DuFeNet: Improve the Accuracy and Increase Shape Bias of Neural Network Models

Zecong Ye^{1,2,3} · Zhiqiang Gao^{1,2} · Xiaolong Cui² · Yaojie Wang^{1,2,3} · Nanliang Shan^{1,2,3}

Received: 10 March 2021 / Revised: 19 August 2021 / Accepted: 24 October 2021 / Published online: 4 January 2022
© The Author(s) 2021

Abstract

In image classification field, existing work tends to modify the network structure to obtain higher accuracy or faster speed. However, some studies have found that the neural network usually has texture bias effect, which means that the neural network is more sensitive to the texture information than the shape information. Based on such phenomenon, we propose a new way to improve network performance by making full use of gradient information. The dual features network (DuFeNet) is proposed in this paper. In DuFeNet, one sub-network is used to learn the information of gradient features, and the other is a traditional neural network with texture bias. The structure of DuFeNet is easy to implement in the original neural network structure. The experimental results clearly show that DuFeNet can achieve better accuracy in image classification and detection. It can increase the shape bias of the network adapted to human visual perception. Besides, DuFeNet can be used without modifying the structure of the original network at lower additional parameters cost.

Keywords Gradient information · Neural network · Shape bias · Image classification

1 Introduction

Image classification is to determine what category the input image belongs to according to some classification algorithms and it is an important basis for object detection [13] and image segmentation [20] tasks. In recent years, with the development of deep learning, image classification has made remarkable progress and reached a level beyond human beings in many applications [23].

From recent research, it can be found that convolutional neural networks are not as good at learning shape features as learning texture features. It often needs to learn shape features by some methods such as style transfer, which undoubtedly customize for each data set. In order to solve the problem of neural network in learning shape features, and to simplify the

operation process, we propose the DuFeNet, which not only can improve the accuracy and increase shape bias of neural network models, but also can be used without modifying the structure of original network at lower additional parameters cost. This study aims to effectively alleviate the problem of texture bias. Our method does not require complex operations to change the data set and through the experience the method also improve the accuracy.

The chapters of this paper are arranged as follows: The second section mainly introduces the study of neural network structure and the effect of gradient information in the image classification network. The third section introduces the overall framework and various components of the DuFeNet. The fourth section discusses the experimental details, the quantitative analysis of DuFeNet, checks the effect of DuFeNet on shape bias through specific dataset, uses Grad-CAM [26] method for visual analysis and the method use in detection task. The fifth section summarizes the content of this paper and puts forward the ideas of future work.

✉ Xiaolong Cui
beifangyun@126.com

¹ School of Information Engineering, Engineering University of PAP, Xi'an 710086, China

² Counter-Terrorism Command Information Engineering Research Team, Engineering University of PAP, Xi'an 710086, China

³ Graduate team, Engineering University of PAP, Xi'an 710086, China

2 Related work

2.1 Deep neural network

Since AlexNet [17] wins the 2012 race on the ImageNet competition [3], some top teams have been working on computer vision applications based on convolutional neural networks. Different network structures are designed to improve accuracy. The VGGNet [27] increase the depth of convolutional neural networks. Kaiming He et al. propose ResNet [8], adding shortcut connections to early layers to reuse the features of the early layers. Jie Hu et al. put forward the SE blocks [11] which can adaptively recalibrate channel-wise feature responses. While the accuracy of network is constantly improving, some researchers enable the neural networks to run on mobile devices. MobileNet [9,10,25], ShuffleNet [21,31] and other network structures are put forward. These lightweight network structures can greatly reduce network parameters and calculations without losing much precision. And the design of the network structure relies on the experience and prior knowledge seriously. Some researchers begin to research the algorithm to design the architecture of the network automatically [24]. They try to use reinforcement learning, evolutionary, etc. algorithms to find the best network structure. In addition, some researchers try to apply the transformer to the CV field which is a type of deep network mainly based on self-attention mechanism and achieved results comparable to the CNN method [7].

2.2 The effect of gradient information

It is still important to understand how convolutional neural networks work. Once we find the rules of neural network, we can optimize it pertinently. A common explanation of how the networks work is that continuous convolution layers and pooling layers learn to represent edges and shapes in images [15]. Shapes is a gradient information, and it is well known that, compared with other cues like size or texture, shape is the most important cue for human object recognition [19]. However, recent studies have shown that convolutional neural networks may include texture and color deviations in their renderings [18,28], which is contrary to people's intuition about the shape of image content and human biology. For example, Robert Geirhos et al. [6] put such conflicting hypotheses to a quantitative test by evaluating difference between convolutional neural networks and human observers on images with conflicting texture shape cues. The outcome shows that convolutional neural networks trained through ImageNet dataset [3] are strongly biased towards recognizing textures. Figure 1 is the result of three experiments they have done. Besides, they attempt to increase the shape bias by using style-transfer method that can remove texture cues from the data in order to train more robust and accurate net-

Fig. 1 Accuracies and example stimuli for three different experiments without cue conflicts [6]. The numbers on the histogram represent the accuracy rates of different networks or methods

works. Brochu [1] uses domain-adversarial training in order to further increase the shape bias. Table 1 are the characteristics of different methods.

3 Methods

To solve the problem of neural networks in learning shape features, we propose the DuFeNet. In this section, we introduce the component of DuFeNet and explain why the proposed structure can have a good impact on the performance of the model. Firstly, we introduce the overall framework of DuFeNet. Then we present the details of Gradient Branch, Texture Branch which extract different kinds of features from original image and gradient image. Finally we present Fusion Block and special training method.

3.1 Overall framework design

As shown in Fig. 2, we propose the design of the framework, which includes three parts, namely Gradient Branch (GB), Texture Branch (TB), and Fusion Block (FB). The input of Gradient Branch is a single channel with only edge features image obtained by the edge detection algorithm such as Sobel [5], Prewitt [29], Canny [2] and so on. The main function of Gradient Branch is to make convolutional neural networks learn the knowledge in the case of only the edge feature to classify the image. Texture Branch uses the traditional convolutional neural network to learn the knowledge from the original image. Fusion Block combines Gradient Branch and Texture Branch. The input of Fusion Block is done by fusing features, which are in front of the fully connected layer in Gradient Branch and Texture Branch. Fusion Block can be regarded as the fusion of image gradient information (edge information) and the texture feature to enhance the sensitivity of convolutional neural networks to gradient information.

Table 1 Characteristics of different methods

method	advantages	disadvantages
Style-transfer [6]	Just build the data set and can improve accuracy through special train method	Need to be customized
Style-transfer + domain-adversarial train [1]	Increases the robustness and shape bias	Not provides a gain in accuracy
Our method	Increase the shape bias, improve accuracy, and do not have to change the dataset	–

Fig. 2 The design of the DuFeNet, mainly includes three parts, Gradient Branch, Texture Branch and Fusion Block

We denote the input of DuFeNet as X and its output as Y . The process of edge detection can be expressed as

$$X_E = G(X) \quad (1)$$

where X_E and $G(X)$ denote the gradient image generated from original image and the edge detection algorithm, respectively. In this paper, we take Canny [2] as the edge detection algorithm.

3.2 Gradient Branch and Texture Branch

The target of the Gradient Branch is to make a convolutional neural network classify image by gradient information (edge image). The process of Gradient Branch can be described as follows:

$$F_{\text{gradient}} = GB(X_E, \text{Para}_{\text{GB}}) \quad (2)$$

where Para_{GB} and F_{gradient} denote the parameters of convolutional neural network in Gradient Branch and the vector of

the gradient feature generated from Gradient Branch, respectively. As is shown in Fig. 2, the vector of the gradient feature is the yellow block on the right in Gradient Branch. Since gradient information contains much less information than the original image, a small amount of convolutional layer can be used to learn the gradient information in Gradient Branch. In Fig. 2, we design a three-layer convolutional network without residual or other complex network structures. As a small branch network structure, Gradient Branch can be added to the original network at any time to improve the accuracy and increase the shape bias of neural network models at the cost of a small amount of calculation and parameters.

On the other hand, compared with Gradient Branch, Texture Branch uses a convolutional neural network to learn more texture information of the original image. The process of Texture Branch can be described below:

$$F_{\text{texture}} = TB(X, \text{Para}_{\text{TB}}) \quad (3)$$

where Para_{TB} and F_{texture} denote the parameters of convolutional neural network in Texture Branch and the vector of texture feature generated from Texture Branch, respectively.

Texture Branch uses the convolutional neural network structure that came up before, such as Vgg19 [27], ResNet18 [8], Mobilenetv2 [25], Shufflenetv2 [21]. It can make use of the advantages of network structure to get higher accuracy.

3.3 Fusion Block and special training method

Fusion Block can fuse the gradient features and texture features learned before. The concrete way is to concatenate the vectors from Gradient Branch and Texture Branch. Then we add a layer of fully connected layer (Fc layer). This process can be formulated as follows:

$$Y = FB(F_{\text{gradient}}, F_{\text{texture}}, \text{Para}_{\text{FB}}) \quad (4)$$

where Para_{FB} denotes the parameters of neural network in Fusion Block. Fusion Block makes full use of two features and achieves better performance by combining these two features. Gradient Branch fully exploits the image gradient information that Texture Branch learning is not enough. Fusion Block combines the gradient knowledge with the original network that has texture bias to supplement the texture knowledge.

As for DuFeNet, the network structure is clear and simple, but the training method is special. The training method is divided into two stages. Stage 1: individual train, Stage 2: frozen, and fine-tune.

Stage 1: Train the two networks separately to obtain Para_{GB} and Para_{TB} . Gradient Branch only uses the gradient images as its input while Texture Branch uses the original images. The expression is as follows:

$$Y_{\text{GB}} = F_{\text{CGB}}(\text{GB}(X_E, \text{Para}_{\text{GB}})) \quad (5)$$

$$Y_{\text{TB}} = F_{\text{CTB}}(\text{TB}(X, \text{Para}_{\text{TB}})) \quad (6)$$

where Y_{GB} , Y_{TB} represent the outputs of the individual training process, corresponding to Gradient Branch and Texture Branch respectively. F_{CGB} , F_{CTB} are the functions of the fully connected layer that is added after Gradient Branch and Texture Branch.

Stage 2: When we get Para_{GB} , Para_{TB} in stage 1, we load these parameters in DuFeNet, and frozen these parameters before the Fusion Block. Finally, we fine-tune Fusion Block to make gradient features and texture features fully utilized and achieve the improvement of model performance.

Obviously, there is no complicated structure since each part is distinct and is not mixed together. However, it is worth noting that if we train DuFeNet directly, the performance can not be improved. It may be because when the two branch networks are trained at the same time, their internal parameters will affect each other, the texture and gradient information can not be distinguished for learning. This special method preserves the characteristics of their respective learning.

4 Results

In this section, we describe the details of the experiment, make some quantitative experiments to find out which factors can greatly improve the model accuracy, and use a special dataset to verify the enhancement of shape bias. In order to make the observation more intuitive, we use Grad-CAM method [26] to visualize it. Besides, we use this method in detection experience.

4.1 Experiments settings

The proposed method is evaluated on cifar100 [16], which is a common dataset. In the Texture Branch, we use ResNet18 [8], MobileNetv2 [25], Vgg19 [27], ShuffleNetv2 [21] while we design a network with only a few layers in Gradient Branch. All the experiments are implemented with PyTorch [22] under the ubuntu system on one RTX 2080 GPU. Optimization is performed by using Adam Algorithm [14] with a learning rate of 0.001 and a batch size of 300. All models are trained for 200 epochs from scratch. The Canny Algorithm is implemented with the OpenCV framework. We save the model weights with the highest accuracy on the validation set.

4.2 The advantage of DuFeNet

This subsection mainly verifies the superiority of DuFeNet. And analyze the impact of neural networks with different layers and widths in Gradient Branch on its own accuracy and on the accuracy that brings for DuFeNet. We adopt different networks in Texture Branch and DuFeNet to test the advantage of Gradient Branch. For Gradient Branch, we use the network structure with three convolutional layers as shown in Fig. 2. Table 2 is the results of different networks. We evaluate the networks with four evaluation index, Paras, Mult-Adds, PrecisionT1, PrecisionT5.

It can be found from the Table 2 that DuFeNet can improve the accuracy of any network structure, the increase range of T1 is 0–3%, the increase range of T5 is 0–2.2%, the increase in parameters is about 0.4M, and the increase in computational cost is about 39M. It can be concluded that Gradient Branch is suitable for different network structures. Gradient Branch have a small increase in parameters and calculations, in line with the requirements of being a small part, which can be added to a network at any time to increase the accuracy.

And then we analyze Gradient Branch. As shown in Table 3, a total of 5 different networks are set up, namely Anet, Bnet, Cnet, A2net, and A4net.

Firstly, we analyze the impact of neural networks with different layers and widths in Gradient Branch on its own accuracy. It can be seen from the Table 4 that the accuracy continues to increase as the network width widens or the

Table 2 The advantage of DuFeNet

Network	Paras(M)	Mult-Adds(M)	PrecisionT1 (%)	PrecisionT5 (%)
TB(Vgg19)	39.32	417.42	65.35	83.68
DuFeNet(Vgg19)	39.72	455.78	66.27 (+0.92)	84.07 (+0.39)
TB(ResNet18)	11.22	555.47	69.39	89.85
DuFeNet(ResNet18)	11.61	593.83	71.44 (+2.05)	90.99 (+1.14)
TB(MobileNetv2)	2.36	64.96	62.63	86.84
DuFeNet(MobileNetv2)	2.7	103.00	64.96 (+2.33)	88.16 (+1.32)
TB(ShuffleNetv2)	1.36	43.68	63.01	86.15
DuFeNet(ShuffleNetv2)	1.75	82.04	65.84 (+2.83)	88.32 (+2.17)

Table 3 Different setups of neural network in Gradient Branch

Layer name	Output size	Anet	Bnet	Cnet	A2net	A4net
conv1_x	32 × 32	3 × 3, 64	3 × 3, 64 3 × 3, 64	3 × 3, 64 3 × 3, 64	3 × 3, 32	3 × 3, 16
maxpool	16 × 16	2 × 2, stride 2				
conv2_x	16 × 16	3 × 3, 128	3 × 3, 128	3 × 3, 128 3 × 3, 128	3 × 3, 64	3 × 3, 32
maxpool	8 × 8	2 × 2, stride 2				
conv3_x	8 × 8	3 × 3, 256	3 × 3, 256	3 × 3, 256 3 × 3, 256	3 × 3, 128	3 × 3, 64
	1 × 1	Average pool, 100-d fc				

Table 4 The accuracy of Gradient Branch

Network	Paras(M)	Mult-Adds(M)	PrecisionT1 (%)	PrecisionT5 (%)
Anet	0.39	38.36	31.19	58.32
Bnet	0.43	76.11	31.75	58.88
Cnet	1.17	151.61	37.45	65.51
A2net	0.01	9.74	28.58	55.22
A4net	0.003	2.51	24.42	50.47
ResNet18	11.21	554.29	37.90	64.41

network depth deepens. And it can be seen from the table that when the number of convolution layers is 6 (Cnet) in Gradient Branch, the accuracy of ResNet18 can basically be reached, which shows that the gradient image does not need too many parameters to learn its characteristics and the neural network with only 6 layers of convolution can learn most of the internal information.

However, our method combines ignored edge features with Texture Branch instead of only using the gradient information. We combine the different networks in Gradient Branch and Texture Branch to see how their different combinations affect the accuracy of DuFeNet. Table 5 is the results of the experiment. The first column in the table is the network selected in Gradient Branch, the first line is the network selected in Texture Branch and the second line is results of single Texture Branch.

From Table 5, it is found that different networks in Texture Branch can use Gradient Branch component to improve

the accuracy to different degrees, Vgg19 improves the least, ShuffleNetv2 the most. When using Anet, A2net, and A4net to make network width comparisons, we can find that the network width has little effect on accuracy. When using Anet, Bnet, Cnet, ResNet18 to do network depth comparison, the depth impact the network accuracy is large. The deeper the network, the greater the improvement in the accuracy. Cnet and ResNet18 have the greatest improvement in the accuracy, which reflects that the improvement of the accuracy under multi-layered such as ResNet18 is limited. And Cnet, a network of 6 convolution layers structures, has learned well about the edge features of the image. For other researchers, they can balance the relationship between model accuracy and inference speed to choose a suitable network structure in Gradient Branch for their original network.

Table 5 Impact of different combinations

PrecisionT1 (%)	Vgg19	ResNet18	MobileNetv2	ShuffleNetv2
TB	65.35	69.39	62.63	63.01
Anet	66.27 (+0.92)	71.44 (+2.05)	64.96 (+2.33)	65.84 (+2.83)
Bnet	66.59 (+1.24)	71.61 (+2.22)	65.04 (+2.41)	66.12 (+3.11)
Cnet	66.93 (+1.58)	72.08 (+2.69)	66.20 (+3.57)	67.34 (+4.33)
A2net	66.32 (+0.97)	71.53 (+2.14)	64.68 (+2.05)	65.78 (+2.77)
A4net	66.21 (+0.86)	71.42 (+2.03)	64.49 (+1.86)	65.73 (+2.72)
ResNet18	67.55 (+2.20)	72.24 (+2.85)	66.23 (+3.60)	67.04 (+4.03)

Fig. 3 The increase in shape bias of DuFeNet. On the left is a brief flow chart of DuFeNet and the output of it compares the results of Texture Branch and Gradient Branch. On the right are partial results of the

special dataset [6]. The label is below the test picture and the histogram shows the probability of the true label in the network outputs

4.3 Increase shape bias of neural network

We use three networks trained by the cifar100 dataset, namely ResNet18 + Cnet (DuFeNet), ResNet18 (single TB), and Cnet (single GB). And we use the dataset provided in the paper [6] (using only the categories contained in cifar100) to explore the influence of Gradient Branch on shape bias by comparing the correct category probability which can be generated by softmax operation on Y_{TB} , Y_{GB} , Y . Y_{TB} , Y_{GB} denote the outputs of the individual training processes, corresponding to Gradient Branch and Texture Branch, respectively. Y denotes the output of DuFeNet. Figure 3 shows the results of some of the images.

Figure 3 that only a few outputs get high probability because the model is not trained on the special data field (Silhouette), still the differences of networks' shape bias degree can be clearly seen. There is a big difference between categories. For example, in the terms of Bottle pictures, the results of DuFeNet are better than Texture, but for other labels, Gradient Branch is the best; Fusion Block is the second, and Texture Branch is the worst. And we can draw a conclusion that for different categories, the accuracy impact caused by

Fig. 4 Visual presentation by Grad-CAM. The first line is the original images. The second line is Grad-CAM and Guided Grad-CAM pictures generated from the network in Texture Branch. The third line is Grad-CAM, Guided Grad-CAM pictures generated from the network in Gradient Branch, and binary pictures generated by the Canny Algorithm edge detection

DuFeNet varies. At the meantime we can find that the shape bias degree of Texture Branch is 0–1% in most cases which indicates that Texture Branch does learn most texture features but ignores gradient (edge) features in the image.

To make it easier to understand the role of Gradient Branch, we visualized some pictures of cifar100 dataset using the Grad-CAM method [26]. The method makes full use of the gradients of any target concept to produce a coarse localization map. It highlights the important regions in the image for predicting the concept. Thus it makes 'visual explana-

Table 6 AP for different backbone

Backbone	AP50	APs	APm	APl
TB	62.03	3.61	22.34	49.81
GB	42.19	1.27	11.54	32.80
DuFeNet	63.07	4.13	22.79	51.09

tions' for decisions from the convolutional neural network models, making them more transparent and easier to understand. The results are shown in Fig. 4.

In Fig. 4, it can be seen from Grad-CAM pictures that Gradient Branch is more dependent on global features to identify objects because the red area from Gradient Branch is larger than the red area from Texture Branch in the heat map. The result is also in line with a characteristic that the edge features are generally sparse in the image and not as dense as the texture. And it can be found from Guided Grad-CAM pictures that Gradient Branch is more in line with the intuitive human feeling while Texture Branch is more in favor of other features such as textures in pictures. All the results above show that Gradient Branch effectively extracts edge features and is capable of image classification by edge features.

4.4 Detection experiment

To highlight the breadth and practicality of this method, we use this method in detection task. We change the backbone of centernet [32], a SOTA detection algorithm. Pascal VOC [4] is a popular detection dataset. Firstly we use TB(ResNet18) and GB(MobileNetv2) as backbone to train on VOC 2007 and VOC 2012 trainval and test on VOC2007 test set with 384×384 resolution. And then combine TB and GB to get DuFeNet backbone. Different from the classification task, the method in target detection concat two feature maps from GB and TB before the up-sample stage, and the classification task is to concat two vectors. We use batchsize 32, adam algorithm, learning rate $1.25e-4$ and train 70epochs with learning

rate dropped $10 \times$ at 45 and 60 epochs based on the project [12]. Table 6 shows AP at IOU threshold 0.5 and AP for small, middle, large object. From this we can know that the GB also can work in detection task as a backbone because the loss function can converge and get 42.19 AP50. DuFeNet can improve AP on large, medium, and small object. Figure 5 shows the AP of different categories in VOC. We can conclude that for some categories with relatively simple internal textures, our method can greatly increase its AP, just like boat and cow. This also provides us with a new idea for optimizing target detection algorithms.

5 Conclusion

In this paper, we propose a novel DuFeNet, including three parts, namely Gradient Branch, Texture Branch and Fusion Block. DuFeNet uses dual networks to extract diverse features for boosting the representing ability of the learned features for classification. We have found that the gradient information of the image not only can increase the shape bias of neural networks but also strengthen the learning ability of networks. Meanwhile, it improves the accuracy of the classification network and enhances the robustness of the network. And DuFeNet can help well take advantage of this gradient information. This method verifies that it is also possible to further improve the gradient deviation of the model by changing the network structure. Extensive experiments demonstrate the high performance and computation efficiency of DuFeNet. And the original network can be easily transformed to DuFeNet structure within a few parameters and amount of calculation to get greater performance. In the future, as the gradient information is relatively sparse in the image, some convolution methods that can rapidly expand the receptive field, such as dilated convolution [30], can be used to quickly obtain the shape features and further simplify the neural network structure in Gradient Branch. Therefore, we can use this method to achieve the same precision effect with fewer parameters and calculations.

Fig. 5 AP for different categories in VOC

Funding This work is supported by the National Natural Science Foundation of China (NSFC) under Grant U1603261 and the Natural Science Foundation Project of Xinjiang Province of China under Grant 2016D01A080.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Brochu, F.: Increasing shape bias in imagenet-trained networks using transfer learning and domain-adversarial methods. arXiv preprint [arXiv:1907.12892](https://arxiv.org/abs/1907.12892) (2019)
- Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 679–698 (1986)
- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
- Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>
- Gao, W., Zhang, X., Yang, L., Liu, H.: An improved sobel edge detection. In: 2010 3rd International Conference on Computer Science and Information Technology, vol. 5, pp. 67–71. IEEE (2010)
- Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint [arXiv:1811.12231](https://arxiv.org/abs/1811.12231) (2018)
- Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., Yang, Z., Zhang, Y., Tao, D.: A survey on visual transformer. arXiv preprint [arXiv:2012.12556](https://arxiv.org/abs/2012.12556) (2020)
- 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)
- Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1314–1324 (2019)
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint [arXiv:1704.04861](https://arxiv.org/abs/1704.04861) (2017)
- 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)
- JDAI-CV: centerx. <https://github.com/JDAI-CV/centerX> (2020)
- Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., Qu, R.: A survey of deep learning-based object detection. *IEEE Access* (2019). <https://doi.org/10.1109/ACCESS.2019.2939201>
- Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
- Kriegeskorte, Nikolaus: Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annu. Rev. Vis. Sci.* **1**(1), 417 (2015)
- Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
- 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)
- Kubilius, J., Bracci, S., Op de Beeck, H.: Deep neural networks as a computational model for human shape sensitivity. *J. Vis.* **16**, 759 (2016). <https://doi.org/10.1167/16.12.759>
- Landau, B., Smith, L., Jones, S.: The importance of shape in early lexical learning. *Cognit. Dev.* **3**, 299–321 (1988). [https://doi.org/10.1016/0885-2014\(88\)90014-7](https://doi.org/10.1016/0885-2014(88)90014-7)
- Long, J., Shelhamer, E., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation, pp. 3431–3440 (2015). <https://doi.org/10.1109/CVPR.2015.7298965>
- Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: practical guidelines for efficient CNN architecture design. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 116–131 (2018)
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, pp. 8026–8037 (2019)
- Rawat, W., Wang, Z.: Deep convolutional neural networks for image classification: a comprehensive review. *Neural Comput.* **29**(9), 2352–2449 (2017)
- Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Chen, X., Wang, X.: A comprehensive survey of neural architecture search: challenges and solutions. arXiv preprint [arXiv:2006.02903](https://arxiv.org/abs/2006.02903) (2020)
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted Residuals and Linear Bottlenecks, pp. 4510–4520 (2018). <https://doi.org/10.1109/CVPR.2018.00474>
- Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626 (2017)
- 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)
- Xiao, B., Jaiswal, M., Poovendran, R.: Assessing Shape Bias Property of Convolutional Neural Networks, pp. 2004–20048 (2018). <https://doi.org/10.1109/CVPRW.2018.00258>
- Yang, L., Wu, X., Zhao, D., Li, H., Zhai, J.: An improved prewitt algorithm for edge detection based on noised image. In: 2011 4th International Congress on Image and Signal Processing, vol. 3, pp. 1197–1200. IEEE (2011)
- Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint [arXiv:1511.07122](https://arxiv.org/abs/1511.07122) (2015)
- Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6848–6856 (2018)
- Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint [arXiv:1904.07850](https://arxiv.org/abs/1904.07850) (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.