Multiclass Image classification on Static Facial Expression in the Wild Dataset using Convolutional Neural Network

Pranav Rawat

Computer Science and Information Technology Building, Research School of Computer Science, Australian National University, Acton ACT, 2601Canberra, Australia

u6637058@anu.edu.au

Abstract. Convolutional Neural Networks are widely used as feature extractors. In this report a convulutional network is introduced that does multiclass image classifcation on SFEW dataset [2]. There are two key aspects that are discussed. The former aims at implementation of a convolution neural network. The latter is about performance comparison with a simple neural network. The convolutional neural network is considered to establish accuracy baseline to the other versions of the neural netowork that were implemented. The final results of this report are also compared to various other research papers utilizing the same dataset.

Keywords: Convolutional Neural Networks; Facial Expressions; Multiclass Image Classification; Feature extractors.

1 Introduction

The field of Face and its emotions Recognition finds its application in diverse areas of research. Image processing, pattern recognition, and computer vision are relevant subjects to the face recognition field [1]. Human Body expresses different facial changes in order to respond to a person's internal state of mind when exposed to a certain environment. Facial Recognition and emotion detection find its use in various security systems deployed at public places by the government. Apart from security systems and biometric devices, Emotion detection is also deployed in smart cameras and features like Windows Hello. In this paper, A multiclass image classification problem is devised, and a convolutional neural network is implemented. The convolutional neural network performance is compared to different versions of neural network that were making use of the Static Facial Expressions in the Wild (SFEW) database [2] in past researches that were conducted.

1.1 Dataset Selection

Facial Recognition datasets are popular in various fields of research including computer vision. The dataset used in this research report is Static Facial Expressions in the Wild (SFEW) dataset [2]. For this research an extended version of the base dataset is used along with the base dataset. The base dataset contains 10 dimensions and 7 classes from 1 to 7. This dataset was an extraction from Acted Facial Expressions in the Wild Database. The classes/labels from 1 to 7 depict facial expressions like Anger, Disgust, Fear, Happy, Neutral, Sad & Surprise [3]. However, the extended version of this dataset has 675 images placed under 7 folders classified under their facial expressions as discussed above.

There are three major reasons to select this dataset. First reason is that the dataset has 675(674 used as one row had NaN values) instances which provides enough base for training data to train the neural network. Secondly, the dataset is very clear, and it is observed that the 10 dimensions are divided into two feature sets of 5; one for local phase quantization and the second for pyramid of histogram of gradient feature. Finally, there is a lot of research done in this area. Therefore, this dataset is utilized in other papers as well which broaden the scope of comparison of the results.

1.2 Problem & Modelling

When doing image classification using a simple neural network, often it is observed that there is loss of spatial orientation and case of too many parameters that increase the number of hidden layers. To resolve this issue, use of convolution neural network is suggested. One of the major focus of the report is to understand the use and performance of a convolutional neural network as feature extractors. The images in the extended dataset are used to build the model for convolutional neural network.

1.4 Use of Extended Dataset for CNN

For image classification, both the base dataset and the extended dataset are used together. Using the base dataset i.e. the csv file, labels and the image IDs(names) are extracted. The train file contains 580 image IDs with their labels and the test file contains 96 image IDs without their label. These labels that are missing in test files must be predicted by our convolutional neural network. These predicted outcomes are later stored in the sample submission file.

	A	B
1	id	label
2	50_50_001805320_00000009	4
3	DeepBlueSea_004442200_00000001	3
4	OceansTwelve_002840120_00000045	7
5	AlexEmma_004507280_00000022	5
6	DidYouHearAboutTheMorgans_003731807_000000	3
7	HarryPotter_Deathly_Hallows_1_003239120_0000	5
8	IAmSam_003343120_00000058	1
9	HauntingMollyHartely_002326760_00000018	2
10	OceansTwelve_003845720_00000019	7
11	OceansTwelve_002551440_00000006	1
12	AlexEmma_004117920_00000035	6

	A	B
1	id	
2	CryingGame_001419640_00000062	
3	Juno_003321880_00000052	
4	AlexEmma_003716360_00000040	
5	ThereIsSomethingAboutMary_005941880_00000039	
6	AmericanHistoryX_012427760_00000011	
7	HarryPotter_Deathly_Hallows_1_011317280_00000001	
8	MarotAtTheWedding_010431280_00000042	
9	Juno_011833760_00000059	
10	OceansTwelve_011847320_00000054	
11	Juno_001934160_00000002	
12	Bridesmaids_000059880_00000039	

Fig. 1. Depicting content inside the training and testing csv files extracted from the dataset

1.5 Methods of Analysis

The performance of the network is evaluated using various methods. Training accuracy is one of the methods that is utilized. In order to examine the training result in real time, following every epoch of the training, the analysis program will calculate the accuracy of the how network performs on the training dataset. The accuracy analysis on the training set cannot be enough to show the actual capacity of the network because the network may experience overfitting by the training and only recognize training instead of generalizing the patterns of the data [4]. Therefore, to mitigate this problem, the report also does analysis on the accuracy of testing data. Loss accurately depicts the learning done by the network making it an integral analysis method for this network [5]. For the convolutional neural network, Validation is also used along with other methods listed above.

2 Method

There are two versions of the network that are implemented in this report. The first version that is a simple neural network acts a baseline to measure and compare performance with the other version. As a second version, we have introduced the convolutional neural network that does multiclass image classification making use of the base and extended dataset.

2.1 Data Preprocessing for Simple Neural Network

The first basic step is to process the data. Making use of pandas, the data is loaded. Since there are no characters in the given dataset; the next step is to transfer data into numeric. Then, the pandas data frame will be converted into an array, which will be divided into x-array and y-array. Ultimately, both these arrays will be wrapped by tensors and variables so that the data can be feed into the network.

Note: Testing the data involves similar pre-processing steps.

2.2 Simple three-layer neural network

A simple three-layer neural network with 10 input neurons, 9 hidden neurons and 7 output neurons is implemented. In order to improve the performance of the neural network, the optimizer that is used is Adam. Cross-entropy is used as it is generally used for most classification networks. Also, other hyperparameters (like learning rate = 0.01) are kept constant. Testing and training accuracies were defined as two different function like in the case of the first version. Each of these functions were called during every epoch.

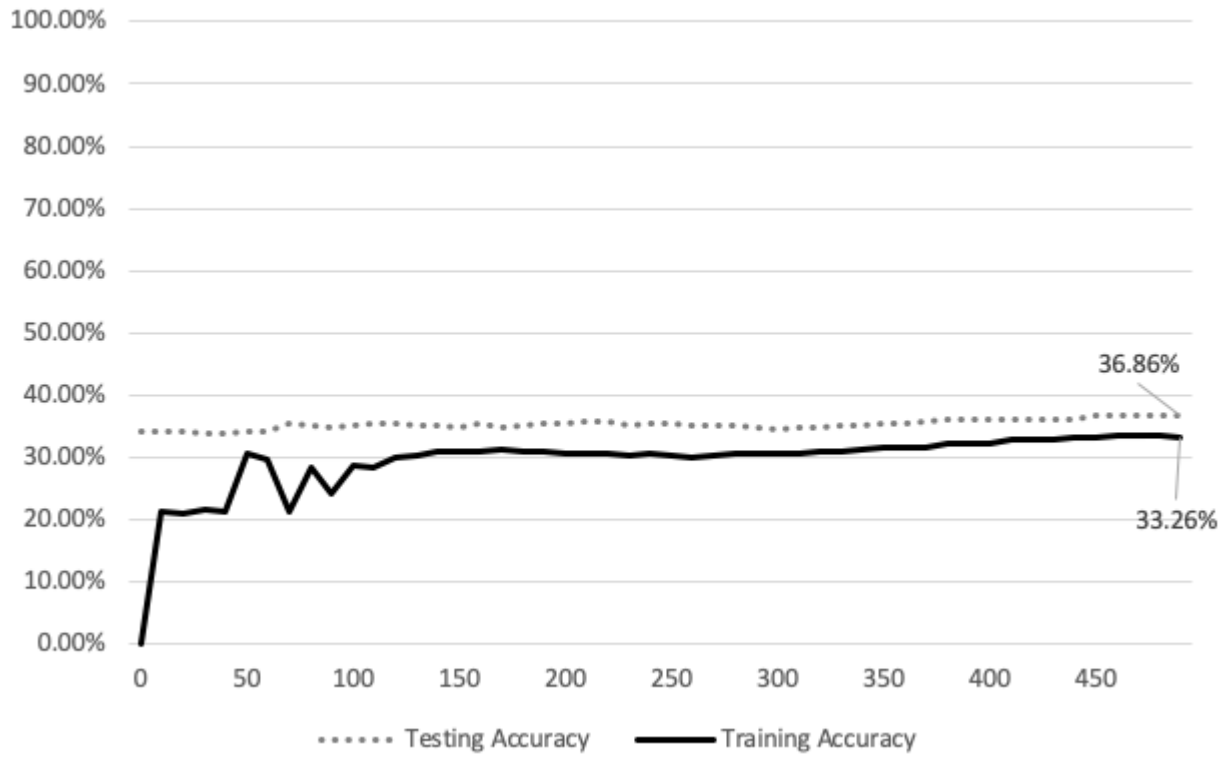

Fig. 2. Testing & Training Accuracy w.r.t number of epochs

Figure 2 depicts the training and testing accuracies of the network. It was observed that for both the accuracies started to drop around 200 epochs. However, both the accuracies were way above 30% by then. Finally, by 500th epoch the final accuracy for testing was 36.86% and training was 33.26%. However, the gap between training and testing does mean that there is a case of overfitting possible in the neural network.

372	0	1	0	2	0	0	0	0	1
0	368	4	1	0	0	1	1	8	6
0	0	375	2	0	0	1	0	1	1
0	0	1	378	0	5	0	1	1	3
0	5	0	0	365	0	7	1	5	4
1	0	3	1	0	363	0	0	1	7
0	3	0	0	1	0	373	0	0	0
0	2	1	1	4	0	0	377	0	2
2	16	0	1	6	5	1	0	347	2
2	8	0	5	8	1	0	9	3	346

Fig. 3 Confusion matrix for testing

Figure 3 gives the confusion matrix for testing. This matrix also certifies that the neural network can classify data over a reasonable range without any extreme errors.

2.3 Data Preprocessing for Convolutional Neural Network

The extended dataset had 675 images which contain 100 images in each folder angry, fear, happy, neutral, sad, surprise and 75 images in folder for disgust. All these images are of dimension 720x526. Therefore, the first step is to convert the size into 28x28 and normalize the pixel values. The images are read one by one and stack one over the other in an array. Division of pixels of the images by 255 is carried out so that the pixel values of images comes in the range [0,1]. This step is done to improve the overall performance of the CNN model.

2.4 Label Prediction using Convolutional Neural Network

The CNN model introduced in this paper has two Conv2d layers and a Linear layer followed by a dense fully connected layer to classify features under their respective labels. The linear layer A kernel of size 3x3 is used in both the Conv2d layers. Cross-entropy loss function is used as it is generally used for most classification network. There is 10% of the data in the validation set and the remaining in the training set. Accuracy is recorded as two separate lists for validation and training.

```
Net(
  (cnn_layers): Sequential(
    (0): Conv2d(1, 4, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
    (1): BatchNorm2d(4, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (4): Conv2d(4, 4, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
    (5): BatchNorm2d(4, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (6): ReLU(inplace=True)
    (7): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  )
  (linear_layers): Sequential(
    (0): Linear(in_features=196, out_features=7, bias=True)
  )
)
```

Fig. 4 CNN model architecture

After training the network and calculating losses for training & validation, 95 test images are pre-processed the same way as for training mentioned above and predictions are generated for the test set. These predictions are saved and stored in the sample submission file. The number of epochs is kept at 500 and the learning rate is 0.01. Most hyperparameters are kept constant.

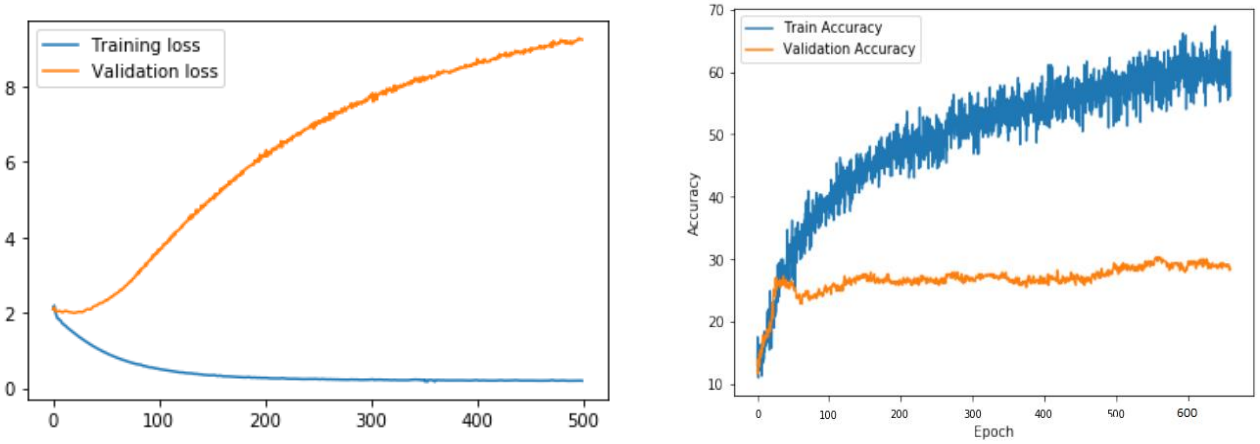

Fig. 5 The left figure plots the train loss and validation loss. The right figure plots the training accuracy and validation accuracy of the convolutional neural network

The left plot in figure 5 visualizes the validation and training loss, it is observed that there might be a case that the model is not able to generalize well on the validation set The right figure plot given in Figure 5 presents the training and validation accuracy of the convolutional neural network. The training accuracy reaches a maximum of 69.3 % and the validation accuracy is 26.7%.

3 Results and Discussion

This report has proposed a convolutional neural network for label prediction on the dataset. Along with that in the early section of the report, a simple neural network is also implemented. Every version of these networks account for their accuracies on both testing and training/validation dataset which are considered as key evaluation methods. To further evaluate the work of this report, a comparison of performance between these two networks and SPI baseline of the SFEW dataset [2].

As compared to the simple neural network where the best training accuracy was 33.26%, the proposed convolutional neural network reached at a 69.7% training accuracy. Therefore, it is concluded that the convolutional neural network has a better accuracy as compared to the neural network. The SPI baseline [2] for the dataset is average 19% for validation accuracy. The proposed convolution network had an average validation accuracy of 24%.

4 Conclusion and Future Work

The report introduced a convolutional neural network for image classification. It is observed how CNNs can be useful for extracting features from images. It highlights the superiority of using convolutional neural network over simple neural network. The accuracy difference between the two network implementations explicitly state that convolutional neural network has better performance. As compared to the SPI baseline [2], the best result of the convolutional neural network improves the average accuracy by approximately 11%.

For future work, the hyperparameters of the CNN model can be adjusted and further tuned to attain even better accuracy levels. An area of investigation to optimize network performance is that apart from the learning rate & batch size, other hyper parameters must be studied that affect the performance. Detailed study on these hyperparameters to tune like the number of filters in each convolutional layer, number of convolutional layers, number of epochs, number of dense layers, number of hidden units in each dense layer can be conducted [6]. The CNN model proposed can be tested on other dataset which have large number of images inside them like MNIST and CIFAR-10 [7].

References

1. Kak, Shakir & Mustafa, Firas & Valente, Pedro. (2018). A Review of Person Recognition Based on Face Model. 4. 157-168. 10.23918/eajse.v4i1sip157.
2. Dhall, A., Goecke, R., Lucey, S., & Gedeon, T. (2011, November). Static facial expressions in tough conditions: Data, evaluation protocol and benchmark. In 1st IEEE International Workshop on Benchmarking Facial Image Analysis Technologies BeFIT, ICCV2011.
3. A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Acted Facial Expressions in the Wild Database. In Technical Report, 2010.
4. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15, 1929 - 1958.
5. N. Murata, S. Yoshizawa and S. Amari, "Network information criterion-determining the number of hidden units for an artificial neural network model," in IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 865-872, Nov. 1994.
6. Z. He, B. Gong and D. Fan, "Optimize deep convolutional neural network with ternarized weights and high accuracy," in 2019, . DOI: 10.1109/WACV.2019.00102.
7. R. F. Alvear-Sandoval, J. L. Sancho-Gómez and A. R. Figueiras-Vidal, "On improving CNNs performance: The case of MNIST," *Information Fusion*, vol. 52, pp. 106-109, 2019.