Computers, Materials & Continua CMC, vol.60, no.2, pp.575-597, 2019

A Review on Deep Learning Approaches to Image Classification and Object Segmentation

Hao Wu[1], Qi Liu[2, 3, *] and Xiaodong Liu[4]

Abstract: Deep learning technology has brought great impetus to artificial intelligence, especially in the fields of image processing, pattern and object recognition in recent years. Present proposed artificial neural networks and optimization skills have effectively achieved large-scale deep learnt neural networks showing better performance with deeper depth and wider width of networks. With the efforts in the present deep learning approaches, factors, e.g., network structures, training methods and training data sets are playing critical roles in improving the performance of networks. In this paper, deep learning models in recent years are summarized and compared with detailed discussion of several typical networks in the field of image classification, object detection and its segmentation. Most of the algorithms cited in this paper have been effectively recognized and utilized in the academia and industry. In addition to the innovation of deep learning algorithms and mechanisms, the construction of large-scale datasets and the development of corresponding tools in recent years have also been analyzed and depicted.

Keywords: Deep learning, image classification, object detection, object segmentation, convolutional neural network.

1 Introduction

With the continuous improvement of computing system, such as the help of GPU and distributed computing system, the training of large-scale multi-layer artificial neural network has become possible. In addition, the project of ImageNet [Deng, Dong, Socher et al. (2009)] makes a fertile soil for deep learning to develop. It offers a very giant image database which contains 12 subtrees with 5247 synsets and 3.2 million images in total. This storage was going to bigger after it became public. More than a million of the images have a clear category tag and an annotation of the object in the image. The tags and annotations are up corrected or updated when mistakes found. The giant data quantity and highly tagging make the ImageNet almost become the standard the image processing

[1] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), School of Computer and Software, Nanjing University of Information Science & Technology, No. 219, Ningliu Road, Nanjing, 210044, China.

[2] Shandong BetR Medical Technology Co., Ltd.

[3] School of Computer and Software, Nanjing University of Information Science & Technology, No. 219, Ningliu Road, Nanjing, 210044, China.

[4] School of Computing, Edinburgh Napier University, 10 Colinton Road, Edinburgh, EH10 5DT, UK.

[*] Corresponding Author: Qi Liu. Email: qrankl@163.com.

in deep learning especially computer vision. Researchers from all the world compete their image classification, object detection and location algorithms in the ImageNet Large Scale Visual Recognition Competition (ILSVRC). The most attractive achievement at the beginning is the work from Krizhevsky et al. [Krizhevsky, Sutskever and Hinton (2012)] in 2012. They give a very efficient and effective network architecture that has a great impact on subsequent researches. This network was named as AlexNet by the researchers which is an homage to Yann LeCuns pioneering LeNet 5 network [Lecun, Boser, Denker et al. (1989)]. They propose many efficient optimization techniques, such as Dropout, ReLU (Rectified Linear Unit) and LRN (Local Response Normalization). Many follow-up networks were put forward at the basement of the AlexNet such as VGG [Simonyan and Zisserman (2014)], GooLeNet [Szegedy, Liu, Jia et al. (2015)], ResNet [He, Zhang, Ren et al. (2016)]. They extend depth and width of the AlexNet basically and optimize the structures of networks. These networks have a great impact on computer vision and pushed artificial intelligence technology to a new height. They are very inspiring for later researchers and have had far reaching influence.

Deep learning has become a hot research topic for its excellent performance. More and more researchers from different fields devote their efforts into deep learning and combine it with their own research projects. Countless innovations have emerged constantly in recent years. For example, Google made a great progress with the neural machine in their translation systems [Wu, Schuster, Chen et al. (2016)]. Further, Van Den Oord et al. [Van Den Oord, Dieleman, Zen et al. (2016)] design a WaveNet which can generate human likely voice from text. On the other hand, deep learning shows great strength in the field of games. The famous AlphaGo [Silver, Huang, Maddison et al. (2016); Silver, Schrittwieser, Simonyan et al. (2017)] defeat many human top Go masters in 2016 and 2017. Before that, the machine equipped with deep reinforcement learning technologies is realized the human like video games control in 2015. Even some robot algorithms have surpassed humans' recordings [Mnih, Kavukcuoglu, Silver et al. (2015)]. Besides, deep learning also shows its ability in artistic creation. GAN [Goodfellow, Pouget-Abadie, Mirza et al. (2014)] is a network that can generate fraudulent data based on training data. The generated images from the large data set are hard to find differences from the training data. Gatys et al. [Gatys, Ecker and Bethge (2015)] propose an artistic style transfer neural network. It can redraw the image according to other artistic works following the style of the learnt images. Besides, deep learning also shows its powerful performance in the tradition artificial intelligence such as NLP, auto driving, medical industry and robot industry et al. Innumerable specific neural networks and optimization methods have been invented.

This paper mainly discusses the major progress of deep learning in image processing in recent years, especially in the fields of image classification, target detection and object segmentation. The three research areas have both connected and progressive relationships. The connected thing is that they are all based on the basic idea of a convolutional neural network. The progressive relationship is that their difficulty is getting higher and higher. Target detection and object segmentation all use some basic network models in image classification. The image classification algorithm based on convolutional neural network gives a lot of new ideas to target detection and object segmentation and achieves very good results.

On the other hand, efficient experimental tools, high-performance computing platforms high-quality and large-scale training data sets are also an essential part of achieving an outstanding model. Therefore, this paper compares the typical experimental tools and training datasets in Section 2 firstly. To achieve better results, only increasing the width and depth of the network is far from enough. Next, this paper compares the typical deep learning networks developed in recent years and refine their advantages in Section 3. In the Section 4, the paper lists and compares several networks that have the most recognized results in the field of target detection. Finally, the Section 5 briefly describes the research progress in semantic segmentation and instance segmentation.

2 Typical experimental tools and training datasets

A workman must sharpen his tools if he is to do his work well. Good tools can make the research process go more efficiently and successful. For deep learning, the tools can be divided into two parts. The first part is a good computing platform, including hardware and software. Another part is the high-quality large-scale training datasets.

2.1 Typical experimental tools

Deep learning is a class of computationally intensive algorithms. They rely on high performance computing resources heavily, especially floating-point operations. Fortunately, with the development of technology, the floating-point operation ability of GPU has been emphasized in the industry. More and more researchers are using GPU and even distributed GPU clusters to speed up the training of large-scale artificial neural networks [Li, Zhang, Huang et al. (2016)]. Therefore, to facilitate the adoption of the high-performance computing resources well and help researchers to focus on the implement of algorithms agilely, a variety of programming tools and frameworks emerge as the times require. Besides, most of these tools are not only for deep learning. They can also do a lot different kinds of scientific speed-up calculation in a very easy way.

The frameworks in the Tab. 1 are very popular both in the industry and academia. They provide rich convenient interfaces for mathematical computation. Users can use these interfaces to build their own neural network models conveniently with powerful GPU/CPU even their clusters without worrying about tedious computing details.

Some of these frameworks lay particular emphasis on the underlying design of mathematical interfaces. Therefore, there is still a lot of work to do before users realize their own experiments. To further simplify programming, a more advanced framework arises at the historic moment. The high-level frameworks, such as Keras, use some of above frameworks as the backend so that combine ease of use and high performance together but sacrifice certain expansibility.

Table 1: Comparison of programming tools supporting deep learning

Features	**TensorFlow**	**MXNet**	**PaddlePaddle**	**Caffe**	**Torch**	**Theano**
Main develop language	C++/ CUDA	C++/ CUDA	C++/ CUDA	C++/ CUDA	C++/Lua/ CUDA	Python/C++/CUDA
Sub-language	Python	Python/R/ Julia/Go	Python	Python/Matlab	-	-
Hardware	CPU/GPU/ Mobile	CPU/GPU/ Mobile	CPU/GPU/ Mobile	CPU/GPU	CPU/GPU	CPU/GPU
Cluster Enable	Yes	Yes	Yes	No	No	No
Speed	Medium	Fast	Fast	Fast	Fast	Medium
Extensibility	Good	Good	Good	Medium	Good	Good
Documents	Common	Rich	Rich	Rich	Rich	Common
OS support	Linux, OSX, Win	Linux, OSX, Win	Linux, OSX	Linux, OSX, Win	Linux, OSX	Linux, OSX, Win
Command (CMD)/ Configuration (Conf)	Conf	Both	Conf	Conf	CMD	Conf
Net structure	Tensor graph	Uncertain	Layered	Layered	Layered	Tensor graph

Table 2: Common datasets of image processing

Name	Release time	Type	Size	Description
MNIST	1998	Image, binary	60,000 train samples and 10,000 test samples, a total of about 12M zipped	Classical dataset, only handwritten digital images, academic standards.
PASCAL VOC	2005	Image	20 classes. The train/val data has 11,530 images containing 27,450 ROI annotated objects and 6,929 segmentations. 2 GB of final version.	The quality of the data set is good, and the annotation is complete. It is very suitable for testing the performance of the algorithm. It was updated from 2005 to 2012.
CIFAR	2009	Image, RGB	CIFAR-10: 10 categories and 50000 training images and 10000 test images CIFAR-100: 100 classes, each has 600 pictures.	CIFAR is a very good small and medium scale data set for the image classification algorithm test.
ImageNet	2009	Image	14,197,122 images, 1,034,908 images with bounding box annotations, 1.2 million images with SIFT features, 1 TB in total.	It is very convenient to use. It has been widely applied in computer vision field and has almost become the "standard" dataset for deep learning algorithm performance in image domain.
COCO	2014	Image	Multiple objects per image, more than 300,000 images, more than 2 million instances, 80 object categories, 5 captions per image, 40 GB.	Image annotations information of COCO not only has category, location information, but also describes the semantic text of image, almost become the standard data set of image semantic understanding algorithm performance evaluation.
Open Image	2016	Image	9 million images URL, more than 6000 classes, 1.5 GB (images excluded).	High quality for network training, but only provides picture URLs, which may not be easier to use.
Youtube-8M	2016	Video	8 million videos, 50 million hours in total, 4800 classes, less than 1.5 TB after compression.	Only uses more than 1000 of the public video resources on YouTube.

As the excellent performance of deep learning is more and more favored by the market, the customized chips and computing systems come into being, such as NVIDIA DGX with NVIDIA Tesla V100s, Google TPU. However, they are all expensive or no public for sale. In addition to speeding up on general-purpose chips, researchers are also trying to design specialized chips for accelerated calculations, such as FPGA, ASIC, and achieved good performance [Nurvitadhi, Sheffield, Sim et al. (2017)]. What is more, researchers have already successfully ported models to embedded devices [Hegde, Ramasamy and Kapre (2016)] and mobile devices [Tsung, Tsai, Pai et al. (2016)].

2.2 Typical data set comparison

The development of deep learning cannot be separated from the development of data sets. The next table shows the typical dataset of image processing fields. They all play an important role in the recent neural network researches whatever in industry application or academic research.

Tab. 2 shows the common data sets of image processing which is used frequently. Some of these data sets have been unable to meet the needs of modern machine learning but they are still popular, such as MNIST. Because they can still verify the basic function of algorithms just like the first "hello world" program or be the standard of the contrasts in some performances of different algorithms. Some of the data sets were continually updated, such as ImageNet and PASCAL VOC. But the updates are stopped now. Because they cannot meet the needs like the MNIST.

The Fig. 1 shows some pictures of several typical image datasets. The picture (a) is some handwritten digits samples from MNIST, which is a subset of National Institute of Standards and Technology (NIST). The digits have been size-normalized and centered in a fixed-size image. This data set has a long history, and there are only ten kinds of pictures, but the quality of data set is very high. So many researchers regard it as the simplest data sets for testing algorithms. It is a very simple and meaningful benchmark in the field of image classification. The pictures (b) and (c) are two more modern practical datasets. They are very suitable for modern artificial neural network training. The ImageNet is a large-scale hierarchical image database which aims to populate the majority of the 80,000 synsets of WordNet with an average of 500-1000 clean and full resolution images. The hierarchical structure of each node is depicted by thousands of images. An average of more than five hundred of each node. They are all annotated by manual labelling. When ImageNet was born, deep learning based on big data has not been widely concerned. It's a very forward-looking dataset. On the basis of ImageNet, ImageNet Large Scale Visual Recognition Competition (ILSVRC) also has a great impact on the promotion of machine learning and artificial intelligence. Researchers and enterprises from all over the world compete against their algorithm performance on the basis of ImageNet. This competitive activity has greatly promoted the progress of related intelligent algorithms. At the same time, similar competitions emerge in an endless stream. Many datasets of other research fields are emerged accompanied by related competitions that attract a lot of researchers' interests from all over the world. For example, The Microsoft COCO (Common Objects in Context) [Lin, Maire, Belongie et al. (2014)] dataset is designed not only for image classification, but also object recognition

in the context of the broader question of scene understanding. Specifically, the COCO can be used to do object segmentation and recognition in context. Most pictures have multiple objects which are very different from ImageNet that only each object per image. It has more than 300,000 images and more than 2 million instances in total. The objects can be classified into 80 categories and you can find more than 100, 1000 people from the dataset. Besides, each image has 5 captions in texts. This dataset aims at scene understanding, which is mainly intercepted from complex everyday scenes. The targets in the images are calibrated by precise segmentations.

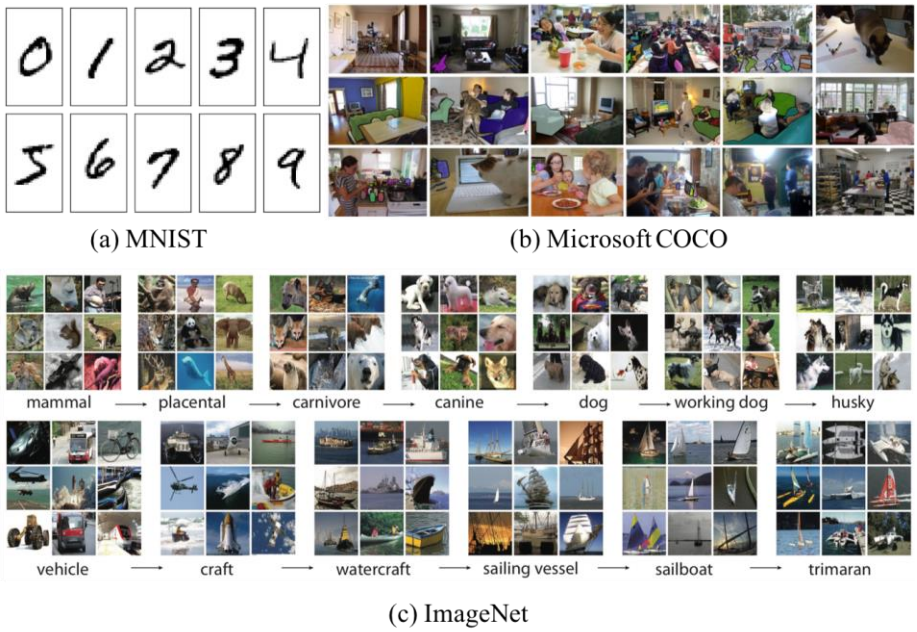

Figure 1: Samples of some typical datasets

Most of the deep learning algorithms are typical supervised machine learning. Therefore, accurate high-quality training samples are very important, which can be as important as the algorithm. Not a single one can be omitted. Collecting so much data, classifying and tagging it one by one is a quite great project. Thanks to the authors of these open source data sets for their unselfish dedication to their work which is meaningful. These data sets have greatly promoted the development of artificial intelligence technology. Many researchers, engineers, and students with limited funds and strength can easily verify their innovation in deep learning with these open source data set and made a leap forward. Since then, CNN has become a hot research topic in the academic world. The COCO dataset has 91 categories, although there are fewer than the ImageNet category, but there are more images in each category, which is conducive to obtaining more capabilities in each category in a particular scene. Compared with PASCAL VOC, there are more categories and images.

The researchers from Google and CMU [Li, Wang, Agustsson et al. (2017)] find that the quality and quantity of data are crucial to the training of deep learning models. This

conclusion is based on an already existing JFT image dataset [Sun, Shrivastava, Singh et al. (2017)]. The data set was first proposed by scientists such as Geoffrey Hinton et al. and expanded by Hinton et al. [Hinton, Vinyals and Dean (2015)]. The JFT dataset has over 300 million images and is marked with 18291 categories. It is much larger than the size of ImageNet, but the accuracy is not high. They find that the performance of visual tasks continues to increase linearly with the magnitude of the training data size. Network performance increases linearly with the magnitude of training data. Representation learning (or pre-training) is still of great use. By training a better basic model, it can improve the performance of visual tasks. Besides the hardware performance of computer system, the training datasets are also very important to deep neural network models.

3 An analysis of convolutional neural network based image classification

Image classification means that the image is structured into a certain category of information, and the image is described with a previously determined category or instance ID. This task is the simplest and most basic image understanding task, and it is also the task of the deep learning model to achieve the first breakthrough and realize large-scale application. Among them, ImageNet is the most authoritative evaluation set. Each year, ImageNet Large Scale Visual Recognition Competition (ILSVRC) has spawned a large number of excellent in-depth network structures, providing the basis for other tasks. In the application field, the recognition of faces, scenes, etc. can be classified as classification tasks.

In the domain of image classification, the convolutional neural network shows excellent performance. In general, the CNNs are the state-of-the-art contrast to the classic algorithms. The structural features of the convolutional neural network are more suitable for solving the problem of image field. Through the continuous research and improvement of its structure, a series of network models have been formed, which have been successful in a wide range of practical applications. The convolutional neural network can reduce the parameters needed to learn by using the spatial structure relation, thus improving the training efficiency of the back-propagation algorithm. In the convolutional neural network, the first volume layer will accept a pixel level image input, each operation only a small image processing convolution, and convolution change and then spread to the back of the network, each layer of convolution will have the most effective feature extraction data. This method can extract feature the most basic image, such as edges or corners in different directions, and then combined and abstract formation characteristics of higher order, so CNNs can cope with various situations, the theory of image with zoom, rotation and translation invariance.

LeCun has used the back-propagation algorithm to train multi-layer neural networks to identify handwritten postcodes, which is the first modern CNN [Lecun, Bottou, Bengio et al. (1998)] in 1998. However, limited to the development of computing performance and data sets, it was not until 2012 that Krizhevsky et al. [Krizhevsky, Sutskever and Hinton (2012)] proposed a CNN to adapt to the large data set. On the millions of ImageNet data sets, the effect is much more than the traditional method, and the classification accuracy has risen from more than 70% to more than 80%.

Table 3: Architectures of typical convolutional neural networks

Features	AlexNet	VGG	Inception-v1	ResNet	Inception-v2	Inception-v4
1st release time	2012	2014	2014	2015	2015	2016
Layers	8	19	22	152	22	22
Top-5 error	16.4%	7.3%	6.7%	3.57%	4.8%	3.08%
Data Augmentation	✓	✓	✓	✓	✓	✓
Convolutional layers	5	16	21	152	21	21
Convolutional kernel size	11,5,3	3	7,1,3,5	7,1,3,5	7,1,3	7,1,3
Inception	✗	✗	✓	✗	✓	✓
Full connected layers	3	3	1	1	1	1
Full connected size	4096,4096, 1000	4096,4096,1000	1000	1000	1000	1000
Dropout	✓	✓	✓	✓	✗	✗
Local response normalization	✓	✗	✓	✗	✓	✓
Batch normalization	✗	✗	✗	✗	✓	✓

After that, because of the great performance, CNNs have attracted the attention of many researchers. Until the last ILSVRC in 2017, the image classification accuracy of the deep learning algorithm on ImageNet has approached or even surpassed that of humans.

Because of this, the ILSVRC competition is no longer continuing. Researchers will focus on more challenging projects, such as WebVision Challege [Ioffe and Szegedy (2015)].

Tab. 3 shows the best CNNs in the ILSVRC competition from 2012. It lists the new features of the successive generations of networks. As we can see from the table, more and more optimizations are applied into the network design, such as ReLU, Dropout, Inception, Local Response Normalization (LRN), Batch Normalization, etc. With the development of the network, not all of the optimization methods are applied to the latest algorithms. For example, not all the networks use the LRN method. The author of VGG believes that LRN does not play an optimization role in their network structure. However, the Inceptions [Szegedy, Vanhoucke, Ioffe et al. (2016); Szegedy, Ioffe, Vanhoucke et al. (2017); Girshick (2014)] all adopt the LRN. On the other hand, the Dropout method has always been considered to be able to effectively improve the network generalization ability and reduce overfitting. But after the first version of the Inception, the method has been abandoned. Because batch normalization in the Inceptions regularizes the model and reduces the need for Dropout. It can be either removed or reduced in strength in a batch-normalized network

The performance results of the above networks are based on the ImageNet data set in the ImageNet Large Scale Visual Recognition Competition (ILSVRC). However, the competition stopped after the 8 sessions from 2009 to 2017. The accuracy rate of identifying objects from the original algorithm is only 71.8% up to 97.3% of the present, and the error rate of recognition is far below the 5.1% of human. Although the ImageNet challenge has ended its short life cycle, ImageNet data set will continue to exist. Up to now, there are more than 13 million pictures and will grow in the future and continue to contribute to the field of computer vision. In the future, ImageNet will remain open for free use by researchers. Even if the ImageNet competition itself is over, its legacy will continue to affect the entire industry. Since 2009, dozens of newly developed datasets have introduced computer vision, neural language processing and speech recognition and other subdomains. ImageNet has changed researchers' thinking mode. Although many people still care about models, they are also concerned about data. Data redefine our way of thinking about models.

4 Two significant target detection ideas with CNNs

It is not enough to just classify images. Classification is the basic of computer vision. Object localization, object recognition, semantic segmentation, instance segmentation and key point detection are more hard and meaningful tasks. The classification task is concerned with the whole, given the content description of the entire picture, while the detection is focused on the specific object target, and it is required to obtain the category information and location information of this target at the same time. Compared with classification, target detection gives the understanding of the foreground and background of the image. The algorithm needs to separate the target of interest from the background and determine the description of the target, such as the category and location of the target. Therefore, the output of the target detection model is a list. Each item of the list uses a data group to give the category and position of the target, which is commonly represented by the coordinates of the rectangular detection frame.

Target detection research has been conducted for many years, and there are many methods that have been widely recognized and applied in the industry, such as Jones et al. [Jones and Viola (2001); Zhu, Yeh, Cheng et al. (2006)]. But most classic methods are very dependent on finding effective features which is hard to be found in a common simple method. The introduction of CNNs changed the main research ideas in the field of target detection. It frees researchers from complex feature engineering. In the next part, several typical target detection algorithms are introduced in the field of target detection with the help of deep learning.

At present, there are two main ideas in the field of target detection based on deep learning. One is the RCNN series method based on the region proposal, and the other is the YOLO series algorithm based on the regression.

4.1 RCNNs: regions with CNN features

4.1.1 RCNNs: Regions with CNN features

The classical target detection algorithm uses a sliding window method to determine all possible areas in turn. Girshick [Girshick (2015)] extract a series of more likely candidate regions in advance with the method of selective search, and then the CNNs based extraction features are used only in these candidate regions for judgment. The RCNN algorithm can be divided into 4 steps:

1. Candidate region generation: An image generates 1K~2K candidate regions (using the Selective Search method).

2. Feature extraction: Using deep convolution network to extract feature (CNN) for each candidate region.

3. Category judgment: The feature is sent to each class of SVM classifier to distinguish whether it belongs to the class.

4. Location: Refinement using regressor fine correction candidate frame position.

The drawback of this method is that RCNN has the problem of repeated computation. There are thousands of regions in proposal, most of which overlap each other, and the overlapped parts will be repeatedly extracted from feature. To solve the problem, Kaiming He et al. [He, Zhang, Ren et al. (2014)] propose the SPP-Net to improve the RCNN.

4.1.2 SPP-net: spatial pyramid pooling in deep convolutional networks.

An image has 1~2 k candidate boxes in RCNN, and each one has to enter a CNN to do convolution. SPP-net proposes to extract the RoI (Region of Interest) features on the feature map, so that only a convolution is needed on the entire image. The overall process is similar to RCNN. But because all Z the features of the RoI are extracted directly from the feature map, the convolution operation is greatly reduced, and the efficiency is improved.

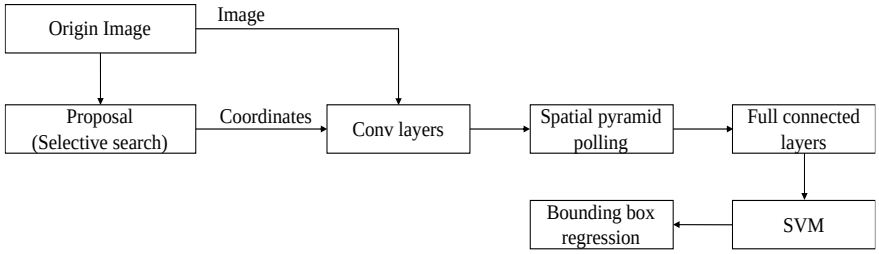

Figure2: SPP-net data flow graph

4.1.3 FAST and FASTER RCNN

Girshick et al. combined with SPP-net to improve the performance of RCNN networks [Girshick (2015); Ren, He, Girshick et al. (2015)]. The Faster RCNN even realize object recognition in video. Specifically, in Fast-RCNN, the author put bounding box regression into the internal neural network and classified it into region and became a multi-task model. The actual experiment also proved that these two tasks can share convolutional feature and promote each other. However, the performance can still be promoted. Faster-RCNN is an end-to-end CNN object detection model. The authors propose that the convolutional level features in network can be used to predict category dependent region proposal, and do not need to perform algorithms such as selective search in advance. The author integrates region proposal extraction and Fast-RCNN part into a network model. Although the training stage is still multi-step, the detection phase is very convenient and fast, and the accuracy rate is not much different from the original Fast-RCNN. At last, for the very deep VGG-16 model, the Faster-RCNN detection system has a frame rate of 5fps (including all steps) on a GPU in 2016.

4.1.4 Mask R-CNN

Mask-RCNN [He, Gkioxari, Dollar et al. (2017)] is a parallel detection and segmentation results, which means that the two results can be got at one time, but unlike the previous segmentation after do classification. The general framework of Mask-RCNN is still like the Faster-RCNN framework. It can be said that the fully connected subdivision network is joint after the basic feature network. The task of the network is from the original two tasks (classification+regression) to three tasks (classification+regression+segmentation).

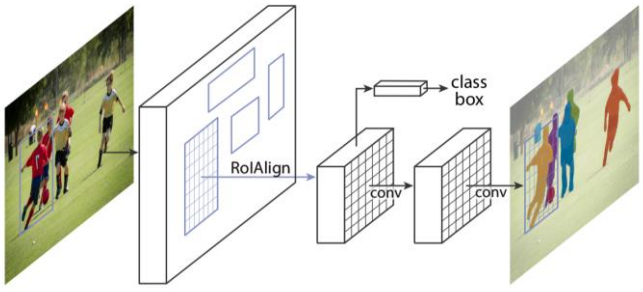

Figure 3: The Mask R-CNN framework for instance segmentation

Mask R-CNN is mainly divided into two stages:

Stage 1: Generate the candidate box area. This process is the same as Faster R-CNN, the RPN (Region Proposal Network) is used.

Stage 2: RoIPool is used in the candidate box area to extract features and to classify and border box regression, and a two-element mask is generated for each RoI.

In target detection, there are some errors in the given bounding box and the original graph, which does not affect the results of the classification detection. But in pixel level image segmentation, such spatial location error will seriously affect the segmentation results. Therefore, this network proposes the use of bilinear interpolation to solve this problem, that is, RoIAlign. The image is passed through RoIAlign instead of RoIPool, making the area of the feature map selected by RoIPool more accurately corresponding to the area of the original image.

Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework.

Although the detection accuracy of RCNNs is high, if it is applied to video information processing, the excessive calculation amount required by the algorithm makes it difficult to process the video information in real time in a single machine condition. The emergence of YOLOs has brought the target detection speed to a new level with the lowest loss of accuracy rate.

4.2 YOLO: you only look once

YOLO [Redmon, Divvala, Girshick et al. (2015)] is a convolutional neural network that can predict multiple Box locations and categories at a time. It can achieve end to end target detection and recognition, and its biggest advantage is fast speed. In fact, the essence of target detection is regression, so a CNN that implements a regression function does not require a complex design process. YOLO does not train the network in the way of selecting sliding windows or extracting proposal, but directly selecting the whole training model. The advantage of this way is that it can better distinguish the target and background area. In contrast, the Fast R-CNN trained by proposal often mistakenly checks the background area as a specific target. Of course, YOLO has sacrificed some precision while improving the speed of detection. Until now, the YOLO series of algorithms have developed three generations

4.2.1 YOLO Version 1: beginning of a regression-based target detection algorithm

YOLO [Redmon, Divvala, Girshick et al. (2015)] is the most popular object detection algorithm. It is fast and simple. As the name implies, this algorithm recognizes all objects by looking at the image only once. This algorithm can achieve real-time object detection, about 40 frames per second with the help of the Titan X GPU. The accelerated version of YOLO is almost 150 fps. YOLO reasons globally about the image when making predictions. It can also learn generalizable representations of objects which mean that its generalization ability is relatively strong. YOLO supports

end-to-end training, which will reduce a lot of unnecessary work, and the entire model is actually a convolutional neural network.

YOLO's computing process can be divided into the following steps:

Step 1: Divide the original image into an S×S grid.

Step 2: Each grid predicts B bounding boxes and confidence scores which can be represented by $(x, y, w, h, \Pr(Object) * IOU_{pred}^{truth})$, where $Pr(Object)$ represents the probability that the current position is an Object, IOU is the overlap probability between the predicted box and ground truth. The x, y is the center coordinate and the w, h is the size of the box.

Step 3: The probability of each grid prediction class $Ci = Pr(Classi|Object)$.

Step 4: When predicting, multiply the class conditional probability and confidence:

$$Pr(Class_i|Object) * Pr(Object) * IOU_{pred}^{truth} = Pr(Class_i) * IOU_{pred}^{truth}$$

Similar to R-CNN and DPM, when a large object is encountered, it is still necessary to perform non-maximally suppressed operations. Since B has a value of 2, that is, a grid will only return two boxes, and a grid will have only one category, so if there are multiple categories in a grid, there will be problems. YOLO has better results for images like small objects such as a flock of birds.

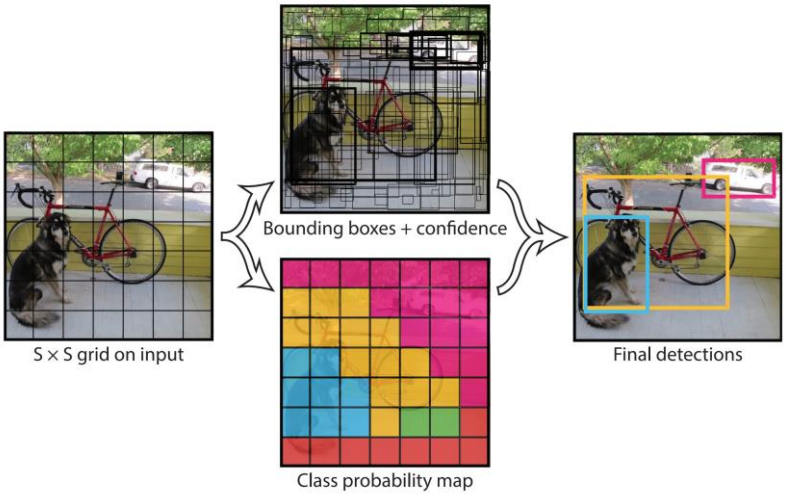

Figure 4: The YOLO models detection as a regression problem. It divides the image into an S×S grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an S×S×(B∗5+C) tensor

4.2.2 YOLO Version 2: a more comprehensive improvement

There are still many places in the original YOLO that can be improved, so the original author made many improvements in the first version and achieved remarkable results. In summary, there are two major improvements. Because the network can distinguish approximately 9000 objects under combining training with ImageNet and COCO, it is also called YOLO9000 [Redmon and Farhadi (2016)].

First, the author used a series of methods to improve the original YOLO multi-objective inspection framework. With the advantage of maintaining the original speed, the accuracy was improved. Using the VOC 2007 data set test, the mAP (Mean Average Precision) at 67 FPS reached 76.8%, and the mAP at 40 FPS reached 78.6%. The overall performance is comparable to Faster R-CNN and SSD.

Second, the author proposes a joint training method for target classification and detection. Through this method, YOLO9000 can simultaneously train in COCO and ImageNet. The trained model can achieve real-time detection of up to 9000 objects.

Some specific improvements are as follows: YOLOv2 adds batch normalization behind the volume base layer, removes the dropout layer, and increases its mAP by 2%. During network training, the network is changed from 224*224 to 448*448, and in order to ensure that there are only an odd number of positioning positions in the feature map, only one central cell is guaranteed, and the network is finally set to 416*416. Finally achieved 4% mAP increase. In addition, previous YOLO uses the data of the full connection layer to complete the prediction of the border, resulting in the loss of more spatial information and inaccurate positioning. In this version, the author draws on the anchor idea of Faster R-CNN, removes the full connectivity layer from the network, and combines Dimension Clusters and Direct location prediction to improve the mAP by 5%.

In terms of training, unlike the method of fixing the picture size of the input network, the author finetunes the network after several iterations. Every 10 epochs, new image sizes are randomly selected. The down-sampling parameter used by the YOLO network is 32, then a scaled multiple of 32 is used for {320, 352...608}. The final minimum size is 320*320 and the largest size is 608*608. Then adjust the network according to the input size for training. This mechanism allows the network to better predict pictures of different sizes, meaning that the same network can perform detection tasks with different resolutions. YOLOv2 runs faster on small pictures, achieving a balance in speed and accuracy.

4.2.3 YOLO Version 3: the latest improvement

The YOLOv3 model is much more complex than the previous model and can be weighed against the speed and accuracy by changing the size of the model structure. It improved multi-scale prediction, and better basic classification networks and classifiers. YOLOv3 does not use Softmax to classify each box. Instead, it is replaced by multiple independent logistic classifiers, and the accuracy does not decrease.

4.3 General comparison of RCNNs and YOLOs

Compared to the RCNNs methods, the YOLO series methods have the following advantages:

- High speed as previous introduction.
- False positive rate of background is low.
- Versatility. YOLO also applies to object detection in artistic works. Its detection rate for non-natural image objects is much higher than DPM and RCNN series detection methods.

Tab. 4 lists the main features of RCNNs and YOLOs. In general, from R-CNN, SPP-NET, Fast R-CNN to Faster R-CNN, and from YOLOv1 to YOLOv2, the detection process based on deep learning has become increasingly streamlined and accurate. The speed is getting faster and faster. It can be said that the R-CNN series target detection methods based on the region proposal and the YOLO series target detection methods based on the regression are the two main branches in the current target detection field.

The target detection, such as RCNNs and YOLOs, only gives the positions and labels of the objects in the image. However, many times, there is a need to detect the edges of objects and give relevant descriptions of the objects at the same time. The re-search of object segmentation is to achieve this goal.

Table 4: Features of RCNNs and YOLOs

RCNN	• Extracting 2000 bottom-up region proposals using the selective-search • A large CNN network computing feature for each region proposal • Classifying each region proposal using a linear SVMs classifier • Regression analysis to adjust the region area
Fast-RCNN	• Read the entire picture and a set of RoIs as input • Extract features from the entire image to get the feature map with convolutional network • For each RoI region, the pooling layer extracts a fixed-size feature factor from the feature map • Feature factor is sent to full connected layer and mapped to two parts. One part is to evaluate k target classes and another part to generate bounding box regressor
Faster-RCNN	• For the entire picture, use CNN for feature map • Perform full-connected operations on feature maps using RPN networks and get the feature information of the candidate box • Use the classifier to determine whether a feature belongs to a particular class for the features extracted in the candidate frame • For a candidate box belonging to a certain feature, use a regression to further adjust its position
YOLOv1	• Dividing the input image into S×S grids, each grid is responsible for detecting objects that fall into it • If the center of the object falls into the grid, the grid is responsible for detecting the object • Each grid outputs the number of bounding boxes and the number of objects belonging to a class of confidence.
YOLOv2	• Add Batch Normalization to avoid overfitting • Remove a pooling layer to increase the resolution of the convolutional output • Use K-means to automatically select the best initial boxes • Feature map of 26×26×512 turned into 13×13×2048 compared to YOLOv1

5 Pixel-level object segmentation

Object detection only gives the locations and labels of the objects, which is not specific enough. It is more difficult to separate out all the pixels related to the object and give the

categories. This operation is called object segmentation. Object segmentation includes semantic segmentation and instance segmentation. The former is an extension of the pre-background segmentation. It requires the separation of image parts with different semantics, while the latter is an extension of the detection task and requires the outline of the objects, which is more refined than the detection frame. Object segmentation is a pixel-level description of an image. It gives each pixel category meaning and is suitable for understanding demanding scenes, such as the segmentation of roads and non-roads in auto pilot, geographic information system, and medical image analysis, etc.

5.1 Semantic segmentation

Before deep learning was developed, the semantic segmentation method was diverse, and the effect levels were uneven, such as thresholding methods, clustering-based segmentation methods, graph partitioning segmentation methods and even the pixel-level decision tree classification [Shotton, Johnson and Cipolla (2008); Shotton, Fitzgibbon, Cook et al. (2011); Shi and Malik (1997); Rother, Kolmogorov and Blake (2004)]. After computer vision entered the era of deep learning, semantic segmentation also entered a new stage of development. A series of semantic segmentation networks based on convolutional neural networks represented by Fully Convolutional Networks (FCNs) are proposed and repeatedly refresh the semantic segmentation accuracy of images.

5.1.1 Full convolutional neural network (FCN) for semantic segmentation

The idea of the FCN [Long, Shelhamer and Darrell (2015)] is very intuitive. It directly performs pixel-level end-to-end semantic segmentation. It can be implemented based on the mainstream deep convolutional neural network model. It reuses ImageNet's pre-training network for se-mantic segmentation and uses deconvolutional layer up-sampling. At the same time, it also introduced a skip connection to improve the up-sampling of coarse pixel positioning.

The SegNet [Badrinarayanan, Kendall and Cipolla (2017)] improves use of encoder and decoder. It applies the result of the pooling layer to the decoding process. With other improvements, the segmentation accuracy is slightly better than that of the FCN, and the overall efficiency is slightly higher than that of the FCN.

5.1.2 Dilated convolutions

One deficiency of the FCN is that due to the presence of the pooling layer, the size of the tensor, i.e., the length and width, becomes smaller and smaller. However, the original design of the FCN required an output that was consistent with the input size, so the FCN did an up-sampling. But up-sampling cannot retrieve lost information completely without loss.

Dilated Convolution [Yu and Koltun (2015)] is a good solution for this. Since pooled down-sampling operations can result in information loss, the pooling layer is removed directly. However, the removal of the pooling layer will bring about a smaller receptive field in each layer of the network, which will reduce the prediction accuracy of the entire model. The main contribution of Dilated Convolution is to remove the down-sampling operation of the pool without reducing the receptive field of the network.

5.1.3 Following-up development

After this, most semantic separation networks are inseparable from the FCN architecture. The DeepLab [Chen, Papandreou, Kokkinos et al. (2018); Chen, Papandreou, Schroff et al. (2017)] also uses Dilated Convolutions and fully connected CRF added and proposes atrous spatial pyramid pooling. Dilated Convolution has several disadvantages, such as large amount of computation and a large amount of memory. Therefore, RefineNet [Lin, Milan, Shen et al. (2017)] designs the Encoder-Decoder architecture with well thought-out decoder blocks and all the components follow residual connection design which reduces the computing requirements. Besides, in the recent years, more and more improvements are emerged, such as PSPNet [Zhao, Shi, Qi et al. (2017)], Large Kernel Matters [Peng, Zhang, Yu et al. (2017)] and DeepLab v3 [Chen, Papandreou, Schroff et al. (2017)]. The semantic segmentation is developing fast. However, there are still many challenges waiting to be resolved.

Semantic segmentation technology based on deep learning can achieve a segmentation effect that is faster than traditional methods, but its requirement for data annotation is too high. Not only does it require huge amounts of image data, they also need to provide Semantic labels that are accurate to the pixel level. Therefore, more and more researchers have begun to turn their attention to the problem of image semantic segmentation under Weakly-supervised conditions. In this kind of problem, the image only needs to provide the image level annotation, and the semantic segmentation accuracy comparable to the existing method can be obtained without the need of expensive pixel level information.

5.2 Instance segmentation

Instance segmentation is a complex of object detection and semantic segmentation. It can both detect the object, give its bounding box, and it can be segmented to the edge of the object. Relative semantic segmentation, instance segmentation can label different individuals of the same type of object on the picture. Therefore, instance segmentation is a very comprehensive problem that combines object detection, semantic segmentation and image classification.

The MASK R-CNN and FCIS [Li, Qi, Dai et al. (2017)] are the most significant research results in the past two years. The MASK R-CNN has been introduced in section 4.1. In fact, this method can effectively detect the simultaneous occurrence of each target and generate a high-quality segmentation mask for each instance. Mask R-CNN has a simple and straightforward idea: For Faster R-CNN, it has two outputs for each target object. One is the class label and the other is the bounding-box offset. Based on this, the Mask R-CNN method adds the output of the third branch: the object mask. The difference between the object mask and the existing class and box output is that it requires a finer refinement of the spatial layout of the object.

FCIS solves the problem of instance segmentation through a multitasking network. It inherits all the merits of FCNs for semantic segmentation [Dai, He, Li et al. (2016)] and instance mask proposal First, primary features are first extracted through a convolutional neural network, and a RoI (Region of Interest) is proposed. Then, for each RoI region, corresponding features are extracted by RoI Warping and RoI Pooling. Next, use the full connected layer to perform the foreground and background division. Finally, use the full

connected layer for image classification for each RoI.

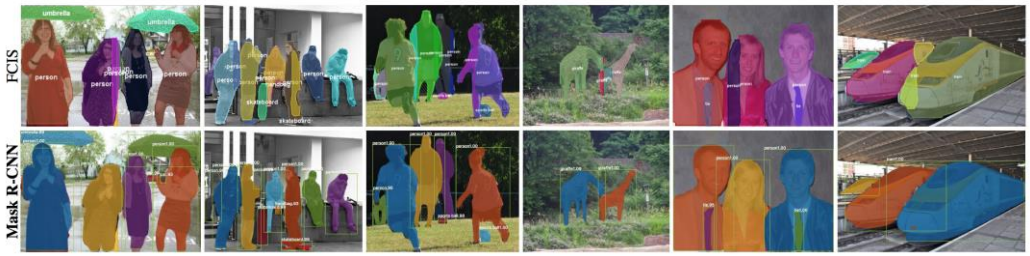

Figure 5: FCIS *vs.* Mask R-CNN. FCIS exhibits systematic artifacts on overlapping objects, suggesting that it is challenged by the fundamental difficulty of instance segmentation. Mask R-CNN shows no such artifacts

There are three shortcomings of the instance-based partitioning architecture based on proposal. First, if two objects share the same or similar boundaries, the algorithm cannot accurately identify them, especially for low-fill-rate linear objects such as windmills and windows. Second, there is no solution in the architecture that can prevent two instances from sharing pixels. Finally, the number of identifying instances is usually limited by the number of proposals that the network can handle. Some researchers took other ideas to avoid the above problems, such as the idea of instance embedding. Each pixel in the network output is a point in the embedding space. Points belonging to the same object are relatively close in the embedding space, while points belonging to different classes are far apart in the embedding space. Each pixel in the network output is a point in the embedded space [De Brebandere, Neven and Van Gool (2017); Fathi, Wojna, Rathod et al. (2017); Kong and Fowlkes (2017)]. The drawback is that compared to the methods based the idea of proposal, the results of these methods are not as good as Mask R-CNN or FCIS at present.

6 Conclusion

This paper has briefly reviewed typical deep convolutional neural networks in recent years and compared their differences and similarities. Effective network structures and optimization methods in these convolutions neural network is summarized. Besides, target detection and object segmentation algorithms based on deep convolution neural network are also summarized. These models have been becoming or will be recognized as new hotspots in deep learning and convolutional neural networks to effectively solve problems in computer vision, multi-object classification and/or relevant fields in these years and are therefore recognized as effective methods and/or de-facto tools in the industry and academia.

Besides the performance of the algorithm, however, most algorithms still highly rely on their own training datasets, which directly determines the functionality and performance of the algorithms. In that case, careful preparation of a dataset becomes critical and even tricky sometimes, so that it is imminent to develop migration learning methods based on deep learning.

Acknowledgment: This work has received funding from 5150 Spring Specialists (05492018012), the European Union Horizon 2020 research and innovation programme under the Marie Sklodowska Curie grant agreement no.701697, Major Program of the National Social Science Fund of China (Grant No.17ZDA092), the PAPD fund, and 333 High-Level Talent Cultivation Project of Jiangsu Province (BRA2018332).

References

Badrinarayanan, V.; Kendall, A.; Cipolla, R. (2017): Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481-2495.

Chen, L. C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A. L. (2018): DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834-848.

Chen, L. C.; Papandreou, G.; Schroff, F.; Adam, H. (2017): Rethinking atrous convolution for semantic image segmentation. *Computer Vision and Pattern Recognition.*

Dai, J.; He, K.; Li, Y.; Ren, S.; Sun, J. (2016): Instance-sensitive fully convolutional networks. *European Conference on Computer Vision*, pp. 534-549.

De Brabandere, B.; Neven, D.; Van Gool, L. (2017): Semantic instance segmentation with a discriminative loss function. arXiv:1708.02551v1.

Deng, J.; Dong, W.; Socher, R.; Li, L. J.; Li, K. et al. (2009): ImageNet: a large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248-255.

Fathi, A.; Wojna, Z.; Rathod, V.; Wang, P.; Song, H. O. et al. (2017): Semantic instance segmentation via deep metric learning. *Computer Vision and Pattern Recognition.*

Gatys, L. A.; Ecker, A. S.; Bethge, M. (2015): Image style transfer using convolutional neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414-2423.

Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. (2014): Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580-587.

Girshick, R. (2015): Fast R-CNN. *IEEE International Conference on Computer Vision*, pp. 1440-1448.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D. et al. (2014): Generative adversarial nets. *International Conference on Neural Information Processing Systems*, vol. 3, pp. 2672-2680.

He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. (2017): Mask R-CNN. *IEEE International Conference on Computer Vision*, pp. 2980-2988.

He, K.; Zhang, X.; Ren, S.; Sun, J. (2016): Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778.

He, K.; Zhang, X.; Ren, S.; Sun, J. (2014): Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 9, pp. 1904-1916.

Hegde, G.; Ramasamy, N.; Kapre, N. (2016): CaffePresso: An optimized library for deep learning on embedded accelerator-based platforms. *International Conference on Compliers, Architectures, and Sythesis of Embedded Systems*, pp. 14.

Hinton, G.; Vinyals, O.; Dean, J. (2015): Distilling the knowledge in a neural network. *Computer Science*, vol. 14, no. 7, pp. 38-39.

Ioffe, S.; Szegedy, C. (2015): Batch normalization: accelerating deep network training by reducing internal covariate shift. *OALib Journal*.

Jones, M. J.; Viola, P. (2001): Robust real-time object detection. *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137-154.

Kong, S.; Fowlkes, C. (2017): Recurrent pixel embedding for instance grouping. *Computer Vision and Pattern Recognition*.

Krizhevsky, A.; Sutskever, I.; Hinton, G. E. (2012): Imagenet classification with deep convolutional neural networks. *International Conference on Neural Information Processing Systems*, vol. 60, pp. 1097-1105.

LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E. et al. (1989): Backpropagation applied to handwritten zip code recognition. *Neural Computation*, vol. 1, no. 4, pp. 541-551.

Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. (1998): Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324.

Li, W.; Wang, L.; Li, W.; Agustsson, E.; Berent, J. et al. (2017): Webvision challenge: Visual learning and understanding with web data. *Computer Vision and Pattern Recognition.*

Li, X.; Zhang, G.; Huang, H. H.; Wang, Z.; Zheng, W. (2016): Performance analysis of GPU-based convolutional neural networks. *International Conference on Parallel Processing*, pp. 67-76.

Li, Y.; Qi, H.; Dai, J.; Ji, X.; Wei, Y. (2017): Fully convolutional instance-aware semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2359-2367.

Lin, G.; Milan, A.; Shen, C.; Reid, I. (2017): RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition.*

Lin, T. Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P. et al. (2014): Microsoft COCO: common objects in context. *European Conference on Computer Vision*, pp. 740-755.

Long, J.; Shelhamer, E.; Darrell, T. (2015): Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431-3440.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J. et al. (2015): Human-level control through deep reinforcement learning. *Nature*, vol. 518, no. 7540, pp. 529-533.

Nurvitadhi, E.; Sheffield, D.; Sim, J.; Mishra, A.; Venkatesh, G. et al. (2017): Accelerating binarized neural networks: comparison of FPGA, CPU, GPU and ASIC. *International Conference on Field-Programmable Technology*, pp. 77-84.

Peng, C.; Zhang, X.; Yu, G.; Luo, G.; Sun, J. (2017): Large kernel matters-improve semantic segmentation by global convolutional network. *IEEE Conference on Computer Vision and Pattern Recognition*, 1743-1751.

Redmon, J.; Farhadi, A. (2016): YOLO9000: Better, faster, stronger. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6517-6525.

Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. (2016): You only look once: unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788.

Ren, S.; He, K.; Girshick, R.; Sun, J. (2015): Faster R-CNN: towards real-time object detection with region proposal networks. *International Conference on Neural Information Processing Systems*, vol. 39, pp. 91-99.

Rother, C.; Kolmogorov, V.; Blake, A. (2004): Grabcut: interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309-314.

Shi, J.; Malik, J. (2000): Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905.

Shotton, J.; Fitzgibbon, A.; Cook, M.; Sharp, T.; Finocchio, M. et al. (2011): Real-time human pose recognition in parts from single depth images. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1297-1304.

Shotton, J.; Johnson, M.; Cipolla, R. (2008): Semantic texton forests for image categorization and segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L. et al. (2016): Mastering the game of go with deep neural networks and tree search. *Nature*, vol. 529, no. 7587, pp. 484-489.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A. et al. (2017): Mastering the game of go without human knowledge. *Nature*, vol. 550, no. 7676, pp. 354-359.

Simonyan, K.; Zisserman, A. (2014): Very deep convolutional networks for large-scale image recognition. *Computer Science*.

Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. (2017): Revisiting unrea-sonable effectiveness of data in deep learning era. *IEEE International Conference on Computer Vision*, pp. 843-852.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. A. (2017): Inception-V4, inception-resnet and the impact of residual connections on learning. *AAAI*, vol. 4, pp. 12.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S. et al. (2015): Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. (2016): Rethinking the inception architecture for computer vision. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818-2826.

Tsung, P. K.; Tsai, S. F.; Pai, A.; Lai, S. J.; Lu, C. (2016): High performance deep neural network on low cost mobile GPU. *IEEE International Conference on Consumer Electronics*, pp. 69-70.

Van Den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O. et al. (2016): Wavenet: a generative model for raw audio. *Sound*.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M. et al. (2016): Google's neural machine translation system: bridging the gap between human and machine translation. *Computation and Language*.

Yu, F.; Koltun, V. (2015): Multi-scale context aggregation by dilated convolutions. *OALib Journal*.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. (2017): Pyramid scene parsing network. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2881-2890.

Zhu, Q.; Yeh, M. C.; Cheng, K. T.; Avidan, S. (2006): Fast human detection using a cascade of histograms of oriented gradients. *IEEE Computer Vision & Pattern Recognition*, vol. 2, pp. 1491-1498.