Journal of High School Science

An Empirical Analysis of Machine Learning Based Emotion Recognition

Pei S[1]

Abstract

Emotion recognition is widely used to understand human interaction and social behavior. The tendency for subtle differences in facial expressions to influence the emotion completely and the limitations in dataset collection pose a challenge for machines to accurately and automatically detect emotion from one image. The study was conducted using a facial expression dataset from Kaggle's emotion recognition competition in 2013. We empirically compared different machine learning techniques, explored data augmentation methods, and applied a multiplier technique to further improve accuracy. With both quantitative and qualitative evaluation, we find that Convolutional Neural Network, Random Noise of standard deviation 1, and a multiplier value of 3 have the highest accuracy. We also provide insight on several counter-intuitive results in the study.

[1] Corresponding author: Serena Pei, Horizon Academics, 1376 Trailside Lane, San Jose, CA, serenapei123@gmail.com, +14087082258

Introduction

Emotion is an important part of daily interaction and communication. The interpretation of an emotion is subject to deviations of certain facial features. Figure 1 from (1) shows how a slight deviation in physical features can alter the classification of an emotion. Recognizing human emotion is crucial for machines to communicate and interact with users. A deep learning model can essentially learn how to recognize emotion from different features by training with pre-labeled data.

FIGURE 1: The figure shows subtle differences in emotion between two photographs. Although the physical features appear similar, the expression on the left looks angry whereas the expression on the right looks happy.

Convolutional Neural Networks are generally accepted as one of the most effective models in image processing and recognition because they preserve the locality of pixels. Other models such as Random Forest Classifier, Support Vector Machine, Multi-layer Perceptron, and K-Nearest-Neighbors are less often used for image classification because they treat two-dimensional images as vectors. Therefore, it is difficult to apply any useful transforms to the image. We constructed a table with the overall accuracy of these models to verify that Convolutional Neural Networks are most effective for emotion recognition.

Data augmentation techniques expand a dataset and improve the training process. The application of data augmentation allows for more training data essentially with the same cost. Some types of transforms include Random Horizontal Flip, Random Resized Crop, Center Crop, Random Noise, and Mixup. These transforms create more variability during training, which produces more robust models which are able to generalize better from unfamiliar images. This study will focus on analyzing the overall accuracy of a Convolutional Neural Network (CNN) trained with transformed data. Another technique we explore in the study is the multiplier, a constant that is multiplied to the loss of a specific emotion during training. Increasing loss during training allows the model to learn one emotion better, with higher accuracy. Theoretically, using a multiplier for one emotion indicates that the accuracy for other emotions will decrease, so it is important to weigh priorities before deciding to use a multiplier.

Qualitative observations help visualize the results of training by comparing screenshots of each model's performance. This is another way to visually reinforce the conclusions drawn from the quantitative results of accuracy.

Methods and Model development

The study involved the use of Random Forest Classifier, Support Vector Machine, Multi-layer Perceptron, K-Nearest Neighbors, and Convolutional Neural Network. Then, we applied data augmentation techniques to the Convolutional Neural Network, including existing transforms and certain combinations of existing transforms. In addition, we used a multiplier technique to improve the accuracy of a specific emotion, and through quantitative evaluations, we observed some unexpected results. The overall accuracy of all models and techniques are outlined in our quantitative and qualitative results to verify the usefulness of each method. After comparing the overall accuracy of all models, we conclude that the Convolutional Neural Network is generally more effective than all tested models for emotion recognition. Of the transforms and multipliers used in the study, Random Noise with standard deviation 1 and a multiplier of 3 has highest overall accuracy. We also explore and discuss several nuances for specific emotions across different techniques and models.

A study applied facial expression recognition to a Convolutional Neural Network, as we did in our evaluation (2).

Another study introduced new methods and compared its performance to existing methods such as Multi-layer Perceptron, Support Vector Machines, and K-Nearest Neighbors (3). In our analysis, we compared different models, data augmentation, and multiplier techniques to propose new methods by combining multiple techniques.

A recent study used Mixup and Cutout on Convolutional Neural Networks to recognize images in their dataset (4). On this emotion recognition dataset, we combined Mixup with other techniques such as Random Noise and Random Horizontal Flip. These more diverse transforms may result in more robust models.

Another study introduced the use of approximate multipliers on a deep learning model during training to improve accuracy (5). We apply a similar multiplier technique to a Convolutional Neural Network and evaluate the effects on specific and overall accuracy.

Parts of our study were inspired by two main ideas from a previous work: data augmentation to decrease overfitting and qualitative results to draw conclusions about the model (6). For data augmentation, they used horizontal reflections and intensity alterations of RGB channels using Principal Component Analysis (PCA). PCA is a dimensionality reduction used to filter noisy parts of an image and keep the more valuable parts for training. We implemented Random Horizontal Flip in our study for data augmentation and Random Noise to change intensity of pixels in our

Convolutional Neural Network. We used PCA for Support Vector Machine and Multi-layer Perceptron because the dataset size significantly affected run time. The same study introduces a method of qualitative evaluation, where the five images in the last layer of the Convolutional Neural Network with the smallest Euclidean distance from the original image before and after training are compared to make a conclusion about the growth of the model. In addition to comparing last layer features, we also used the same process for the second to last layer, which offers more insight as to how and what the model learned over one layer.

The study utilizes a facial expression dataset offered by the 2013 Kaggle facial expression competition (7). There were no labels available for the test set, so we split the training set into training, validation, and test sets. There were exactly 28,709 images in the original training set. The seven labeled categories of emotions are anger, disgust, fear, happiness, sadness, surprise, and neutral. The dataset is formatted as a CSV (comma-separated values) file, where each row contains a labeled emotion in the form of a number between 0 and 6 and a space separated list of values from 0 to 255 describing the color intensity for each of the 48 by 48 pixels in one image.

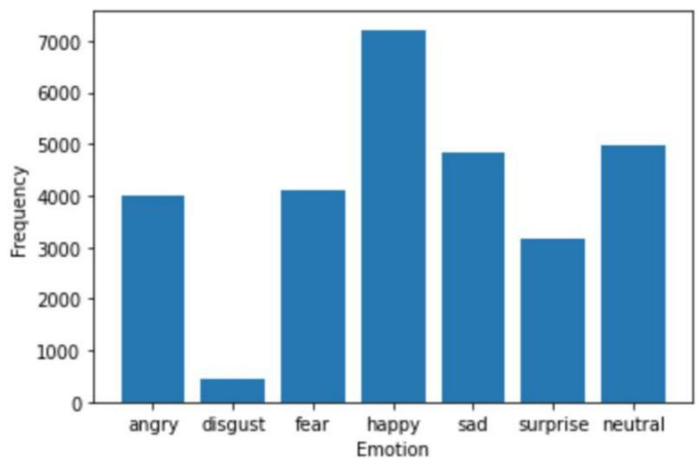


FIGURE 2**:** The bar chart shows the frequencies of the seven emotions from the Kaggledataset: anger, disgust, fear, happiness, sadness, surprise, and neutral. From the chart, happiness is the most frequent emotion and disgust is the least common emotion

As shown in Figure 2, we plot the frequencies of each of seven emotions. Figure 3 shows a random sample of images taken from the dataset along with labeled emotions. This is a random sample, so it should be representative of the entire dataset. Since the proportion of baby faces and colored people's faces are relatively less, we can aim to address that in future studies.

FIGURE 3**:** The figure shows a sample of images from the dataset with its labeled emotion. From this sample of image, it is apparent that faces from different races,genders, and camera angles are represented in the dataset.

The first part involves five widely used models that are generally known to perform well: Random Forest Classifier, Support Vector Machine, Multi-layer Perceptron, and K-Nearest Neighbors implemented with the Sci-kit learn library (8) and Convolutional Neural Network implemented with the Pytorch library (9).

A Random Forest Classifier (RF) combines the "Bootstrap aggregating" method and "random subspace method" method to build a set of decision trees. Bootstrap aggregation takes the characteristics of ensemble learning into account, where a group of weak learners will outperform a single strong learner. Random subspace method refers to the strategy of combining outputs of different models into one final output. The output categories for a RF are determined by the classification results of multiple decision trees. In a single decision tree, the RF uses random selection of training samples and attributes in a sample. The final classification result is decided by equal weight voting after all decision trees are constructed.

A Support Vector Machine (SVM) is a clustering algorithm for classification problems which finds natural grouping within a dataset and finds a separating line between two or more classes. For instance, if there are 2 input features, the separating plane is a single line. If there are 3 input features, the separating plane is two-dimensional. The points closest to the separating boundary are known as the support vectors. All data points located on one side of the arbitrary support vector line are classified in a certain category.

A Multilayer Perceptron (MLP) is a deep, artificial neural network composed of many artificial neurons, which are simple units that take in input signals and compute an output signal using an activation function. The input layer receives a signal, the output layer makes a prediction about the input, and the hidden layers perform the computations. During the training process, the weights and biases are adjusted in order to minimize error. The forward pass compares the ground truth labels to the decision in the output layer. The backward pass alters the weights and biases through the MLP with functions to optimize loss, one such function being stochastic gradient descent.

The K-Nearest Neighbors (KNN) algorithm functions on the assumption that objects with similar features will exist in close proximity with one another. It is versatile in that it can be used in classification, regression, and search. The KNN traverses through the entire dataset to search for images of the closest proximity in terms of Euclidean distance. In other words, 5 Nearest Neighbors of a certain image will be the 5 most similar images in terms of how the model analyzes features. Increasing the number of samples drastically will cause KNN to be significantly slower as it needs to search the dataset to classify each image. For this study, we used 1 Nearest Neighbor because Figure 4 shows that it is most accurate in comparison to other Nearest Neighbors from 1 to 10.

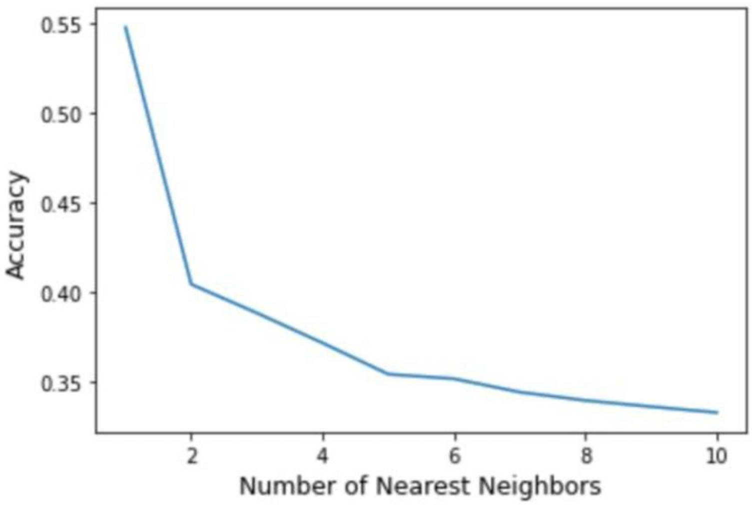


FIGURE 4: The plot shows the accuracy of the number of Nearest Neighbors from 1 to10 for a KNN fitted on the training set. 1 Nearest Neighbor had the highest accuracy.

A Convolutional Neural Network (CNN) has multiple layers including a convolutional layer, non-linearity layer, pooling layer, and fully connected layer. The convolutional layer traverses the two-dimensional image array by iterating through sections of a specified size. In general, a CNN is designed to learn how to classify images by taking in parameters, which refer to the variables that are learned during the training process. The output of a CNN always contains the probability of the image being under each category. In this case, the output of the emotion recognition CNN contains the probabilities of each emotion for each image.

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(16 * 9 * 9, 120)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 7)
```

FIGURE 5: A sample of code was taken from the class Net, which was used to define the layers in our CNN. We structured the network specifically for classifying imagesthat are 48 by 48 pixels. The network contains two convolutional layers, one poolinglayer, and three fully connected layers.

As shown in Figure 5, the net contains 2 convolutional layers separated by 1 pooling layer and followed by 3 fully connected input layers. A general layout of the network is shown in Figure 6. The parameters inputted into the convolutional layers denote the original number of channels, final number of channels, and kernel size for traversing an image. The max-pooling layer summarizes neighboring groups of neurons in the image by returning the maximum value in the given kernel. The fully connected input layers flattens the outputs of the previous layer into a single vector. This shows that CNN is more fit for classifying images because of its preservation of spatial locality.

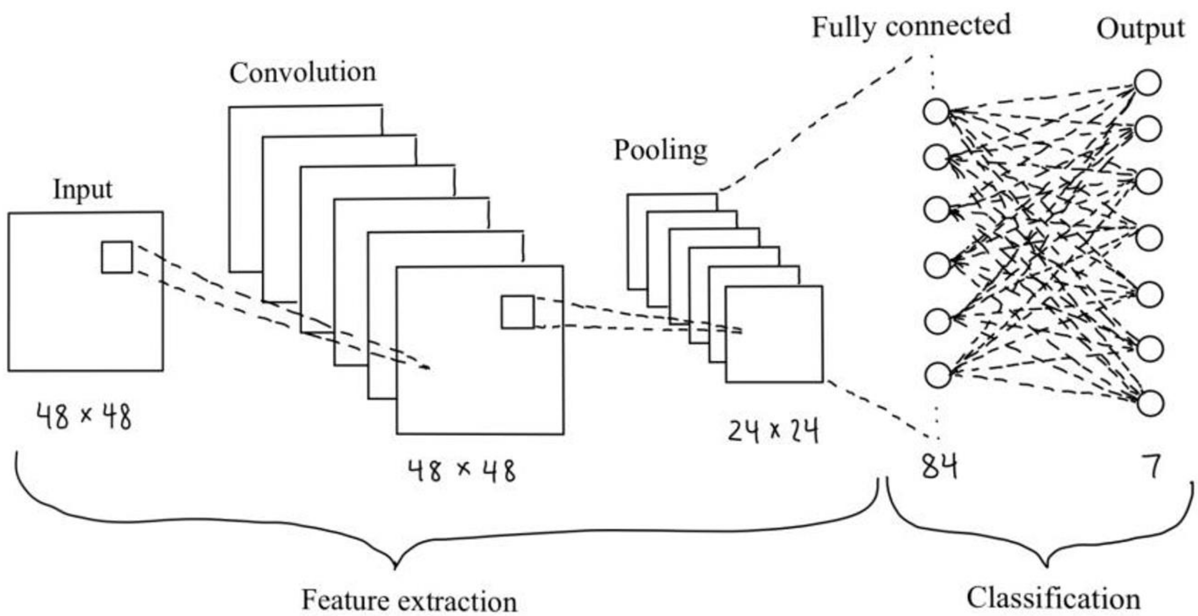


FIGURE 6: The figure visualizes the effect of each layer in the forward pass of the CNN.

The first convolutional layer filters the 48 x 48 x 1 images with kernels of size 5 x 5 with a stride of default 1 pixel. The second convolutional layer filters the output of the pooling layer with kernels of size 5 x 5. The 3 fully connected layers take the output size 16 x 9 x 9 and ultimately returns a vector of size 7, which describes the probabilities for each of the 7 emotions. The last layer returns the probabilities of each of 7 emotions for each image. The outcome is fed into a softmax function, which normalizes the distribution over the 7 emotions so that the individual probabilities sum up to 1. During testing, the model takes the emotion of the maximum probability from the outputs as its prediction and compares the prediction to the true label to obtain the overall accuracy. Figure 7 shows the training and validation loss curves of our CNN. The decreasing loss values prove that the model is improving, which should be the fundamental goal of training a model.

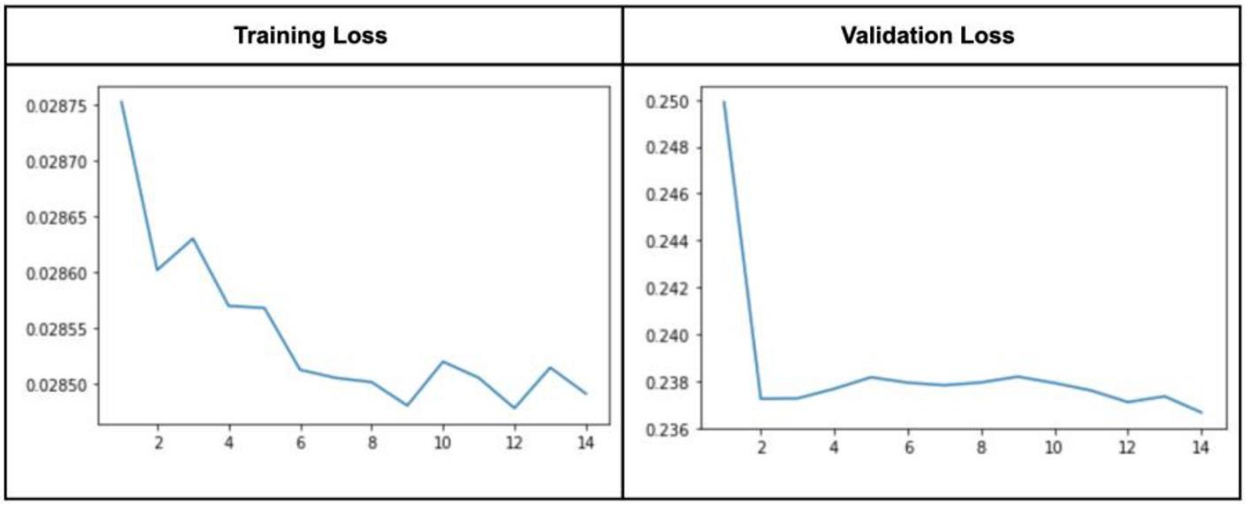


FIGURE 7**:** The figure shows the training and validation loss curves for the CNN. Thedecreasing curve provides evidence that the model is improving over time, which isconsistent with our goal.

Data augmentation is a technique used to expand a dataset by taking existing labeled data and producing transformations of the data while preserving the labels. There is limited data in the world, and the cost to collect and label data is often too expensive both economically and practically. Labeling massive amounts of data requires organizations to hire human workers, which costs money and effort. Data augmentation multiplies the amount of data in a way that is not only cost efficient but also is able to train more robust models. The transformed images are produced directly from the original, so they are computationally free.

Random Resized Crop is the process of producing random subsets of the original images in the dataset, creating more variability for the model to train on. The use of Random Resized Crop builds a more robust model as it needs to learn to recognize images with less information. After Random Resized Crop is performed on an image, the objects of interest are not always completely visible while training and therefore ensures the model does not rely on one specific feature for every classification.

For example, an emotion recognition model may be reliant on eyes to classify emotions, but if Random Resized Crop removes the eyes from an image, the model may have to learn how to classify the same emotion through different features such as the nose or mouth. The dimensions for Random

Resized Crop in this study was 30, meaning that images from the dataset were cropped to 30 x 30 pixels, taken from a random part of each image. Figure 8 shows the visual effect of a Random Resized Crop on one image.

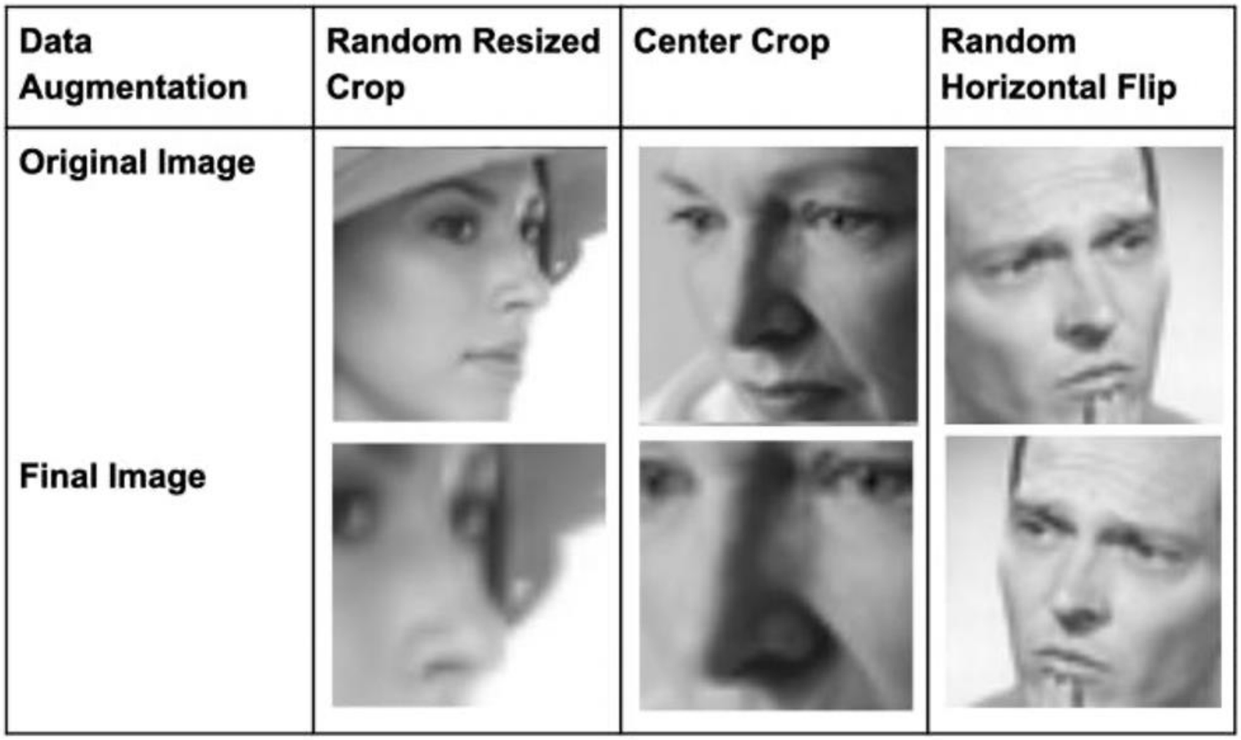


FIGURE 8: The result of random resized crop, center crop and random horizontal flip.

Center Crop takes only the middle subset of an image using the specified size provided in the parameters. In the same way as Random Resized Crop, Center Crop allows the model to learn through different features because some features may be cut out of the training data. In the case of emotion recognition, Center Crop may be more useful because the subject's face in a photo will typically be near the center so that no crucial information will be cropped out. Sometimes, Center Crop even removes irrelevant features around the face that could be distracting when classifying an emotion. The dimension for Center Crop used in this study was 30, meaning that images were cropped to the center 30 x 30 pixels. Figure 8 shows the visual effect of a Center Crop on one image.

Random Horizontal Flip is the process of mirroring an image about its vertical axis given a probability. This can be applied to emotion recognition models because the horizontal orientation of a face should not change the emotion of the face. Since Random Horizontal Flip does not crop parts out of the image, all facial features are kept, which lowers the chance of distorting the emotion. For instance, applying Random Horizontal Flip to an open mouth will still result in an open mouth. Facial features reflected horizontally generally do not have a significant impact on its interpretation. Figure 8 shows the visual effect of a Random Horizontal Flip on one image.

Random Noise creates a noise filter by taking in a mean, standard deviation, and size, where the mean and standard deviation determines the range of values each pixel in the noise filter will return. If the mean is a value other than 0, the distribution of values of the noise filter shifts toward the mean, and a larger standard deviation causes a wider range of values. After the noise filter is created, it is added to the original image, then normalized so that all values are between 0 and 255. For instance, the emotion recognition model may improve with a certain amount of noise because it encourages the model to learn certain emotions with more variability. In this study, we implemented Random Noise with a standard deviation of 1, 5, and 10. Figure 9 shows the visual effect of each standard deviation of Random Noise.

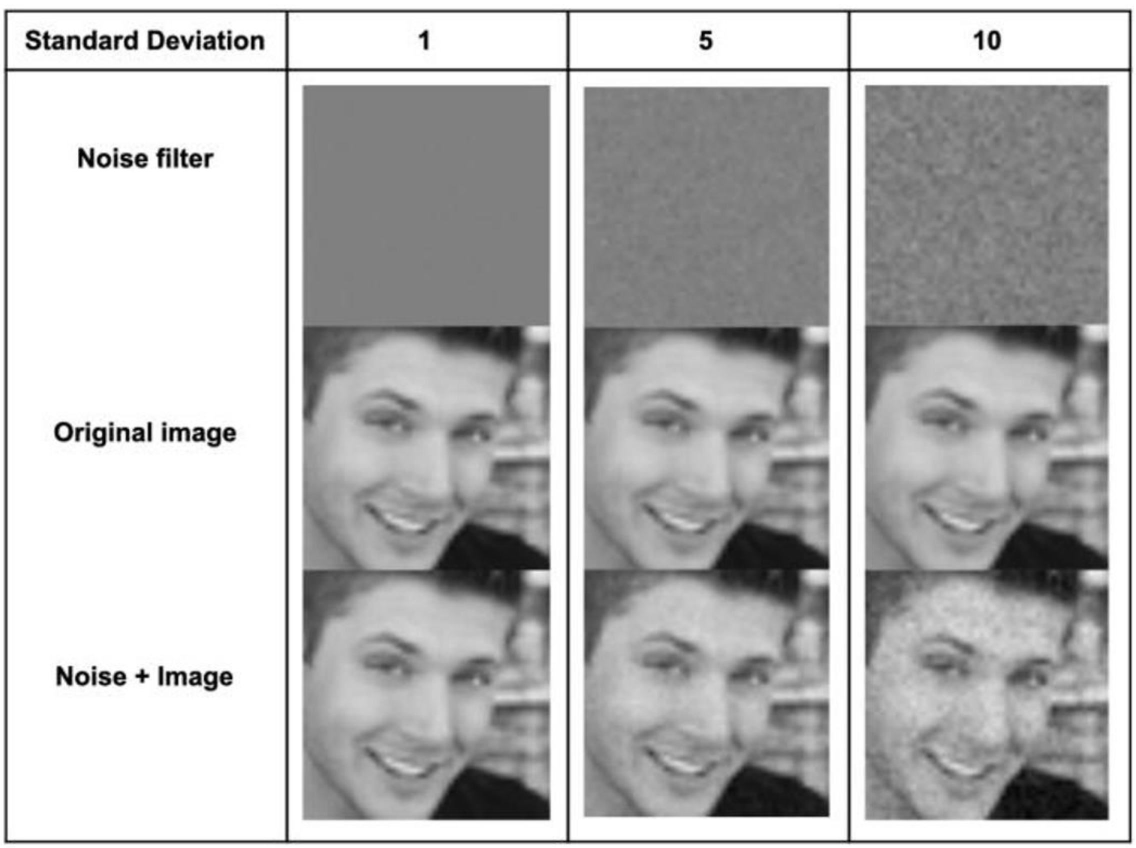


FIGURE 9**:** The figure shows the result of applying Random Noise of different standard deviations on the same image. In order from top to bottom in each column, the noise filter, original image, and final image with noise is shown.

Mixup generates a new image with an image pair from the training data by layering a weighted combination of each image on top of each other. The predicted outputs should include the probabilities of each label as a multi-output classification. For instance, if an image labeled sadness and an image labeled happiness are combined using Mixup with weights of 0.5, the expected outcome would be 0.5 sadness and 0.5 happiness.

Results and discussion

We develop new techniques that combine different data augmentation techniques: Random Horizontal Flip with Noise, Noise with Mixup, and Random Horizontal Flip with Mixup. Random Noise of standard deviation 1 was applied for these techniques. The combination of multiple transforms could potentially create more variance in the training data and ultimately a more robust model. Figure 10 shows the visual effect of applying the three different techniques on random images from the dataset.

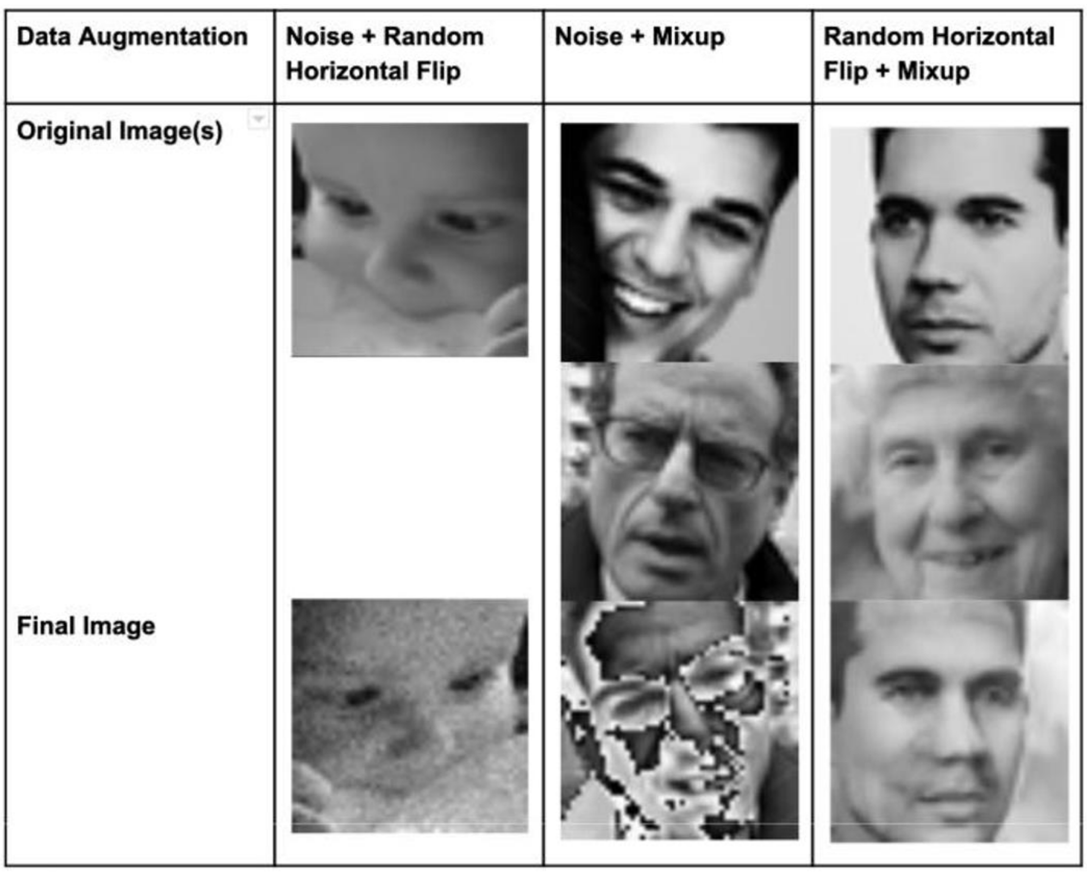


FIGURE 10**:** The figure shows the result of the transforms used in the study: Random Horizontal Flip with Noise, Noise with Mixup, and Random Horizontal Flip with Mixup. We used a Random Noise of standard deviation 5 to more visibly show the effects of the noise filter.

The multiplier is a constant that increases the loss of one specific emotion so that the model learns that emotion better (10). During training, the loss of the specified emotion is multiplied by a constant, which prompts the model to try to balance the loss of that emotion with the loss of other emotions. This increases the accuracy at test time because the model essentially learns that emotion better than before. Table 1 records the overall and disgust accuracy across multiple values of multipliers. The overall test accuracy decreases from multiplier 1 to 2 but increases from 2 to 3 and decreases again from 3 to 5. The disgust test accuracy increases non-linearly from multiplier 1 to 5. Theoretically, the accuracy for disgust should strictly increase due to the trade-off in accuracy between other emotions and the disgust emotion when loss is deliberately increased during training. The reason for this inconsistency could be due to the unbalanced frequencies of each emotion, but we will need further experimentation with other emotions or datasets to draw a definite conclusion.

Multiplier	Overall Accuracy(%)	Disgust Accuracy (%)
Multiplier = 1	65.78	57.35
Multiplier = 1.5	67.72	63.24
Multiplier = 2	57.62	54.41
Multiplier = 3	**72.06**	77.94
Multiplier = 5	71.69	**82.35**

FIGURE 11**:** The table shows the accuracy for multiplier values of 1, 1.5, 2, 3, and 5. The highest accuracies of the overall accuracy and disgust accuracy column are in bold. The overall test accuracy decreases from multiplier 1 to 2 but increases from 2 to 3 and decreases again from 3 to 5. The disgust test accuracy does not strictly increase from multiplier 1 to 5.

In Figure 12, the overall accuracy for the CNN is higher than the Sci-kit learn models in the study. These results verify the accepted belief that the CNN is more effective for classifying 2D images because unlike Sci-kit learn models, it does not unravel the image into a single vector. This is especially important as many useful visual transforms can only be applied to the CNN. Therefore, expanding the dataset at a lower cost is easier for CNN compared to Sci-kit learn models.

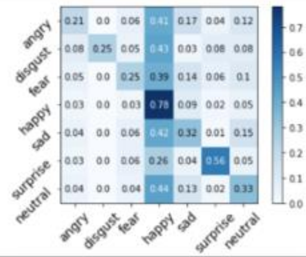
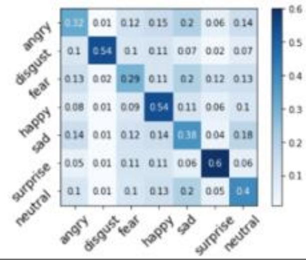
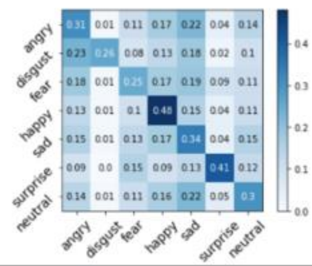
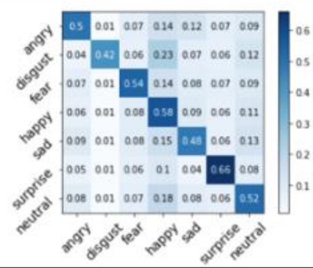
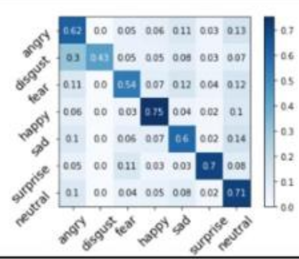
Model	Overall Accuracy (%)	Confusion Matrix
Random Forest Classifier	42.74	
Support Vector Machine	43.12	
Multi-layer Perceptron	35.80	
K-Nearest Neighbors	54.51	
Convolutional Neural Network	**65.78**	

FIGURE 12**:** Overall accuracy and confusion matrices for RF,SVM, MLP, KNN and CNN. The horizontal axis contains the predicted labels while the vertical axis contains the true labels. Darker shades of blue denote higher frequency. The first column includes the type of model and overall test accuracy, and the third column shows the confusion matrices.

Of the four Sci-kit learn models, KNN had the highest overall accuracy. However, the lower accuracy for SVM and MLP may be caused by the limitations of PCA, which was applied to prevent run time from exceeding total session time. Additionally, RF had a greater accuracy for happiness than CNN, which could potentially be useful if the goal was to identify happiness only. In general, RF had a higher concentration of predictions in the happiness category, likely due to the higher frequency of happiness in the original dataset. In KNN, there was a relatively higher frequency of predicting happiness while the actual emotion was disgust. This may be explained by the fact that happiness and disgust have similar physical facial features.

Figure 13 shows the overall accuracy across all techniques applied on the CNN throughout the entire study, including a confusion matrix to compare different multipliers. By comparing the overall accuracy across different models and techniques, we can draw conclusions about the success of each change and whether certain techniques are effective or not. The results show that Random Noise with standard deviation 1 has the highest accuracy rate for emotion recognition compared to other transforms in the study. Also, the combinations of Random Noise, Mixup, and Random Horizontal Flip did not seem to increase performance of the model, which is insightful because it undermines the possibility that more transforms create more robust models.

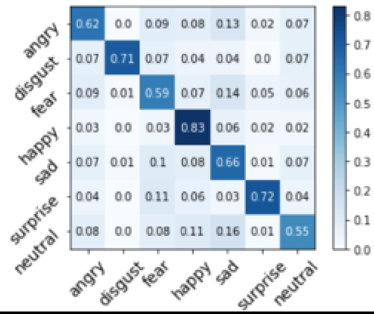
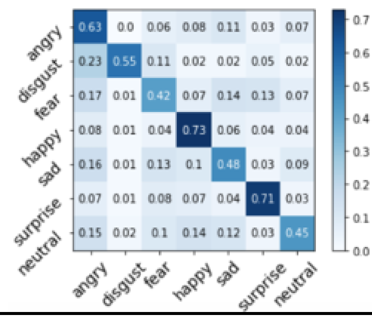
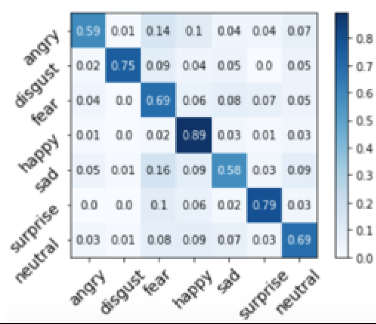
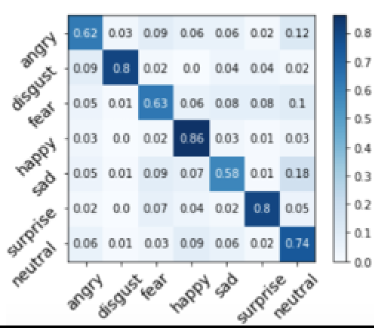
Technique	Overall Accuracy (%)	Confusion Matrix
Multiplier = 1.5	67.72	
Multiplier = 2	57.62	
Multiplier = 3	**72.06**	
Multiplier = 5	71.69	

FIGURE 13: Overall accuracy and confusion matrices of different multiplier techniques applied to the CNN: multiplier 1.5, 2, 3, and 5. Multiplier 3 had the highest overall accuracy of 72.06%. The horizontal axis contains the predicted labels while the vertical axis contains the true labels.

Transform	Accuracy (%)
Random Noise SD = 1	**68.33**
Random Noise SD = 5	66.17
Random Noise SD = 10	25.33
Random Resized Crop	28.08
Center Crop	25.80
Random Horizontal Flip	32.44
Mixup	26.01
Random Flip with Noise	14.49
Mixup with Random Noise	16.47
Random Flip with Mixup	25.55

FIGURE 14: Overall accuracy of the different transforms applied to the CNN: Random Noise of standard deviations 1, 5, and 10, Random Resized Crop, Center Crop, Random Horizontal Flip, Random Flip with Noise, Mixupwith Random Noise, and Random Flip with Mixup. Random Noise with standard deviation 1, in bold, had the highest accuracy of 68.33%.

In Figure 15, we qualitatively assess what the network learned during training by comparing the outputs and actual emotions. The chosen images in the table are randomly selected in order to keep the subset representative of the dataset. Therefore, not all emotions are guaranteed to be included in the table. The orange bar stands for the true label, and the tallest bar stands for the predicted label. Therefore, if the orange bar is tallest, the model has successfully predicted the emotion of that image. From the 6 scenarios shown in the figure, the model was correct 3 out of 6 of the times.

The bar chart provides insight on the labeled data and similarity between emotions. For example, the 4th image was labeled as fear but predicted as disgust. However, from our observation, the image seems to resemble disgust more than fear, so we can assume that not all images in the dataset are labeled correctly. From this, the model seems to be performing well and very close to regular human instinct. Therefore, we can effectively conclude that the accuracy of the model only provides a lower bound for the true accuracy.

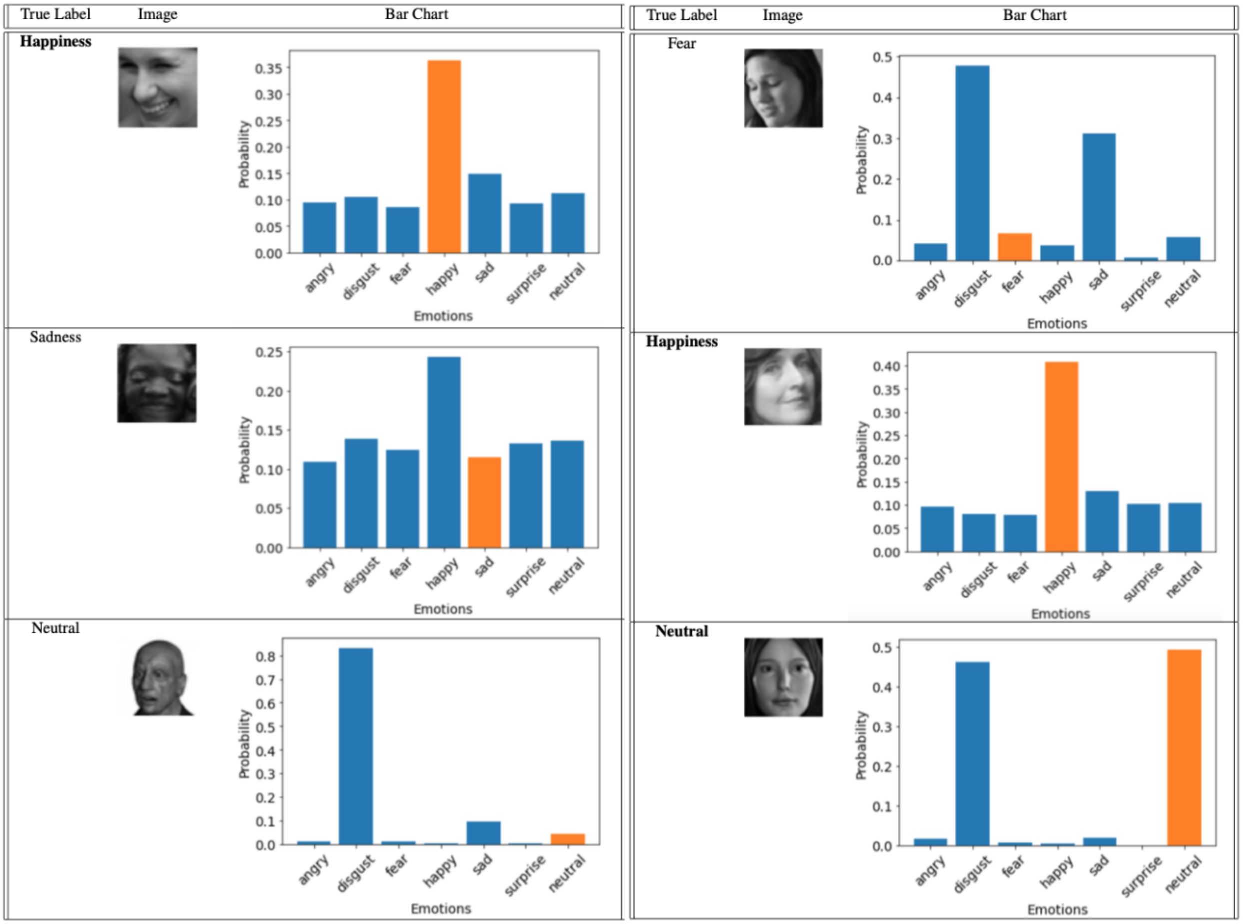


FIGURE 15: The figure shows 6 images with a bar chart of the CNN's predicted probability of each emotion. The horizontal axis is the list of emotions, and the vertical axis is the probability. The orange bar represents the true label, so when thetallest bar is orange, the model has predicted the emotion of the face correctly. The model predicted the correct emotion in 3 out of 6 cases, denoted by the bold emotions.

Figure 16 shows the Nearest Neighbors of five random images before training. Nearest Neighbors are images of the smallest Euclidean distance. The images classified as Nearest neighbors are mostly random and have little pattern, with a rough average accuracy of 8/25.

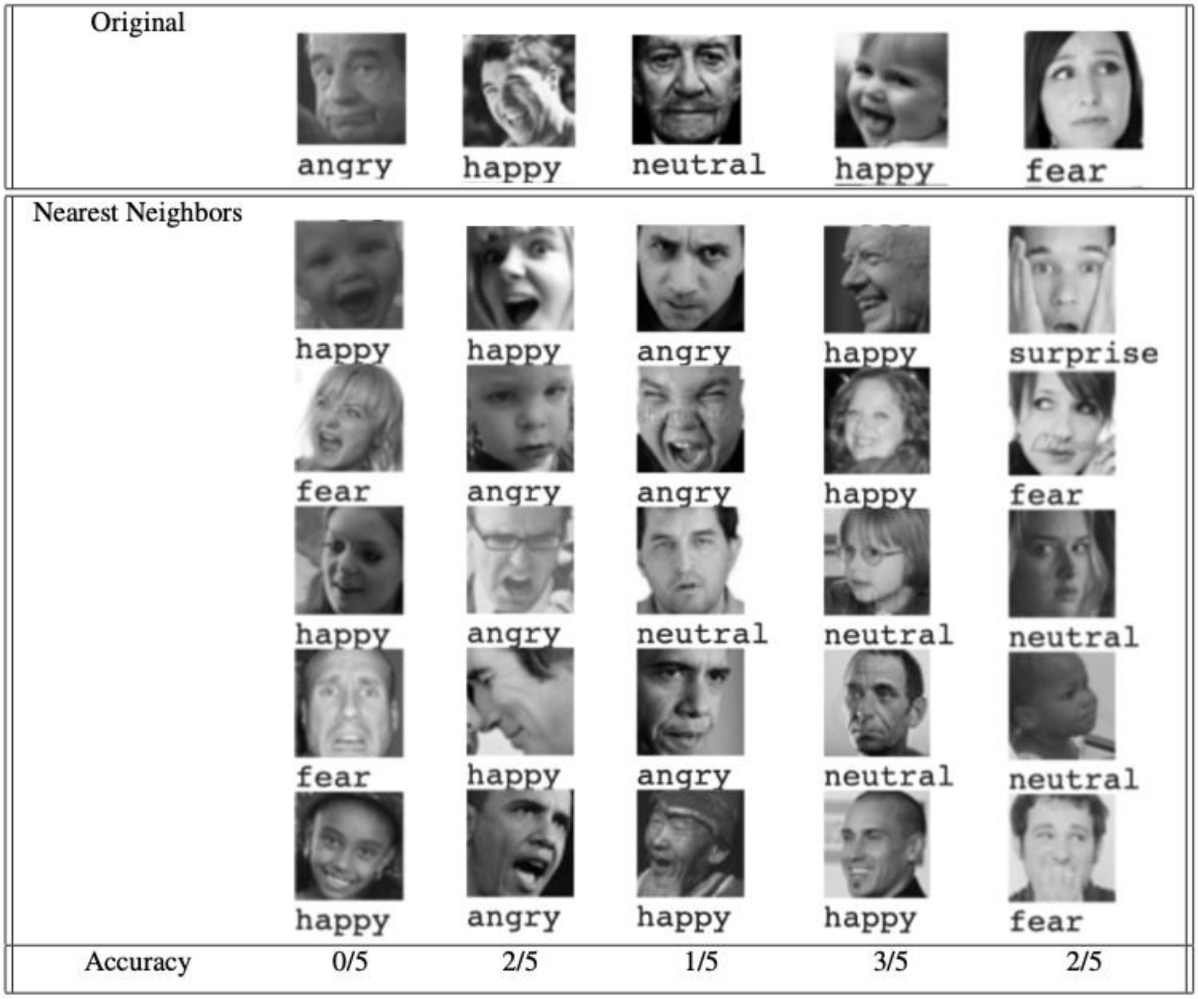


FIGURE 16: The figure shows 5 Nearest Neighbors of 5 random images from the datasetbefore training. The last row shows the rough accuracy of the column, and the number in the numerator represents the number of images that are classified correctly

In Figure 17, the average accuracy is 13/25, which is greater than the theoretical 1/4 probability of random guessing for happiness, the most frequent emotion, and the 8/25 accuracy obtained before training. The reason why none of the Nearest Neighbors for sadness were correct in the first column may be because the angle in the original image and that the intense makeup around the eyes can be misleading. When we analyze the visual similarities between those images classified as Nearest Neighbors, we find that mouth shape seems to play a significant role in determining distance between images. For example, because the mouth in the first image of the first column appears to be open due to heavy makeup, the images of smallest Euclidean distance all have open mouths.

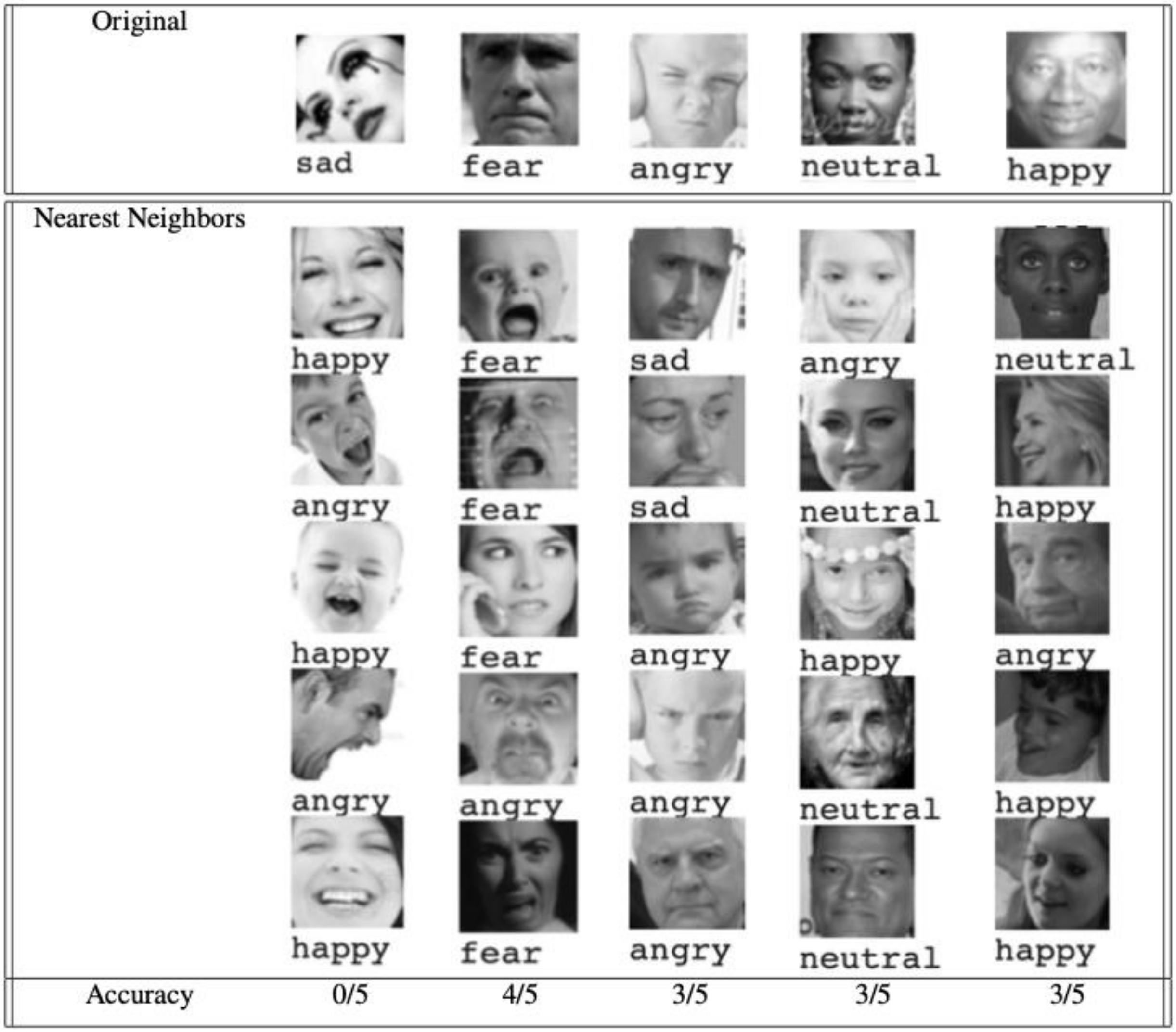


FIGURE 17: The figure shows 5 images of the smallest Euclidean distance in the lastlayer after training, for each of the original images in the first row. The last row shows the approximate accuracy of the column.

In Figure 18, the second to last layer features seem to be sorted by skin color more than actual emotion. Although the images in one column seem very similar at first glance, the actual emotions are not necessarily all the same. Additionally, hand gestures, face positioning, and photo angles seem to play another role in determining Euclidean distance in the second to last layer. For example, all faces in the last column are mostly centered and the full face is included in the image. The Nearest Neighbors of the last layer focus on the facial features more than the Nearest Neighbors of the second to last layer, which focus on the orientation of the face rather than the actual emotion.

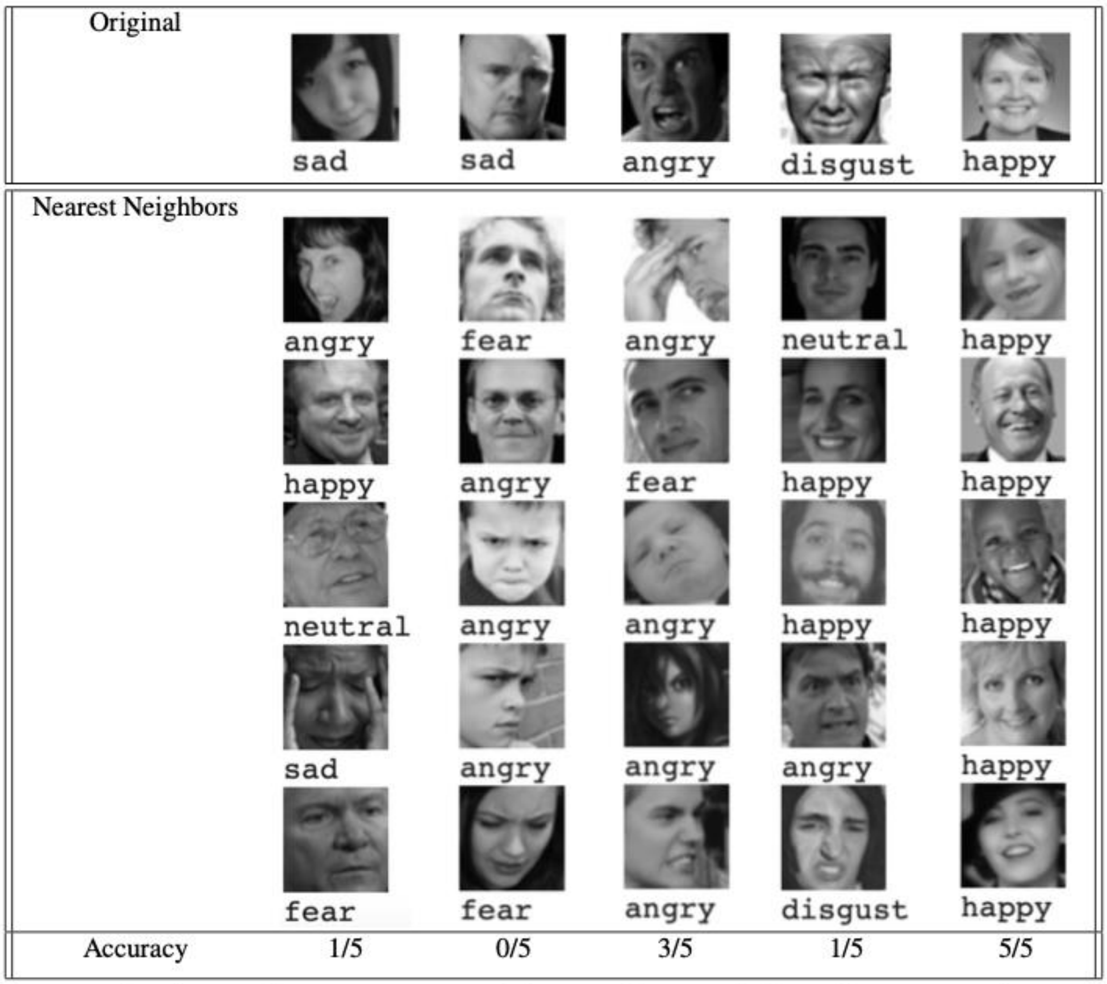


FIGURE 18: The figure shows 5 images of the smallest Euclidean distance in the secondto last layer after training, for each of the images in the first row. The last row shows the approximate accuracy of the column.

Of the 4 Sci-kit learn models, KNN had the highest overall accuracy. However, the lower accuracy for SVM and MLP may have been caused by the limitations of PCA, which was applied to prevent run time from exceeding total session time. Although CNN was more accurate than RF overall, RF had a greater accuracy for happiness than CNN, which could potentially be useful if the goal was solely to identify happy expressions.

Furthermore, while a multiplier is theoretically only useful when there is one main target that needs improvement, our results show otherwise. A multiplier of 3 had the highest overall accuracy while a multiplier of 5 had a higher individual emotion accuracy. For future studies, we can confirm this by using the multiplier on different emotions or datasets and analyzing the fluctuations in accuracy. Our quantitative results across all models and techniques showed that a multiplier of 3 and Random Noise with standard deviation 1 applied on the CNN resulted in the highest accuracy, closely followed by the standard CNN with no transforms. The combinations of multiple transforms did not result in a higher accuracy, but instead resulted in relatively low performance. Therefore, we can conclude that using more transforms simultaneously will not necessarily result in better models. Our qualitative results suggest that the accuracy recorded for each model provides only a lower bound for the true accuracy because some images may be mislabeled or subject to interpretation.

These methods of emotion recognition can be applied to practical situations (11). The researchers turned the seven emotions into emoticons in their application design, which could categorize a user's emotion in a specific category. They also scored the emotion with positive, negative, or neutral based on the user's behavior. These techniques may allow better interaction between humans and machines. We may also try detecting emotion with other signals, including physiological or auditory ones. (12). Then, by combining multiple approaches, we can potentially obtain more accurate results and create better methods of emotion recognition. Emotion recognition models can be applied to video feed with depth sensors to analyze spontaneous expressions (13). Depth sensors use infrared light technology and are more resistant to unbalanced lighting or other factors. These methods will eventually create even more robust and convenient systems for emotion recognition in the real world, such as in video game feedback, website customization, and education.

Most importantly, we hope to explore the racial biases in emotion recognition models, specifically testing if certain skin colors lead to higher or lower accuracy rates. Algorithmic biases can explicitly or implicitly harm certain racial groups and cause disunity within society (14). By correcting algorithmic biases through more robust techniques like the ones introduced in the study, we can aim to solve these problems more effectively, creating equal technological treatment for all ethnicities in the digital age. However, several ethical issues of emotion based machine learning are proposed in another recent study (15). We should recognize the complexity behind emotions and that emotion detectors and other applications of emotion recognition often oversimplify these complexities. Going forward, the emotion recognition community should honor the privacy of users and analyze the ethical and societal impacts of technology.

Conclusions

We conclude that the Convolutional Neural Network is generally more effective than

Random Forest Classifier, Support Vector Machine, Multi-layer Perceptron, and K-Nearest Neighbors. Of all techniques applied, Random Noise with standard deviation 1 and the multiplier of 3 has highest overall accuracy. We discover that combining different transforms does not necessarily improve accuracy and that using a multiplier does not necessarily strictly decrease overall accuracy. By combining physical information with the successful methods described in this study, we can aim to achieve even better results for emotion recognition. Additionally, we can explore the role of ethical implications and racial biases in the model to improve the larger societal issues behind machine learning. With the techniques verified in our study, we can build more robust models that can potentially solve prevalent inequalities in current technological systems.

Abbreviations and Formal definitions

Random Forest Classifier (RF) - a classification model that functions by building sets of decision trees in order to classify an object.

Support Vector Machine (SVM) - a classification model that functions by searching for natural grouping within a dataset and constructing a separating line between two or more classes.

Multi-layer Perceptron (MLP) - a deep, artificial neural network composed of artificial neurons, simple units that take in input signals and compute an output signal using an activation function.
K-Nearest Neighbors (KNN) - a model that searches for images of the closest proximity in terms of Euclidean distance.

Euclidean distance - the arbitrary length of space between two objects, determined by the objects' features.

Convolutional Neural Network (CNN) - a deep learning model that learns spatial hierarchies of parameters through backpropagation through parameters during the training process.

References

1. Bruk D. 2016. This Freaky Gif Shows There's Only A Very Subtle Difference Between Daniel Radcliffe and Elijah Wood. https://www.seventeen.com/celebrity/news/a40703/this-freaky-gif-shows-theres-onlya-very-subtle-difference-between-daniel-radcliffe-and-elijah-wood/

2. Ghaffar F, Facial Emotions Recognition using Convolutional Neural Net, Social Networks and Human Centered Computing Program, 2018.

3. Dilbag S., Human Emotion Recognition System, International Journal of Image Graphics and Signal Processing. IJIGSP, 4(8), 50-56, 2012.

4. Yun S, Han D, Oh SJ, Chun S, Choe J, Yoo Y; CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features, Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 6023-6032, 2019.

5. Hammad I, El-Sankary K, Jason G, Deep Learning Training with Simulated Approximate Multipliers, 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), 47-51, 2019.

6. Krizhevsky A, Sutskever I, Hinton GE, ImageNet Classification with Deep Convolutional Neural Networks, NIPS'12: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, Pages 1097–1105, December 2012.

7. Carrier, Pierre-Luc, and Aaron Courville. 2013. Challenges in Representation Learning: Facial Expression Recognition Challenge, Available at <https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data>.

8. Pedregosa, F., Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, et al. Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12, 2825–2830, 2011.

9. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Paper presented at the Advances in Neural Information Processing Systems 32, 8024–8035. Curran Associates, Inc. Available at <http://papers.neurips.cc/paper/9015- pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.

10. M. S. Ansari, V. Mrazek, B. F. Cockburn, L. Sekanina, Z. Vasicek and J. Han, Improving the Accuracy and Hardware Efficiency of Neural Networks Using Approximate Multipliers, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 28(2), 317-328, 2020, doi: 10.1109/TVLSI.2019.2940943.

11. H. Lee and K. Hong, A study on emotion recognition method and its application using face image, International Conference on Information and Communication Technology Convergence *(ICTC)*, 2017, pp. 370-372, 2017. doi: 10.1109/ICTC.2017.8191005.

12. Lee MS, Lee YK, Lim MT, Knag TK, Emotion Recognition Using Convolutional Neural Network with Selected Statistical Photoplethysmogram Features. Appl. Sci., 10, 3501-3516, 2020.

13. Landowska, A., Szwoch M, Szwoch W, Wróbel MR, Kołakowska A. Emotion Recognition and Its Applications. Advances in Intelligent Systems and Computing. 300, 51-62, 2014.

14. Lee NT, Detecting racial bias in algorithms and machine learning. Journal of Information, Communication and Ethics in Society, Vol. 16 No. 3, pp. 252-260, 2018. https://doi.org/10.1108/JICES-06-2018-0056

15. Stark L, Jesse H, The Ethics of Emotion in Artificial Intelligence Systems. FAccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency March 2021 Pages 782–793 https://doi.org/10.1145/3442188.3445939