

PSNet: Parametric Sigmoid Norm Based CNN for Face Recognition

Yash Srivastava, Vaishnav Murali and Shiv Ram Dubey

Computer Vision Group,

Indian Institute of Information Technology, Sri City, Chittoor, Andhra Pradesh, India

{srivastava.y15, murali.v15, srdubey}@iiits.in

Abstract—The Convolutional Neural Networks (CNN) have become very popular recently due to its outstanding performance in various computer vision applications. It is also used over widely studied face recognition problem. However, the existing layers of CNN are unable to cope with the problem of hard examples which generally produce lower class scores. Thus, the existing methods become biased towards the easy examples. In this paper, we resolve this problem by incorporating a Parametric Sigmoid Norm (PSN) layer just before the final fully-connected layer. We propose a PSNet CNN model by using the PSN layer. The PSN layer facilitates high gradient flow for harder examples as compared to easy examples. Thus, it forces the network to learn the visual features of difficult examples. We conduct the face recognition experiments to test the performance of PSN layer. The suitability of the PSN layer with different loss functions is also experimented. The widely used YouTube Faces (YTF) and Labeled Faces in the Wild (LFW) datasets are used in the experiments. The experimental results confirm the relevance of the proposed PSN layer.

Index Terms—Sigmoid Function, Deep Learning, Face Recognition, Parametric Sigmoid Layer, PSNet.

I. INTRODUCTION

The face recognition is a very challenging problem due to several factors ranging from the personal to environmental factors. A person can make different appearances of the face by changing the hair style, beard style, cap, glasses, etc. Many times the capturing device is not able to capture very good face image and contains various challenges such as difficult pose, bad lighting condition, etc. However, the face based recognition system has a huge potential to automate many applications in real-life.

Nowadays, the deep learning has pushed the performance of any recognition system with a great success in forward direction. The first Convolutional Neural Network (CNN), named as AlexNet [1] introduced by Krizhevsky et al. completely changed the dynamics in visual recognition area. Finally, the AlexNet became a revolutionary model for the problem of image categorization. It also won the ImageNet Large Scale Challenge in 2012. After that, various CNN based architectures have been proposed for different applications such as VGGNet [2], GoogleNet [3], ResNet [4] for image classification; Local Bit-plane Decoded CNN [5], ChexNet [6] and RCCNet [7] for biomedical image retrieval/classification; CNN for face spoof detection [8], [9]; and many more.

The CNN has also shown its impact in the face recognition area as well. The DeepFace [10], DeepID2 [11], FaceNet

[12] and two-branch CNN [13] are few state-of-the-art methods which have shown a significant performance gain when compared to the hand-crafted features. This advancement is achieved by the evolution of face datasets at scale for testing and training of the CNN models. The major face datasets include Youtube Faces (YTF) [14], CASIA-Webface [15] and Labeled Faces in the Wild (LFW) [16] among other face datasets.

The study of Convolutional Neural Networks over time have shown that the deep CNNs are generally better in performance when compared to the shallower networks. It leads to the rise of deep architectures like ResNet [4] and Inception [17] which use multiple trainable convolutional layers to achieve higher accuracy. At the same time, experiments showed that after certain depth, not much increment in accuracy was observed, i.e., the performance saturated at great depths [4]. Moreover, the rise of large-scale face recognition leading to heavy architectures would require significant resources to run the recognition systems which hinders its applications on resource constrained devices. Considering the above two points, researchers have started exploring and working over different parts of the architecture system like non-linearity layers, loss functions, and many more. The Rectified Linear Unit (ReLU) [1], Parametric ReLU [18], Average Biased ReLU [19], and Linearly Scaled Hyperbolic Tangent (LiSHT) [20]. Various loss functions have been developed like ArcFace [21], SphereFace [22], and Hard-Mining Loss [23], specifically designed for the face recognition task. Other type of layers used in CNN is Dropconnect [24], Dropout [25], Batch Normalization [26], etc. A performance analysis of loss functions over face datasets is conducted in [27].

Most of these types of layers are proposed to increase the learning capacity and generalization ability of CNN models. However, it turns out that after a certain limit the correctly classified samples introduce the biasedness and hampers the learning of incorrectly classified samples. We believe that it happens because the abstract features for easy examples become different from the difficult examples. This paper tackles this problem by introducing a parametric sigmoid norm layer just before the final class score layer to force the abstract features of both easy and hard examples to be closer to each other. It leads to the better training of model for harder examples also.

In this paper, first we present the proposed Parametric-

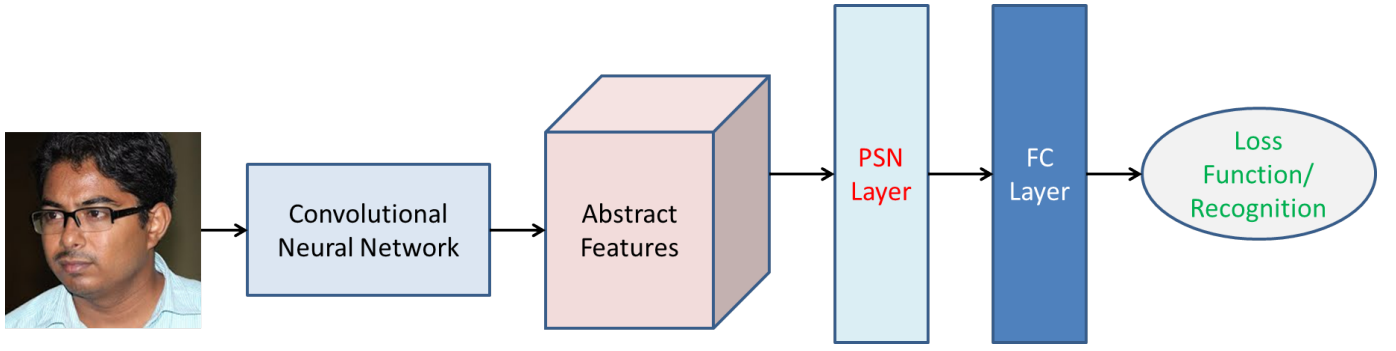


Fig. 1. The proposed PSNet CNN model for face recognition with Parametric-Sigmoid-Norm (PSN) layer represented. Given an input image, the image is passed through the convolutional layers of the CNN and features are obtained just before the final class score layer. These feature tensors are passed through the Parametrize-Sigmoid-Norm layer to enforce the feature distribution as close as possible for both easy as well as hard examples. It leads to better learning of hard examples also along with easy examples. The face image taken in this illustration is provided by one of the author this paper.

Sigmoid-Norm (PSN) layer in Section 2. We also discuss its importance just before the final fully-connected layer. The experimental setup, architecture details and datasets used is presented in Section 3. The experimental results and analysis are illustrated in Section 4. Finally, the concluding remarks are summarized with future directions in Section 5.

II. PROPOSED PARAMETRIC-SIGMOID-NORM LAYER

Considering a range of available face datasets like CASIA-Webface [15], LFW [16] among others, we observe that these datasets have many face images from a single identity. Due to this, there is a big cluster of easy samples which might be far in feature space from the significantly small cluster of hard samples when the distance is measured between the centroids of both these clusters. This leads to a bias in the model which is trained on such a dataset and might perform poorly in real-case scenarios. We tackle this problem by enforcing the features to follow the parametric sigmoid norm. Generally, sigmoid is used as the loss as it converts the input into the range of 0 and 1. We leverage this property of sigmoid over the CNN features before class score computation. The sigmoid is defined as follows:

$$\mathcal{F}_{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

where x is the input and $\mathcal{F}_{Sigmoid}(x) \in [0, 1]$. Note that using this sigmoid over abstract CNN features might disturb the distribution in an adverse manner. In order to avoid it, we use the parametric sigmoid which allows some flexibility in sigmoid using some parameters. The parametric sigmoid function is defined as follows:

$$\mathcal{F}_{PSigmoid}(x) = \frac{\alpha}{1 + e^{-\beta(x-\gamma)}} \quad (2)$$

where α , β and γ are the parameters and/or hyper-parameters which have been kept either trainable or fixed under different setting scenarios. Keeping α equal to 1, the parametric-sigmoid function clips the output value between 0 to 1. Note that the proposed PSN layer uses $\mathcal{F}_{PSigmoid}$ function for all its inputs. The effect of the application of the above function helps to bring the centroids of hard and easy sample clusters

closer in abstract feature space. Hence, making it easier for the model to learn the training dataset irrespective of easy or hard examples.

We use the parametric-sigmoid-norm (PSN) layer with the existing CNN as shown in Fig. 1. The PSN layer is used just before the final fully-connected layer (i.e., class score layer) to enforce the abstract features of all samples to follow a certain distribution irrespective of its difficulty level. Basically, it tries to reduce the cluster distance between easy and hard samples for a class. In other words, it generalizes the capacity of model with high priority for harder examples.

III. EXPERIMENTAL SETUP

In this section, first we introduce the CNN models used for face recognition, then we provide a summary of training and testing datasets used, and finally we state the other settings related to training of networks.

A. CNN Architectures

The residual networks (ResNet) have shown very appealing results in various applications of computer vision [4]. We use the ResNet model along with proposed parametric sigmoid norm (PSN) layer as the PSNet model. Thus, PSNet18 model consists of ResNet18 and PSN layer and PSNet50 model consists of ResNet50 and PSN layer.

B. Training Datasets

The CASIA-Webface face dataset [15] is adapted in this paper for the training purpose. This dataset is used for training in most of the recent work of the face recognition. This dataset has 10,575 persons leading to total of 4,94,414 face images. Thus, the number of classes is 10,575 in CASIA-WebFace dataset.

C. Testing Datasets

The YouTube Faces (YTF) [14] and Labeled Faces in the Wild (LFW) [16] are used as our testing datasets. A total of 5749 different people leading to 13233 images is present in the LFW dataset. However, 3425 videos from 1595 identities are present in the YTF dataset. The image frame numbers are

TABLE I

THE EXPERIMENTAL RESULTS WITH DIFFERENT SETTINGS OF PROPOSED PSNET MODEL (I.E., THE PARAMETRIC-SIGMOID-NORM (PSN) ON RESNET ARCHITECTURES). THE RESULTS ARE COMPARED USING PSNet18 (I.E., RESNET18 WITH PSN) AND PSNet50 (I.E., RESNET50 WITH PSN) MODELS. THE TRAINING IS DONE OVER CASIA-WebFace FACE DATASET. HOWEVER, THE TESTING IS PERFORMED USING LFW AND YTF FACE DATASETS. HERE, THE TERM *Trainable* MEANS THE CORRESPONDING HYPERPARAMETER IS LEARNED AUTOMATICALLY THROUGH TRAINING.

Model	Parametric-Sigmoid-Norm Hyperparameters	Accuracy on LFW	Accuracy on YTF
PSNet18	Without PSN (i.e., ResNet18)	95.35	91.80
	$\alpha = 1, \beta = 20, \gamma = 1$	95.68	92.60
	$\alpha = \text{Trainable}, \beta = 20, \gamma = 1$	95.12	91.90
	$\alpha = 1, \beta = \text{Trainable}, \gamma = 1$	96.40	93.34
	$\alpha = 1, \beta = 20, \gamma = \text{Trainable}$	96.09	93.43
	$\alpha = 1, \beta = \text{Trainable}, \gamma = \text{Trainable}$	96.87	94.10
	$\alpha = \text{Trainable}, \beta = \text{Trainable}, \gamma = \text{Trainable}$	95.95	93.20
PSNet50	Without PSN (i.e., ResNet50)	97.42	94.50
	$\alpha = 1, \beta = 20, \gamma = 1$	97.50	94.47
	$\alpha = \text{Trainable}, \beta = 20, \gamma = 1$	96.98	94.34
	$\alpha = 1, \beta = \text{Trainable}, \gamma = 1$	97.71	94.64
	$\alpha = 1, \beta = 20, \gamma = \text{Trainable}$	97.89	94.67
	$\alpha = 1, \beta = \text{Trainable}, \gamma = \text{Trainable}$	98.10	94.90
	$\alpha = \text{Trainable}, \beta = \text{Trainable}, \gamma = \text{Trainable}$	97.53	94.33

TABLE II

THE EXPERIMENTAL RESULTS COMPARISON USING ANGULAR-SOFTMAX AND ARCFACE LOSS FUNCTIONS OVER RESNET AND PSNET CNN MODELS. THE RESULTS ARE COMPARED FOR DIFFERENT NETWORK DEPTH SUCH AS 18 AND 50. THE CASIA-WebFace DATASET IS USED FOR TRAINING. THE TESTING IS PERFORMED USING LFW AND YTF FACE DATASETS. THE HYPERPARAMETER SETTINGS IN PSNET ARE KEPT AS $\alpha = 1$ AND β & γ ARE NETWORK TRAINABLE.

Loss Function	Architecture	Performance on LFW	Performance on YTF
Angular-Softmax	ResNet18	97.12	93.90
	PSNet18	97.28	94.22
ArcFace	ResNet18	97.79	94.54
	PSNet18	97.78	94.65
Angular-Softmax	ResNet50	99.10	96.10
	PSNet50	99.21	96.50
ArcFace	ResNet50	99.20	96.30
	PSNet50	99.26	97.10

provided to be used for training and testing purpose in the YTF dataset. We use the standard as per the LFW benchmark for face verification and report the performance for both test face datasets in terms of the accuracy. These accuracies have been used as the performance measure in face recognition works and hence reported here also for comparison.

D. Input Data and Network Settings

We use the MTCNN [28] model for the alignment of the face images similar to [21], [22]. A normalization of face images is performed for each pixel by dividing by 128 after subtracting 127.5. The training is performed with 64 batch size for 20 epochs for each model. The learning rate is initialized with 0.01 and reduced by a factor of 10 at the 8th, 12th and 16th epochs. The Stochastic Gradient Descent (SGD) with momentum optimization technique is used with default settings for training using backpropagation.

IV. EXPERIMENTS, RESULTS AND OBSERVATIONS

In this section, first we present the experiments and results analysis of proposal PSNet method and then we show its generalization using different loss functions.

A. Result Analysis

We summarize the experimental results in Table 1 using the proposed PSNet model with the PSN layer with depth 18 (i.e.,

PSNet18) and 50 (i.e., PSNet50). The training is done over CASIA-WebFace dataset and testing is done over YTF and LFW datasets. The Cross-entropy loss is used in this experiment. The results are compared under different settings of PSN layer such as α , β , and γ are either fixed or trainable. If it is not trainable, then it is fixed based empirical observation. Note that under first setting, PSN layer is not used, thus it is same as the ResNet model. It is discovered that the performance of PSNet18 as well as PSNet50 is better when $\alpha = 1$ and β and γ are trained using backpropagation. The performance of PSNet18 and PSNet50 is significantly improved as compared to ResNet18 and ResNet50, respectively under best setting of PSN layer. Another important finding is that the proposed PSNet performs relatively much better when the depth of the network is less which can lead to better lightweight models for resource constrained devices.

B. PSNet with Other Losses

In this paper, we compare the results with existing methods in terms of the existing loss functions. The softmax loss is used to compute the results discussed in the previous section. We also compute the results of ResNet and PSNet using specialized loss functions including Angular-Softmax [22] and ArcFace [21] to show the generalization ability of the proposed PSNet model. Table 2 illustrates the results for ResNet18,

PSNet18, ResNet50, and PSNet50 using Angular-Softmax and ArcFace losses. The training is done using CASIA-WebFace dataset and testing is done over YTF and LFW face datasets. It is evident from the experimental results using different loss functions that the proposed PSNet model outperforms the state-of-the-art ResNet model for the same depth. It shows the importance of PSN layer which converts some of hard examples into easy by enforcing the parametric sigmoid norm in abstract feature space.

As we can see from the experimental results, with Loss functions like ArcFace and SphereFace, there is an improvement in performance with the usage of shallower networks. The proposed PSN layer is added to reduce the distance between hard and easy faces. The added robustness leads to the improvement in accuracy over LFW and YTF faces.

V. CONCLUSION

This paper introduces a parametric sigmoid norm (PSN) layer in the abstract feature space of CNN to enforce the better learning of data irrespective of its difficulty level. The proposed PSNet model introduces the PSN layer just before the final class score layer of ResNet. The face recognition experiments are performed to judge the performance of the proposed model. The results are computed over widely adapted LFW and YTF face datasets while CASIA-WebFace dataset is used for training. It turns out from the experiments that the hyperparameters $\alpha = 1$ while β and γ as trainable lead to better performance of PSNet. The proposed model is also tested with state-of-the-art losses including Angular-Softmax and ArcFace and has shown the improved performance. It is also observed that the proposed idea has a great impact over the relatively shallow model. It can lead to a better light weight CNN model for resource constrained devices.

ACKNOWLEDGMENT

This research is funded by Science and Engineering Research Board (SERB), Govt. of India under Early Career Research (ECR) scheme through SERB/ECR/2017/000082 project fund.

REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [5] S. R. Dubey, S. K. Roy, S. Chakraborty, S. Mukherjee, and B. B. Chaudhuri, "Local bit-plane decoded convolutional neural network features for biomedical image retrieval," *Neural Computing and Applications*, pp. 1–13, 2019.
- [6] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.
- [7] S. S. Basha, S. Ghosh, K. K. Babu, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Rccnet: An efficient convolutional neural network for histological routine colon cancer nuclei classification," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 1222–1227.
- [8] C. Nagpal and S. R. Dubey, "A performance evaluation of convolutional neural networks for face anti spoofing," *arXiv preprint arXiv:1805.04176*, 2018.
- [9] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 389–398.
- [10] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.
- [11] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in neural information processing systems*, 2014, pp. 1988–1996.
- [12] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.
- [13] E. Zangeneh, M. Rahmati, and Y. Mohsenzadeh, "Low resolution face recognition using a two-branch deep convolutional neural network architecture," *Expert Systems with Applications*, vol. 139, p. 112854, 2020.
- [14] L. Wolf, T. Hassner, and I. Maoz, *Face recognition in unconstrained videos with matched background similarity*. IEEE, 2011.
- [15] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.
- [16] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.
- [17] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.
- [19] S. R. Dubey and S. Chakraborty, "Average biased relu based cnn descriptor for improved face retrieval," *arXiv preprint arXiv:1804.02051*, 2018.
- [20] S. K. Roy, S. Manna, S. R. Dubey, and B. B. Chaudhuri, "Lisht: Non-parametric linearly scaled hyperbolic tangent activation function for neural networks," *arXiv preprint arXiv:1901.05894*, 2019.
- [21] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *arXiv preprint arXiv:1801.07698*, 2018.
- [22] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.
- [23] Y. Srivastava, V. Murali, and S. R. Dubey, "Hard-mining loss based convolutional neural network for face recognition," *arXiv preprint arXiv:1908.09747*, 2019.
- [24] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *International conference on machine learning*, 2013, pp. 1058–1066.
- [25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.
- [27] Y. Srivastava, V. Murali, and S. R. Dubey, "A performance comparison of loss functions for deep face recognition," *arXiv preprint arXiv:1901.05903*, 2019.
- [28] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.