Emotion Recognition from Webcam Video Streams Using Deep Learning

Grayson Lewis Nelms[1,2]*, Chayne Thrash[3], Soheil Kolouri[3]

[1]*School for Science and Math at Vanderbilt, Nashville, TN, US 37203*
[2]*Hume-Fogg Academic High School, Nashville, TN, US 37203*
[3]*Department of Computer Science, Vanderbilt University, Nashville, TN, US 37203*

KEYWORDS. Emotion, Recognition, Training, Facial

BRIEF. Utilizing ResNet18 to predict emotion from human faces for benefit of persons with neuropsychological disorders such as schizophrenia.

ABSTRACT. Facial detection and recognition are tasks easily performed by most humans and are even used in technologies to differentiate between human and automated activity [1]. However, certain neuropsychiatric disorders like schizophrenia can impair the ability to recognize emotions from facial expressions. This project aimed to train a ResNet classifier to recognize emotion in over 35,000 testing and training images with a low enough processing time to be able to use live webcam capture, using a FaceNet embedder to detect faces and identify their emotions, removing noise and returning a value for one of 7 emotion choices. The finished model used various data augmentations and was trained over 100 epochs, where with a high training accuracy, it attained a testing accuracy of 71% on a completely unseen dataset. It was very successful with positive emotion classification. The model struggled more with negative classifications, with some possible explanations discussed later in this article such as labeling issues for our training data or more general ambiguity for some expression of certain emotions. However, it is still largely accurate at classifying emotion into its corresponding class and more so at identifying the positive or negative connotation. This can be utilized in video call tools such as Apple's FaceTime, as an assistive technology—similar to subtitles for the deaf and hard of hearing, a live readout to assist persons whose ability to perceive emotion is impaired by a neuropsychiatric or psychosocial disorder. Even for entertainment such as film, this technology could be implemented automatically.

INTRODUCTION.

Recognition of emotion is a traditionally challenging task for AI models. It requires detection of a face and then a model capable of discerning feelings from just an image of a human face. This is something that is often taken for granted by humans—we've been conditioned for life to be able to interpret body language, including facial expressions such as a smile or a frown to distinguish between expressed emotions without a verbal commitment or affirmation [2]. However, artificial intelligence has traditionally struggled to consistently identify emotion, and this project sought to achieve high accuracy of facial detection and emotion recognition.

Some neuropsychiatric disorders, namely schizophrenia, impair a patient's ability to discriminate emotion by facial expression, largely due to the additional observed symptom of "flat affect" where an affected person suffers difficulty in expressing and interpreting emotions [3]. This condition is more prevalent in men, but the difficulty in recognition affects many and can exacerbate the pre-existing symptoms of the disorder. Bull et. al., 2006 states that "The hypothesis that flat affect has an adverse effect on course of illness was strongly supported. Patients with FA had poorer premorbid adjustment, worse current quality of life, and worse outcome 1 year after affect was assessed" [3]. In addition, this difficulty to discriminate between negative emotions can, while not directly, encourage violent action due to emotional misinterpretation [4, 5], and we hypothesize that by mitigating this deficiency, some of these adverse effects may be slowed or regulated.

We used a deep learning model to learn from a dataset of labeled emotions, and construct a network based on ResNet18 with roughly 11 million parameters [6] to take in frame-by-frame inputs of a camera and predict a subject's emotion class. This required a dataset for training and testing, as well as a way to intake webcam data and transform it into a form that the model can process. The model must be able to recognize faces and predict each emotion based on the weights that it has optimized over the training period on the training/testing dataset of facial examples.

MATERIALS AND METHODS.

Model Architecture and Data.

For training and testing, we utilize the FER2013 emotion recognition dataset [7], which consists of 35,685 grayscale images (28,507 training, 7,178 testing, *Fig. 1*) of size 48x48 pixels. Each image is labeled as one of seven classes: 'angry,' 'disgusted,' 'fearful,' 'happy,' 'neutral,' 'sad,' or 'surprised.'

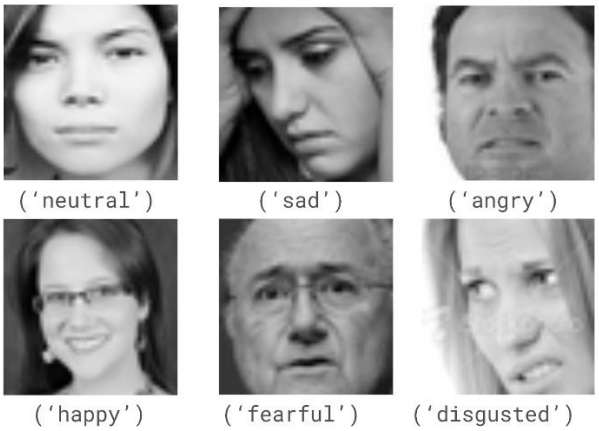

Figure 1. Examples of training data and their subsequent labels in the FER2013 dataset.

We chose to utilize a pretrained ResNet18 architecture [6] via Python's PyTorch package, pulling from a computer vision file. This was because ResNet18 has shown proficiency at deep neural networking tasks such as feature extraction and image classification. Additionally, this project aimed to use a model to, in real-time, interpret image data and calculate via our set of parameters, so speed of processing is of the utmost importance in order to be utilized quickly and accurately to determine emotions. The first stage of the model's training is solely image classification, based upon intaking 48x48 images and sorting them into 7 distinct classes.

We also utilized the Adam SGD (stochastic gradient descent) method. According to Kingma et. al., 2014 [8], the method is "computationally efficient, has little memory requirement, invariant to diagonal rescaling of gradients, and is well suited for problems that are large in terms of data/parameters". This proves ideal to train models quickly and adjust the data augmentation efficiently.

The optimizer uses a learning rate of 1e-3 and a weight decay of 1e-4. The step size of the torch StepLR scheduler is 30, with a gamma value of 1e-1.

Training Phase.

To improve the model's resilience to noisy or varying inputs, we used data augmentations with several methods of image manipulation. As mentioned earlier, the dataset [7] was broken into 28,507 training samples (~80%) and 7,178 testing samples (~20%). These were already designated between training/testing, and the data distribution was roughly equivalent for each distinct class between the two groups, shown in Table 1. The Adam optimizer and ResNet18 model architecture (pretrained) meant that the speed of the training period was very efficient. The entire training period of 100 epochs took only just over an hour using an A5000 GPU with 4 workers, allowing easy modification to the functionality.

Table 1. Training and Testing Data Distribution by Emotion

Emotion	Training Points	Testing Points	Training (%)	Testing (%)
Angry	3995	958	13.915	13.346
Disgusted	426	111	1.519	1.546
Fearful	4097	1024	14.271	14.266
Happy	7215	1774	25.131	24.714
Neutral	4965	1233	17.294	17.177
Sad	4830	1247	16.824	17.373
Surprised	3171	831	11.045	11.577

Our data augmentations were some of PyTorch's default transforms, such as resizing and cropping to fit various sized images, because not all facial regions will be exact squares, and faces in testing may not actually be entirely in frame. There are also random flips, rotations, crops, and color jittering. Fig. 2 shows examples of these transformations.

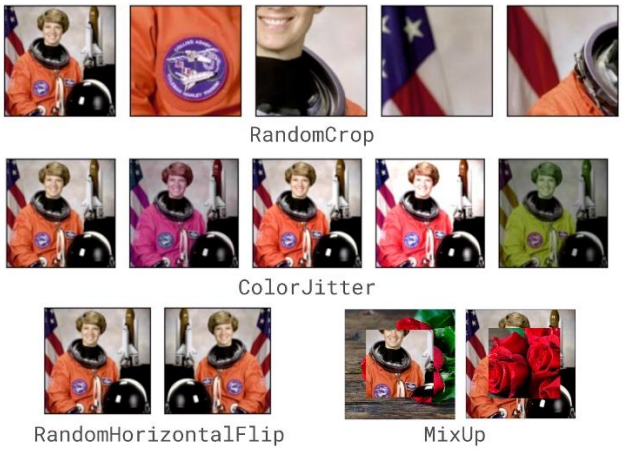

Figure 2. Examples of utilized PyTorch data augmentations (transforms) and timm's Mixup.

We also chose to utilize PyTorch's image model 'timm' for Mixup. This augmentation lowered the training accuracy's potential to roughly 78%. However, without Mixup, the training accuracy reached a peak of 99.97%. The model's accuracy and loss were recorded (see Fig. 3) over the period using a standard cross entropy loss function and reaching a training loss point of ~0.9 and final testing point of ~0.93.

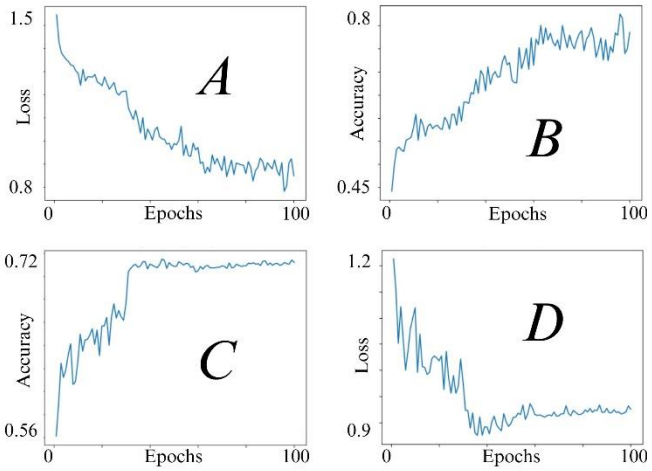

Figure 3. These graphs show the training(A, B)/testing(C, D) loss(A, D) and accuracy(B, C) over the 100-epoch period.

Facial Detection.

The FaceNet architecture implements a triplet loss function to attempt to match similar or identical faces by minimizing the Euclidean distance between an anchor and a positive embedding and maximizing distance between the anchor and the negative embedding. However, in this project, we only need the ability of the embedder to extract the region which contains a face.

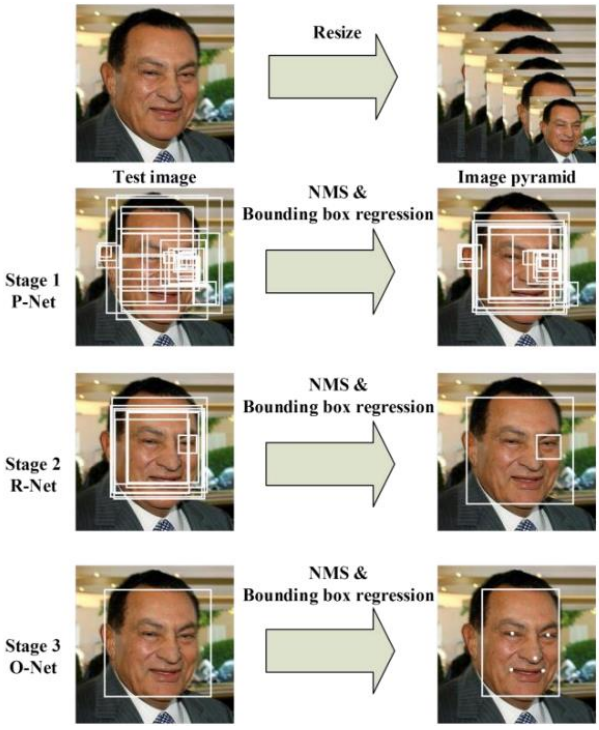

Figure 4. MTCNN workflow chart shows the process of identifying key features and subsequently locating the most probable area for face location, denoted by the box to increase confidence.

Our usage of this function returns four-pixel values to create a rectangular bounding box around every face in frame. We use a confidence value of 0.95 in our extractions of the facial region of interest. (R.O.I.)

The FaceNet architecture utilizes the Multi-task Cascading Convolutional Network (MTCNN) [9] to identify facial features within the bounding box to further assist the embedder in maximizing accuracy

in placement of the region's landmark pixels, as shown in Fig. 4. We are also using cv2 to constantly stream each frame from the available camera before converting it to a PIL image and then a tensor for model calculation- these values are similar to those of the training and testing data after all the data transformations, and have a bounding box drawn upon them for visualization of the FaceNet's embedder's ability to locate faces, where the embedder is pretrained with a high rate of detection as sourced in the original development [9].

RESULTS.

After training, the model reached a peak 71% testing accuracy of correct emotion recognition, without any of the data augmentations. Without rotation and color jittering, the model achieved roughly 64%. Training accuracy remained around 80% with Mixup and Cutmix, although it reached 99.7% when these were removed. We used a t-SNE (t-distributed Stochastic Neighbor Embedding) plot to visualize the 2D embeddings of this 500-dimensional space in Fig. 5 and Fig. 6, where it is important to take note of which classes the model finds challenging to classify.

We planned to utilize a method of facial 'memory-' where the program records the past 3-4 facial emotions and concatenates them and refreshes them frame by frame. If all items in this list are the same, then

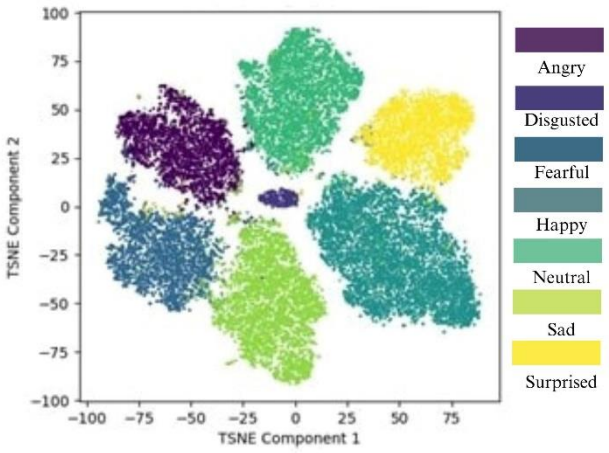

Figure 5. Training Data t-SNE relative embedding plot demonstrates the model's embedding space with over 99% accuracy by minimizing the Kullback-Leibler divergence and clustering the data points by minimization of their Euclidean distance.

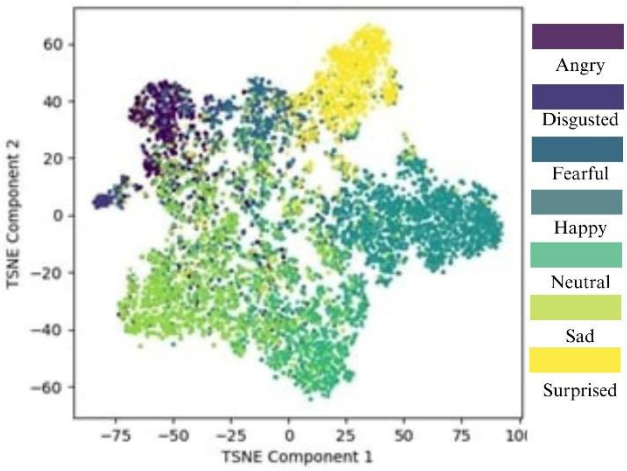

Figure 6. Testing Data t-SNE of the model's embedding space (accuracy of 71.28%) shows the model's difficulty to interpret some classes in the wild.

the model is highly confident that the face in frame is demonstrating that emotion. If they are not the same, the 'memory' is unconfident that there is a new class being emoted and reports no change in the facial data. This effectively eliminates all these one-frame 'noises,' and allows the model to report consistent, accurate data—only 'refreshing' the displayed class when necessary. This also required a method to remember all faces in frame—the model will read all faces and output individual probabilities and concatenate them to their own lists—but if a face drops from the embedder for one frame, a null value is inputted rather than a replacement emotion. This way should the face reappear in frame, the memory system will still be intact. These results are read right-to-left in frame, and the overall functionality of our product is displayed below (Fig. 7).

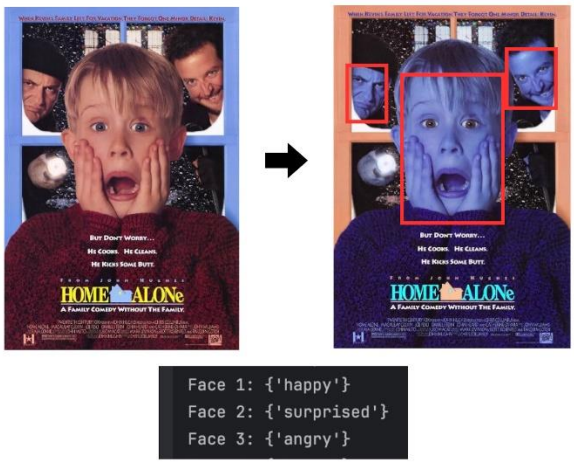

Figure 7. This is the final functionality of our code—the intake of some image, such as a frame from a webcam, applying certain color transforms before drawing a bounding box for the region of interest, sending that R.O.I. to the trained ResNet18 model which outputs a de-noised probable emotion class.

The obscuration of the 'fearful' class in Fig. 6 is likely due to the faulty data labeling, and the small, distinct cluster of class 1 (disgusted) in both Fig. 5 and Fig. 6 is due to the uneven total distribution of data as shown in Table 1.

Webcam Integration.

After some testing of the system with webcam utilization, we discovered that there was a large amount of data 'noise' in the collection period. Certain faces would drop out for a single frame due to an unusually low confidence rate from FaceNet only to reappear in the next frame, or while one emotion was being shown, another would have an irregularly high probability from the model and would therefore be displayed instead.

DISCUSSION.

In conclusion, this project has achieved a viable method to use live webcam feed for facial detection and emotion recognition. If implemented properly, this could be the equivalent of emotional subtitles for assistance to persons impacted by certain disorders which may impair emotional reasoning and accurate discernment.

Upon some review, the FER2013 dataset we utilized [7] appeared to be very poorly sorted—many images were wrongly labeled (See Fig. 8), and others were subjective as to their label. This likely encouraged much inaccuracy and failure to generalize—additionally, some emotions were more difficult to consistently classify. As visualized in Fig. 5 and Fig. 6, for example, the fearful class contains several false positives from the surprised class, demonstrating (as observed during our

debugging period) that these emotions are more difficult to classify—an issue possibly due to the poor labeling of the dataset (Fig. 8) or due to the ambiguity of the training and testing points' labels between these two classes. Some proposed future directions are, first, the implementation of a larger, better-labeled dataset. The misclassification of some emotions is likely due in part to the poor sorting, and having a training/testing dataset with data points that could not be considered ambiguous as to their label would help the model generalize to difficult new data.

Figure 8. Examples of misclassified data in the FER2013 dataset [7], likely causing misclassification of data and poor training of the recognition model.

However, the project did succeed with a relatively high accuracy in its task and is a flexible framework for further application—for example, the formation of infrastructure for the uploading of movie files and video for 'emotional transcription' would be one that would be able to automate the detection and recognition for, instead of live feed, a long-form recorded media input. These could all be implemented in tandem with a system to learn which *multiple* emotions are being expressed at once, instead of displaying one 'trump' emotion among the seven probabilities. This would allow for deeper emotional 'intelligence' of the model. But overall, the model performs well and has thoroughly demonstrated its theoretical and practical efficiency.

ACKNOWLEDGMENTS.

I would like to acknowledge Mr. Chayne Thrash of the Machine Intelligence and Neural Technologies Laboratory for his assistance in debugging and in research, as well as Dr. Soheil Kolouri for his contributions to the MINT Lab's success. I would also like to acknowledge the advisors of the SSMV.

REFERENCES

1. G. Goswami, B. M. Powell, M. Vatsa, R. Singh, and A. Noore, FaceDCAPTCHA: Face detection based Color Image captcha. *Future Generation Computer Systems* **31,** 59-68 (2014)
2. P. Ekman, W. Friesen, Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology* **17,** 124-129 (1971)
3. R. E. Gur et al., Flat affect in schizophrenia: Relation to emotion processing and neurocognitive measures. *Schizophrenia bulletin* **32,** 279-287 (2006)
4. C. Arango, Violence in schizophrenia. *Dialogues in clinical neuroscience* **2,** 392-393 (2000)
5. W. Cho et al., Biological aspects of aggression and violence in schizophrenia. *Clinical psychopharmacology and neuroscience: the official scientific journal of the Korean College of Neuropsychopharmacology* **30,** 475-486
6. K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition. *Institute of Electrical and Electronics Engineering Conference on Computing* 770-778 (2015)
7. I. J. Goodfellow, Challenges in Representation Learning: A report on three machine learning contests. *Neural Networks* **64**, 59-63 (2015)
8. D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization. *Computing Research Repository (arXiv.org)* **1412.6980** (2014)
9. K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, Joint face detection and alignment using multi-task cascaded convolutional networks. *Institute of Electrical and Electronics Engineering Signal Processing Letters* **23**, 1499-1503 (2016)
10. A. Kumar, Multi-View Face Recognition Using Deep Learning. *International Journal of Advanced Trends in Computer Science and Engineering* **9(3)**, 3769-3775 (2020)

Grayson Nelms is a student at Hume-Fogg Academic High School in Nashville, TN. He participated in a research internship through the School for Science and Math at Vanderbilt.