Convolutional Neural Networks for Facial Emotion Recognition with Image Processing
Final Project CSC 481
Tianyi (Jocelyn) Tan

Abstract
Human interact with each other mainly through speech but also through facial expression to display emotions. Facial expressions are important part of communication. Automatic recognition of facial emotions can be an essential component of human-machine interfaces, behavioral science and clinical practices.
This project aims to implement the complex Convolutional neural networks (CNN) proposed by previous studies to FER2013 dataset and also show the influences of image processing techniques on deep learning performance. Techniques including face detection, cropping, median filtering, Gaussian filtering, global contract normalization, histogram equalization, noise adding and combination between them are applied to the images in the data. The performance of different models using the same CNN architecture are compared. Visualization of the CNN model is also explored to have a deeper understanding of the black box.
In this project, modern CNN is implemented and obtain good performance. Proposed processing methods by literature have been successfully implemented to make comparison between methods for facial emotion recognition. CNN is also visualized to show the pattern of filters and feature maps. Based on the experiment results obtained, histogram equalization achieve the best improvement for the CNN performance on test set. Models trained by processed images tend to perform better when predicting unseen images and certain emotions such as Angry, Disgust and Neutral which are difficult emotions for the original model. Thus, by applying image processing with cautious and clear purpose, it might enhance the facial emotion recognition by CNN.

1. Introduction

Automatic facial expression recognition has been a challenging and important task in the computer vision field. The application of automatic facial expression can be applied in wide variety of areas such as human-computer interaction (HCI), lie detection and clinical psychology. Convolutional neural networks (CNN), one of the machine learning approaches, are commonly used to do facial recognition and emotion recognition. Data augmentation, an effective pre-processing technique including cropping, scaling, flipping to increase the size of the training set, is usually applied to the image input to improve the performance of CNN.
This project aims to implement the complex CNN model (mini Xception) proposed by previous studies to FER2013 dataset and also show the influences of image processing techniques on deep learning performance. Techniques including face detection, cropping, median filtering, Gaussian filtering, global contract normalization, histogram equalization, noise adding and combination between them are applied to the images in the data. The performance of different models using the same CNN architecture are compared. Visualization of the CNN model is also explored to have a deeper understanding of the black box.

2. Background

Convolutional neural networks are often utilized in emotion recognition and facial expression analysis. For the commonly used CNNs, they use fully connected layers as final layers which uses a huge number of parameters, e.g. VGG16 with 138,357,544 parameters (123,642,856 parameters for the last three fully connected layers, about 90% of total parameters). Modern

CNN architectures proposed by Arriaga et, al. [1] aims to eliminate completely the fully connected layers by combining the two of most successful experimental assumptions in CNNs: the use of residual modules[2] and depth-wise separable convolutions[3]. By using the modern CNN model and reduce the number of parameters to approximately 60,000, it provides a better generalization. More importantly, for this project, it enables us to re-train the model from scratch using images processed by different techniques at a faster speed (approximately 8 hours for each experiment).

To enhance the deep learning models, Pitaloka et, al. [4] propose data processing methods: resizing, face detection, cropping, add noises and data normalization consisting of local normalization, global contrast normalization and histogram equalization. By combining those techniques, it boosts performance of the conventional CNN model.

In the study of Hemalatha and Sumathi [5], face detection involves preprocessing the image, extracting facial features from the image and classify the image as face and non-face from the extracted features. The study uses median filter for noise removal in the pre-processing process and feedforward conventional CNN for classification.

In this project, image processing methods from the literature are combined and explored. The performance of models with different processing techniques are compared with each other and the performance of original model. By identifying the influences of different image processing techniques, this project can evaluate the necessity of image processing for modern CNN models and provide recommendations for enhancement.

3. Methods

a. Data and Features

The FER2013 dataset was obtained by Kaggle website, which consists of 35,888 48x48 pixel gray-scale images of faces. These facial emotions have been categorized as: 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, and 6=Neutral shown in the image below. The images were stored as strings in a csv file and conversion was needed for model training and image processing at later stage.

Figure 1 Sample Images for Different Emotions

For each emotion, there are different number of images in the dataset shown in the table below. Thus, when splitting training and testing sets for model training, stratified sampling is used. The imbalance of the class labels might have an influence on the model performance for each emotion.

Table 1 Class Distribution

Emotion	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Counts	4953	547	5121	8989	6077	4002	6198

b. Experiments

i. Haar Feature-based cascade classifiers

The Haar Feature-based cascade classifiers detect frontal face in an image with high accuracy. This detector can be used for real time system and has faster speed than other face detectors. The project uses an implementation from OpenCV [6]. It is a machine learning based approach where the cascade function is trained from images with faces and images without faces proposed by Viola et, al.[7]. Features are extracted from the model shown in the images below. They are extracted by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle. Features with minimum error rate are selected and final classifier is a weighted sum of all the weak classifiers using Adaboost. Instead of using all 6000 features selected, the concept of Cascade of Classifiers applies groups of features of different stages of the classifier one by one. The best two features are shown below.

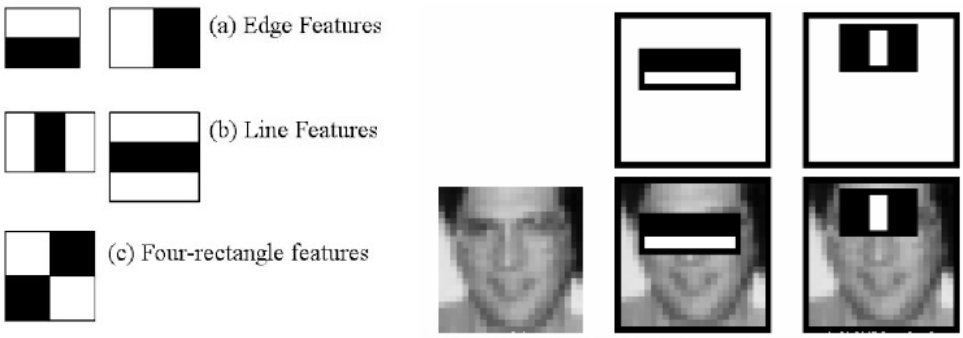


Figure 2 Haar Features (Left) Examples of Best Two Features: (The top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose.)

ii. CNN Model

This Mini Xception model was proposed by Arriaga et, al. [1] as the convolutional neural networks for emotion and gender classification in a real-time system with high accuracy in all 7 emotions. It takes bounded face as input and predicts probabilities of 7 emotions in the output layer.

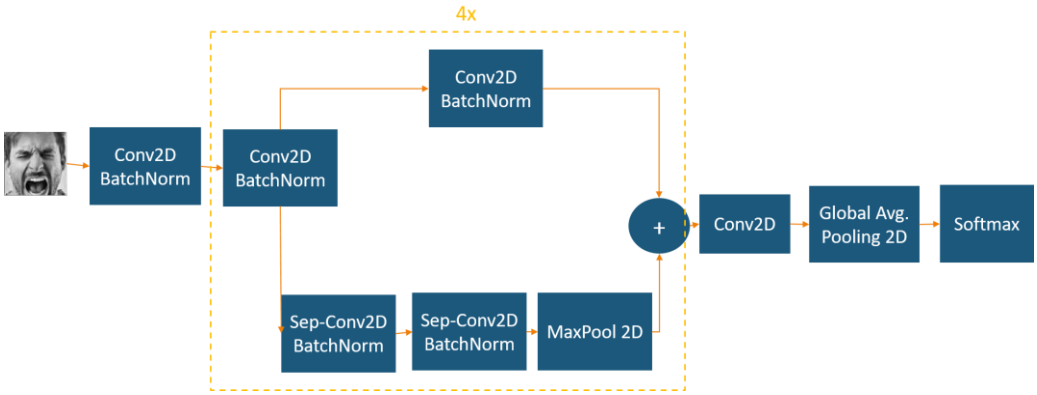


Figure 3 CNN Architecture

This architecture combines depth-wise separable convolutions and residual modules. Global Average Pooling is used to remove fully connected layers. Residual modules modify the optimal mapping between two subsequent layers. The learned features become the difference of the original feature map and the desired features. The depth-wise separable convolutions consist of two layers: depth-wise convolutions and point-wise convolutions, which separate the spatial cross-correlations from the channel cross correlations [8]. The model contains only 58,423 parameters which reduces the number of parameters hugely. The Mini Xception model includes 4 residual depth-wise separable convolutions followed by batch normalization operation [9] and a ReLU activation function. The final layer uses a global average pooling which sums out spatial information and force the model to extract global features from the input image [10] and a softmax activation function to obtain the classification results.

To further improve the performance of the CNN model, some other techniques are also used. Data augmentation which generates more data by rotation, crop, shifts, shear, zoom, flip, reflection, normalization and etc. It also uses kernel regularizer to apply penalty on layer parameters to avoid overfitting.

iii. CNN Visualization

CNNs are often considered as black-boxes and often the features obtained by the model are invisible. Visualization of layers might help to observe some features extracted by the model. In the paper of Arriaga et, al.[1], the authors utilize guided back-propagation visualization of their model using the images in the dataset. The visualization is shown in Figure 4.

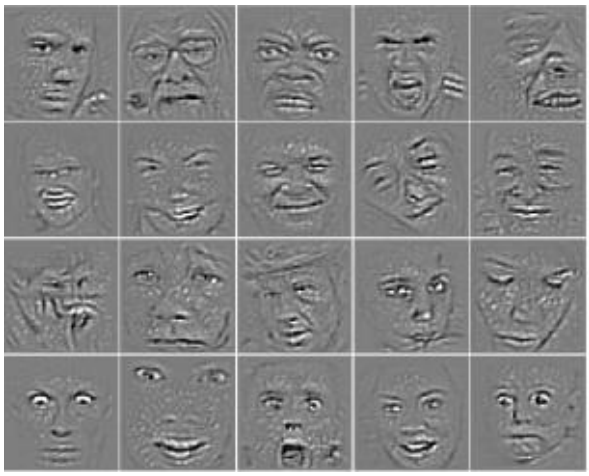


Figure 4 Guided Back-propagation Visualization[1]

This project uses simpler techniques to visualize output by each module of the complex model, filters of the first hidden layer and the feature maps of the fourth hidden layer using unseen image.

iv. CNN with Image Processing

Inspired by the work of Pitaloka et, al.[4], some image processing techniques are used to show the influences on model performance. Different techniques are first explored on a test image to see the impact on the prediction using the model trained by original images from the dataset. The same technique is then applied to all the images of the dataset as the pre-processing step. New models with the same architecture of mini Xception are trained on the processed images.

There are 5 different pre-processing steps:

(a) Normalized raw data (without other pre-processing steps)

The network training will have a faster convergence when the inputs are linearly transformed to have zero means and unit variance. Thus, the image data is transformed to the range from -1 to 1 for all the steps before feeding to the CNN network. This is used to train the original model and considered as baseline model.

face detection and cropping using Haar Feature-based cascade classifiers

Face detection utilizes the Haar algorithm and provides a bounding box on the facial region. The original image will then be cropped using the coordinates of the bounding box.

(b) global contrast normalization

Global contrast normalization (GCN) subtracts each pixel value of image by the mean and divide it by standard deviation. It aims to prevent images from a variety of contrast. The implementation of the normalization is different from the standard normalization by introducing positive regularization parameterλ. Dividing by the true standard deviation might amplify the sensor noise. The regularization parameter is used to bias the estimate of standard deviation. ε is used to constraint the denominator to be at least equal to some default value.

$$X'_{i,j,k} = s\frac{X_{i,j,k} - \bar{X}}{\max\left\{\varepsilon, \sqrt{\lambda + \frac{1}{3rc}\sum_{i=1}^{r}\sum_{j=1}^{c}\sum_{k=1}^{3}(X_{i,j,k} - \bar{X})^2}\right\}}$$

where $X_{i,j,k}$ *is the image of shape* $(i,j,k), \bar{X}$ *is the mean intensity of the image*

(c) Histogram equalization

Histogram equalization is commonly used to enhance the contrast of the image by spreading out the most frequent intensity values effectively on the histogram of an image.

(d) Adding noises

Salt-and-pepper noise which is a form of noise sometimes seen on images. By adding noises, it might help with data augmentation which might help the model to predict unseen data.

(e) Median and Gaussian Filters

Median and Gaussian filters of size 3x3 are applied. They are used to smooth an image to reduce image noise.

4. Results

4.1 CNN Visualization

4.1.1 Filters

In neural networks, the learned filter are simply weights. The weights have a spatial relationship to each other and plotting each filter as a two-dimensional image could see how different filters in different layers are trying to highlight or activate different parts of the image. It is easier to see the patterns in starting layers because as you go deeper the pattern captured might be sparser.

The filters (left below) are generated from the weights of the first two-dimensional convolutional layers and normalized to range 0-1 for easy visualization. The dark squares indicate small weights and the light squares represents large weights. 8 filters in the layer for the only 1 channel of each filter (gray image) are plotted below.

The filters (right below) are generated from the weights of the second two-dimensional convolutional layer. 8 filters in the layer and each of the 8 channels of each filter are plotted below.

We can observe from the filters of the first layer that some of the filters act as edge detector (first one), other are detecting a particular region of the image like its center region (the second one), and some act as background detectors.

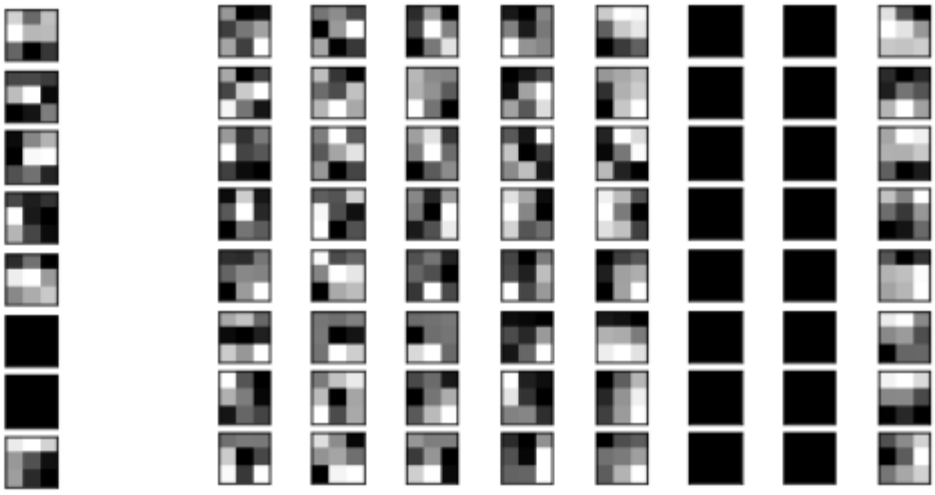


Figure 5 weights of the first two-dimensional convolution layer (left), weights of the second 2D convolution layer (right)

4.1.2 Feature Maps

The feature maps, also called activation maps, capture the results of applying the filters to input (both input images or from another feature map). Visualizing a feature map for a specific input image might help us understand what features are detected and kept in a certain feature map. General expectation would be that the feature map closer to the input detects finer details while the feature map close to the output capture more generalized features. An image with happy emotion is used for the demonstration, which is not included in the dataset.

Figure 6 Original Image with Happy Emotion for Visualization

Feature maps from each main block are collected in a single run and images of each are created. The number of each feature maps varies so the number of maps for each layer is capped at 4 for consistency and simplicity. The feature maps of 1st, 4th, 12th, 21st, 30th and 39th hidden layers (2x2 for each layer) are shown below. The pattern confirms the expectation.

Figure 7 Feature Maps Visualization

4.2 CNN with Image Processing

4.2.1 Exploration using Unseen Data

Using one test image which is not in the dataset, using different techniques or the combination of those techniques to observe the influence of the prediction.

The image below shows different types of image processing methods explored using the same image: Angry man. All the methods except adding noise are tried on the same image and run the model for prediction. Adding noise will be used for the training images in the dataset directly to train new models to see if there is any improvement of the model performance.

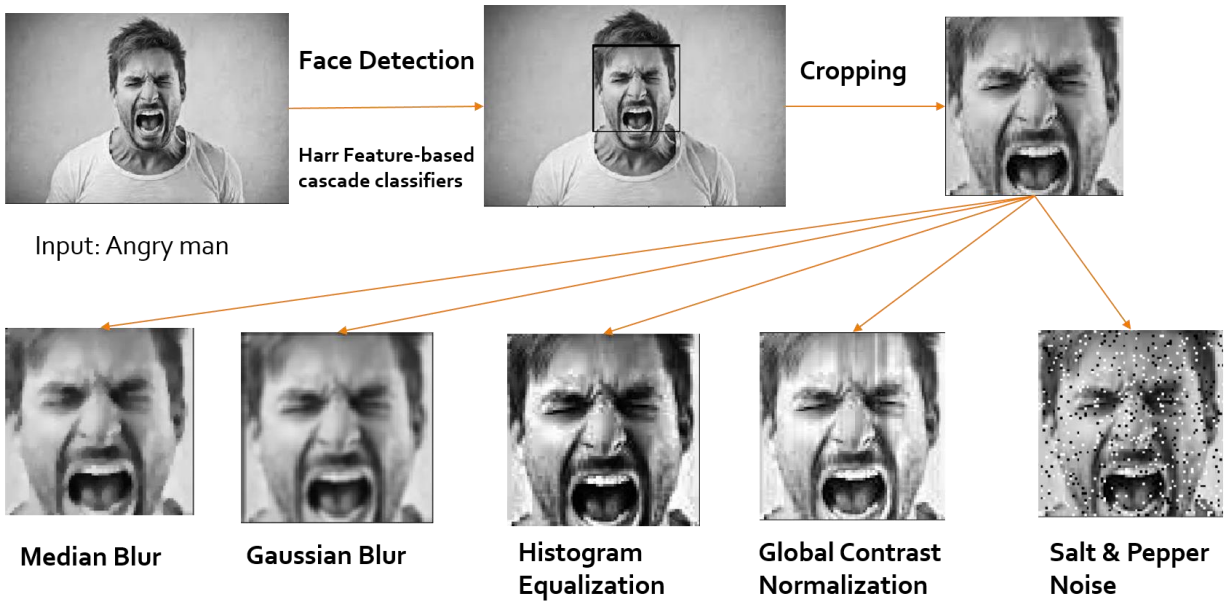


Figure 8 Demonstration of the Image Processing Techniques

The original image is labeled as scared instead of angry (the correct label) shown in Figure 9 below. The red bounding box is created by the face detected and the label is automatically provided by the prediction of the original model.

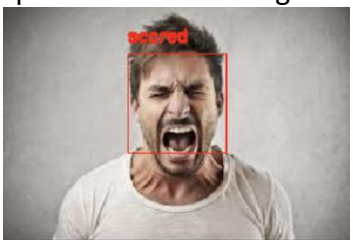


Figure 9 Original Image with Original Model Prediction

We can observe from the experiment results shown below that the original image is predicted to scare with high confidence from the model. The confidence (category probability provided by softmax activation map) of scare decreases using different image processing techniques. To explore the prediction performance of new models trained at later stage, the new models trained by images with noise and image normalized are applied to this image. Using the combination of GCN and Histogram Equalization (HE) gives the best results of this particular image. After using the combination of the processing methods on the original dataset to train a new model and applying the new model for prediction, this image is correctly predicted as angry with high confidence.

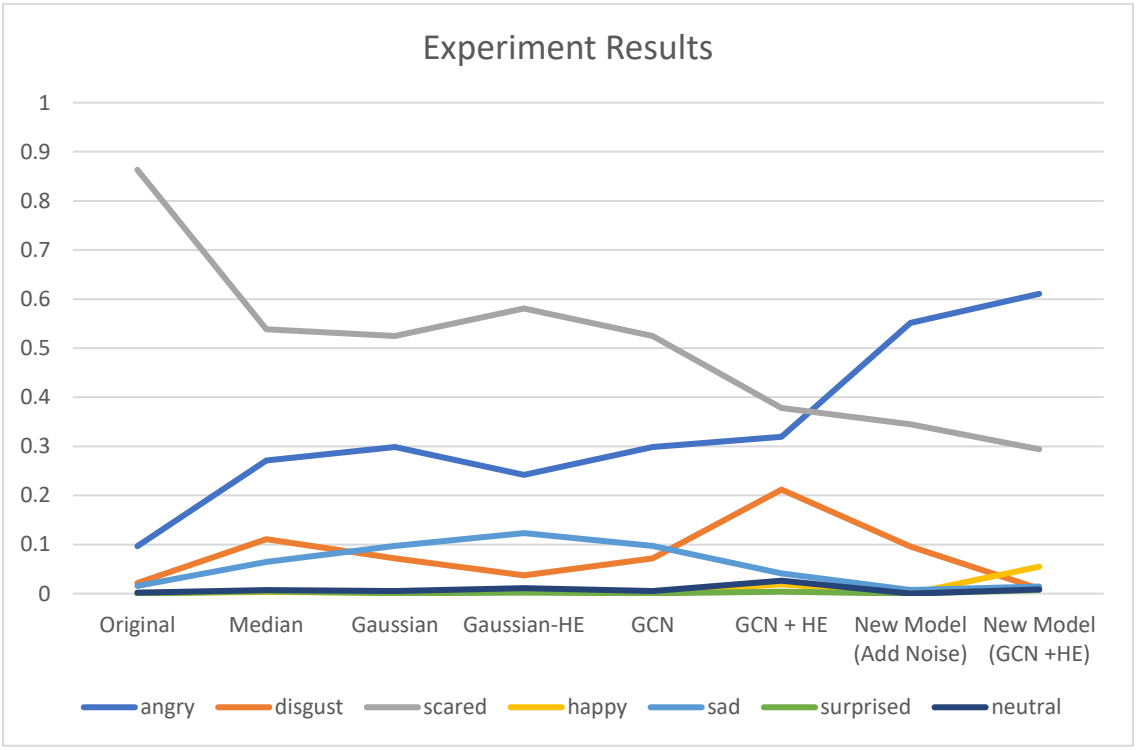


Figure 10 Line Graph of Experiment Results

This flow chart below shows the interactive process of using the best processing method identified by the exploration to re-train the model and obtain new model for prediction.

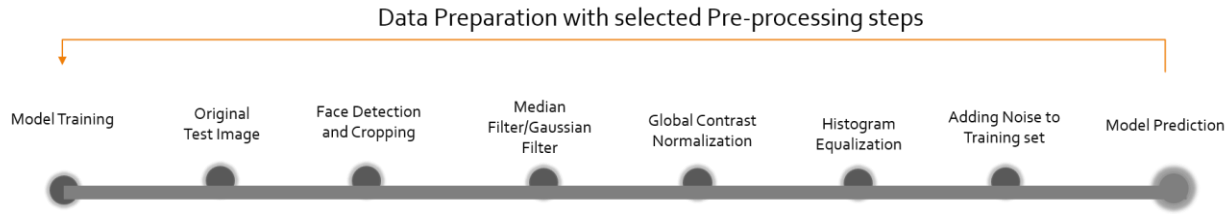


Figure 11 Flow Chart of Interactive Process

4.2.2 Model Performance using Selected Processing Techniques

From the exploration above, GCN, HE and combination of the two seems to improve the prediction on this unseen image. Adding noise is also a method to experiment. Blurring filters are not selected for further exploration due to the worse performance in the previous experiments.

Preprocessed Images in the dataset (shown as one image from each class):
For adding noise, the noise is only added to the training set so the images are different.

1) GCN Only:

2) Histogram Equalization:

3) GCN + Histogram Equalization:

4) Add Salt& Pepper Noise:

Figure 12 Sample Images for Each Emotion Class Processed by Different Techniques

Pre-processing steps and CNN are performed using python libraries such as OpenCV, Tensorflow and Sklearn. They are performed on the FER2013 dataset. The dataset is split into 80:20 for training and validation. There are four different processing step to experiment to observe the influence on CNN performance: 1) Global Contrast Normalization (GCN), 2) Histogram Equalization (HE), 3) GCN + HE, 4) Add Salt & Pepper Noise, as shown in the figure above (Figure 11). For each techniques, images are processed and fed into the CNN architecture (mini Xception) to obtain the performance on training and testing. The model will then be applied to the same group of images randomly picked from 20% the dataset as test set to compare the performance of different models.

Table 2 shows the accuracy, loss (categorical cross entropy) of training and validation set. The order is based on the validation loss. We can observe that original model still has the best performance in term of accuracy and loss of validation set. Histogram equalization provides the best performance among all different techniques.

Table 2 Model Performance on Training and Validation Set

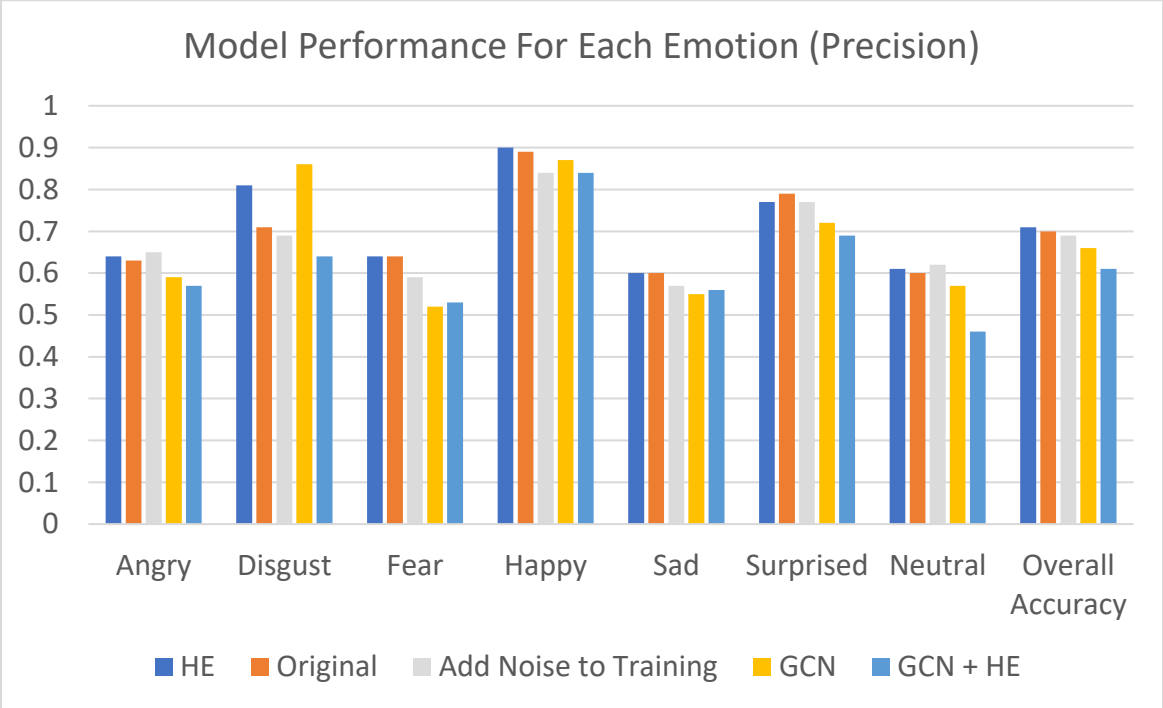
Model (+ Processing)	Training Accuracy	Training Loss	Valid. Accuracy	Valid. Loss
Mini Xception	0.6924	0.8272	0.6456	0.9716
+HE	0.6803	0.8571	0.6410	0.9744
+ Add Noise	0.6648	0.9018	0.6329	1.0007
+GCN	0.6933	0.8276	0.6265	1.0299
+GCN+HE	0.6728	0.8771	0.6240	1.0132

Table 3 shows the precision (true positives/ (true positives + false positives) of each class of different models on test set. The order is based on the overall accuracy. Histogram equalization has the best performance in this case. It provides improvement in identifying Disgust by 10% increase in precision and maintain high precision for other emotions. Adding salt-and-pepper noise can increase the precision of Angry and Neutral but will provide worse performance for other emotions.

Table 3 the influences of Precision for Processing Techniques for each class

Precision	Angry	Disgust	Fear	Happy	Sad	Surprised	Neutral	Overall Accuracy
HE	0.64	0.81	0.64	0.90	0.60	0.77	0.61	0.71
Original	0.63	0.71	0.64	0.89	0.60	0.79	0.60	0.70
Add Noise	0.65	0.69	0.59	0.84	0.57	0.77	0.62	0.69
GCN	0.59	0.86	0.52	0.87	0.55	0.72	0.57	0.66
GCN + HE	0.57	0.64	0.53	0.84	0.56	0.69	0.46	0.61
Support	991	109	1024	1798	1216	800	1240	7178

The grouped bar chart can clearly show the pattern of each emotion by each model. However, the imbalance of label distribution shown in the Support row in Table 3 might affect the performance results. For example, the precision of Happy is high might be due to the fact that it has more instances.



5. Conclusion

In this project, modern CNN is implemented and obtain good performance on the FER2013 dataset. Proposed processing methods by literature have been successfully implemented to make comparison between methods for facial emotion recognition. CNN is also visualized to show the pattern of filters and feature maps. Based on the experiment results obtained, histogram equalization achieve the best improvement for the CNN performance on test set. Models trained by processed images tend to perform better when predicting unseen images such as the particular angry test image. They also provide better precision for certain emotion such as

Angry, Disgust and Neutral which are all hard cases for model trained by raw images. Thus, although CNN is a black-box method, visualization might help find patterns and guided image processing might improve predictions in particular situations. Image processing should be applied with cautious and clear purpose.

Reference

[1] O. Arriaga, M. Valdenegro-Toro, and P. G. Ploger, "Real-time Convolutional Neural Networks for emotion and gender classification," *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, p. 6, 2019.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

[3] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *ArXiv*, vol. abs/1704.04861, 2017.

[4] D. Anggraeni Pitaloka, A. Wulandari, T. Basaruddin, and D. Y. Liliana, "Enhancing CNN with Preprocessing Stage in Automatic Emotion Recognition," *Procedia Computer Science*, vol. 116, pp. 523–529, Dec. 2017.

[5] G. Hemalatha and C. P. Sumathi, "Facial image detection with multiple filters and neural network for classification," in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, 2017, pp. 1414–1418.

[6] "OpenCV: Cascade Classifier." [Online]. Available: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html. [Accessed: 19-Nov-2019].

[7] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, vol. 57(2), pp. 137–154, 2004.

[8] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017, pp. 1800–1807.

[9] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *ArXiv*, vol. abs/1502.03167, 2015.

[10] M. Lin, Q. Chen, and S. Yan, "Network In Network," *ICLR 2013*, Mar. 2014.