RAINBOW UMBRELLA

Facial Expressions Detection With Partial Occlusion Report

Animesh Gandhi	1004980229
Muhammad Khan	1003978591
Jia Nian, Chiah	1004717390

Abstract

For this project, we will be exploring two different texture information extraction methods to extract facial features and how they affect the accuracy rate of the neural network in predicting facial expression on both occluded and non-occluded faces. The two methods used are Gabor filter and Non-negative Matrix Factorization (NMF). Based on our findings, we discover that Gabor filters only works well for non-occluded faces and it performs significantly worse in interpreting facial expression for occluded faces. While NMF requires more data to train, it shows a promising accuracy rate in predicting facial expressions on both occluded and non-occluded faces.

Introduction and Literature Review

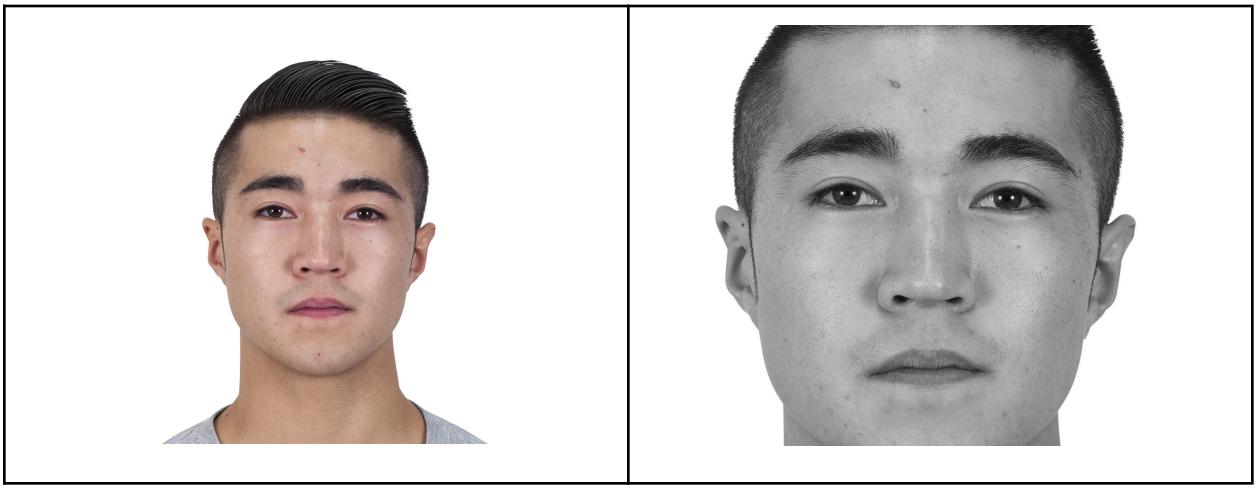
We humans evolved to detect facial expression by combining multiple facial features on a person's face. This includes our mouth, eyebrows, eye frown lines and cheek. However, with the ongoing COVID-19 pandemic, virtually everyone starts wearing a mask and covers the lower half of our face. This reduces the amount of visual features we humans can utilize significantly to detect facial expression accurately. Similar challenge is faced by most of the current computer vision technology as they are usually trained on a dataset with full, uncovered faces. Through this project, we hope to first compare a facial expression recognition model that is trained using partially covered faces and another with the full faces, and then uses the information derived from the comparison to propose a model that would achieve higher accuracy.

To achieve this goal, we are using the paper "An analysis of Facial Expression Recognition Under Partial Facial Image Occlusion" by Irene Kotsia as the backbone of our project. This paper explored three methods of classification which are Gabor wavelets texture information extraction, a supervised image decomposition method based on Discriminant Non-negative Matrix Factorization, and a shape-based method that exploits the geometrical displacement of certain facial features. The author uses the facial images with different parts of the face being occluded to measure the performance of each classification method. The main take away from this paper is that, regardless of classification methods, when the lower part of the face is covered (specifically mouth), one can expect the accuracy to drop significantly.

Methodology, Results, and Experiments

Texture Extraction via Gabor Filter

Before we can apply the Gabor filter on the input images, we have to crop the image to remove all the unnecessary data that would hinder the neural network performance. The cropped image is converted to grayscale for better computation for the Gabor filter.



After that, the cropped image will be downsample from using the image pyramid (**cv2.pyrDown()**). Note that because there will be information lost in the process of downsampling, we have to limit the number of times we downsample using the image pyramid. The cropped image is downsampled from 1280x960 to 80x60.

The Gabor kernel is calculated with varying orientations and frequencies, specifically:

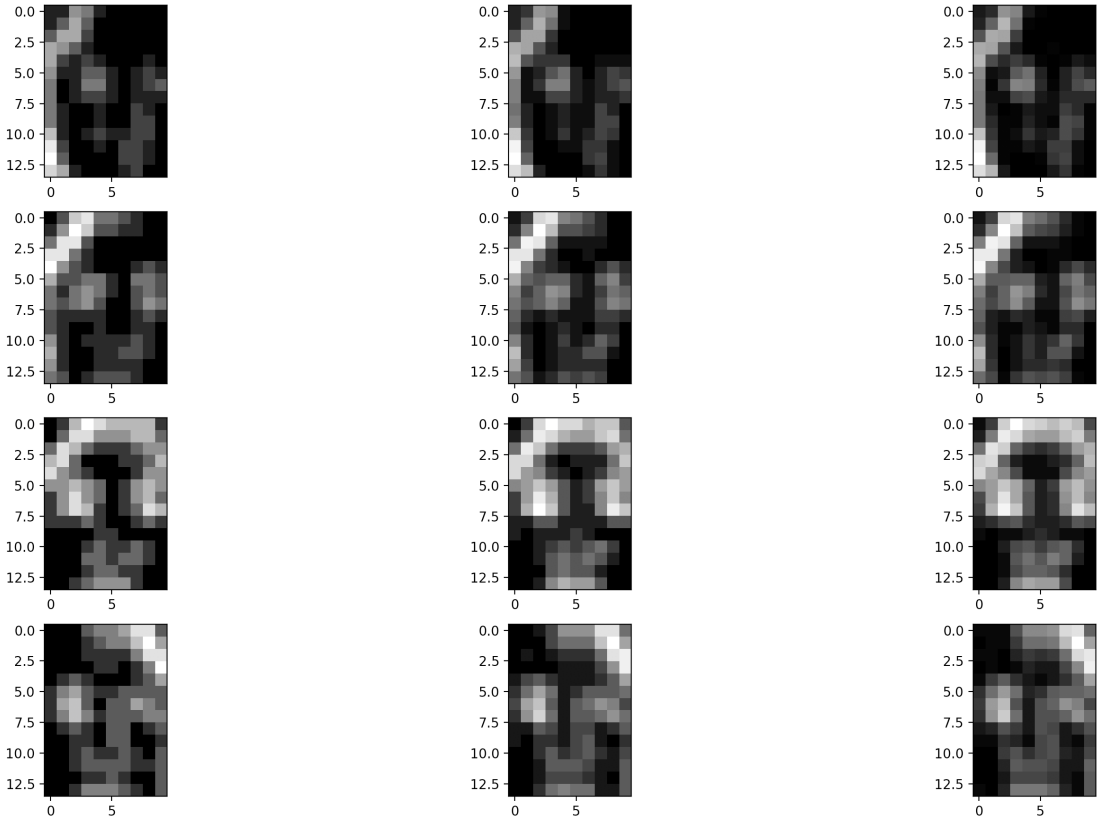
Orientation : 0, π/4, π/2, 3π/4

Frequency : 0, 2, 4

and results in 12 different filters. These 12 filters will be used to perform convolution on the 80x60 image, and the resulting image will be downsampled again to 14x10. All twelve 14x10 images will be converted and combined into a 1680x1 column

vector, which will be used to train our neural network. Each column vector for each input image is stored in a txt file separately.

Note that in the original paper that we referred to, the author downsampled the output to 20x10 instead of 14x10. This is because in our dataset, we notice that we can remove some columns as those columns are noises, and it can help speed up the training of neural networks due to less rows in the final vector. Also, the paper uses K-Nearest Neighbour as a classifier instead of neural networks.



Sample output of input images before converting them to column vector

Calculation of Gabor Filter and Fine Tuning Parameter

The Gabor filters can be expressed in the following equation:

$$\psi_{\mathbf{k}}(\mathbf{z}) = \frac{\|\mathbf{k}\|^2}{\sigma^2}\exp\left(-\frac{\|\mathbf{k}\|^2\|\mathbf{z}\|^2}{2\sigma^2}\right)\left(\exp(i\mathbf{k}^{\mathrm{T}}\mathbf{z}) - \exp\left(-\frac{\sigma^2}{2}\right)\right) \tag{1}$$

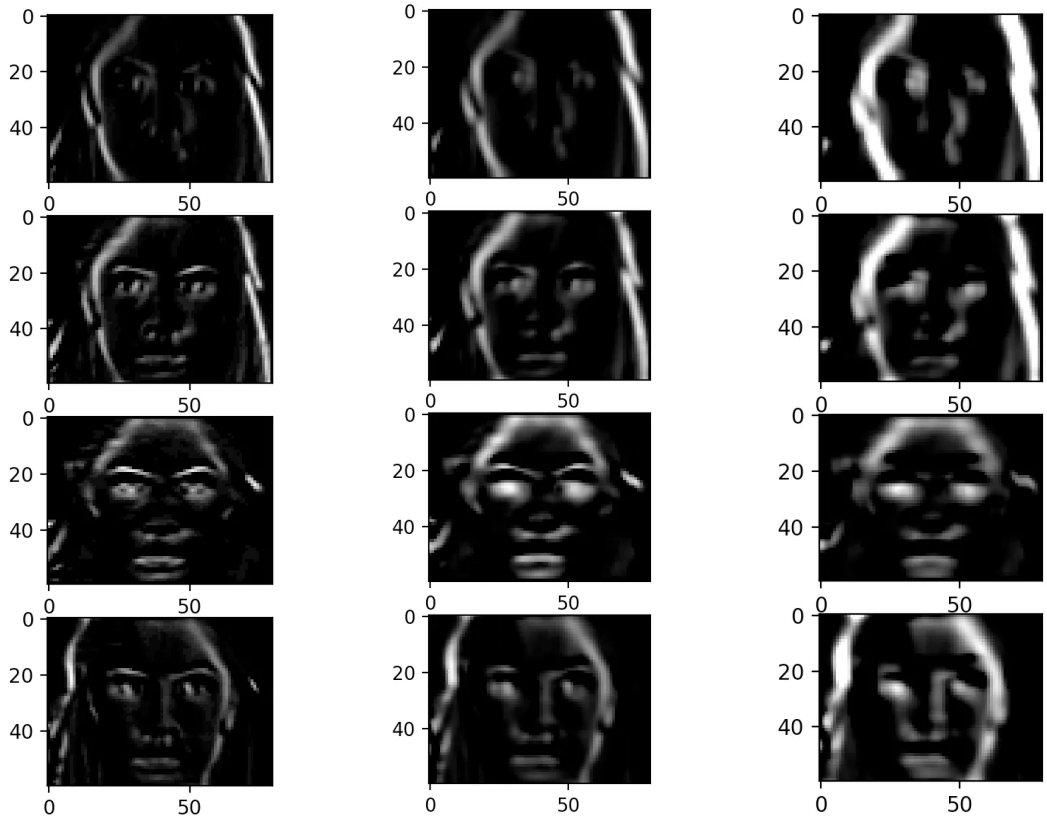
Where **z** = (x,y), σ is equal to 2π and **k** is the characteristic wave vector:

$$\mathbf{k} = \left[k_v \cos\varphi_\mu, k_v \sin\varphi_\mu\right]^{\mathrm{T}} \tag{2}$$

With

$$k_v = 2^{-\frac{v+2}{2}}\pi, \varphi_\mu = \mu\frac{\pi}{8} \tag{3}$$

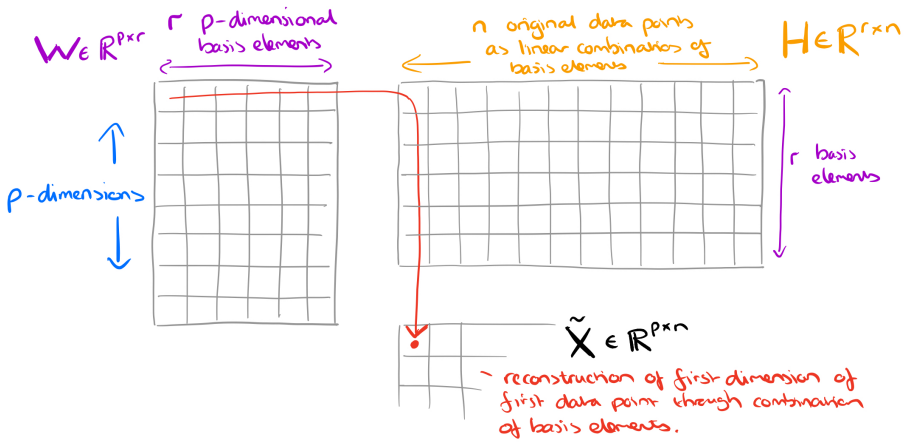
In our project, the v and μ in equation (3) is set to the values of [0, 2, 4] and [0, π/4, π/2, 3π/4] respectively to calculate the wavelength **k**. The **k** value will be fed to the **cv2.getGaborKernel()** function alongside with the a filter size of 3x3. Such filter size is chosen as we discovered that larger filter size will make the input image much blurrier, which makes it hard to extract meaningful information. As for the value choices for both v and μ, they are chosen as they were suggestions from the paper we read to maximize the accuracy (Kotsia 2007).



Sample output of different filter sizes (left to right): 3x3, 5x5, 7x7

Texture Extraction via Non-negative Matrix Factorization (NMF)

Non-Negative Matrix Factorization is a technique where, given an N by M matrix **X**, it is "factorized" into two matrices **H** and **W**, with dimensions N by R and R by M respectively, where the basis dimension R is picked by the user. These matrices are selected such that **X** is approximately equal to **HW**.



NMF based techniques excel at converting matrices to a lower dimension. Since images are usually represented by non-negative matrices, this makes these methods good candidates for feature extraction., converting images to relatively sparse representations of themselves. According to I. Buciu et. al., NMF is a method that unlike other methods like Independent Component Analysis, maintains the interdependence of components in the image. While independent component analysis performs well in nature, where items are much more likely to be independent, in faces we expect a high degree of dependence. When comparing to Gabor filters, it is noted that while they can be very effective, they are reliant on a high degree of manual tuning of hyper parameters for a "complete basis for image representation".

In our implementation, we first cropped and downsampled the images following the same method described above in the Gabor Filters section. Following this an image was selected to provide the basis matrix **H** that all images would be described with, and all images had their respective **W**s extracted. On our end, it was done with the python library sklearn.decomposition, NMF. On the theoretical end, this is done by minimizing the frobenius norm of ||**X**-**HW**||^2**,** where initialization is done by singular value decomposition.

Occlusion

Due to the lack of access to an extensive database with masked faces, we were forced to generate our own occluded faces. To this end, we used face & eye detection based on HAAR cascades, and used this information to estimate the location of the mouth, resulting in 3 types of images: no occlusion, mouth-only occlusion, and full-mask occlusion.

(left to right) sample no occlusion, mouth-only occlusion, full-mask occlusion

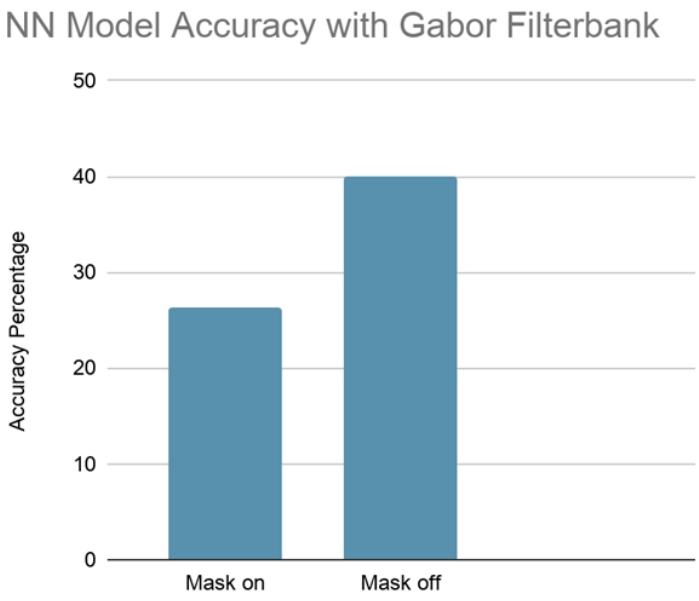
Classification with Feed Forward Neural Network via Gabor Filterbank

Two models were trained using Gabor filterbank convolutions. (1) images of complete faces and (2) images with mouth region occluded to simulate facial masks/coverings.

(1) 12 Gabor filter convolutions with corresponding face image were downsampled to 14x10. The downsampled results were further flattened and added together to form a single vector of size 1680x1. This vector was the input to a feed-forward neural network with 4 layers (2 hidden layers of size 60 each) and 5 classification nodes in the last layer for each expression (neutral, angry, fear, happy open mouth, happy closed mouth). Total number of parameters is 30,240,000, the dataset size is 1295, and the learning rate is 10%. The dataset was not completely balanced, with neutral-expression images making up around half of the dataset. As a result of the imbalance, loss function of Cross Entropy loss was used with a weight of 0.2 on neutral-expression classification and 1 on the other classification outputs.

(2) Same procedure as above. A significant portion around the mouth region of gabor filterbank convolutions with image had value 0. Number of inputs remained the same as 1680.

The model showed reduced accuracy when trained on faces with the mouth region excluded. It had an average accuracy of 26.4% on test sets of occluded faces and 40% on non-occluded faces. Given enough data, we believe the model can be trained to be more accurate.



Classification with Feed-forward Neural Network via Non-negative Matrix Factorization (NMF)

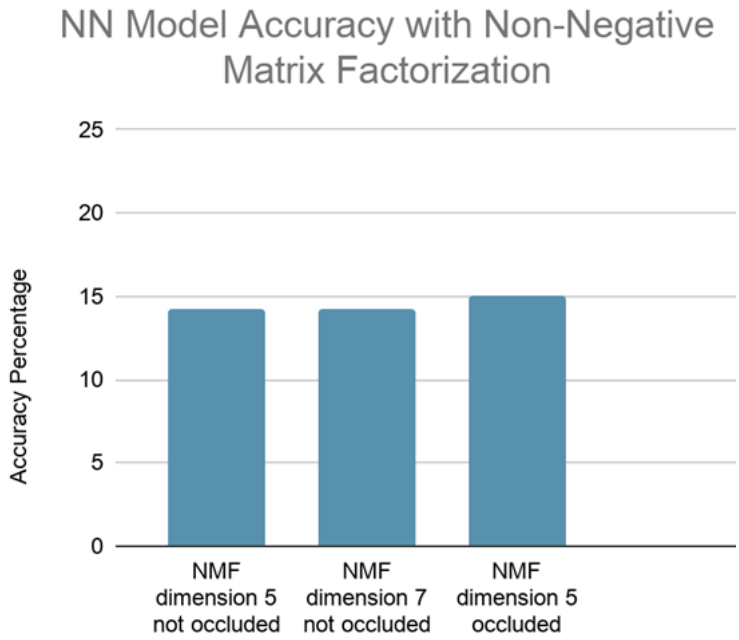
Two models were trained using Non-negative Matrix Factorization. (1) images of complete faces with NMF dimension 5, (2) images of complete faces with NMF dimension 7, and (3) images with mouth region occluded to simulate facial masks/coverings (dimension 5).

(1) NMF with dimension 5 was performed on every image to produce NMF matrix. Matrix is flattened to a 300x1 vector and passed through the feed-forward neural network. The NN has 4 layers (2 hidden layers of size 60 each) and 5 classification nodes for each expression. The model had a total of 5,400,000 parameters, dataset of size 1207, and learning rate of 10%. Cross entropy loss was used with a weight of 0.2 on neutral-expression classification due to the imbalance in the dataset.

(2) Same procedure as above but with NMF dimension 7. This resulted in a 420x1 vector. Total parameters were 7,560,000, dataset of size 1207, and learning rate of 10%.

(3) The mouth region of faces within our dataset were blacked out and the resulting images were passed through the NMF algorithm with dimension 5. This resulted in a vector of 300x1 trained on a model with 5,400,000, a dataset of size 1187, and learning rate of 10%.

The model showed equal accuracy on all three versions of NMF of around 15% on test sets. Given enough data, we believe accuracy would improve across all three versions.



Conclusion

For this project, our aim was to train a Classifier to be able to classify facial expression in the presence of masks. For our first step, we used face/eye detectors using Haar Cascades to extract faces from images. Next, we processed these images using Gabor filterbank and Non-negative Matrix Factorization for both occluded images and non-occluded images. These images were then trained on their own individual Neural Networks with feature vectors extracted using the above two techniques. For Gabor filterbank convolutions, we found that we were able to train our non-occluded images to a higher accuracy than our occluded images (40% v. 26.4%). For NMF, we found no difference among occluded, non-occluded dim 5, or non-occluded dim 7. Suggesting that we require more data to train our model for NMF feature extraction.

Author Contribution

Animesh Gandhi:

- Feature extraction with NMF, tweaking dimensions
- Building scripts to perform automated occlusion based on estimated mouth position derived from face & eye detection

Muhammad Khan:

- Face/eye detection implementation using Haar Cascades
- Structuring data for training and testing purposes
- Building and training Neural network model with appropriate hyperparameters
- Testing NN

Jia Nian, Chiah:

Perform preprocessing on the dataset and texture extraction via Gabor filter. This involves steps such as

- Noise removal from the input images
- Image downsampling using image pyramids
- Convolution of Gabor filter on input images and fine tuning the parameters of the Gabor kernel

Also, helps to train the neural networks

Reference

1. Kotsia, Buciu, Pitas (2007). *Facial expression recognition under partial facial image occlusion.* https://journals-scholarsportal-info.myaccess.library.utoronto.ca/pdf/02628856/v26i0007/1052_aaoferupfio.xml
2. Buciu, Pitas (2006). *NMF, LNMF, and DNMF modeling of neural receptive fields involved in human facial expression perception* https://www.sciencedirect.com/science/article/pii/S104732030600040X
3. C. Boutsidis, E. Gallopoulos (2007). *SVD based initialization : a head start for nonnegative matrix factorization* http://scgroup.hpclab.ceid.upatras.gr/faculty/stratis/Papers/HPCLAB020107.pdf
4. Tipples, J., Atkinson, A. P., & Young, A. W. (2002). The eyebrow frown: A salient social signal. https://psycnet.apa.org/record/2002-18009-008
5. Tian, Ying-Li, Takeo Kanade, and Jeffrey F. Cohn. "Facial expression analysis." *Handbook of face recognition*. Springer, New York, NY, 2005. 247-275. https://www.cs.cmu.edu/~cga/behavior/FEA-Bookchapter.pdf
6. Face Images Database. https://chicagofaces.org
7. https://mlexplained.com/2017/12/28/a-practical-introduction-to-nmf-nonnegative-matrix-factorization/
8. *https://blog.acolyer.org/2019/02/18/the-why-and-how-of-nonnegative-matrix-factorization/*