IOP Conf. Series: Journal of Physics: Conf. Series **1487** (2020) 012024 doi:10.1088/1742-6596/1487/1/012024

PickPatch: A Regularization Method for Deep Face Recognition

Linjun Sun [1,2,3.4], Wei He [3,4], Xin Ning [1,2,3,4,*,a], Weijun Li [1,2,*,b] and Yuan Shi [3,4]

[1] Laboratory of Artificial Neural Networks and High-speed Circuits, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China;
[2] Center of Materials Science and Optoelectronics Engineering & School of Microelectronics, University of Chinese Academy of Sciences, Beijing, China 100049;
[3] Cognitive Computing Technology Joint Laboratory, Wave Group, Beijing 100083;
[4] Beijing Wave Security Technology company limited, Beijing, 102208

[*]Corresponding author: [a]ningxin@semi.ac.cn; [b]wjli@semi.ac.cn

Abstract. This paper proposes a simple and efficient regularization method, called PickPatch, for face recognition based on a deep convolutional neural network (DCNN). The proposed method randomly selects patches in the input face image and the intermediate feature maps as the activation region according to facial landmarks during the training phase. PickPatch is an approximation method that trains a series of models for different face patches and provides a combined model. This strategy introduces the idea of model combination for multiple face patches but does not change the model structure, which is both simple and efficient. Experiments on the public LFW database demonstrate that the proposed regularization method based on current deep convolutional neural networks can achieve obvious improvements of face recognition accuracy.

1. Introduction

Face recognition (FR) technology has become part of daily life, and its applications have been extended into different environments including financial services, public security, government affairs, transportation, and retail services. Recently, significant improvements in computer vision technology, especially in face recognition, have been brought about by the development of deep convolutional neural networks (DCNNs). DeepID [1] regards FR as a face image classification task, and employs the conventional softmax classifier to train the DCNN. FaceNet [2] treats FR as a problem of the metric learning of face similarity, and introduces triplet loss as the supervisor. ArcFace [3] introduces the arc space margin to enhance inter-class differences and ultimately improve FR performance. Ning et al [4][5][6]solve the problem of face recognition by introducing bionic pattern recognition. Current prevailing methods have focused on the enhancement of the discriminative power of face features. Comparatively, little research has investigated regularization methods for FR.

One common generalization method is model combination, which is a method by which to increase the generalization ability for deep face recognition. Model combination [7] aims at improving the overall performance by fusing models that can be complementary to each other. However, training multiple and diverse models with optimal hyperparameters requires massive computing resources, as well as training and inference time. In this sense, dropout [8] is considered to be effective for obtaining the same performance as model combination while utilizing fewer resources. When training

networks with dropout, activations are set to zero with some fixed probability. The network then approximates the result of exponentially-sized ensembles of sub-networks. Existing studies have shown that dropout is powerful for fully-connected layers, but less powerful for convolutional layers[9]. DropBlock [10] claims that information flows can still be transmitted, even with dropout, because of the spatial correlation of the features in the convolutional layers.

In this paper, the PickPatch method is proposed; it was inspired by DropBlock [10], a regularization method professionally applied in face recognition. PickPatch solves the problem of overfitting by injecting noise in a similar fashion to dropout and DropBlock. However, unlike in these algorithms, contiguous patches that constitute a continuous region of input are randomly selected as the activation region according to the facial landmarks during the training phase. PickPatch is applied to the entire network structure. Experiments on the public LFW database demonstrate that the PickPatch model based on current deep convolutional neural networks can achieve obvious improvement of face recognition accuracy.

2. Related work

Deep convolutional neural networks with more data and regularization tend to achieve stronger hierarchical feature representation.

The most direct way to prevent the network from overfitting is data augmentation, which creates new samples to increase the amount and diversity of training data. In addition to simple image transformation, such as mirroring and random cropping, data augmentation has been addressed in many studies. Bengio et al. [11]proposed that perturbed examples effectively improve model robustness. Cutout [12] creates out-of-distribution examples by randomly dropping out a square region of input images; this makes the network more robust to noise and occlusion. Mixup [13] and SamplePairing [14] synthesize a new linear combination of two origin images at the pixel level. The difference between them is that the former also linearly combines the two labels, but the latter uses the label of the first image. The basic principle of this method is the application of noise to training datasets to prevent network overfitting. However, Smart Augmentation [15] utilizes the idea of generative adversarial neural networks [16], and uses a deep neural network to generate new samples by inputting some images from the same class.

In this work, noise is applied to data augmentation in a way similar to Cutout[12], but with two key differences. First, the square regions of input images are determined according to the location of the key points on the face instead of by random selection. Second, rather than dropping, PickPatch chooses selected face regions and sets the rest of the faces to zero.

From the perspective of the network structure and training process, another widely-used regularization method is to add noise to the training step. Dropout [8] provides effective and simple means by which to approximate the model ensemble by dropping out the subset of activations. DropConnect [17] generalizes dropout by dropping each connection with probability p. However, dropout is less powerful for convolutional layers, and structured noise is therefore needed. From this perspective, DropBlock [10] randomly drops out a square region in a feature map, and SpatialDropout [9] drops out an entire channel. Further, StochasticDepth [18] and DropPath [19] benefit from the efficient implicit union of subnetworks by dropping subsets of layers with an identity function and dropping paths in a multi-branched structure separately. The similarity of all these methods lies in the fact that the overfitting information flow can be disturbed with noise.

The proposed method was inspired by DropBlock [10]. Patches in the input image and the intermediate feature maps are chosen according to the facial landmarks during the training phase. In these experiments, five facial landmarks were used to complete face alignment so that the least number of picked patches is one, and the greatest number of picked patches is five, which corresponds to the entire face image. The position of the patch in the image and feature map is consistent, which adds noise not only to the data source, but also to the training steps.

3. PickPatch

In this paper, a regularization method that can be professionally applied to face recognition is proposed. The main motivation for this paper is that little research has investigated regularization methods for FR, which are always very important in large-scale, unconstrained face recognition tasks. As a combination of the cutout method [12] and DropBlock [10], PickPatch chooses patches in the input image and the intermediate feature maps, and sets the rest to zero during the training phase. The positions of patches are decided by facial landmarks; this remains consistent in the entire network and is scaled up in the intermediate feature maps. Figure 1 presents the diagrammatic sketch of PickPatch.

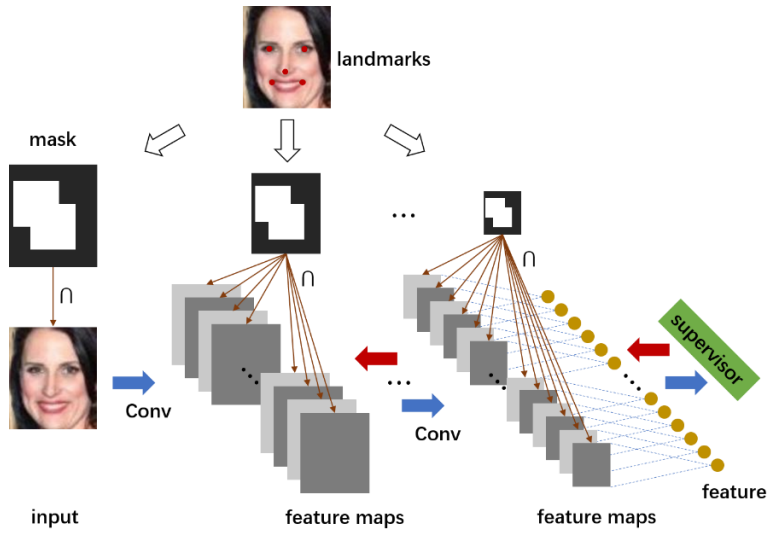

Figure 1. The diagrammatic sketch of PickPatch.

Figure 1 illustrates the whole process of the algorithm. First, the landmarks of all face images are detected by the multitask cascaded CNN algorithm [20], and the detected faces are aligned with similarity transformation according to their landmarks. Consequently, five landmarks are obtained for each image; thus, the number of picked patches can be at most five patches and at least one patch. A mask M is then sampled, in which the picked patches are set to either one or zero. Mask M is then applied onto the input image and the intermediate feature maps, namely

$$A = A \circ M, \tag{1}$$

where A represents the input images or feature maps and $\circ$ represents the Hadamard product. Similar to DropBlock [10] and dropout [8], PickPatch is not applied during inference. This approximates the result of exponentially-sized ensembles of sub-networks.

There are two main hyperparameters of the experiments, namely $patch\_radiu$ and num_layers. The $patch\_radiu$ hyperparameter is the radius of the patch which is obtained by expanding to the periphery with the coordinates as the center. The size of patches is scaled up by the size of the feature map. The num_layers hyperparameter is the number of masked layers in the intermediate feature maps. Their influences on the results are revealed by the experiments.

For face recognition, parts of face images are picked, which makes the network more adaptable to occluded examples. This regularization method also prevents the network weights from collaborating with one facial feature and yielding more powerful representation space. On the other hand, although the mask in the intermediate feature maps may have no real physical meaning, this is equivalent to injecting structured noise in a similar fashion to DropBlock [10]. Furthermore, the closer to the middle position in the feature map, the greater the influence on the output according to the convolution calculation mechanism. Therefore, the proposed method is superior to random selection.

4. Experiments

A series of experiments for FR were performed to verify the effectiveness of PickPatch. All

experiments were implemented with PyTorch[21].

4.1 Implementation Details

4.1.1. Dataset and Face Image Preprocessing. The training dataset was the subset of the publicly available VGGFace2[22]. After removing the identities appearing in LFW[23], the training set consisted of 8529 identities and approximately 3M images. After detected their landmarks, faces images were aligned with similarity transformation according to their landmarks. The size of aligned face images was 112×112, and each pixel was normalized by subtracting 128 and dividing by 128.

4.1.2. Network Architecture. In our experiments, MobileFaceNet [24] and ResNet [25] were employed as the network architectures. In the face recognition field, MobileFaceNet[24] is a representative of lightweight networks, and has extreme efficiency for real-time face verification on mobile devices. The main building block of MobileFaceNet is the residual bottleneck. The detailed architecture is consistent with that used in the original paper. ResNet [25] is also a classic network that is widely used in face recognition tasks. In the experiments, ResNet50 was employed to obtain the final 512-D embedding feature according to ArcFace[3].

4.2. Evaluation on LFW

The trained models were evaluated on the LFW dataset, which contains 13233 images from 5749 different identities that consist of large variations in pose, expression, and illumination. All the models were evaluated using two different protocols; the first was the standard unrestricted protocol with labeled outside data, and the second was the benchmark of large-scale unconstrained face recognition (BLUFR) protocol that used all 13233 images.

Two basic networks, namely MobileFaceNet [24] and ResNet[25], were used to verify the performance of PickPatch. For every batch, num_layers layers was randomly selected, and selected feature maps were masked after the last operation of the residual bottlenecks or residual blocks with some fixed probability $prob$, which was set at 0.5. Fifty images for each identity from VGGFace2[22] were randomly selected to constitute the final training data. Except for PickPatch, no other regularization means, such as random cropping or horizontal flipping, were used. The feature scale s was set to 30 and the angular margin m of AMsoftmax[26] was chosen as 0.4. As determined by experimental verification, when the radius of the patch was set to 30, the proposed method was found to achieve better performance. Table 1 presents the test results of MobileFaceNet and ResNet

Table 1. Test results of PickPatch on MobileFaceNet and ResNet

Architecture	PickPatch Yes/No	num_layers	LFW 6000 pairs	LFW BLUFR VR@FAR=0.1%	LFW BLUFR DIR@FAR=1%
MobileFaceNet (VGGFace2 subset)	No	-	98.58%±0.46%	98.20%±0.59%	85.86%±1.19%
	Yes	0	98.77%±0.63%	98.52%±0.67%	87.39%±1.37%
	Yes	3	**99.03%±0.50%**	98.60%±0.62%	**88.31%±1.38%**
	Yes	7	98.85%±0.56%	**98.66%±0.55%**	88.00%±1.79%
ResNet50 (VGGFace2 subset)	No	-	98.97%±0.46%	97.95%±0.79%	80.18%±2.09%
	Yes	0	98.80%±0.71%	98.45%±0.64%	84.65%±1.66%
	Yes	3	98.97%±0.59%	98.52%±0.61%	85.40%±2.41%
	Yes	9	**99.10%±0.50%**	98.46%±0.63%	85.99%±1.60%
	Yes	12	98.80%±0.46%	**98.54%±0.50%**	**86.03%±1.46%**
ResNet50 (VGGFace2)	No	-	99.52%±0.34%	99.54%±0.21%	92.04%±1.62%
	Yes	3	99.53%±0.34%	99.67%±0.22%	93.06%±1.60%
	Yes	9	**99.65%**±0.34%	99.72%±0.19%	94.64%±1.16%
	Yes	12	99.60%±0.34%	**99.73%±0.17%**	**94.92%±0.94%**

As reported in Table 1, the use of PickPatch in MobileFaceNet and ResNet trained by the VGGFace2 subset outperformed the baselines by a significant margin on the LFW dataset, which demonstrates that PickPatch can notably enhance the discriminative power of embedding features. In addition to the baseline, the models were investigated with different num_layers. The experimental results reveal that MobileFaceNet with $num\_layers = 3$ and ResNet50 with $num\_layers = 9$ achieved the best recognition performance; thus, these settings were applied in the subsequent experiment. The performance of the model trained with $num\_layers = 0$ was also verified, and in this case, PickPatch was only applied to the input layer. The result was between that of the baseline and that of the proposed method, demonstrating the superiority of employing PickPatch in the intermediate feature maps. In the experiments, the layers in the net and the feature maps in a layer were all randomly selected; thus, masking in the intermediate feature maps not only injected structured noise, leading to good generalization ability, but also approximated the model ensemble in a doubly random manner.

ResNet50, which has massive parameters, is difficult to converge with such a small dataset, and easily experiences overfitting. The proposed PickPatch alleviates this problem to some extent, but it is still limited by a small dataset. Based on this consideration, the results of ResNet50 trained with the entire VGGFace2 are also presented in Table 1. With large-scale face images and a large number of parameters, the proposed method also exhibited advantages and outperformed the baseline by an obvious margin.

5. Conclusion

In this work, PickPatch was introduced for professional application to face recognition. PickPatch injects noise in the data source and training step by randomly selecting patches in the input and the intermediate feature maps as the activation region according to the facial landmarks. The randomness in the PickPatch approximation model results in the training of a series of models for different face patches and different network architectures. PickPatch was demonstrated to be an effective regularization method for face recognition by employing it in the MobileFaceNet and ResNet50 architectures.

This work has demonstrated the superiority of PickPatch by randomly dropping a part of spatial features. Future work will return to more automatic work, such as automatically defining the position of picking or dropping.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61901436).

References:

[1] Y. Sun, X. Wang, and X. Tang, “Deep learning face representation from predicting 10,000 classes,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1891–1898

[2] F. Schroff, D. Kalenichenko, and J. Philistine, “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

[3] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “ArcFace: Additive Angular Margin Loss for Deep Face Recognition,” arXiv:1801.07698 [cs], Jan. 2018

[4] Xin N, Weijun L, Haoguang L, et al. Uncorrelated Locality Preserving Discriminant Analysis Based on Bionics[J]. Journal of Computer Research and Development, 2016 (11): 18.

[5] Ning X, Li W, Tang B, et al. BULDP: biomimetic uncorrelated locality discriminant projection for feature extraction in face recognition[J]. IEEE Transactions on Image Processing, 2018, 27(5): 2575-2586.

[6] Ning X, Li W, Xu J. The principle of homology continuity and geometrical covering learning for pattern recognition[J]. International Journal of Pattern Recognition and Artificial

Intelligence, 2018, 32(12): 1850042.
[7] Z. H. Zhou, Ensemble Methods: Foundations and Algorithms, London:Chapman and Hall/CRC, 2012
[8] 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,15(1):1929–1958, 2014
[9] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient Object Localization Using Convolutional Networks," arXiv:1411.4280 [cs], Jun. 2015
[10] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "DropBlock: A regularization method for convolutional networks," arXiv:1810.12890 [cs], Oct. 2018
[11] Y. Bengio, A. Bergeron, N. Boulanger-Lewandowski, T. Breuel, Y. Chherawala, et al. Deep learners benefit more from out-of-distribution examples. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pages 164–172, 2011
[12] T. DeVries and G. W. Taylor, "Improved Regularization of Convolutional Neural Networks with Cutout," arXiv:1708.04552 [cs], Nov. 2017
[13] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," arXiv:1710.09412 [cs, stat], Apr. 2018
[14] H. Inoue, "Data Augmentation by Pairing Samples for Images Classification," arXiv:1801.02929 [cs, stat], Apr. 2018
[15] J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart Augmentation Learning an Optimal Data Augmentation Strategy," IEEE Access, vol. 5, pp. 5858–5869, 2017
[16] I. Goodfellow et al., "Generative Adversarial Nets," in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680
[17] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, "Regularization of Neural Networks using DropConnect," In International Conference on Machine Learning, pages 1058–1066, 2013
[18] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Weinberger, "Deep Networks with Stochastic Depth," In ECCV, pages 646–661. Springer, 2016
[19] G. Larsson, M. Maire, and G. Shakhnarovich, "FractalNet: Ultra-Deep Neural Networks without Residuals," arXiv:1605.07648 [cs], May 2017
[20] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," IEEE Signal Process. Lett., vol. 23, no. 10, pp. 1499–1503, Oct. 2016
[21] Paszke A, Gross S, Massa F, et al. PyTorch: An imperative style, high-performance deep learning library[C]//Advances in Neural Information Processing Systems. 2019: 8024-8035
[22] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," arXiv:1710.08092 [cs], May 2018
[23] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," p. 14, 2008
[24] S. Chen, Y. Liu, X. Gao, and Z. Han, "MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices," in Biometric Recognition, vol. 10996, J. Zhou, Y. Wang, Z. Sun, Z. Jia, J. Feng, S. Shan, K. Ubul, and Z. Guo, Eds. Cham: Springer International Publishing, 2018, pp. 428–438
[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778
[26] F. Wang, W. Liu, H. Liu, and J. Cheng, "Additive Margin Softmax for Face Verification," IEEE Signal Process. Lett., vol. 25, no. 7, pp. 926–930, Jul. 2018