IT Carlow - BSc.
Software development

Automatic Detection of Brand Logos Research Document

Student Name: Zhe Cui
Student Number: C00266169
Lecturer: Paul Barry
Date: 30.04.2021

INSTITUTE of TECHNOLOGY CARLOW
Institiúid Teicneolaíochta Cheatharlach

Table of Contents

Table of Figures

1 Introduction

Great development has taken place in some technology areas such as the computer science, multimedia and the Internet with the rapid growth of the global economy, particularly in recent years with the advent of the big data era, whether in science or industrial production field, the need for automatic image processing is increasing.

It is greatly significant to simulate human do object detection in a complex environment by using Automatic Image Processing Technology, which is a branch of image processing technology, Automatic Detection of Brand Logos technology also has a non-negligible status.

Automatic Detection of Brand Logos is primarily achieved by a series of processing work on input images, locating the logo and identifying the logo. It can help evaluate marketing campaigns, capture user reviews of the product, counterfeit detection and protect brands' intellectual property, personalize product recommendations, improve search algorithms. If a brand is sponsoring a sporting event or serves as a sports team supporter. The logo is seen on the uniforms of athletes, judges, coaches, and spectators, and is featured among many other positions on stadium banners. As it can be viewed by in-person audiences as well as millions of internet followers, the company seeks to raise awareness through brand promotion.

2 The status of Automatic Detection of Brand Logos Technology

The Automatic Detection of Brand Logos Technology is based on object detection. There are 2 tasks of object detection, location and recognition. Automatic Detection of Brand Logos Technology originated in the 1990s, based on the traditional sliding window mechanism and traditional machine learning technology. The idea is to traversal each area of the input image through a sliding window of variable size, the images of each area are then the invariant feature is extracted. Finally, Adaboost, Support Vector Machine (SVM) and other classifiers were used to classify the extracted features. With the development of Artificial Neural Network and Deep Learning, object detection based on Deep Learning is proposed.

The main task of the Automatic Detection of Brand Logos Technology is feature extraction. There are some common feature extraction algorithm, the Oriented FAST and rotated BRIEF (ORB) feature descriptor [1], Scale-Invariant Feature Transform (SIFT) feature descriptor [2], Speeded Up Robust Features (SURF) feature descriptor [3], Haar feature description operator [4], HOG (Histogram of Oriented Gradient) feature descriptor [5], LBP (Local Binary Pattern) feature descriptor [6].

SIFT feature descriptor finds the feature points in different scale-spaces and calculates the direction of the feature points. The feature points found by SIFT are some very prominent points that will not be changed by illumination, affine transformation and noise, such as corner points, edge points, bright spots in dark areas and dark spots in bright areas. SIFT feature is the local feature of the image, which is invariant for rotation, scaling and brightness change, and also stable for the change of perspective, affine transformation and noise. SURF description operator using the Box Filter to find the features points, which improves the SITF feature description operator, makes a qualitative leap in the running time of object detection algorithm. The HOG feature operator mainly captures contour information. Grayscaling the image first and then Gammaing correction image, reducing the local image shadow and light changes caused by the impact, at the same time the noise interference is suppressed. It counts gradient histogram of each cell, and then several cell histograms to form a descriptor of a block, the descriptor of the whole picture is formed by all blocks. The invariance of the geometric and optical deformation could be maintained very well, and it was first used in face detection by Papageorgiou et al [4].

The performance of traditional object detection methods is strongly depend on the manually selected features. The Deep Learning model proposed by Hinton in 2006 could learn the higher representations and features by multiple hidden layers automatically[7]. The deep learning models based on Restricted Boltzmann Machine (RBM) [8], deep learning models based on autoencoder [9], and deep learning models based on the Convolutional Neural Network (CNN) [10] are the popular deep learning models.

With the development of Deep Learning technology, the accuracy of object detection could be greatly improved by deep learning model based on the Convolutional Neural Network. Girshick R proposed the R-CNN (Regions with CNN features) algorithm in 2014 [11], which has made significant breakthroughs in object detection algorithms based on deep learning. In the R-CNN algorithm, the selective search algorithm is used to generate candidate regions instead of the traditional sliding window.

The mainstream method used in traditional object detection and recognition is Deformable Parts Models (DPM), whose mAP is 43% for the VOC2007 data set. Object detection based on Convolutional Neural Network originated in 1994 [12], proposed by Le Cun. In 2013, Szegedy C tried to think the object detection as a regression problem, but mAP was only 30.5% [13] on the VOC2007 data set. Then the Regions with CNN features (R-CNN) algorithm, increased the mAP for the VOC2007 dataset to 58%. Later, the optimization algorithm based on the R-CNN algorithm has proposed, such as the SPP-Net algorithm [14], Fast R-CNN algorithm [15], Faster R-CNN algorithm [16], You Only Look Once (Yolo) algorithm [17], Single Shot MultiBox Detector (SSD) algorithm [18], and so on.

3 Computer Vision

3.1 Reasons for developing Computer Vision

If the data is separated into structured data and unstructured data, as shown in Figure 3-1, the unstructured data was increase exponentially. Computer vision technology is particularly important to understand massive unstructured data.

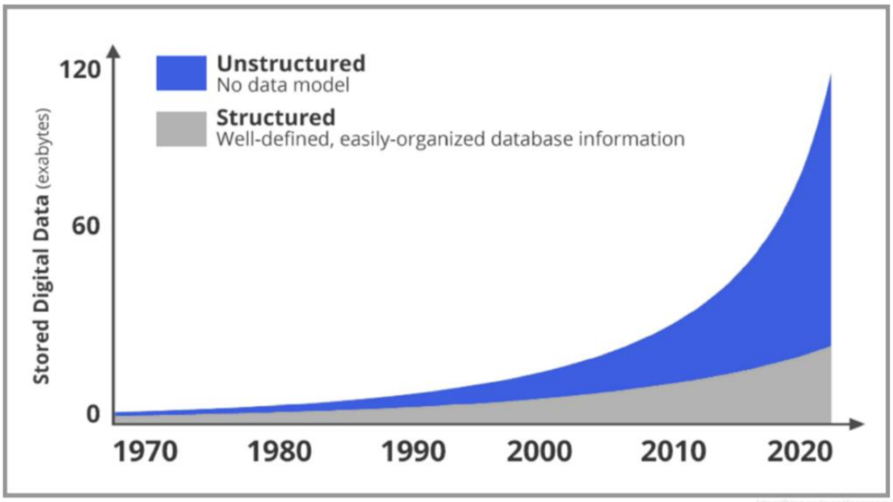
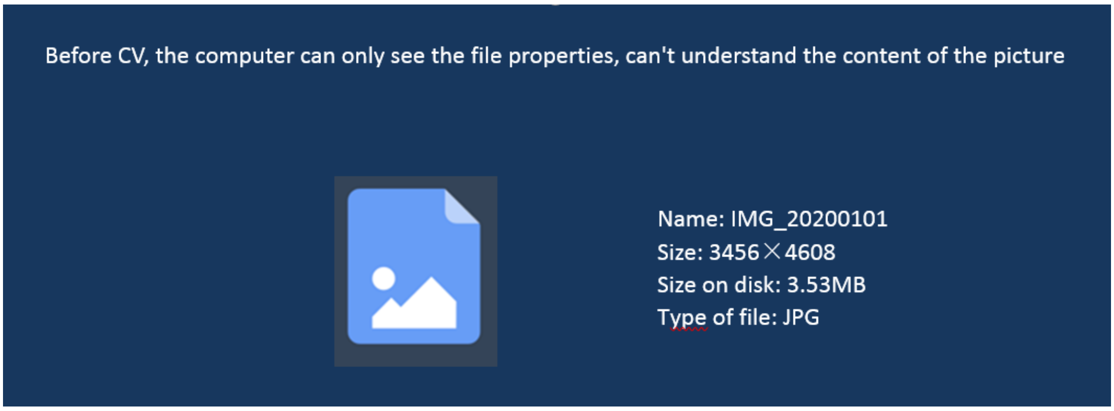


Figure 3-1 Structured data and unstructured data

Source: https://www.m-files.com/blog/what-is-structured-data-vs-unstructured-data/ (2020) What is Structured Data vs. Unstructured Data?

Before computer vision, images were viewed as black boxes, as shown in Figure 3-2, only the file name, size, size on disk, and type of file of the image could be read.

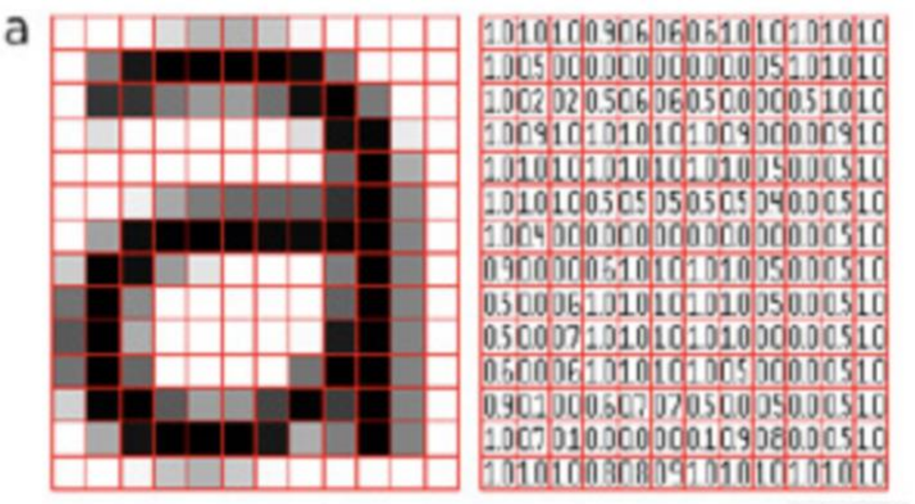


Before CV, the computer can only see the file properties, can't understand the content of the picture

Name: IMG_20200101
Size: 3456×4608
Size on disk: 3.53MB
Type of file: JPG

Figure 3-2 Image properties

When computer vision has made initial developments, it can only saw densely arranged numbers when opening a picture, as shown in Figure 3-3, but the meaning of these numbers can not be known, and these numbers cannot be combined with lowercase ‘a’ correspondingly. To solve this problem, computer vision technology is needed.

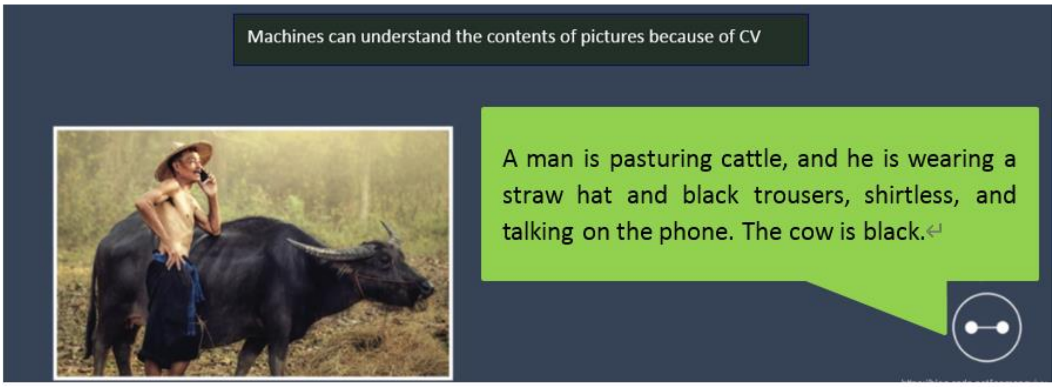


1.0	1.0	1.0	0.9	0.6	0.6	0.6	1.0	1.0	1.0	1.0	1.0
1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0	1.0	1.0
1.0	0.2	0.2	0.5	0.6	0.6	0.5	0.0	0.0	0.5	1.0	1.0
1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.0	0.0	0.9	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5	0.4	0.0	0.5	1.0
1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0
0.9	0.0	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.6	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.7	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.5	1.0
0.6	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.0	0.5	1.0
0.9	0.1	0.0	0.6	0.7	0.7	0.5	0.0	0.5	0.0	0.5	1.0
1.0	0.7	0.1	0.0	0.0	0.0	0.1	0.9	0.8	0.0	0.5	1.0
1.0	1.0	1.0	0.8	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0

Figure 3-3 Computer Vision
Source: http://iesalmadraba.org/dibujo/epv4/elementos-de-la-imagen/color-mapas-de-bits/ (2020)

3.2 Concept of computer vision

In a narrow sense, computer vision is to understand the content of the image, as shown in Figure 3-4, a man with a straw hat and black trousers, shirtless, is pasturing cattle, and he is wearing and talking on the phone. The cow is black.



Machines can understand the contents of pictures because of CV

A man is pasturing cattle, and he is wearing a straw hat and black trousers, shirtless, and talking on the phone. The cow is black.↵

Figure 3-4 Computer vision allows machines to understand images

Human beings are relying on the visual system to get this information. As shown in Figure 3-5, neuroscientists have found that the human visual system is divided primarily into the retina, the V1 layer, the V2 layer, the V4 layer.

- The retina projections object to the retina and through the optic nerve, convert light to electrical signals and send these signals to the cerebral cortex.

- The V1 layer is primarily sensitive to edge information, and its role is similar to the Laplacian operator and the Sobel operator in conventional image processing, to extract edge information from objects.

- The V2 layer aims to synthesize the edge information into simple shapes, which is the local object information.

- The V4 layer obtain semantic information explicitly.

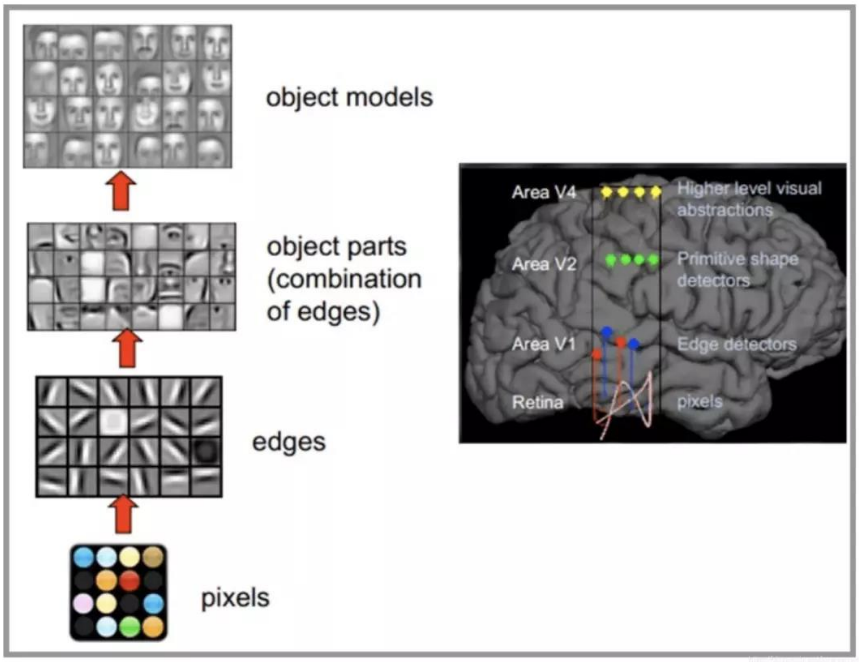


Figure 3-5 The structure of the human visual system
Source: https://www.programmersought.com/article/4092569936/ (2019) Deep Learning

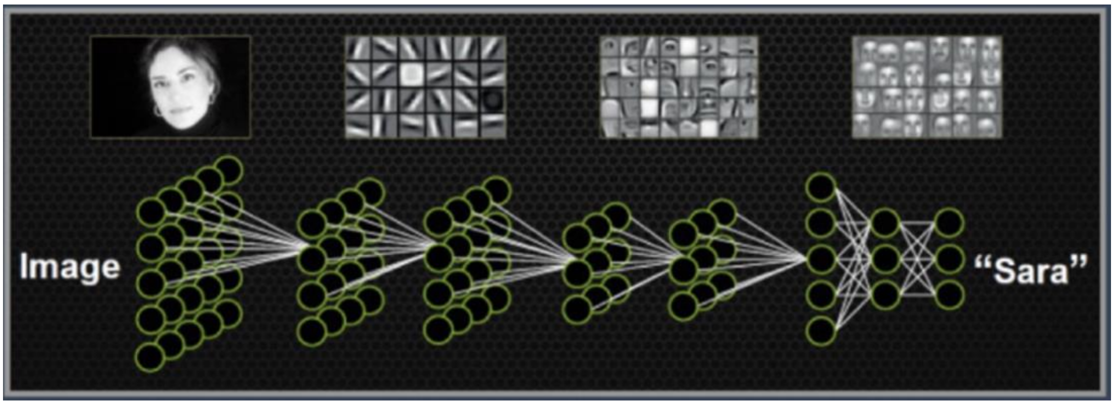


Figure 3-6 A schematic overview of how a basic neural network functions
Source: https://www.aviationtoday.com/2020/05/15/neural-networks-already-showing-future-potential-aerospace/ (2020) How Neural Networks are Already Showing Future Potential for Aerospace?

4 Tools

4.1 Python

Python is a high-level, interpretive, compiled, interactive, and object-oriented, language. Python is very readable, and compared to traditional languages like C/C++, Java, and C#, Python has less strict code formatting requirements. Python has the richest and powerful class libraries in scripting languages, which covers file I/O, GUI, network programming, database access, text manipulation, and most of the application scenarios. The underlying code of these libraries is not necessarily Python, and there is a lot of C/C++. When a piece of code needs to run faster, it can be implemented in C/C++ and then called in Python. Python is known as the "glue language" for its ability to "glue" other languages together. Python, with its rich resources and solid numerical algorithms, ICONS, and data-processing infrastructure, is the perfect language for AI, and the most suitable language for this project.

4.2 OpenCV

OpenCV is a cross-platform computer vision and machine learning software library based on the BSD license, it could run on Linux, Windows, Android, and Mac OS operating systems. It's lightweight and efficient, it's made up of a set of C functions and a handful of C++ classes, providing interfaces to languages like Python, Ruby, and MATLAB, and implementing many common algorithms for image processing and computer vision.

5 History of Computerized Image Recognition

Image recognition refers to the technology used by computers to process, analyze, and understand images to identify various targets and images of different patterns.

Image recognition is an important field of artificial intelligence. To develop a computer program to simulate human image recognition, different image recognition models are proposed. For example, the template matching model. This model states that to identify an image, you must have a memory pattern for that image in your experience or a template. If the present stimuli fit the template in the brain, the image is recognized. For example, if there is a letter ‘A’, if there is a template in the brain, the size, orientation and shape of the letter ‘A’ are the same as the letter ‘A’, then the letter ‘A ‘is recognized. This model is clear and easy to incorporate into practice. But this model emphasizes that only images that match the template in the brain can be recognized. In reality, human beings can recognize not only images that fit the templates in the brain, but also images that do not match the templates. For example, human beings can recognize not only a particular letter ‘A’, but also typed letters, handwritten letters, different sizes of letters ‘A’. At the same time, a vast number of images can be recognized by humans, and it is also impossible to recognize each image as having an appropriate template in the brain.

In order to solve the problem of the template matching model, gestalt psychologists have proposed a prototype-matching model. This model assumes that what is retained in long-term memory is not the various models to be recognized, but some "similarity" in the image. However, this model does not show people identify and process-related stimuli and it is difficult to incorporate in computer programs. A more complex model, the " pandemonium model " has been proposed.

The development of image recognition has gone through three stages: text recognition, digital image processing and recognition, and object recognition. Character recognition research started in 1950, usually recognizing letters, numbers and symbols. It is commonly used for print character recognition and handwritten character recognition.

Yann LeCun has developed a text recognition system based on a convolutional neural network that can recognize handwritten numbers, called Lenet-5 [19], using the MNIST data set. The system has been used to recognize handwritten numbers in banks since the 1990s, and the recognition rate is very high.

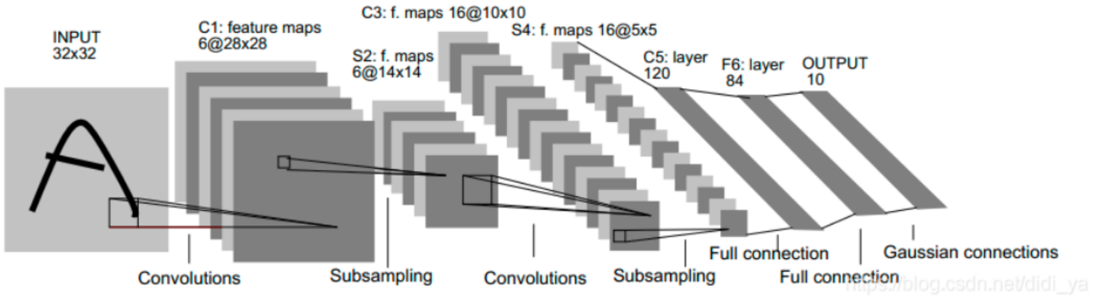


Figure 5-1 LeNet-5

Source: https://ieeexplore.ieee.org/abstract/document/726791 (1998) Gradient-based learning applied to document recognition

5.1 Introduction to the MNIST data set

The MNIST dataset [20] is a popular dataset for image classification machine learning model tutorials. It contains 70,000 images of handwritten digits from American Census Bureau employees and American high school students. There are 60,000 training images(55,000 were used for training and 5,000 for verification) and 10,000 testing images. Each image has a normalized and focused size. The image of this data set is a single-channel image in black and white. Each image has a resolution of 28x28, and the numerical pixel values are in greyscale. Each image is represented by a 28x28 matrix with each element of the matrix an integer between 0 and 255. The label of each image is the intended digit of the handwritten image between 0 and 9. A sample of the dataset is illustrated in the figure below:

Figure 5-2 Sample of MNIST dataset

Source: https://ci.nii.ac.jp/naid/10027939599/ (1998) THE MNIST DATABASE of handwritten digits

5.2 Composition of the MNIST data set

In the MNIST is split into three parts, 55,000 data points of training data (mnist.train), 10,000 points of test data (mnist.test), and 5,000 points of validation data (mnist.validation), mnist.train.images is a tensor (an n-dimensional array) with a shape of [55000, 784]. The first dimension indexes the images and the second dimension indexes the pixels in each image. Each entry in the tensor is the pixel intensity between 0 and 1, for a particular pixel in a particular image.

During the training, the model did not meet data from the validation data set, so the accuracy of the model could be tested using the validation data set. The more reliable this is, the more generalized the code model is. Besides, the three data sets have three corresponding files (label files) that are used to denote the number of each image. Put pictures and marks together, and name them " annotation." The label of the MNIST data set is a number between 0 and 9, which is used to describe the number shown in the given image.

The label data is "one-hot vector", except for the number of a certain digit 1, the other dimensions are all 0. For example, label 0 will be expressed as ([1,0,0,0,0,0,0,0,0,0,0,0]). If the one-hot format is Valid, that is, the parameter of one-hot is True in the above code, then the label in each use case is no longer a number, but an array of length 10. There are 9 zeros in the array, and just a 1 (note that this is a use case). For example, if you write 3 by hand, it becomes [0, 0, 0 , 1, 0, 0, 0, 0, 0, 0, 0, 0] for a one-hot format. A single hot encoding format is used in a multi-class model. The other formats will remain unchanged.

6 Introduction to Deep Learning

There are 3 different types machine learning: Supervised Learning, Unsupervised Learning and Semi-supervised Learning. The main task of supervised learning is to generate an input-to-output mapping function between the labeled input data and output data. Unsupervised learning is processing unlabeled data and building the model directly, and the machine learns to find out the underlying rules between these different types of data, such as clustering. The semi-supervised learning is to build a learner to label unlabeled samples by using the model assumptions of the data distribution.

Traditional machine learning methods select and extract features manually, which is time-consuming and labor-intensive. And using the manual feature is difficult to complete another task. So selecting feature manually is not so effective, to solve the weakness of traditional machine learning. The machine learning tasks is expanded, not only learning the model, but also learning the features, automatically selecting the features. The biggest difference between Deep Learning and the traditional machine learning algorithm is that deep learning is a "end-to-end" method. As shown in Figure 6-1, the deep learning algorithm takes the original form of data as the initial input, extracts features layer by layer automatically. There is no need to manually extract features.

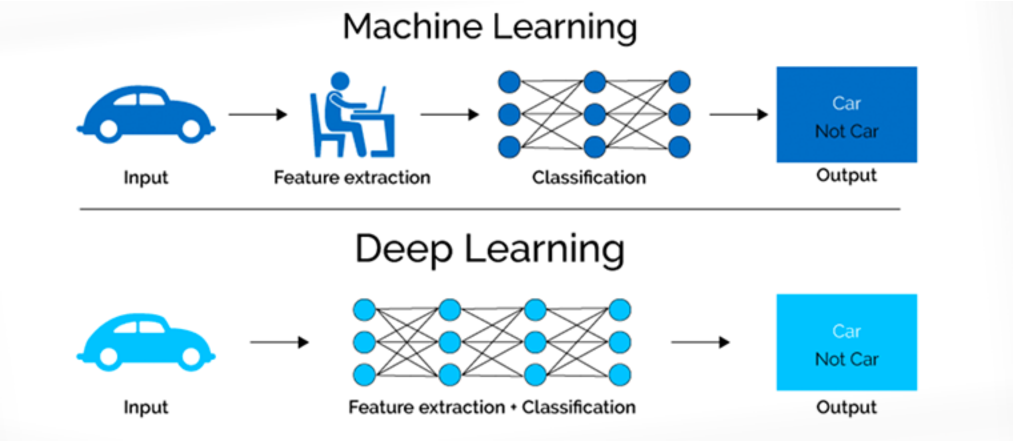


Figure 6-1 Comparison traditional machine learning and deep learning
Source: https://blog.csdn.net/z0n1l2/article/details/83692963 (2018)

6.1 Deep learning model based on AutoEncoder

The AutoEncoder is a variant of the multilayer perceptron, proposed by Rumelhart in 1986. The main task of AutoEncoder is to minimize the reconstruction error, the basic idea is to reconstruct the original input data through the multi-layer neural network. There are two processes of AutoEncoder, encoding and decoding. The structure of AutoEncoder is shown as Figure 6-2.

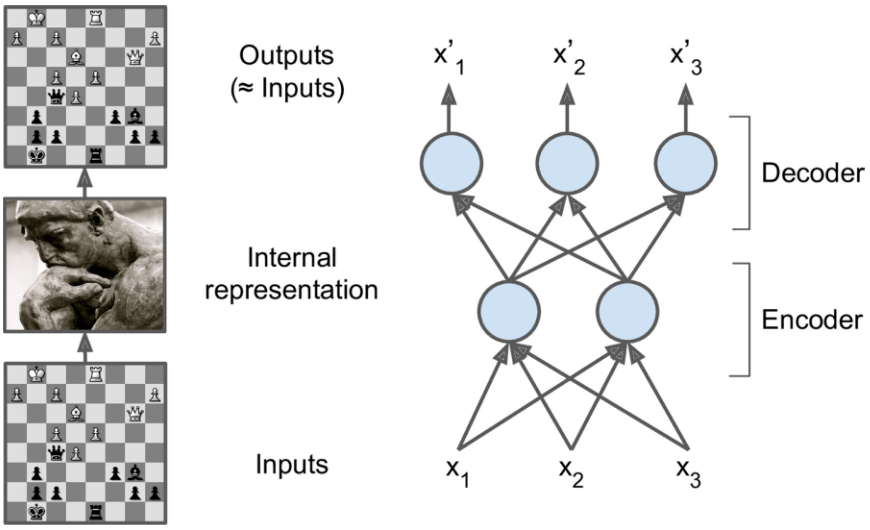


Figure 6-2 Schematic diagram of the basic structure of auto-encoder

The encoding process of the auto-encoder is mapping the input data x to the hidden feature through a certain mapping rule f(x), and the decoding process is reconstructing the hidden feature to input data through a certain decoding rule f'(z). The commonly used functions for nonlinear mapping in the encoding process are the sigmoid function and the Rectified Linear Unit (ReLU) function. These nonlinear functions are also called activation functions. The goal of the autoencoder for parameter training is to minimize the mean square deviation between the reconstructed signal and the input signal, as shown in Formula 6-1.

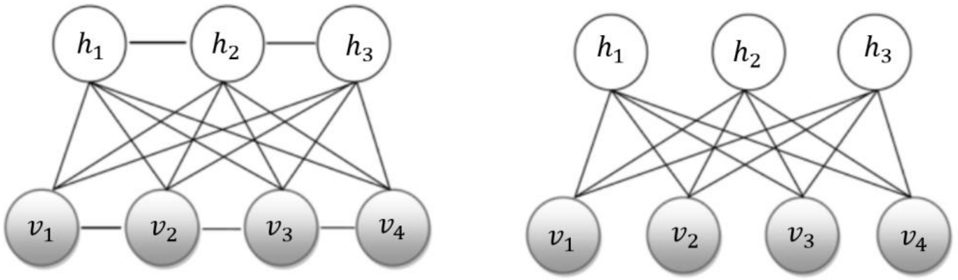
$$target = min \sum_{i} \left(\hat{x}_i \quad - \quad x_i \right)^2$$

Formula 6-1

Multiple AutoEncoders are stacked layer by layer to form a Stacked Autoencoder. The output of the previous layer of the Stacked Autoencoder is the input of the next layer. During the training process, each layer of the encoder is optimized separately, so each layer of the Stacked Autoencoder can encode the original input data. In this way, the characteristics of the input data can be expressed more effectively through the deep data relationship hidden inside the signal.

6.2 Deep learning model based on restricted Boltzmann machine

Boltzmann machine is one of the earliest random recursive artificial neural networks. It can learn the inherent internal representation of data and can solve more complex problems. The restricted Boltzmann machine is an extension of the Boltzmann machine. The structure of Boltzmann machine is a two-layer undirected fully connected graph, while the restricted Boltzmann machine removes the connection between the same layer of the Boltzmann machine to improve the learning efficiency of the network. The restricted Boltzmann machine is used as the basic unit of the Deep Belief Networks, to maximize the probability of qualified samples created by the learned model, this is the deep learning model based on a restricted Boltzmann machine.



H is hidden layer

(a) Boltzmann machine (b) restricted Boltzmann machine

V is visual layer

Figure 6-3 The difference between Boltzmann machines and restricted Boltzmann machines

6.3 Deep learning model based on convolutional neural network

Generally, there are 3 parts of the Convolution Neural Network: Convolution Layer, Pooling Layer and Fully Connected Layer.

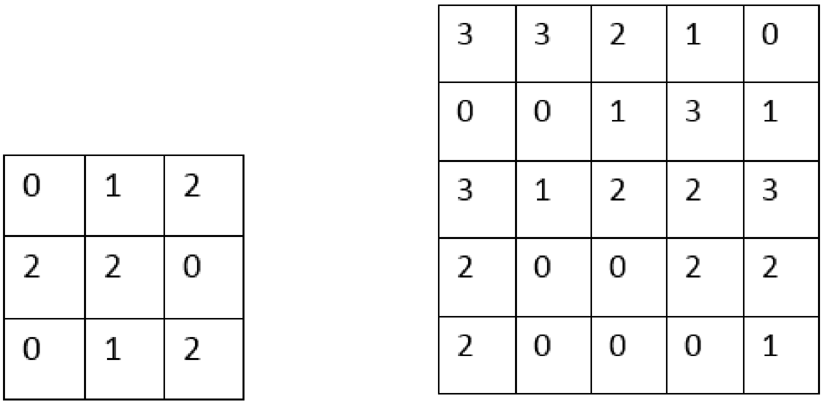
6.3.1 Convolution Layer

The convolution layer is the basic operation in convolutional neural networks. In some networks, the Fully Connected Layer, which is the classifier is replaced by convolution operation in the engineering implementation. The essence of convolution operation is an operation in analytical mathematics, and the convolution operation on images is equivalent to the filtering operation on images, such as the Gaussian filtering and Laplace filtering.

$$f\left(x,y\right)\times g\left(x,y\right)=\sum_{i=-m}^{m}\sum_{j=-n}^{n}g\left(i,j\right)\times f\left(x-i,y-j\right)$$

Formula 6-2

Formula expression of convolution operation is shown in Formula 6-2, the F(x,y) represents the pixel value of x row and Y column in the image, and g(x,y) is the filter, which is also called the convolution kernel. It can be seen from the formula that the essence of the convolution kernel is a weighted template, and the convolution operation is to get the eigenvalue of the corresponding point by weighting the covered pixels through the template. To illustrate the process of convolution operation more vividly, it is assumed that there is a 5×5 convolution input and 3×3 convolution kernels in the two-dimensional spaceplane. Let the convolution operation be a summation operation. The input plane and convolution kernel are shown in Figure 6-4.



0	1	2
2	2	0
0	1	2

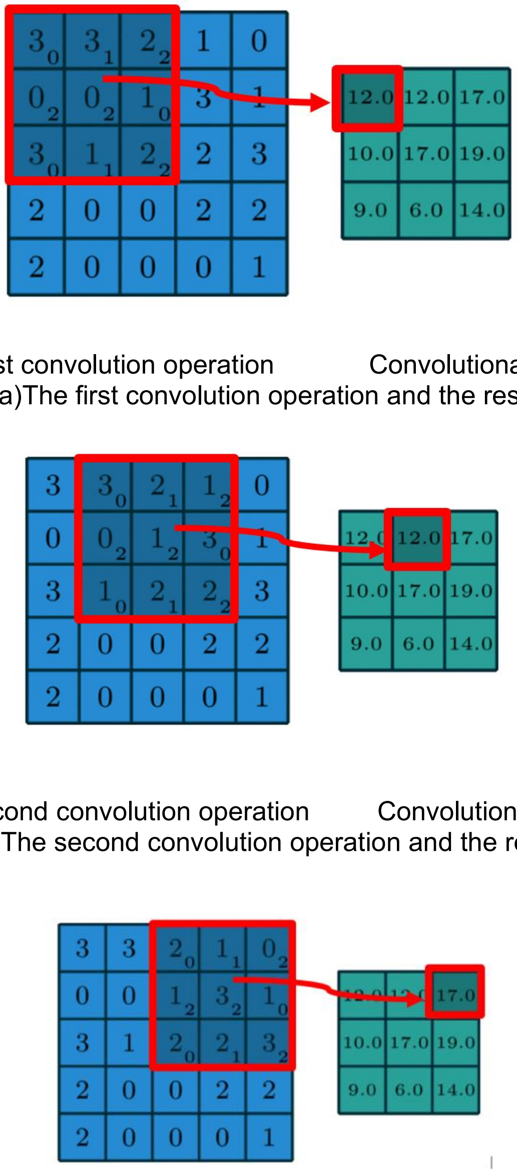
Convolution kernel

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

(b) Convolution input

Figure 6-4 Convolution kernel and convolution input in two-dimensional space

The first convolution operation starts from the position where is the input of the pixel (0, 0), by multiplying and summing the convolution kernel parameters and pixel of corresponding image position as a result of the first time convolution operation, as shown in Figure 6-5 (a), then to step 1 to the right, on the second convolution operation, as shown in Figure 6-5(b), and then from left to right, from top to bottom in turn for convolution operation, finally get a 3 x 3 characteristics of convolution results as shown in Figure 6-5, then the convolution results will enter into the next layer of the convolution operation. Convolution kernel (filter) in this picture is a 3x3 matrix, convolution kernels sliding from top to bottom, from left to right sliding in the input space, for every slide position, convolution kernels and elements multiplication and summation.



3_0	3_1	2_2	1	0
0_2	0_2	1_0	3	1
3_0	1_1	2_2	2	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

The first convolution operation Convolutional feature

(a)The first convolution operation and the result

3	3_0	2_1	1_2	0
0	0_2	1_2	3_0	1
3	1_0	2_1	2_2	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

The second convolution operation Convolutional feature

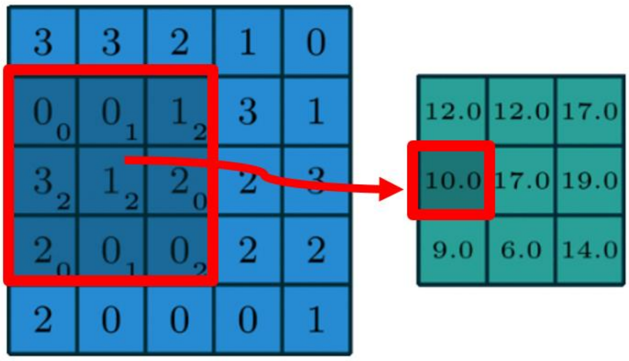
(b)The second convolution operation and the result

3	3	2_0	1_1	0_2
0	0	1_2	3_2	1_0
3	1	2_0	2_1	3_2
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

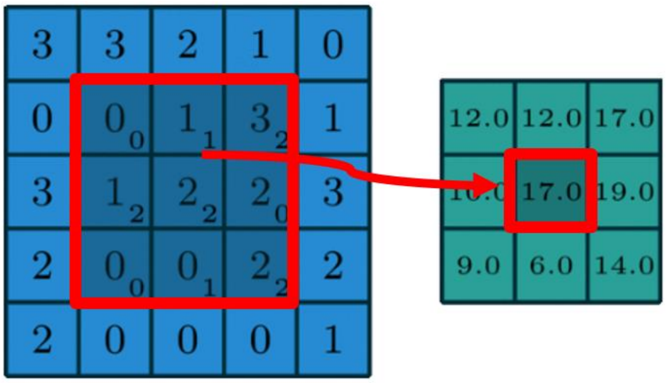
The third convolution operation Convolutional feature

(c)The third convolution operation and the result



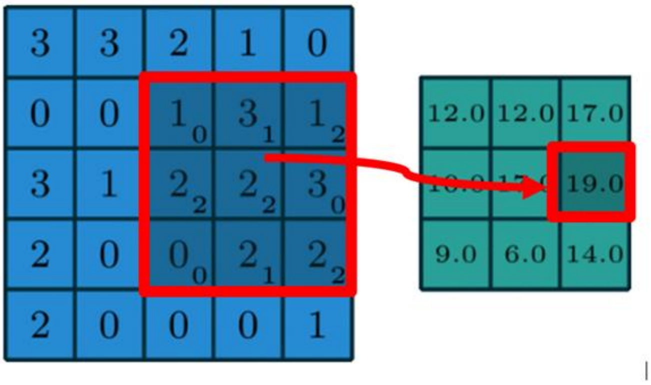
The fourth convolution operation Convolutional feature

(d) The fourth convolution operation and the result



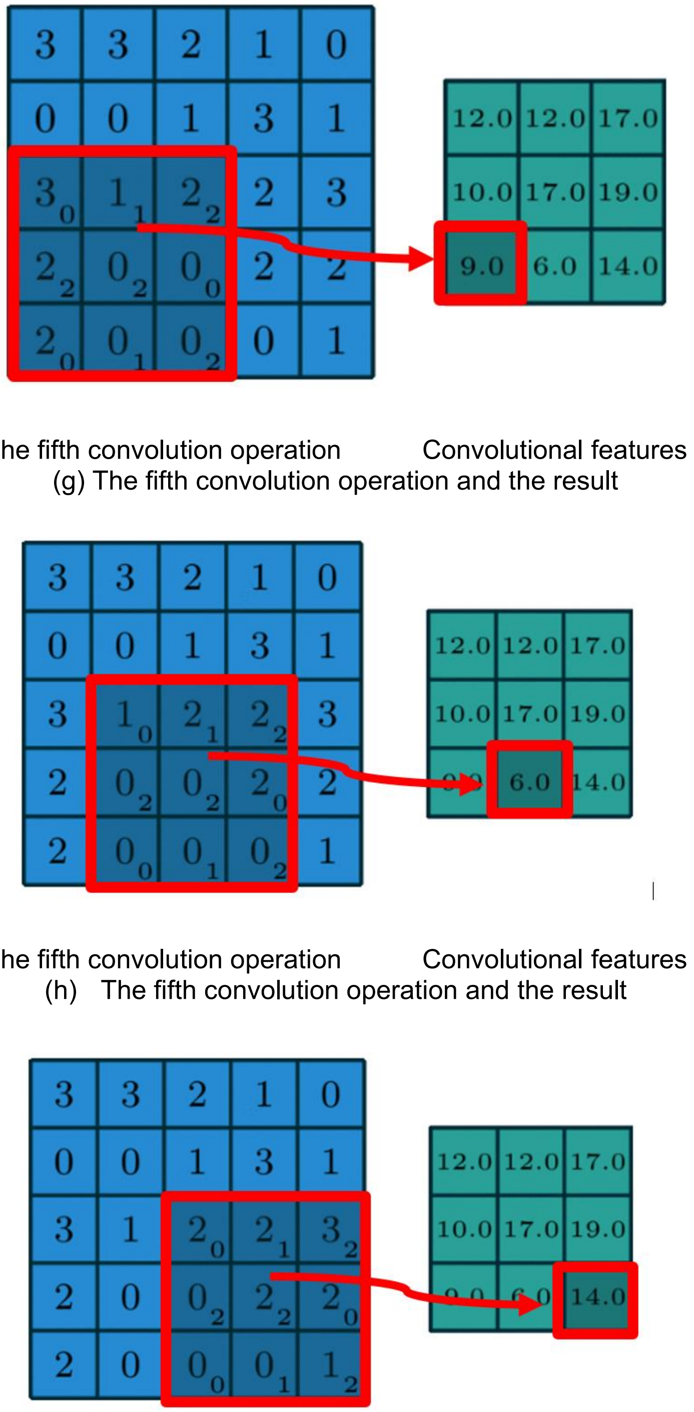
The fifth convolution operation Convolutional features

(e) The fifth convolution operation and the result



The fifth convolution operation Convolutional features

(f) The fifth convolution operation and the result



3	3	2	1	0
0	0	1	3	1
3_0	1_1	2_2	2	3
2_2	0_2	0_0	2	2
2_0	0_1	0_2	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

The fifth convolution operation Convolutional features

(g) The fifth convolution operation and the result

3	3	2	1	0
0	0	1	3	1
3	1_0	2_1	2_2	3
2	0_2	0_2	2_0	2
2	0_0	0_1	0_2	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

The fifth convolution operation Convolutional features

(h) The fifth convolution operation and the result

3	3	2	1	0
0	0	1	3	1
3	1	2_0	2_1	3_2
2	0	0_2	2_2	2_0
2	0	0_0	0_1	1_2

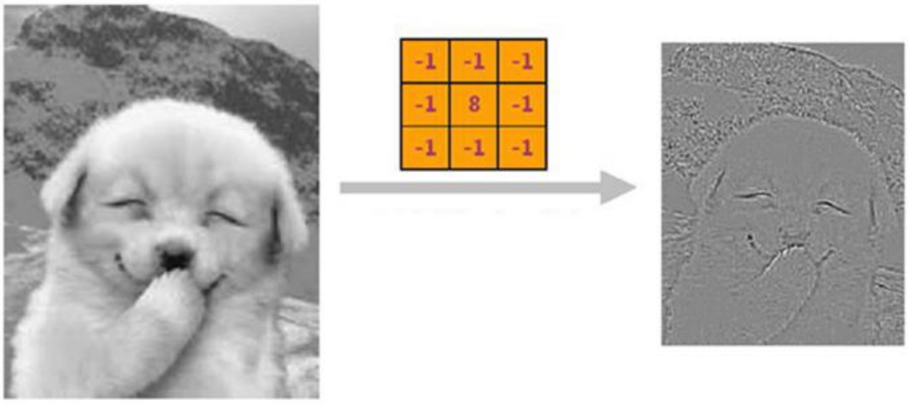
12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

The fifth convolution operation Convolutional features

(i) The fifth convolution operation and the result

Figure 6-5 The operation process and results of two-dimensional plane convolution

The convolution operation is a local operation, which acts on the partial area of the image through a certain size convolution kernel, to calculate the local information of the image. In a convolutional network, convolution kernel parameters are trained through the network, and the trained convolution kernel can play many roles, such as edge filtering, horizontal filtering, vertical filtering, color detection, shape detection, texture detection, etc. The effect of convolution operation on the image is shown in Figure 6-6. In practical neural networks, these convolution kernels are usually combined to achieve complex abstract effects and finally achieve the effect of abstracting high-level semantics.



-1	-1	-1
-1	8	-1
-1	-1	-1

Figure 6-6 The effect of convolution operation on images

6.3.2 Pooling Layer

The Pooling Layer is also called the down-sampling layer. The main function of this layer is to reduce the size of the representation, to speed the computation, as well as make some of the features that it detects a bit more robust. Generally, after one or more layers of convolution operation, the convolution features of input data are obtained. Researchers will use these features to perform classification operations, such as classifying the convolution features, which is the input of Softmax classifier or SVM classifier. In general, for the input of the classifier, the convolution feature data is too large, which will lead to long calculations and overfitting. Therefore, pooling is usually adopted for data dimension reduction to reduce operation time and the possibility of overfitting. Generally, pooling operations include non-overlapping pooling, overlapping pooling and spatial pyramid pooling. The difference between overlapped pooling and non-overlapped pooling is whether there are overlapped areas in adjacent pooling windows. The Spatial Pyramid Pooling (SPP) layer removes the fixed-size constraint of the network, i.e. a CNN does not require a fixed-size input image.

There are two different pooling methods: average pooling and max pooling. Average Pooling is a pooling operation that calculates the average value for patches of a feature map, and uses it to create a downsampled (pooled) feature map. Max Pooling calculates the maximum value for patches of a feature map, and uses it to create a downsampled (pooled) feature map.

In practice, max-pooling can be divided into overlapping pooling and non-overlapping pooling. For example, AlexNet/GoogLeNet use overlapping pooling and non-overlapping pooling in VGG are adopted. However, since ResNet, the pooling layer has been used less and less in the classification network, and the convolution with stride=2 is often used to replace the max-pooling layer. The advantage of max-pooling is that it can learn the edge and texture structure of the image. Average Pooling extracts features more smoothly than Max Pooling, whereas max-pooling extracts more pronounced features like edges.

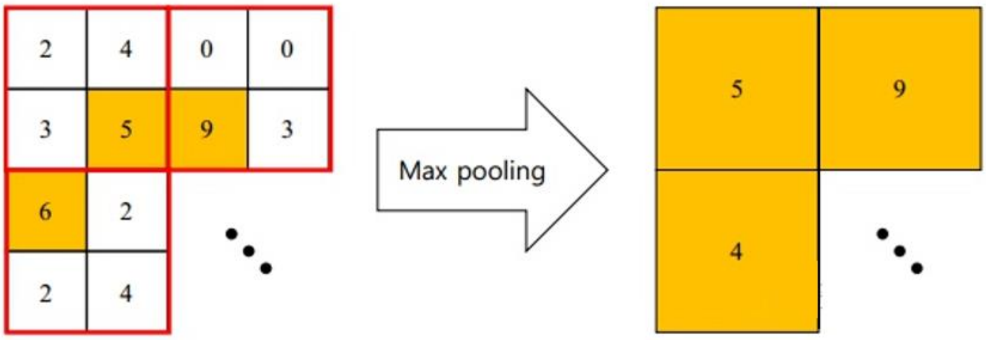


Figure 6-7 Max pooling

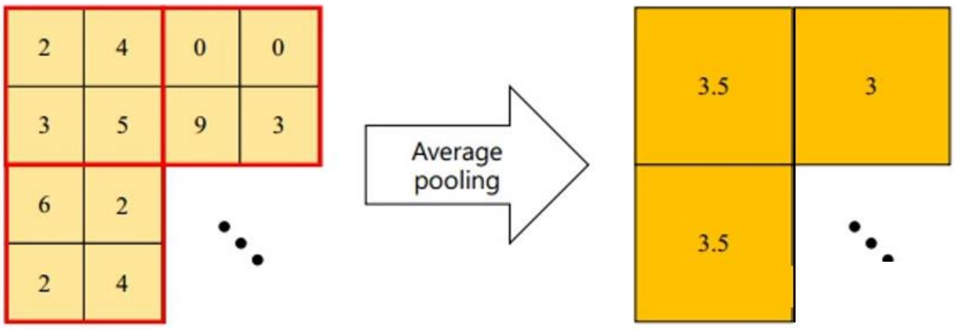


Figure 6-8 Average pooling

The most important feature of the pooling layer is that feature invariability can be guaranteed while feature dimension reduction operation is carried out. And that invariance includes translation, rotation, and scale. The main features are retained while the parameters (dimension reduction, similar to PCA) and calculation amount are reduced to prevent overfitting and improve the generalization ability of the model.

As the pooling layer has such a feature, the learning model pays more attention to the existence of certain features after the convolution operation of input data, rather than the specific location where the features appear. In fact, in many tasks (such as voice recognition, object detection, etc.), researchers hope to obtain translation invariance by calculation, because the characteristics of the target should remain unchanged even if the image has undergone translation and other transformations.

(1) The translation invariance:

Here's a simple example (digital recognition), there is 16 x16 picture and the number ‘1’, the ‘1’ may be written to the left a bit (as shown in Figure 6-9 and Figure 6-10 they are represented by ‘a’ and ‘b’. The positions of ‘a’ and ‘b’ in the original picture will eventually be mapped to the same position.), or the right point (as shown in Figure 6-10), Figure 1 translate 2 unit to the right is the Figure 2 But after the maximum pooling in Figure 1 and Figure 2, they both become the same 8x8 feature matrix, the main characteristics are captured, at the same time, the scale of the problem from 16x16 dropped to 8x8, and has the characteristics of translation invariance. In Figure 6-9 and Figure 6-10 they are represented by ‘a’ and ‘b’. The positions of ‘a’ and ‘b’ in the original picture will eventually be mapped to the same position.

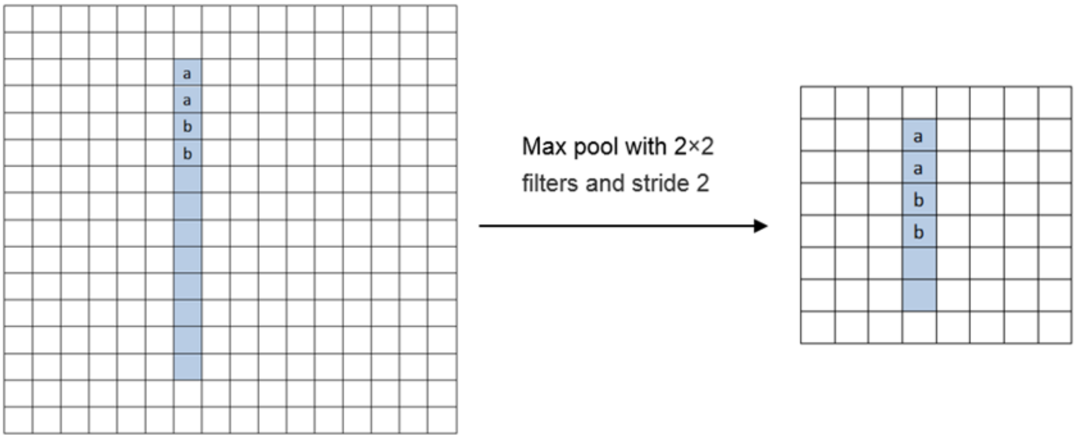


Figure 6-9 The translation invariance

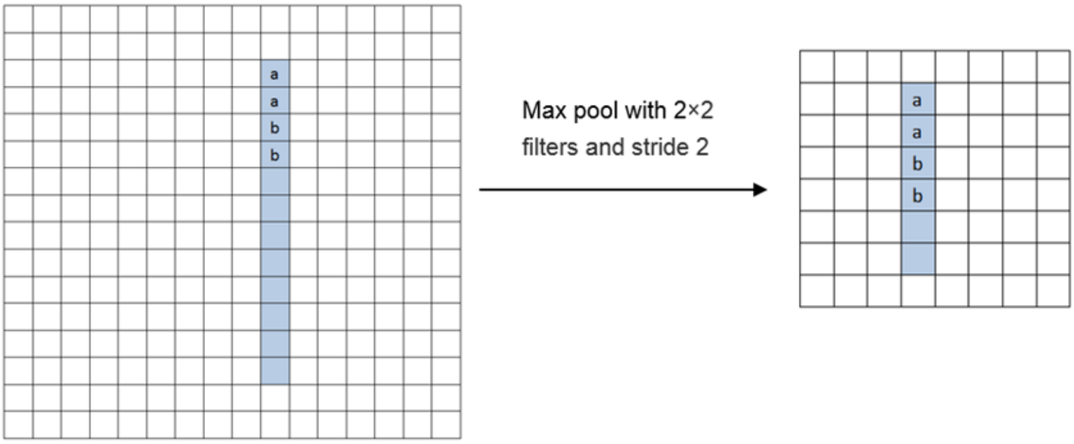


Figure 6-10 The translation invariance

(2) The rotation invariance:

The following figure represents the recognition of Chinese character "一", which looks like a horizontal line. These two pictures are equivalent to being rotated and have the same characteristics after multiple max pooling. Figure 6-11 has a tilt angle relative to the x-axis, and Figure 6-12 is parallel to the x-axis, these two pictures are equivalent to being rotated and have the same characteristics after multiple max pooling.

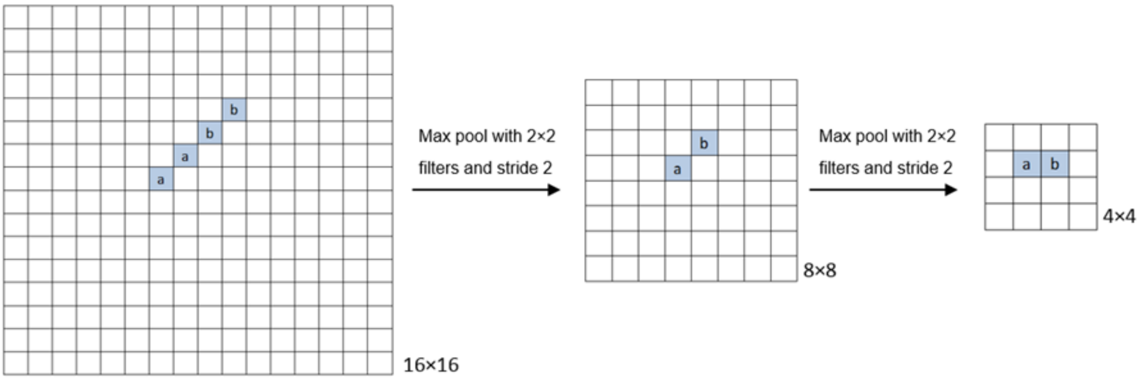


Figure 6-11 The rotation invariance

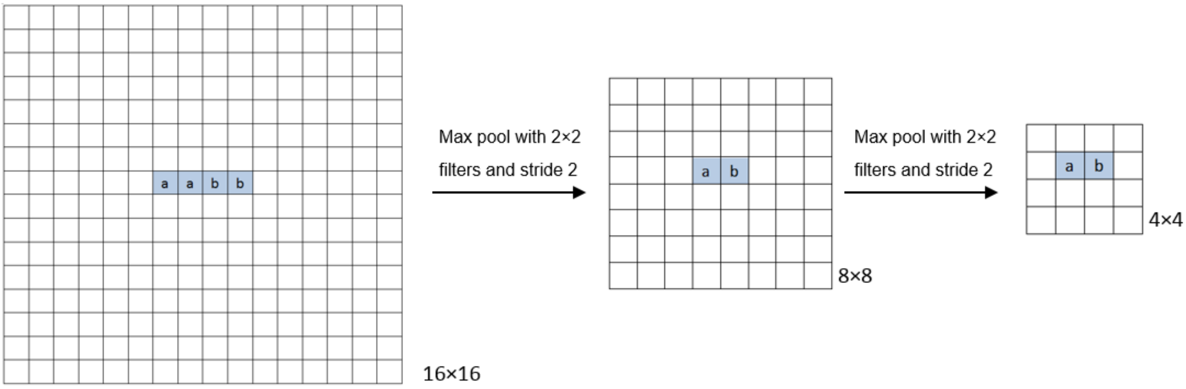


Figure 6-12 The rotation invariance

6.3.3 Fully connected layer

The fully connected layer mainly plays the role of ‘classifier’ in the whole convolution neural network. The convolutional layer and the pooling layer mainly study the characteristics of input data, which is the mapping of the original input data to the feature space. The objective of a fully connected layer is to take the results of the convolution/pooling process and use them to classify the image into a label (in a simple classification example). There is a large number of nuisance parameters in the fully connected layer, for example, only full connection layer parameters can account for about 80% of the entire network parameters. Therefore, some models use global average pooling (GAP) to replace the full connection layer to fuse the learned depth features, and finally use a loss function such as Softmax as the objective network function to guide the learning process. In ResNet and GoogLeNet models, GAP is used to replace the fully connected layer, because the replacement will make the network model have better predictive performance.

7 Detection and identification methods based on traditional methods

The general process of traditional object detection and recognition technology is mainly three steps: image preprocessing, feature extraction, and classifier classification. Image preprocessing mainly includes grayscale, binarization, smoothing, geometric changes, image enhancement and other technologies. Feature extraction is mainly the study of invariant feature operators, Common feature descriptor include SIFT feature descriptor, SURF feature descriptors, ORB feature descriptors, HOG feature descriptors, LBP feature descriptors, and Haar feature descriptors. And for the classifiers, generally use Adaboost classifiers and SVM classifiers.

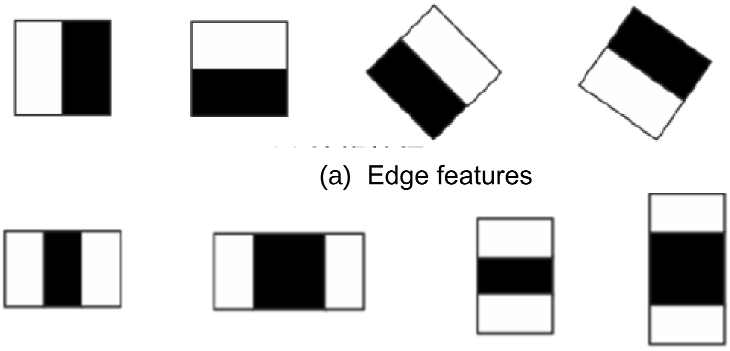
The traditional methods for object detection and recognition are mainly divided into three categories: Adaboost-based object detection algorithms, SVM-based object detection algorithms, and Deformable Part Model (DPM) algorithms.

7.1 Detection and recognition method based on Adaboost

The typical representative of this type of algorithm is the combination of Harr feature operator and Adaboost classifier, which first extracts the Haar feature description operator, and then uses the Adaboost classifier for classification. What makes the Haar feature description operator famous should belong to its application in the field of face detection. Follow-up researchers also apply the Haar feature description operator in the field of Logo detection and recognition [21].

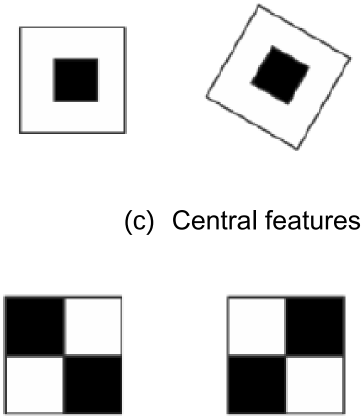
7.1.1 Haar feature descriptor

Haar features are divided into four categories: edge features, linear features, central features and diagonal features, which are combined into feature templates. There are white and black rectangles in the feature template, and the feature value of the template is defined as the sum of white rectangle pixels and minus black rectangle pixels. The Haar eigenvalue reflects the gray level changes of the image. However, the rectangular feature is only sensitive to some simple graphic structures, such as edges and line segments, so it can only describe structures with specific directions (horizontal, vertical, diagonal).



(a) Edge features

(b) Linear features



(c) Central features

(d) Diagonal features

Figure 7-1 Haar feature descriptor

As shown in Figure 7-2, for the characteristics as A, B and D, the characteristics numerical calculation formula is v = Σ white - Σ black. For C, the computation formula is as follows: v = Σ white - 2 * Σ black. The reason for multiplying the black area by 2 is to have the same number of pixels in the two rectangular areas.

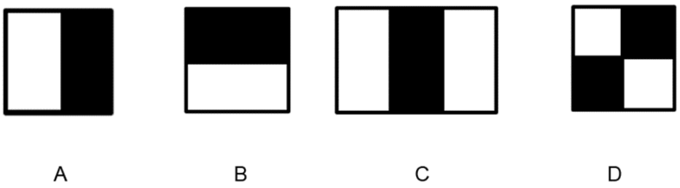


Figure 7-2 Haar feature descriptor

7.1.2 Adaboost classifier

The basic principle of the Adaboost algorithm is to combine multiple weak classifiers reasonably to make it a strong classifier, the single decision tree is generally used as weak classifiers. Adaboost adopts the idea of iteration. Each iteration only trains one weak classifier, and the trained weak classifier will participate in the next iteration. In the Nth iteration, there are a total of N weak classifiers, of which n-1 are trained before and their various parameters are not changed. The Nth classifier is trained in Nth iteration. The Nth weak classifier is more likely to correctly classify the data misclassified by the previous n-1 weak classifiers, and the final classification output depends on the comprehensive effect of the N classifiers.

The structure of the Adaboost classifier is shown as Figure 7-3, the dotted line in the figure represents the iteration effect of different rounds. In the first iteration, there is only the structure of the first row, in the second iteration, there are the structure of the first row and the structure of the second row, and each iteration adds a row of structure. The process of the round i iteration is as follows:

- Adding WeakClassifier(i) and WeakClassifier weight alpha(i).
- WeakClassifier(i) is trained by data set data and data weight W(i), and its classification error rate is obtained to calculate its WeakClassifier weight alpha(i).
- By using the method of weighted voting, the weighted voting of all the weak classifiers is used to get the final predicted output, and the final classification error rate is calculated. If the final error rate is lower than the set threshold value (such as 5%), the iteration is over. If the final error rate is higher than the set threshold, then update the data weight to get W(i +1).

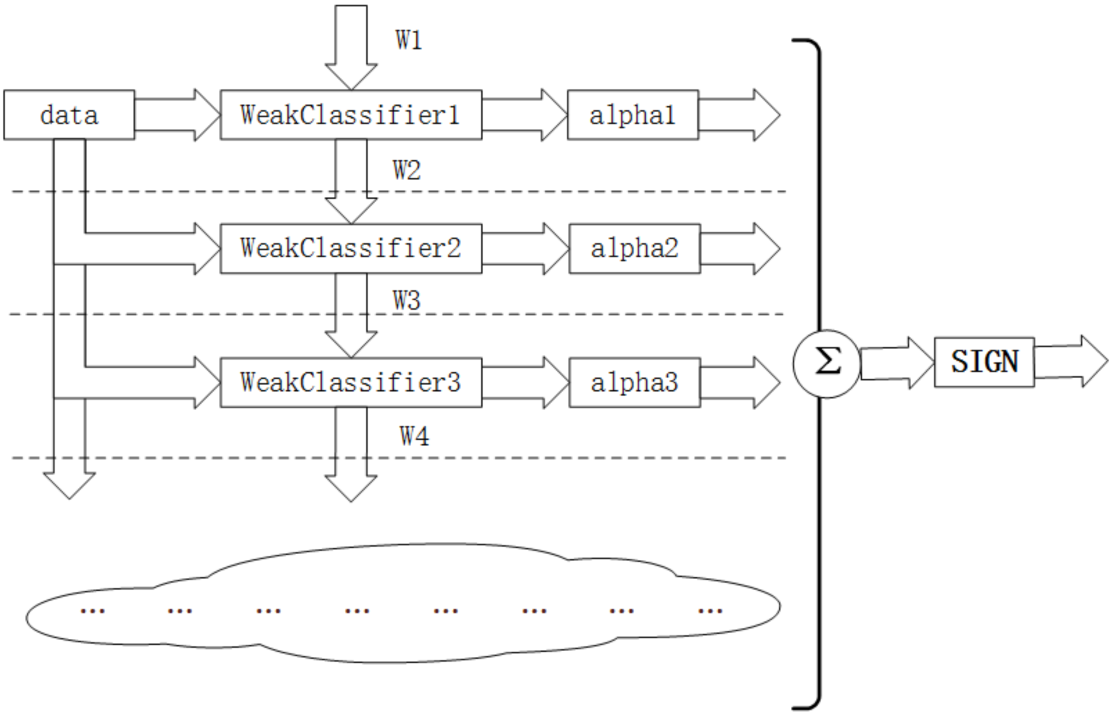


Figure 7-3 The structure of the Adaboost classifier

7.2 Detection and recognition method based on Support-Vector Machine (SVM)

The HOG (Histogram of Oriented Gradient) feature descriptor constitutes features through calculation and statistics of Histogram of Gradient direction of the local area of an image. The combination of HOG feature descriptor for feature extraction and SVM classifier for classification has been widely applied in image recognition, especially achieved great success in pedestrian detection. Subsequently, researchers applied this algorithm to traffic Logo [23] and TV Logo [24] detection and recognition.

7.2.1 Histogram of oriented gradients (HOG) feature descriptor

The HOG feature descriptor is based on the calculation of the normalized local directional gradient histogram in the dense grid. The basic idea of this method is as follows: no matter know or not the local gradient and the distribution position of the edge direction, it can describe the appearance and shape of the local target well.

In practical calculations, the image is generally divided into small cells, and then a one-dimensional gradient direction (or edge direction) histogram is calculated by accumulating in each cell. To ensure that the HOG feature descriptor has better invariance to the effects of lighting and shadows, the cells are usually formed into larger blocks to compare and normalize all the cells in the corresponding block. Then the HOG features of all blocks in the window which is going to be detected are combined to form the feature vector of the corresponding area, and SVM is used for classification finally.

During the detection and recognition, the window to be detected is divided into overlapping blocks, and then the HOG feature descriptor is calculated on these blocks, and the formed feature vector is put into the SVM for target classification finally. The detection window will use different scales to scan the entire image sequentially, and finally use the non-maximum suppression algorithm to merge the target area, thereby outputting the final detection target.

Compared with other feature description methods, HOG has the following two advantages:

- First of all, since HOG feature descriptor operates on the local element of the image during the calculation, it has good retention of the invariability of deformation in plane geometry and optics of the image, resulting in that these two kinds of deformation only appear in a larger space.

- Secondly, since the HOG feature descriptor adopts coarse spatial sampling, fine directional sampling and local normalization, it can allow subtle differences between the target area which is going to be detected and the real target area, and these subtle differences can be ignored by the HOG feature descriptor without affecting the detection effect.

7.2.2 SVM classifier

The SVM classifier is a two-class classification model whose essence is to calculate the maximization interval in the feature space. Due to the existence of the kernel function, SVM can be used not only as a linear classifier, but also as a nonlinear classifier. Since the learning strategy of the SVM classifier is to maximize the interval, the SVM classification problem can be transformed into a convex quadratic programming problem for solving, or into a regularized hinge loss function minimum problem for solving.

SVM classifiers can be divided into three categories according to the construction model from simple to complex: linear support vector machine in linearly separable case, linear support vector machine and non-linear support vector machine. The linear separable support vector machine is shown in Figure 7-4. In the two-dimensional feature space shown in Figure 7-4, "●" represents a positive example and " ○ " represents a negative example. When training a linear classifier, a lot of straight lines can be trained to correctly divide the two types of data. The linear separable support vector machine corresponds to the straight line that correctly divides the two types of data and has the largest interval.

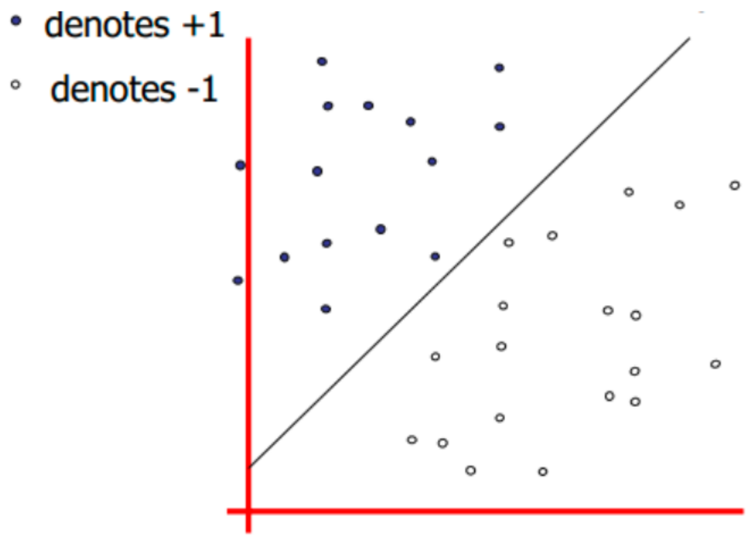


Figure 7-4 Linear support vector machine in linearly separable case

7.2.3 Deformable Part Model (DPM) algorithms

Deformable Part Model (DPM) is a component-based detection algorithm. This model was proposed by Felzenszwalb in 2008 [25] and published a series in CVPR, NIPS. He also won the PASCAL VOC Lifetime Achievement Award in 2010.

- Algorithm thought

Root filter+ Part filter：

The model shown in Figure 7-5 has an 8*8 resolution root filter (left) and a 4*4 resolution part filter (middle). The resolution of the middle image is twice the left image, and the size of the part filter is twice of the root filter so that the gradient is more detailed. The figure on the right shows its two-fold spatial model after Gaussian filtering.

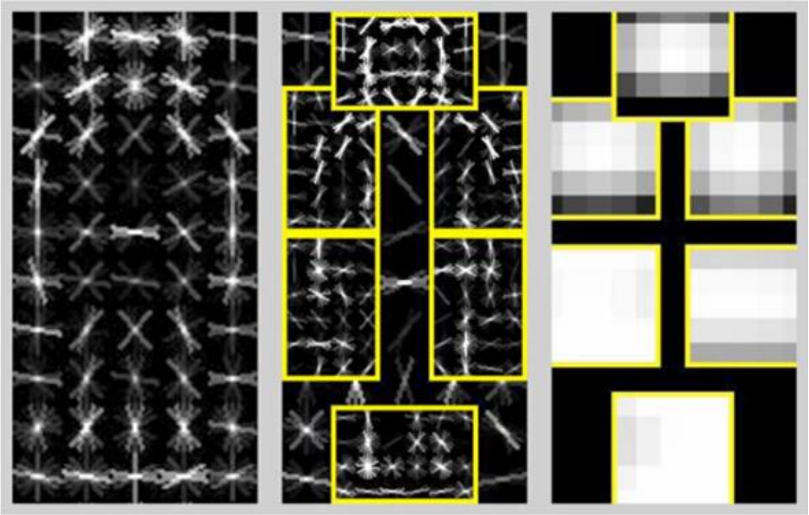


Figure 7-5 (Left)Rootfilter(middle) Part Filter (right) Gaussian filtered model

- Calculation of score

The formula of the score is as follows:

$$\mathrm{score}(x_0, y_0, l_0) = R_{0,l_0}(x_0, y_0) + \sum_{i=1}^{n} D_{i,l_0-\lambda}(2(x_0, y_0) + v_i) + b.$$

Formula 7-1

- DPM detection process:

As shown in Figure 7-6, the DPM feature map of every input image is extracted, and the original image is performed the Gaussian pyramid up-sampling, and the DPM feature map is extracted. The DPM feature map of the original image and the trained root filter are convoluted to obtain a response graph of the root filter. The 2-fold image of the DPM figure and trained Part Filter are convolved to obtain the Part Filter response map. In this way, the response map of the Root Filter and the response map of the Part filter can have the same resolution. The weighted average is then used to produce the final response graph. The higher light, the larger the response value.

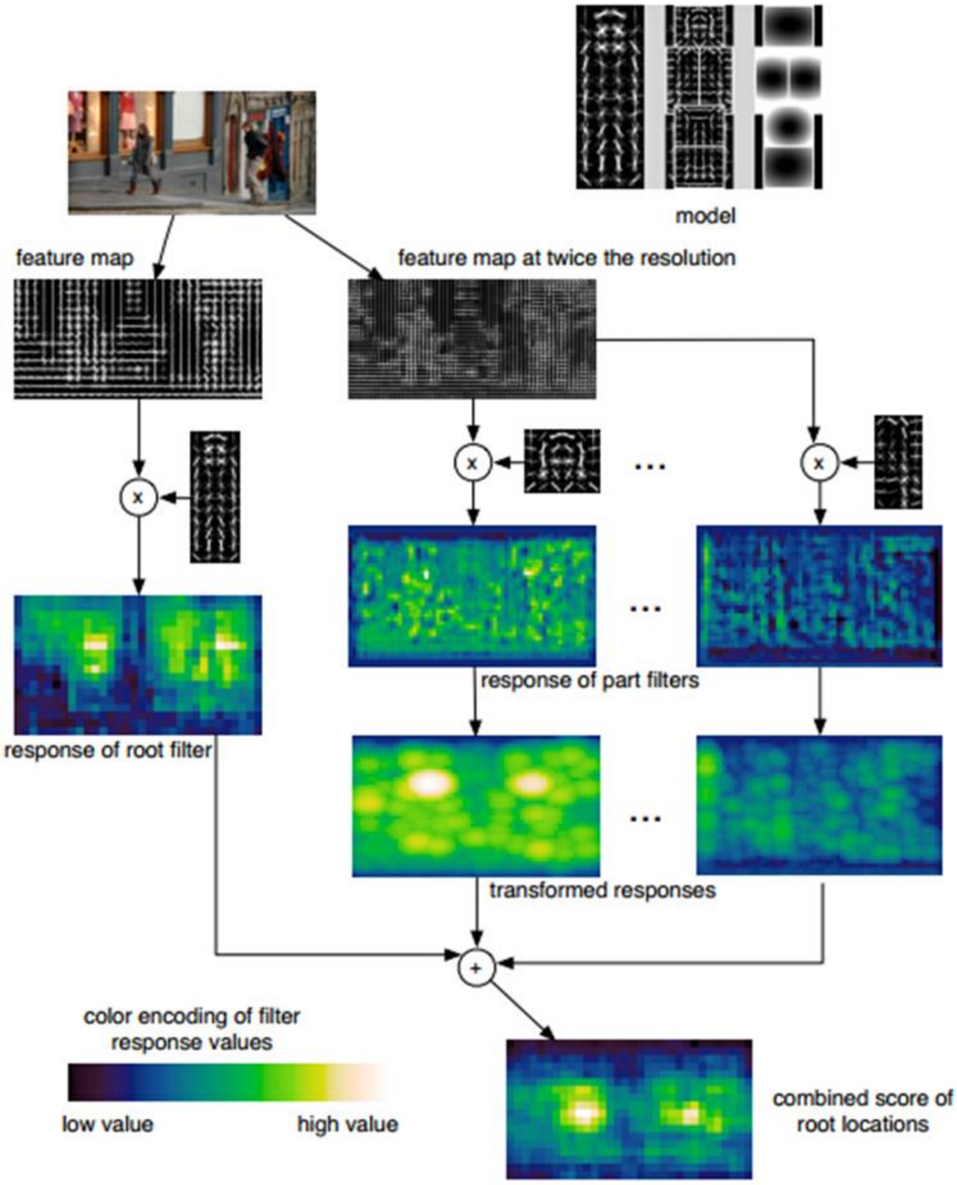
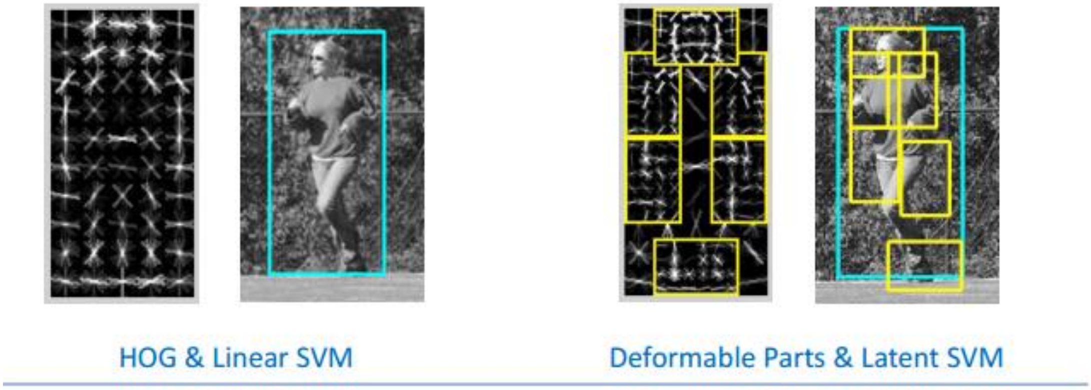


Figure 7-6 DPM detection process

- Latent SVM:

The difference between the traditional Hog+SVM algorithm and the DPM+LatentSVM algorithm is shown as Formula 7-2.



$$f_w(x) = w \cdot \Phi(x)$$

$$w = F_0$$

$$\Phi(x) = \phi(H(x), p_0)$$

$$w^* = \arg\min_{w} \lambda \|w\|^2 + \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i))$$

$$f_w(x) = \max_{z \in Z(x)} w \cdot \Phi(x, z)$$

$$w = (F_0, \ldots, F_n, a_1, b_1, \ldots, a_n, b_n)$$

$$\Phi(x,z) = (\phi(H(x), p_0), \phi(H(x), p_1), \ldots, \phi(H(x), p_n), \tilde{x}_1, \tilde{y}_1, \tilde{x}_1^2, \tilde{y}_1^2, \ldots, \tilde{x}_n, \tilde{y}_n, \tilde{x}_n^2, \tilde{y}_n^2)$$

$$w^* = \arg\min_{w} \lambda \|w\|^2 + \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i))$$

Formula 7-2

In the training sample, the negative sample set must be 100% correct and the positive sample set may have noisy. Because the positive sample is labeled manually, people may make mistakes and some labels are inaccurate. Therefore, the noisy data has to be removed. For the rest of the results, the gradient map of the training model is always divergent and irregular due to different angles and postures. It is also important to pick the data with the same posture, the samples which are nearest to the dividing line between the positive and negative samples. Samples that are very close to the dividing line are called hard examples. On the contrary, Easy-examples are further to the dividing line. The practical effect is shown as Figure 7-7:

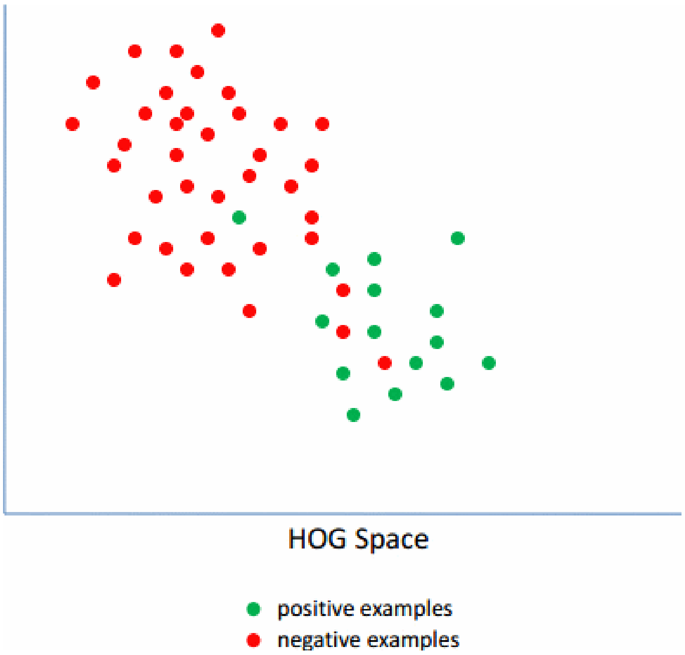


Figure 7-7 HOG Space

8 Detection and recognition methods based on deep learning

There are two key issues with conventional object detection and recognition technology: one is that the sliding window-based region selection approach is not targeted, and window redundancy contributes to high time complexity. Second, the manually designed feature has poor robustness in changing the diversity of the input picture. Since the accuracy of the conventional logo detection and recognition algorithm is difficult to fulfill all application requirements in real production and life, it is an unavoidable trend to study the new logo detection and recognition algorithm. With the impressive success of deep learning in the field of object detection and recognition in recent years, several researchers have applied deep learning to it.

There are two main reasons why the deep learning model can have strong detective characteristics. On the one hand, the deep learning model through independent learning of high-level features extracting to enhances the ability of characteristics expression. On the other hand, because the deep learning model can do feature extraction, feature selection and feature selection and feature classification fusion in the same model, implying the end-to-end training, enhancing the separability of features. Therefore, the hotspot of scientific research is the deep learning model for object detection and recognition. Object detection methods in the field of deep learning are divided into two categories: the detection and recognition method based on regional recommendations and the detection and recognition method based on regression.

(1) Object detection method based on regional recommendations. First, a large number of candidate boxes of targets are obtained by traversal the images through sliding Windows of different scales. Then, candidate boxes are classified to obtain accurate object bounding box and category. This method is called Two-Stage object detection algorithm because it is completed in two steps: obtaining candidate box and classifying candidate box.

(2) Object detection method based regression, the regression and classification of the boundary box are processed by training network directly. Because this method is completed in One step end-to-end, it is called One-Stage object detection algorithm.

The series of variants based on R-CNN are the Two-stage object detection algorithm, the series of variants based on Yolo and SSD are the One-stage object detection algorithm. The flow of the Two-Stage object detection algorithm is shown as Figure 8-1(a), and that of the One-Stage object detection algorithm is shown as Figure 8-1(b).

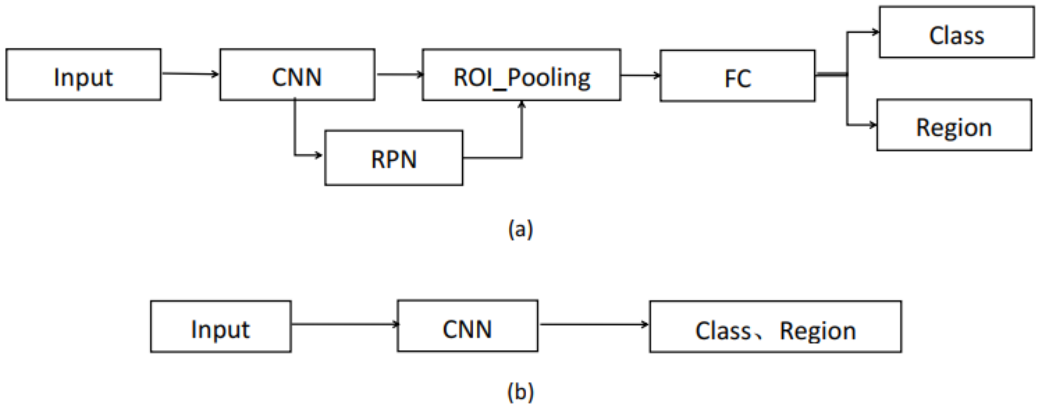


Figure 8-1 Flowchart of object detection algorithm

8.1 Detection and recognition method based on regional recommendation

8.1.1 R-CNN

Since the conventional detection algorithm relies on the sliding window mechanism which can cause high time complexity, so the Region Proposal is used to find the target location possible instead of the traditional sliding window mechanism, which can significantly reduce the proportion of invalid windows, and reduce the computational complexity of object detection, this is the idea of object detection algorithm based on the regional recommendation. The famous one is the Region-based Convolutional Neural Networks (R-CNN) algorithm [11].

R-CNN is to obtain a large number of candidate regions after processing the input images, and then obtain the features of the candidate regions through the convolutional network. According to the features, SVM classification is used to classify the category of the target, and finally, the overlapping candidate boxes are eliminated by Non-Maximum Suppression to obtain the accurate boundary boxes.

The R-CNN test results for the PASCAL VOC2007 dataset have been substantially improved compared to the previous algorithm. The mAP of R-CNN was explicitly increased to 66%. The structure of the R-CNN algorithm is as shown in Figure 8-2. There are 4 steps of the R-CNN algorithm: region proposals with selective search, the convolutional network feature extraction with CNN, classify features, and the regressor improves the bounding.

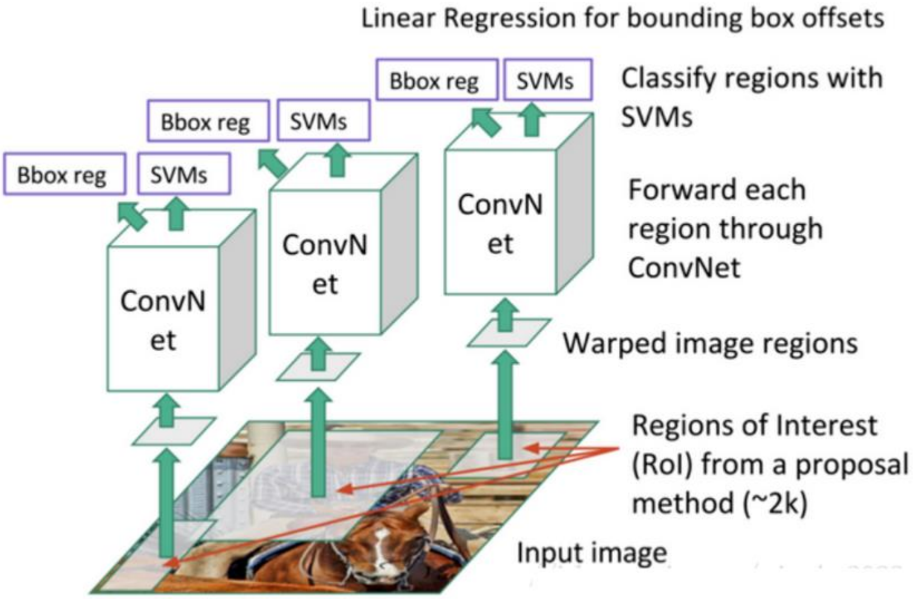

Linear Regression for bounding box offsets
Bbox reg
SVMs
Bbox reg
SVMs
Bbox reg
SVMs
Classify regions with SVMs
ConvN et
ConvN et
ConvN et
Forward each region through ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image

Figure 8-2 R-CNN algorithm
Source: https://www.codenong.com/cs106162355/ (2020)

8.1.1.1 Region proposal

Selective Search proposed by J.R.R. Uijlings [27] is used for the selection of candidate regions in the R-CNN model. The Selective Search algorithm aims to classify all possible target positions for object class recognition. Relative to the conventional single approach, selective search offers a range of search methods that can substantially minimize the space and time complexity of searches compared to sliding window searches.

Since obtaining candidate regions by the selective search algorithm are rectangular regions of different sizes, images of a fixed size are needed to be inputted in the CNN model, the image needs to be normalized.

8.1.1.2 Feature extraction

The AlexnNet network [28] and the VGG16 network [29] have been considered as the backbone network of the R-CNN model. By testing the VOC 2007 data set, AlexNet network mAP and VGG16 network mAP are 58.5 % and 66 %. However the calculation time of the VGG16 network was 7 times that of the AlexNet network, so the AlexNet network was adopted to explain in the paper. The extraction function of the AlexNet network architecture consists of 5 convolution layers and 2 fully connected layer layers. The number of neurons in Pooling Layer 5 is 9216, the number of neurons in fully Connected Layer 6 and fully Connected Layer 7 is 4096. After training, the features are extracted from the network. Each input candidate box will get an eigenvectors with 4096 dimensions.

8.1.1.3 Category judgment

The SVM classifier is used in the R-CNN algorithm. Since SVM is a two-class classifier, the N SVM classifiers need to be trained (N represents the number of target classes to be detected) for the classification. During CNN training, because CNN training is easy to overfit, it needs a lot of training data, all labeling of training data is loose, the candidate areas containing only part of the target object will be viewed as positive samples.

8.1.1.4 Improving the bounding

The essence of improving the bounding is to train a linear regression model to scale and shift the candidate regions in both horizontal and vertical directions. Candidate regions after position refining have higher IoU values and are more SVM-friendly, which could increases the accuracy.

8.1.2 Fast R-CNN

Fast R-CNN makes an improvement on the basis of R-CNN and inputs the candidate box and the whole image into the convolutional network. The convolutional layer learns the full image features [15] , generates the global feature map, and then uses the ROI Pooling layer to intercept the corresponding candidate box regional features on the global feature map.

Finally, classification is carried out through the fully connected layer, which can avoid the process of repeated convolution to obtain feature vectors for each candidate region in R-CNN, and effectively improve the detection rate.

8.1.3 Faster R-CNN

Faster R-CNN is modified on the basis of Fast R-CNN, eliminating the process of generating candidate boxes, directly generating the boundary boxes and global feature maps of candidate regions through an RPN network, and then classifying and modifying the boundary boxes of candidate regions by using the fullly connected layer. Faster R-CNN achieves the purpose of obtaining candidate boxes through RPN network, which improves the algorithm efficiency again.

8.2 Detection and recognition method based on the regression algorithm

The regional recommendation-based detection and recognition algorithm makes full use of the advantages of a convolution neural network in the field of image processing. However, due to the complicated layout of the network and the long-running time, it cannot be used in real-time detection. It is difficult for many algorithms to consider both accuracy and running time so they effectively achieve real-time detection and identification requirements by sacrificing accuracy to a large extent. To solve these problems, the researchers revamped the structure of the convolution neural network for the object detection and recognition algorithm. They also suggested that use the neural network as a regressor and the entire image which is going to be detected as a candidate region, as the input of the convolutional neural network. In this report, these methods are collectively referred to as regression-based object detection algorithms.

The method of using convolutional neural network to regress the target position information to the image to be detected was first proposed by Szegedy et al [13]., setting the output layer of AlexNet as a regression function and using part of the image to be detected as the input of convolution neural network. Then a binary mask image is generated, and then the target position and frame are judged by the mask image.

Based on the idea of a regression method for object detection and recognition proposed by Szegedy et al., Erhan [30], Girshick [31], Wan Li [32], Li Xudong [33], Redmon [17], Liu Wei [18] and other researchers, have researched this type of method. In particular, the Yolo algorithm proposed by Redmon dramatically improves the speed of goal positioning by combining the same category and intersecting boundary frames. The model SSD proposed by Liu Wei et al. uses different features in different locations, which dramatically increases the speed and precision of detection.

The Yolo algorithm and the SSD algorithm have good performance when it comes to real-time detection. The detection and recognition system based on the regression algorithm and the regional recommendation detection method has a different recommendation process mechanism. The detection and recognition method based on the regional recommendation must obtain the candidate regions firstly and then perform the extraction and classification of features, while the detection and recognition method based on the regression algorithm uses the integrated training idea and the scoring mechanism for the direct classification and regional box regression.

8.2.1 You Only Look Once: Unified, Real-Time Object Detection (Yolo)

The core idea of Yolo is to divide the image into regions for prediction. Yolo divides the input image into several grids with the unit of 32×32. For example, if there are n input images with the size of [416,416], then the output feature map size of Yolo is 13×13, as shown in Figure 8-3.

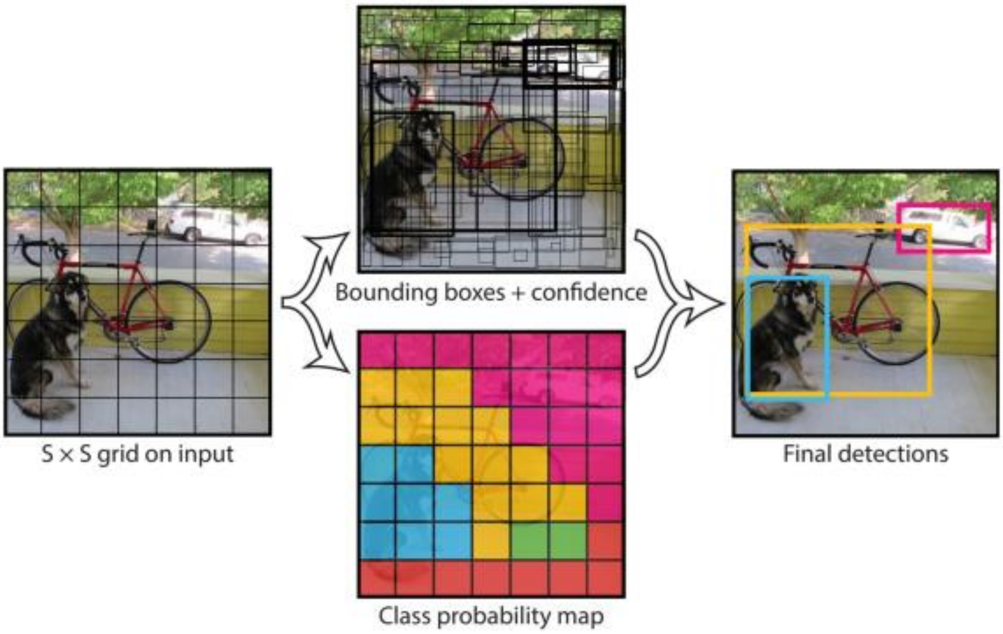


Figure 8-3 Yolo
Source: Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

When the coordinate of the center point of the detected target is located in a grid of the feature map, the grid is responsible for outputting the category confidence and regression bounding box of the object. Each grid can set the number of regression bounding boxes and the number of object categories. If the number of bounding boxes to be predicted in each grid is B and the number of categories is C.

Then the output tensor of the final convolutional layer is $CONV_{output} = [n, 13,13, (B \times 5 + C)]$, n is the number of input images, and $B \times 5$ is the parameters of the regression bounding box $[x, y, h, w, confidence]$, including the center offset x and y of the bounding box, the height h and width w of the feature map, and the confidence of the bounding box, C represents the classification confidence of the detected target corresponding to each category. $CONV_{output} = [n, 13,13, (B \times 5 + C)]$ is input to the fully connected layer for outputting the bounding box and confidence of the target.

8.2.2 Yolov2

Yolov2 improved on Yolov1. First, using the concept of candidate box of R-CNN, five candidate box sizes were obtained by K-means clustering according to the bounding box of training data. In the prediction process, the size of the bounding box was not predicted directly, but the compensation was predicted according to the size of the candidate box. Finally, the corresponding candidate box size is used to fit the target boundary box according to the compensation value when analyzing the output tensor. Yolov2 bounding box regression principle is shown as Figure 8-4. t_x, t_y is the coordinate offset of the bounding box center point, which is the output of the model. t_w, t_h is the width and height of the bounding box, which is the output of the model. c_x, c_y is the grid coordinate. p_w, p_h is the width and height of the candidate box.

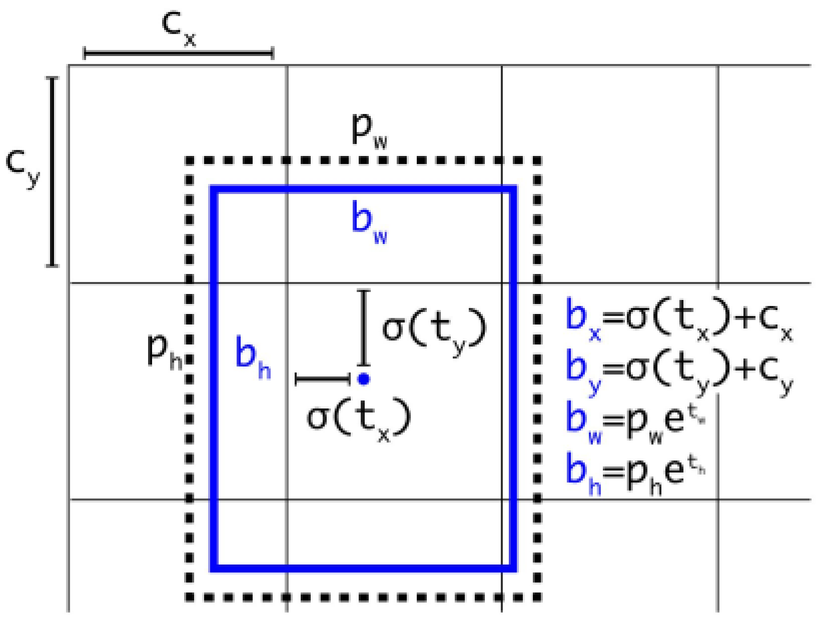


Figure 8-4 Yolov2 bounding box regression principle

Yolov2 also innovated in the backbone network. Based on VGG16, the Darknet-19 network structure was proposed and the fully connected layer was removed. Compared with Yolov1, the number of network parameters was greatly reduced. In addition, the Batch Normalization layer was added after each convolution layer to normalize the feature map, which greatly improves the generalization ability and robustness of the algorithm, and avoids the gradient explosion. Compared with Yolov2, Yolov1 improves the detection accuracy and speed greatly.

However, the detection performance of small objects is not as good as SSD due to its large output feature map.

8.2.3 Yolov3

Yolov3 mainly improved 2 sections based on the Yolov2. On the one hand, it uses Darknet 53 (with 52 convolutional layers) as the backbone network. Compared with Yolov2, the accuracy is improved, and the processing speed of the algorithm is also guaranteed.

Yolov3 proposed a pyramid structure, which is similar to the Feature Parymid Network (FPN). The feature maps output by the Feature Network were upsampled twice and then merged with the corresponding shallow Feature maps in the Feature Network, finally obtaining the multi-scale output tensor. In this way, the detection effect of small targets has been improved. The Yolov3 network structure is shown in Figure 8-5.

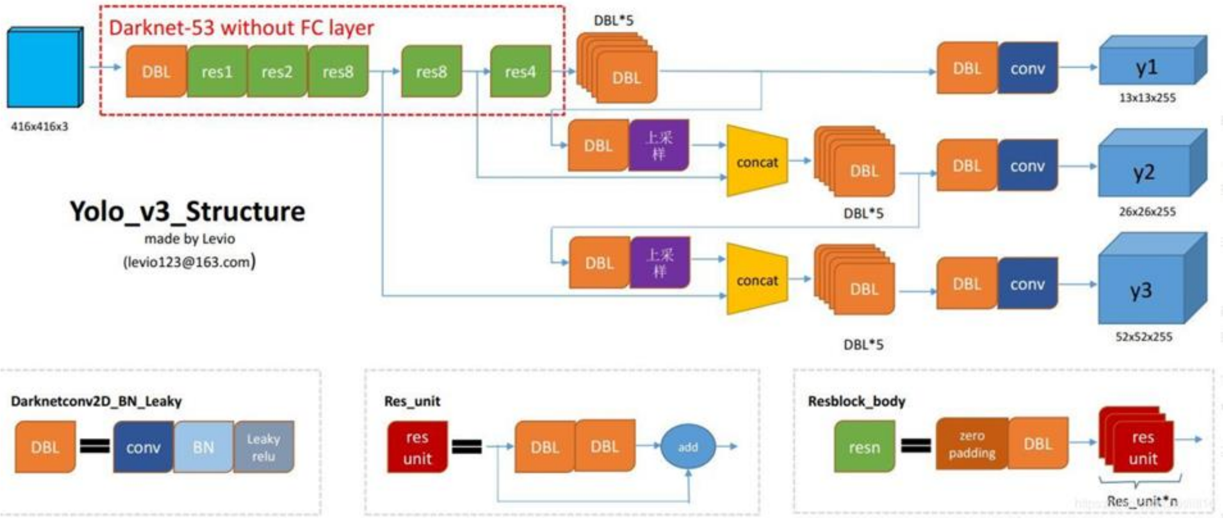


Figure 8-5 Architecture of Yolov3
Source: https://blog.csdn.net/leviopku/article/details/82660381(2018)

For example, if there are n 416×416 input images, then the size of the feature map output by the feature network is [n,13,13,c], n is the number of input images, c is the number of channels of the output feature map, and the size of the feature map after the first up-sample [n,26,26,c]. After the second up-sample, the size of the feature map is [n,52,52,c], then the output of the final network is the tensor whose size is [n,13,13,c], [n,26,26,c], [n,52,52,c], and the tensor is composed of three scales feature map.

Compared with Yolov2 and Yolov1, which only have a scale of [n,13,13,c] feature map, Yolov3 has two more detailed scale feature maps, so the detection effect of small targets has been improved.

8.2.4 Single Shot MultiBox Detector (SSD)

SSD is a One-Stage object detection algorithm proposed after Faster R-CNN and Yolo. SSD is a single-shot detector. It predicts the boundary boxes and classes directly from feature maps in one single pass. The architecture of SSD is shown as Figure 8-6. The backbone feature extraction network of SSD is the traditional image classification network, such as VGG16. If there are n 300×300 images. Then the size of the output feature map will be [n,38,38,512], and n 38×38 $feature\ map_1$with 512 channels of will be obtained. Then, the convolution whose stride is 2 is used to down-sample $feature\ map_1$ to obtain the $feature\ map_2$ with the size of [n,19,19,1024]. Continuing to down-sample to obtain the $feature\ map_3$ with the size of [n,10,10,512]. Then down-sample continuously, the [n,5,5,256] $feature\ map_4$ and [n,3,3,256] $feature\ map_5$ and [n,1,1,128] $feature\ map_6$ are obtained.

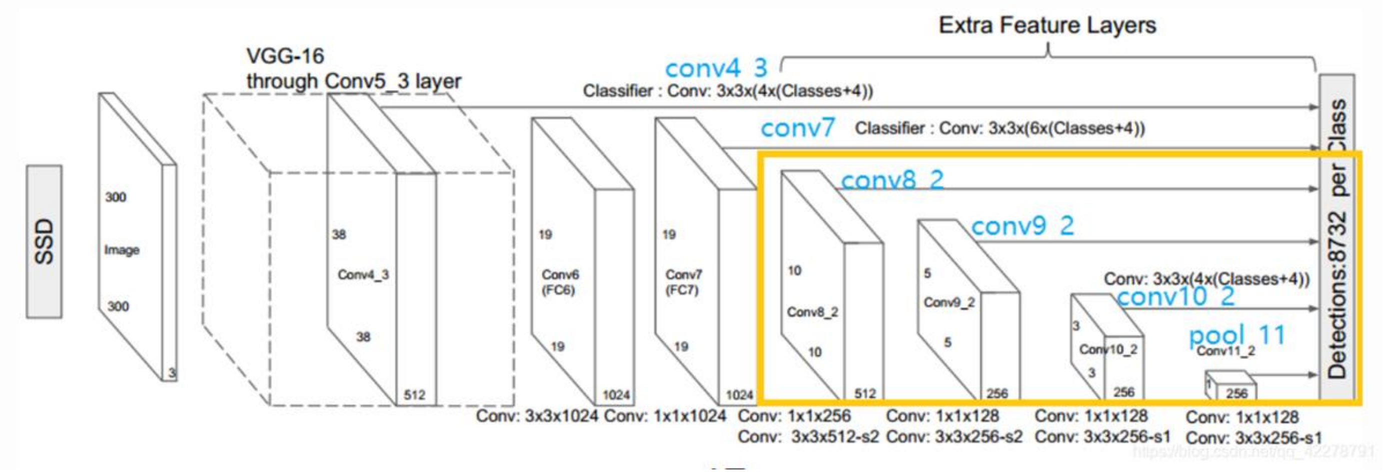


Figure 8-6 Architecture of Single Shot MultiBox Detector

9 Conclusion

Popular models of deep learning and two types of object detection methods are briefly introduced in this report, detection and identification methods based on traditional methods and detection and identification methods based on deep learning.

The accuracy of conventional algorithms depends heavily on the manually selected features, and the ability of conventional algorithms to detect different object categories is poor. Detection method based on deep learning can learn high-level features automatically, solving the problem of traditional algorithms relies heavily on artificial selection features. However, the deep learning model is prone to over-fitting, so a large training data set is needed, which is a very serious challenge for a situation with fewer training data sets.

While the detection and recognition method based on deep learning has made good progress in the field of object detection and recognition, there are still some issues that need to be solved. On the one hand, while current detection and identification methods based on the deep learning model have good results, how to set the training parameters relies mainly on experience and practice. On the other hand, in the field of multi-scale and multi-category object detection, there is still much room for the promotion of the deep learning model, which needs to be addressed by developing a more reasonable network structure.

Bibliography

[1] Rublee, E., Rabaud, V., Konolige, K. and Bradski, G., 2011, November. ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision* (pp. 2564-2571). Ieee.

[2] Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), pp.91-110.

[3] Bay, H., Tuytelaars, T. and Van Gool, L., 2006, May. Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404-417). Springer, Berlin, Heidelberg.

[4] Papageorgiou, C.P., Oren, M. and Poggio, T., 1998, January. A general framework for object detection. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)* (pp. 555-562). IEEE.

[5] Dalal, N. and Triggs, B., 2005, June. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (Vol. 1, pp. 886-893). IEEE.

[6] Ojala, T., Pietikainen, M. and Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, *24*(7), pp.971-987.

[7] Hinton, G.E., Osindero, S. and Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), pp.1527-1554.

[8] Hinton, G.E., 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, *14*(8), pp.1771-1800.

[9] Hinton, G.E. and Zemel, R.S., 1994. Autoencoders, minimum description length and Helmholtz free energy. In *Advances in neural information processing systems* (pp. 3-10).

[10] LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E. and Jackel, L.D., 1990. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems* (pp. 396-404).

[11]Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).

[12]Vaillant, R., Monrocq, C. and Le Cun, Y., 1994. Original approach for the localisation of objects in images. *IEE Proceedings-Vision, Image and Signal Processing*, *141*(4), pp.245-250.

[13]Szegedy, C., Toshev, A. and Erhan, D., 2013. Deep neural networks for object detection. *Advances in neural information processing systems*, *26*, pp.2553-2561.

[14]He, K., Zhang, X., Ren, S. and Sun, J., 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, *37*(9), pp.1904-1916.

[15]Girshick, R., 2015. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[16]Ren, S., He, K., Girshick, R. and Sun, J., 2016. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, *39*(6), pp.1137-1149.

[17]Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

[18]Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016, October. Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.

[19] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp.2278-2324.

[20] LeCun, Y., 1998. The MNIST database of handwritten digits. http://yann. lecun. com/exdb/mnist/.

[21] Hao, K., Wang, Y. and Ding, Y., 2010, March. Recognition of Robot Logo Based on Haar-Like Features. In *2010 International Conference on Measuring Technology and Mechatronics Automation* (Vol. 2, pp. 992-995). IEEE.

[22] Freund, Y. and Schapire, R.E., 1996, July. Experiments with a new boosting algorithm. In icml (Vol. 96, pp. 148-156).

[23] Llorca, D.F., Arroyo, R. and Sotelo, M.A., 2013, October. Vehicle logo recognition in traffic images using HOG features and SVM. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)* (pp. 2229-2234). IEEE.

[24] Ye, F., Zhang, C., Zhang, Y. and Ma, C., 2013, June. Real-time TV logo detection based on color and HOG features. In *2013 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)* (pp. 1-5). IEEE.

[25] Felzenszwalb, P., McAllester, D. and Ramanan, D., 2008, June. A discriminatively trained, multiscale, deformable part model. In 2008 IEEE conference on computer vision and pattern recognition (pp. 1-8). IEEE.

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

[27] Uijlings, J.R., Van De Sande, K.E., Gevers, T. and Smeulders, A.W., 2013. Selective search for object recognition. *International journal of computer vision*, *104*(2), pp.154-171.

[28] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), pp.84-90.

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

[30] Erhan, D., Szegedy, C., Toshev, A. and Anguelov, D., 2014. Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2147-2154).

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

[32] Wan, L., Eigen, D. and Fergus, R., 2015. End-to-end integration of a convolution network, deformable parts model and non-maximum suppression.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 851-859).

[33]Li, X., Ye, M., Liu, D., Zhang, F. and Tang, S., 2016, July. Memory-based object detection in surveillance scenes. In *2016 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 1-6). IEEE.