Fully CapsNet for Semantic Segmentation

Su Li, Xiangyu Ren, and Lu Yang(✉)

School of Automation Engineering,
University of Electronic Science and Technology of China, Chengdu 611731, China
yanglu@uestc.edu.cn

Abstract. Fully convolutional networks (FCNs) are powerful models for semantic segmentation. But convolutional networks fail to perform well in recognizing and parsing images with spatial variation. In this paper, a novel Capsule network called Fully CapsNet is proposed. We introduce Capsule to FCN and improve Equivariance of the neural network in image segmentation. Compared with traditional FCN based networks, a trained Fully CapsNet shows robustness in recognizing image pixels with more or less spatial variation. Each capsule layer is connected by dynamic routing algorithm. The effectiveness of the proposed model is verified through PASCAL VOC. Results show that Fully CapsNet outperforms the FCN in understanding both original images and rotated images.

Keywords: Fully convolutional network · Semantic segmentation
Capsule network · PASCAL VOC

1 Introduction

Image segmentation is one of the main research field in image processing. Semantic segmentation can understand images at pixel level. Image semantic segmentation can be regarded as the gist of image understanding which plays an important role in many applications. For example, street view recognition in robot guiding system [17], determination of landing site of UAV [6] and wearable device application [9] etc.

The idea of semantic segmentation has been raised before deep learning is popularized. Many semantic algorithms such as Thresholding methods [7], Clustering-based segmentation methods [16] and Graph partitioning segmentation methods [11,14] have been proposed in computer vision. The work of semantic segmentation at that time was to segment the image according to low-level visual cues. For instance, abstracting images to the form of graphs and then achieve semantic segmentation on the basis of Graph theory. Other methods require supporting information such as bounding box and scribbled lines. These algorithms perform poorly when applied to complex images without enough supporting information. Semantic segmentation algorithms attracted growing research interests as deep learning popularized. Fully convolutional networks (FCN) [10,13] and deep convolutional Nets [2–4] for semantic segmentation are

J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11257, pp. 392–403, 2018.
https://doi.org/10.1007/978-3-030-03335-4_34

widely employed by deep learning based approaches. Key observation of FCN is that the fully connected layers in classification networks can be viewed as convolutions with kernels that cover their entire input regions. Feature maps still need to be upsampled because of pooling operations in CNNs. Instead of using simple bilinear interpolation, deconvolutional layers can learn the interpolation by themselves. However, information of images losses because of pooling layers. Therefore, skip connections are introduced to address this problem from higher resolution feature maps [13].

One of the main drawbacks of convolutional networks is their lacking in '*comprehension*' to images. Human vision builds up coordinate frames when recognizing images. Coordinate frames effect the way human observe images through comprehension to space. However, coordinate frame does not exist in convolutional networks. A novel neuron structure called '*Capsule*' proposed by Hinton in [12] manages to solve this problem. Hinton et al. believes that the relationship between objects and observers (e.g. pose of objects) should be described by a set of neurons instead of a single neuron. Priori knowledge of coordinate frames can then be expressed effectively. This set of neuron in general is called 'Capsule'. Furthermore, Capsule network offers equivariant mapping, which means that both location information and pose information of objects can be reserved. Routing tree in Capsule networks maps the partial hierarchy of the target, therefore each part is assigned to a whole. In summary, Capsule network is robust to rotation, translation and other forms of transformations.

In this paper, we managed to leverage both FCN and Capsule network and proposed a novel neural network called Fully CapsNet. To our best knowledge, this is the first work to modify neural structure in FCN with Capsule for semantic segmentation. Fully CapsNet introduces the principle of capsule to Fully convolutional network. It adds several capsule layers and linked each layer by *dynamic routing algorithm* [12]. Fully CapsNet improves the robustness of convolutional network to pose transformation of objects as well as accuracy in semantic segmentation.

The rest of this paper is organized as follows: Related literatures for the construction of our model are introduced in Sect. 2. In Sect. 3, details of the proposed work is demonstrated and reviewed. The network structure of Fully CapsNet is presented in Sect. 4. Comparison experiments are demonstrated and discussed in Sect. 5. Segmentation results are evaluated on PASCAL VOC where Fully CapsNet will show the significant outperformance compared to the original FCN.

2 Related Work

2.1 Deconvolutional Network

Deconvolutional network is first proposed by Zeiler et al. in [18], which is a framework that permits the unsupervised construction of hierarchical image representations. Compared with CNN, which obtain feature map from input images convoluted by feature filter, deconvolutional network restores input images through

feature map convoluting feature filter. For each input image x, there are several feature z to represent its latent features, and feature filters F can be obtained through learning the given images $X = \{x^1, x^2, \ldots, x^N\}$. The trained filters can be then used to infer the feature map when a new image is given.

The input of the first layer of deconvolutional network is the original image and the output is the feature map z^1 extracted from the input image. The rest of the layers has an input of the previous layer's output feature map z^{L-1} and an output of feature map z^L. Several cost functions are introduced to optimize the parameters F and z. To learn the parameters in filters, deconvolutional network alternately minimizes cost functions over the feature maps while keeping the filters fixed (i.e. perform inference). The trained filters can be then used to infer the feature map when a new image is given.

Deconvolutional network is a powerful tool for mid and high level feature learning. Visualization and understanding of each feature map obtained from convolutional layers can be achieved through deconvolution.

2.2 VGG

VGG is a convolutional neural network model proposed by Simonyan et al. in [15]. VGG is made up a concise structure. There are 5 convolutional layers, 3 fully connected layers and a softmax output layer. Each layer is separated by a max-pooling layer and the activation unit of the hidden layer adopts ReLu function. In this paper a VGG-16 structure is adopted. VGG-16 contains 16 layers in the framework in total. Small-scale kernel function is one of the main feature of VGG. Convolutional layers in VGG consist of several small-scale kernel functions (3×3), where kernel functions in other structure such as AlexNet [8] are bigger in size (7×7). On the one hand, the amount of parameters can be reduced by bringing down the size of kernel functions. On the other hand, more nonlinear mapping can be carried out, which can increase the fitting and expressing ability of the network.

Our work is similar to [5] in the sense that an architectural change to layers is proposed. The authors propose to modify several layers in VGG.16 to formulating them as CapsNets, creating a new class of FCN, called Fully CapsNet. The idea can be extended to other forms of FCNs.

3 Preliminaries

3.1 Fully Convolutional Network

FCN is proposed by Long et al. in [13]. It replaces the fully connected layers with several convolutional layers. A net with only layers of convolutional layers computes a nonlinear filter is called fully convolutional network. Each layer of data in a convolutional network is a three-dimensional array of size $h \times w \times d$, where h and w are spatial dimensions, and d is the feature or channel dimension. The first layer is the input image, with pixel size $h \times w$, and d color channels.

Locations in higher layers correspond to the locations in the image they are path-connected to, which are called their receptive fields. Writing x_{ij} for the data vector at location (i, j) in a particular layer, and y_{ij} for the following layer, these functions compute outputs y_{ij} by

$$y_{ij} = f_{ks}(\{x_{si+\delta}\}) \tag{1}$$

where k is called the kernel size, s is the stride, and f_{ks} determines the layer type. This functional form is maintained under composition, with kernel size and stride obeying the transformation rule:

$$f_{ks} \cdot g_{k's'} = (f \cdot g)_{k'+(k-1)s',ss'} \tag{2}$$

A real-valued loss function composed with an FCN defines a task. If the loss function is a sum over the spatial dimensions of the final layer, $l(x;\theta) = \Sigma_{ij} l'(x_{ij};\theta)$, its gradient will be a sum over the gradients of each of its spatial components. Thus stochastic gradient descent on l computed on whole images will be the same as stochastic gradient descent on l', taking all of the final layer receptive fields as a minibatch. In order to obtain the original size of image, an upsampling layer or deconvolutional layer is applied in FCN. It simply reverses the forward and backward passes of convolution. However, upsampling produces coarse segmentation maps because of loss of information during pooling. Thus, skip architecture is introduced to FCN. A skip architecture is learned end-to-end to refine the semantics and spatial precision of the output. In addition, FCN ignores spatial regularization procedure which is normally used in pixel-level segmentation. Researches have been done regarding to these problems, such as RFCN [5], ResNet, GoogLeNets [1] etc.

3.2 Capsule Networks

Capsules are locally invariant groups of neurons that learn to recognize the presence of visual entities and encode their properties into vector outputs, with the vector length (limited to $[0, 1]$) representing the presence of the entity. To achieve the limitation of the vector length, a squashing function (Eq. 3) is used.

$$v_j = \frac{||s_j||^2}{1 + ||s_j||^2} \frac{s_j}{||s_j||} \tag{3}$$

where v_j is the vector output of capsule j and s_j is its total input.

Each capsule can learn to identify certain objects or object-parts in images. Within the framework of neural networks, several capsules can be grouped together to form a capsule-layer where each unit produces a vector output instead of a scalar activation.

Sabour et al. introduced a routing-by-agreement mechanism in [12] for the interaction of capsules within deep neural networks with several capsule-layers, which works by pairwise determination of the passage of information between capsules in successive layers. For each capsule h_i^l in layer l and each capsule

$h_j^{(l+1)}$ in the layer above, a coupling coefficient c_{ij} is adjusted iteratively based on the agreement (cosine similarity) between h_i's prediction of the output of h_j and its actual output given the product of c_{ij} and h_i's activation:

$$S_j = \sum_i c_{ij} \hat{u}_{j|i}, \quad \hat{u}_{j|i} = W_{ij} u_i \tag{4}$$

where $\hat{u}_{ij}$ is the prediction vector of all the capsule below, u_i is the output of one capsule and W_{ij} is a weight matrix.

The coupling coefficients c_{ij} inherently decide how information flows between pairs of capsules. Sabour et al. proposed a *routing softmax* which enables the coupling coefficients between capsules in layer i and above sum up to 1. The initial logits b_{ij} of coupling coefficients are the log prior probabilities that capsule i should be coupled to capsule j:

$$c_{ij} = \frac{\exp(b_{ij})}{\Sigma_k \exp(b_{ik})} \tag{5}$$

For a classification task involving K classes, the final layer of the CapsNet can be designed to have K capsules, each representing one class. Since the length of a capsule's vector output represents the presence of a visual entity, the length of each capsule in the final layer can then be viewed as the probability of the image belonging to a particular class. The algorithm is shown in Procedure 1 [12].

Procedure 1 . ROUTING ALGORITHM

1: Routing $(\hat{u}_{j|i}, r, l)$
2: for all capsule i in layer l and capsule j in layer $l+1$: $b_{ij} \leftarrow 0$
3: **for** r iterations **do**
4: for all capsule i in layer l: $c_i \leftarrow \text{softmax}(\text{b}_\text{i})$
5: for all capsule j in layer $l+1$: $s_j \leftarrow \sum_i c_{ij} \hat{u}_{j|i}$
6: for all capsule j in layer $l+1$: $v_j \leftarrow \text{squash}(\text{s}_\text{j})$
7: for all capsule i in layer l and capsule j in layer $l+1$: $b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \times v_j$
8: **return**
9: **end for**

4 Fully CapsNet

Capsule network is essentially parallel attention network. Each capsule layer focuses on linking to the capsules in next layer, which are more active to the information extracted in the previous capsule layer and then ignore those inactive capsules. The idea of capsule is more close to that how human react to information processing: information processing between neurons are vector instead of scale. For example, CNNs have the ability to recognize a human face with all facial features, even they are not in their correct position. This is because

the pooling layers in CNNs simply learn each of these features separately abandoning the spatial connection among them. Capsule, however, builds a feature group containing both features and their spatial connections. Thus, CapsNet would not recognize it a human face if the facial features are not in the correct order. The transition of information between capsules is conducted by *dynamic routing*. On the basis of discussion above, the idea of Capsule Network and its routing algorithm (*dynamic routing*) is applied in our model.

4.1 Construction of Fully CapsNet

Fully CapsNet is similar in structure to the FCN model in general. A traditional FCN structure *VGG-16* is selected as the feature extractor of Fully CapsNet. *VGG-16* is a mature and widely used Convolutional Neural Network structure and the original FCN is proposed based on it. Then we modified the output layer of *VGG-16*. Instead of upsampling directly with the output of last convolutional layer, several capsule layers are added after them. Firstly, the feature map from conv.6 is transformed into the form of a vector. Information of the map is extracted by dynamic routing method and capsules in each layer are activated according to Margin Loss function. Finally, upsampling (or deconvolution) method is used. Deconvolution helps to restore the size of initial image. The *Skip* structure in FCN is also applied in Fully CapsNet in order to fine-tune the output results. It learns to combine coarse, high layer information with fine, low layer information. A demonstration for the architecture of Fully CapsNet is shown in Fig. 1.

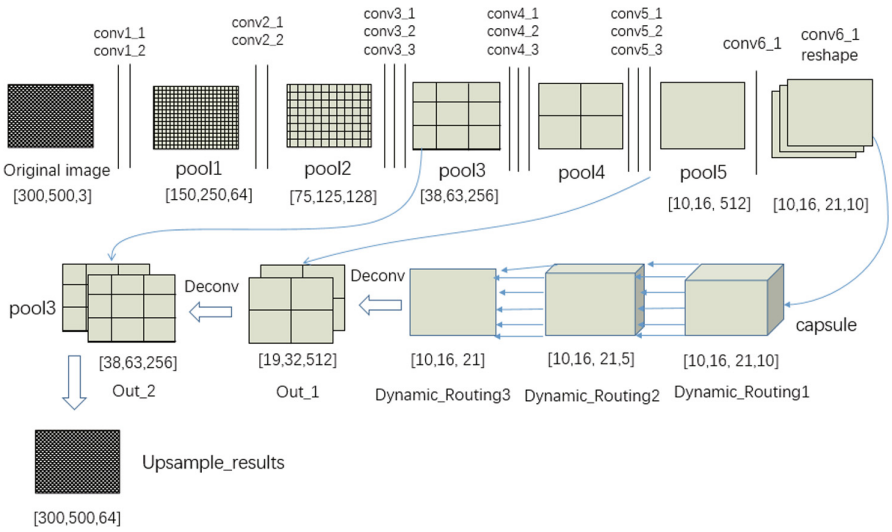


Fig. 1. Framework of fully CapsNet

The whole process can be simplified to several progress: Feature extraction → dynamic routing → upsampling, or Encoder → Decoder process.

It is noticeable in Fig. 1 that the size of the featured map in the con6$_1$ *reshape* stage increased by one dimension. The purpose of reshaping the size is aimed at preparing for the introduction of Capsule structure.

4.2 Modification on Routing Algorithm

The Dynamic routing algorithm is fully connected in every pixel. In other word, it requires adequate storage in computers to store data and powerful abilities for calculation, which is hardly achievable for general users. Thus, we modified the routing algorithm through a *partial connection method*. For example, given a image with the size 20×30, the original dynamic routing algorithm requires a $20 \times 30 \times 20 \times 30$ (360000 in total) space to store the weighting value between each pixel, which can be space consuming when the size of input images increase. As for partial connection method, which splits the original image into 150 (10×15) small images of size 2×2. Therefore, the space required is reduced to $10 \times 15 \times 2 \times 2 \times 2 \times 2$ (2400 in total). What's more, in order to reduce the storage space for calculations required by Capsule Network the dynamic routing algorithm is applied only on image data in higher layers, such as data in con6 layers. The data in higher layers are the feature maps extracted from lower layers, which contain essential information that represent the nature of the input images. In this way, the adjusted routing algorithm largely cuts down the usage of storage and calculation complexity.

Fully CapsNet has the advantage of taking input of arbitrary size and produce correspondingly-sized output with efficient inference and learning inherited form FCN. Besides, Fully CapsNet also has the ability of equivariance inherited from Capsule. It is robust to rotation, translation and other forms of transformations. In the following section, the effectiveness of Fully CapsNet is verified and analyzed through PASCAL VOC.

5 Experimental Analysis

The performance of Fully CapsNet is evaluated through a set of experiments, in which we compare Fully CapsNet with FCN on their accuracy rate. The experiments are based on PASCAL VOC. The segmentation results are evaluated by two methods: pixel-wise accuracy rate and MAP value. The PASCAL VOC dataset is analyzed in Sect. 5.1. Experimental results are displayed and analyzed in Sect. 5.2.

5.1 Datasets

A major part in computer vision is about object recognition, detection and classification, which are fundamental functions in application field. Therefore, the correctness and efficiency of an algorithm is verified through whether or not these

three functions can be completed. Large quantities of images are then collected by researchers to be applied to their algorithms. PASCAL VOC Challenge is a platform where algorithms are contrasted based on the same data set. PASCAL VOC provides adequate standardized image data sets for pixel-wise scene understanding as well as a common set of tools for accessing the data sets and annotations. There are twenty object classes in PASCAL VOC and are divided into four categories: Person, Animal, Vehicle and Indoor objects. In this paper PASCAL VOC 2012 is selected for semantic segmentation. Figures 2 and 3 shows the original images and their groundtruth segmentation results obtained from PASCAL VOC 2012.

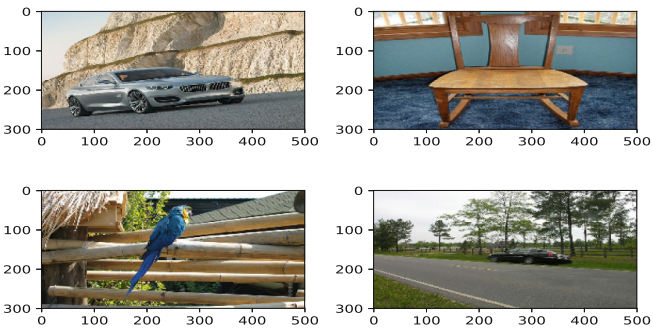


Fig. 2. Original images

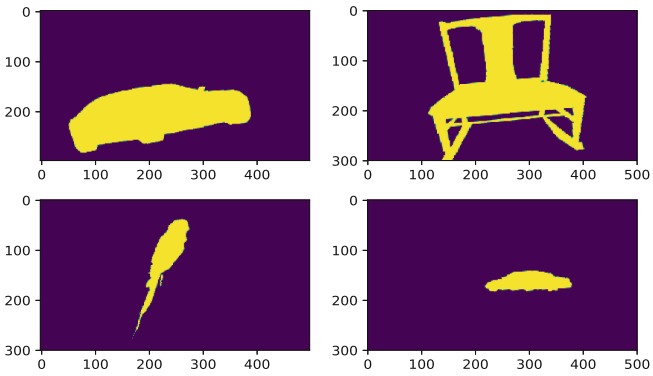


Fig. 3. Groundtruth

5.2 Segmentation Results and Analysis

The authors qualitatively compare images generated randomly using both Fully Convolutional Network and Fully CapsNet. Figure 4 shows segmentation results of a image by the two algorithms, where the first column is the groundtruth and the second and third column are segmentation results generated by Fully

CapsNet and Fully Convolutional Network respectively. Obviously, Fully CapsNet's segmentation results outperforms initial Fully Convolutional Network's. The accuracy rate of the segmentation results of other objects are shown in Table 1.

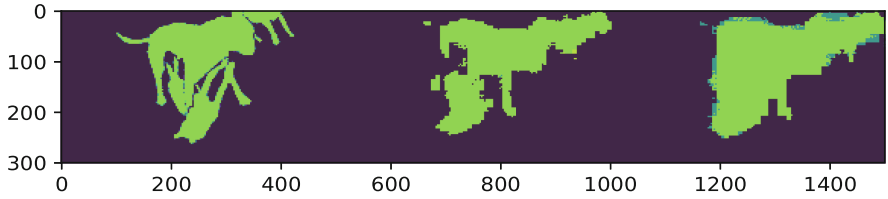


Fig. 4. Segmentation results of original images

Table 1. Accuracy rate of upright images by the two methods

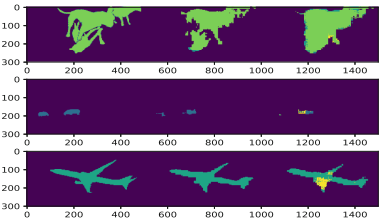
Object	Aeroplane	Sofa	Bird	Boat	Bottle	Bus	Car	Cat
Fully caps	**0.88**	**0.62**	**0.73**	**0.68**	0.48	**0.91**	0.62	**0.86**
FCN	0.77	0.37	0.69	0.49	**0.60**	0.75	**0.75**	0.78
Object	Chair	Cow	Diningtable	Dog	Horse	Motorbike	Person	Sheep
Fully caps	0.35	**0.94**	**0.55**	**0.79**	**0.79**	**0.80**	0.70	**0.88**
FCN	0.21	0.63	0.47	0.72	0.64	0.77	**0.74**	0.72

As it can be seen in Table 1, Fully CapsNet outperforms FCN in segmenting normal position images. Although segmentation of some objects such as chair (colored in red) and dining table is not accurate because the background of the image effects segmentation results in boundary regions.

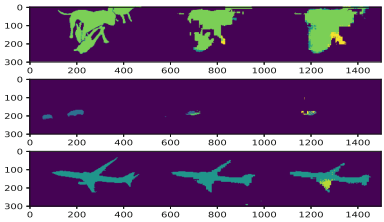
For example, similar color exists between target objects and backgrounds, fuzzy object edges etc. All of these factors contribute to low accuracy segmentation rate. Apart from improvements in accuracy, Fully CapsNet also has the ability in parsing rotated images.

In order to show 'Equivariance' in Fully CapsNet, we managed to rotate several images obtained from the training set and then set them as input of the trained network. The objects in the selected images are rotated by 5, 10, 15 and 20° while the size of each image remain fixed. Figure 5 shows some of the segmentation results from Fully CapsNet and FCN.

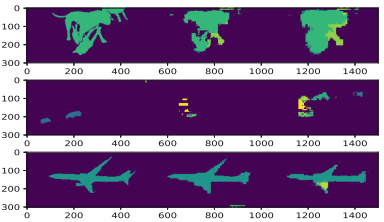
In Fig. 5, the first column of each image shows the ground truth, the second column shows segmentation results from Fully CapsNet and the third column shows segmentation results from FCN. It is obvious that Fully CapsNet shows better equivariance compared with FCN when the pose of objects varies to a small extent. Take Fig. 5(a) as an example, Fully CapsNet can segment the edge of the target object while FCN performs badly in segmenting the edges as well as classifying the target objects. Experiments with more degree of rotation of the objects are carried out and the results are shown in Table 2.



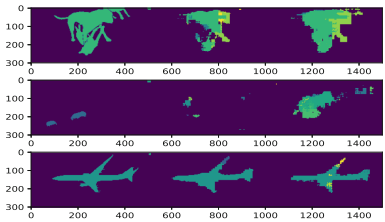
(a) Results with 5 degree rotation



(b) Results with 10 degree rotation



(c) Results with 15 degree rotation



(d) Results with 20 degree rotation

Fig. 5. Results with different rotation degree

Table 2. Segmentation results of rotated objects

Fully CapsNet	*Aeroplane*	*Bicycle*	*Bird*	*Boat*	*Bottle*	*Bus*	*Car*	*Cat*
5	0.85	0.37	0.71	0.66	0.44	0.89	0.60	0.86
10	0.84	0.34	0.71	0.62	0.43	0.88	0.56	0.85
15	0.83	0.34	0.69	0.61	0.42	0.87	0.57	0.85
20	0.80	0.34	0.67	0.59	0.43	0.83	0.54	0.85
FCN	*Aeroplane*	*Bicycle*	*Bird*	*Boat*	*Bottle*	*Bus*	*Car*	*Cat*
5	0.75	0.24	0.62	0.59	0.39	0.84	0.53	0.76
10	0.73	0.24	0.60	0.57	0.38	0.82	0.53	0.77
15	0.73	0.23	0.60	0.53	0.34	0.79	0.51	0.75
20	0.70	0.22	0.60	0.48	0.33	0.76	0.48	0.74
Fully CapsNet	*Cow*	*Diningtable*	*Dog*	*Horse*	*Motorbike*	*Person*	*Train*	*Sheep*
5	0.94	0.54	0.78	0.78	0.79	0.67	0.90	0.87
10	0.90	0.51	0.79	0.78	0.78	0.66	0.88	0.84
15	0.90	0.48	0.77	0.74	0.77	0.64	0.85	0.85
20	0.86	0.49	0.75	0.73	0.75	0.62	0.80	0.83
FCN	*Cow*	*Diningtable*	*Dog*	*Horse*	*Motorbike*	*Person*	*Train*	*Sheep*
5	0.80	0.50	0.82	0.75	0.73	0.63	0.82	0.77
10	0.80	0.47	0.79	0.73	0.72	0.61	0.80	0.73
15	0.78	0.42	0.77	0.72	0.71	0.58	0.79	0.76
20	0.76	0.40	0.74	0.70	0.68	0.55	0.75	0.73

6 Discussion and Future Work

Fully convolutional networks are powerful deep learning models for semantic segmentation. Motivated by the success of Capsule network over CNNs at improving the network's ability to comprehend images, we proposed a Fully CapsNet, a FCN framework but incorporates Capsule network instead of CNNs as discriminators when modeling image data. Fully CapsNet adapts to recognizing spatial transformation of objects in trained images. The effectiveness of the model is verified through PASCAL VOC and compared with original Fully convolutional network. Results show that Fully CapsNet out performs FCN in parsing both original images and rotated images.

However, the proposed method shows robustness in recognizing rotated images only to a small extent of rotation. In addition, Capsule network requires tremendous space to store data and powerful calculating ability due to its full connection structure in routing algorithm. Simply applying partial connection reduces the performance of Capsule network. Further research works need to handle these problems.

Acknowledgement. This research was supported by NSFC (No. 61871074) and Fundamental Research Funds for the Central Universities (ZYGX2018J064).

References

1. Ballester, P., Araujo, R.M.: On the performance of GoogLeNet and AlexNet applied to sketches. In: Thirtieth AAAI Conference on Artificial Intelligence, pp. 1124–1128 (2016)
2. Branson, S., Horn, G.V., Belongie, S., Perona, P.: Bird species categorization using pose normalized deep convolutional nets. eprint Arxiv (2014)
3. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: delving deep into convolutional nets. Computer Science (2014)
4. Chen, L.C., 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)
5. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems, pp. 379–387 (2016)
6. Fitzgerald, D.L.: Landing site selection for UAV forced landings using machine vision. Unmanned Arial Vehicle (2007)
7. Han, S.Q., Wang, L.: A survey of thresholding methods for image segmentation. Syst. Eng. Electron. **41**, 233–260 (2002)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
9. Lee, C.M., Schroder, K.E., Seibel, E.J.: Efficient image segmentation of walking hazards using IR illumination in wearable low vision. In: International Symposium on Wearable Computers, pp. 127–128 (2002)

10. Li, H., Qian, X., Li, W.: Image semantic segmentation based on fully convolutional neural network and CRF. In: Yuan, H., Geng, J., Bian, F. (eds.) GRMSE 2016. CCIS, vol. 698, pp. 245–250. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-3966-9_27
11. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut": interactive foreground extraction using iterated graph cuts. In: ACM SIGGRAPH, pp. 309–314 (2004)
12. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Advances in Neural Information Processing Systems, pp. 3859–3869 (2017)
13. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 640–651 (2014)
14. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. IEEE Computer Society (2000)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Computer Science (2014)
16. Wang, S.L., Cao, A.J., Chen, C., Wang, R.Y.: A comparative study on fuzzy-clustering-based lip region segmentation methods. Commun. Comput. Inf. Sci. **234**, 376–381 (2011)
17. Wong, Y.W., Tang, L., Bailey, D.: Vision system for a robot guide system. In: Fourth International Conference on Computational Intelligence, Robotics and Autonomous Systems, pp. 337–341 (2007)
18. Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: Computer Vision and Pattern Recognition, pp. 2528–2535 (2010)