Deep Superpixel Convolutional Network for Image Recognition

Xianfang Zeng, Wenxuan Wu, Guangzhong Tian, Fuxin Li, and Yong Liu, *Member, IEEE*

***Abstract*—Due to the high representational efficiency, superpixel largely reduces the number of image primitives for subsequent processing. However, superpixel is scarcely utilized in recent methods since its irregular shape is intractable for standard convolutional layer. In this paper, we propose an end-to-end trainable superpixel convolutional network, named SPNet, to learn high-level representation on image superpixel primitives. We start by treating irregular superpixel lattices as a 2D point cloud, where the low-level features inside one superpixel are aggregated to one feature vector. We replace the standard convolutional layer with the PointConv layer to handle the irregular and unordered point cloud. Besides, we propose grid based downsampling strategies to output uniform 2D sampling result. The resulting network largely utilizes the efficiency of superpixel and provides a novel view for image recognition task. Experiments on image recognition task show promising results compared with prominent image classification methods. The visualization of class activation mapping shows great accuracy at object localization and boundary segmentation.**

***Index Terms*—Deep learning, representation learning, image recognition, superpixel.**

I. Introduction

SUPERPIXELS are over-segmented image fragments that are formed by aggregating local pixels with similar low-level features. They provide perceptionally meaningful segmentation of image content, reducing the number of basic image units for subsequent image processing. Such property makes it become a fundamental representation in various vision tasks [1], [2] such as saliency object detection [3], image matting [4], semantic segmentation [5], [6], and tracking [7], [8]. However, in the deep learning era, we can rarely find the utilization of superpixel in recent vision algorithms that are mostly developed on convolutional neural networks (CNNs). The behind reasons boil down to: 1) most superpixel algorithms are in the iterative optimization framework, which disobeys the general end-to-end training pipeline of deep models. 2) superpixels usually have an irregular shape and thus the lattice constructed by superpixel is intractable for the standard convolution operation.

To tackle the second challenge, several methods have been proposed to fuse superpixel into standard CNNs. Raghudeep *et al.* [5] propose the bilateral inception module for semantic segmentation task to utilize the structure information contained in superpixel. This module can be inserted to standard CNNs and propagates feature between pixels along with object edges. Another strategy for employing superpixel is extending general grid pooling layer to superpixel pooling, as done in [6], [9]. This pooling layer samples feature inside each superpixel, which can better group similar features.

In contrast to existing methods, we do not explore using superpixel as an auxiliary tool for standard CNNs. We propose the superpixel convolutional network aiming at directly learning high-level representation on superpixel image primitives. To this end, we view the irregular lattice formed by superpixel as 2D point cloud and it is worth nothing that some operations on point processing can be extended to construct deep superpixel network. We start by transforming the image features to a 2D point cloud format via aggregating the similar low-level features grouped by superpixel mask. This processing transfers the representation primitive from pixel to superpixel, reducing redundant calculations inside superpixel. The corresponding challenge is that the neighbor features of points can occur at any position around the reference point, while standard convolutional layer can only handle regular neighbors. Hence, we replace the standard convolutional layer with PointConv layer [10] to fuse local features and learn high-level representation on points. The PointConv layer treats the weight of convolutional kernel as continuous function of local coordinates and generalizes the powerful convolution operation to irregular 2D point data. Besides, we propose three grid based downsampling strategies for transformed 2D points. Those methods sample a specific point from each small grid and output uniform sampling result. As a comparison, the general 3D point sampling methods tend to keep the outline of points. With above fundamental layers, we can build end-to-end trainable convolutional networks for superpixel primitives.

In summary, our contributions are two folds. 1) We propose an end-to-end trainable superpixel convolutional network, which takes superpixel as image primitives and provides a novel view for image recognition task. 2) Experiments on image recognition task show comparable results to prominent image classification methods. The visualization results show high precision at object

Manuscript received February 22, 2021; accepted April 14, 2021. Date of publication April 27, 2021; date of current version May 20, 2021. This work was supported by the National Natural Science Foundation of China under Grant 61836015 and Grant 61771193. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jianxin Li. *(Corresponding author: Yong Liu.)*

Xianfang Zeng and Guangzhong Tian are with the Institute of Cyber-System and Control, Zhejiang University, Hangzhou 310027, China (e-mail: zzlongjuanfeng@zju.edu.cn; gztian@zju.edu.cn).

Wenxuan Wu and Fuxin Li are with the CORIS Institute, Oregon State University, Corvallis, OR 97331 USA (e-mail: wuwen@oregonstate.edu; lifuxin2001@gmail.com).

Yong Liu is with the State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China (e-mail: yongliu@iipc.zju.edu.cn).

Digital Object Identifier 10.1109/LSP.2021.3075605

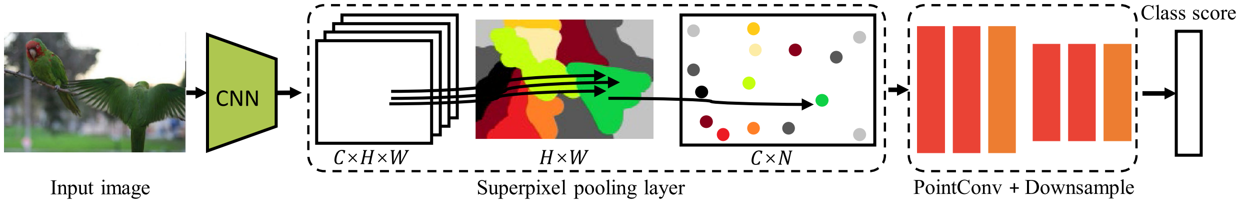


Fig. 1. Pipeline of our superpixel convolutional network. The input image is first fed to several convolutional layers to learn low-level representation. The extracted feature is transformed to a 2D point cloud format via the superpixel pooling layer, where the features grouped in one superpixel are aggregated to one feature vector. For the final class score, we stack the 2D PointConv layers and point downsampling layers to extract high-level features from the points. Note that the main representation learning of our model is performed at the 2D point stage, where the basic unit is superpixel.

localization and boundary segmentation, which indicates a potential application for weakly supervised semantic segmentation task.

II. Method

We propose a novel superpixel convolutional network to learn representation on superpixel primitives. The overall idea is that we treat the irregular lattice formed by superpixel as 2D point cloud and extend some operations on points to build deep neural network on this lattice. Fig. 1 illustrates the training pipeline of our model when given a pair of image and superpixel mask. We start by transforming low-level features to 2D point cloud via aggregating the similar features grouped by superpixel mask. We then extend the general convolutional layer to handle irregular point cloud data. The challenge is that the neighbor features of points can occur at any position around the reference point. To tackle this problem, we view the weight of convolutional kernel as continuous function of local coordinates and thus a MLP is used to estimate the convolutional kernel given the relative coordinates with respect to the reference point. Besides, we propose grid based downsampling strategies for uniform sampling result. The full architecture involves three fundamental layers: superpixel pooling layer, 2D PointConv layer, and grid based downsampling layer. The main details of our method are presented in three parts:

Superpixel pooling layer. As illustrated in Fig. 1, the superpixel pooling layer [9] provides a simple and flexible strategy for aggregating local features. Unlike regular pooling layer which operates the rigid neighbor features, it utilizes the superpixel mask [11]–[15] to group near features. The grouped features over a superpixel are aggregated to one feature vector via the average function. This step transfers the basic feature unit from pixel to superpixel and fuses the redundant information in the input image. Suppose the intermediate feature map is $I \in \mathbb{R}^{C\times H\times W}$, composed of C channels and $H \times W$ pixels. The one-channel superpixel segmentation is denoted as $S \in \mathbb{R}^{H\times W}$, where $S_{i,j} = [1, N]$ is the integer label of the pixel at position (i, j). The superpixel pooling layer transforms the feature map to 2D point cloud, represented by a feature matrix $P \in \mathbb{R}^{C\times N}$. This process is formulated as

$$P_{c,n} = \frac{1}{\mathcal{N}(n)} \sum_{i}^{\mathcal{N}(n)} I_{c,i}, \tag{1}$$

where $\mathcal{N}(n) = \{(i, j) : S_{i,j} = n\}$ denotes the set of pixels belong to n-th superpixel. Note that the feature is aggregated independently at the channel dimension.

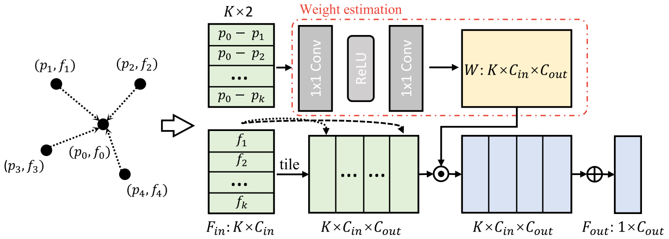


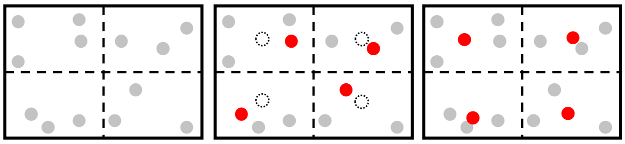
Fig. 2. The process of conducting 2D PointConv on local regions around the center point (p_0, f_0), where p means coordinates and f is the learned feature. The PointConv layer takes the local coordinates to estimate the convolution kernel, which is utilized to fuse the input features of the K nearest neighbors.

2D PointConv layer. This extends general convolutional layers to conduct convolution operation on irregular point cloud data. Fig. 2 illustrates this dynamic filtering process over the local region around one point. The input of PointConv consists of the features coming from the K nearest neighbors and the corresponding local coordinates. Similar to [10], we consider the convolutional kernel estimation as nonlinear functions of the local coordinates. The coordinates are fed to a multilayer perceptron to estimate the convolutional weight with respect to the center point. Then a convolution operation is performed on the corresponding local features with the estimated kernel. Same as the common convolution, this operation is an element-wise multiplication followed by a summation. Consider the local region $\mathcal{G}$ around the center point $o = (i, j)$, and (δ_i, δ_j) can be any possible position in this region. We formulate the 2D PointConv layer at position (i, j) as

$$\mathcal{F}(W, f)_{i,j} = \sum_{(\delta_i,\delta_j)\in\mathcal{G}} W(\delta_i, \delta_j) f(i + \delta_i, j + \delta_j), \tag{2}$$

where W represents the estimated convolutional kernel, and $f(i + \delta_i, j + \delta_j)$ is the feature at corresponding position.

Grid based downsampling layer. The irregular and unordered data is intractable to general 2D pooling layers. We extend the common pooling layers, e.g. the average pooling, to down sample irregular points. As shown in Fig. 3(a), we split the whole point cloud into small grids for grouping unordered points. We propose three grid-based downsampling methods for pooling the points in one grid: grid-mid, grid-mean, and grid-max. As illustrated in Fig. 3(b), grid-mid chooses the point nearest to the grid center for the pooled set. As displayed in Fig. 3(c), we calculate the mean point for each grid and select it as the new feature point, named grid-mean. In grid-max, the average reduction function is replaced by the maximum



(a) Input points (b) grid-mid (c) grid-mean

Fig. 3. Grid based 2D point downsampling methods, where the gray points are original and the red one is pooled. (a) The input points are splited to different grids. (b) Grid-mid: the point nearest to the grid center is chosen to construct the new set. (c) Grid-mean: calculate the average point for each grid and select it.

function to select new points. These grid based methods behave more like general 2D pooling layers, generating more uniform sampling result. As a comparison, the common 3D downsampling methods, e.g. farthest point sampling, tend to keep the geometric outline. Given the original point cloud $P \in \mathbb{R}^{C \times N}$, the downsampling process is formulated as

$$\bar{P}_{c,k} = reduce\{P_{c,i} | i : g_i = k\}, \tag{3}$$

where g_i means the grid index of one point, and $reduce$ represents a reduction function like $average$ or max.

III. Experiment

We first evaluate the effect of different downsampling strategies and superpixel algorithms. Then we provide comparisons against mainstream image classification models. Finally, we visualize some class activation mappings of our model.

Implementation details. We implement our superpixel convolutional network using PyTorch framework. Our model is trained by SGD with weight decay 0.0001, momentum 0.9, and a mini-batch of 256 on 4 Tesla V100 GPUs. The learning rate starts from 0.1 and is divided by 10 every 30 epochs. All models are trained for 90 epochs with the standard cross entropy loss. The MLP in PointConv consists of two 1 × 1 convolutional layers and the number of hidden units is 8, while this number is set as 4 in our lite variation.

Datasets and baselines. In order to evaluate the effect of downsampling strategies, we report results on the MNIST dataset [16]. To evaluate the representational capacity of superpixel convolutional network, we further conduct our ablation study and comparisons on ImageNet-1K [17] dataset, which collects 1.28 million training images and 50 k validation images from 1000 classes. The input in 224 × 224 resolution is randomly cropped from a 256 × 256 resized image, with random horizontal flipping. On ImageNet-1 k dataset, the superpixel mask is extracted by Watershed [14] algorithm in pre-processing where the markers is set as 784 and compactness is 0.001. We compare our SPNet with three mainstream image recognition networks: VGG [18], ResNet [19], and DenseNet [20]. All models, including baselines and our model, are trained under the same experimental protocol.

A. Experiments on Downsample Methods

We conduct quantitative experiments on MNIST dataset to evaluate the effect of different downsample methods. The results are shown in Table I. The input image is resized to two resolutions: 14 × 14 and 7 × 7, where a 2 × 2 or 4 × 4 patch is treated as a superpixel. The classic convolutional neural network, LeNet [16], is chosen as the performance reference. We construct the models with 5 PointConv layers for performance evaluation. We first evaluate random sampling and farthest point sampling (FPS), which are widely used in point cloud recognition task. Both two methods perform significantly worse than the baseline. We believe the reason is that those methods tend to keep the overall geometry structure. However, image classification is rely more on discriminative fragments to recognize objects. We then assess three grid-based downsample methods: grid-mid, grid-max, and grid-mean. Those methods split the whole point cloud into different grids and choose the specific point from each grid. For instance, grid-mean choose the mean point to represent all points in one grid. Therefore, these methods can output more uniform sampling result. As displayed in Table I, all grid-based methods can achieve comparable performance to the baseline. Among all variations, grid-mean achieves the best performance, outperforming the baseline by 0.5% approximately on classification accuracy.

TABLE I
Classification Accuracy on MNIST With Different Downsampling Strategies

Model	14 × 14	7 × 7
LeNet [16]	97.55	94.13
Our-random	84.33	76.22
Our-FPS	83.84	78.12
Our-grid-mid	97.17	93.80
Our-grid-max	97.95	93.66
Our-grid-mean	**98.01**	**94.61**

TABLE II
Top-1 and Top-5 Test Error(%) on ImageNet Dataset of SPNets Using Different Superpixel Methods. Patch Means That a Image Patch Is Treated as a Superpixel

Method	top-1 err.(%)	top-5 err.(%)
Patch	25.75	8.28
Felzenszwalb [13]	26.23	8.44
Quickshift [12]	25.65	7.94
SSN [15]	25.43	7.65
SLIC [11]	24.65	7.48
Watershed [14]	**23.11**	**6.37**

B. Experiments on Superpixel Algorithm

To evaluate the effect of superpixel mask in our framework, we conduct an ablation study on ImageNet dataset. Table II shows top-1 and top-5 test error of SPNets using different superpixel algorithms including Felzenszwalb [13], Quickshift [12], SSN [15], SLIC [11], and Watershed [14]. We can see that the best result outperforms the worst by 3.12% on top-1 and 2.07% on top-5. This shows that the selection of superpixel algorithms has important influence on final recognition accuracy. The best result is achieved by Watershed, which tends to segment superpixels with uniform size. Since the 2D points are transformed via superpixel mask, the result also indicates that 2D points with

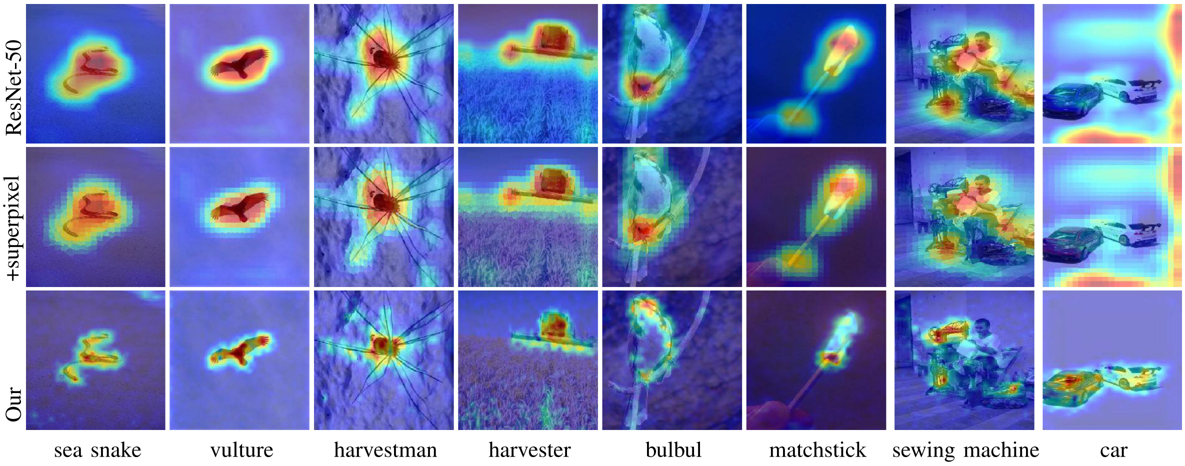


Fig. 4. Visualization of class activation mapping for different methods. '+superpixel' means the maps of ResNet post-processed by superpixel. The brighter regions represent the more discriminative parts. Best viewed in color.

TABLE III
TOP-1 AND TOP-5 TEST ERROR(%) AS WELL AS MODEL SIZE AND COMPLEXITY ON IMAGENET DATASET

Model	top-1 err.(%)	top-5 err.(%)	Params(M)	GFLOPs
VGG-19 [18]	25.76	8.15	143.68	19.7
DenseNet-169 [20]	24.00	7.00	14.15	**3.42**
ResNet-34 [19]	26.70	8.58	21.80	3.68
ResNet-50 [19]	23.85	7.13	25.56	4.12
Our-lite	24.93	7.23	**13.40**	3.46
Our	**23.11**	**6.37**	21.15	6.98

uniform distribution are able to achieve higher performance on image recognition task.

C. Classification on ImageNet-1 k

Table III shows the top-1 and top-5 test error on the ImageNet-1 k dataset. We build models with approximately 50 layers to perform quantitative comparisons against mainstream image recognition networks. The number of learnable parameters and GFLOPs are utilized to assess model size and complexity. Our full model outperforms other mainstream classification networks. With a smaller model size, our model achieves an improvement of 0.74% on top-1 error and 0.76% on top-5 error over ResNet-50. Our lite variation (Our-lite) reduces 37% model size by decreasing the number of hidden units in PointConv from 8 to 4. For similar model size, Our-lite achieves a decrease of 0.93% on top-1 over DenseNet-169. It is a reasonable result since Our-lite has less layers and no dense connection mechanism. For the similar network structure, it has a improvement of 1.77% on top-1 and 1.35% on top-5 over ResNet-34, with less GFLOPs and model size.

D. Class Activation Mapping

To understand superpixel convolutional network, we show the class activation mapping (CAM) on Fig. 4 via Grad-CAM [21], where the more discriminative areas are covered with the brighter color. Compared with ResNet-50, the CAM generated by superpixel convolutional network performs better at object boundary and localization accuracy. For object boundary, CAM generated by SPNet has more clear boundary for thin and long objects such as 'sea snake' and 'vulture'. For some complicated classes like 'harvestman' and 'harvester,' both methods can cover the object regions, while our method tends to focus on body parts to avoid involving the background. As for localization accuracy, our SPNet has activation map that marks head rather than feet as discriminative area for 'bulbul,' in line with human intuition. Another case is that the CAM of ResNet improperly highlights the human finger for 'matchstick' class. As displayed in the last two columns, we also show some failure case of ResNet: It wrongly recognizes 'people' as 'sewing machine,' and incorrectly highlights the background for 'car'.

To further evaluate the effect of superpixel in activation maps, we also visualize the CAM of the revised ResNet ('+superpixel'), which adopts the watershed superpixel algorithm [14]. Specifically, we replace the last average pooling layer in ResNet-50 with superpixel pooling layer. Both the original and the revised ResNet-50 have similar activation maps on all classes. This indicates that the difference of visualization results between the ResNet-50 and SPNet is more likely caused by the network architecture rather than the utilization of superpixel. The ability of SPNet in precisely locating the class activation mapping provides a potential application for weakly supervised semantic segmentation task.

IV. CONCLUSION

We propose a novel superpixel convolutional network, named SPNet, which is an end-to-end trainable deep network on image superpixel primitives. Due to the representational efficiency, our model can reduce redundant calculations inside superpixels for subsequent processing. Experiments on image recognition task show that our SPNet achieves comparable results to prominent image classification methods. Meanwhile, the visualization results indicate a potential application of SPNet in weakly supervised semantic segmentation task.

REFERENCES

[1] Q. Wang, Y. Yuan, P. Yan, and X. Li, “Saliency detection by multiple-instance learning,” *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 660–672, Apr. 2013.

[2] Q. Wang, J. Gao, and Y. Yuan, “A joint convolutional neural networks and context transfer for street scenes labeling,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1457–1470, May 2018.

[3] S. He, R. W. Lau, W. Liu, Z. Huang, and Q. Yang, “Supercnn: A superpixelwise convolutional neural network for salient object detection,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 330–344, 2015.

[4] X. Li, K. Liu, and Y. Dong, “Superpixel-based foreground extraction with fast adaptive trimaps,” *IEEE Trans. Cybern.*, vol. 48, no. 9, pp. 2609–2619, Sep. 2018.

[5] R. Gadde, V. Jampani, M. Kiefel, D. Kappler, and P. V. Gehler, “Superpixel convolutional networks using bilateral inceptions,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.

[6] S. Kwak, S. Hong, and B. Han, “Weakly supervised semantic segmentation using superpixel pooling network,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, 2017, pp. 4111–4117.

[7] F. Yang, H. Lu, and M.-H. Yang, “Robust superpixel tracking,” *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1639–1651, Apr. 2014.

[8] Y. Yuan, J. Fang, and Q. Wang, “Robust superpixel tracking via depth fusion,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 15–26, Jan. 2014.

[9] M. Schuurmans, M. Berman, and M. B. Blaschko, “Efficient semantic image segmentation with superpixel pooling,” 2018, *arXiv:1806.02705*.

[10] W. Wu, Z. Qi, and L. Fuxin, “Pointconv: Deep convolutional networks on 3D point clouds,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9621–9630.

[11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “Slic superpixels compared to state-of-the-art superpixel methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[12] A. Vedaldi and S. Soatto, “Quick shift and kernel methods for mode seeking,” in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 705–718.

[13] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.

[14] Y. Deng, B. S. Manjunath, and H. Shin, “Color image segmentation,”. in *Proc IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 1999, pp. 446–451.

[15] V. Jampani, D. Sun, M.-Y. Liu, M.-H. Yang, and J. Kautz, “Superpixel sampling networks,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 352–368.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[17] O. Russakovsky *et al.*, “Imagenet large scale visual recognition challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[18] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. Int. Conf. Learn. Represent.*, 2015.

[19] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[21] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.