ECE324 Final Report

MojiMi: Categorizing Facial Emotion for Emoji Placement

November 11, 2019

Word Count: 1992
Overlimit Penalty: 0

Eric Li (1004015852)
Kailin Hong (1003870876)

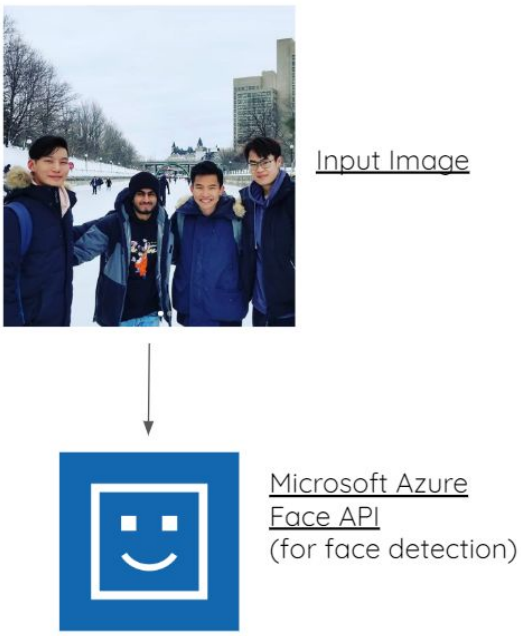
1.0 Introduction

Given an input photo, the goal of the project is twofold: to first detect the location of human faces and to recognize corresponding happy, sad, surprised, neutral, and angry emotions of detected faces. The ultimate output will be the input photo with emojis matching the detected emotions over the locations of detected faces. Based on similar emotion projects[1], the validation accuracy goal for this project is 60%.

For facial detection and recognition operations, machine learning is an appropriate tool. Applied in industry, such as pedestrian detection for autonomous vehicles[2], machine learning (neural networks specifically) can apply scanning kernels to detect faces at any location within a picture and iteratively adjust parameters used to evaluate facial features for emotion classification.

Motivation for this project is based on trends in social media and anonymity. With the final output image having emojis covering people's faces, the project can be applied as a photo filter- a now quintessential feature in social media- or as an anonymity tool to hide identity in the world of growing privacy concerns.

2.0 Illustration



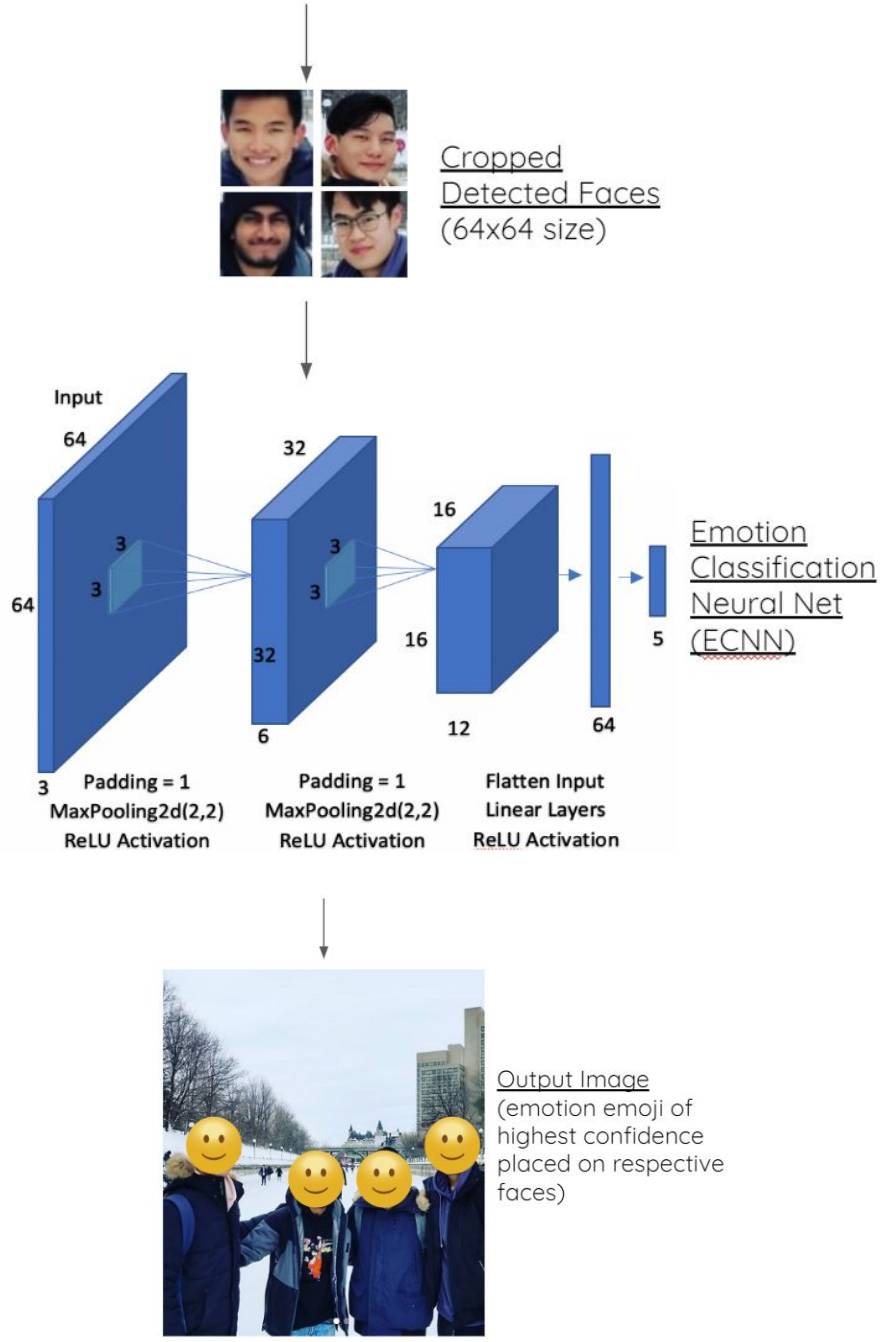


Figure 2.1 The overall illustration of the program, including the input image, Azure Face API, detected faces, Emotion Classification Neural Net (ECNN) model, and final output image

3.0 Background of Related Work

In a paper submitted to the EMotiW 2016 Challenge[1], a hybrid CNN-RNN and C3D network capable of classifying emotions in video is presented with 59.02% accuracy. Face features are extracted by using a pretrained CNN network- the VGG-16 Face Model, which is tuned and retrained to more suitable for solving the challenge. The features are inputted in each time step to the RNN. The CNN-RNN and 3D CNN network, which extract facial features and keep track of time steps, are combined together to increase accuracy. With the paper showing good results through applying the CNN to extract facial expression, a CNN is chosen as the main identifier of the MojiMi project. Since the MojiMi requires static images, a 2D CNN is considered. Moreover, from the EMotiW paper's application of transfer learning, a similar VGG transfer learning model is explored and tested to MojiMi's specific dataset and setting.

4.0 Dataset and Data Processing

Flickr Faces High Quality (FFHD)[2] and AffectNet[3] datasets are used in this project. For images in the FFHD dataset (*Figure 4.1*), Azure Face Detection API is used to obtain emotion labels and coordinates of detected faces. Since happy and neutral labelled faces significantly out-number faces labelled with the other three emotions, data augmentations (noise adding, brightness adjusting, and horizontal flipping) are performed. Distributions after data augmentation is shown in *Figure 4.2*.

Figure 4.1: Flickr Faces HQ dataset Example Images

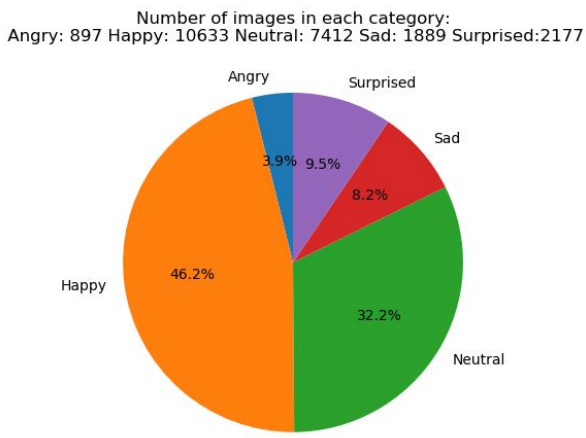


Figure 4.2. Emotion label distribution of collected and augmented Flickr Dataset data.

With emotion labels manually labelled by a team of researchers and coordinates for detected faces given, AffectNet is introduced to create a more balanced dataset, forming the majority of the surprised, sad, and angry image categories. The resulting distribution is shown in *Figure 4.3*. Images from both datasets are cropped to span detected faces and resized to 64x64 pixels. AffectNet examples can be seen in *Figure 4.4*.

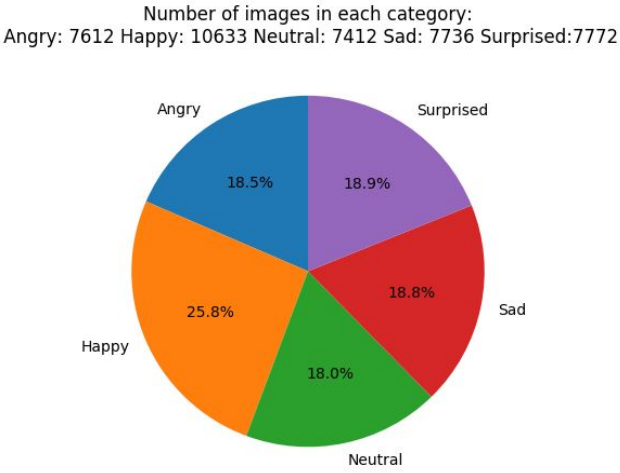


Figure 4.3. Emotion distribution of all collected and preprocessed data.

Figure 4.4: Image examples and accompanying labels from the AffectNet dataset

Selected in equal amounts from each emotion category in the dataset, these images are transformed into tensors, normalized within range [0,1] and split into 70% training, 20% validation, and 10% test data respectively. Data processing procedures are illustrated in *Figure 4.5* and *Figure 4.6*.

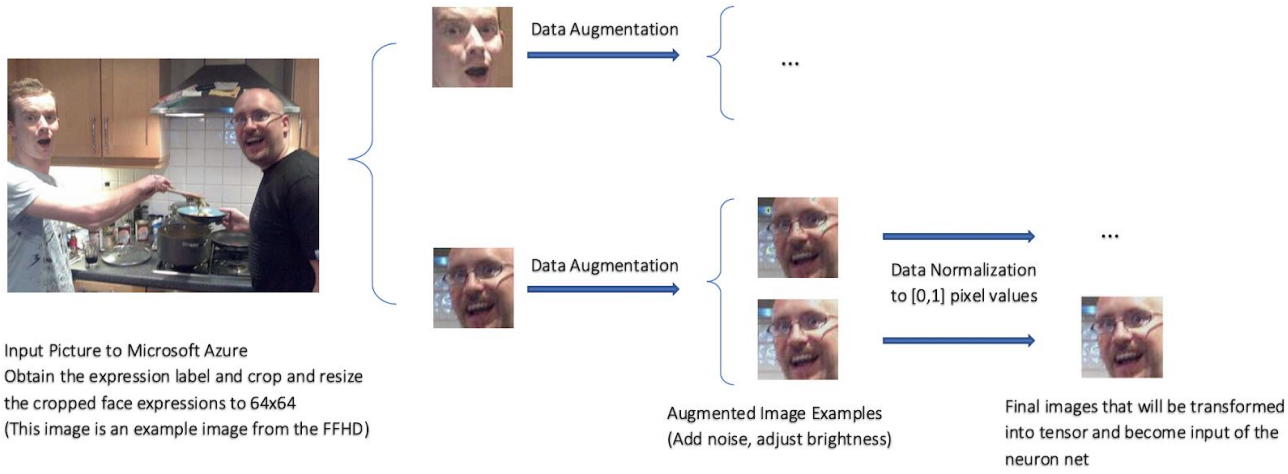
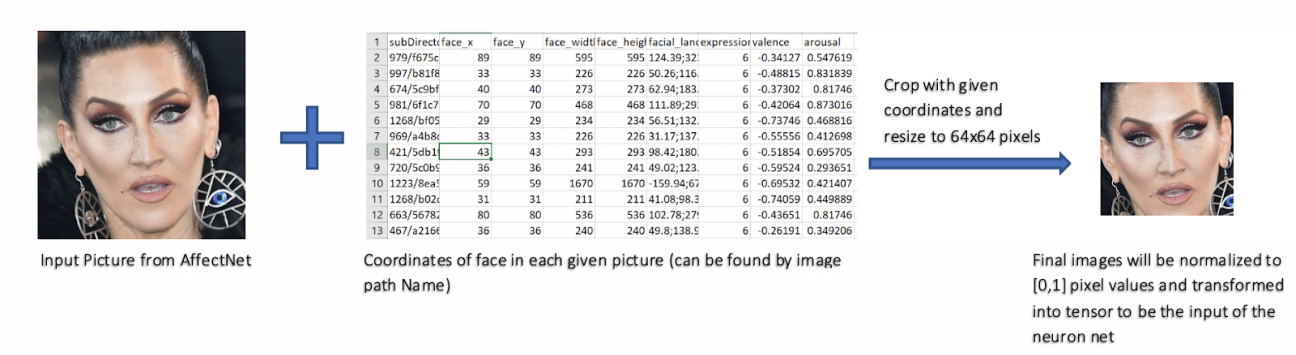


Figure 4.5. Flickr Dataset Images Preprocessing Procedure



	subDirect	face_x	face_y	face_widtl	face_heigl	facial_lanc	expressior	valence	arousal
2	979/f675c	89	89	595	595	124.39;32	6	-0.34127	0.547619
3	997/b81f8	33	33	226	226	50.26;116.	6	-0.48815	0.831839
4	674/5c9bf	40	40	273	273	62.94;183.	6	-0.37302	0.81746
5	981/6f1c7	70	70	468	468	111.89;29.	6	-0.42064	0.873016
6	1268/bf05	29	29	234	234	56.51;132.	6	-0.73746	0.468816
7	969/a4b8c	33	33	226	226	31.17;137.	6	-0.55556	0.412698
8	421/5db1	43	43	293	293	98.42;180.	6	-0.51854	0.695705
9	720/5c0b5	36	36	241	241	49.02;123.	6	-0.59524	0.293651
10	1223/8ea5	59	59	1670	1670	-159.94;67	6	-0.69532	0.421407
11	1268/b02c	31	31	211	211	41.08;98.3	6	-0.74059	0.449889
12	663/56782	80	80	536	536	102.78;27!	6	-0.43651	0.81746
13	467/a216(	36	36	240	240	49.8;138.9	6	-0.26191	0.349206

Figure 4.6. AffectNet Dataset Images Preprocessing Procedure

5.0 Architecture

The final ECNN architecture contains two convolution layers followed by two fully connected layers(1 hidden and 1 output). The input is a batched RGB image tensor of shape [batch_Size,3, 64, 64]. Having six and twelve kernels respectively, the first ans second convolution layer apply one pixel padding and have kernels of size 3x3. The ReLU activation function and max pooling operations (2x2 size with stride 2) are applied after each convolution layer. The output tensor of the convolution layers is flattened into shape [batch_size, 16*16*12] and inputted into the fully connected hidden layer, which has 64 neurons. A ReLU activation function follows the hidden layer before the 5 neuron output layer, which generates an output of shape [batch_size,5] to represent the probability of the image being in each of the 5 emotion classes. For adjusting parameters, the CrossEntropy Loss function and Adam optimizer are applied. A softmax function is applied to the output to obtain the probability distribution for the five observed emotions and the final prediction. The final ECNN model is illustrated on *Figure 2.1*.

The ECNN model presented has the highest accuracy and the lowest loss found among previous architecture designs.

6.0 Baseline Model

The baseline model used as a comparison benchmark for ECNN is a two layered Multi-Layered-Perceptron which receives a flattened RGB image tensor of shape [batch_size,3*64*64]. The first fully connected hidden layer has 3*16*16 neurons followed by a ReLU activation function. The model's output layer has 5 neurons and has its 5 outputs passed into a softmax function to obtain the final prediction probabilities. The baseline model is also trained with the CrossEntropy Loss function and Adam optimizer. The architecture of the baseline is shown in *Figure 6.1*.

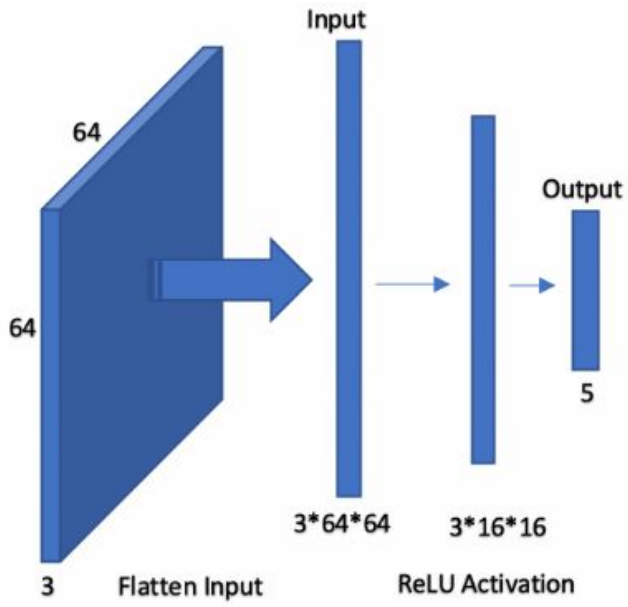


Figure 6.1: The architecture of the Baseline Model

7.0 Result

7.1 Quantitative Results

To measure the overall performance of the ECNN in assigning correct emotions to photos of faces, test and validation accuracies and training and validation losses are considered. From *Figure 7.1.1*, it is seen that the ECNN architecture overfits. Consequently, early stopping is applied at epoch 15 to obtain our final ECNN model parameters before validation accuracy starts decreasing and overfitting occurs. The final model achieves a test accuracy of 74.07%, which aligns with the validation accuracy.

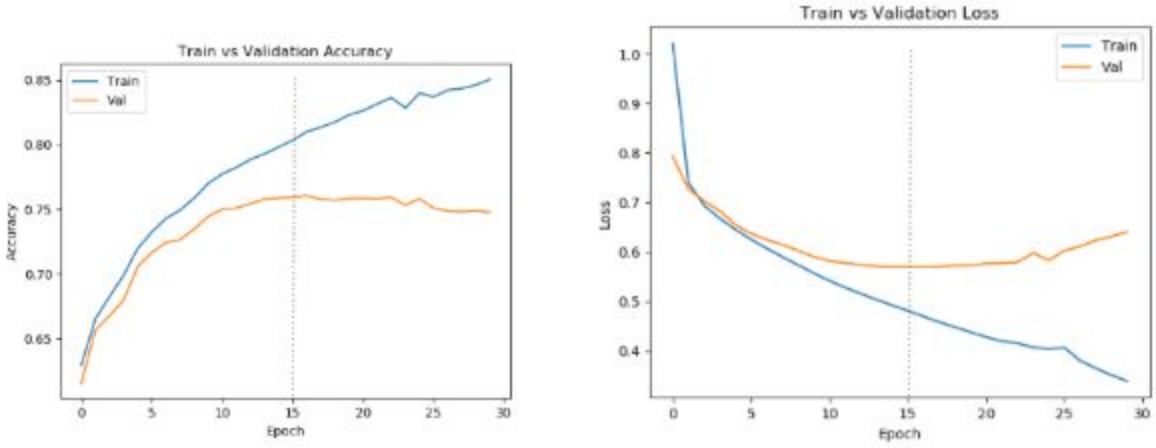


Figure 7.1.1: ECNN architecture's training (accuracy and loss) curves, with blue representing training curves and orange representing validation curves

To observe and evaluate the correctness of labelling in each respective category, the confusion matrix of the final ECNN model is considered. From *Figure 7.1.2*, it is seen that happy and neutral photos are most often labelled correctly, while angry, sad, and surprised photos are often confused among.

	Angry	Happy	Neutral	Sad	Surprised
Angry	[[882	0	1	250	170]
Happy	[0	1343	83	1	3]
Neutral	[3	136	1393	1	2]
Sad	[481	3	5	1079	439]
Surprised	[116	0	0	151	868]]

Figure 7.1.2: Confusion matrix produced by the final ECNN model

From the 69.1% test accuracy and curves of the baseline (*Figure 7.1.3*), it is concluded that the ECNN slightly outperforms the baseline model with about 5% in accuracy increase.

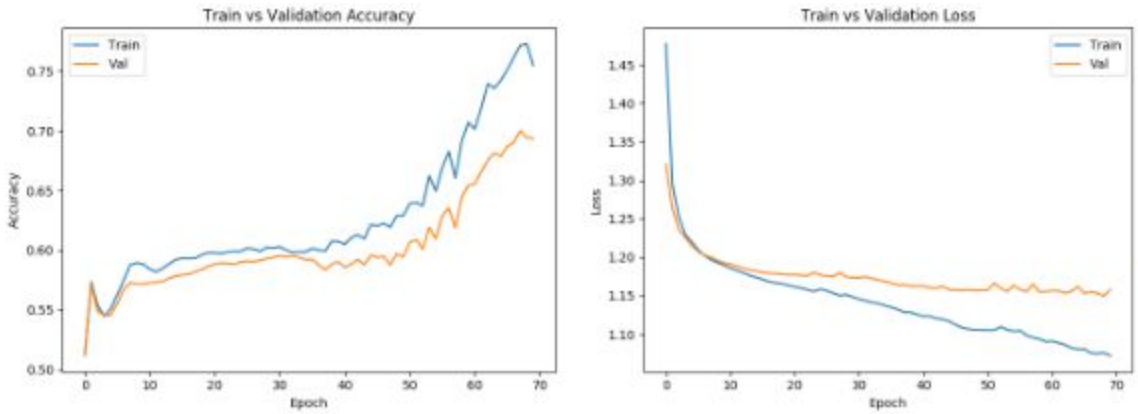


Figure 7.1.3: Baseline Model training (accuracy and loss) curves, with blue representing training curves and orange representing validation curves

7.2 Qualitative Results

Generally, the program can perform well in relation to happy and neutral facial expressions. Shown in *Figure 7.2.1,* with an input image taken from Instagram, the program crops the faces detected by Azure and predicts the corresponding emotions with the ECNN model. Emotion emojis are then placed onto the input image for the final output.

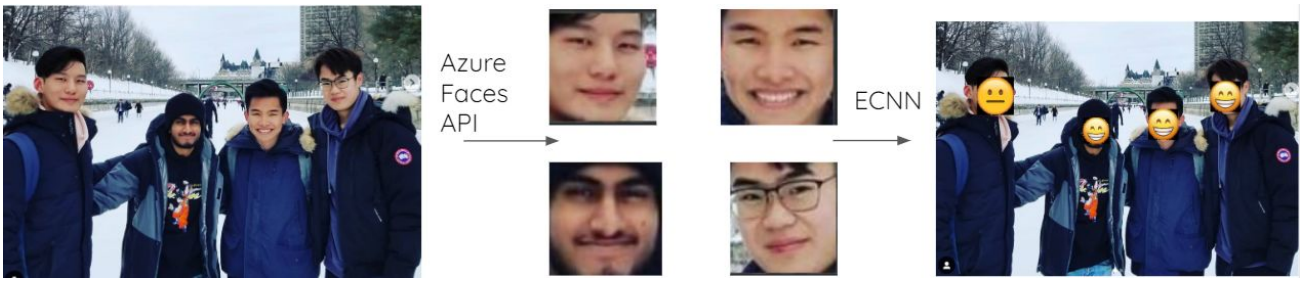


Figure 7.2.1: The input, cropped, and output images over the program's applied Azure Face API and ECNN model.

It is important to note several limitations. Since face detection is performed by the Azure Face Detection API, faces undetectable by the API are not processed by the ECNN. This is shown in *Figure 7.2.2* and *Figure 7.2.3*, which display the inability of face detection for side angled faces and covered and widely varying sized faces respectively.

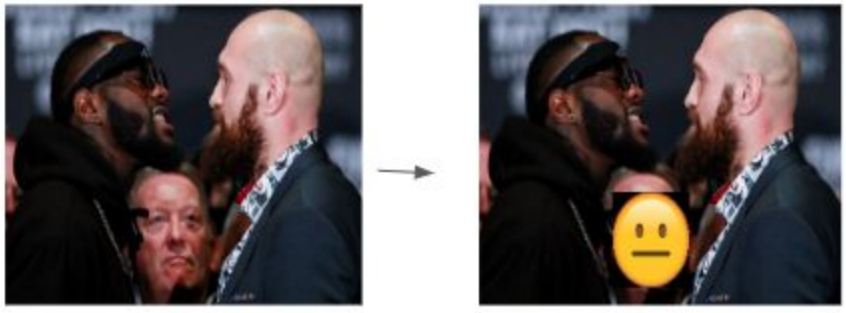


Figure 7.2.2: The input and output of an image consisting of side face angles

Figure 7.2.3: The input and output of an image consisting of faces that are covered and faces that significantly vary in size.

The model also predicts a neutral emotion when given an input image of angry, sad, or surprised emotions ("sad" example in *Figure 7.2.4*). With the angry, sad, and surprised

emotion data comprised predominantly from AffectNet, it is observed that many images between the three categories are very similar to one another and to those from the “neutral” category (*Figure 7.2.5*). This is suspected to be the main cause of this mis-classification.

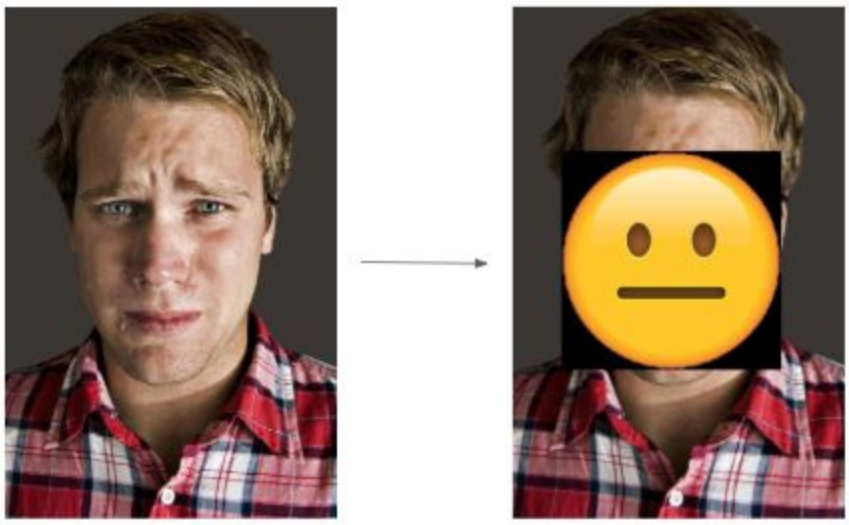


Figure 7.2.4: The input and output of an “sad” image

Figure 7.2.5: The similarity of photos between emotion labels

8.0 Discussion and Learning

From *Figure 7.1.1*, it is observed that the ECNN model exhibits overfitting characteristics. During Neural Net architecture design iterations, many designs were considered to prevent such results. Attempts include implementing dropout on fully connected layers, applying L2 regularization, and tuning possible hyperparameters; however, no design combinations were found to prevent the model from overfitting.

Transfer learning is also explored with applying inputs to a pretrained VGG-16 model, which had its last classifier layer modified and tuned to fit MojiMi’s classification goal. Despite a significant decrease in training time, the best accuracy without overfitting is around 55% (*Figure 8.1*), which is much lower than the baseline model’s accuracy. The

cause of this accuracy outcome is suspected to be due to VGG-16 model's primary design to perform image categorization for the imageNet challenge. Low image resolution may also have contributed to transfer learning's low accuracy.

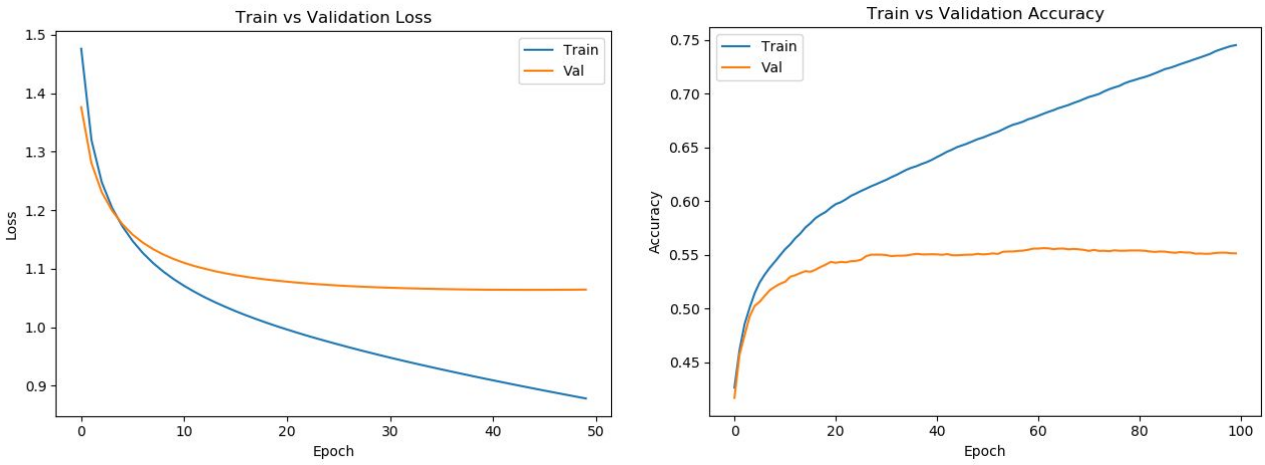


Figure 8.1: The training and validation loss (left) and accuracy (right) curves for tuned and modified VGG-16 model

Consisting of around 60% of the entire dataset and almost all of the sad, surprised, and angry picture categories, AffectNet's largely varying photo details and noise may be the main contributor to ECNN's overfitting and inability to reach higher validation accuracies. The source and quality of data also results in an accuracy and confusion matrix that does not accurately represent input photos taken from other sources, such as social media. According to the confusion matrix, the model performs worse when input pictures are from sad, surprised and angry emotion categories, which is believed to be associated with the fact that the majority of sad, surprised and angry image data are sourced from AffectNet. Due to time constraints, ECNN is trained with input photos of 64x64 dimensions, which may have furthermore contributed to overfitting due to bad image resolution quality.

For similar projects, a diverse dataset should be more strictly considered to fit the project's application. For example, for this project in hindsight, the Facebook API could have been used to scrape and to obtain photos directly from social media for labelling with Azure Face API. This could possibly result in less overfitting, higher prediction accuracy, and a better representation of potential input data for the overall program.

9.0 Ethical Discussions

The project is developed with the main intention to benefit social media users through two main methods. By being able to automatically detect the location of faces and recognize corresponding emotions, the project provides an automatic method for users to apply filters- in this case an emoji- onto social media photos and posts, ultimately resulting in saved time. Since the faces of users are being covered, the project further benefits users by hiding their identity from potential online web scraper or strangers attempting to collect personal information. This respects and protects the users' right of portrait.

On the other hand of benefit and autonomy, justice is a concern. The dataset's bias relative to setting or demographic will affect the prediction accuracy of the ECNN model, possibly impacting some minority groups, which arouses equity concerns. The inaccuracy may also lead to false detections, which is undesirable and can generate potential harm to users in certain scenarios- for example: identifying sad faces in a funeral setting with a happy prediction, which leads to deeper sorrow in those involved with the picture.

References

[1] Y. Fan, X. Lu, D. Li and Y. Liu, (2016). “Video-based emotion recognition using CNN-RNN and C3D hybrid networks,” Proceedings of the 18th ACM International Conference on Multimodal Interaction, pp. 445-450. [Online]. Available: ResearchGate, https://www.researchgate.net/publication/308453418_Video-based_emotion_recognition_using_CNN-RNN_and_C3D_hybrid_networks?fbclid=IwAR1MtMcRMpIoTUY2gx2kedeAecShZpF89EoKQM813yvxMpU94dIF_xwk6zc. [Accessed October 24, 2019]

[2]GitHub. (2019). Flickr-Faces-HQ Dataset (FFHQ). [online] Available at: https://github.com/NVlabs/ffhq-dataset [Accessed 15 Nov. 2019].

[3]Mahoor, M. (2019). AffectNet. [online] Mohammadmahoor.com. Available at: http://mohammadmahoor.com/affectnet/ [Accessed 15 Nov. 2019].

[4] Krizhevsky, A., Sutskever, I. and Hinton, G. (2019). ImageNet Classification with Deep Convolutional Neural Networks. [online] Toronto. Available at: https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf [Accessed 25 Oct. 2019].

References

Permission to Post Video: Yes
Permission to Post Final Report: Yes
Permission to Pos Source Code: Yes