




A Visual Inductive Priors Framework for Data-Efficient Image Classification

Pengfei Sun^(✉) , Xuan Jin, Wei Su, Yuan He, Hui Xue, and Quan Lu

Alibaba Group, Hangzhou, China
{yeqing.spf, jinxuan.jx, junyu.sw, heyuan.hy,
hui.xueh, luquan.lq}@alibaba-inc.com

Abstract. State-of-the-art classifiers rely heavily on large-scale datasets, such as ImageNet, JFT-300M, MSCOCO, Open Images, etc. Besides, the performance may decrease significantly because of insufficient learning on a handful of samples. We present Visual Inductive Priors Framework (VIPF), a framework that can learn classifiers from scratch. VIPF can maximize the effectiveness of limited data. In this work, we propose a novel neural network architecture: DSK-net, which is very effective in training from small data sets. With more discriminative feature extracted from DSK-net, overfitting of network is alleviated. Furthermore, a loss function based on positive class as well as an induced hierarchy are also applied to further improve the VIPF's capability of learning from scratch. Finally, we won the **1st Place** in VIPriors image classification competition.

Keywords: Learn from scratch · Classification · Visual inductive priors · DSK-net · Induced hierarchy

1 Introduction

Convolutional Neural Networks (CNNs) have achieved state-of-the-art performance in image classification, object detection, semantic segmentation, etc. With the appearance of AlexNet [14], VGG [18], Inception [12, 19–21], ResNet [9], EfficientNet [22], ResNeSt [29], etc., the top-1 accuracy on ImageNet has been increased from 62.5% (AlexNet) to 84.5% (ResNeSt-269). Besides different network backbones, there are also many plug-and-play modules which can significantly improve accuracy, such as SE (Squeeze-and-Excitation) [11], CBAM (Convolutional Block Attention Module) [26], ECA (Efficient Channel Attention) [24], etc.

However, due to the limitation of label data, the performance of CNN is greatly limited. Pre-trained models are the most common solution that can get a fine result because of the prior knowledge. But there are only a few pre-trained models which are fixed architectures and proposed like Inception, ResNet, EfficientNet, etc. For training from scratch on VIPriors classification dataset which has only 50 training samples per class, the effectiveness of learning plays an

important role. Effective and sufficient augment strategies are necessary, such as rand erasing [32], Mixup [30], CutMix [28], Cutout [7], AutoAugment [5], RandAugment [6], etc. On the other hand, models would overfit easily with little training data, so it is crucial to lighten the overfitting with appropriate regularization.

In this work, a novel network architecture Dual Selective Kernel network (DSK-net) is proposed to improve the effectiveness on small scale datasets. For more data-efficient learning, positive class classification loss and intra-class compactness loss are applied to enhance discriminative power of the deeply learned features. An induced hierarchy is used which is easier for models to learn from scratch. Methods are evaluated on VIPriors Image Classification dataset. The dataset is derived from ImageNet and contains 50 images per class for training and testing. Experimental results show that our methods achieve the best performance on VIPriors classification dataset.

2 Related Works

2.1 Data Augmentation

Augmentation is an effective way to improve CNNs' performance especially in the case of insufficient data. Mixup [30] trains a model on convex combinations of pairs of examples and their labels together. Cutout [7] randomly erases square regions on input images during training. CutMix [28] cuts and pastes patches among training images where the training labels are also mixed proportionally to the area of patches. It can efficiently make use of training pixels and retain the regularization effect of regional dropout. GridMask [3] drops pixels on the input images with multiple squares and different ratios. Recently, with the emergence of AutoML, network learning strategies also can be searched from data. Auto-Augmentation [5] is a series of augmentation operation strategies searched on ImageNet which needs a huge space for searching. Hence RandAugmentation [6] proposes a simplified search space which has less computational expense.

2.2 Translation Invariance in CNNs

It is generally known that CNNs are not shift-invariant. A small shift or translation of input will result in a quite different output. To reduce the influence of translation, several augmentation operations are often used such as scaling, rotation and reflection [2, 4, 8, 17, 27]. [31] integrates low-pass filtering to anti-alias which is a common signal processing module. [13] proposes a full convolution architecture by removing spatial location as feature which improves equivariance and invariance of the inductive convolutional prior.

2.3 Important Feature Learning

For image classification, locating and recognizing the discriminative feature is the key to a better performance. And most of discriminative feature extraction modules are based on attention mechanisms which is inspired by human

brain neural units. SE (Squeeze-and-Excitation) [11] and ECA [24] are channel attention architectures. Channel and spatial attention modules are applied in CBAM [26]. Inspired by adaptive field sizes of neurons, [15] proposes Selective Kernel (SK) convolution which is based on soft-attention manner to improve feature extraction efficiency. Except attention architecture, loss function can also help model learn more discriminative feature. Center loss [25] is implemented by increasing inter-class dispersion and intra-class compactness. It learns centers form deep features of each class, and then penalizes the distances between deep features and their corresponding class centers.

3 Proposed Method

To be more data-efficient, firstly, a 3-branched network called Dual Selective Kernel (DSK) network is proposed in Fig. 1. DSK has the advantages of discriminative feature extraction, translation invariant and regularization. Secondly, a composite loss function is designed to improve feature discrimination. It helps models not only classify correctly but also increase the diversity of different classes.

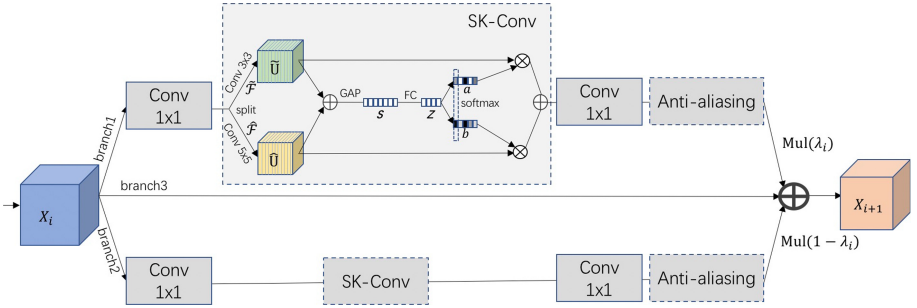


Fig. 1. Dual selective kernel residual block.

3.1 Dual Selective Kernel Network

Discriminative Feature Extraction. To adjust the receptive fields of neurons automatically, selective kernel convolution [15] is added into residual block. For any given feature map $X_i \in \mathbb{R}^{H \times W \times C}$, X_i is respectively conducted by convolutions of kernel size 3 and 5. Then two transforms are conducted: $\hat{\mathcal{F}}: X_i \rightarrow \hat{\mathcal{U}} \in \mathbb{R}^{H \times W \times C}$ and $\tilde{\mathcal{F}}: X_i \rightarrow \tilde{\mathcal{U}} \in \mathbb{R}^{H \times W \times C}$. Both $\hat{\mathcal{F}}$ and $\tilde{\mathcal{F}}$ are composed with depthwise convolution, Batch Normalization and ReLU. Feature \mathcal{U} is a element-wise sum of $\hat{\mathcal{U}}$ and $\tilde{\mathcal{U}}$. For \mathcal{U} , global average pooling is used for information embedding. Further, a compact feature $s \in \mathbb{R}^C$ is created by passing feature embedding to fully connected layer for squeeze. Then Batch Normalization, ReLU and another two fully connected layers are applied for different kernel

excitation. Finally, a soft attention is conducted to select information in different spatial scales. The weights $\hat{\omega}$ and $\tilde{\omega}$ for attention is calculated by a channel-wise softmax operation of per channel between a and b . The final feature map is obtained by applying attention weights to feature \hat{U} and \tilde{U} :

$$\mathcal{V} = \hat{\omega} \cdot \hat{U} + \tilde{\omega} \cdot \tilde{U} \quad (1)$$

Translation Invariant. The reducing spatial resolution operations in CNNs including max pooling, average pooling and strided convolution are harmful to shift-equivariance. Blur pool [31] is an anti-aliased architecture which is compatible with above architectures components. For example, max pooling with stride=2 in CNNs will be split into max pooling with stride=1 and blur pool with stride=2. Strided convolution with activation function will be split into convolution with stride=1, activation function and blur pool. As for blur pool kernel, it has several anti-aliasing filters from size 2×2 to 5×5 with increasing smoothing. In DSK, 3×3 filter is applied in max pooling and strided convolution.

Regularization. Like data augmentation techniques applied to input data, it is reasonable to apply corresponding techniques to representation branch in residual block. Let X_i denotes the input tensor of residual block i . \mathcal{W}_i^1 and \mathcal{W}_i^2 denote weights associated with the two residual units. \mathcal{F} denotes the residual function and X_{i+1} denotes the outputs from i . The 3-branch architecture can be represented as:

$$X_{i+1} = X_i + \lambda_i \mathcal{F}(X_i, \mathcal{W}_i^1) + (1 - \lambda_i) \mathcal{F}(X_i, \mathcal{W}_i^2) \quad (2)$$

When forward and backward during training, λ_i is a random value of 0 or 1, which means that only one of branch1 and branch2 will be randomly selected. And λ_i is 0.5 for inference, which means that half of each branch's output will be used for inference.

3.2 Loss Function

Categorical cross-entropy (CE) loss after softmax is widely used in multi-class classification. But for VIPriors classification dataset, CE is suboptimal. Because it forces models to only focus on training image and ignore the compactness of intra-class. In this section, several loss functions will be discussed and a combined loss is proposed as Eq. 3 for a better performance.

$$L = \alpha L_{PCL} + \beta L_{CL} + \gamma L_{TSL} \quad (3)$$

Positive Class Loss. CE loss is showed in Eq. 4. Let p represents the output of a model and l represents one-hot labels. CE not only directs model to classify the ground truth class correctly but also forces the prediction of other classes as low as possible.

$$L_{CE} = -\frac{1}{N} \sum (l * \log(p) + (1 - l) * \log(1 - p)) \quad (4)$$

But is it suitable to use a loss on a small dataset in which the number of classes is far greater than the number of samples per class? Additionally, [16] proves that there are many label errors in ImageNet including actual multi-label images but only labeled with single class label. We have reasons to believe that there is the same question on VIPriors classification dataset. Based on the above, making models only focus on ground truth label may be more beneficial during learning. Consequently, the positive class loss (PCL) is proposed as:

$$L_{PCL} = \frac{1}{N} \sum (-l * \log(p) + (1 - \cos(l, p))) \quad (5)$$

PCL has two parts: the former is from CE, the latter is cosine loss [1].

Center Loss. Although PCL can directly model for a better learning, it is easily overfitting with less data. Therefore, center loss (CL) [25] in Eq. 6 is used for more discriminative feature extraction. Let $x_i \in \mathbb{R}_d$ denote the i th deep feature belonging to the y_i th class. The y_i th class center of deep features $c_{y_i} \in \mathbb{R}_d$ is computed by averaging y_i th class features of the corresponding classes in each iteration.

$$L_{CL} = \frac{1}{2} \sum_i \|x_i - c_{y_i}\|_2^2 \quad (6)$$

Tree Supervision Loss. The semantic relations of classes in VIPriors can be induced as a hierarchical tree. Child nodes of the tree represent 1000 classes in the dataset and parent nodes represent superclasses such as animal, vehicle and etc. For every parent node, its child nodes often have some commonalities which is helpful for classification. Inspired by Neural-Backed Decision Trees (NBDT) [23], a hierarchical architecture is defined according to the semantic relationship based on 1000 classes. Tree supervision loss (TSL) is used for model training. Let $x \in \mathbb{R}_d$ denotes featurized sample, $w_{r \rightarrow i}$ denotes weights of the path from root nodes r to leaf node n_i . TSL can be represented as:

$$L_{TSL} = L_{CE}([\prod x * w_{r \rightarrow n_1}, \prod x * w_{r \rightarrow n_2}, \dots], l) \quad (7)$$

4 Experiments

4.1 Implementation Details

Following data augmentation methods are used in our models: random resize and crop, random horizontal flip and CutMix (with a probability of 0.5). All models are trained with 16 GPUs and 64 samples per CPU. In the training stage, warm up with initial lr of 0.0001 in 5 epochs, cosine learning rate [10] with initial lr of 0.1, dropout with probability of 0.2, weight decay of 0.0001 and label smooth are used for learning. For coefficients in Eq. 6, α , γ and β are set to 1, 0.0005 and 1. In early time of the competition, we trained model on training

set for methods attempt and verification. And in the final stage, we trained models on both training set and most of validation set. Only a little samples in validation set were reserved for validation. For final prediction, Test Time Augmentation (TTA) with 10-crop was used. Additionally, experimental results prove that increasing training epochs from 90 to 360 improve model accuracy by 5.3%.

4.2 Results

Table 1 shows the results for ResNeXt, D-ResNeXt, SK-ResNeXt, DSK-ResNeXt, PSL and CL on validation set. Models are trained with 360 epochs.

Table 1. Performance of DSK-net, PSL and CL on validation set.

	top-1 acc. (%)
ResNeXt50_32x4d	52.01
D-ResNeXt50_32x4d	54.45
SK-ResNeXt50_32x4d without anti-aliasing	54.06
SK-ResNeXt50_32x4d	54.37
DSK-ResNeXt50_32x4d	55.97
DSK-ResNeXt50_32x4d+PSL	56.48
DSK-ResNeXt50_32x4d+PSL+CL	57.51

Table 2 shows results of TSL for EfficientNet and ResNeSt in the final stage. Models are trained with 720 epochs and tested on partial validation set.

Table 2. The experiment results of TSL.

	top-1 acc. (%)
EfficientNet-b3	62.42
EfficientNet-b3+TSL	63.15
EfficientNet-b5	65.43
EfficientNet-b5+TSL	65.85
EfficientNet-b6	65.67
EfficientNet-b6+TSL	66.26
ResNeSt-101(320x320)	65.96
ResNeSt-101(320x320)+TSL	67.15
ResNeSt-200(320x320)	67.40
ResNeSt-200(320x320)+TSL	67.81

Table 3 shows the results of DSK-net in the final stage. Models are trained with 540 epochs and tested on partial validation set. 69.59% is the best single model performance we achieved.

Table 3. The experiment results of DSK-net.

	top-1 acc. (%)
DSK-ResNeXt50_32x4d(224x224)	67.35
DSK-ResNeXt50_32x4d(320x320)	69.20
DSK-ResNeXt101_32x4d(224x224)	68.02
DSK-ResNeXt101_32x4d(320x320)	69.59

4.3 Other Tricks

Results for CutMix showed in Table 4 indicate that the global semantic information and local area feature are equally import.

Table 4. The experiment results on validation set for CutMix. Input size is 320×320 , training epoch is 90.

	top-1 acc. (%)
ResNeXt50_32x4d	45.23
ResNeXt50_32x4d+CutMix with prob=0.3	45.73
ResNeXt50_32x4d+CutMix with prob=0.5	46.25
ResNeXt50_32x4d+CutMix with prob=0.7	45.85
ResNeXt50_32x4d+CutMix with prob=1.0	44.35

Results of label smooth, dropout and dual pool are showed in Table 5:

Table 5. The experiment results of label smooth, dropout and dual pool on validation set. Models are trained with 360 epochs.

	top-1 acc. (%)
ResNeXt50_32x4d	50.56
ResNeXt50_32x4d+dual pool	50.87
ResNeXt50_32x4d+label smooth	50.70
ResNeXt50_32x4d+dropout with prob=0.2	50.90
ResNeXt50_32x4d+dropout with prob=0.4	50.84
ResNeXt50_32x4d+dual pool+label smooth+dropout with prob=0.2	51.82

4.4 Ensembling

For a better performance, we ensembled predictions of above methods in total 16 models including EfficientNet-b5, EfficientNet-b6, ResNeSt-101, ResNest-200, DSK-ResNeXt50, DSK-ResNeXt101. Finally, a weighted score average method was used that the weight of higher performance models was 3, the rest was 1. Finally, we got the score of 73.08% on test set.

Figure 2 shows an overview of methods and appearances. No external image/video data or pre-trained models were used throughout the competition.

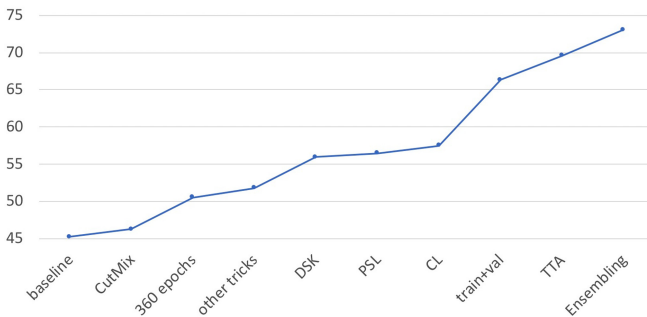


Fig. 2. Performance overview.

5 Conclusions

In this paper, we discuss and explore data-efficient learning, visual inductive priors and training from scratch. In VIPF, we propose a novel architecture called DSK-net, which is robust to translation. Sufficient experiment results fully proved that DSK-net learns efficiently from insufficient data and outperformed EfficientNet, ResNeSt on VIPriors classification dataset. Then a loss based on positive class is applied for model constraint. An induced hierarchy is used which can direct models to learn discriminatively and easily. Experimental results show that VIPF we proposed is effective. Finally we won the 1st place in VIPriors image classification competition.

References

1. Barz, B., Denzler, J.: Deep learning on small datasets without pre-training using cosine loss. In: The IEEE Winter Conference on Applications of Computer Vision, pp. 1371–1380 (2020)
2. Bruna, J., Mallat, S.: Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(8), 1872–1886 (2013)
3. Chen, P.: Gridmask data augmentation. arXiv preprint [arXiv:2001.04086](https://arxiv.org/abs/2001.04086) (2020)

4. Cohen, T., Welling, M.: Group equivariant convolutional networks. In: International Conference on Machine Learning, pp. 2990–2999 (2016)
5. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: AutoAugment: learning augmentation policies from data. arXiv preprint [arXiv:1805.09501](https://arxiv.org/abs/1805.09501) (2018)
6. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: RandAugment: practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 702–703 (2020)
7. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint [arXiv:1708.04552](https://arxiv.org/abs/1708.04552) (2017)
8. Esteves, C., Allen-Blanchette, C., Zhou, X., Daniilidis, K.: Polar transformer networks. arXiv preprint [arXiv:1709.01889](https://arxiv.org/abs/1709.01889) (2017)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
10. He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., Li, M.: Bag of tricks for image classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 558–567 (2019)
11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
12. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint [arXiv:1502.03167](https://arxiv.org/abs/1502.03167) (2015)
13. Kayhan, O.S., Gemert, J.C.v.: On translation invariance in CNNs: convolutional layers can exploit absolute spatial location. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14274–14285 (2020)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
15. Li, X., Wang, W., Hu, X., Yang, J.: Selective kernel networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 510–519 (2019)
16. Northcutt, C.G., Jiang, L., Chuang, I.L.: Confident learning: estimating uncertainty in dataset labels. arXiv preprint [arXiv:1911.00068](https://arxiv.org/abs/1911.00068) (2019)
17. Sifre, L., Mallat, S.: Rotation, scaling and deformation invariant scattering for texture discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1233–1240 (2013)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
19. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
20. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
21. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
22. Tan, M., Le, Q.V.: EfficientNet: rethinking model scaling for convolutional neural networks. arXiv preprint [arXiv:1905.11946](https://arxiv.org/abs/1905.11946) (2019)
23. Wan, A., et al.: NBDT: neural-backed decision trees. arXiv preprint [arXiv:2004.00221](https://arxiv.org/abs/2004.00221) (2020)

24. Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.: ECA-Net: efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11534–11542 (2020)
25. Wen, Y., Zhang, K., Li, Z., Qiao, Yu.: A discriminative feature learning approach for deep face recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 499–515. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_31
26. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_1
27. Worrall, D.E., Garbin, S.J., Turmukhambetov, D., Brostow, G.J.: Harmonic networks: deep translation and rotation equivariance. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5028–5037 (2017)
28. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: CutMix: regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 6023–6032 (2019)
29. Zhang, H., et al.: ResNeSt: split-attention networks. arXiv preprint [arXiv:2004.08955](https://arxiv.org/abs/2004.08955) (2020)
30. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: beyond empirical risk minimization. arXiv preprint [arXiv:1710.09412](https://arxiv.org/abs/1710.09412) (2017)
31. Zhang, R.: Making convolutional networks shift-invariant again. arXiv preprint [arXiv:1904.11486](https://arxiv.org/abs/1904.11486) (2019)
32. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: AAAI, pp. 13001–13008 (2020)