EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

A Review of Convolutional Neural Network Development in Computer Vision

Hang Zhang[1,*]

[1]School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan 454000, P R China

Abstract

Convolutional neural networks have made admirable progress in computer vision. As a fast-growing computer field, CNNs are one of the classical and widely used network structures. The Internet of Things (IoT) has gotten a lot of attention in recent years. This has directly led to the vigorous development of AI technology, such as the intelligent luggage security inspection system developed by the IoT, intelligent fire alarm system, driverless car, drone technology, and other cutting-edge directions. This paper first outlines the structure of CNNs, including the convolutional layer, the downsampling layer, and the fully connected layer, all of which play an important role. Then some different modules of classical networks are described, and these modules are rapidly driving the development of CNNs. And then the current state of CNNs research in image classification, object segmentation, and object detection is discussed.

Keywords: Convolutional Neural Networks, computer vision, deep learning, IoT.

Received on 08 March 2022, accepted on 26 March 2022, published on 13 April 2022

doi: 10.4108/eetiot.v7i28.445

*Corresponding author. Email: zh@home.hpu.edu.cn

1. Introduction

CNNs are prevalent in deep learning architecture. Local awareness and parameter sharing are features of the convolutional structure, which can lower the model's complexity and the number of parameters. CNNs are also a very flexible machine learning model containing multi-level nonlinear transformations, and their design is inspired by the way the animal visual cortex is organized. After being inspired, CNNs have developed rapidly, solved many difficult problems in the field of artificial intelligence in the past, have strong robustness and fault tolerance [1], and are easy to train and optimize. In practical applications, at the same time, it is also combined with the Internet of Things (IoT) to improve the accuracy of face recognition. In terms of medical treatment, the IoT technology can be used to obtain data, load data, and put it into the convolutional model, which can complete the intelligent management of people and things.

In 1998, LeCun proposed LeNet-5 [2] network, which designed local receptive field, shared weight, and downsampling is designed to keep the translation, scale, and distortion invariance of handwritten digits [3-5]. In small-scale handwritten digit recognition, the system performed well. In 2012, in the ImageNet competition, the new model AlexNet [3] won the best performance in the image classification challenge competition, which attracted wide attention. Compared with the LeNet network, AlexNet network has a complicated design, as ReLU (Rectified Linear Unit) is built as a nonlinear activation function, using Dropout randomly inactivating neurons to address the issue of parameters as well as prevent network overfitting. With the success of AlexNet, researchers continue to improve on this basis to improve its performance, among which the representative architecture ZFNet [4], NIN [6], and VGGNet [5], GoogLeNet [7], ResNet [8]. With the development of this architecture, although these networks continue to improve the accuracy of ImageNet classification tasks, they also

lead to the trend of wider, deeper, and more complex networks. While this has helped networks get better feature extraction, researchers have also observed that these networks are increasingly hardware demanding and difficult to train. Researchers have proposed various improvements to address these issues.

All in all, the combination of the Internet of Things and the convolutional network is getting closer and closer, and the future life and the development of science and technology are inseparable from the organic combination of the two. Next, we will explain the knowledge about convolutional neural networks from several aspects.

2. CNNs Architecture

CNNs take the original image as the network input [9-11]. After the simple transformation of the data, it carries out a series of operations such as convolution, pooling, and nonlinear activation function mapping and abstracts, the original image layer by layer into the final feature [12] representation needed by its task. Finally, it ends with the linear mapping from the feature to the task target. Although there are many variants of CNNs, their structures are very similar [13-15], generally composed of the input layer, convolutional layer, pooling layer, the fully connected layer, and output layer. Figure 1 depicts the LeNet network.

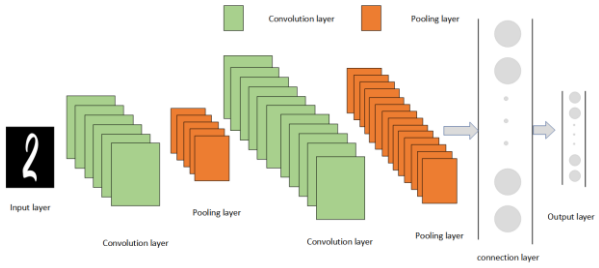

Figure 1. LeNet Network example

2.1 Convolutional Layer

As the first layer of image processing, the convolutional layer aims to learn the feature representation of the input image. The convolutional layer consists of multiple filters that map different features [16, 17]. In a convolutional neural network, an element in the output of a certain layer is determined when the region size of the corresponding input layer is called the receptive field. The new feature mapping can convolution the input with the learning filter [18], and then apply the nonlinear activation function to the convolutional result to obtain the output result [19-21]. The filters in the low convolutional layer are used to detect low-order features such as linear textures at edges and corners. But the filters in the high layer are used to learn abstract and more specific features [22-24]. By stacking multiple convolutional layers, the network model can gradually extract higher-level feature representation. Figure 2 illustrates the convolution operation.

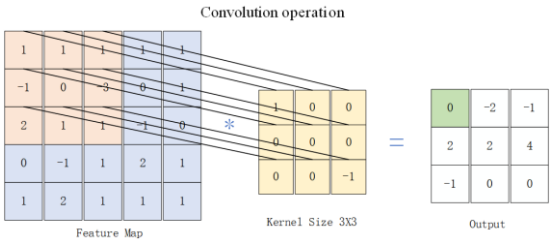

Figure 2. convolution operation

The preceding layer's feature map is convolution using a learnable convolutional kernel, and further output feature map is then generated using an activation function. The values of numerous feature maps can be convoluted using every output feature map. The calculation process is as follows(1)(2).

$$x_j^l = f\left(u_j^l\right) \tag{1}$$

$$u_j^l = \sum_{i \in M_j}^{n} x_i^{l-1} * k_{ij}^l + b_j^l \tag{2}$$

Among them, the net activation of the jth channel of the convolutional layer l is referred to as u_j^l, this is created by convolution and summation of the preceding layer's output feature map x_i^{l-1} plus the bias, and x_j^l is the output of the jth channel of the convolutional layer l, $f(\cdot)$ called the activation function, and sigmoid or ReLU are examples of activation functions are frequently used. The M_j represents the subset of input feature maps used to calculate u_j^l, k_{ij}^l is the convolutional kernel matrix, and b_j^l is the bias of the feature map after convolution. The convolutional kernel k_{ij}^l corresponding to each input feature map x_i^{l-1} may differ form an output feature map x_j^l, and "*" is the convolutional representation.

2.2 Pooling Layer

Pooling layers are in the middle of consecutive convolutional layers. The feature map of the pooling layer corresponds to the upper network layer [25, 26], that is, the pooling operation does not change the number of feature maps. The traditional network is often too large when processing images, which is inconvenient to process. The introduction of a pooling layer to reduce the size of the intermediate parameter matrix is to prevent overfitting. In addition, the pooling layer can maintain the translation and rotation invariance of convolution, that is, rotate or translate the image, and can also extract image features and improve the model's ability to flourish [27-29]. Commonly used pooling methods include max pooling, which takes the point with the largest value in the local receptive field, mean pooling, which averages all parameters in the local receptive field, and stochastic pooling, which randomly takes a value from the value in the local receptive field [30-32].

In addition, there are pooling methods such as mixed pooling and spatial pyramid pooling. The function of pooling is to reduce the size of the model. We know that

the amount of information contained in an image is huge, and the features are also very large. However, some information is not very useful for us to identify key features. Therefore, it is necessary to compress a relatively complex large matrix into a relatively small matrix through pooling to reduce its complexity to improve the operation speed and improve the robustness of the model to deal with more complex problems. [33-35]. The max pooling is shown in Figure 3

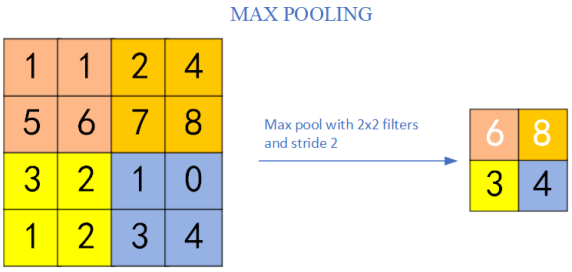

Figure 3. Max Pooling

Average pooling can be regarded as a structural regularization, which can improve the consistency between feature surfaces and categories [36-38]. There are no parameters that need to be optimized in the globally averaged sampling layer, so overfitting can be avoided. Furthermore, the globally averaged sampling layer sums the spatial information and is, therefore, more robust to spatial variations of the input. The average pooling is shown in Figure 4.

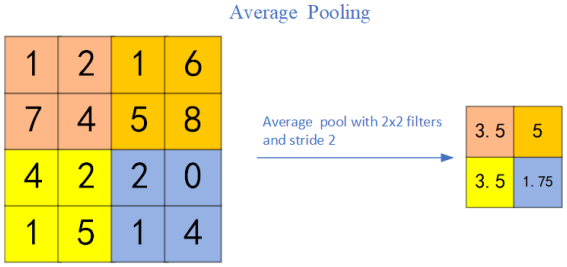

Figure 4. Average Pooling

2.3 Fully Connected Layer

We know that the convolution operation generates local features and that the fully connected layer's purpose is to add up the prior local features before generating the classification result [39, 40]. It's the same as translating the feature space of many distinct local representations previously learned to the machine's sample feature space [41-43] which is convenient for handing over to the final classifier or regression. The calculation formula is(3) (4).

$$x^l = f\left(u^l\right) \tag{3}$$

$$u^l = w^l x^{(l-1)} + b^l \tag{4}$$

Where u^l is called the net activation of the fully connected layer l, which includes by weight and bias the output feature map $x^{(l-1)}$ of the previous layer, w^l is the weight coefficient of the fully connected network, and b^l is the bias of the fully connected layer l, $f(\cdot)$ is called the activation function.

2.4 Activation Function

In deep CNNs, the activation function is a critical component. The traditional network can only cope with some linearly separable issues when there is no activation function. The nonlinear activation function is presented, which is useful for boosting the model's resilience, increasing its nonlinear expression ability, and eliminating difficulties like gradient disappearance [44-46].

2.4.1 Sigmoid

Sigmoid the function formula is(5). Mapping the function to (0~1).

$$\sigma(x) = \frac{1}{1+e^x} \tag{5}$$

The Figure 5 is shown in Sigmoid activation function.

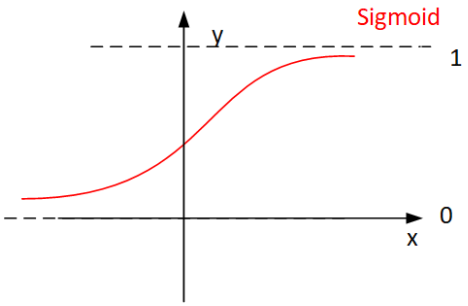

Figure 5. Sigmoid activation function

2.4.2 ReLU

ReLU is the most frequently used activation function in various models so far. Specifically, when the input is negative, the ReLU function's output is 0; when the input is positive, the ReLU function's output is x. Compared with Sigmoid, ReLU, the convergence speed of the model is faster, and it is more beneficial to the gradient update of backpropagation [47-49]. At the same time, the function of neurons in the hidden layer is set to 0, which brings sparseness and makes it easy for the network to obtain sparse representation, reduce the number of parameters [50], and reduce overfitting. [51, 52] Experiments show that ReLU has better performance than Sigmoid, and can be better to solve the gradient vanishing problem. The function formula of the ReLU is(6).

$$f(x) = \max(0, x) \tag{6}$$

The ReLU activation function is shown in Figure 6.

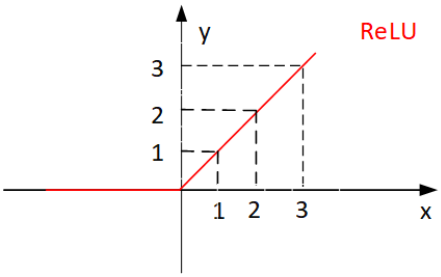

Figure 6. ReLU activation function

3. CNNs Improvement Block

3.1 Network in Network

The standard CNN is generally connected by linear convolutional layer, pooling layer, and the fully connected layer. The convolutional layer carries on the linear convolutional operation through the filter, then uses the nonlinear activation function to process the convolutional result, and finally generates the characteristic graph. Because the convolutional layer uses a linear filter, the acquired features have a strong linear representation [53, 54], so it is more suitable for learning linearly separable features and limits the task application scenarios to a great extent. However, the features of the sample to be extracted are generally highly nonlinear. For example, in face recognition, human ears, noses, and mouths all have different features. Therefore, Lin et al. [6] designed a Network in Network (NIN) model, whose main idea is to replace the traditional convolutional layer with a Multilayer Perceptron (MLP) composed of multiple the fully connected layers with nonlinear [55] activation functions. The Linear convolutional layer and MLP layer are depicted in Figure 7. So the nonlinear neural network is used to replace the linear filter, which enables it to approach more abstract representations of potential features and have stronger generalization ability.

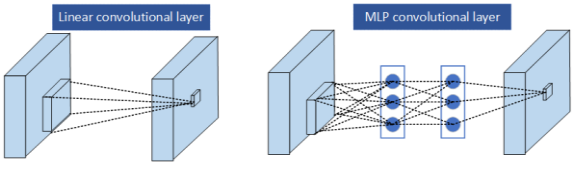

Figure 7. Linear convolutional layer and MLP layer

MLP is a nonlinear convolutional layer of the NIN structure, which replaces the original generalized linear model with MLP. NIN obtains feature maps of convolutional layers by sliding a miniature neural network through the input. Similar to the weight sharing of convolutional, MLP also shares all local receptive fields of the same feature surface, that is, the same for the same feature map MLP. The reason why NIN chooses MLP is that MLP uses the backpropagation algorithm for training and can be integrated with the CNNs structure [56]. At the same time, MLP is also a deep model with the idea of feature reuse.

In the traditional CNNs structure, the fully connected layers have too many parameters and are prone to overfitting [57], so it relies heavily on the dropout regularization technique. The NIN structure uses global average pooling to replace the original the fully connected layers, which greatly reduces the parameters of the model. It averages each feature surface of the last MLP convolutional layer through the global average pooling method and then concatenates these values into a vector, which is finally input into the softmax classification layer [58]. MLP convolutional layers can handle more complex nonlinear problems and extract more abstract features.

3.2 Inception Block and Improved Inception Block

Szegedy et al. [7] proposed an Inception model in 2014, which uses dimensionality reduction (1×1 convolution) to reduce the amount of computation and the cost of computation. Its main idea uses three small-scale filters of different sizes to extract feature information of different scales from the previous input layer, and then use this feature information and transmit it to the next layer. Inception has 1×1, 3×3, and 5×5 filtering among them, the 1×1 filter is mostly used for data dimensionality reduction, which may drastically reduce calculation time. This greatly improves the running speed of the code. Through the feature fusion of 4 channels, more useful features are extracted [59-61]. Szegedy et al. thought of a method to improve the accuracy of CNNs, which used decomposed convolutional and dimensionality reduction in the network [62-64]. The improved Inception model reduces the number of parameters and speeds up computation by replacing the 5×5 convolution in the Inception model with two 3×3 convolutions. The Inception and the improved Inception block are shown in Figure 8.

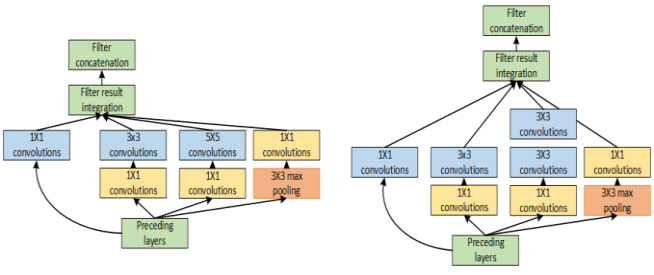

Figure 8. Inception block and improved Inception block

3.3 Residual Block and Improved Residual Block

With the development of CNNs, the depth and width of the network increase, the network will become more complex, and it is theoretically easier to fit complex feature representations. However, the network will face degradation problems and gradient exploding problems or gradient vanishing problems instead of overfitting problems [65, 66]. The specific performance is that the network performance no longer improves when depth deepens, and even when the network depth further increases, the model performance declines seriously [32, 67].

Residual block proposed by He et al. ResNet [8] is similar to Highway network [68], which also allows input information to spread across multiple hidden layers. The difference is that the threshold mechanism of the residual network is no longer learnable, that is, it always maintains a smooth state of information [69], which is extremely the number of hyperparameters is greatly reduced [70], the network convergence is accelerated, and a series of problems caused by network degradation is greatly reduced. The residual module is shown in Figure 9. The input of the residual module is defined as X, and the result is described as H(X)=F(X)+X, the residual is defined as F(X), and the network learns the residual F(X) during the training process, which is easier than directly learning the output H(X).

The proposal of residual networks marks a new stage in the development of convolutional neural networks. Residual blocks can be used to train deep network structures, and then a large number of studies have been carried out to improve residual structures [71]. By using the residual block to superimpose the depth, the accuracy is slightly improved, so the researchers tried to study the influence of the width on the network and found that the width is more important than the depth, and it is unnecessary to train a network with more than 50 layers [72], so there is currently a lot of research work to optimize the structure of the residual network from the network width. Zagoruyko et al. [73] think that ResNet cannot the fully feature reuse during training, which is manifested in that the gradient cannot flow through each Residual block during backpropagation, and there are only a few the residual module can learn useful feature representations.

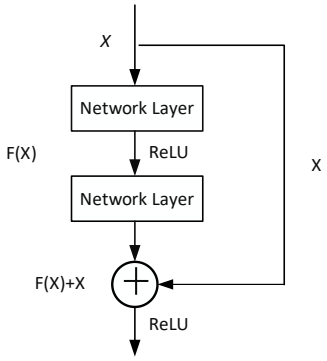

Figure 9. Residual block

The author proposes a Wide Residual Network (WRN) [73]. By widening the network width, reducing its depth, the training speed of WRN is increased by 2 times compared with the previous residual network, but the number of network layers is reduced by 50 times, which greatly reduces the amount of computation. Targ et al. [74] proposed a generalized residual network that combines the residual network and the standard convolutional neural network in parallel, removes invalid information while retaining the effective feature expression, improves the expressive ability of the network, and has a significant effect on the CIFAR-100 dataset. Zhang et al. [75] by adding additional bypass connections to the residual network and increasing the width to improve the learning ability of the network, the proposed Residual networks of residual networks (RoR) [76] can be used as a general module for constructing the network. Abdi et al. [77] experimentally support that the residual network is the hypothesis obtained by the fusion of several shallow networks, the model proposed by the author increases the number of residual functions in the residual module to improve the model's expressive capabilities.

3.4 DenseNet

In 2016, inspired by the idea of skip connections in ResNet [8], Huang et al. proposed a DenseNet [78] model. The model first uses forward propagation in the convolutional layer to connect each layer with other layers in the network and then uses the feature maps of all previous layers as the input of each subsequent layer to construct DenseNet [79, 80]. On popular image classification benchmarks, DenseNet can achieve comparable accuracy to ResNet in the ImageNet image classification competition, but it requires significantly fewer parameters [81, 82]. In addition, it improves the gradient vanishing problem, and at the same time, it strengthens the feature propagation process and promotes feature reuse through the reorganization of feature maps, reducing the amount of irrelevant computation. The DenseNet model structure is shown in Figure 10.

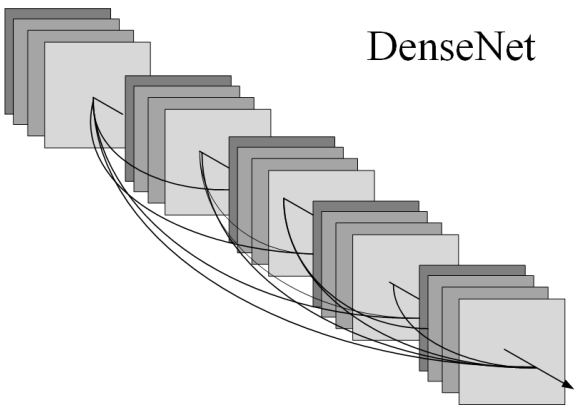

Figure 10. DenseNet block

3.5 Other Innovative Blocks

In the exploration of the design space of network structure, a great deal of effort has gone into fine-tuning module design, and a series of research results have been obtained. Such as to reduce the training parameters of the fully connected layer, NIN first proposed using global average pooling to replace the fully connected layer, which is equivalent to the whole connection layer. The network structure is regularized to prevent overfitting [83-85]. The global mean pooling establishes a connection between the feature map and the output category label, which is more interpretable than the fully connected layer, and then GoogLeNet adopts this structure to obtain performance improvement Huang et al. believe that the success of extremely deep networks comes from the introduction of bypass connections, and their proposed Dense block has direct connections between any two-layer network. For any network layer, its input comes from the output of all previous network layers. The output is used as the input of the following layers. This dense connection improves the propagation speed of the gradient and has a regularizing effect on the network, which makes the overfitting problem on small datasets optimized. Another advantage of dense connection is that it allows feature reuse. The trained DenseNet has a smaller number of parameters and is easier to train.

Traditional models like VGG models are too large to be enabled on lightweight devices. The lightweight network constructed by MobileNet [86] proposed by Howard et al. can be used on mobile embedded devices. Specifically, the traditional convolutional process is decomposed into two steps: depthwise convolutional and pointwise convolutional, which reduces the size of the model and the amount of computation. Sandler et al. combined the residual module with the depth-wise separable convolutional and Inverse residual with linear bottleneck are proposed, and the MobileNetv2 [87] constructed from this is superior to MobileNet in speed and accuracy. Zhang et al. further proposed ShuffleNet [75] of pointwise group convolution and channel shuffle on the basis of MobileNet, which achieved greatly improved in both image classification and target detection tasks. The Depthwise Convolutional block is shown in Figure 11.

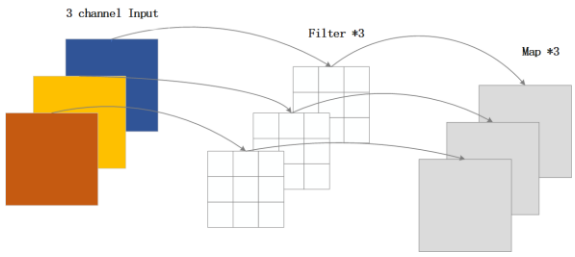

Figure 11. Depthwise Convolutional block

4. Applications of CNNs

4.1 Image Classification

Image classification is one of the core problems in image processing. Image classification refers to predicting the category of an image given an image.

In the last several years, CNNs have been widely used in the field of image processing. Krizhevsky et al. [3] used CNNs in the LSVRC-12 competition for the first time, they use the ReLU + Dropout technology for the first time, the depth of the CNNs model was deepened, and the best classification results were obtained at that time, which was called the AlexNet model. Compared with traditional CNNs, there is a great improvement. Because the nonlinear ReLU activation function is used in AlexNet, the ability of the model to deal with nonlinear functions is improved, the computational complexity of the model is reduced, and the training speed of the model is reduced. In addition, through dropout technology, some neurons are set to 0 at random throughout the training process, that is, some neurons in the middle are randomly inactivated in each cycle, the model has stronger robustness, and reduces the number of parameters of the fully connected layer to avoid overfitting [88].

Through the success of AlexNet, Szegedy et al. [7] try to increase the depth of CNNs, proposing a CNNs structure with more than 20 layers (called GoogLeNet). Convolution operations have been enhanced to three types (1*1, 3*3, and 5*5) that are used in the GoogLeNet structure. The main feature of this structure is that the convolution computation is reduced, the parameters are 12 times less, and the GoogLeNet The accuracy rate is higher, and it won the first place in the "specified data" group of image classification in LSVRC-14.

In the deep network model with very deep layers, in addition to the gradient vanishing problem and gradient exploding problem, there is also a degradation problem. Batch Normalization (BN) [89] is an efficient method to solve the gradient diffusion problem. The so-called degradation problem is: as the depth increases, the network accuracy will first rise and tend to saturate, and then rapidly decline. But the performance drop is not caused by overfitting, due to increasing the depth of the network so that its training error also increases. He et al. [8] adopt Residual Networks to solve the degradation problem. The main feature of ResNet is the cross-layer connection, which adds the input cross-layer pass and the convolution result by introducing Shortcut Connections. In other words, the unit's input is directly added to its output before being activated. Experiments show that the residual network can indeed solve the degradation problem of deep neural networks due to the depth of the network. ResNet enables the underlying network to be fully trained, the extracted shallow features are more abundant, and the accuracy is significantly improved with the deepening of the depth. When demonstrated in the LSVRC-15 competition using a deep ResNet with a depth of 152 layers, it achieved the 1st place result in image classification.

4.2 Object Detection

In the subject of computer vision, object detection has long been a major research focus [90-92], Its purpose is to locate the image target accurately and determine the target category. [93]. The use of CNNs for target detection can be traced back to the 1990s [94, 95]. However, due to the lack of training data and limited data processing capability of computing equipment, at that time, target detection based on CNNs developed very slowly before 2012 [96]. The great success of CNNs in the ImageNet challenge in 2012 re-inspired researchers interest [62] in based CNNs object detection, and it also led to the improvement of object detection accuracy. Object detection has also produced many more classical networks including R-CNN [97], OverFeat [88], Fast R-CNN [98], Faster R-CNN [99], FPN [100], and Mask R-CNN [101]. Nowadays, object detection has been widely used in security, military, transportation, and other fields.

Although the R-CNN [97] algorithm has achieved significant performance improvement in the object detection task, the CNNs feature extractor is executed for each candidate region, thus consuming a lot of computing time, resulting in high computational cost. The researchers in order to solve this problem proposed OverFeat [88]. OverFeat is the first time to use this model for multiple tasks. The characteristics of CNNs are fully utilized. First, the basic features are extracted from the image through CNNs, and then the basic features are extracted and assigned to different feature tasks. Because the weights are reused, the network propagation calculation is reduced. It solved the problem of long operation time. Later, Fast R-CNN [98] was introduced to improve the network by using the end to end training method. All convolutional layers can update parameters during fine-tuning, which improves the efficiency of code execution and improves the accuracy of detection.

4.3 Object Segmentation

Deep convolutional neural networks have been successfully applied to image detection, classification, and other tasks, many researchers have applied CNNs to the field of image segmentation.

The Fully Convolutional Network (FCN) proposed by Long et al. [102] at CVPR2015 can learn from end to end. Object classification results belong to pixel-level learning. Unlike classical CNNs architectures, traditional CNNs can only accept pixel inputs of fixed size. FCN replaces the traditional fully connected layer with a fully convolutional layer with a convolution kernel size of 1, which allows FCN to accept pixel inputs of any size. The network uses pooling and deconvolution operations to ensure input and output. With the same size, richer feature information is obtained by fusing the low-level features of the shallow network with the high-level semantic features of the deep network. That is, the semantic information from the deep convolutional layers is combined with the appearance information of the shallow convolutional layers through the residual connection structure to generate accurate and detailed image semantic segmentation. The schematic structure of FCN is shown in Figure 12.

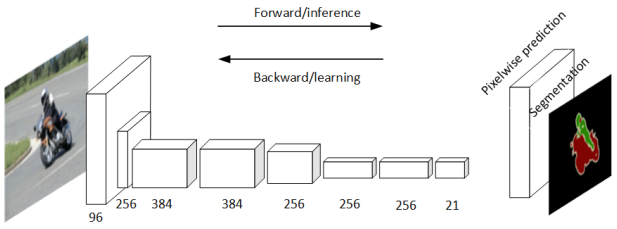

Figure 12. Architecture of FCN

5. Conclusion

Over the last few years, CNNs have made continuous breakthroughs in the field of computer vision. So far, the Internet of things (IoT) technology has penetrated our daily life, such as unmanned vending machines, driverless cars, smoke alarms, etc. Among them, the Internet of Things uses convolutional networks to develop related unmanned driving technologies, body temperature detection systems, and intelligent safety equipment, which has been widely used.

This paper expounds the modules in the classic model from the convolutional layer, pooling layer, and activation function, and finally summarizes some research progress of CNNs in image classification, object detection, and object segmentation is presented. Although convolutional neural networks have been widely used, there is still room for exploration in computer vision and other fields [103].

First, since CNNs architectures are getting deeper and deeper, large-scale annotated datasets and huge computing power are required for training. Secondly, the current research on CNNs in computer vision is almost all supervised learning, so manual collection of labelled data sets requires a lot of manpower and financial resources, so it becomes particularly important for unsupervised learning exploration; At the same time, when testing, CNNs deep models need to take up a lot of video memory, and the training time is very long. Large networks sometimes require several months of training time, which makes them unsuitable for deployment on mobile platforms with limited resources. Reducing the complexity of the model and being able to run the model on the underlying device without loss of accuracy is very important for the development of convolutional neural networks [104].

Finally, choosing appropriate hyperparameters has always been a major obstacle to applying CNNs to new tasks, such as the size of the learning rate, the size of the convolution kernel, the selection of the stride and the number of convolutional layers, and the selection of the

optimizer. These hyperparameters have strong internal dependencies, and any small adjustment may cause the final training result a big impact [105, 106].

In general, IoT devices can generate a large amount of data, and convolutional networks can use data to analyse products and systems. In turn, new smart devices can be created. It can be said that there are still many things in the process of combining IoT and convolutional networks. As a possibility, related work on the future development of convolutional is constantly moving closer to the Internet of Things.

References

[1] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2892-2900.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, no. 11, pp. 2278-2324, 1998.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems,* vol. 25, 2012.

[4] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, 2014: Springer, pp. 818-833.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[6] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400,* 2013.

[7] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[9] S. Srinivas and R. V. Babu, "Data-free parameter pruning for deep neural networks," *arXiv preprint arXiv:1507.06149,* 2015.

[10] X. Liu, J. Pool, S. Han, and W. J. Dally, "Efficient sparse-winograd convolutional neural networks," *arXiv preprint arXiv:1802.06367,* 2018.

[11] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," *arXiv preprint arXiv:1405.3866,* 2014.

[12] H. Zhou, J. M. Alvarez, and F. Porikli, "Less is more: Towards compact cnns," in *European conference on computer vision*, 2016: Springer, pp. 662-677.

[13] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation," *Advances in neural information processing systems,* vol. 27, 2014.

[14] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," *Advances in neural information processing systems,* vol. 29, 2016.

[15] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems,* 2021.

[16] C. Hwang, D. Kim, and T. Lee, "Semi-supervised based Unknown Attack Detection in EDR Environment," *KSII Transactions on Internet and Information Systems,* vol. 14, no. 12, pp. 4909-4926, Dec 2020.

[17] H. Jung and B. G. Lee, "The Impact of Transforming Unstructured Data into Structured Data on a Churn Prediction Model for Loan Customers," *KSII Transactions on Internet and Information Systems,* vol. 14, no. 12, pp. 4706-4724, Dec 2020.

[18] V. Lebedev and V. Lempitsky, "Fast convnets using group-wise brain damage," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2554-2564.

[19] Y.-D. Lv, "Alcoholism detection by data augmentation and convolutional neural network with stochastic pooling," *Journal of Medical Systems,* vol. 42, no. 1, 2018, Art no. 2.

[20] C. Tang, "Twelve-layer deep convolutional neural network with stochastic pooling for tea category classification on GPU platform," *Multimedia Tools and Applications,* vol. 77, no. 17, pp. 22821-22839, 2018.

[21] C. Pan, "Abnormal breast identification by nine-layer convolutional neural network with parametric rectified linear unit and rank-based stochastic pooling," *Journal of Computational Science,* vol. 27, pp. 57-68, 2018.

[22] N. Altwaijry, "Keystroke Dynamics Analysis for User Authentication Using a Deep Learning Approach," *International Journal of Computer Science and Network Security,* vol. 20, no. 12, pp. 209-216, Dec 2020.

[23] M. Raveendra and K. Nagireddy, "Inter frame Tampering Detection based on DWT-DCT Markov Features and Fine tuned AlexNet Model," *International Journal of Computer Science and Network Security,* vol. 20, no. 12, pp. 1-12, Dec 2020.

[24] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 173, pp. 24-49, 2021.

[25] P. Murugeswari and S. Vijayalakshmi, "New Method of Internal Type-2 Fuzzy-Based CNN for Image Classification," *Int. J. Fuzzy Log. Intell. Syst.,* vol. 20, no. 4, pp. 336-345, Dec 2020.

[26] S. Joshi, R. Kumar, and A. Dwivedi, "Hybrid DSSCS and convolutional neural network for peripheral blood cell recognition system," *IET Image Processing,* vol. 14, no. 17, pp. 4450-4460, Dec 2020.

[27] C. Pan, "Multiple sclerosis identification by convolutional neural network with dropout and parametric ReLU," *Journal of Computational Science,* vol. 28, pp. 1-10, 2018/09/01/ 2018.

[28] C. Huang, "Multiple Sclerosis Identification by 14-Layer Convolutional Neural Network With Batch Normalization, Dropout, and Stochastic Pooling," (in English), *Frontiers in Neuroscience,* Original Research vol. 12, 2018-November-08 2018, Art no. 818.

[29] G. Zhao, "Polarimetric synthetic aperture radar image segmentation by convolutional neural network using graphical processing units," *Journal of Real-Time Image Processing,* vol. 15, no. 3, pp. 631-642, 2018.

[30] C. Vance *et al.*, "Learning to detect the onset of slow activity after a generalized tonic-clonic seizure," *Bmc Medical Informatics and Decision Making,* vol. 20, Dec 2020, Art no. 330.

[31] H. Sim and J. Lee, "Bitstream-Based Neural Network for Scalable, Efficient, and Accurate Deep Learning Hardware," *Frontiers in Neuroscience,* vol. 14, Dec 2020, Art no. 543472.

[32] H. K. Shin, S. H. Park, and K. W. Kim, "Inter-floor noise classification using convolutional neural network," *Plos One,* vol. 15, no. 12, Dec 2020, Art no. e0243758.

[33] K. Muhammad, "Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation," *Multimedia Tools and Applications,* vol. 78, no. 3, pp. 3613-3632, 2019.

[34] S.-H. Wang and J. Sun, "Cerebral micro-bleeding identification based on a nine-layer convolutional neural network with stochastic pooling," *Concurrency and Computation: Practice and Experience,* vol. 32, no. 1, p. e5130, 2020.

[35] A. K. Sangaiah, "Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization," *Neural Computing and Applications,* vol. 32, pp. 665-680, 2020.

[36] G. Antonelli, P. Gkolfakis, G. Tziatzios, I. S. Papanikolaou, K. Triantafyllou, and C. Hassan, "Artificial intelligence-aided colonoscopy: Recent developments and future perspectives," *World Journal of Gastroenterology,* vol. 26, no. 47, pp. 7436-7443, Dec 2020.

[37] R. Majji, P. G. O. Prakash, R. Cristin, and G. Parthasarathy, "Social bat optimisation dependent deep stacked auto-encoder for skin cancer detection," *Iet Image Processing,* vol. 14, no. 16, pp. 4122-4131, Dec 2020.

[38] N. Padmasini and R. Umamaheswari, "Automated detection of multiple structural changes of diabetic macular oedema in SDOCT retinal images through transfer learning in CNNs," *Iet Image Processing,* vol. 14, no. 16, pp. 4067-4075, Dec 2020.

[39] P. Sinthia and M. Malathi, "Cancer detection using convolutional neural network optimized by multistrategy artificial electric field algorithm," *International Journal of Imaging Systems and Technology,* vol. 31, no. 3, pp. 1386-1403, Sep 2021.

[40] M. S. Yildirim and E. Dandil, "Automatic detection of multiple sclerosis lesions using Mask R-CNN on magnetic resonance scans," *Iet Image Processing,* vol. 14, no. 16, pp. 4277-4290, Dec 2020.

[41] Y. D. Zhang, "A seven-layer convolutional neural network for chest CT based COVID-19 diagnosis using stochastic pooling," *IEEE Sens. J.,* pp. 1-1. doi: 10.1109/JSEN.2020.3025855

[42] S.-H. Wang, "Covid-19 Classification by FGCNet with Deep Feature Fusion from Graph Convolutional Network and Convolutional Neural Network," *Information Fusion,* vol. 67, pp. 208-229, 2020/10/09/ 2021.

[43] Y.-D. Zhang, "A five-layer deep convolutional neural network with stochastic pooling for chest CT-based COVID-19 diagnosis," *Machine Vision and Applications,* vol. 32, 2021, Art no. 14.

[44] M. Taskiran, N. Kahraman, and C. E. Erdem, "Hybrid face recognition under adverse conditions using appearance-based and dynamic features of smile expression," *Iet Biometrics,* vol. 10, no. 1, pp. 99-115, Jan 2021.

[45] M. M. Rahman and F. H. Siddiqui, "Multi-layered attentional peephole convolutional LSTM for abstractive text summarization," *Etri Journal,* vol. 43, no. 2, pp. 288-298, Apr 2021.

[46] P. Dey, "The emerging role of deep learning in cytology," *Cytopathology,* vol. 32, no. 2, pp. 154-160, Mar 2021.

[47] D. S. Guttery, "Improved Breast Cancer Classification Through Combining Graph Convolutional Network and Convolutional Neural Network," *Information Processing and Management,* vol. 58, 2, 2021, Art no. 102439.

[48] X. Cheng, "PSSPNN: PatchShuffle Stochastic Pooling Neural Network for an Explainable Diagnosis of COVID-19 with Multiple-Way Data Augmentation," *Computational and Mathematical Methods in Medicine,* vol. 2021, 2021, Art no. 6633755.

[49] W. Zhu, "ANC: Attention Network for COVID-19 Explainable Diagnosis Based on Convolutional Block Attention Module," *Computer Modeling in Engineering & Sciences,* vol. 127, 3, pp. 1037-1058, 2021.

[50] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 33, no. 12, pp. 2368-2382, 2011.

[51] K. Kiwitz, C. Schiffer, H. Spitzer, T. Dickscheid, and K. Amunts, "Deep learning networks reflect cytoarchitectonic features used in brain mapping," *Scientific Reports,* vol. 10, no. 1, Dec 2020, Art no. 22039.

[52] A. S. Nencka *et al.*, "Split-slice training and hyperparameter tuning of RAKI networks for simultaneous multi-slice reconstruction," *Magn. Reson. Med.,* p. 9. doi: 10.1002/mrm.28634 Article; Early Access. [Online]. Available: <Go to ISI>://WOS:000599191800001

[53] E. Kotze and B. Senekal, "Not just a language with white faces: Analysing #taalmonument on Instagram using machine learning," *Td-the Journal for Transdisciplinary Research in Southern Africa,* vol. 16, no. 1, Dec 2020, Art no. a871.

[54] D. Marima, B. Hadad, S. Froim, A. Eyal, and A. Bahabad, "Visual data detection through side-scattering in a multimode optical fiber," *Opt. Lett.,* vol. 45, no. 24, pp. 6724-6727, Dec 2020.

[55] Y. Pang, M. Sun, X. Jiang, and X. Li, "Convolution in convolution for network in network," *IEEE transactions on neural networks and learning systems,* vol. 29, no. 5, pp. 1587-1597, 2017.

[56] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," *arXiv preprint arXiv:1612.03928,* 2016.

[57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research,* vol. 15, no. 1, pp. 1929-1958, 2014.

[58] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4133-4141.

[59] R. K. Pandey and R. A. Ganesan, "DeepInterpolation: fusion of multiple interpolations and CNN to obtain super-resolution," *Iet Image Processing,* vol. 14, no. 15, pp. 4000-4011, Dec 2020.

[60] Q. Zhou, "ADVIAN: Alzheimer's Disease VGG-Inspired Attention Network Based on Convolutional Block Attention Module and Multiple Way Data Augmentation," *Front. Aging Neurosci.,* vol. 13, 2021, Art no. 687456.

[61] S. C. Satapathy and D. Wu, "Improving ductal carcinoma in situ classification by convolutional neural network with exponential linear unit and rank-based weighted pooling," *Complex Intell. Syst.,* vol. 7, pp. 1295-1310, 2020/11/22 2021.

[62] R. Girshick, F. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2015, pp. 437-446.

[63] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*, 2014: Springer, pp. 184-199.

[64] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646-1654.

[65] Y. T. Xiao, "TReC: Transferred ResNet and CBAM for Detecting Brain Diseases," *Front. Neuroinformatics,* vol. 15, Dec 2021, Art no. 781551.

[66] S. Lu, "Detecting pathological brain via ResNet and randomized neural networks," *Heliyon,* vol. 6, no. 12, p. e05625, 2020.

[67] M. Mora, J. Naranjo-Torres, and V. Aubin, "Convolutional Neural Networks for Off-Line Writer Identification Based on Simple Graphemes," *Applied Sciences-Basel,* vol. 10, no. 22, Nov 2020, Art no. 7999.

[68] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387,* 2015.

[69] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251-1258.

[70] G. Larsson, M. Maire, and G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," *arXiv preprint arXiv:1605.07648,* 2016.

[71] J. Cheng, P.-s. Wang, G. Li, Q.-h. Hu, and H.-q. Lu, "Recent advances in efficient computation of deep convolutional neural networks," *Frontiers of Information Technology & Electronic Engineering,* vol. 19, no. 1, pp. 64-77, 2018.

[72] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," *arXiv preprint arXiv:1710.09282,* 2017.

[73] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146,* 2016.

[74] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: Generalizing residual architectures," *arXiv preprint arXiv:1603.08029,* 2016.

[75] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848-6856.

[76] K. Zhang, M. Sun, T. X. Han, X. Yuan, L. Guo, and T. Liu, "Residual networks of residual networks: Multilevel residual networks," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 28, no. 6, pp. 1303-1314, 2017.

[77] M. Abdi and S. Nahavandi, "Multi-residual networks: Improving the speed and accuracy of residual networks," *arXiv preprint arXiv:1609.05672,* 2016.

[78] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.

[79] S.-H. Wang, "DenseNet-201-Based Deep Neural Network with Composite Learning Factor and Precomputation for Multiple Sclerosis Classification," *ACM Trans. Multimedia Comput. Commun. Appl.,* vol. 16, no. 2s, p. Article 60, 2020.

[80] S. C. Satapathy, "Covid-19 diagnosis via DenseNet and optimization of transfer learning setting," *Cognitive Computation*. doi: 10.1007/s12559-020-09776-8

[81] M. Astaraki, G. Yang, Y. Zakko, I. Toma-Dasu, O. Smedby, and C. L. Wang, "A Comparative Study of Radiomics and Deep-Learning Based Methods for Pulmonary Nodule Malignancy Prediction in Low Dose CT Images," *Frontiers in Oncology,* vol. 11, Dec 2021, Art no. 737368.

[82] H. S. Shad *et al.*, "Comparative Analysis of Deepfake Image Detection Method Using Convolutional Neural Network," *Computational Intelligence and Neuroscience,* vol. 2021, Dec 2021, Art no. 3111676.

[83] D. Sulot, D. Alonso-Caneiro, D. R. Iskander, and M. J. Collins, "Deep learning approaches for segmenting Bruch's membrane opening from OCT volumes," *OSA Continuum,* vol. 3, no. 12, pp. 3351-3364, Dec 2020.

[84] A. Woloshuk *et al.*, "In Situ Classification of Cell Types in Human Kidney Tissue Using 3D Nuclear Staining," *Cytometry Part A,* vol. 99, no. 7, pp. 707-721, Jul 2021.

[85] K. Wu, "SOSPCNN: Structurally Optimized Stochastic Pooling Convolutional Neural Network for Tetralogy of Fallot Recognition," *Wireless Communications and Mobile Computing,* vol. 2021, p. 5792975, 2021/07/02 2021, Art no. 5792975.

[86] G. Andrew and Z. Menglong, "Efficient convolutional neural networks for mobile vision applications," ed: Mobilenets, 2017.

[87] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510-4520.

[88] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229,* 2013.

[89] X. Han and Q. Dai, "Batch-normalized Mlpconv-wise supervised pre-training network in network," *Applied Intelligence,* vol. 48, no. 1, pp. 142-155, 2018.
[90] D. T. Nguyen, W. Li, and P. O. Ogunbona, "Human detection from images and videos: A survey," *Pattern Recognition,* vol. 51, pp. 148-175, 2016.
[91] Y. Li, S. Wang, Q. Tian, and X. Ding, "Feature representation for statistical-learning-based object detection: A review," *Pattern Recognition,* vol. 48, no. 11, pp. 3542-3559, 2015.
[92] M. Pedersoli, A. Vedaldi, J. Gonzalez, and X. Roca, "A coarse-to-fine approach for fast deformable object detection," *Pattern Recognition,* vol. 48, no. 5, pp. 1844-1853, 2015.
[93] S. Zagoruyko *et al.*, "A multipath network for object detection," *arXiv preprint arXiv:1604.02135,* 2016.
[94] P. N. Sabes and M. I. Jordan, "Advances in neural information processing systems," in *In G. Tesauro & D. Touretzky & T. Leed (Eds.), Advances in Neural Information Processing Systems*, 1995: Citeseer.
[95] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *International conference on machine learning*, 2015: PMLR, pp. 597-606.
[96] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *IEEE transactions on Neural Networks,* vol. 21, no. 10, pp. 1610-1623, 2010.
[97] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580-587.
[98] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440-1448.
[99] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems,* vol. 28, 2015.
[100] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117-2125.
[101] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961-2969.
[102] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 39, no. 4, pp. 640-651, 2016.
[103] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis," *Future Generation Computer Systems,* vol. 115, pp. 279-294, 2021.
[104] M. Torres and F. Cantú, "Learning to see: Convolutional neural networks for the analysis of social science data," *Political Analysis,* vol. 30, no. 1, pp. 113-131, 2022.
[105] D. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: a survey," *Evolutionary intelligence,* pp. 1-22, 2021.
[106] A.-A. Tulbure, A.-A. Tulbure, and E.-H. Dulf, "A review on modern defect detection models using DCNNs–Deep convolutional neural networks," *Journal of Advanced Research,* vol. 35, pp. 33-48, 2022.