Emotion Detection From Facial Expression

1st Anuj Verma
CSE Department
Indraprasth Institute Of Information Technology
Delhi, India
anuj17026@iiitd.ac.in

2nd Sohaib Fazal
CSAM Department
Indraprasth Institute Of Information Technology
Delhi, India
sohaib17267@iiitd.ac.in

***Abstract*—Emotion recognition plays an important role in interpersonal relationships. The automatic recognition of human emotion is an active area of research in the early eras. There are many things that reflects a human emotion like speech, hand gestures, body movements but we are considering only facial expressions to find the human emotion. Extraction of human emotion can be quite useful in case of human and machine interactions. There are many approaches discussed for classifying the human emotion. The main objective of this paper is to detect human recognition in real time.**

***Index Terms*—emotion recognition, facial expressions, human and machine interaction, real time**

I. Problem Statement And Motivation

Our problem statement is to extract the emotion of a human from its facial expressions. The emotions that we want to extract have seven classes [Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral]. Today facial behaviour is important in many applications such as gaming, criminal identification, advertising, safety of persons in cars etc. Many companies are now interested in this application and we have tried many approaches to detect it with higher accuracy. Even this can be very useful for human and machine interaction.

II. Literature Review

A. Best Model For Image Classification

With several models available the best model that can be used for image recognition involves CNN as shown in [2]. The paper demonstrates that shallow networks(Multilayer perceptron) and SVM reported only 0.39 test accuracy whereas CNN reported 0.557 on a particular dataset. Moreover [5] shows that state of the art results can be obtained on smaller CNNs and has scope for faster training time also.

B. Current Pretrained Models For Image Classification

There has been various upcoming models for image classification like AlexNet, ResNet, ShuffleNet,SqeezeNet, GoogleNet, VGG-19, DenseNet and Inception and an approach for image classification is shown in [1] where they train using the Image Net, AlexNet, VGG and combine it with CNN to train on smaller datasets to get accuracy in the ranges of 0.53 to 0.55.

III. DataSet Details

The Dataset is obtained from the ICML 2013 competition named Challenges in Representation Learning: Facial Expression Recognition, hosted on Kaggle. The dataset consists of 28,709 training images. There were 3,589 public test files and 3,589 private test files used at the end of competition. Each image is of dimension 48*48 and grayscale. There are 7 types of labels associated with every image. The categories are as follows (0 =Angry, 1= Disgust, 2= Fear, 3 = Happy, 4 = Sad, 5= Surprise, 6 =Neutral). I have used the public test files as validation images and private test files as the test files.

TABLE I
Distribution Of Training Datasets

Labels	***Angry***	***Disgust***	***Fear***	***Happy***	***Sad***	***Surprise***	***Neutral***
Sample	***3995***	***436***	***4097***	***7215***	***4830***	***3171***	***4965***

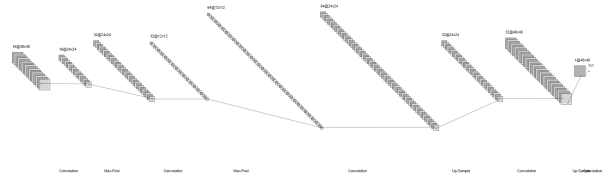
After obtaining the dataset we have preprocessed the dataset by making it balanced(using SMOTE) and we have also normalized every image. After balancing the dataset we have got 7215 samples of each kind to get total of 50505 samples. Also we have checked if there are any outliers in the dataset by checking the z score method. There were no outliers in the dataset.

IV. Proposed Architecture

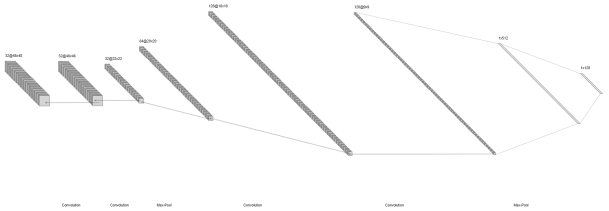
We have designed a CNN model with first four layers being the convolutional layers with filters 32,32,64 and 128 respectively. The kernel sizes used is (3,3) and stride is 1. After that we have used three dense layers of sizes 512,128 and 7 respectively the last layer being for the output. We have used relu activation everywhere except the last layer for output where we have used softmax. We have applied batch Normalization between each layer in order to reduce overfitting and faster training. Also between each layer except the first two layers we applied a dropout of .25 except the last two pair of dense layers where dropout is 0.5 the reason for increasing the drop out in later case to avoid overfitting as the size of the dense layer is quite large. The model is trained on 12 epochs only. The loss function used for fitting the model is categorical–cross entropy as our data was categorical. Optimizer used is Adam(faster and shows consistent results),

metrics used is just for the accuracy.

Autoencoder Model



Convolution Model

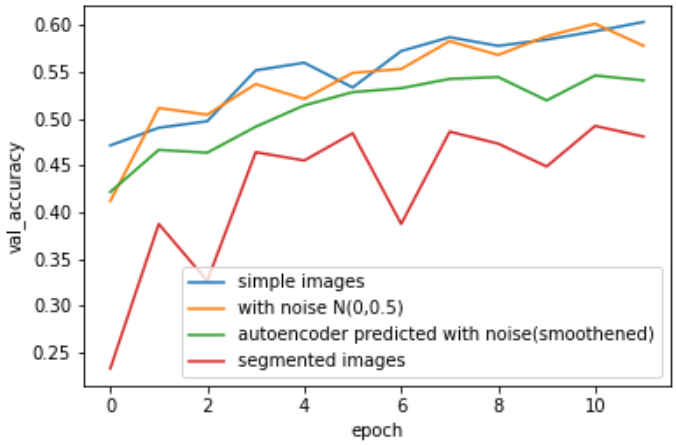


A. Early Models

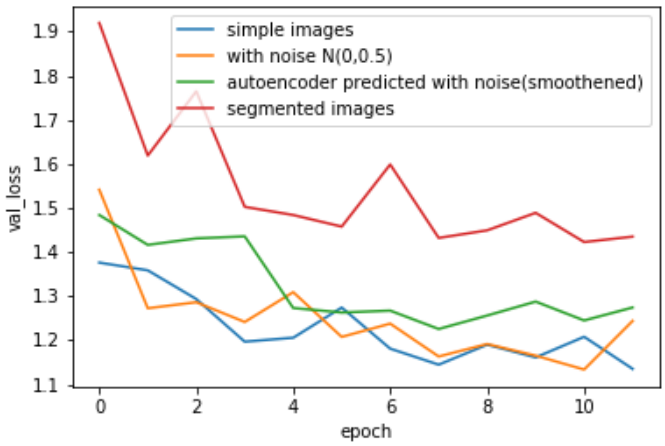
- Earlier we used a 3 convolutional layer model with sizes 32,64,128 respectively and with 1-2 dense layers" with batch normalization in between and dropout of 0.25 which could only produce a testing accuracy of 0.52 - 0.54 so we decided to add another convolutional layer for better fitting.
- Without adding dropout layers and Batch Normalization our model was overfitting and produced testing accuracy of only 0.24 whereas the model was fitted to 0.90. So we added a dropout of 0.25 and used Batch Normalization.
- Kernel regularizer didn't bring much impact to our model and also after applying batch Normalization and dropout our model was not overfitting much so we didn't use it.
- Smaller batch size of 32 brought better accuracy in comparison to earlier batch sizes of 128,256 on which we were training the dataset as it may lead to better generalization.
- We used only 12 epochs for determining the model because of computation time and also the model after 8th epoch was not showing a significant improvement.
- We also used early stopping which was monitored on validation accuracy with a patience of 3 just in case the model does not overfit.

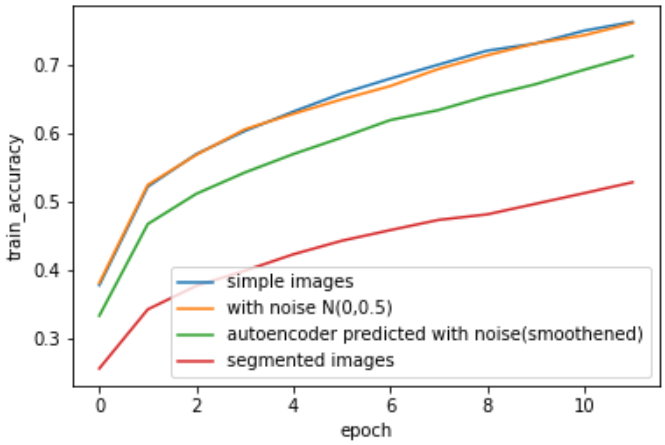
V. Visualization

- Training Images With CNN–**Model 1**
- Training images with added Gaussian Noise(0,0.5) With CNN–**Model 2**
- Training images with added Gaussian Noise(0,0.5) predicted via autoencoder.(The autoencoder was a convolutional AE with 3 layers of filters 16,32,64 and kernel size =(3,3)) With CNN–**Model 3**
- Segmented Images With CNN–**Model 4**



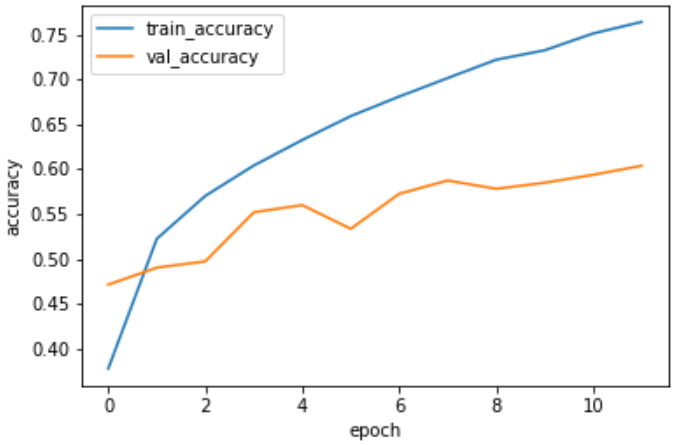
We can see that simple images and noisy images performed similarly on this parameter where as autoencoder predicted images performed relatively poor followed by segmented images.



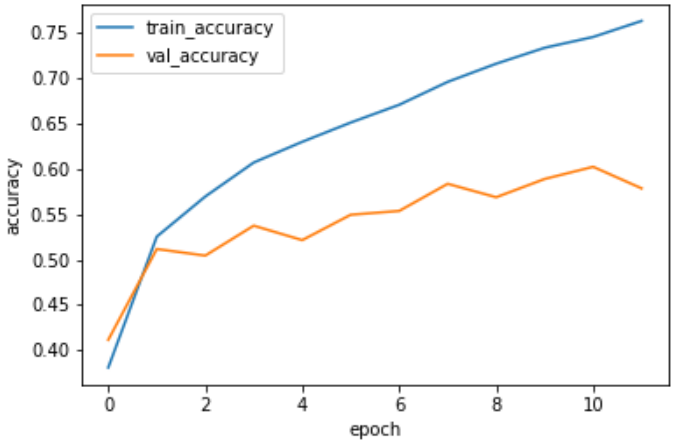


Other Models That We Have Used

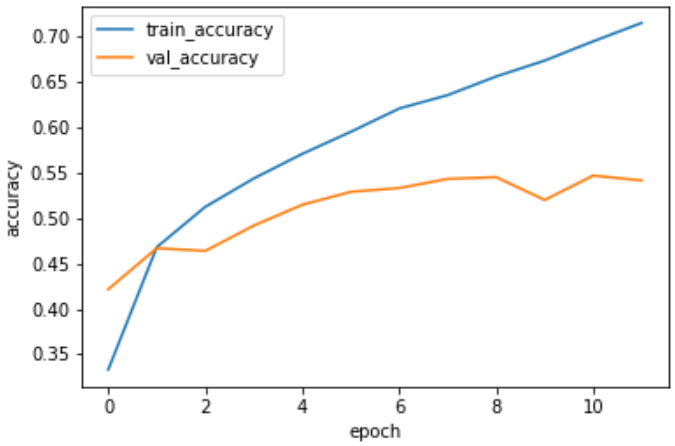
Model 1



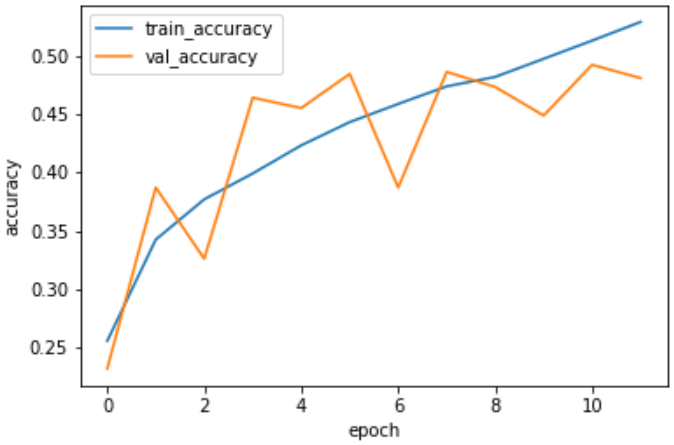
Model 2



Model 3



Model 4



Fisrt three approaches perform similar.
But the segmented images that is when we are removing the background pixels from the images did not overfits and on some epochs it underfits as well showing that CNN need more epoch to learn the general behaviour for this model.

VI. Results

TABLE II
ACCRACY FOR DIFFERENT MODELS

Type	*Model 1*	*Model 2*	*Model 3*	*Model 4*
Train	*0.8793*	*0.7797*	*0.7576*	*0.6809*
Test	*0.616*	*0.5814*	*0.5486*	*0.4758*

TABLE III
CONFUSION MATRIX FOR THE MODEL 1

	Anger	*Disgust*	*Fear*	*Happy*	*Sad*	*Surprise*	*Neutral*
Anger	*241*	*6*	*47*	*33*	*93*	*9*	*62*
Disgust	*9*	*32*	*4*	*5*	*4*	*0*	*1*
Fear	*55*	*3*	*208*	*20*	*113*	*49*	*80*
Happy	*31*	*0*	*29*	*709*	*55*	*19*	*36*
Sad	*57*	*0*	*53*	*42*	*319*	*11*	*112*
Surprise	*14*	*2*	*24*	*21*	*15*	*318*	*22*
Neutral	*38*	*2*	*27*	*49*	*116*	*8*	*386*

Every Expression Ratio

correct angry ratio: 0.4908350305498982
correct disgust ratio: 0.5818181818181818
correct fear ratio: 0.3939393939393939
correct happy ratio: 0.8065984072810012
correct sad ratio: 0.5370370370370371
correct surprise ratio: 0.7644230769230769
correct neutral ratio: 0.6166134185303515

TABLE IV
CONFUSION MATRIX FOR THE MODEL 2

	Anger	*Disgust*	*Fear*	*Happy*	*Sad*	*Surprise*	*Neutral*
Anger	*183*	*9*	*26*	*44*	*149*	*8*	*72*
Disgust	*7*	*37*	*1*	*1*	*6*	*1*	*2*
Fear	*52*	*7*	*136*	*37*	*191*	*45*	*60*
Happy	*16*	*3*	*11*	*701*	*96*	*13*	*39*
Sad	*30*	*8*	*21*	*36*	*404*	*8*	*87*
Surprise	*4*	*2*	*37*	*35*	*39*	*273*	*26*
Neutral	*19*	*5*	*11*	*56*	*175*	*7*	*353*

Every Expression Ratio

correct angry ratio: 0.3727087576374745
correct disgust ratio: 0.6727272727272727
correct fear ratio: 0.25757575757575757
correct happy ratio: 0.7974971558589306
correct sad ratio: 0.6801346801346801
correct surprise ratio: 0.65625
correct neutral ratio: 0.5638977635782748

TABLE V
CONFUSION MATRIX FOR THE MODEL 3

	Anger	*Disgust*	*Fear*	*Happy*	*Sad*	*Surprise*	*Neutral*
Anger	*222*	*4*	*40*	*47*	*69*	*19*	*90*
Disgust	*17*	*28*	*1*	*1*	*4*	*1*	*3*
Fear	*68*	*1*	*124*	*58*	*114*	*59*	*104*
Happy	*43*	*0*	*25*	*684*	*47*	*17*	*63*
Sad	*71*	*2*	*47*	*66*	*226*	*9*	*173*
Surprise	*18*	*1*	*26*	*35*	*17*	*279*	*40*
Neutral	*35*	*0*	*17*	*78*	*80*	*10*	*406*

Every Expression Ratio

correct angry ratio: 0.45213849287169044
correct disgust ratio: 0.509090909090909
correct fear ratio: 0.23484848484848486
correct happy ratio: 0.7781569965870307
correct sad ratio: 0.38047138047138046
correct surprise ratio: 0.6706730769230769
correct neutral ratio: 0.6485623003194888

TABLE VI
CONFUSION MATRIX FOR THE MODEL 4

	Anger	*Disgust*	*Fear*	*Happy*	*Sad*	*Surprise*	*Neutral*
Anger	*186*	*10*	*52*	*98*	*32*	*30*	*83*
Disgust	*11*	*29*	*3*	*6*	*1*	*2*	*3*
Fear	*76*	*4*	*120*	*99*	*50*	*81*	*98*
Happy	*47*	*4*	*38*	*618*	*31*	*39*	*102*
Sad	*95*	*8*	*47*	*131*	*133*	*36*	*144*
Surprise	*12*	*2*	*31*	*44*	*10*	*297*	*20*
Neutral	*41*	*4*	*40*	*144*	*40*	*32*	*325*

Every Expression Ratio

correct angry ratio: 0.3788187372708758
correct disgust ratio: 0.5272727272727272
correct fear ratio: 0.22727272727272727
correct happy ratio: 0.7030716723549488
correct sad ratio: 0.2239057239057239
correct surprise ratio: 0.7139423076923077
correct neutral ratio: 0.5191693290734825

A. PSNR

We picked a random sample(image 100) and calculated the psnr of image 100 under different models and found out the following results :

- Training image (PSNR = 100 As it is the true image)
- Training image with noise (PSNR = 54.31)
- Autoencoder Predicted Images after adding noise(PSNR = 17.414)
- Segmented Images(PSNR =35.375569)

TABLE VII
PSNR OF DIFFERENT MODELS

Type	*Model 1*	*Model 2*	*Model 3*	*Model 4*
Image 100	*100*	*54.31*	*17.41*	*35.37*

VII. ANALYSIS

- Model 1 and Model 2 performed better than Model 3 and Model 4. Although Model 1 produced better accuracy than Model 2 but still Model 2 provides significant less gap between training and testing accuracy(less overfitting) so it can considered better.From our results we found that Model 4 is worst.
- By analysing confusion matrix, we can say that the best predicted emotions in all the cases are happy,surprise,neutral, disgust where as sad,angry and fear are not well predicted.
- The model with better psnr for image 100 performs better in comparison to the one with lower psnr.

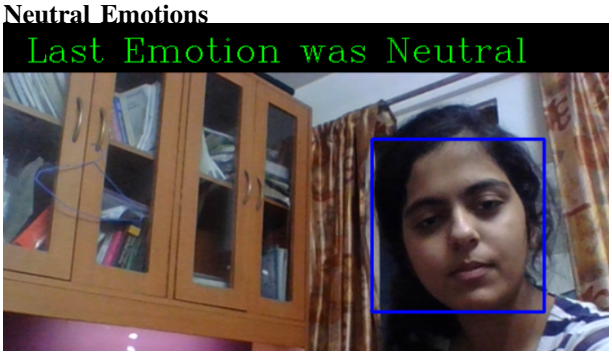
VIII. EXPERIMENTS

Here we are attaching some photos that we have taken during testing our model. We have used the camera of our laptop to do this task. This thing also gives clarity about how the model works in real time.

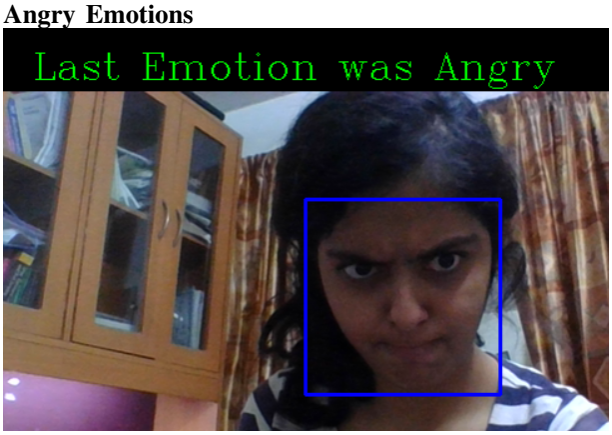
Happy Emotions

Fear Emotions

Neutral Emotions



Angry Emotions



Surprise Emotions

IX. Discussion And Conclusion

By seeing all the comparative graphs we can say that Model 2>= Model 1> Model 3> Model 4.

This shows that CNN tend to perform better when noisy images are provided as it increases the generalization power.

And also lets CNN to denoise the images itself and reduce noise. The autoencoder predicted images performed poorly as they were blurry and smooth therefore with less noise,it was difficult for CNN to train on it. Segmented images performed worst as they were lost some features and provided less chance for CNN to learn from the data with less features.

For the segmented images we have seen that the CNN does not get saturated with the 12 epochs but it is highly probable that it can give better results. As we are ignoring some image features in the segmented images the gap between train and test accuracy is very low or we can say lowest of all the models so we expect CNN can work best in this case also if we use right value of parameters for training the model.

The expressions sad, angry, fear were less distinguishable and provided confusion leading to conclusion that they need to be further looked upon and more of these expressions need to be augmented which are of higher quality as one model could not produce good results on them.

The higher the PSNR the better is the prediction of the sample as shown by image 100 (random sample) that image with higher psnr tend to produce better results.

X. Individual Contribution

We both have equal contribution in the coding part.

- Anuj Verma– I have provided the relevant resources and also gave relevant solutions to the problems.
- Sohaib Fazal– Final Model is made by him and as well he has trained the final model.

References

[1] Deep Learning for Emotion Recognition on Small Datasets Using Transfer Learning Hong-Wei Ng, Viet Dung Nguyen, Vassilios Vonikakis, Stefan Winkler Advanced Digital Sciences Center (ADSC) University of Illinois at Urbana-Champaign, Singapore

[2] Learning facial expressions from an image (Bhrugurajsinh Chudasama, Chinmay Duvedi, Jithin Parayil Thomas.

[3] A Method for Improving CNN-Based Image Recognition Using DCGAN Wei Fang1, 2 , Feihong Zhang1, * , Victor S. Sheng3 and Yewen Ding CMC, vol.57, no.1, pp.167-178, 2018.

[4] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[5] Enhanced Image Classification With a Fast-Learning Shallow Convolutional Neural Network Mark D. McDonnell and Tony Vladusich.

[6] How to control the stability of training neural networks with the batch size by Jason Brownlee.

[7] Image denoising using deep CNN with batch renormalization.

[8] https://medium.com/datadriveninvestor/real-time-facial-expression-recognition-f860dacfeb6a

[9] https://towardsdatascience.com/the-4-convolutional-neural-network-models-that-can-classify-your-fashion-images-9fe7f3e5399d