Emotion Detection through Static Images using Pruned Convolutional Neural Networks

Jihirshu Narayan , Wenbo Ge

Australian National University, Canberra ACT, Australia
{jihirshu.narayan, wenbo.ge}@anu.edu.au

Abstract. Image classification is perhaps one of the most widely implemented application of Neural Networks in industry today and Convolutional Neural Network is the most effective tool for the same. However, the most basic requirement of a such a deep neural network is a massive dataset. We aimed to demonstrate that CNNs, without the use of pre-trained models, can produce reasonable multi label image classification results even when the model is trained on a dataset of modest size. Additionally, we aimed to demonstrate the effects of feature map pruning on a deep convolutional neural network. It was observed that feature map pruning based on output vector in the pattern space is an effective method of building lighter neural networks without compromising on the model performance. The model retained its predictive accuracy even after the feature maps in the last layer were pruned by as much as 75% in some instances.

Keywords: Image Classification, Pruning, Emotion Detection, Deep Neural Network, Convolutional Neural Network

1 Introduction

Image processing is rapidly developing and being put to use on an unprecedented scale. Its applications are diverse in all possible aspects, ranging from trivial tasks like recognizing individuals on social media to highly critical tasks such as object recognition in autonomous driving. Deep Neural Network is one of the tools that has been immensely successful in image classification. Emotion detection is a very specific application of image processing. Due to the dynamic nature of facial expression, its features are best extracted from videos. However, under certain circumstances, it becomes imperative that these features are extracted from static images. This classification problem is quite advanced and complicated due to the fact that it has multiple target classes and it's difficult to generalize emotional expressions across individuals. For the purpose of this report we will be using the dataset presented by Dhall et al. [3]. The database is a static facial expression database that comprises of unconstrained expressions captured from various angles and at varying resolution. The images are diverse in terms of pose, spatial orientation, subject age, etc. It is named as Static Facial Expressions in the Wild (SFEW) database [3].

The classification model used for SFEW dataset by Dhall et al. [3] is based on a non-linear support vector machine which achieved an averaged accuracy of 19% on the Strictly Person Independent dataset. We believe that similar results, if not better, can be achieved with a convolutional neural network model, which in recent times, has emerged as one of the most popular tools for image classification, especially in single label image classification [10]. If the provided database is large and diverse enough, CNNs can handle multiple labels efficiently as well [1]. The designed neural networks must be large and robust enough to learn from massive datasets, while at the same time, they should be capable of producing reasonable results with smaller datasets as well. Pruning is one of the most effective methods through which a neural network can be reduced in size while maintaining its efficiency, resulting in a lighter and faster model [4]. In this paper, we present a model that performs reasonably well across datasets of varied size. To build a more compact model that works well with small datasets, we extend the idea of neuron pruning based on the concept of "Distinctiveness" [4] by applying it as pruning technique on a feature map level. We also provide an analysis of the relationship between the threshold angle for pruning with respect to the model accuracy and how the selection of the correct threshold angle may reduce overfitting in the model.

2 Dataset

The SFEW comprises of frames selected from AFEW (Acted Facial Expressions in the Wild). It contains a total of 700 images of multiple subjects that vary on various aspects such as resolution, angles and illumination [3]. The 700 samples are labeled with 7 expressions; angry, disgust, fear, happy, sad, surprise and neutral. Each sample had a 720x576 pixel resolution. We also utilized the Kaggle dataset [12], which is a facial expression dataset with the exact same classification labels, to demonstrate how our model performed on larger datasets. It has around 37,000 grayscale images with a 48x48 pixel resolution.

3 Methodology

Kotsiantis et al. [6] state that data pre-processing affects the generalizing performance of any supervised learning model notably. It is therefore extremely important that we clean and process the data in such a way that it facilitates pattern learning by the model. Considering that the task focuses on emotion detection, the first step was to detect faces in each image using a combination of multi-task cascaded convolutional neural network and haar cascade classifier. Once the face was detected, it was cropped out and resized to 48x48 pixels. The training dataset was also augmented for each epoch of training by random rotation, addition of noise to brightness, contrast and saturation values, and randomly flipping the image across the vertical axis. Such data augmentation methods are necessary in order to expose the learning model to more variety, thus resulting in a more robust classifier. It is especially relevant in our case because in terms of deep

neural networks, a dataset of 675 samples is considered quite small. For instance, a similar 7 label emotion detection classifier based on CNN, achieved an average accuracy of 65% while working on a dataset comprising of 37,000 images provided by Kaggle website [1].

3.1 Convolutional Neural Network Model

Local descriptor-based feature sets, such as PHOG [2], SIFT, LPQ [8], etc. have been traditionally employed for the purpose of image recognition. However, in recent years, the popularity of these models has been overshadowed by the emergence of CNN. CNNs extract features through layers and thus produce better generalization [11]. This aspect of learning through CNN is extremely relevant for image recognition because there is high variance in features even in the same target class, thus making the nature of image recognition much more abstract when compared to other forms of data-based learning.

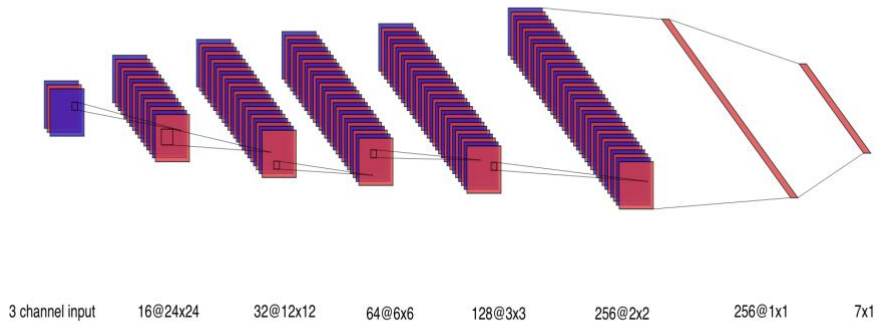

Fig 1. Visual representation of the feature maps

Our CNN model consists of 6 convolution module and each one of them further consists of a convolution layer, batch normalization layer and a pooling layer. The first five layers are summarized through max pool method while the last layer uses an adaptive average pooling method. The output size of the first 5 layers gradually increases from 16 to 256 and the last layer condenses the information to 7 output neurons, one for each target label. A visual representation of the feature map obtained as the output of each layer is shown in Fig 1.

Hyperparameter Tuning is the most important step while optimizing any learning model. Pinto et al. [9] argue that image classification performance can be improved by finding the optimum hyperparameters rather than inventing new learning techniques. Since the pruning technique being implemented for our model is based on the angle between feature map output vectors in the pattern space [4], the threshold angle for feature map removal was also treated as a hyperparameter. Contradictory to the 15° threshold suggested [4], it was found that a threshold angle of 8° - 10° removed the optimal number of feature maps so that the model performance was not affected. A learning rate of 0.0001 was found to be most effective. The number of epochs of training also has a bigger impact on our model accuracy than in general. The reason for this behavior lies in our method of data augmentation, which was discussed earlier. We do not augment the dataset to a fixed size. Rather, we modify the dataset based on random parameters for each epoch of training. Therefore, from the model's perspective, a higher number of epochs of training means that it has more variety of data available at its disposal.

For the purpose of model testing, we split the original dataset into 3 sets. 80% of the original data was kept for training while the remaining 20% was split into test and validation sets. The test set was used to keep track of the accuracy through each epoch of training and the validation set was used as a final hurdle to evaluate the model.

3.2 Pruning

When constructing a neural network, the designer has no empirical method to determine the correct size of the network. It completely depends on the dataset and the pattern that resides within it. Data visualization and exploration may provide some sense regarding it, but ultimately it comes down to logical guess work, instinct and experience. As such, it is not uncommon for designers to create networks that are larger than required. Network pruning is a useful tool in such cases as it helps to trim down the network to its lightest version without compromising its accuracy.

Network pruning methods have been developed on various levels such as pruning based on weights [5], filter pruning [7] and neuron pruning [4]. We have used the feature map pruning method based on the concept of "distinctiveness" [4]. The degree of distinctiveness is measured by the angle between vectors formed by the layer 5 output activation values formed in the pattern space [4]. In simpler words, it means that each feature map output value obtained by an instance of sample data serves as a co-ordinate for the n-dimensional vector, where n is the total number of samples in the data. Thus, we have a vector associated with each feature map and we determine the angle between all possible pairs. In order to get values ranging from 0° to 180°, the vectors are normalized in the range [-0.5, 0.5]. A close to 0° angle represents feature maps providing similar functionality and an angle nearing 180° represents complimentary feature maps. If similar feature maps are found, one of them is neutralized and its weight is added to the other. If complementary feature maps are found, both are neutralized because they were cancelling out each other's functionality anyway. The threshold for

pruning should be tuned like a hyperparameter and we will see the effects of a large threshold in the next section.

Once the angles for each pair of feature map vectors are calculated, we can start the process of neutralizing the redundant feature maps. It is important to keep track of the feature maps that have been removed so that over the course of the entire process, we do not end up removing both the feature maps of a similar pair. It is also advisable to keep a back-up of the original weights so that it can be referred when we need to add the weight of a feature map that has been already removed. After updating the weights, the validation data is processed through the network and the model accuracy is calculated again. For our model, we pruned the weights from the final convolution layer to the averaging layer. It should be noted that since we are dealing with weights originating from a convolution layer, the weight matrix is 3 dimensional. We operate on the weight dimension representing an individual feature map. So, if two feature maps are providing similar or complimentary functionality, the entire weight matrix for that feature map is neutralized. We use the newly calculated model accuracy to compare how the pruning process has affected the model performance.

4 Results and Discussion

Dhall et al. [3] reported an average accuracy of 19 % for the Strictly Person Independent category. The results were achieved with using one of the principal component feature set at a time classified by a non-linear support vector machine. The low accuracy is attributed to the complex nature of problem and large number of target classes [3]. Our best performing model achieved an accuracy of 49.09%. In order to demonstrate that our model works well with larger datasets as well, we trained and tested our model with the same hyper-parameters on the Kaggle dataset [12] used by Alizadeh and Fazel [1] and achieved a validation accuracy of 61.5 %.

Table 1. Convolutional Neural Network Performance Analysis on the SFEW dataset

Feature Maps in Final Layer	Epochs	Learning Rate	Threshold Angle for Pruning	Average Test Accuracy	Validation Accuracy	Number of Feature Maps Pruned	Validation Accuracy after Pruning
256	10	0.001	8º, 172º	21.83%	21.82%	111	25.45%
256	10	0.0001	8º, 172º	26.05%	32.73%	7	34.55%
256	20	0.001	10º, 170º	26.83%	45.45%	148	43.64%
256	20	0.0001	10º, 170º	27.81%	41.82%	36	43.64%
256	40	0.0001	8º, 172º	30.11%	49.09%	16	43.64%
256	50	0.001	10º, 170º	33.74%	32.73%	191	36.36%
256	50	0.0001	10º, 170º	31.15%	40.00%	43	40.00%

The pruning process was only applied to the last global averaging layer. Even a threshold of 8° had observable impact on accuracy. We noticed that a lot of iterations showed a slight positive growth in accuracy in spite of significant feature map pruning. This increase in accuracy might be an indication of overfitting in the original model. Overall, we can safely claim that feature map pruning based on the concept of "Distinctiveness" [4] can be effectively applied to even deep CNN based neural networks. A complete summary of results for this model is listed in Table 1.

As mentioned before, the angle between the two activation value vectors in the pattern space represents the relationship between the functionality provided by two feature maps. This quantifiable attribute helps us in identifying similar and complimentary feature maps. In Fig 2, we can observe the effect of varying threshold angle for pruning on model performance. We can observe that with low threshold values, there is minimal pruning which has no noticeable effect on the model performance. If the model is overfitted, which is the case in the example we have chosen, the model performance improves with increasing threshold angle, before deteriorating again due to loss of vital feature maps. Such an exercise for model designers would help them identify the optimum size for their networks for a given dataset.

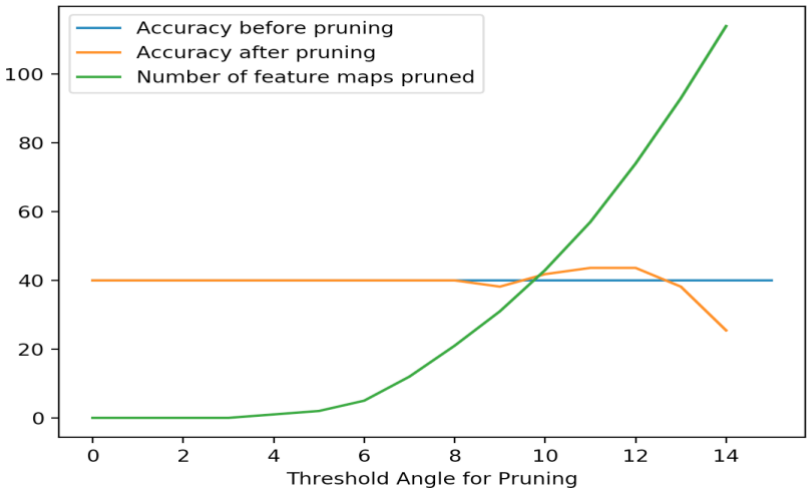

Fig 2. Effect of varying Threshold Angle

5 Conclusion and Future Work

The demand for efficient image classification models is rising and it seems unlikely that it is going to decline in the near future. Since there is a lot of scope for

improvements as well, we can expect significant development in the coming years. For instance, the concepts we have discussed in this paper need to be built upon for dynamic application such as image classification in videos. Pruning techniques such as the one implemented in this paper will aid the development of models with a shorter response time.

The model accuracy obtained in this report is significantly better than the one reported by Dhall et al. [3]. Despite the improved model accuracy, we believe that there is a considerable scope of improvement in the prediction accuracy. The use of pre-trained models has proven immensely successful in recent years [1]. These models are trained on a larger dataset and therefore they are able to extract and generalize features better. The output of these models can then be connected to our own custom designed fully connected layers which can be further trained on our own dataset, keeping the weights of the pre-trained model fixed. However, it must be noted that such models would require more resources such as a GPU (Graphics Processing Unit) or Cloud TPU (Tensor Processing Unit). It would also be interesting to explore pruning in convolution layers based on the concept of "Distinctiveness" [4] in a more comprehensive manner, unlike how we have limited our pruning to the last layer. Moreover, there is scope for pruning entire convolution layers, which would result in a much lighter model. We can expect that a filter level pruning technique would be highly successful in such a model [7]. Overall, we believe that this paper would serve as a solid foundation for future work related to image classification on small datasets.

References

1. Alizadeh, S. and Fazel, A., 2017. Convolutional Neural Networks for Facial Expression Recognition, 1704.06756.
2. Dalal, N. and Triggs, B., 2005, June. Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 886-893). IEEE.
3. Dhall, A., Goecke, R., Lucey, S. and Gedeon, T., 2011, November. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops) (pp. 2106-2112). IEEE.
4. Gedeon, T.D. and Harris, D., 1991. Network reduction techniques. In Proceedings International Conference on Neural Networks Methodologies and Applications (Vol. 1, pp. 119-126).
5. Han, S., Pool, J., Tran, J. and Dally, W., 2015. Learning both weights and connections for efficient neural network. In Advances in neural information processing systems (pp. 1135-1143).
6. Kotsiantis, S.B., Kanellopoulos, D. and Pintelas, P.E., 2006. Data preprocessing for supervised leaning. International Journal of Computer Science, 1(2), pp.111-117.

7. Luo, J.H., Wu, J. and Lin, W., 2017. Thinet: A filter level pruning method for deep neural network compression. In Proceedings of the IEEE international conference on computer vision (pp. 5058-5066).
8. Ojansivu, V. and Heikkilä, J., 2008, July. Blur insensitive texture classification using local phase quantization. In International conference on image and signal processing (pp. 236-243). Springer, Berlin, Heidelberg.
9. Pinto, N., Doukhan, D., DiCarlo, J.J. and Cox, D.D., 2009. A high-throughput screening approach to discovering good forms of biologically inspired visual representation. PLoS computational biology, 5(11).
10. Wei, Y., Xia, W., Lin, M., Huang, J., Ni, B., Dong, J., Zhao, Y. and Yan, S., 2015. HCP: A flexible CNN framework for multi-label image classification. IEEE transactions on pattern analysis and machine intelligence, 38(9), pp.1901-1907.
11. Zheng, L., Yang, Y. and Tian, Q., 2017. SIFT meets CNN: A decade survey of instance retrieval. IEEE transactions on pattern analysis and machine intelligence, 40(5), pp.1224-1244.
12. I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio. Challenges in representation learning: A report on three machine learning contests. Neural Networks, 64:59--63, 2015. Special Issue on "Deep Learning of Representations", <https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data>