NEW YORK INSTITUTE OF TECHNOLOGY

Image Classification: Feature Selection, Data Augmentation, and Transferred Learning

Final Paper

DTSC 870 Project I

Advisor: Houwei Cao

Members:

Michael Trzaskoma	1202901	mtrzasko@nyit.edu
Hui (Henry) Chen	1242445	hchen60@nyit.edu

Table of Contents

Abstract - Computer vision is a growing area of interest utilizing computers to evaluate visual input for various tasks, one of these tasks being image classification. When working with visual data for classification, one of the limitations is the lack of available labeled image data to train models. This project explores several methods to combat the limited image data which are feature modification and selection, data augmentation, and transfer learning. These methods were applied to convolution neural networks and support vector machines where applicable within experiments trying to optimize the accuracy of image classification. The two domains of classification explored in this project are medical data utilizing a MRI brain tumor dataset , and emotion recognition using the facial emotion recognition 2013 dataset.

I. Introduction

Computer vision is a growing field where machines make use of image data through the extraction of key features prevalent in an image, one of the most common applications is with image classification. Computers are able to make classification decisions based on the feature information that is extracted from images and determine what are the distinguishing features between classes within an imageset. The benefit of utilizing a computer over a human to classify images is that computers may observe features of an image unobservable to the human eye, allowing for the possibility of a higher classification accuracy. Such advances in accuracy can be applicable to improving the quality of several domains that utilize images to make decisions. Such domains could be: in the medical field with evaluating the health of the patient based on images of tests being done, or with product evaluation determining if a product is faulty based on key determining features. Two of the common models utilized for image classification are Support Vector Machines (SVM) and Convolutional Neural Networks (CNN).

While image classification can be incredibly useful, it has its challenges. Image classification requires a lot of image data to be accurate, however the availability of usable image data to train a model is very limited. To be a usable image for training, it requires the image to be labeled with the class it pertains to due to classification being a supervised learning task needing data to be labeled. The amount of image data available is also restricted due to privacy concerns, such as people with their medical test results for example, as well as industry regulations by governing bodies that have rules/laws limiting available image data.

As a result of the limited images that can be utilized in classification, methods were developed to combat this situation and enhance the accuracy of computer image classification models. The methods include feature selection/modification, data augmentation, and transfer learning. We used the following handcrafted features in this project: Local Binary Patterns (LBP), Histogram of Oriented Gradients (HoG), Scale-Invariant Feature Transform (SIFT), and Principal Component Analysis (PCA). Based on the gray levels co-occurrence matrix, the LBP compares the neighbors of each pixel to the center pixel to determine the descriptor for the image as a binary number. The feature extraction is robust to variations in illumination and pose. While HoG is a powerful way to detect variations and extract the image features by computing a histogram of oriented gradients of a squared image. For instance, when facial expressions change. In addition, SIFT detects and describes features in images, and matches them against key points in images. Finally, PCA is a method for reducing the number of features in a data set while preserving as much information as possible. This dimensionality reduction technique identifies the hyperplane that lies close to the data and then projects the data onto it under the assumption that: there is no unique variance, the total variance is equal to common variance, and there must be linearity in the data set.

Furthermore, Data augmentation is the concept of having an existing image and making modifications to it through methods such as rotating the image, brightening the image, and reflecting the image over an axis. The purpose of data augmentation in the image classification area, is to allow for the creation of more images from an original image set trying to tackle the challenge of having a limited amount of images to train classification models with. Transfer learning is the concept of utilizing a pre-trained CNN, and molding it to a classification task. How this is done is through preserving a portion of weights on the pretrained CNN while allowing other weights to be modified when the model is being trained on the new dataset. The reason for utilizing a pre-trained CNN is that these CNN's are already trained on millions of images, and some features that these models have extracted from previous image sets can be beneficial to the current classification task. It requires less images to adjust a CNN compared to training a CNN from scratch [1], tackling the issue of having a limited amount of images for classification tasks.

II. Related Works

For the related works, we analyzed previous papers that tackled utilizing various feature change/selection techniques, data augmentation, and transfer learning to understand how it was applied towards their experiment and to display a proof of concept that these techniques have been beneficial in previous experiments.

Ravi et al [3] utilize the handcrafted features from the LBP with combination of the number of neighbors $P = 8$ and within radius $R = 1$ to extract the facial features without any image augmentation methods. They did experiments with different SVM kernels in order to compare it against the CNN approach. As a result, they are able to achieve 76.23% accuracy with a polynomial kernel. However, the value of C and Degree are not mentioned in the paper. The second paper we found is from Kalsum et al [4] where they utilize Local Intensity Order Pattern (LIOP, one of variations of SIFT) and HoG to extract the facial details from FER2013 dataset to achieve 63% accuracy without any image augmentation. Lastly, Sachdeva et al [2] demonstrate an accuracy boost from 80.8% to 89% after utilizing PCA in medical data.

Tang et al. [5] explores data augmentation with several medical datasets, one being a dataset of chest x-ray images. The paper dives into the creation of an algorithm to see which augmentation methods would be beneficial. For this paper they explored 3 augmentations, rotation, cutout, and greyscale. The purpose of this was due to the fact that they were curious if every augmentation would be beneficial in the classification task. Through their experimentation, at least for the medical data, rotation and greyscale seemed to have the best performance in enhancing the accuracy of the chest classification, boosting the accuracy from 87.51% to 91.82%. They deemed that a method such as cutout was harmful more than helpful due to the fact that the cutout had the possibility to cut out a key feature in the classification task within medical data. This cut out feature could be the tipping point since with medical data, small details can be the difference maker between classes.

Two papers highlighted the impact that transfer learning had on improving their experimentation. Thota and Reddy [6] explored the utilization of transfer learning within the medical domain where they were classifying the diabetic retinopathy severity. This dataset contained roughly thirty four thousand images, with 5 classes. The reason for the utilization of transfer learning is that the current top of the line accuracy with this field is roughly 50% and wanted to explore improving the accuracy. They proposed an architecture that utilized VGG-16 for transfer learning, and when this methodology applied, their model's best performance resulted in 74% accuracy for all class accuracy which was over a 20% improvement highlighting how beneficial transfer learning can be. The second paper was Junaidi et al's [7] where their classification task was to classify if an image was either a chicken, an egg, or a hatched egg. Their premise was to evaluate two transfer learning networks, VGG-16, but also VGG-19 to see which transfer learning network would perform better with their dataset. These networks were tested against a baseline CNN to evaluate how beneficial transfer learning could be. Through experimentation their final results were that the initial CNN performed with an 87% accuracy, where the VGG-16 performed with a 90% accuracy and the VGG-19 performed with a 92% accuracy. Through this experimentation, while not as significant of an improvement, it still highlighted that both transfer learning networks outperformed the CNN they developed, displaying the positive impact that transfer learning has on the image classification within this experiment.

III. Tools and Technologies

In this project, we utilized the following tools and technologies (Figure 1).

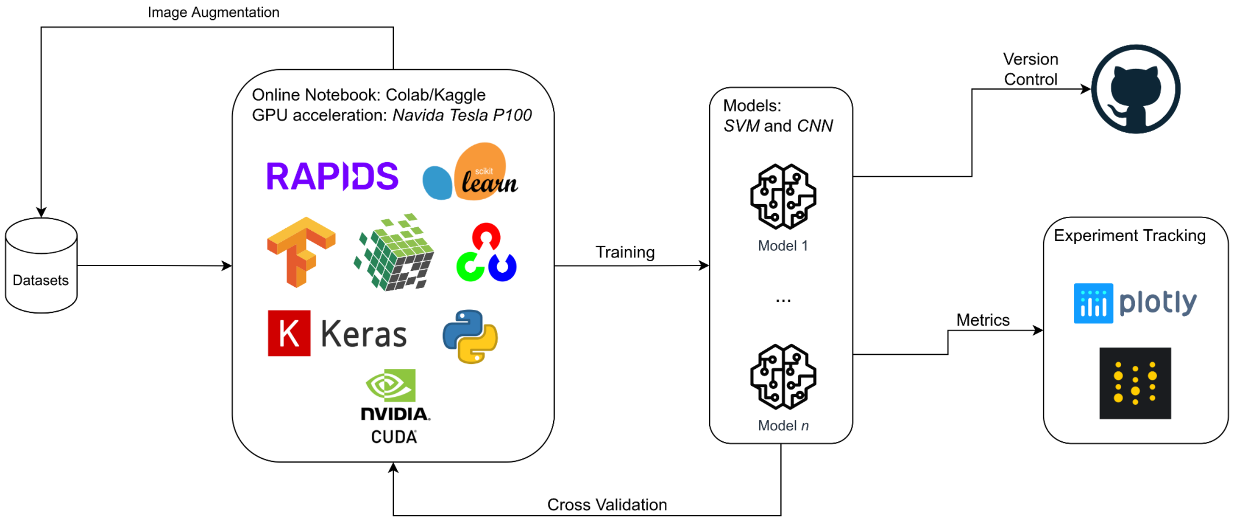


Figure 1: How the Tools and Technologies are Used In this Project

The models were built and trained on Google Colab and Kaggle Notebook with Nvidia Tesla P100 GPU acceleration along with the Python, TensorFlow, Keras, Nvidia Cuda, Rapids, scikit-lean, OpenCV and cupy. At the same time, we utilized plotly and Weights & Biases for the experiment tracking, and GitHub as our model version control.

IV. Experiments

The experiments utilizing SVM and CNN were performed on two domains of image classification: medical data classification, and facial emotion recognition. The models performed classification tasks without any performance enhancing techniques (feature selection, data augmentation, and transfer learning) at first, and then applied these techniques to evaluate how beneficial they are. When performing experiments, Stratified K Fold Cross Validation was utilized with 5 folds to get a better overall sense of how impactful a method was on improving the accuracy. To note, this also reduced the amount of data used for training due to the split of training data for validation. The metrics that we utilized to evaluate the models are overall accuracy, accuracy per class, and ROC curve.

A. MRI Brain Tumor Classification

The dataset utilized for medical data classification was located on kaggle [8] containing a total of 400, 256x256 pixel RGB images, where 170 images pertained to the “Normal” class and 230 images pertaining to the “Tumor” class (Figure 3).

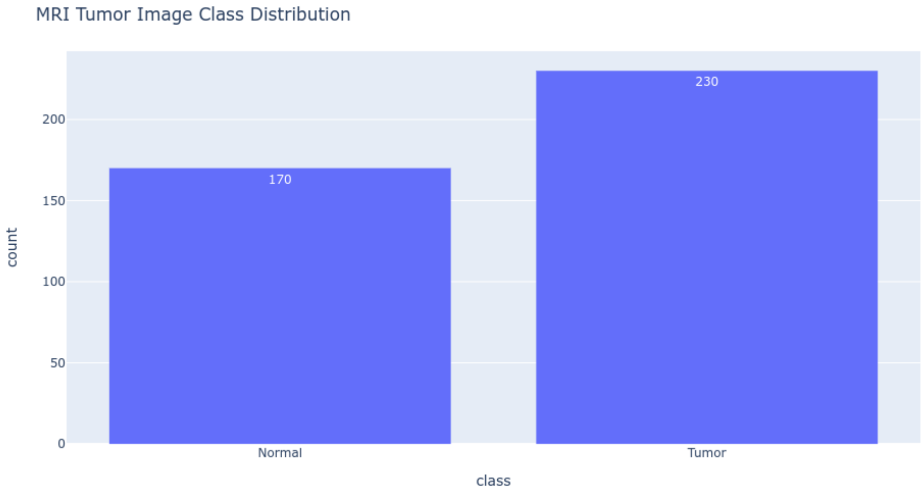


Figure 2: Class Distribution of MRI Brain Tumor Dataset.

1. Image Augmentation: Rotation and Reflection

There is a number of research that suggests applying image augmentation techniques to the data set might improve the model's performance. Throughout the entire project, we trained our models without and with the image augmentations and drew comparative studies. The image augmentation techniques we utilized for the MRI data set were randomly rotation (clockwise or anticlockwise 90°) and horizontal reflection. The image augmentation allows us to enrich the training set.

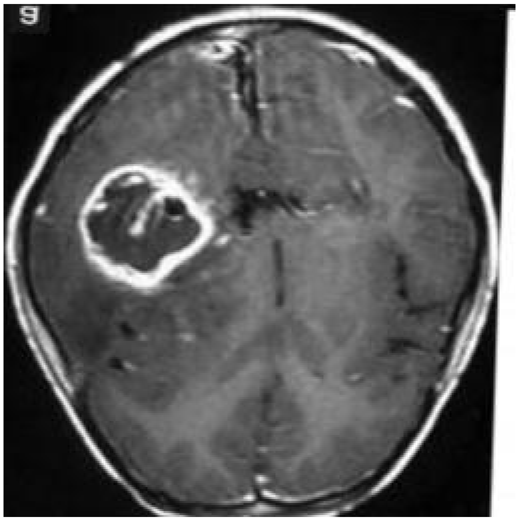


Figure 3.1: No Rotation and Reflection

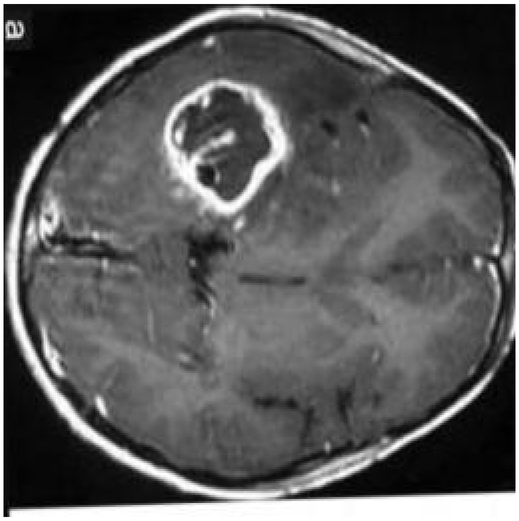


Figure 3.2: After Applying Rotation and Reflection

2. Support Vector Machine (SVM)

a) Initial SVM

The initial SVM was trained without extracting any features or feature selection methods. This enabled us to have a glance of how the model could perform. The preprocessing steps were flattening the image into raw pixel, 256 * 256 * 3, which results in a total of 196608 pixels per image and each pixel will be served as a single feature. Then standardization was applied to the raw pixel data before training the model. This step was done on the data without and with image augmentation (rotation and reflection) and the experiment results are below in figure 4.

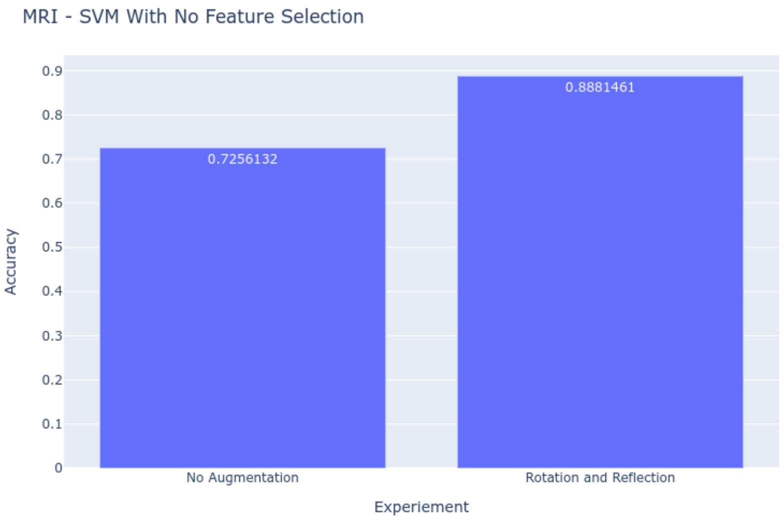


Figure 4: MRI Dataset For SVM Without Feature Selection

As we can see from Figure 4, after applying rotation and reflection on the dataset, it significantly improved the model's performance from 72.56% to 88.81%. This experiment demonstrated the image augmentation does help the performance of the model.

b) PCA

In the second experiment, we performed an experiment by applying the PCA to the raw pixel data to see how it would impact the performance of the model. Firstly, we applied the PCA to the entire training set (both no augmentation and rotation and reflection) without specifying the variance in order to calculate the cumulative variance along with the number of components (features) in the PCA. The following variance is what we selected from the cumulative variance and utilized for the experiment: 100%, 99%, 97%, 95%, 90%, 80%, and 70%. After applying the PCA to the raw pixel data, the dimension of the features are reduced as shown in the following table:

Raw Pixel Features (without PCA)	PCA Variance	# of Features
196608	100%	279
	99%	245
	97%	206
	95%	179
	90%	131
	80%	77
	70%	46

Table 1: Raw Pixel Data and PCA Features for MRI dataset

The original raw pixel data reduced from 196608 to 279 features for 100% variance while 70% variance reduce the dimension to the 46 features. After applying the PCA, each set is trained with SVM as shown in figure 5.

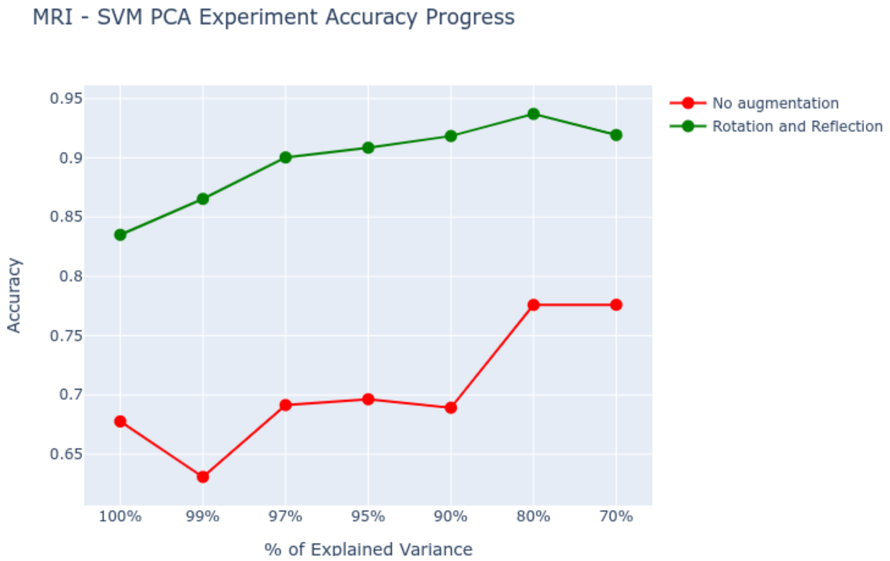


Figure 5: The Performance of SVM with Different PCA Variance

Based on figure 5, after the augmentation, the performance of the model significantly improved as the variance gets lower. However, under the rotation and reflection, there is a slight performance drop as the variance increases from 80% to 70%. On the other hand, without augmentation, the model performance only reached around 77.5%. Out of all PCA variance, the best performance is at 80% variance on rotation and reflection with 93.70% accuracy.

c) LBP and PCA

For the third experiment, we reduced the dimension through PCA after extracting the handcrafted features from LBP. This step is applied on both no augmentation and rotation and reflection sets. In this experiment, we experimented the following combination of *P* (the number of neighbor) and *R* (radius) for the LBP function: (*P*: 8, *R*: 1), (*P*: 16, *R*: 2) and (*P*: 24, *R*: 3) since a number of researches suggested that, the higher the neighbor and radius are, generally, the better of the model's performance is.

Raw Pixel Features (without PCA)	PCA Variance	Number of Features Without LBP	LBP (*P*: 8, *R*: 1)	LBP (*P*: 16, *R*: 2)	LBP (*P*: 24, *R*: 3)
196608	100%	279	279	279	279
	99%	245	272	273	273
	97%	206	262	264	265
	95%	179	253	255	257
	90%	131	232	235	239
	80%	77	193	200	205
	70%	46	158	166	173

Table 2: LBP features and PCA

As we can see from Table 2, the dimension of the handcrafted features from LBP are maintained the same dimension as pure PCA from Table x with 100% variance. However, as the value of *P* and *R* increase, the effectiveness of PCA is less. For instance, 70% PCA variance on LBP (*P*: 8, *R*: 1) is 158 features while 166 features obtained from LBP (*P*: 16, *R*: 2) on the same variance.

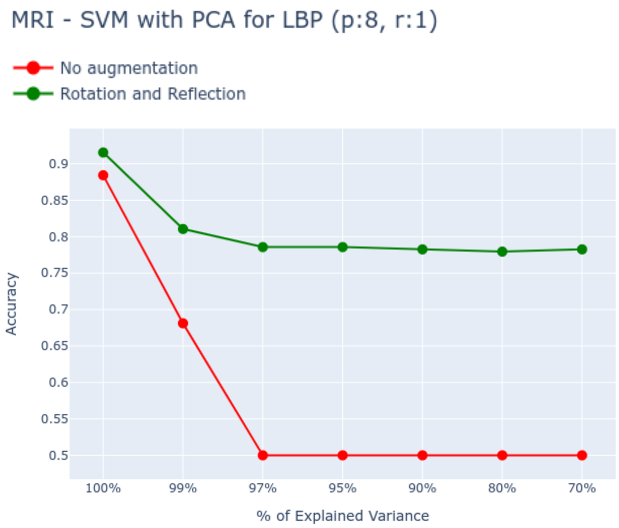


Figure 6: SVM with PCA for LBP (*P*: 8, *R*: 1)

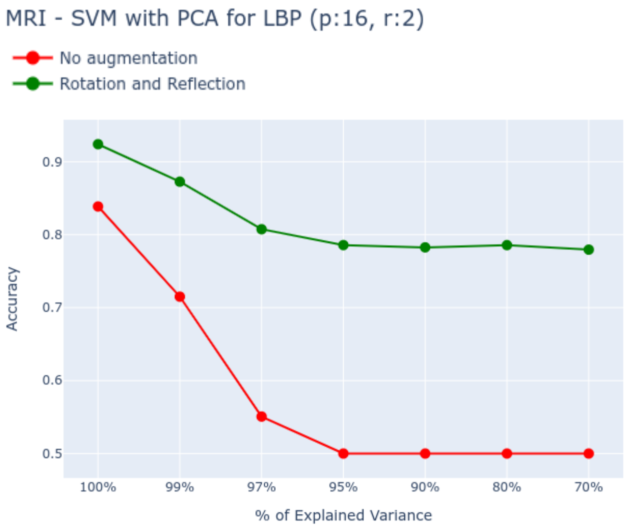


Figure 6.1:SVM with PCA for LBP (*P*: 16, *R*: 2)

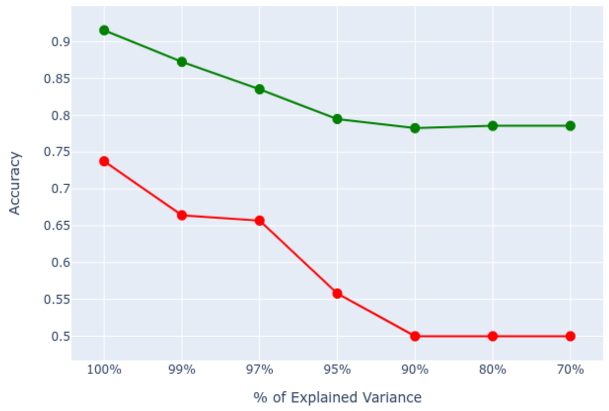


Figure 6.2: SVM with PCA for LBP (*P*: 24, *R*: 3)

Based on the experiment, the rotation and reflection outperforms the no image augmentation. The best performance is PCA at 100% variance for LBP (*P*: 16, *R*: 2) results 92.40% accuracy. One interesting observation is that the less the variance on PCA, the poor the performance of the model are. For instance, when no image augmentation applied, for the PCA variance range between 100% to 97%, the model's performance tends to be better, and 97% to 70% PCA variance the models tend to perform poorly.

3. Convolutional Neural Network (CNN)

a) Initial CNN Creation

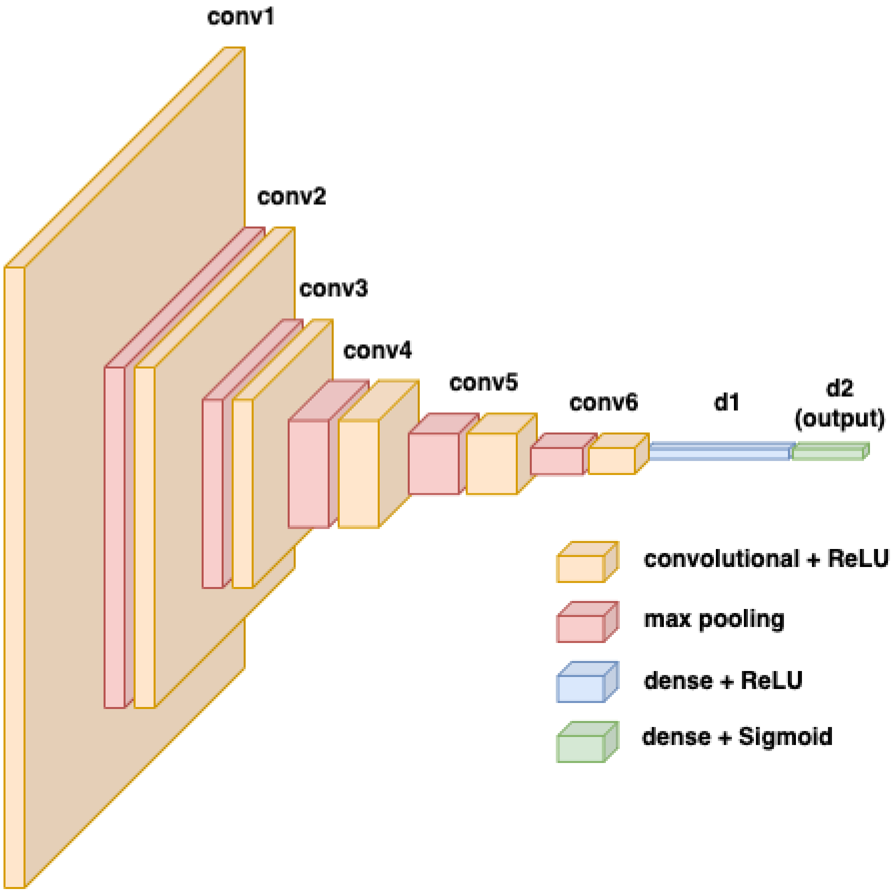


Figure 7: MRI CNN Structure

The initial CNN structure was based on Tensorflow's documentation [9] on image classification, where the number of convolutional layers was increased till the accuracy performance felt it was not improving the model's accuracy for classification. Every addition of a convolutional layer was accompanied by a max pooling and dropout except for the last convolution layer connecting to the dense layer. This is how the initial CNN structure was established. The dropout utilized for this model was 20% after every max pooling layer. Every layer utilized ReLU as an activation function with the final dense layer containing a Sigmoid activation function for the binary classification as portrayed in Figure 7.

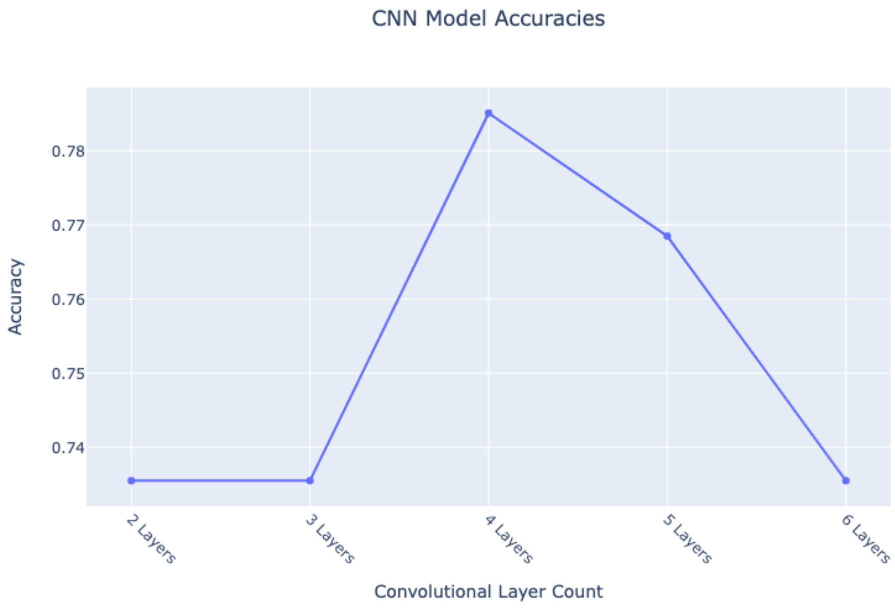


Figure 8: Initial CNN Experiment performance

When referring to Figure 8 for the results of this experiment, it's noted that at convolutional layer count 4 it performs the best with an accuracy of 75.51%, when going passed 4 layers, the accuracy starts to lessen.

b) Transfer Learning

To start with transfer learning, first the model needs to be selected to perform transfer learning from. Looking at Kang et al's [10] paper working in a similar domain of image classification, while DenseNet169 did not outperform their top methodology, they did make a case that it had merit in performing well within this classification domain so this was the network that transfer learning will be performed on. To fine tune the model, we will be varying the amount of trainable layers that the model will have. Tensorflow's Densenet169 contains 695 trainable layers in total, with fine tuning we will be looking to freeze the weights of the top layers while looking to train the weights of the bottom layers. The experimentation seeks to evaluate the optimal amount of trainable layers for the transfer learning model.

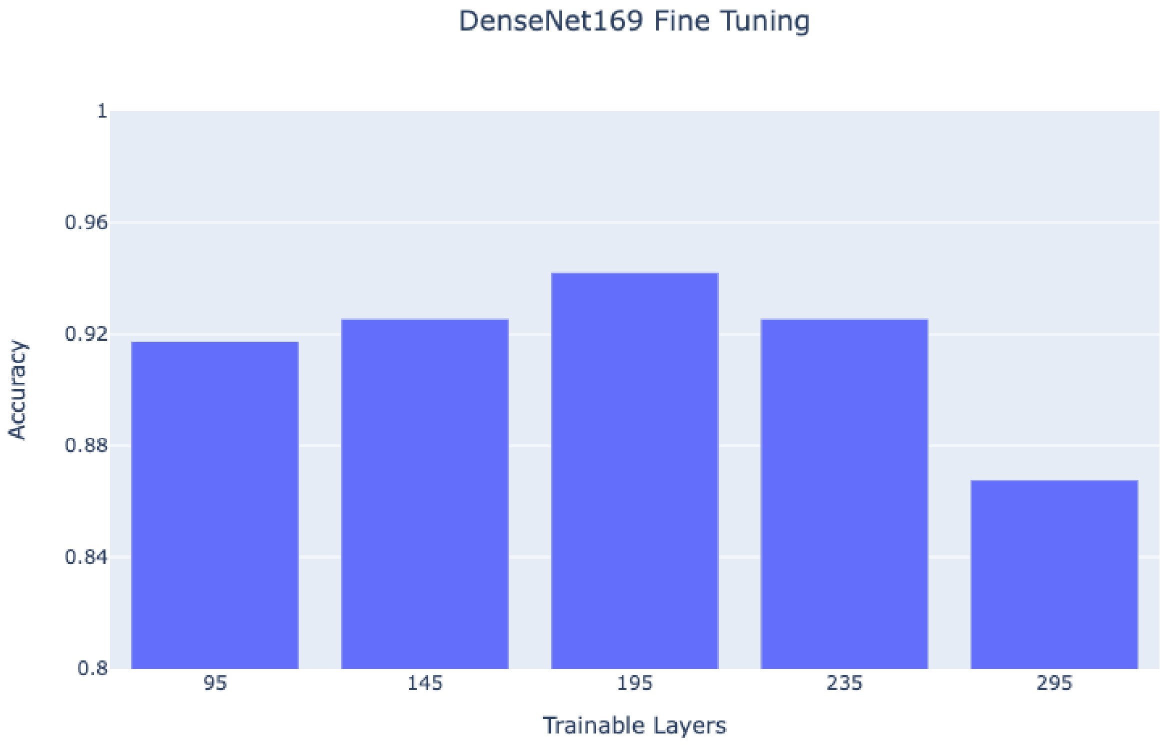


Figure 9: DenseNet169 Fine Tuning

Referring to figure 9 with the results of the experiment, 195 trainable layers performs the best, where having more trainable layers starts to decrease the performance. This could be due to the fact that as we increase the amount of layers to be trained, there are more parameters needed to be tuned, causing for overfitting of the data. This overfitting worsens the performance of the model. With the proposed 195 trainable layers with DenseNet169, we resulted in a performance of 94.21% accuracy.

c) Data Augmentation

Utilizing the horizontal flip and 90 degree rotation either clockwise or counterclockwise was then performed on the top performing model of previous experiments. The purpose of augmentation is to increase the training size of the model, allowing it to be trained with more information in the hopes to benefit the accuracy performance. Before the augmentation was performed, figure 9 portrays the confusion matrix of the previous best performing model, densenet169 with 195 trainable layers.

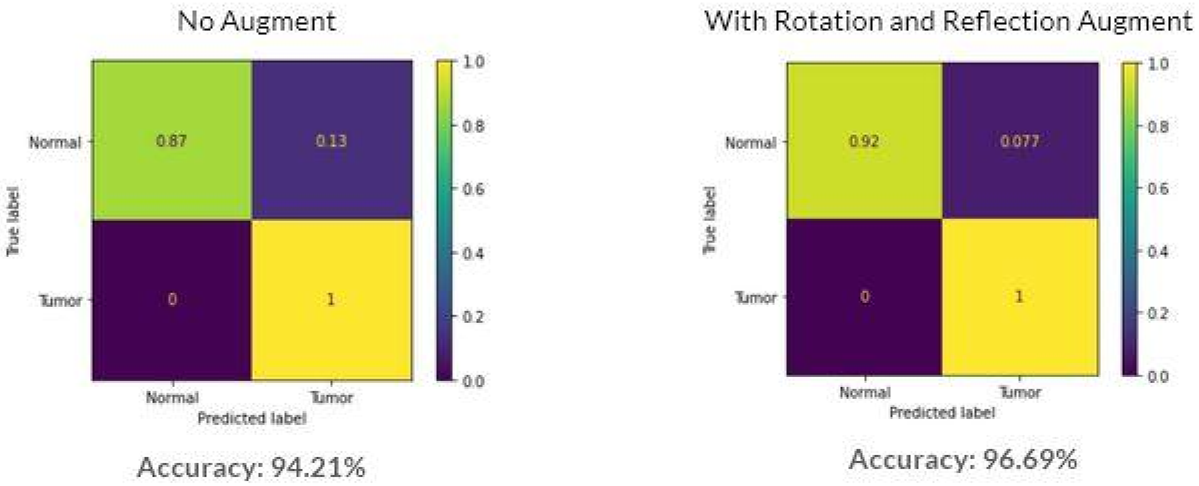


Figure 10: Pre and Post Augmentation Confusion Matrix

Once the augmentation is applied to the dataset the normal class improves by 0.05 which is roughly 2 images, pushing the overall accuracy from 94.21% to 96.69%. While it's a very minor improvement, the initial model already performed very well leaving very little improvement to be achieved.

B. FER 2013 Classification

The dataset utilized for emotion recognition was located on kaggle [11] containing a total of 32298 images, 48x48 pixel black and white images, where 4953 images belong to "Angry" class, 547 images belong to "Disgust" class, 5121 images belong to "Fear" class, 8989 images belong to "Happy" class, 6198 images belong to "Neutral" class, 6077 images belong to "Sad" class, and 4002 images belong to "Surprise" class (Figure 12)

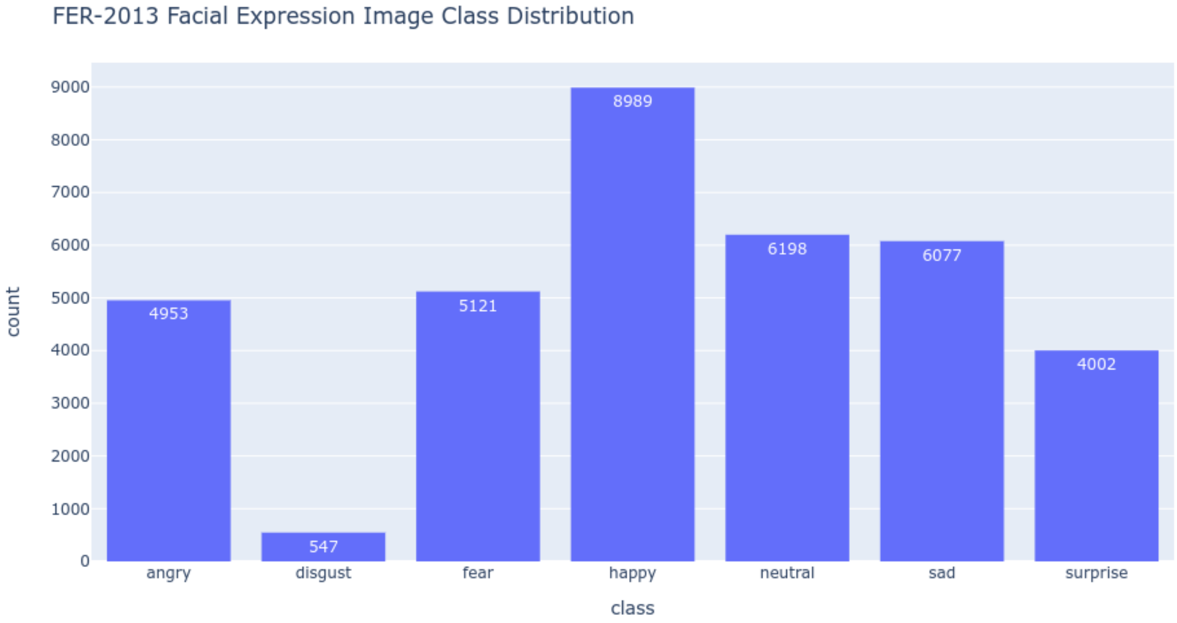


Figure 12: Class Distribution of FER-2013 Dataset.

1. Image Augmentation: Reflection and Translation

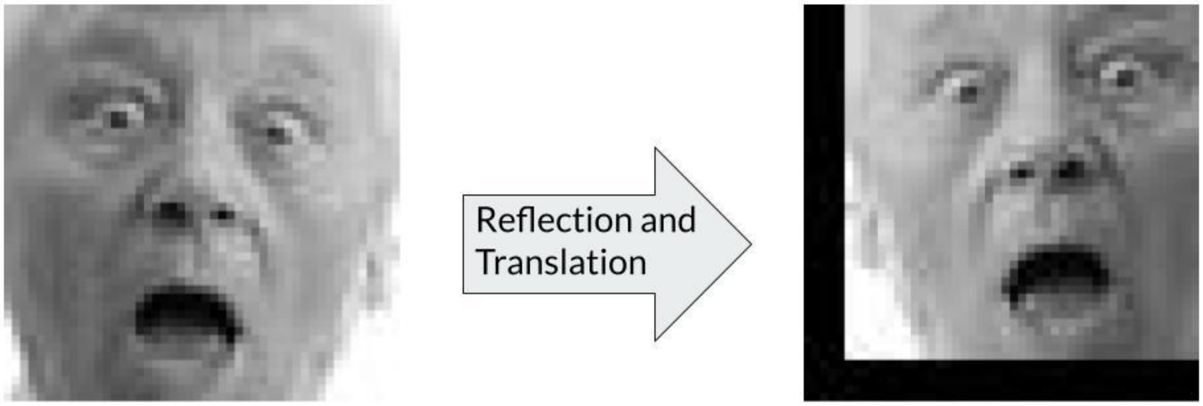


Figure 12: Reflection and Translation Augmentation

Based on [12] one of the impactful methodologies for augmenting data of emotion recognition is translation and reflection. For these data set including the augmentation performed on the MRI set, another set of augmentation was done where the images were translated 5 pixels in one of the 8 compass directions (due to the image being original 48x48 pixels a small translation was done) randomly having the shited image be padded with black pixels for the missing pixels in the image after the shift, and then a horizontal reflection was performed on the shifted image to generate the augmented dataset as displayed in figure 11.

2. Support Vector Machine (SVM)

a) Initial SVM

For the initial experiment on the FER-2013 dataset, we experimented how the model would perform without any features and feature selection techniques. After flattening each image into 1D raw pixel data, we obtained 2304 features (48x48) per image. The same process is applied to the three sets: no image augmentation, rotation and reflection, and reflection and translation, and the results are shown in figure 13.

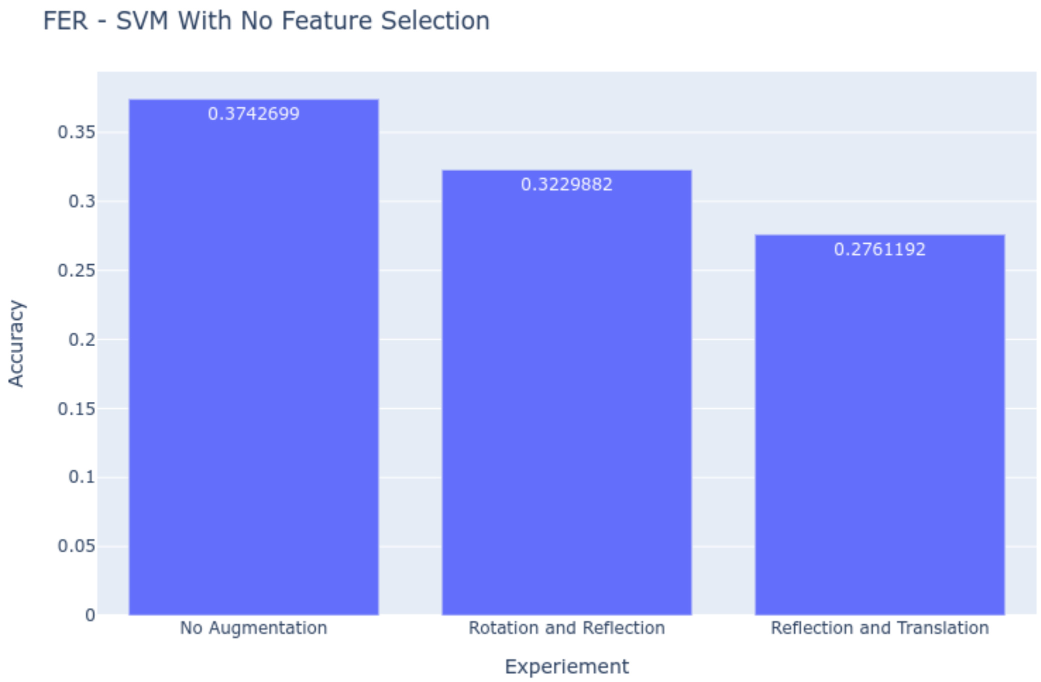


Figure 13: SVM with Raw Pixel Data

After training the SVM with three different datasets, the best performance is no image augmentation with a result of 37.42% accuracy. While the rotation and reflection and reflection and translation accuracy drop around 5% one after the other.

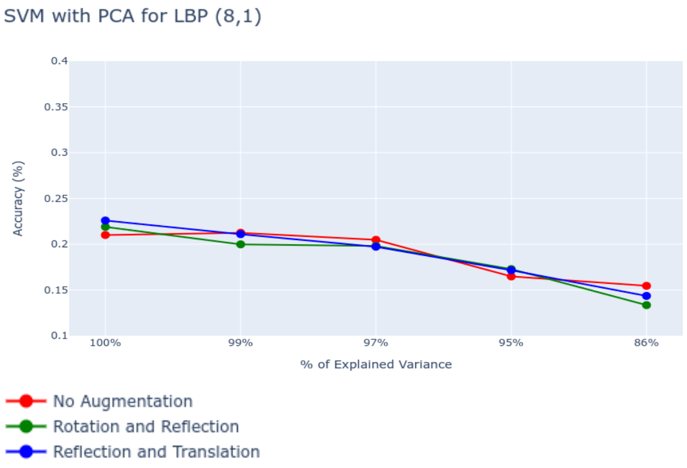
b) PCA

For the second SVM experiment for the FER-2013 dataset, we applied the PCA on three datasets (no augmentation, rotation and reflection, and reflection and translation). The following variance is what we selected from the cumulative variance and utilized for the experiment: 100%, 99%, 97%, 95%, 90%, 80%, and 70%. After applying the PCA to the raw pixel data, the dimension of the features are reduced as shown in the following table:

Raw Pixel Features (without PCA)	PCA Variance	LBP (*P*: 8, *R*: 1)	LBP (*P*: 16, *R*: 2)	LBP (*P*: 24, *R*: 3)
2304	100%	10	18	26
	99%	8	13	18
	97%	7	10	13
	95%	4	6	7
	86%	2	2	2

Table 3:: Raw Pixel Data and PCA Features for FER-2013 dataset

As we can see after the LBP extracts the handcrafted features, the dimension is reduced dramatically. For example, the LBP combination of (*P*: 8, *R*: 1) with 100% variance only has 10 features, while 86% variance only has two features.



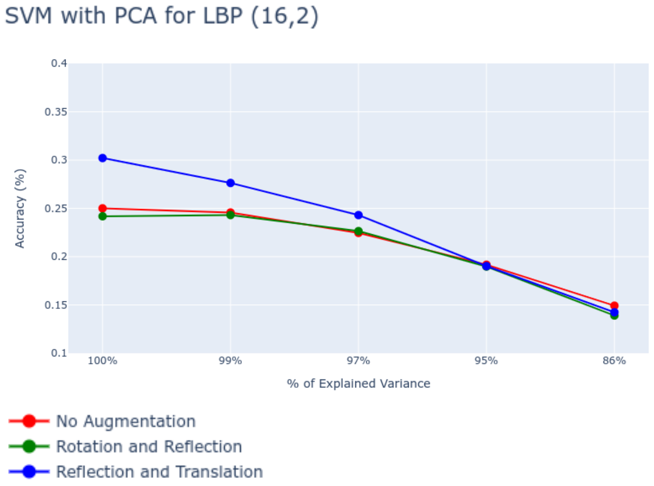


Figure 14: FER2013 with PCA for LBP (*P*: 8, *R*: 1)

Figure 14.1: FER2013 with PCA for LBP (*P*: 16, *R*: 2)

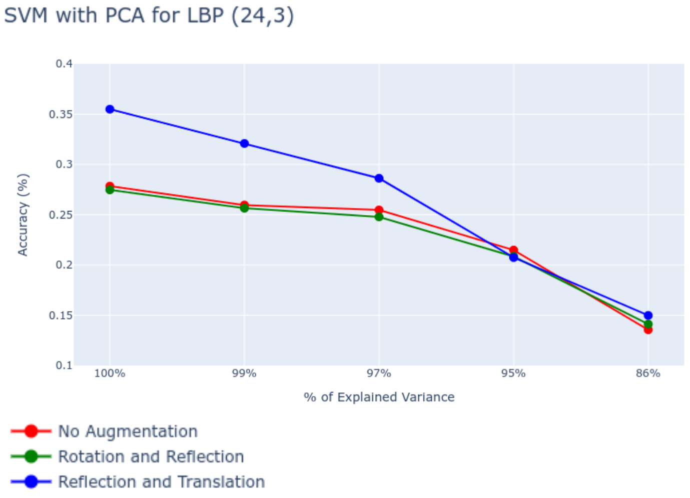


Figure 14.2: FER2013 with PCA for LBP (*P*: 24, *R*: 3)

After applying PCA with different *P* and *R* in LBP features, we observed that reflection and translation helps the model to improve the performance. While on the other hand, the rotation and reflection did not help to improve the model's performance. In fact, in some of the cases the rotation and reflection actually drop the model's performance. For instance, 99% variance for LBP (*P*: 8, *R*: 1), no image augmentation performs better than rotation and reflection. As a result, out of all LBP and PCA experiments for the FER-2013 dataset, the best performance is 100% PCA variance with LBP (*P*: 24, *R*: 3), which results in 35.50% accuracy.

c) HoG

For the third experiment, we examine the model's behavior by utilizing features from HoG. In our experiment setup, we utilized 16x16 for the pixel per cell and 9 for the orientation, which is inline with the paper that we found [4]. The HoG feature extraction is applied on the three datasets (no image augmentation, rotation and reflection, and reflection and translation) with the following result shown in figure 15.

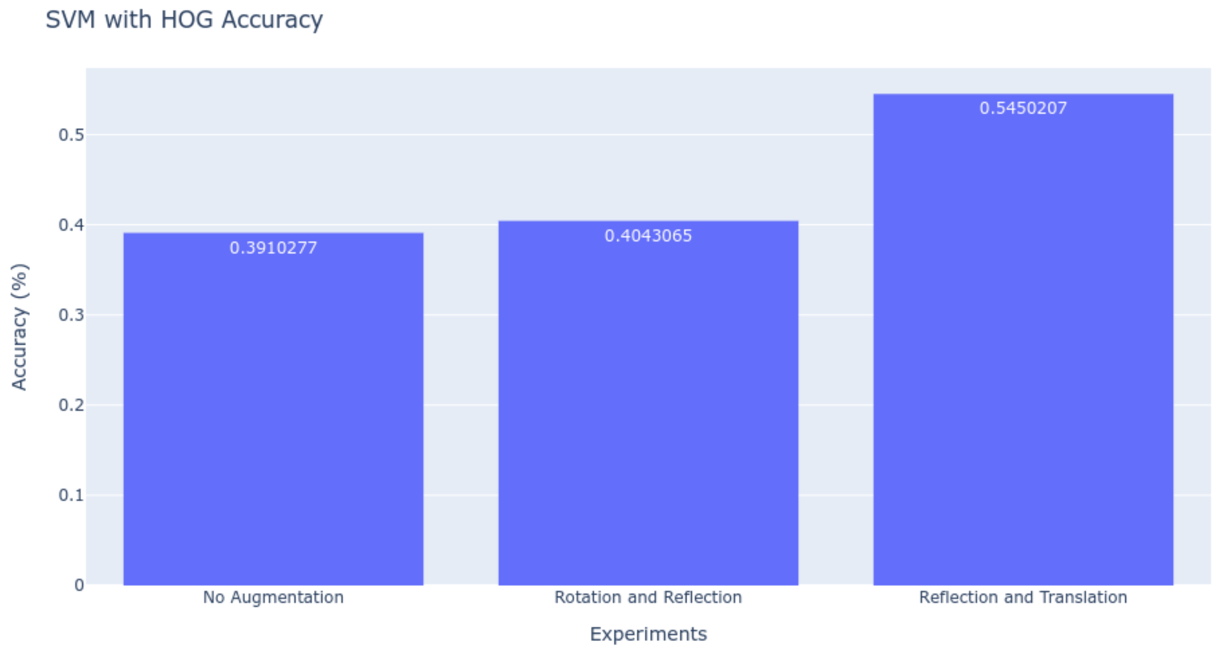


Figure 15: SVM Performance on HoG Features

As we can see, reflection and translation improved the model's performance significantly by around 14% accuracy. While on the other hand, the rotation and reflection from the HoG feature did not improve the model's performance significantly. As a result, the best model's performance was 54.50% accuracy.

d) SIFT

The last experiment for the FER-2013 dataset under SVM utilizes the SIFT feature. After the SIFT feature extracted from the images, we applied K-Means as the Bag-of-Words approach to clustering the features. Unfortunately, the features extracted from SIFT are different numbers for each image. As a result, we could not utilize the elbow method to find the optimal *K* for the

K-Means prior the algorithm, and we exam the effectiveness of *K* by looking at the model's performance as shown infigurex.

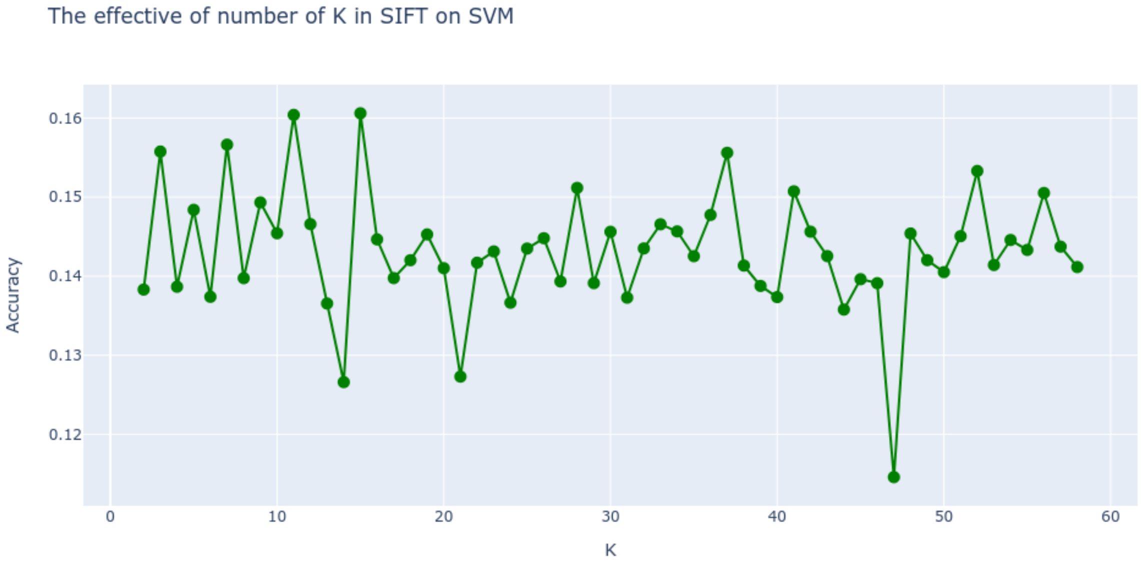


Figure 16: The Effective of K in SIFT Impacts on SVM

Based on the experimental results, we found out that the highest performance is at $K = 15$. With that setup, we applied the SIFT algorithm to extract features on the three datasets (no augmentation, rotation and reflection, and reflection and translation) we have and the results are shown below infigurex.

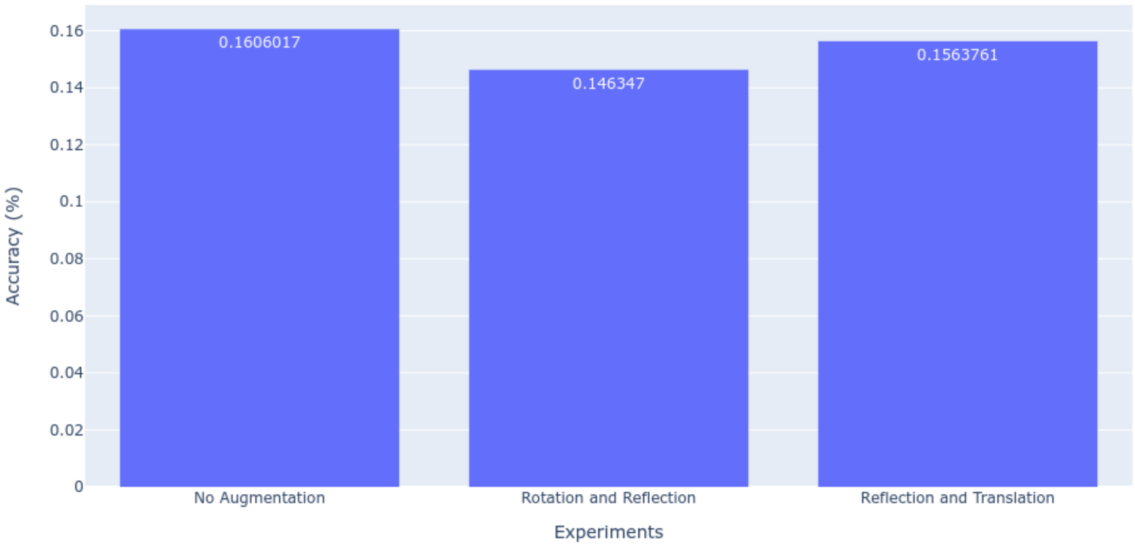


Figure 17: SVM with SIFT Features

As we can see, the data augmentation did not help to improve the model's performance. In fact, it decreased the model's performance. As a result, the best performance model in this experiment is no image augmentation achieved 16.06% accuracy.

3. Convolutional Neural Network (CNN)

a) Initial CNN

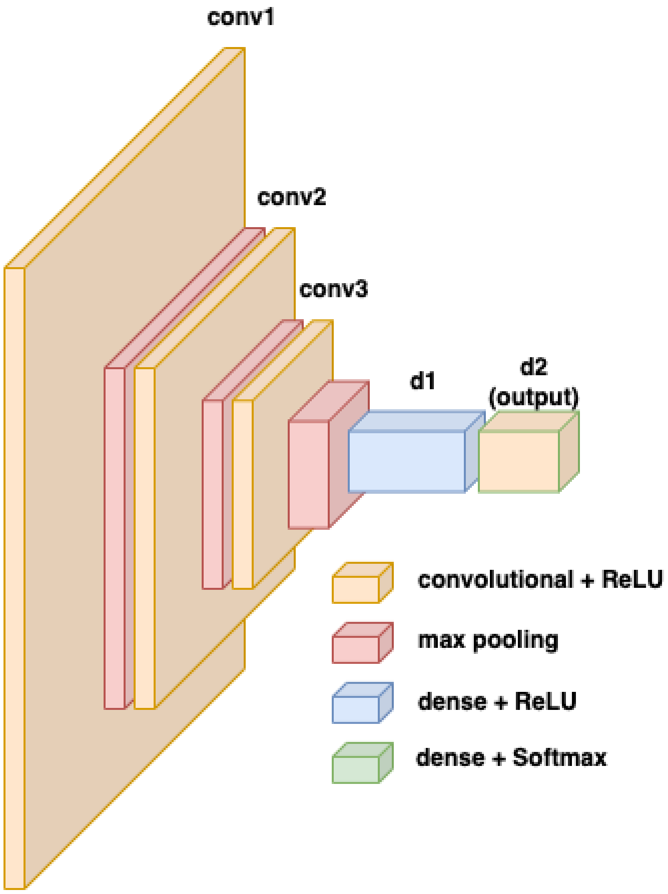


Figure 18: Initial CNN Architecture For FER 2013 Dataset Classification

The initial CNN structure was similar to that of the one create in the MRI experimentation however no experiments outside of a single CNN was conducted due to the performance of the initial CNN within the medical data experimentation being worse than the transfer learning experimentation, as well as papers [13] and [14] performed transfer learning due to the explanation that the complexity of the problem would require a large amount of data to train a successful model from scratch. As a result, only a single CNN was established to be viewed as the baseline CNN with no techniques utilized. The architecture of the CNN is found in figure 18. This architecture yielded a 54.35% accuracy based on the best fold during stratified k-fold.

b) Transfer Learning

Transfer learning performed with this dataset utilized two models, VGG-16 and ResNet50. The reason for these two models is their prominence in papers such as [15] and [16] where VGG-16 seems to perform rather well within the domain of the dataset being worked on. Similar to the MRI transfer learning section, each model will be fine tuned on variations of the number of bottom trainable layers while freezing the remaining top layers. This is to maintain knowledge of the pretrained models while adjusting it to the current task at hand. The VGG 16 that tensor flow provides has 19 trainable layers while the ResNet50 network contains 175 trainable layers in total. Similar to how transfer learning was handled in the MRI set, the experimentation was done testing various amounts of frozen and trainable layers finding the optimal amount of layers for each respective model.

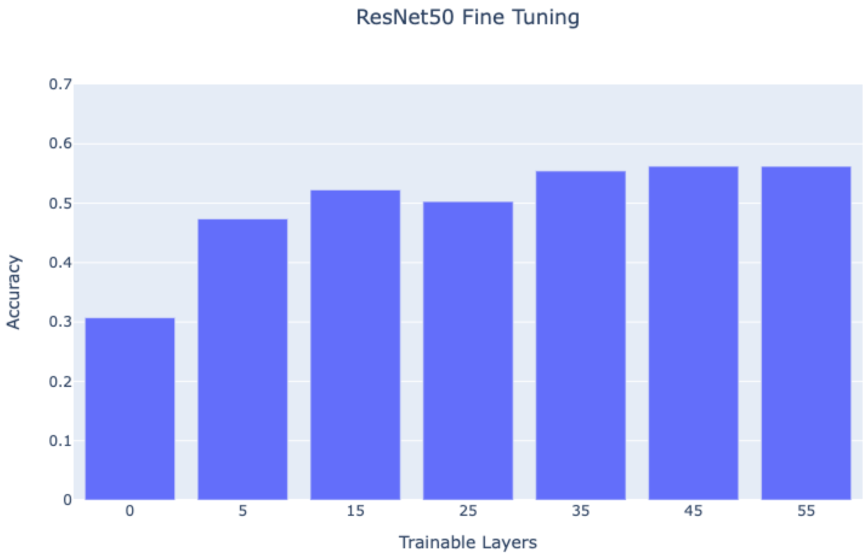


Figure 19: ResNet50 Fine tuning

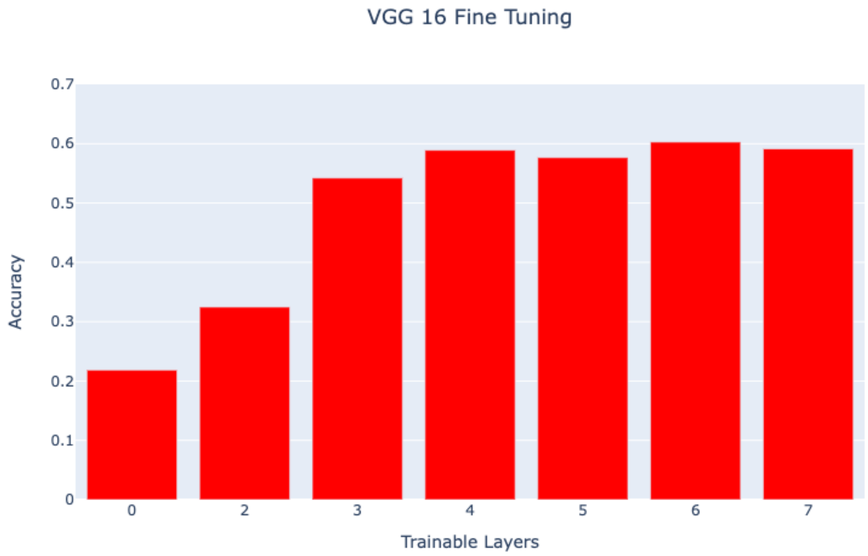


Figure 20: VGG 16 Fine Tuning

Referring to figure 19 the ResNet50 training started to plateau after 35 trainable layers with very little improvement having the best performance being 56.23% at 45 trainable layers. The VGG16 fine tuning had a similar behavior of plateauing to occur around 5 trainable, where the best performing structure was with 6 trainable layers, outperforming the ResNet50's best performance with an accuracy of 60.28%.

c) Data Augmentation

Similar augmentation with horizontal reflection and 90 degree rotation was performed on the dataset to supply more data to train the transfer learning model. However shocking results are found where the data augmentation actually harmed the performance of the model instead of enhancing it compared to how the MRI experimentation reacted to the augmentation.

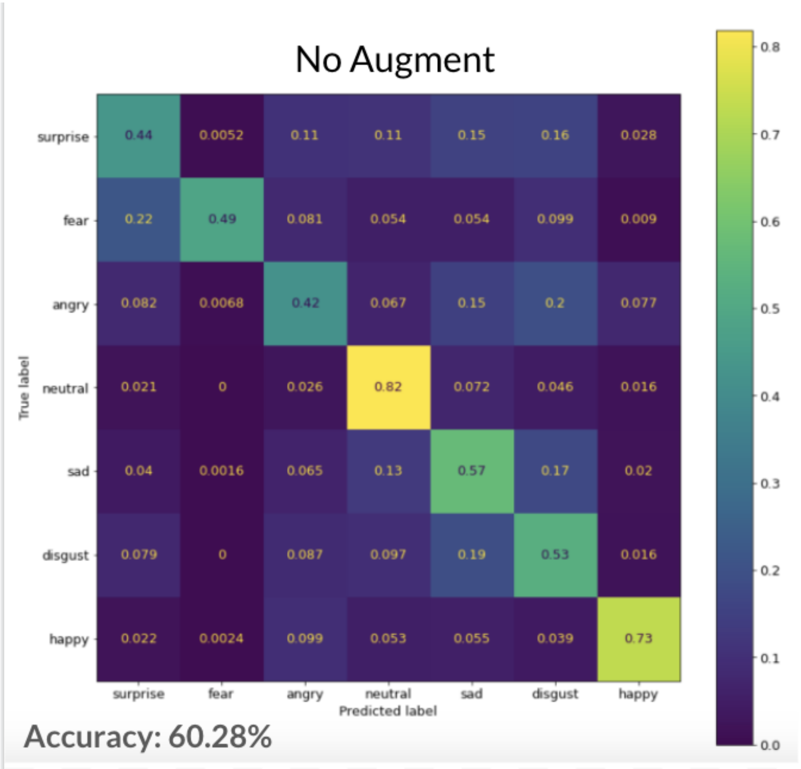


Figure 20: Confusion Matrix of Accuracy Without Augmentation

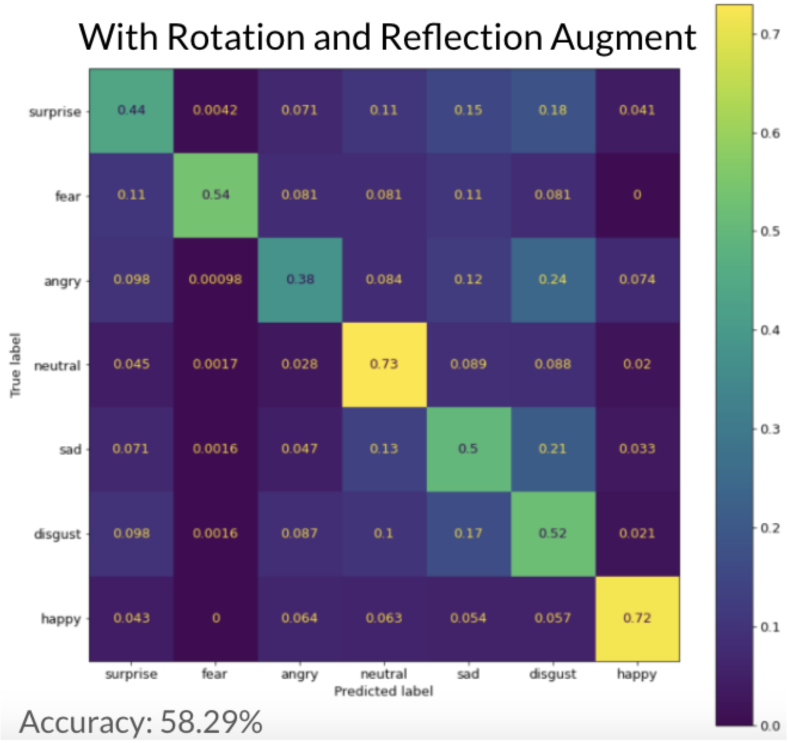


Figure 20.1 :Confusion Matrix of Accuracy With Augmentation

When analyzing the confusion matrices displayed in figures 20 and 20.1 it's shown that the utilization of data augmentation actually harmed the performance for all class accuracies except for fear, where the overall accuracy worsened from the initial 60.28% utilizing VGG-16 to now 58.29%.

d) Image Resizing

Based on the previous experiment not much change occurred when performing data augmentation, as a result a new direction was taken by increasing the dimension of the data image. The original dataset has a size of 48x48 pixels, and according to [16] this dimensionality is small for CNN's including those that are performing transfer learning, to extract features from an image. As a result based on their findings, the images were resized to 224x224 pixels to see if this would impact the accuracy of the model.

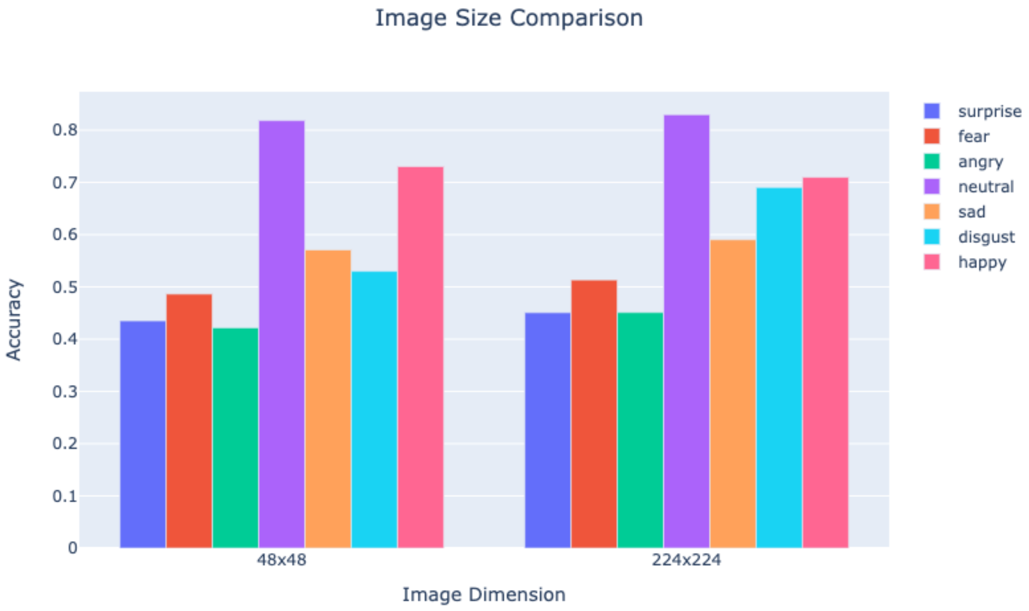


Figure 21: Per Class Performance utilizing 48 pixels vs 224 pixels

Analyzing Figure 21 we can see that nearly all classes improved on performance when the size of the images were increased, this could be due to the acknowledgement of features not observed before when doing the class distinction. The overall accuracy of VGG-16 transfer learning improved from 60.28% to 64.11% accuracy.

e) Application of New Data Augmentation

When referring to [4], one aspect stood out which is the idea that not all augmentations are beneficial based on the domain. Through research, [11] explores some of the best augmentation methods for the FER dataset to be translation and reflection. With this experiment we evaluated how the model with resized imaging would interact with the original augmentation of rotation and reflection, as well as utilizing the translation and reflection.

Method	Accuracy
No Augmentation	64.11%
Rotation and Reflection	62.52%
Translation and Reflection	64.43%

Table 4: Augmentation Methods and their Accuracy

When referring to table 4 for the colluding results of the experimentation, the rotation and reflection augmentation that was performed similar to the MRI again had the harmful impact on accuracy, worsening it by over 1%, similar to earlier in experimentation. However when applying the translation and reflection, the accuracy slightly improved by 0.32%, while it's a slight margin, it improved the accuracy instead of harming it when utilized.

V. Results and Discussion

A. MRI SVM

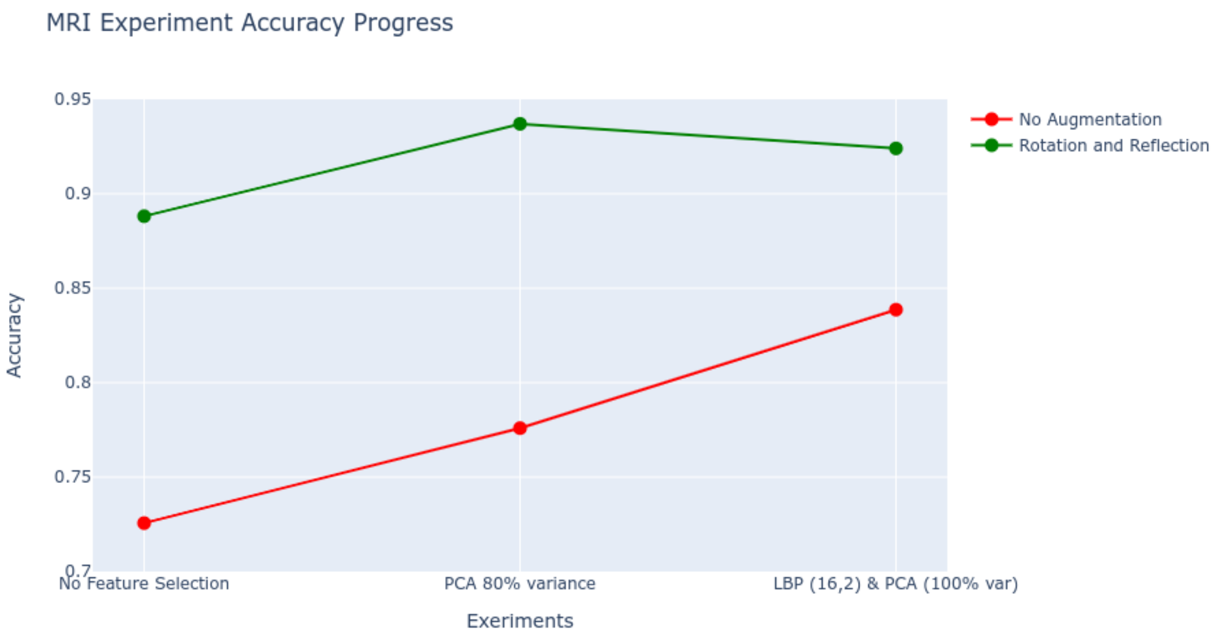


Figure 22: MRI SVM Experiment Accuracy Progress

Throughout the entire MRI SVM experiments, we can clearly see that feature selection with PCA and handcraft features with LBP helped to improve the model's performance compared to no feature or feature selection. In particular, the rotation and reflection improved the model's performance significantly. As a result, the best formance SVM model on MRI dataset is PCA with 80% variance (77 features) on rotation and reflection, which results in 93.70% accuracy. The SVM model detail is as follows: kernel = RBF, C = 3.0, Gamma = 0.003636364.

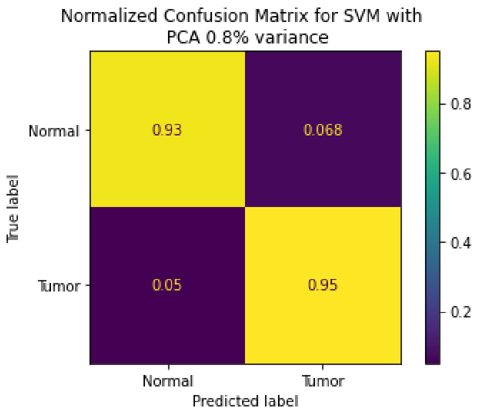


Figure 23: Post Augmentation Confusion Matrix for the Best SVM Model with MRI Dataset

Based on figure 23, we can see that the model is able to classify more Tumor class than the Normal class by 2% accuracy. This phenomenon is better than the other way around because we do not want to wrongly detect a tumor patient as a normal patient in the real world.

B. MRI CNN

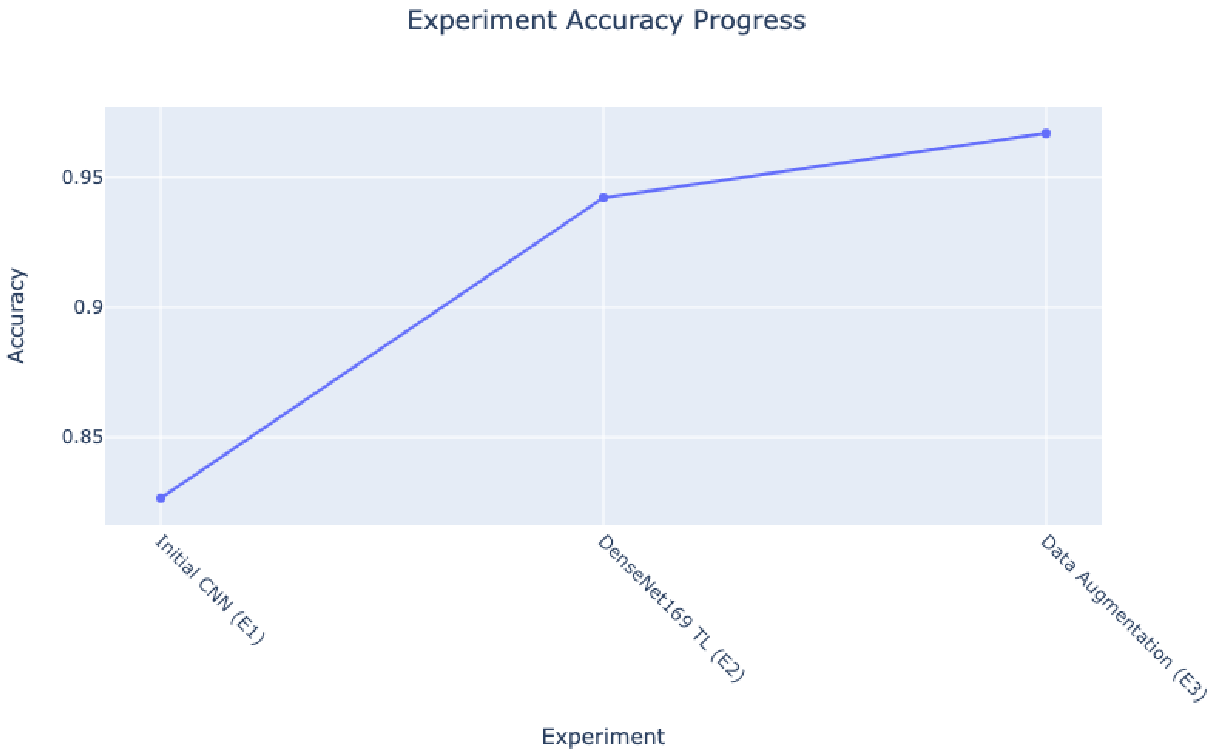


Figure 24: MRI CNN Experiment Progression

Through the utilization from an initial CNN to utilizing transferred learning to lastly applying data augmentation had an improvement every step of the way as highlighted in figure 24. The initial CNN performed with an accuracy of 75.51% after determining that the best amount of convolutional layers in the architecture to be 4 layers. When experimenting with transfer learning utilizing densenet 169 we concluded that utilizing 195 trainable layers would be the best performing, where the accuracy was 94.21% pushing the accuracy to improve over 19% from the initial cnn. The last experimentation through utilizing data augmentation with rotation and reflection pushed the accuracy to 96.69% which was the best accuracy of the CNN experimentation.

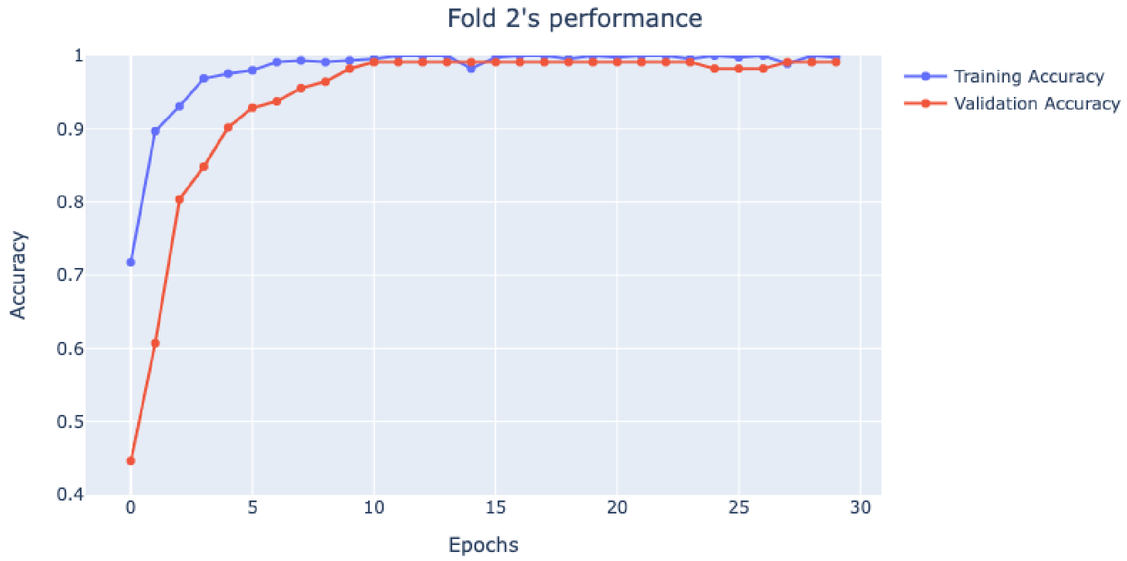


Figure 25: MRI CNN Best Model Training and Validation

Referring to figure 25, it highlights that the training of the model did not exhibit overfitting, having the validation accuracy match the training accuracy of 100%. This performance was observed on the second fold from the five fold cross validation.

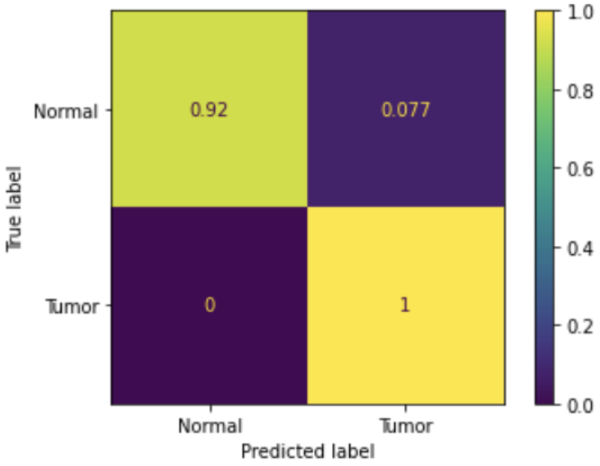


Figure 26: MRI CNN Best Model Confusion Matrix

When looking at the confusion matrix (figure 26) we can see that the tumor class which was the majority class had a 100% accuracy where the normal class had a 92.3% accuracy.

C. MRI Discussion

When comparing the SVM without any features or feature selection technique with CNN utilizing no techniques, SVM achieved 88.81% accuracy while the CNN achieved 78.51% accuracy. However, once the CNN utilized transfer learning and image augmentation, CNN's performance surpassed the SVM model utilizing PCA feature selection and augmentation with an accuracy of 96.69%.

D. FER SVM

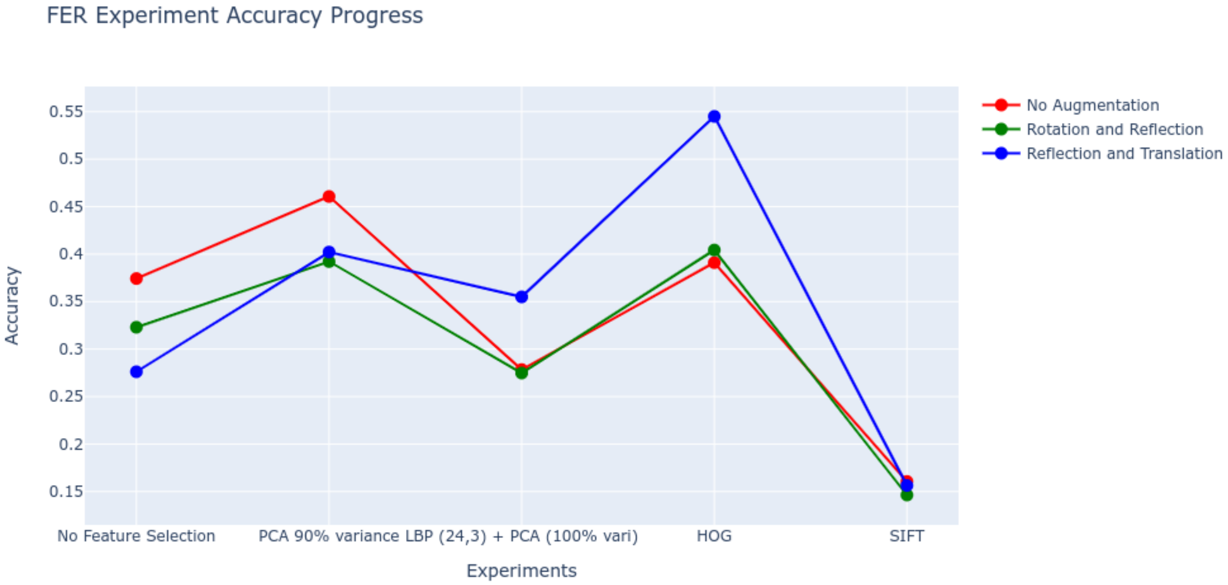
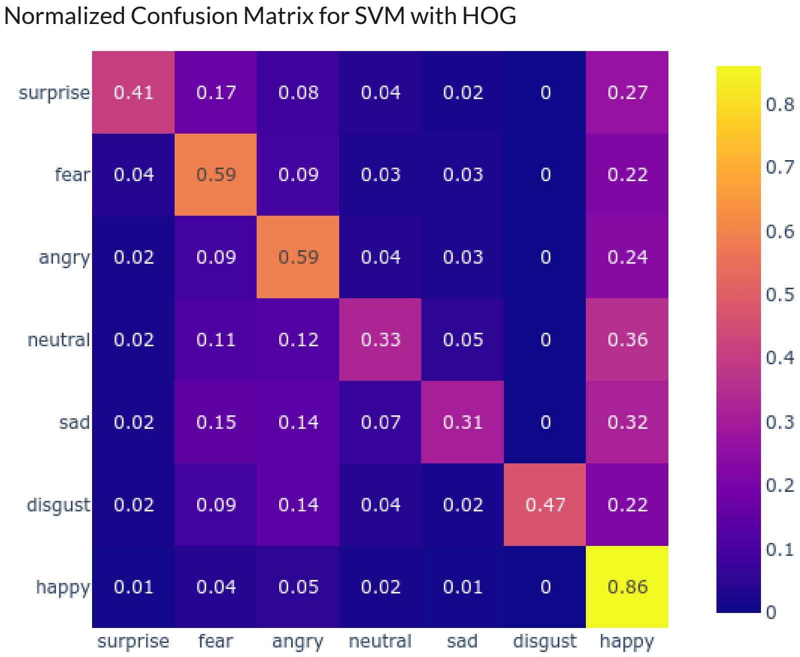


Figure 27: SVM Experiment Accuracy Progress on FER2013 Dataset

Based on figure 27, we can clearly see that reflection and translation really improved the model performance when utilizing HoG features by around 15% accuracy. On the other hand, reflection and translation made the model performance worse when we have raw pixel data. Also, the SIFT feature performed poorly throughout all experiments. As a result, the best performance model is HoG with rotation and reflection at 54.50% accuracy, and the model detail as follows: kernel = Polynomial, C = 1.0, Degree = 5, Decision Function = one-vs-rest, and apply the class weights.



	surprise	fear	angry	neutral	sad	disgust	happy
surprise	0.41	0.17	0.08	0.04	0.02	0	0.27
fear	0.04	0.59	0.09	0.03	0.03	0	0.22
angry	0.02	0.09	0.59	0.04	0.03	0	0.24
neutral	0.02	0.11	0.12	0.33	0.05	0	0.36
sad	0.02	0.15	0.14	0.07	0.31	0	0.32
disgust	0.02	0.09	0.14	0.04	0.02	0.47	0.22
happy	0.01	0.04	0.05	0.02	0.01	0	0.86

Figure 28: Post Augmentation Confusion Matrix on SVM with HoG Features

Examining the confusion matrix on figure 28, perclass accuracy for “disgust” class is 47% outperformed the “sad” class, which is significantly since the “disgust” label is minority class in the dataset. However, the class label “sad” and “neutral” per class accuracy dropped with the sacrifice of boosting “disgust” class accuracy. In addition, we can clearly see that “happy” class has the highest class accuracy this because it has highest number of samples in the dataset.

E. FER CNN

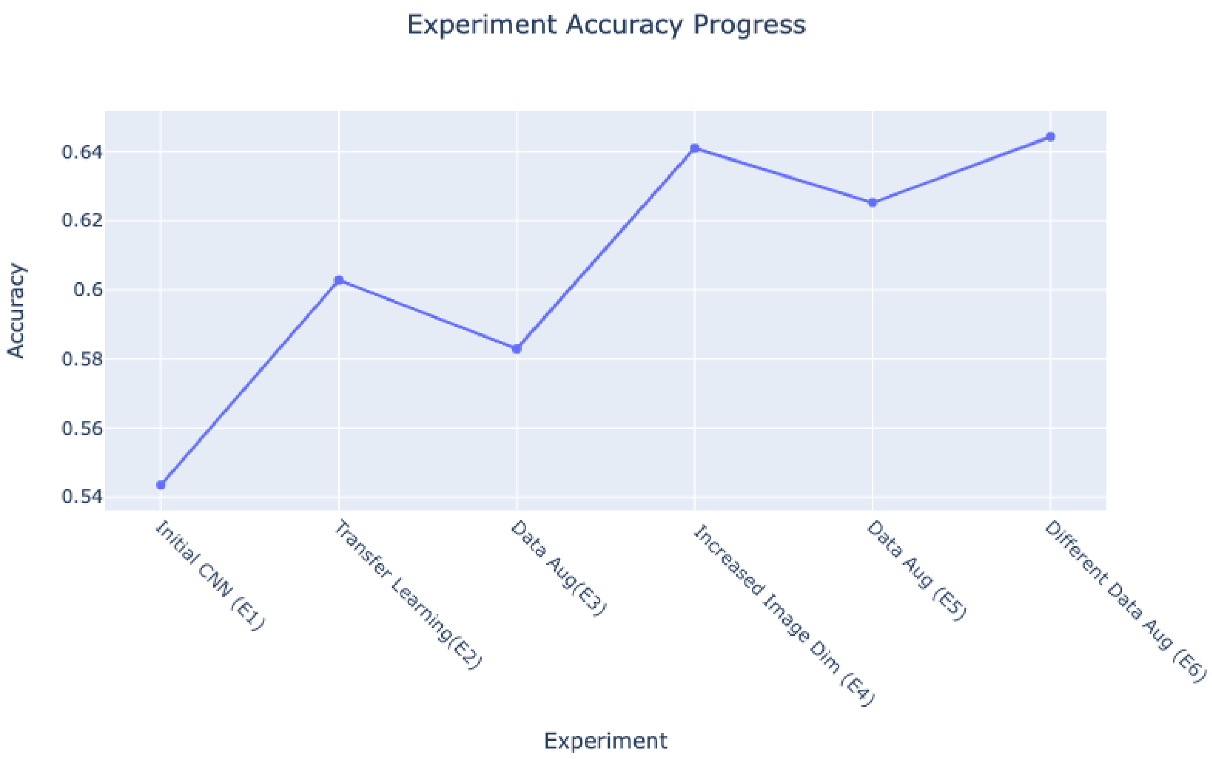


Figure 29 FER CNN Experiment Progress

The key takeaway from Figure 29 is the hiccups in improving the accuracy of the model when we tried to introduce “Data Aug” which was the rotation and reflection augmentation introduced for the MRI dataset. This highlights what [4] mentioned, discussing that certain augmentations will not benefit all domains of classification, calling for a new type of augmentation to be utilized, highlighted with “Different Data Aug”. Two of the largest jumps in accuracy were going from the initial CNN to transfer learning and from the transfer learning structure to increasing the dimension size. The initial CNN structure was with 3 convolution layers performing an accuracy of 54.35%. When performing transfer learning, the best performing CNN was VGG-16 utilizing 5 trainable layers, where the performance surpassed the 60% mark performing at a 60.28% accuracy. The modification of image size to 224x224 pixels allowed for the performance to increase roughly 4% to 64.11%. The final and best performing iteration was when VGG-16 was with 5 trainable layers, the images were resized to 224x224 pixels, and the data augmentation of reflection and translation utilized, having the final performance to be 64.48% accuracy.

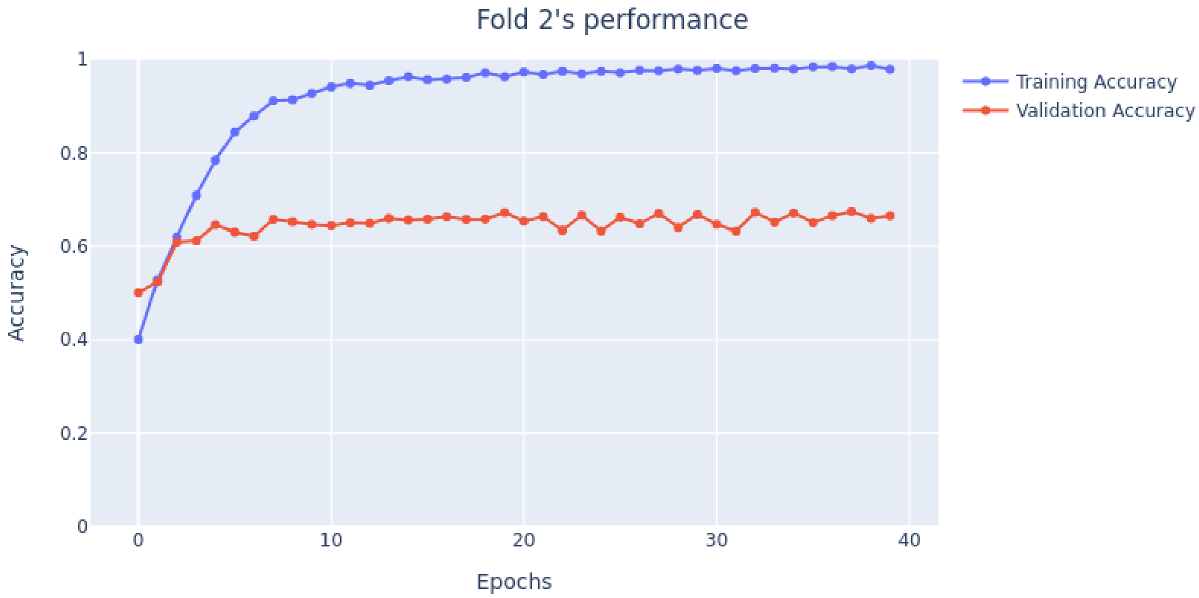
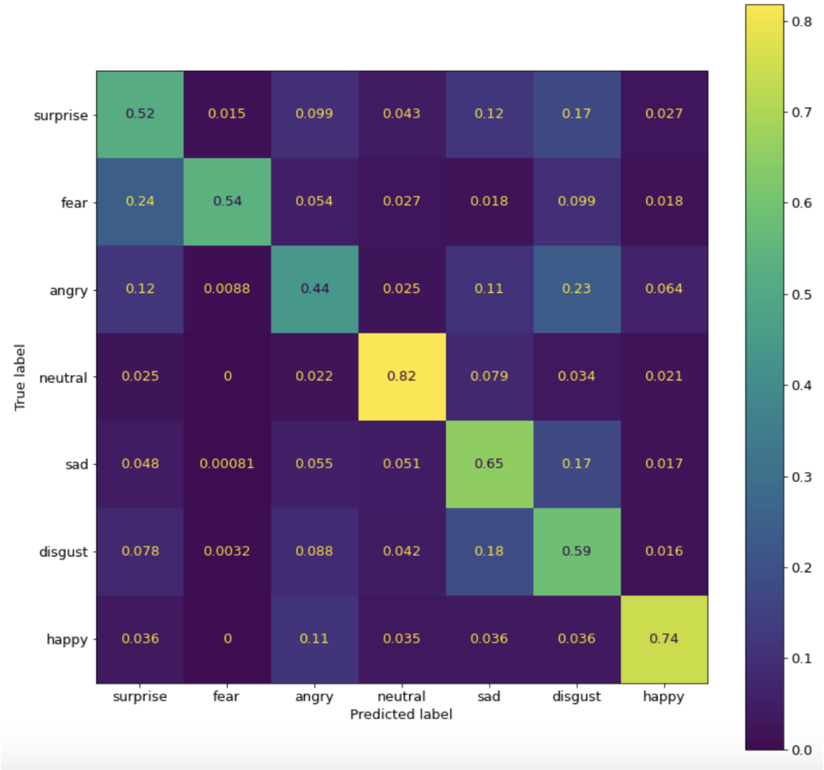


Figure 30: FER CNN Best Model Training and Validation

Unlike the MRI training curve, the one performed by the best performing model never had the validation reach the accuracy of the training set. This trend seemed to be common when comparing this result to other training and validation curves such as in [17]



True label \ Predicted label	surprise	fear	angry	neutral	sad	disgust	happy
surprise	0.52	0.015	0.099	0.043	0.12	0.17	0.027
fear	0.24	0.54	0.054	0.027	0.018	0.099	0.018
angry	0.12	0.0088	0.44	0.025	0.11	0.23	0.064
neutral	0.025	0	0.022	0.82	0.079	0.034	0.021
sad	0.048	0.00081	0.055	0.051	0.65	0.17	0.017
disgust	0.078	0.0032	0.088	0.042	0.18	0.59	0.016
happy	0.036	0	0.11	0.035	0.036	0.036	0.74

Figure 31: FER CNN Best Model Confusion Matrix

Looking at the confusion matrix presented in figure 31, the best performing class was neutral which had the largest amount of images within the dataset (nearly 9000 images). The worst performing was angry, where the classification model seemed to misclassify angry images with surprise, sad, and disgust. The reason for the minor class not being the worst performing is due to the application of class weights when training the model to make it so all classes are valued equally when trying to classify. Due to this, it could be the reason for the performance being hindered trying to classify all images equally instead of having a bias to majority classes.

F. FER Discussion

When comparing the initial CNN with the initial SVM performance, the CNN performs at an accuracy of 54.35% accuracy while the SVM performs at an accuracy of 37.43%. When the SVM applies feature selection and data augmentation is when it surpasses the initial CNN performance to an accuracy of 54.50%. Utilizing transfer learning alone allowed for the CNN to perform better than SVM's best performing model with an accuracy of 60.28%. The CNN due to requiring no augmentation means that it performed better than the SVM with less data presented to the model.

VI. Summary

Throughout the entire project, we conclude that CNN greatly benefits from transfer learning, being able to outperform SVM utilizing handcrafted features or feature selections and data augmentation. At the same time, we found out that applying image augmentation is a way to enrich the training data and boost the model performance. However, the same augmentation method for one specific domain may not be applicable to another domain's problem. For instance, the same image augmentation method for the MRI dataset (medical domain) does not work for the FER2013 dataset (facial emotion domain). While the PCA works well on raw pixel data, but is less effective on handcrafted features such as LBP, and this is true for both medical and emotional data. The other handcrafted features such as LBP, HoG, and SIFT, end up performing worse.

VII. Challenges

For the entire project, we have three major challenges. Firstly, the heavy computation required for the FER2013 dataset. We did not realize the amount of computation resources required for training the SVM and CNN at the beginning until one iteration of the cross validation fold took around 2 hours. Therefore, we transferred from Google Colab to Kaggle Notebook and scikit-learn to Nvidia Rapids in order to have and maximize the GPU acceleration. After the utilization of GPU, we benchmarked the performance and the results are follow:

1. Reduced the amount of time of PCA to process on average from 8 mins to 2 mins 12 sec.
2. Reduced time per epoch when training CNNs on average from 5 mins to 40 sec.

Secondly, the issue of Nvidia Rapids environment setup. Due to inconsistent package versioning, some of the Nvidia Rapids dependencies will be unable to recognize the virtual environment packages. As a result, the necessary packages were missing. Credit to Ashwin Srinath, Software Engineer at Nvidia Rapids Team, for fixing a bug that we reported [18] and helped us to set up the environment. Lastly, the large quantity of factors to account for when optimizing classifier accuracy such that:

1. CNN needs to look more into the amount of trainable parameters, optimizers, learning rate scheduler, layer structure, and regularization.
2. SVM needs to look more into the amount of trainable parameters and regularization.

VIII. Future Works

Future experimentation for these datasets would be to explore the methodologies of balancing the dataset. One of the noticeable aspects with the FER 2013 dataset is that disgust had a very small amount of images and happy had a very large amount of images. This disparity could lead to models having bias towards the majority class. While methods to tackle this include training the model with class weights, we would like to explore balancing methods to see if it would outperform training a model with class weights. The methods of balancing to explore include upsampling, downsampling, SMOTE, and utilizing generative adversarial networks (GAN). Another avenue to be explored is to look into video data, a field with even less available data where the data would contain dynamic visual and audio information, expanding from the current static images worked on within the project.

Project Site: https://hchen98.github.io/dtsc870/

Project Source Code: https://github.com/hchen98/dtsc870

Reference

[1] M. K. Nalini and K. R. Radhika, "Comparative analysis of deep network models through transfer learning," 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2020, pp. 1007-1012, doi: 10.1109/I-SMAC49090.2020.9243469.

[2] Sachdeva, J., Kumar, V., Gupta, I., Khandelwal, N. and Ahuja, C., 2016. A package-SFERCB-“Segmentation, feature extraction, reduction and classification analysis by both SVM and ANN for brain tumors”. Applied Soft Computing, 47, pp.151-167.

[3] R. Ravi, S. V. Yadhukrishna and R. prithviraj, "A Face Expression Recognition Using CNN & LBP," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020, pp. 684-689, doi: 10.1109/ICCMC48092.2020.ICCMC-000127.

[4] Kalsum, T., Mehmood, Z., Kulsoom, F., Chaudhry, H., Khan, A., Rashid, M. and Saba, T., 2021. Localization and classification of human facial emotions using local intensity order pattern and shape-based texture features. Journal of Intelligent &amp; Fuzzy Systems, 40(5), pp.9311-9331.

[5] X. Tang, C. Zhou, L. Chen and Y. Wen, "Enhancing Medical Image Classification via Augmentation-based Pre-training," 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2021, pp. 1538-1541, doi: 10.1109/BIBM52615.2021.9669817.

[6] N. B. Thota and D. Umma Reddy, "Improving the Accuracy of Diabetic Retinopathy Severity Classification with Transfer Learning," 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), 2020, pp. 1003-1006, doi: 10.1109/MWSCAS48704.2020.9184473.

[7] A. Junaidi, J. Lasama, F. D. Adhinata and A. R. Iskandar, "Image Classification for Egg Incubator using Transfer Learning of VGG16 and VGG19," 2021 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT), 2021, pp. 324-328, doi: 10.1109/COMNETSAT53002.2021.9530826.

[8] A. M. Hashan, “MRI based brain tumor images,” Kaggle, 04-Apr-2021. Available: https://www.kaggle.com/mhantor/mri-based-brain-tumor-images.

[9] “Image classification | TensorFlow Core.” TensorFlow. https://www.tensorflow.org/tutorials/images/classification.

[10] M. K. Nalini and K. R. Radhika, "Comparative analysis of deep network models through transfer learning," 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2020, pp. 1007-1012, doi: 10.1109/I-SMAC49090.2020.9243469.

[11] M. Sambare, “Fer-2013,” Kaggle, 19-Jul-2020. Available: https://www.kaggle.com/msambare/fer2013.

[12] S. Porcu, A. Floris, and L. Atzori, “Evaluation of Data Augmentation Techniques for Facial Expression Recognition Systems,” Electronics, vol. 9, no. 11, p. 1892, Nov. 2020, doi: 10.3390/electronics9111892

[13] J. L. Joseph and S. P. Mathew, "Facial Expression Recognition for the Blind Using Deep Learning," 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON), 2021, pp. 1-5, doi: 10.1109/GUCON50781.2021.9574035.

[14] N. S. Abdulsattar and M. N. Hussain, "Facial Expression Recognition using Transfer Learning and Fine-tuning Strategies: A Comparative Study," 2022 International Conference on Computer Science and Software Engineering (CSASE), 2022, pp. 101-106, doi: 10.1109/CSASE51777.2022.9759754.

[15] G. C. Porușniuc, F. Leon, R. Timofte and C. Miron, "Convolutional Neural Networks Architectures for Facial Expression Recognition," 2019 E-Health and Bioengineering Conference (EHB), 2019, pp. 1-6, doi: 10.1109/EHB47216.2019.8969930.

[16] J. Luo, Z. Xie, F. Zhu and X. Zhu, "Facial Expression Recognition using Machine Learning models in FER2013," 2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC), 2021, pp. 231-235, doi: 10.1109/ICFTIC54370.2021.9647334.

[17] B. G. K. Reddy, P. Yashwanthsaai, A. R. Raja, A. Jagarlamudi, N. Leeladhar and T. T. Kumar, "Emotion Recognition Based on Convolutional Neural Network (CNN)," 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), 2021, pp. 1-5, doi: 10.1109/ICAECA52838.2021.9675688.

[18] GitHub. (n.d.). Error with ‘Expected 48 from C header, got 40 from PyObject’ · Issue #10187 · rapidsai/cudf. [online] Available at: https://github.com/rapidsai/cudf/issues/10187 [Accessed 19 May 2022].