Confusion-Aware Convolutional Neural Network for Image Classification

Liguang Yan[1], Baojiang Zhong[1(✉)], and Kai-Kuang Ma[2]

[1] School of Computer Science and Technology, Soochow University, Suzhou, China
lgyan@stu.suda.edu.cn, bjzhong@suda.edu.cn
[2] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore
ekkma@ntu.edu.sg

Abstract. In image classification, it is often encountered that the decision boundaries of some image categories are ambiguous and easy to confuse with each other, thus yielding inferior accuracy on image classification. In this paper, a novel *confusion-aware* convolutional neural network (CNN) is proposed to address this issue. Different from the *coarse-to-fine* strategy that has been practiced in existing hierarchical classifiers, our proposed method performs *predict-then-correct* strategy. At the training stage, a conventional classifier (referred to as the *prediction* classifier) is trained, and its confusion matrix is estimated by exploiting a cross validation process conducted on the training set. Based on this estimated confusion matrix, a *confusion-aware model* is then established, and it is used as a decision maker to train a set of *correction* classifiers for those confusing categories. At the classifying stage, the prediction and correction classifiers collaboratively work together via a hierarchical structure, and the confusion-aware model is used again as a decision maker to select a proper prediction classifier for each confusing category. Experimental results conducted on the Mnist and CIFAR-10 datasets show that the proposed confusion-aware network outperforms the existing CNN classifiers on image classification.

Keywords: Image classification · Convolutional neural networks · Confusion matrix · Cross-validation · Confusion-aware model

1 Introduction

Most of the existing convolutional neural network (CNN) classifiers for conducting image classification task have a 'flat' structure [1], which treat all classes as independent ones and ignore their visual separability. However, some categories could be substantially more difficult to be differentiated than others and hence more sophisticated classifiers are needed. For example, in the CIFAR-10 dataset [2] it is easy to distinguish a 'cat' from a 'truck', but could be very difficult to distinguish a 'cat' from a 'dog' due to an ambiguous decision boundary between this two categories. A CNN classifier, LeNet-5 [12], can achieve an accuracy of

T. Gedeon et al. (Eds.): ICONIP 2019, LNCS 11953, pp. 151–161, 2019.
https://doi.org/10.1007/978-3-030-36708-4_13

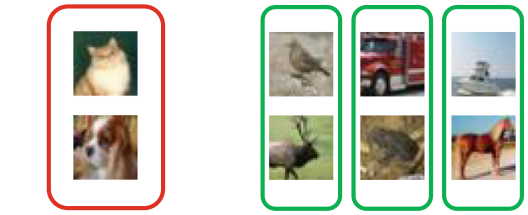

Fig. 1. An example of confusing categories. By using the LeNet-5 [12] method to the dataset, the ratio of misclassifications between 'cats' and 'dogs' categories reaches 16.65%, while the highest ratio of misclassifications between any other two categories is only 6.35%.

76.92% on this dataset; however, the ratio of misclassifications between 'cats' and 'dogs' reaches 16.65%, which is much higher than that of any other two categories with less ambiguous decision boundary or confusion (as demonstrated in Fig. 1).

In this paper, a *confusion-aware* CNN is proposed with incorporation of a confusion-aware model in our developed *predictor-corrector* hierarchical framework. The proposed method is much more capable on distinguishing those image categories that have ambiguous decision boundaries, and it comprises two stages as follows. First, we train a conventional CNN (called the *prediction* classifier) using a training set to conduct a cross validation on the set for computing its confusion matrix. Based on this estimated confusion matrix, a confusion-aware model is then established. Second, by using the confusion-aware model as a decision-making system, a set of *correction* classifiers are trained for yielding a more discriminated decision boundary for each pair of ambiguous categories. Finally, the prediction and correction classifiers as obtained above are collaboratively used via a hierarchical structure for delivering much improved image classification.

The remainder of the paper is organized as follows. In Sect. 2, related works are briefly reviewed, and the relationship between our proposed method and the existing ones are clarified. In Sect. 3, the proposed confusion-aware CNN is described in detail. In Sect. 4, we present the experimental results of the confusion-aware CNN on the Mnist and CIFAR-10 datasets. Finally, Sect. 5 concludes this paper.

2 Backgrounds

2.1 Convolutional Neural Networks

Many CNN-based algorithms have shown their capability on delivering state-of-the-art performance in various computer-vision tasks, including image classification [4,17], object detection [7,16], semantic segmentation [8,15], and so

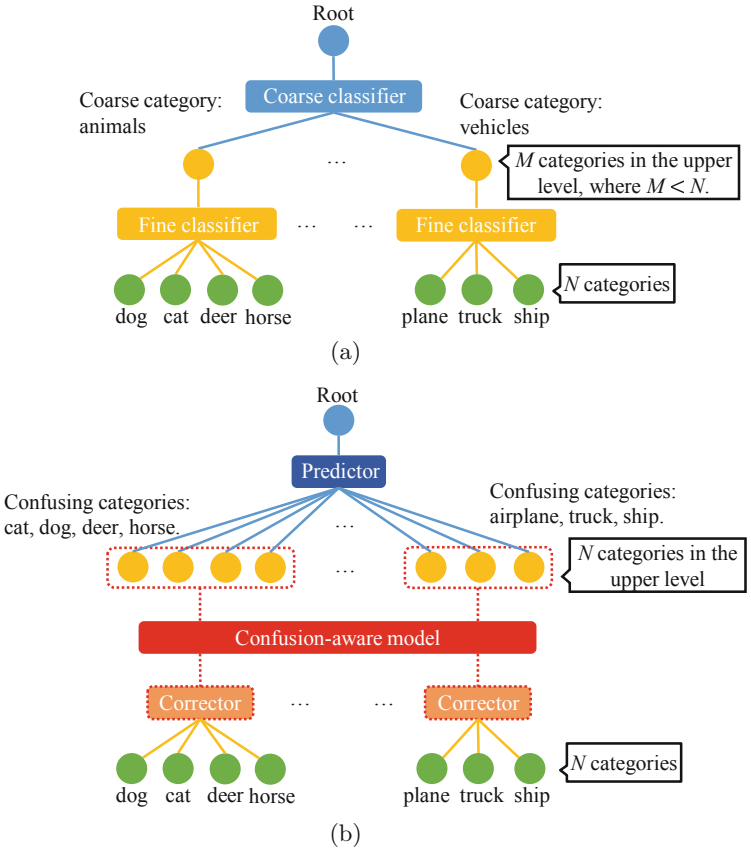

Fig. 2. A comparison of the architectures of the conventional hierarchical classification as shown in (a) and our proposed confusion-aware CNN as presented in (b).

on. Previous investigations mainly focus on enhancing the CNN's components, such as its pooling layers [5], nonlinear layers [21], or activation units [9]. In this work, there is no attempt to modify any of these components. Instead, a new and generalized CNN architecture (i.e., the *predictor-corrector* hierarchical framework) is proposed, in which any CNN classifier can be used. In other words, any superior CNN classifier developed in the future can be simply substituted into our proposed framework for yielding better performance.

2.2 Hierarchical Classification

For image classification, most of the existing deep CNNs are trained as an N-way 'flat' classifier. Since certain hierarchical relationships might be existing among some categories, *hierarchical* classification has been considered as an effective approach for conducting large-scaled visual recognition task [10,11,20,22].

It solves the classification problem by embedding classifiers into two or more category hierarchies, as demonstrated in Fig. 2. The upper-level classifiers produce coarse classification results, which are further discriminated by the lower-level classifiers. The hierarchy of such classifications can be predefined [10,20] or learned by a top-down (or bottom-up) method [11,22].

An earlier work of a category hierarchy using the CNN is reported in [18]. It is mainly used to improve the results for the categories with insufficient training examples by transfer learning. Later on, a hierarchical CNN, called the Hierarchical Deep CNN (HD-CNN), is proposed in [6], for which a set of CNN models based on a two-level category hierarchy are trained to achieve superior classification results over the standard CNN. In [19], a method is developed for regularization and model selection that simultaneously learns both the hierarchical architecture and model parameters. Indeed, hierarchical classification improves the accuracy of classification. However, it faces the problem of error propagation [1]; that is, the classification errors yielded from the upper-level classifiers will be propagated to the classifiers in the next level and therefore lead to more classification errors.

2.3 Present Work

There is a significant difference between the architecture of our proposed confusion-aware CNN and the conventional hierarchical classification as presented in Fig. 2. To be precise, our confusion-aware CNN does *not* follow the coarse-to-fine strategy of the existing hierarchical architecture [6,17,19]. It instead adopts a prediction-correction strategy to conduct a hierarchical classification, which is motivated by the predictor-corrector numerical approach that has been exploited to solve various mathematical and engineering problems (e.g., [23–25]). In other words, the fundamental difference between existing CNNs and our proposed lies in *coarse-to-fine* versus *predict-then-correct*.

3 Confusion-Aware Convolutional Neural Network

3.1 Outline

Components of our proposed confusion-aware CNN include one prediction classifier and a set of correction classifiers. A confusion-aware model is generated with the estimated confusion matrix of the prediction classifier, and the outputs of the prediction classifier and correction classifier are integrated by a probabilistic averaging layer, as demonstrated in Fig. 3 and described in detail as follows.

3.2 Prediction Classifier

The prediction classifier is a 'flat' classifier trained on the training set with all categories. At the classifying stage, it is employed to produce a prediction category of the input image. This prediction is usually not accurate enough

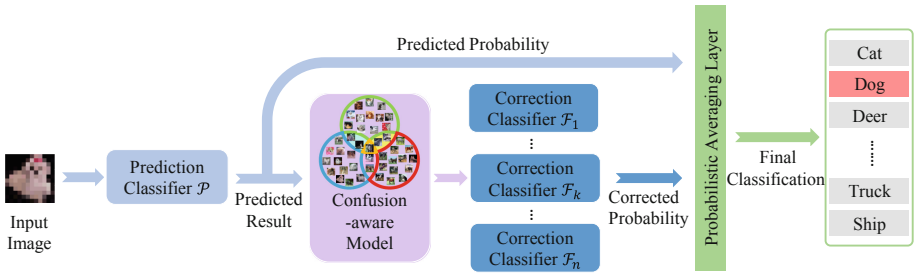

Fig. 3. A full picture of the proposed confusion-aware convolutional neural networks.

and hence need to be corrected. Let a_{ij} be the number of misclassified images between the ith and the jth categories; that is, the number of those images that belong to the ith category but have been misclassified into the jth category by using the predictor. Further define r_{ij} as the ratio of a_{ij} to the total number of misclassified images over every pair of categories; i.e.,

$$r_{ij} = \frac{a_{ij}}{\sum_{i \neq j} a_{ij}}, \qquad \text{for } i, j = 1, ..., K. \tag{1}$$

Based on the set of $\{r_{ij}\}$, those more confusing categories can be recognized from the others by simply applying a threshold to $\{r_{ij}\}$ which will be exploited as a prior to establish our confusion-aware model.

3.3 Confusion-Aware Model

Our confusion-aware model plays an important role in the proposed method. It serves as a decision maker both in the training stage (for training a set of correction classifiers) and in the classifying stage (for selecting a proper correction classifier). The confusion-aware model is constructed on the estimated confusion matrix of the prediction classifier, which is a specific table layout that represents the distribution of those easily confused categories. In detail, each column of the confusion matrix marks the instances in a predicted category while each row marks the instances in an actual category. The confusion matrix F is defined as:

$$F = (a_{ij}), \qquad \text{for } i, j = 1, ..., K. \tag{2}$$

A cross-validation process is used to estimate the confusion matrix of the prediction classifier. For that, we first divide the training set into N clusters and thus apply an N-fold cross-validation process to perform classification. In detail, $N-1$ clusters of the training set are selected in turns to train the model, followed by testing the trained model on the remaining cluster to yield a misclassification result. Then, after every cluster has been selected for testing, all the misclassification results are integrated to produce the confusion matrix (as demonstrated in Fig. 4). Compared with the existing validation approach that

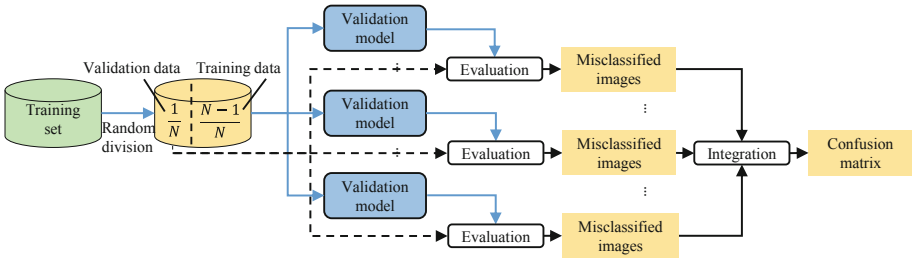

Fig. 4. A N-fold cross-validation step is adopted to estimate the confusion matrix of the prediction classifier.

only uses *one* randomly-sampled image cluster to generate the confusion matrix [6], our cross validation can maximize the use of the training set and obtain a more reliable confusion matrix. The prediction classifier could yield many confusing categories with ambiguous decision boundaries. As shown in Fig. 5, images in the overlapping areas of adjacent categories are easily misclassified. With the estimated confusion matrix, those easily-confusing categories can be recognized. To be specific, a proper threshold T which is less than 30% of the elements in the confusion matrix is used to select the top 30% ambiguous categories based on a ranking of a_{ij}, and then a set of correction classifiers can be trained to generate their clear decision boundaries.

3.4 Correction Classifiers

The confusion-aware model is established as a decision-making system to guide the training of correction classifiers. To be precise, for a predicted category (say, the kth), if $a_{ik} \geq T$ and $a_{jk} \geq T$ in the confusion matrix, a correction classifier $\mathcal{F}_k$ with the ith, jth, and the kth categories (a union of these three categories is denoted as $\mathbb{C}_k$) is trained. At the stage of classifying, if the prediction classifier $\mathcal{P}$ classifies the query image into the kth category, then the correction classifier $\mathcal{F}_k$ will again be selected by the confusion-aware model to conduct a correction classification. The final classification result is generated with a probabilistic averaging layer, where the results of the prediction classifier and the selected correction classifier will be integrated.

3.5 Probabilistic Averaging Layer

As mentioned in Sect. 2.2, error propagation could occur if the classification result is produced by using one classifier *only*. To overcome the difficulty, a probabilistic averaging layer is used to integrate the results of the prediction and the correction classifiers. For that, individual output of each classifier should be normalized with a softmax function, which is defined as

$$\sigma(z_j) = \frac{\exp(z_j)}{\sum_{i=1}^{K} \exp(z_i)}, \qquad \text{for } j = 1, ..., K, \tag{3}$$

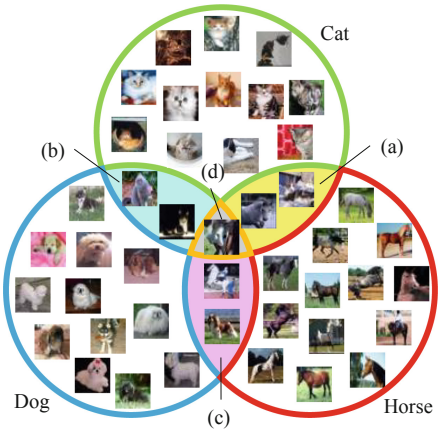

Fig. 5. An exemplary demonstration of the confusion yielded by the prediction classifier among three image classes—*Cat*, *Dog*, and *Horse*; upon which, our confusion-aware model is established. The images commonly shared by adjacent classes denote the confusion yielded between: (a) *Cat* and *Horse*; (b) *Cat* and *Dog*; (c) *Dog* and *Horse*; and (d) *Cat*, *Dog*, and *Horse*.

where z is a K-dimensional vector and its j-th element z_j is mapped onto the interval $(0, 1)$ as a probability $\sigma(z_j)$. Then, the probabilistic averaging layer is applied to average the two probabilities with different dimensions; that is,

$$p(y = j|X) = \begin{cases} \frac{1}{2}(B_j + p_c^k(y = j|X)), & j \in \mathbb{C}_k; \\ B_j, & j \notin \mathbb{C}_k, \end{cases} \tag{4}$$

where X is the input image and y is its category label. The probabilities predicted by the prediction classifier $\mathcal{P}$, and the correction classifier for category j are denoted as B_j and $p_c^k(y = j|X)$, respectively. Note that the $\mathcal{F}_k$ is trained on the union category $\mathbb{C}_k$. The category with the highest probability in p is the final classification of confusion-aware CNN.

4 Experiments

The proposed confusion-aware convolutional neural network is evaluated on the Mnist [12] and the CIFAR-10 [2] datasets. Experiments are implemented in PyTorch [13] and on a single NVIDIA Titan X card. The network is trained with back propagation [4]. The Mnist is a handwritten digital image dataset, with a size of 28×28 in each image. There are 10 categories, corresponding to numbers 0 to 9, respectively. In total, it contains 70,000 gray-scaled images of handwritten digits, of which 60,000 are used as the training data and the remaining 10,000 are the test data. The CIFAR-10 is a commonly-used computer vision dataset, containing a total of 60,000 images, with a size of 32×32 in each image.

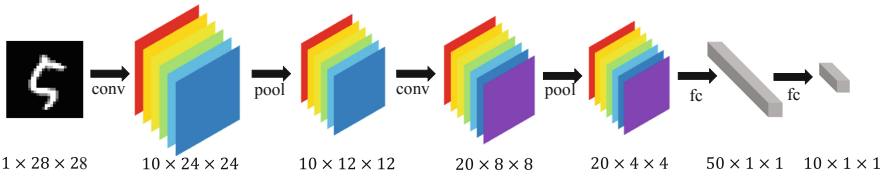

Fig. 6. Structure of the Mnist ConvNets [13].

It also has 10 categories. Among them, 50,000 images are used as the training set and the rest as the test set.

4.1 The Mnist Dataset

The Mnist ConvNets [13] is employed as the base model in our confusion-aware CNN framework; that is, it is exploited as the prediction classifier and also used to construct the correction classifiers. The Mnist ConvNets consists of two convolution layers and two fully-connected layers, as shown in Fig. 6. On the basis of the Mnist ConvNets, a confusion-aware CNN and a hierarchical CNN called HD-CNN [6] are trained. At the training stage of the confusion-aware CNN, the prediction classifier is iterated 500 epochs on the training set with learning rate 0.01 and momentum 0.9. The training set is divided into six sub-training sets and each set has 10,000 images. With these sub-training sets the six-fold cross-validation method is adopted to arrive at the confusion matrix. The correction classifiers are trained with confusing categories selected from the confusion matrix. The confusion-aware CNN is then tested on the test set, and the experimental results are shown in Table 1.

Table 1. Performance evaluation of different CNN models conducted on the Mnist dataset (without data augmentation).

Networks	Layers	Parameters	Accuracy
Mnist ConvNets [13]	4	22K	99.05%
ResNet-32 [3]	32	460K	99.18%
HD-CNN [6]	4	216K	99.21%
Confusion-Aware (Ours)	4	238K	99.31%

Experimental results as documented in Table 1 have shown that the accuracy of the Mnist ConvNets [13] is 99.05%. When one layer is replaced by our proposed confusion-aware CNN, the accuracy of the network is increased to 99.31%. The error rate of the confusion-aware CNN is about 25% lower than that of the single CNN. The number of parameters used in the confusion-aware CNN is 238,000, which are about 10 times more than that of Mnist ConvNets. Compared with

Fig. 7. Structure of the first two networks incorporated in our proposed method.

the ResNet-32 [3] (460,000 parameters) and the HD-CNN (216,000 parameters), the confusion-aware CNN is able to maintain high accuracy with a small amount of parameters, as documented in Table 1.

4.2 The CIFAR-10 Dataset

Three different CNNs are employed as the base model to further incorporate our developed confusion-aware model for evaluating the resulted performance. The depth and total parameters of these networks are increased gradually. The first network is structurally the same as that of LeNet-5 [12]. The only difference is that the size of the input image is changed from $28 \times 28 \times 1$ to $32 \times 32 \times 3$. The second network also has the same structure, however, the number of convolution kernels and the number of hidden nodes of the first network have been increased. The structure of the first two networks is shown in Fig. 7. Classifiers are trained with a learning rate of 0.1, which is decreased by a factor of 10 for every 100 epochs. These are iterated for 500 epochs over the training set with momentum 0.9 and weight decay 0.0005 [14]. The last one uses a residual network of 18 layers, called as the ResNet-18 [3], which includes 17 convolution layers and a fully-connected layer. Each classifier is iterated for 200 epochs. Initial learning rate is set to be 0.01 and is decreased by a factor of 10 for every 50 epochs. Randomly cropped and flipped strategies are used in the training.

Table 2. Performance comparison of three state-of-the-art CNNs (the second column) and the corresponding ones after incorporating our proposed confusion-aware model into these CNNs (the third column), respectively.

Networks	Conventional classifier	Confusion-aware CNN
LeNet-5 [12]	76.92%	80.99%
LeNet-5 (enhanced)	85.23%	86.97%
ResNet-18 [3]	94.63%	94.84%

As observed, and expected, from Table 2, the gain of accuracy decreases from 4.07% to 0.21% as the complexity of the base model increases. Therefore, it is considered that the simpler the base CNN is, the more gain of our confusion-aware CNN can achieve. For the exemplary experiment presented in Fig. 1, The ratio of the misclassified images between 'cats' and 'dogs' in our proposed confusion-aware CNN significantly drops, as shown in Fig. 8.

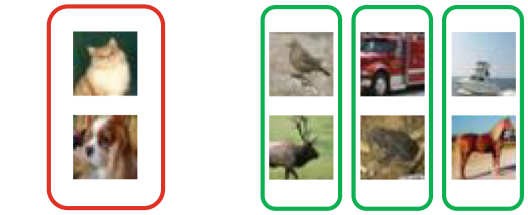

Fig. 8. The ratio of the misclassified images between 'cats' and 'dogs' shows a appreciable decrease with the use of confusion-aware CNN.

5 Conclusion

A novel CNN framework, called the confusion-aware CNN, is proposed in this paper and have clearly shown that it is able to improve the accuracy of image classification. By incorporating the confusion-aware model into a prediction-correction hierarchical structure, our proposed method is able to distinguish those image categories with ambiguous boundaries. Experiments conducted on the Mnist and CIFAR-10 datasets clearly show that the confusion-aware CNN can deliver the superior classification performance over the existing CNN models.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (NSFC No. 61572341) and a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

1. Silla, N., Freitas, A.: A survey of hierarchical classification across different application domains. Data Min. Knowl. Disc. **22**(1), 31–72 (2011)
2. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Technical report (2011)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
4. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
5. Zeiler, M.D., Fergus, R.: Stochastic pooling for regularization of deep convolutional neural networks. In: Proceedings of the International Conference on Learning Representation (2018)
6. Yan, Z., et al.: HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2740–2748 (2015)
7. Dai, D., Li, Y., He, K.M., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems, pp. 379–387 (2016)

8. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
9. Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. In: Proceedings of International Conference on Machine Learning, pp. 1319–1327 (2013)
10. Marszalek, M., Schmid, C.: Semantic hierarchies for visual object recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–7 (2007)
11. Salakhutdinov, R., Torralba, A., Tenenbaum, J.: Learning to share visual appearance for multiclass object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1481–1488 (2011)
12. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
13. Paszke, A., et al.: Automatic differentiation in PyTorch. In: Conference on Neural Information Processing Systems (2017)
14. Krogh, A., Hertz, J.: A simple weight decay can improve generalization. In: Proceedings of the Conference on Neural Information Processing Systems, pp. 950–957 (1991)
15. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2018)
16. Lin, T., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell. **99**, 1 (2018)
17. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.: Learning transferable architectures for scalable image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 8697–7710 (2018)
18. Srivastava, N., Salakhutdinov, R.: Discriminative transfer learning with tree-based priors. In: Advances in Neural Information Processing Systems, pp. 2094–2102 (2013)
19. Murdock, C., Li, Z., Zhou, H., Duerig, T.: Blockout: dynamic model selection for hierarchical deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2583–2591 (2016)
20. Liu, B., Sadeghi, F., Tappen, M.F., Shamir, O., Liu, C.: Probabilistic label trees for efficient large scale image classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 843–850 (2013)
21. Lin, M., Chen, Q., Yan, S.: Network in network. In: International Conference on Learning Representations (2014)
22. Marszałek, M., Schmid, C.: Constructing category hierarchies for visual recognition. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5305, pp. 479–491. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88693-8_35
23. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations. Wiley, Chichester (2016)
24. Chaudhuri, N.R., Chakraborty, D., Chaudhuri, B.: Damping control in power systems under constrained communication bandwidth: a predictor corrector strategy. IEEE Trans. Control Syst. Technol. **20**(1), 223–231 (2012)
25. Simonetto, A., DallAnese, E.: Prediction-correction algorithms for time- varying constrained optimization. IEEE Trans. Signal Process. **65**(20), 5481–5494 (2017)