Human facial emotion recognition using Convolution Neural Networks

Praneeth Kumar Chityala and Sesha Sai krishna Valluri

pkchitya@iu.edu , svallur@iu.edu

Luddy School of Informatics and Computer Science, Indiana University Bloomington, IN, USA

***Abstract* - Face detection has been around for ages, but recent advancements in the technology paved the way for better human emotion recognition models. Accurately predicting human emotions is essential for synthetic intelligence systems to emulate and gauge feelings from facial expressions. In this study, we explore, analyse multiple existing and custom built CNN architectures and rigourously fine tune their hyper parameters to improve model performance.**

Keywords – Human emotion recognition, convolutional neural network (CNN), model hyperparameter fine-tuning, LeNet, AlexNet, VGG-16, Custom CNN

I. INTRODUCTION

Facial expressions are an important nonverbal communication medium for humans. Different facial expressions are indicators of different feelings, and it allows a human being to express his/her current emotional state. The task of predicting human facial emotions has been quite challenging for major part due to the limitations of existing traditional classification models, like KNN, SVM, and the limited computing capabilities. The recent developments in deep neural networks and convolutional neural networks and the GPU processing capabilities have taken the task of classifying and recognition to whole another plateau.

Convolutional Neural Network (CNN) is a class of artificial neural network, predominantly used for image analysis. Since inception, CNN architectures have gone through rapid evolution and in recent years have achieved results which were previously considered possible only via human execution/intervention. Depending on the task at hand, and the corresponding constraints, a wide variety of architectures are available today ranging from simpler models like Shallow CNN, LeNet to more complex architectures like AlexNet, VGG and ResNet.

In this work, we aim to improve prediction accuracy on FER2013 using CNNs. We initially consider shallow CNNs and fine tune their parameters to check for the best accuracy model (baseline) and compare more complex CNNs with the baseline accuracies to study relative performance. We construct various experiments to explore different optimization algorithms, learning rate schedulers, activation functions. We thoroughly tune the model and training hyperparameters to build better performance models.

II. RELATED WORK

During, 1980s to 1990s, with the introduction of Neocognitron and LeNets, CNNs have shown great capabilities in image processing. A typical CNN has a convolutional layer, a pooling layer, and a fully connected layer. This makes it efficient in extracting different features at each level. Even though CNNs showed great potential, their popularity tapered off in the next decade due to lack of sufficient training data and computing power. It was only after the 2010s, the growth of computing power and the collection of larger datasets made CNNs a much more viable tool in feature extraction and image classification. These CNNs were able to process and analyze huge data with decent accuracies.

The recent years saw various methods being developed to boost the performance even more. Like, the Sigmoid activation function being substituted by Rectified Linear Unit (ReLU) activation. Batch normalization was created to keep gradients from vanishing. various pooling approaches such as average pooling and max pooling have been introduced to reduce the dimensionality of the data and to generalize the data. To avoid overfitting, dropout, regularization, and data augmentation are used. Starting with the Stochastic gradient optimizer, various other variations optimizers like Adam, RMSProp, AdaGrad have been proposed to suit different case scenarios.

With advanced computing power and data gathering capabilities, a large image dataset FER2013, was compiled and it became a benchmark in comparing model performance in emotion recognition. Many CNN variants have achieved remarkable results with a classification accuracy between 65 % and 72.7%. Ensemble models has been shown to improve performance. For instance, Liu et. al. ensembled of 3 CNNs and improved their accuracy by 2.6%. However, to improve the ensemble performance even further, we aim to first optimize the building blocks of these ensembles, a single network.

III. DATA SET

A. Datset Description

We have used the standard FER2013 image dataset for our analysis. This dataset consists of about 35,887 well labelled 48 × 48-pixel gray-scale images of human faces. This dataset has 28,709 images for training and 7,178 images for the testing sets. Testing set is further divided in to PublicTest and PrivateTest images. Each image is categorized into one of the seven classes that express different facial emotions - Angry, Disgust, Fear, Happy, Sad, Surprise and Neutral. The images are processed in such a way that the faces are almost centered, and each face occupies about the same amount of space in each image. Below is the sample image of various emotions in the dataset for reference:

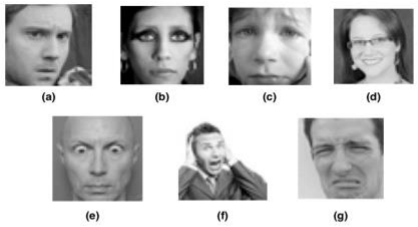
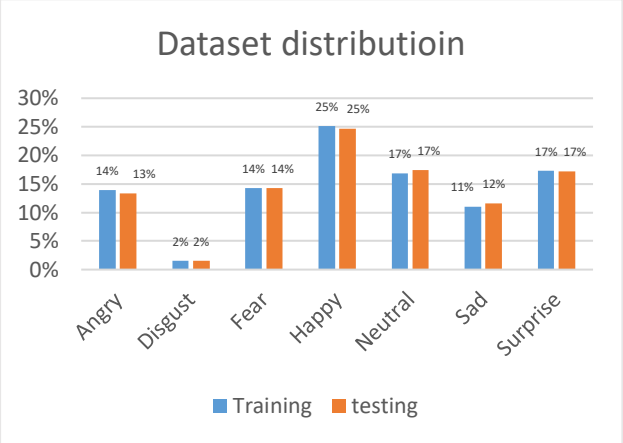


Figure 1: Examples of seven facial emotions that we consider in this classification problem. (a) angry, (b) neutral, (c) sad, (d) happy, (e) surprise, (f) fear, (g) disgust

We noticed that Happy emotion contributed to around 25% of the training and testing. There were very few images for Disgust in both the datasets. We have noticed similiar distribution in the testing set too. Below bar graph shows the distribution of images in training and testing.



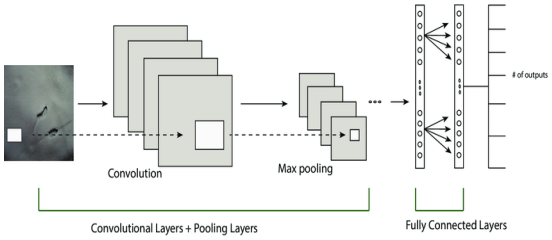
B. Data augmentation:

We have randomly performed rotation (by ± 10 degrees) to about 10% of images and added it to the training set. As FER2013 images are already scaled to to 48×48 and face is well centred into the image, we haven't done any resizing to these images.

IV. MODEL BUILDING

A. Convolutional Neural Networks

Any CNN architecture has four main layers - **Input** – Image input later as arrays or tensors; **CNN** – Convolution layer with integrated convolution, batch normalization, activation and pooling layers. **FC** – Fully connected neural networks. **Output** – Final output prediction layer depending on the number of classes. Below is the base architecture of a CNN model.



B. Computing:

We leveraged the new Mac M1 (MacBook pro) with integrated 8 core GPU, built on System On Chip (SOC) architecture, to train all of our models. The average runtime for a model with 50epochs, 4M parameters and 36,000 input images was around 2 hours.

C. CNN – Model Tuning

Before exploring complex CNN architectures, we analyzed simpler models and fine tuned the models to understand the impact of each of the parameters on the model performance.

Below are a few hyper parameters involved in the CNN architecture:

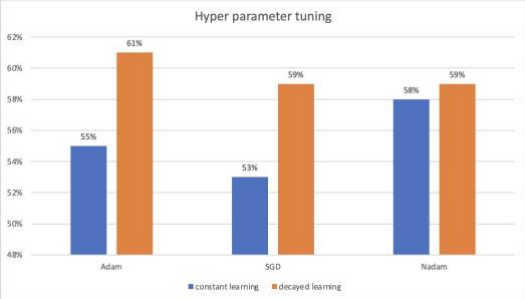
1. *Number of CNN layers and FC layers*
2. *Convolution pooling techniques*
3. *Activation functions*
4. *Learning rate*
5. *Drop out rates*
6. *Batch Size*
7. *Epochs*
8. *Optimizers*
9. *Early stopping criterion*

Grid search is performed to determine the optimal values for the above parameters on convolution and FC layers. Once the architecture has been optimized, we then set up a final experiment to try and fine-tune the trained model's weights and improve its performance.

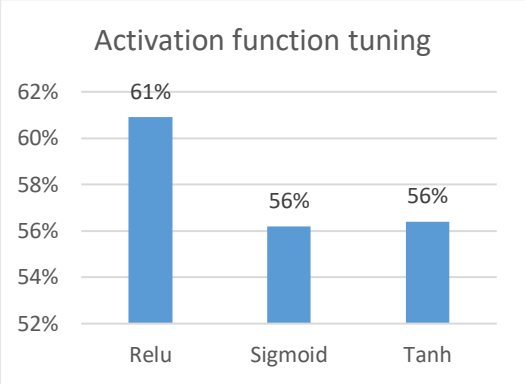
Note: For tuning a parameter, we keep other parameters (tuned) fixed and estimate model performance.

Results of a few crucial parameters tuning are as below:

- **Optimizer:** Our first experiment intends to find the best optimizer in training our architecture. For this, we explore 3 different algorithms: SGD, Adam, and Nadam. The Adam optimizer performed relatively better with 61% accuracy.
- **Learning rates:** We run 2 different variations of this experiment. In the first variation, we run all algorithms with a fixed learning rate of 0.001. This learning rate was determined using a grid search. In the second variation, we set up a simple learning rate scheduler with an initial learning rate of 0.001 and it is reduced by a factor of 0.2 if the validation accuracy plateaus for 5 epochs. The parameters of this scheduler were also determined using a grid search.



- **Activation Function:** We have tried different experiments with some of the popular activation functions and chose ReLu for both the convolution as well as the FC layers.



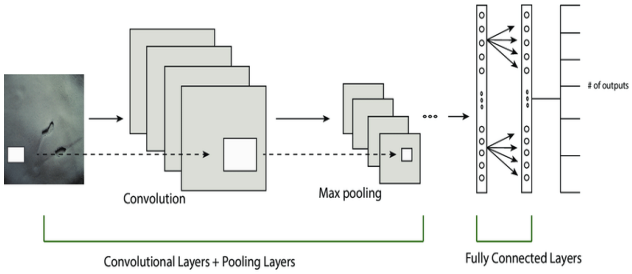
D. Various CNN architectures

Shallow CNNs:

We built a shallow CNN to use as base line. This network has, one convolutional layer and one FC layer.

Convolutional layer: We have 64 3×3 filters, with the stride of size 1, along with batch normalization, with max-pooling and Rectified Linear Unit (ReLU) as the activation function.

Fully connected layer: We have a hidden layer with 512 neurons and SoftMax as the loss function.



Even though the shallow model performed well on the training data, it failed to generalize well for the testing data with ~50% accuracy. We kept this as the baseline for our future models. Any architecture with accuracy below to this base line is neglected.

LeNet and AlexNet CNNs:

We have explored one shallow (LeNet) and one deep (AlexNet) models which performed well on some of the image classifications in other research papers. We wanted to check the effectiveness of these models in FER2013 dataset. Results on the FER2013 gave accuracies of 54% for LeNet and 49% for AlexNet. We noticed problem of overfitting with AlexNet. As we planned to explore Deep Models, outputs of AlexNet gave a good case to configure our custom models with less parameters than AlexNet (which has about 23million trainable parameters).

Custom Deep CNNs:

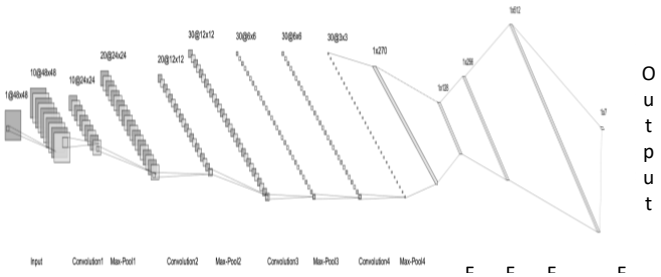
With all the learnings we have had with CNN models and parameter tuning from grid search in our above explorations, we started to build our own custom deep CNN model. We started out with 4 CNN layers. In Fully Connected (FC) layers we explored 2 and 3 layers. We have build about 7 models from this architecture with differet parameters as per our parameter tuning done earlier. We finalized on a 4 CNN + 3 FC architecture which gave out best accuracy of about 66% which is on par with human accuracy of 65%. Below is the description of our model.

7 layer CNN Model (4 Convolutions + 3 FC layers):

4 CNN layers: The first convolutional layer has 64 3×3 filters, the second one has 128 5×5 filters, the third one has 512 3×3 filters and the last one has 512 3×3 filters. In all the convolutional layers, we have a stride of size 1, batch normalization layer, max-pooling layer, dropout and ReLU as the activation function.

3 FC Layers: The hidden layer in first FC layers has 128 neurons, second FC layer has 256 neurons and third layer has 512 neurons. In all FC layers, same as in the convolutional layers, we used batch normalization layer, dropout and ReLU activation. Also, we used SoftMax as

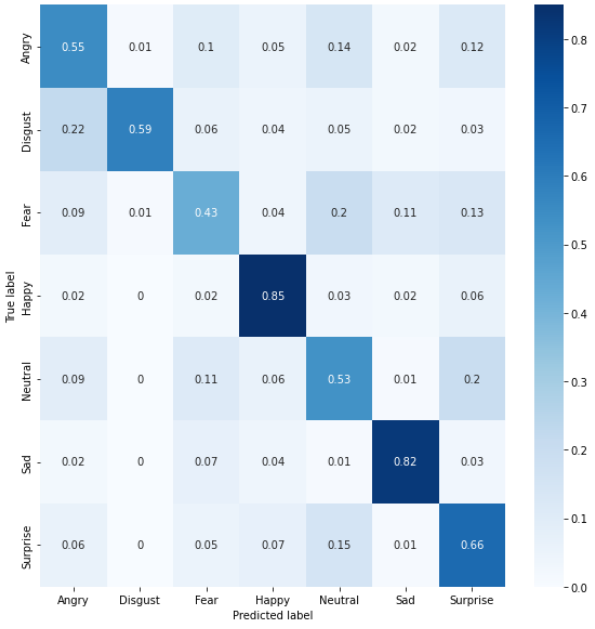
our output activation function and cross entropy as loss funtion.

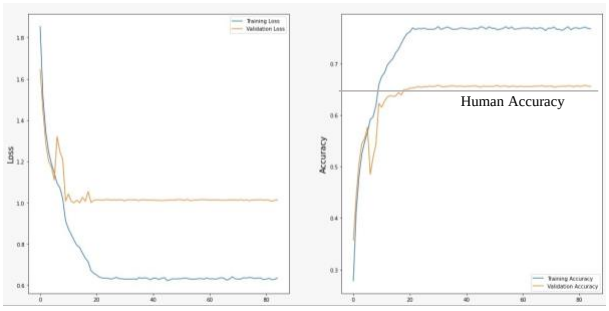


Our custom CNN model has about 4 million trainable parameters which is far less when compared to the 23million trainable parameters of AlexNet as previously explored.

V. RESULTS

After exploring more than 7 custom deep CNN models and 100+ hours of GPU computing our best deep CNN model gave an accuracy of 66.7% which is greater than average human accuracy of 62-65% for FER2013 data set. Below is the loss and accuracy plots of our model along with the confusion matrix. We noticed highest accuracies for sad at 85% and happy at 82% and lowest accuracies for fear at 43%. Highest miss classification came from disgust labled as angry at 22%, this in turn turned out be difficult for even humans to differentiate between disgust and angry given the close relation of the facial expression between them. As we have trained very high number of models we provide the summary of those models below with out final comments to each model.





Sl.No	Model	CNN Layers	Pooling	Activation	FC Layers	Epocs	Train Accuracy	Test Accuracy	Comments
1	Base	1	Average	relu	1	57	84%	52.7%	Overfitting
2	Model1	4	Max	sigmoid	2	15	55%	54%	Model didn't converge
3	Model2	4	Max	sigmoid	2	16	56%	55%	Model didn't converge
4	Model3	4	Max	sigmoid	2	48	66%	62%	Below Human Accuracy
5	Model4	4	Average	relu	3	38	82%	65.4%	Human Accuracy
6	Model5	4	Average	sigmoid	3	48	48%	54%	High training error
7	**Model6**	**4**	**Max**	**relu**	**3**	**82**	**82%**	**66.7%**	**Best CNN Model yet**

VI. CONCLUSION

As a final step we have used OpenCV to provide live facial expression feed and it turned out to be mostly true with our expressions. As noted above our present best model is on par with human accuracy, while state-of-the-art model accuracy at 75%. In order to increase out accuracy we have tried an ensemble model with 5 base shallow CNNs but accuracy was only held at 60% for it.

As next steps to increase the accuracy, we want to focus on better ensemble techniques like boosting and bagging. Once we get the best model we can, we want to use that model and build a small software app which would take live video as input and predict facial emotions real time. For this to happen we need to develop knowledge on the software development side too, which we are trying to focus in our upcoming semesters.

VII. REFERENCES

1. A. Dhall, R. Goecke, J. Joshi, K. Sikka, T. Gedeon, "Emotion recognition in the wild challenge 2014: Baseline, data and protocol," in ICMI 2014 - Proceedings of the 2014 International Conference on Multimodal Interaction, 461-466, 2014, doi:10.1145/2663204.2666275.

2. P.-L. Carrier, A. Courville, Challenges in representation learning: Facial expression recognition challenge, 2013.

3. E. Owusu, E.K. Gavua, Z. Yong-Zhao, "Facial Expression Recognition - A Comprehensive Review," International Journal of Technology and Management Research, 1(4), 29-46, 2020, doi:10.47127/ijtmr. v1i4.36.

4. S. Li, W. Deng, "Deep Facial Expression Recognition: A Survey," IEEE Transactions on Affective Computing, 2020,

doi:10.1109/TAFFC.2020.2981446.

5. K. Liu, M. Zhang, Z. Pan, "Facial Expression Recognition with CNN Ensemble," Proceedings - 2016 International Conference on Cyberworlds, CW 2016, 163-166, 2016, doi:10.1109/CW.2016.34.

6. A. Agrawal, N. Mittal, "Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy," isual Computer, 36(2), 405-412, 2020, doi:10.1007/s00371-019-01630-9.

7. T. Connie, M. Al-Shabi, W.P. Cheah, M. Goh, "Facial expression recognition using a hybrid CNN-SIFT aggregator," Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10607 LNAI, 139-149, 2017, doi:10.1007/978-3-319-69456-6_12.

8. P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar,and I. Matthews, “The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression,” in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops. IEEE, 2010, pp.94–101.