CNN for Facial Expression Recognition

Final Project (SYDE 552)

Dorodi Afroze Krishty
University of Waterloo
dkrishty@uwaterloo.ca

Abstract

Picking up facial expression cues - what is an effortless task for humans is a surprisingly challenging task in the framework of computer vision. Facial expression recognition is a rather complex task for machines to learn and perform autonomously. In this work, we research on an automated facial expression recognition (FER) system, which in recent years has developed into a widely studied problem in the field of image processing. FER has applications in many interesting fields from robotics to biometrics. Over the years, many feature extraction and classification techniques have been researched and developed for FER applications in the domains of machine learning and deep learning. The inspiration behind this work is to understand the implementation of Convolutional Neural Network (CNN)-based architectures, namely VGG on the Facial Expression Recognition (FER)-2013 dataset. In the process we investigate varying topologies of VGG, and the nuances of hyperparameter tuning, which influences the performance of the model and its capability to make predictions. We also use a popular pre-trained network, namely VGG16 and pass our data to the model to investigate transfer learning. Ultimately, we aim to test and improve a CNN that aims to accomplish the task of recognition of seven basic human expressions: 'anger', 'sad', 'happy', 'neutral', 'surprise', 'fear', 'disgust from a set of input images.

Table of Contents

1. Introduction

Facial expression is the common signal for all humans to convey the mood and to emphasize certain aspects of their speech. Facial expressions convey this non-verbal cue that can be an important scope for developing in the area of human-machine interface of automatic facial expression recognition. There have been several attempts at developing an automated system in order to accomplish tasks like face detection, feature extraction techniques and classification of expressions. Deep Learning is a large field of study and its application has found an important place in computer vision. From a high-level there is one method from deep learning that has received massive attention for application in computer vision - it is Convolutional Neural Networks (CNNs) [1]. One of the reasons being that, CNNs are specifically designed for image data. The development of such an automated system can be attempted by a CNN for an automated facial expression recognition system.

1.1 Literature Review

Facial expression recognition is a growing sub-field in image processing this is due to the fact that an automated facial expression analysis tool has potential application in many fields such as human–computer interaction [16], psychiatric observations [17], drunk driver recognition [18] to a lie detector [19]. Since twentieth century, facial expressions have come to be defined as seven prototypical expressions and they are anger, feared, happy, sad, contempt [20], disgust, and surprise [5],[6]. Some techniques have been developed to recognize individual muscle movements induced by an expression on the face [7]. The benchmark psychological framework for describing facial movements by their appearance is called the Facial Action Coding System (FACS) [6]. An expression is typically an accumulation of these facial movements contained in FACS [8]. In addition to FACS, there have been several developments in the machine learning strategies used for facial expression recognition: Bayesian Networks, Multi-level Hidden Markov Model (HMM) [10] [11], Support Vector Machines [12], Histogram of Oriented Gradients (HOG) [13], However, they have shown to achieve lower accuracy compared to deep neural network [14]. Also, now a days, the development in computer vision technniques makes emotion identification much more accurate and accessible to the general population. However, there are still many lingering issues when it comes to facial expression recognition tasks, such as pose appearance with face detection [15]. However, there has been a recent development of a solution for variability in facial pose appearance where they have used three-dimensional pose invariant approach using descriptors [21], [22]. In addition to pose, there are other issues too like excessive makeup [23] and expression [24] that are deemed to make the task more challenging. Recent accomplishments in facial expression detection has contributed towards developments in neurosciences and cognitive sciences [26] that is an important driver in the future advancement of research in the field of facial expression recognition / emotion detection.

The proposed method is based on a CNN framework.

2. Proposed Method

We investigate a baseline model for the FER 2013 dataset [2]. A baseline model will establish the minimum model performance to which all of our other models can be compared, as well as provide the model architecture that we can use as the basis of sour investigation. For the purpose of this research, a VGG model's network topology is referred to as the feature detector part of the model. This is further coupled with a classifier part of the model to interpret features and make predictions, in this scenario, on which of the 7 classes the particular expression in the photo belongs to.

2.1 Network Architecture

At first we test three different neural network topologies and these were VGG-1, VGG-2, VGG-3 (figure 1). The VGG blocks involve stacking of convolutional layers with 3x3 kernels, followed by single Max pooling layer. The number of filters in the convolutional layer were increased with the depth of the network in increments of 2 as 32,64,128,256. Padding was used to ensure the height and the width of the output feature maps matches the inputs. We used ReLU activation function and He weight initialization as best practice. The feature map from the extraction part of the model was flattened with fully connected layers where the final output layer has 7 neurons and softmax activation function, that outputs a prediction. All the VGG-based architecture would have a filters of the size 3x3 with stride of one, Max pooling layers used would also be of size 2x2 and the same stride of one. The baseline model will be optimized with stochastic gradient descent with a modest learning rate of 0.001 with SGD optimiser as a starting point.

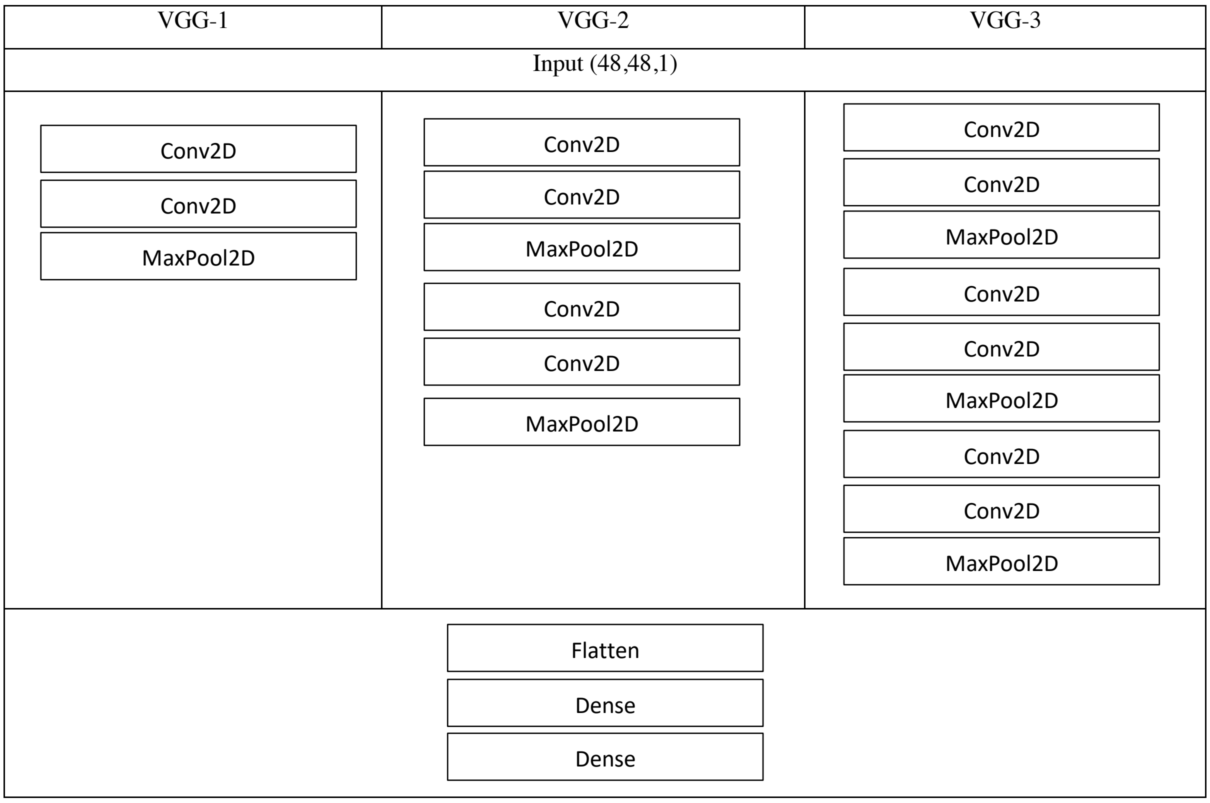

Figure 1: VGG-based model architectures

2.2 Performance Metric

This is a multiclass problem with 7 different classes, so we have considered the following performance metric:

- Categorical cross-entropy loss function: We have used deep learning model with cross-entropy layer in the end with softmax unit, so our goal is to reduce the cross-entropy loss.
- Accuracy: This tells us how well our model performs in predicting the expressions.

2.3 Image Processing

In a computer vision system, it is the process of creating a new image from an existing image, typically simplifying or enhancing the content in some way. The Keras deep learning library provides sophisticated API for loading, preparing and augmenting image data. Normalization is a good default practice for data preparation that allows minimum and maximum pixel values of 0.0 and 1.0 presented as 8-bit unsigned integer. We scale the pixel values for the model by normalizing (i.e. rescaling to a range [0,1]).

2.4 Transfer Learning

Deep convolutional neural network models may take days or even weeks to train on very large datasets, transfer learning enables a shortcut to the process by way of re-using model weights from pretrained models like VGG16 trained on very large datasets such as ImageNet image recognition tasks [1]. For the purpose of this research, we use the flexibility of transfer learning and integrate into our new models. In this task, we utilised transfer learning as a way of weight initialization scheme; were weights in the re-used layers are a starting point for the training process and are adaptable in response to the new problem. Since our task of face emotion recognition is quite similar to the ImageNet task of classification, we choose the output from layers of the fully connected layer prior to the output layer, we can even choose output from deeper in the model.

Here the output of the model from the last stack of VGG block to the output layer of the model is used as the input to a our classifier. Taking into account that convolutional layers close to the input layer learn low-level features such as lines, that layers in the middle of the layer learn more complex abstract features that combine the lower level features extracted from the input and layers close to the output interpret the extracted features in the context of a classification task.

2.5 Hardware & Software Details

Deep neural networks often require powerful computing nodes and big RAM stores and ample storage and caching capabilities, for ease of use we have opted for cloud computing services like Amazon Web Services (AWS) that has CPU instances to compute for Keras and allowed for comparatively faster but more expensive training of our neural network.

2.6 Dataset

The FER-2013 dataset consists of 48x48 pixel grayscale images of faces. The dataset is part of a larger ongoing project, and it was created by using Google image search API to search for images that relate to 184 emotion-related keywords [3]. The images where then mapped to 7 broad categories used in the Toronto Face Database [4]. The faces have been automatically registered so that the face is more or less centred and occupies about the same amount of space in each image. The CNN based algorithms presented in this work aims to categorize each face based on the emotion into seven facial expressions, which are angry, disgust, fear, happy, sad, surprise, neutral [2]. The training set consists of 28,709 examples and the public test set consists of 3,589 examples. The Disgust expression has the minimal number of images - 600, while other expressions have nearly 5000 samples each. Figure 2 shows the FER dataset has good diversity of expression type which is important from an evaluation perspective. The dataset also provided a specific training and test set.

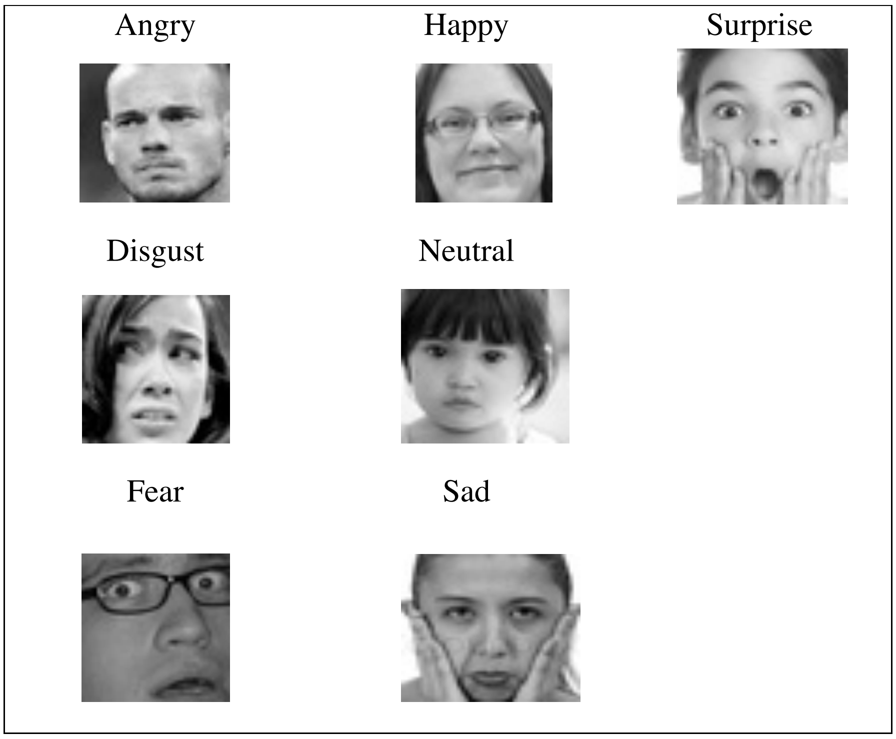

Figure 2: Diversity of expressions in the FER2013 dataset. Example faces for each expression type is shown.

3. Experimental Results

3.1 Develop a Baseline Model

In a nutshell, our model will generate 7 probability values corresponding to seven classes, where sum of all the 7 probabilities is equal to 1. The result will be feed to cross-entropy loss function which is minimized by backpropagation. In this way our model is trained to classify facial expression recognition.

As a starting point for developing our baseline model from scratch we built a The VGG-1 model, observing the plot (figure 3) over 50 epochs we see that the model overfits the data very quickly both for the loss and accuracy. However, the model's performance on the training dataset continues to improve, and in contrast, the performance of the test dataset starts to deteriorate from at ~ 15 epochs. Running the VGG-2 model (figure 4), we continue to see strong overfitting from ~ 20 epochs onwards. While, running the VGG-3 model (figure 5) we observe a modest improvement in the accuracy as we increased the number of layers in the neural network architecture.

The three models we have explored with VGG based architecture can be summarised as :

- VGG 1: 45.92%
- VGG 2: 47.20%
- VGG 3: 50.07%

All three models showed dramatic overfitting around 15-20 epochs.

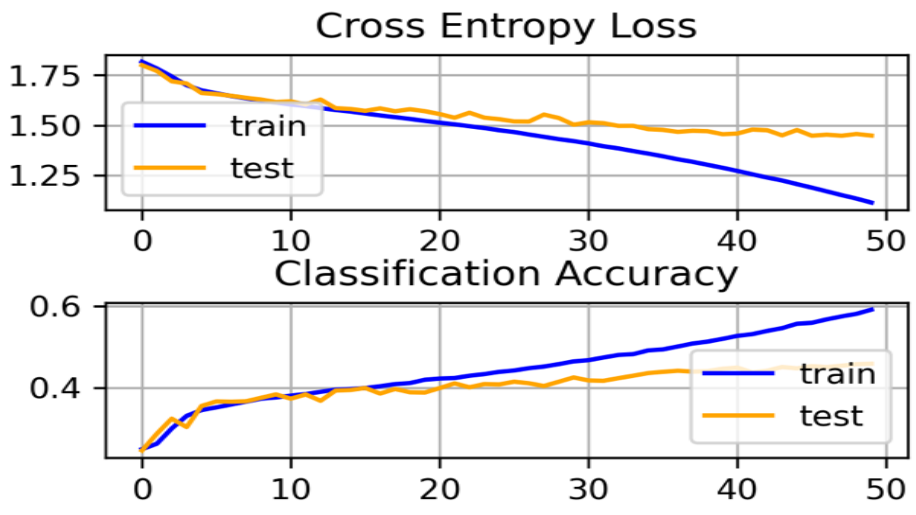

Figure 3: Learning curves for VGG 1 baseline

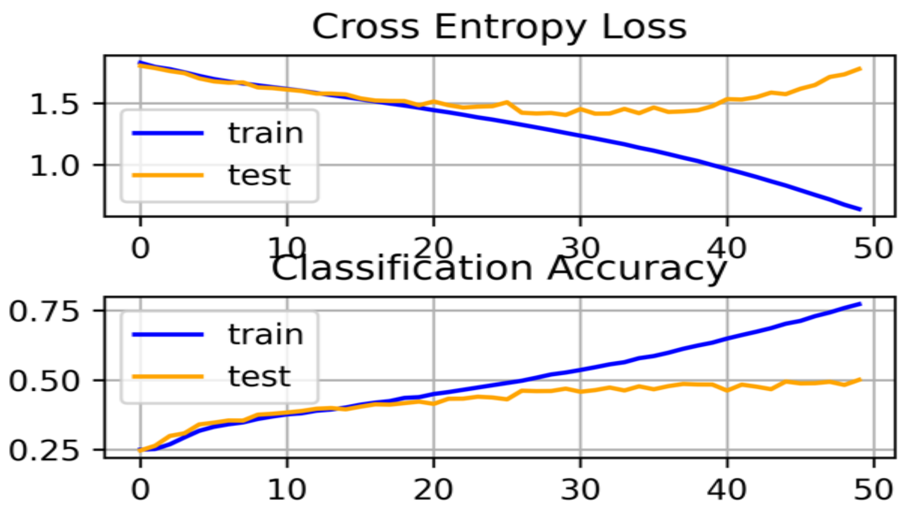

Figure 4: Learning curves for VGG 2 baseline

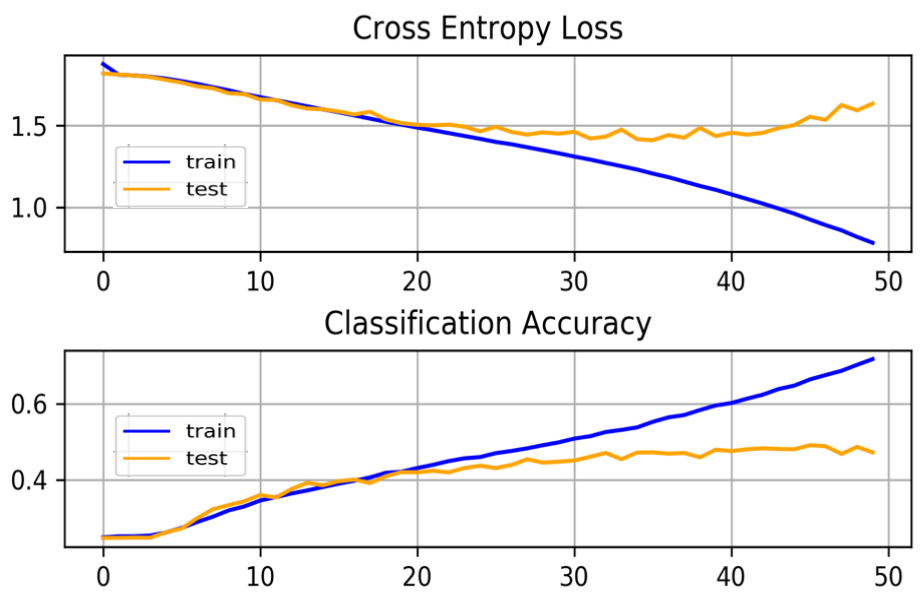

Figure 5: Learning curves for VGG-3 baseline

3.2 Develop an Improving Model

Now that we have established a baseline model i.e. VGG-3, we performed further iterations to modify the model and the training algorithm to improve its performance on our dataset. We look at two ways to do this by regularization and data augmentation.

3.3 Learning Rate

We use stochastic gradient descent optimizer and require that the learning rate is specified so that we can evaluate different rates. Our model is trained for 50 training epochs and now we investigate the dynamics of different learning rates on the accuracy and loss of the model. We can see that with a larger learning rate of 0.01, for this problem results in poor performance, overfitting from less than 5 epochs compared to 20 epochs using learning rate of 0.001. With the current configuration, the result suggest, a moderate learning rate of 0.001 results in good model convergence in the training and validation set.

Furthermore, we study the dynamics of adaptive learning rate by – Adam – on our problem. The default parameter (learning rate = 0.001) is used and the performance was observed. From the experiments a learning rate less than 0.001 simply failed to learn the data. In comparison to SGD on our dataset, using Adam resulted in inferior performance both in terms of accuracy and loss. However, Adam learns the data faster than SGD as we can observe within the first 10 epochs from figure 6 and figure 7. Also, a higher SGD learning rate of 0.01 compared to the default 0.001 helps model to learn faster even though the errors are higher (more overfitting).

In conclusion, we learn that large learning rate results in unstable training and smaller one resulted in longer training (more epochs) but more stability in our case. So we proceed with SGD with learning rate of 0.001 in the next phase of improving our baseline model.

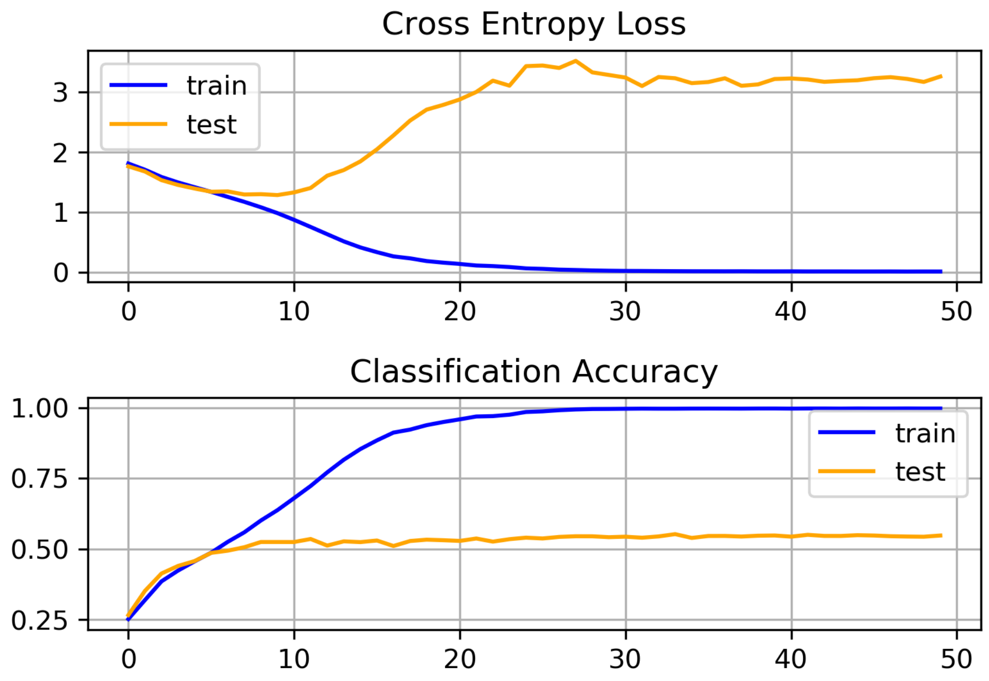

Figure 6: Learning curve using SGD (learning rate = 0.01) optimiser on VGG3 baseline

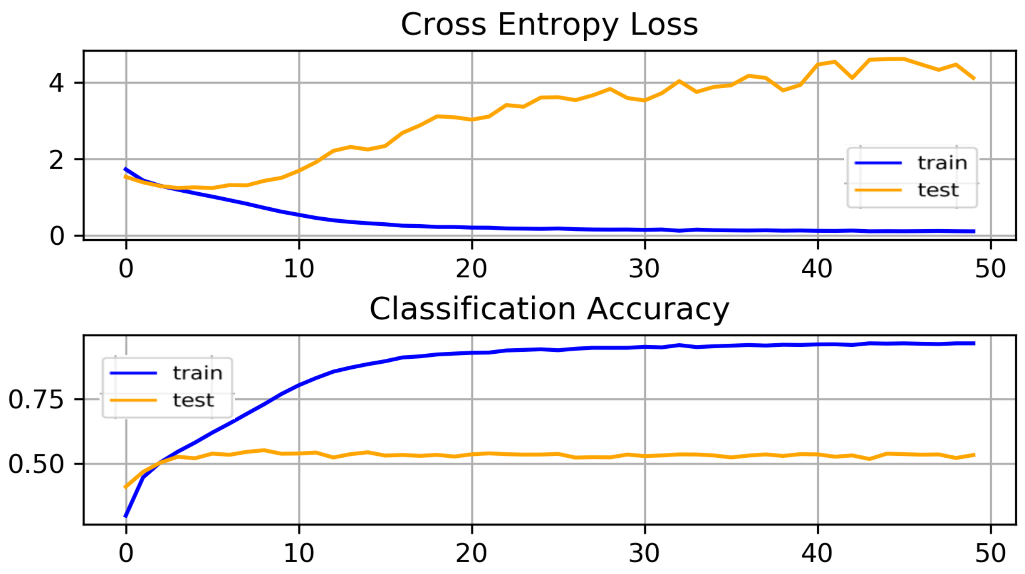

Figure 7: Learning curve using Adam (learning rate =0.001) optimiser on VGG3 baseline

3.4 Dropout, Data Augmentation and Batch Normalization

The performance of deep learning neural networks improve with the amount of data available. Image data augmentation technique is used to artificially expand the size of the training dataset by creating modified versions of the images in the training dataset that belong to the same class as the original image. Unlike data preparation that is performed across the dataset, data augmentation is typically only applied to the training dataset. By the aid of image data augmentation we studied that CNNs enabled a transform-invariant approach to learning. CNN can learn features that are invariant to their location in the image [1]. However, it is important to expand the training dataset with new plausible examples, that are likely to be seen by the model. As such, we use data augmentation to limit overfitting on the baseline model; specifically by horizontal flip, 10% shift in the height and width of the images.

To further counter the issue of overfitting we add increasing dropout layer, in which we start with a fixed dropout of 0.2. But we do not know yet that 0.2 is the best dropout rate. In this case, the combination of fixed dropout and data augmentation has resulted in ~ 52% classification accuracy. Specifically, using both regularization techniques together results in better performance than either technique used alone (table 1).

We further added Batch Normalization in an effort to stabilize the learning and accelerate the process. In order to offset the acceleration we further explored with increasing the dropout rate at 10% interval increases from 20% to 50% at different position of the dropout layers in the model architecture. The intention was to regularize more closer to the output than the input. Reviewing the learning curve (figure 9) the plot suggests learning may have improved if allowed to continue (even if very moderately).

Observing the training and testing loss, in this case, it may indicate that the validation dataset may be easier for the model to predict than the training dataset after applying data augmentation. Another explanation could be that the validation dataset does not provide sufficient information to evaluate the ability of the model to generalize, i.e. because of too few examples in the validation dataset compared to the training dataset.

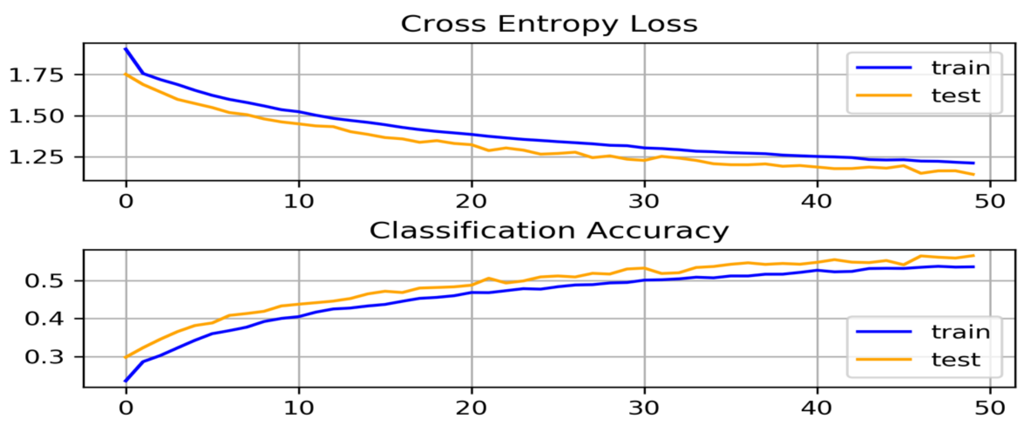

Figure 8: Baseline + Dropout (0.2) + Augmentation model

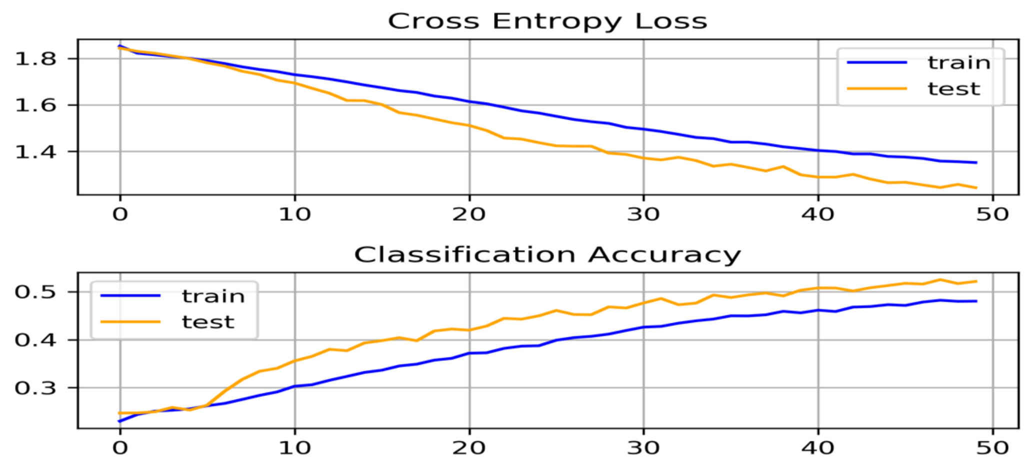

Figure 9: Baseline + Dropout (0.2,0.3,0.4,0.5) + Augmentation model + Batch Normalization model

Sl.	Model	Test Loss	Test Accuracy
1	Baseline	1.633	50.07%
2	Baseline + Augmentation + Dropout (fixed)	1.243	52.15%
3	Baseline + Dropout (increasing) + Augmentation + BatchNormalization	1.443	56.49%

Table 1: Summary of model tuning based on VGG3 architecture.

Our specific results may vary given the stochastic nature of the learning algorithm. In future improvements we can run the problem several times and compare the average performance. Next we will compare the VGG-3 model with dropout, augmentation and batch normalization to a pre-trained network of VGG-16 architecture.

3.5 Pretrained VGG-16

Finally, we compared our modified VGG-3 architecture with another modern convolutional neural network, namely a pre-trained VGG-16 architecture. This model was found to be not only deep and but also complex. So they were hard to train, and it was understood that very large number of images are necessary to train these networks without overfitting . The deeper neural network helped to improve accuracy of the model and help the model to learn faster that is within the first 5 epochs the model achieved >50% accuracy (figure 10).

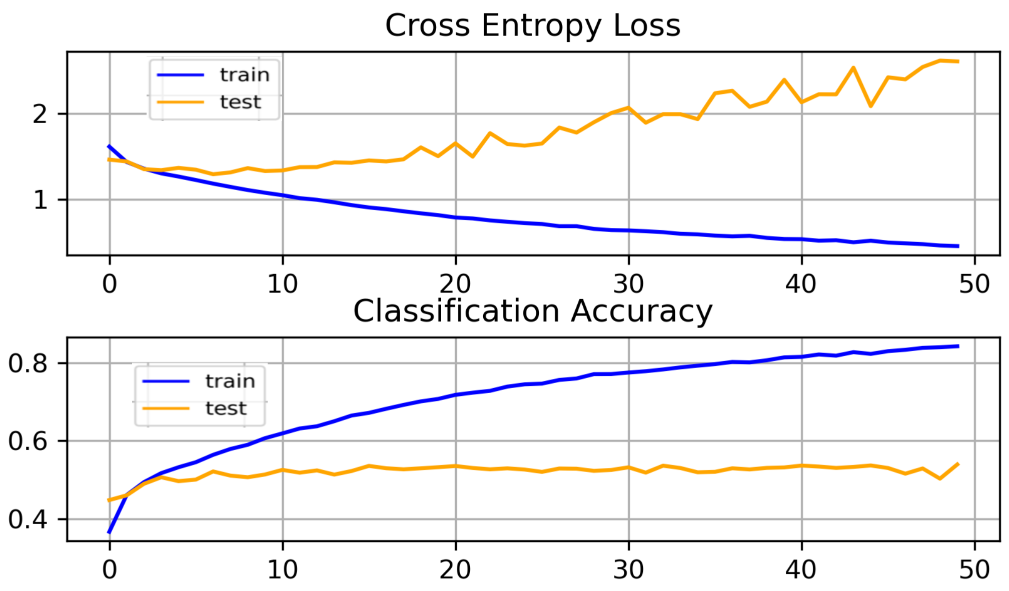

Figure 10: Learning curve on pretrained VGG-16 model

For our task we have chosen VGG-16 pre-trained neural network to generate features. VGG-16 contains 16 layers out of which 13 are convolutional. It is well trained on the ILSVRC 2014 [25] dataset with a million images. In the classifier portion of our network after feature extraction we add a Flatten layer, followed by two fully connected layers with 256 and 128 neurons. For restricting overfitting we added two layers of dropout regularization of 20% between the fully connected layers.

In comparison to the VGG-3 that we trained from scratch, the pretrained VGG-16 improved the accuracy value. The VGG-16, learned the data faster and reached ~50% accuracy within the first few epochs (figure 10) in contrast to our modified baseline model (figure 9). We can still make a number of further improvements to the VGG-16, but that was limited by computational capacity in the present work.

Algorithm	Accuracy	Computational Complexity
VGG-16	2.33	58.03%
VGG-3 (baseline)	1.443	56.49%

Table 2: Comparison table of modified pre-trained VGG16 with modified baseline (modified)

There was significant complexity experienced with running the deep neural network of VGG-16 but the performance observed to be within reason. Various reasons may contribute to this, primarily the size of the dataset. For reasons of limited computational capacity the neurons in the fully connected layers were kept at a lower range and the number of epochs were also relatively less in the experiments. We also observe that instead of freezing weights from all layers of the VGG16 block, taking two layers of Convolution and a layer of Max pooling from the last block and training those weights helps at adopting better abstractions of the model, even though it makes run times significantly longer.

The VGG models are a good starting point because they achieved reasonable performance on the FER 2013 dataset but most notably, the structure of the VGG based network was easy to understand and implement.

3.6 Further Scope

We have pretty decent result given the limited computational resources. However, there is still huge scope of improvement.

1. It is clear that images are indeed very small, thus, it can be challenging to see what is represented in some of the images given the low resolution. This low resolution is likely the cause of the limited performance that top-of-the-line algorithms are able to achieve. We can use images over 400 * 400 pixels.
2. Design our own neural network given we have the time and the computational capacity. For instance, CNN for feature extraction followed by a classifier such as SVM, use Autoencoder for feature extraction followed by a classifier.
3. To get better accuracy we can use a bigger dataset with more variance among them.
4. Develop model in PyTorch and exploit Graphics Processing Unit (GPU) computation to expedite the process

4. Conclusion

After the extraction through CNN, the CNN-based features were trained in the fully connected layer to classify emotions. There are several variations of neural network architecture that can be experimented with for this FER2013 dataset. Interestingly, there is no right or wrong ways of going about it, however, using pretrained models provides us a tool for estimating our modified network's performance. We also encounter that, a deep neural network does not necessarily mean a better network, the performance of these networks vary from a case by case basis. However, developing our VGG from scratch has provided an interesting window to understand that it is able to learn features of the FER2013 dataset, even though a great deal of improvement can be made to enhance its performance.

References

[1] J. Brownlee., Deep Learning for Computer Vision, Image Classification, Object Detection and Face Recognition in Python, 2020

[2] Sambare, M. (2020, July 19). FER-2013. Retrieved April 15, 2021, from https://www.kaggle.com/msambare/fer2013

[3] Goodfellow, I. J., Erhan, D., Luc Carrier, P., Courville, A., Mirza, M., Hamner, B., . . . Bengio, Y. (2015). Challenges in representation Learning: A report on three machine LEARNING CONTESTS. Neural Networks, 64, 59-63. doi:10.1016/j.neunet.2014.09.005

[4] Joshua Susskind, Adam Anderson, and Geoffrey E. Hinton. The Toronto face dataset. Technical Report UTML TR 2010-001, U. Toronto, 2010.

[5] G. Littlewort, M. Bartlett, I. Fasel, J. Susskind, and J. Movellan. Dynamics of facial expression extracted automatically from video. Image and Vision Computing, 24(6), 2006.

[6] P. Ekman,W. Friesen, Facial Action Coding System: A Technique for the Measurement of Facial Movement, Consulting Psychologists Press, 1978

[7] M.S. Bartlett, G. Littlewort, M.G. Frank, C. Lainscsek, I. Fasel, and J.R. Movellan. Automatic recognition of facial actions in spontaneous expressions. Journal of Multimedia, 2006.

[8] S. Alizadeh, A. Fazel, Convolutional Neural Networks for Facial Expression Recognition.

[9] Cohen, Ira, et al. "Evaluation of expression recognition techniques." Image and Video Retrieval. Springer Berlin Heidelberg, 2003. 184-195.

[10] Padgett, C., Cottrell, G.: Representing face images for emotion classification. In: Conf. Advances in Neural Information

[11] Philipp Michel and Rana El Kaliouby. Real time facial expression recognition in video using support vector machines. In Proceedings of the 5th international conference on Multimodal interfaces, pages 258–264. ACM, 2003.

[12] Pranav Kumar, SL Happy, and Aurobinda Routray. A realtime robust facial expression recognition system using hog features. In 2016 International Conference on Computing, Analytics and Security Trends (CAST), pages 289–293. IEEE, 2016.

[13] Lee, J. R., Wang, L., &amp; Wong, A. (2021). EmotionNet nano: An efficient deep convolutional neural network design for real-time facial expression recognition. Frontiers in Artificial Intelligence, 3. doi:10.3389/frai.2020.609673

[14] Mehendale, N. (2020). Facial emotion recognition using convolutional neural NETWORKS (FERC). SN Applied Sciences, 2(3). doi:10.1007/s42452-020-2234-1

[15] Kim DH, An KH, Ryu YG, Chung MJ (2007) A facial expression imitation system for the primitive of intuitive humanrobot interaction. In: Sarkar N (ed) Human robot interaction. IntechOpen, London

[16] Ernst H (1934) Evolution of facial musculature and facial expression. J Nerv Ment Dis 79(1):109

[17] Kumar KC (2012) Morphology based facial feature extraction and facial expression recognition for driver vigilance. Int J Comput Appl 51:2

[18] Hernández-Travieso JG, Travieso CM, Pozo-Baños D, Alonso JB et al (2013) Expression detector system based on facial images. In: BIOSIGNALS 2013-proceedings of the international conference on bio-inspired systems and signal processing

[19] Matsumoto D (1992) More evidence for the universality of a contempt expression. Motiv Emot 16(4):363

[20] Ratyal NI, Taj IA, Sajid M, Ali N, Mahmood A, Razzaq S (2019) Three-dimensional face recognition using variance-based registration and subject-specific descriptors. Int J Adv Robot Syst 16(3):1729881419851716

[21] Ratyal N, Taj IA, Sajid M, Mahmood A, Razzaq S, Dar SH, Ali N, Usman M, Baig MJA, Mussadiq U (2019) Deeply learned pose invariant image analysis with applications in 3D face recognition. Math Probl Eng 2019:1–21

[22] Sajid M, Ali N, Dar SH, Iqbal Ratyal N, Butt AR, Zafar B, Shafique T, Baig MJA, Riaz I, Baig S (2018) Data augmentation-assisted makeup-invariant face recognition. Math Probl Eng 2018:1–10

[23] Ratyal N, Taj I, Bajwa U, Sajid M (2018) Pose and expression invariant alignment based multi-view 3D face recognition. KSII Trans Internet Inf Syst 12:10

[24] Ernst H (1934) Evolution of facial musculature and facial expression. J Nerv Ment Dis 79(1):109

[25] ImageNet large Scale visual Recognition Challenge 2014 (ilsvrc2014). (n.d.). Retrieved April 15, 2021, from http://www.image-net.org/challenges/LSVRC/2014/