs with batch size 5. During the training process, all the loss values are recorded, and metrics are printed out finally and 20% of the data set is used for model test.

3 Results and Discussion

3.1 Results for Decision Tree Classifier

Table 1. Performances on Raw data and Histogram Equalization data (HE)

	Raw	HE
Accuracy	31.8519%	24.4444%

Table 2. Performances on Raw data

Labels	Precision	Recall	F1-score
0	0.39	0.37	0.38
1	0.22	0.27	0.24
2	0.52	0.48	0.50
3	0.19	0.24	0.21
4	0.17	0.12	0.14
5	0.29	0.33	0.31
6	0.44	0.41	0.42

Table 3. Performances on Histogram Equalization data (HE)

Labels	Precision	Recall	F1-score
0	0.17	0.16	0.16
1	0.31	0.27	0.29
2	0.32	0.24	0.27
3	0.11	0.12	0.11
4	0.24	0.25	0.24
5	0.26	0.33	0.29
6	0.33	0.35	0.34

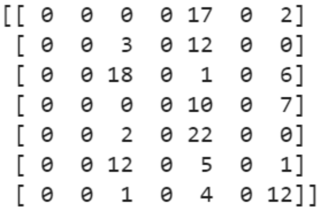
3.2 Results for LeNet

The training loss for LeNet is always very big for the first epoch then it decreases to about 1.948389 at the second epoch, and then decreases continually to 1.947919 at the end. And the accuracy for this method is 12.5926%.

3.3 Results for Customized Convolutional Neural Network

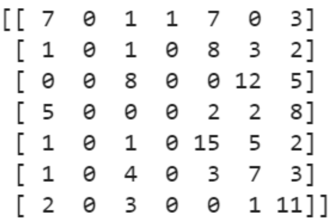
Table 4. Performances on Raw data and Histogram Equalization data (HE)

	Raw	HE
Accuracy	38.5185%	35.5556%
Training Loss	0.1798506	1.564982



```
[[ 0  0  0  0 17  0  2]
 [ 0  0  3  0 12  0  0]
 [ 0  0 18  0  1  0  6]
 [ 0  0  0  0 10  0  7]
 [ 0  0  2  0 22  0  0]
 [ 0  0 12  0  5  0  1]
 [ 0  0  1  0  4  0 12]]
```

Figure 3. Confusion Matrix for Raw Data



```
[[ 7  0  1  1  7  0  3]
 [ 1  0  1  0  8  3  2]
 [ 0  0  8  0  0 12  5]
 [ 5  0  0  0  2  2  8]
 [ 1  0  1  0 15  5  2]
 [ 1  0  4  0  3  7  3]
 [ 2  0  3  0  0  1 11]]
```

Figure 4. Confusion Matrix for Histogram Equalization data

3.4 Discussions

For results of Decision Tree Classifier, we can see that the accuracy for raw data is 31.8519% and the accuracy for histogram equalization data is 24.4444%, so the performance is better using raw data (***Table 1, Table 2, Table 3***). And as talked before, the reason I choose decision tree algorithm is that it is interpretable, and people can get to know the inner logic for classification. Thus, ***Figure 5*** is the trained tree plot with raw data and ***Figure 6*** is part of ***Figure 5*** since the original plot is too big to visualize. Also, ***Figure 7*** is the trained tree plot by histogram equalization data and ***Figure 8*** is part of ***Figure 7***. Furthermore, from ***Figure 6*** and ***Figure 8***, people can conclude about how does the classification conduct and what is the inside logic within it.

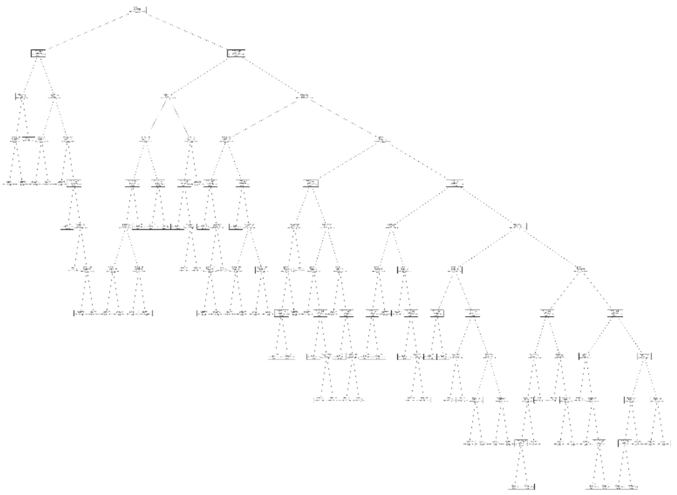


Figure 5. The plot of tree for Decision Tree Classifier model with Raw Data (overview)

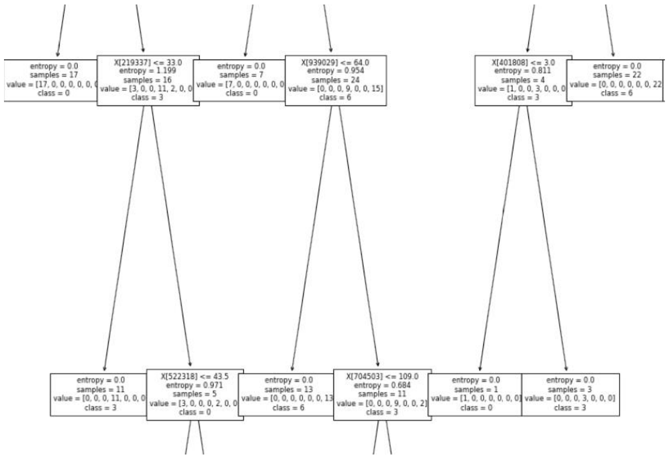


Figure 6. Partial plot of tree for Decision Tree Classifier model with Raw Data (zoom in)

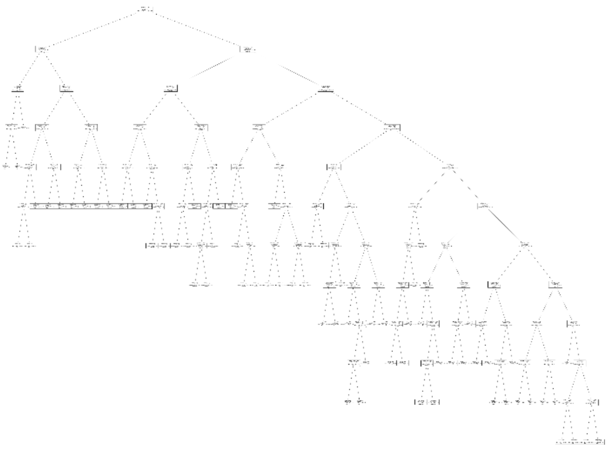


Figure 7. The plot of tree for Decision Tree Classifier model with HE (overview)

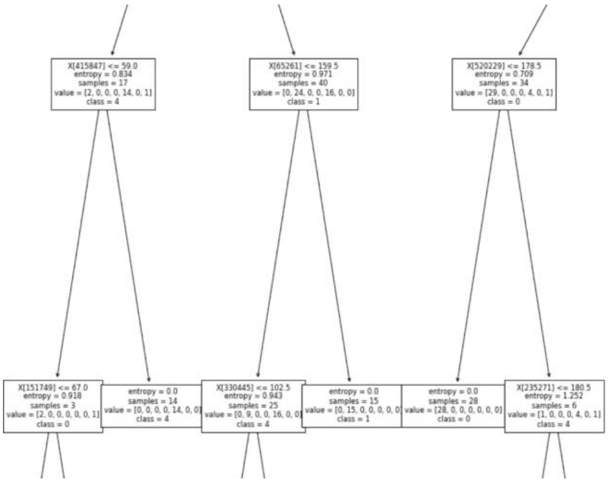


Figure 8. Partial plot of tree for Decision Tree Classifier model with HE (zoom in)

From the results showed in ***Table 4.***, I can see that the performance of customized convolutional neural network is better with raw data rather than the histogram equalization data, which is different from what I expect, since I think histogram equalization can strengthen the features. However, I also find out that the training loss for raw data is 1.958672 at the beginning and then decreases to 1.798506 lastly while the training loss for histogram equalization data is 1.920846 at the beginning and decreases to 1.564982 at the end. Combining the above two situations together, one possible reason that causes this could be overfitting and that is why although the training loss is smaller for histogram equalization data, the accuracy is higher with raw data. And from ***Figure 3,*** I can know that there are many images that are classified into the label 4, namely ‘neural’ emotion incorrectly and the second misleading class is label 6 namely ‘surprise’ emotion. From ***Figure 4***, there are many images are misclassified into label 5 and 6, namely ‘Sad’ and ‘Surprise’ emotions. And since the input scale is not very big, the testing set only contains 135 images and it is chosen randomly, which could cause imbalance in the testing set and influence the performance.

As for the results of LeNet, the accuracy is pretty low with traditional parameters, almost the same as the natural rate. I try to adjust and modify some of them to see if they can achieve better performance but find it hard to make a boost in the accuracy. So I only train it on the raw data since I suppose the performance for histogram equalization data would also be bad under the circumstance that the parameters are not suitable for the data, then it is meaningless to have the result.

So overall, among all the three methods, the customized convolutional neural network has the best performance. And in the dataset paper, for dataset SFEW, the accuracy is 43.71% for descriptor LPQ and 46.28% for descriptor PHOG[1], which is slightly higher than my research, but the performance is still nice compared to the natural possibility of 14.29% and the model is still effective in facial expression classification. One thing to note is that I only set the random state for split the training set, testing set and decision tree classifier, and I do not set the random state for the convolutional neural network method, thus the performance every time for training of two latter methods would be slightly different.

4 Conclusion and Future Work

4.1 Conclusions

In this research, the purpose is to classify the human face emotions from the input images into one of seven emotions. The inputs are original raw images or images with histogram equalization. I implement three algorithms of Decision Tree Classifier, LeNet and Customized Convolutional Neural Network and train Decision Tree Classifier and Customized Convolutional Neural Network on both kinds of input data, then compare the performances between them. And from the results, I can conclude that, the use of histogram equalization is useful in representing features since the accuracy is higher and the training loss is lower than the raw data. Among three methods, the Customized Convolutional Neural Network has best performance, but it is not explanatory while the Decision Tree Classifier also holds a similar good performance, but it is more interpretable, and people can get to know its logic by plotting the trained tree model.

4.2 Future Work

One possible and easy way for further research could be training the model in a broader dataset which contains more information to improve the performance and trying to use some other descriptors to extract the features.

From the technique's perspective, one way to extend the CNN model could be the FRR-CNN, namely feature redundancy-reduced convolutional neural network. Its convolutional kernels are reduced to be divergent by presenting a more discriminative mutual difference among feature maps of the same layer, resulting in generating less redundant features and having a more compact representation of an image. Moreover, the transformation-invariant pooling technique is used to extract representative features cross-transformations. [7]

Another considerable extension for future work is to use histogram of oriented gradients (HOG) and support vector machine (SVM) classifier for facial expressions' recognition.[8] And researchers can consider predicting human actions based on facial expressions for the next stage.

References

1. Dhall, A., Goecke, R., Lucey, S., & Gedeon, T. (2011, November). Static facial expressions in tough conditions: Data, evaluation protocol and benchmark. In 1st IEEE International Workshop on Benchmarking Facial Image Analysis Technologies BeFIT, ICCV2011.
2. Wang J. (2021). Classify Human Face Emotions from SFEW with three-layer neural network and compare performance between descriptors.
3. Gedeon T D, Turner H S. Explaining student grades predicted by a neural network[C]//Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan). IEEE, 1993, 1: 609-612.
4. K. Liu, M. Zhang and Z. Pan, "Facial Expression Recognition with CNN Ensemble," 2016 International Conference on Cyberworlds (CW), 2016, pp. 163-166, doi: 10.1109/CW.2016.34.
5. M. Shin, M. Kim and D. Kwon, "Baseline CNN structure analysis for facial expression recognition," 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2016, pp. 724-729, doi: 10.1109/ROMAN.2016.7745199.
6. Tra, Viet & Kim, Jaeyoung & Khan, Sheraz & Kim, Jongmyon. (2017). Bearing Fault Diagnosis under Variable Speed Using Convolutional Neural Networks and the Stochastic Diagonal Levenberg-Marquardt Algorithm. Sensors. 17. 2834. 10.3390/s17122834.
7. Xie, Siyue; Hu, Haifeng: 'Facial expression recognition with FRR-CNN', Electronics Letters, 2017, 53, (4), p. 235-237, DOI: 10.1049/el.2016.4328
   IET Digital Library, https://digital-library.theiet.org/content/journals/10.1049/el.2016.4328
8. Sajjad, M., Zahir, S., Ullah, A. et al. Human Behavior Understanding in Big Multimedia Data Using CNN based Facial Expression Recognition. Mobile Netw Appl 25, 1611–1621 (2020). https://doi.org/10.1007/s11036-019-01366-9