

AoML2

by Vikram Kalta

Submission date: 03-May-2023 10:25PM (UTC+0100)

Submission ID: 204591814

File name: 42274_Vikram_Kalta_AoML2_498640_1188173393.pdf (720.56K)

Word count: 3082

Character count: 15539

Training convolutional neural networks with FER-2013

Vikram Kalta (2209281)

MSc Data Science and Artificial Intelligence
University of Wales Trinity Saint David
Swansea, UK

Abstract—The world is not new to the power of artificial intelligence (AI) anymore. With the onset of chatbots such as chatGPT, it is evident that it has taken the world by surprise. Machine learning being a part of AI brings in wide variety of models which is used for building systems which learns from the data and does the prediction job. This report takes on a challenge in the field of facial expression recognition by experimenting on dataset such as FER2013 and train a deep neural network on it. A number of experiments are carried out to achieve a better accuracy rate. This is done by following the previous works and learning about the most effective CNN architectures such as VGG. The best accuracy rate for these experiments achieved was 71.5%. Due to the computational constraints the experiments were very limited. This gives space for another area to study which is to make computers used for training utilize the power of parallel processing and hence reducing the training time of the models doing so.

Index Terms—CNN, Neural networks, VGG-16, FER2013

I. INTRODUCTION

Facial expressions are a form of non-verbal communication [1]. In the study of non-verbal communication eye contact is considered to be the most important. The most common facial expressions are happy, sad, surprised, angry, fear etc. Eyes are a key feature that aids in conveying emotions. The eyes takes various shapes and movements such as widening, blinking etc when conveying different emotions. For example the eyes tends to get curvy when a human is happy. When someone is angry the eyes may become narrowed. Similarly other features of the face might be correlated with the different expressions such as opening of mouth when a human conveys surprise emotion. Facial expression recognition (FER) is an interesting area in the field of computer vision and is focussed on developing a system which recognizes human emotions based on facial expressions. FER has been used in human-computer interaction such as gaming, virtual reality etc. FER is also used for marketing and advertisement to capture the emotions of the consumer to various stimuli such as ads, packaging etc. This report will explore FER2013 dataset which is one of the most common used datasets used in the research for FER. FER2013 has 35887 grayscale images in total. This dataset is further split into training and test dataset containing 28709 and 7178 each respectively. Each image is size 48*48 pixels. The images are grouped by 7 different emotions. The emotions are as follows:

- Anger
- Disgust
- Fear
- Happy
- Sad
- Surprise
- Neutral

With the help of machine learning methods a system can be developed which finds the patterns in the pixel positions in the images and attempts to classify the emotions. The objective of this report is to develop such a system which recognizes the image using state of the art techniques and methods. In recent years convolutional neural networks (CNN) has achieved state of the art performance on FER2013. The CNNs are traced back to 1970s, when the first neocognitron model was proposed by Kuniyiko Fukushima [2]. Neocognitron is a type of neural network architecture inspired by the structure and function of the visual cortex. Later Yann Le Cunn played a major role to bring Convolutional Neural Network to a level it has reached today. The core building blocks of CNN are convolutional layers, pooling layers and fully connected layers. The convolutional layers apply filters to the input image, resulting in a feature map. The pooling layer further reduces the spatial size of the feature map, which aids in reducing the computational complexity of the model. Finally the fully connected layers process the flattened feature map and output the final classification results. Each of the layer is discussed further below:

- Convolutional layer: Convolutional layers works with filters which might be in the shape of 2*2 or 3*3 matrix. This filter is then applied to the original image starting from top left. Each part which is equivalent in size to the filter is then multiplied with the filter and then all the values are added and stored in the feature map. This also reduces the size of the original image preserving some important feature of the image as shown in figure 1. The stride is the shift of receptive field to the next [3]. The stride value plays an important role to build the feature map in the next layer. For example if a 5*7 input layer is connected to 3*4 layer using 3*3 receptive fields with a stride of 2.
- Pooling layer: Pooling layer further reduces the size

of the feature map. This also helps in reducing the computational load. One of the common type of pooling layer is max pooling layer. Depending on the size of the max pooling layer selected it will get the maximum value from the corresponding matrix in the feature map from the previous layer and store it in the next layer.

- Fully connected layer: This being the last layer, the data is flattened before being passed on to produce a result.

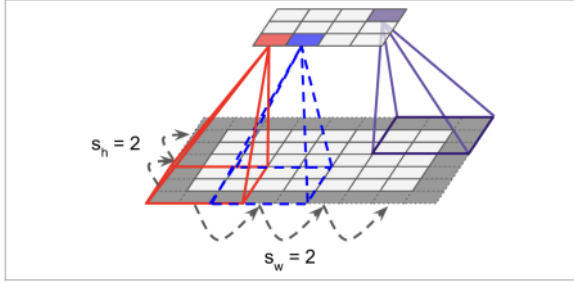


Fig. 1. Working of a convolutional layer

Although the state of the art performance achieved for this dataset is 75.2%. It is worth mentioning that the human accuracy achieved is 65.5% as per [4].

In the following section the author will review the previous work in the literature that has worked with FER2013. Section after that will discuss about the experiments done for this report and then followed by results and discussions. In the conclusions the author will talk about future works that could be done to improve the efficiency of CNN model.

II. LITERATURE REVIEW

This literature review targets to provide a comprehensive overview of the existing research on facial expression recognition datasets such as FER2013. This section will talk about various architectures of CNN adopted by the authors in this area to achieve the best results. Authors in [5] has utilized FER2013 dataset to train the CNN models and has achieved state of the art performance by achieving 75.2% accuracy. Author has carried out initial experiments with 3 different CNN architectures which are summarized in the figure 2. The architecture used are VGG, Inception, ResNet.

TESTED DEEP ARCHITECTURES AND THEIR TEN-CROP TEST ACCURACY ON FER2013. 3R MEANS GROUP OF THREE RESIDUAL BLOCKS.

Name	Architecture	Depth	Parameters	Accuracy
VGG	CCPCCPCPCPF	10	1.8 m	72.7%
Inception	CIPPIPIPIPF	16	1.2 m	71.6%
ResNet	3R4R63RPF	33	5.3 m	72.4%

Fig. 2. CNN architectures in [5]

VGG architecture used by the author is similar to VGG-B [6] but has one less block of CCP. Kindly note that C is used as an acronym for convolutional layer and likewise P

is for pooling layer. The authors have also added a dropout layer after every block and managed to improve the accuracy by 1%. The next architecture i.e Inception is similar to GoogLeNet [7]. The net uses 3*3 features maps. The n is initialized to 32 and increased by 32 after every Inception layer. ResNet is used without the initial CP block. After final pooling layer a dropout layer is added. The authors have used ensemble of 8 such CNN models to achieve accuracy of 75.2%. Its important to note that the authors have not utilized any additional training data or features, data augmentation techniques.

Authors in [8] has experimented with implementing face detection techniques to capture most of the face detection to capture the most of the face. These experiments again are carried out on FER2013. The author has intended to handle issues such as occlusion, illumination and head pose. The authors have applied dlib facial landmark detector pre-trained on iBUG 300-W dataset for feature extraction and extracted 8 most prominent parts of a face, both eyes, including both eyebrows, the jaw, the inner and outer outlines of the mouth and the nose. The CNN architecture used by the authors here are as follows:

- 6 Convolutional layers using ReLU.
- 3 Max pooling layers. Out of which the first 2 using pool size (3,3) and stride (2,2) and third uses pool size (2,2) and stride (2,2), every max pooling is followed by 2 convolutional layers.
- 2 drop out layers with value 0.2.
- 1 flattened layer and 2 dense layers: One dense layer with ReLU and the other with Softmax as activation function. Authors have achieved 61.7% accuracy without any preprocessing techniques applied. Best accuracy was found with batch size of 512 and 10 epochs.

In [9] authors have carried out experiments on FER2013 dataset and have taken a bottom-up approach to design a network. The authors proposed 2 architectures and refers to them as Model1 and Model2. Model2 is a reduced variant of Model1. As part of preprocessing the images are rescaled to 64*64. The optimizer chosen here is ADAM. Network with 16 convolutional layers has helped achieve good results. The achieved accuracy is 65% similar to the human accuracy achieved. Also stochastic gradient descent has provided a slightly better improvement when compared to ADAM which is the default optimizer.

III. EXPERIMENTS

This report will work with several CNN models which will be trained with FER2013. For this report the author has utilized a popular open-source neural network library called keras widely used for deep learning applications. It is written in python and is built on top of tensorflow. The original dataset has 28709 images in 7 classes in training set and 7178 images in 7 classes in test dataset. But for this report the author has chosen 4 classes from the dataset which are happy, sad, angry

and surprise. Further the test set is divided into 2 halves for test and validation set each. A normalization technique is applied to training, test and validation datasets by dividing all pixels by 255. This will keep the values under 1 and is known to be good for reducing computational load. Initial experiment is carried with a very basic architecture which includes the following layers:

- Convolutional layer with 32 filters and 3,3 kernel.
- A max pooling layer with 2,2 kernel.
- Convolutional layer with 64 filters and 3,3 kernel.
- A max pooling layer with 2,2 kernel.
- Convolutional layer with 128 filters and 3,3 kernel.
- A max pooling layer with 2,2 kernel.
- A fully connected layer with 64 neurons.
- Final output layer with activation function set as softmax.

Please note that all convolutional layers use ReLU as activation function. This model uses ADAM as optimizer and loss function is categorical cross entropy. Adam is passed learning rate as 0.01. The learning rate value was tested for 0.1, 0.01 and 0.001 values but 0.01 showed the best results. This architecture has achieved 67.5% accuracy on the test set. VGG has shown promising results in image classification tasks as reviewed in the previous works. This has inspired the author to experiment with VGG and attempt to improve the accuracy rate for FER2013 classification. The authors in [6] have introduced VGG and explained very well different variants of it.

Initially the author attempted to build a model using VGG-16 but due to increased the computational load on the machine it was decided that 3 blocks from the original VGG-16 architecture will be dropped to reduce the training time. The machine used for the experiments has following specifications:

- OS - macOS Monterey version 12.5.1
- Memory - 8 GB
- Chip - Apple M2

In deep neural networks dropout layer is often added for model regularization purposes. Co-adaptation of the neurons happens when the 2 or more neurons start working as a group and become interdependent on each other in the training process [10]. A dropout layer is known to break such interdependency by dropping some neurons randomly and set them to 0. It is generally a good practise to add a dropout layer after convolutional layers. The weight initialization is a very important step in the training. There are certain factors which are considered when initializing the weights. Weights should never be the same as it will force all the neurons to learn the same [11]. The weights should be small values. The weights should have good variance so that the neurons learn something new in every iteration through inputs. For this report the author has used a technique called He initialization. It works by initializing the weights of each layer with random values sampled from a Gaussian distribution with zero mean and variance calculated based on the number of inputs to the layer. This technique is

effective in preventing the vanishing gradient problem and it is commonly used with ReLU activation function. The author has used 'adam' as an optimizer. In the experiments it was found that the stochastic gradient descent performed poorly in comparison with adam. The optimizers were set with a learning rate set to 0.01 after a few experiments. The learning rate determines the step size of the weight updates in each iteration. If the step size is small the training takes longer to converge and if it is too large it may overshoot the most optimal solution. Monitoring the training loss would determine what value would be optimal for learning rate. Loss function is set to categorical_crossentropy which is a special type of cross-entropy function used for 2 or more classes.

So the final architecture is as follows:

- Convolutional layer with 64 filters and 3,3 kernel.
- Convolutional layer with 64 filters and 3,3 kernel.
- A max pooling layer with 2,2 kernel with strides (2,2).
- A dropout layer with 0.25 probability.
- Convolutional layer with 128 filters and 3,3 kernel.
- Convolutional layer with 128 filters and 3,3 kernel.
- A max pooling layer with 2,2 kernel with strides (2,2).
- A dropout layer with 0.25 probability.
- A fully connected layer with 512 neurons.
- Final output layer with activation function set as softmax.

This architecture has shown significant improvement in accuracy increasing it to 71.5%.

IV. RESULTS AND DISCUSSIONS

This section presents the results obtained from training and testing the CNN model on the FER-2013 dataset. As performance measure confusion matrix is employed for these experiments. Initial experiments were carried out with a basic architecture of CNN with convolutional layer each followed by a max pooling layer. The convolutional layers have filters 32, 64, 128 respectively. The fully connected layer has 64 neurons. The training time was around 5 minutes and a test accuracy of 67.5% was reported. In order to improve the accuracy rate a variant of VGG-16 was employed with 3 blocks less and a dropout layer added at the end of each block. This model had relatively higher training time which was around 35 minutes and managed to get a accuracy of 71.5%. The figure 3 shows the confusion matrix for this model.

The "LabelEncoder" encodes the labels in alphabetical order. Hence 0 is angry, 1 is happy, 2 is sad and 3 is surprise. As evident the model performed poorly for angry with 143 times it being miss-classified as sad. 97 times the model miss-classified sad as angry. Angry is classified correctly at 51.7% which is the lowest. Angry and sad share similar visual features such as furrowed eyebrows or narrowed eyes. This might be the reason of miss-classification. Happy is classified correctly at 83.6%.

Figure 5 shows the training and validation accuracy for each epoch and training and validation loss of the model for

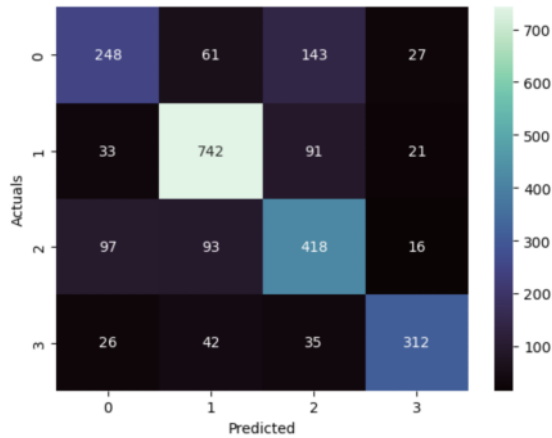


Fig. 3. Confusion matrix

each epoch. This training and validation loss graph helps us determine how well the model is able to generalize to new, unseen data. If the test loss starts to increase after a certain point it might indicate that the model is over-fitting to the training data. As evident in the figure 4 the validation loss is slowly increasing after 6th epoch and there is a gap created between training and validation loss.

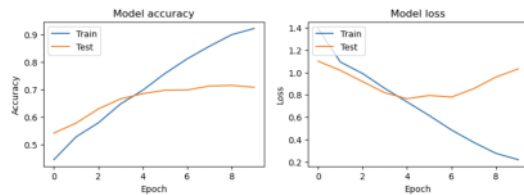


Fig. 4. Model accuracy and training/validation loss graph

V. CONCLUSIONS

In conclusion, this study aimed to explore the effectiveness of CNN for facial expression recognition using the FER2013 dataset. A couple of CNN architectures were implemented, their hyperparameters were tuned to achieve the best accuracy rate. The models were compared in terms of their accuracy and computational efficiency. There were no modifications done to the dataset, although only 4 classes were filtered out for the training and testing purposes. The best model achieved 71.5% accuracy so it still keeps a lot of room for improvement. The possible direction for future work would be to apply data augmentation techniques to increase the size of the dataset such that the neurons are able to learn more about the patterns. There is also a need for upgrading the machine used for training such as a RAM with 32 GB space. Probably employing a GPU for the training can bring down the training period tremendously. GPU is designed for parallel

computation. This will allow the training load to be split across multiple cores. This means in order to implement VGG-16 or VGG-19 completely without removal of any blocks and perhaps more dropout layers for better regularization. It was also seen that the validation loss was higher than the training loss resulting in over-fitting. This can be overcome by increasing the size of the dataset. An early stopping can also be applied if the validation loss starts going above test loss.

VI. APPENDICES

A. Appendix

Additional Resources:

- [Video link for the models training](#)

REFERENCES

- [1] I.Michael Revina, W.R. Sam Emmanuel, "A Survey on Human Face Expression Recognition Techniques", 2018
- [2] Arohan Ajit, Koustav Acharya, Abhishek Samanta, "A Review of Convolutional Neural Networks", 2020
- [3] Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition, Chapter 14", 2019
- [4] Ian J. Goodfellow, "Challenges in Representation Learning: A report on three machine learning contests", 2013
- [5] Christopher Pramerdorfer, Martin Kampel, "Facial Expression Recognition using Convolutional Neural Networks: State of the Art", 2016
- [6] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", 2014.
- [7] C.Szegedy, W.Liu, Y.Jia, P.Sermanet, S.Reed, D.Anguelov, D.Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions", 2015
- [8] Shekhar Singh, Fatma Nasoz, "Facial Expression Recognition with Convolutional Neural Networks", 2020
- [9] Abhinav Agrawal, Namita Mittal, "Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy", 2019
- [10] Sangchul Hahn, Heeyoul Choi, "Understanding dropout as an optimization trick", 2020
- [11] Meenal V. Narkhede, Prashant P. Bartakke Mukul S. Sutaone , "A review on weight initialization strategies for neural networks", 2021

FINAL GRADE

79 / 100

GENERAL COMMENTS

Instructor

A very good submission. Abstract's first few sentences to be reduced/ made shorter. Summary results included. Good.

References used are of a good quality in general and used in an appropriate manner.

Structure of report is good and follows expected format.

Review of related work covers appropriate material which is used later to inform the experiments and architectures used. This is good.

Summarize early parameter experimentation results in a table.

Explain why you chose the four classes.

Your plots labels need to match your figure titles - see fig 4 for example.

Your experiments and results are interesting - so good accuracies.

Pleased to see you provided a detailed report of your parameter values -good.

When discussing results please provide details of your results at the class level using established metrics. Place them in a table, or better still, use a bar graph for comparison.

Overall a good effort. Next time heed comments regarding results and conclusions,



Comment 1

Summarize early parameter experimentation results in a table.

GRADING FORM: SMU GENERIC_GJ

VIKRAM KALTA

0

RESEARCH-INFORMED LITERATURE

KNOWLEDGE AND UNDERSTANDING

ANALYSIS

PRACTICAL APPLICATION AND DEP.

SKILLS FOR PROFESSIONAL PRAC.