电子科技大学
格拉斯哥学院
Glasgow College, UESTC

Final Year Project Report

Bachelor of Engineering

Deep clothes wearing classification in video

Student: Wang Ruixian

GUID: 2429194W

1st Supervisor:

2nd Supervisor:

2021-22

Coursework Declaration and Feedback Form

The Student should complete and sign this part

Student Name: Wang Ruixian	Student GUID:2429194
Course Code : UESTC4006P	Course Name : INDIVIDUAL PROJECT 4
Name of 1st Supervisor:	Name of 2nd Supervisor:
Title of Project: Deep clothes wearing classification in video	

Declaration of Originality and Submission Information

I affirm that this submission is all my own work in accordance with the University of Glasgow Regulations and the School of Engineering requirements Signed (Student) : 	
Date of Submission : 28/4/2022	
Feedback from Lecturer to Student – to be completed by Lecturer or Demonstrator	
Grade Awarded: Feedback (as appropriate to the coursework which was assessed):	

Lecturer/Demonstrator:	Date returned to the Teaching Office:

Abstract

At present, the apparel products are one of the commodities with the largest demand and category in China's e-commerce platform. Therefore, the sales volume of clothing and apparel products will directly affect the revenue of e-commerce platforms. At the same time, with the rapid development of network live broadcast and short video industries, the way of recommending relevant clothing purchase links in live broadcast or short video will provide a new traffic entrance for clothing sales on e-commerce platforms. Therefore, how to build a video-based clothing retrieval model to improve the accuracy of retrieval has become the key to improve the sales of clothing products.

According to the programme, clothing recognition and retrieval based on video, the first stage in this project is extracting the key frame from video, and then the traditional image detection method is used to clothing detection. In the terms of dataset, we chose the DeepFashion2 with large numbers of images and annotations, and we chose YOLO v3 model with outstanding classification effect as the network to detection clothing.

Keywords: Clothing suggestion, Clothing detection and retrieval in video, Clothing segmentation, Deep learning.

Acknowledgements

This graduation design is not only a breakthrough study for me, but also a summary of my past four years of university life. There was a time when I was the same young man who had just entered the campus with great enthusiasm, but after four years of settling down, apart from gaining age in vain, I was also like a new-born baby crying, sucking greedily at the breast milk and taking in knowledge. The results of this research could not have been achieved without any book, a lesson of instruction, or even a mistake or an argument. This essay is the best representation of what I have learned and experienced in four years.

I would like to firstly thank UST Glasgow for giving me this teaching platform, allowing me to grow wildly over the four years and expand my international horizons while learning. Secondly, I would like to thank my supervisor for his conscientiousness and patience in guiding me through my final project. I would then like to thank my other teacher for his care and companionship over the past four years, for his unfailing support and guidance when I made major mistakes in my life, he is the teacher I am most grateful to on my journey. Finally, I would like to thank my friends who have tolerated my mistakes time and time again, and who have shared in each other's happiness while also bearing the pain of school and life. They have been a great help to me in my studies, my work and my life. Finally, I would like to thank my family, my parents, brother and relatives, for their moral support and for often being the last shot of comfort when I have an emotional breakdown. Special thanks to my friends who were used for my test samples.

I hope that in my future research and work, I will continue to be rigorous, practical and conscientious, and contribute to society in a small way.

Contents

List of Tables…….

1 Introduction

In recent years, with the expansion of the target range of online shopping and consumers' preference for online shopping, the consumption mode of online retail has gradually become an important driving force of consumption growth in China. According to the Statistical Report on Internet Development in China released by The China Internet Network Centre in 2020, the retail sales of physical goods sold online accounted for 20.7% of the total retail sales of consumer goods in China. This shows that China's retail industry is gradually shifting to online channels. Among all online shopping commodities in China, the second clothing category is one of the commodities with the largest demand and the fastest pace of update and iteration. Data show that about 70.4% of Chinese consumers have online shopping experience in 2019. Therefore, how to transfer the real and effective information of clothing quality, material and style to consumers through the network has become an urgent problem for domestic platforms and their businesses to consider and solve.

Generally speaking, the traditional way of information transmission is mainly based on pictures and texts. This way of information transmission can transmit limited information, and the authenticity of its graphic description cannot be guaranteed to consumers. Network broadcast, short video and other media forms are rising with the continuous development of the Domestic Internet environment, which is characterized by strong real-time and interactive, and compared with the way of graphic description, the contents displayed will be more comprehensive and real. Therefore, at present, major e-commerce platforms are trying to build a new consumption mode through live broadcasting or short-sighted frequency band goods, so as to further promote people's consumption desire, so that online sales can have a new flow entrance.

Currently, the major domestic e-commerce platforms provide consumers with quick access to product information through either text search or image search. From the perspective of text search, it is generally assumed that the text entered by the consumer is the keyword of the product information, and then the system internally matches these keywords with the keywords describing the product in the database, returning exact results if the match is successful, or rough results if the match fails. The advantage of this retrieval method is that it is fast and the system overhead is relatively low, but the disadvantage is that it requires labelling of the product information in the platform, and as the size of the product information in the platform increases by orders of magnitude, labelling the product information at a detailed granularity becomes too time-consuming and labour-intensive.

From the perspective of image retrieval, a series of techniques and methods in computer vision, both traditional and deep learning based, are used to extract features from the image to be retrieved and obtain a feature vector representation of the image, which is then compared with the feature vector representation of the image in the image library to measure the distance and return the product corresponding to the nearest image, or a list of the nearest products. This method has two advantages for the platform: firstly, it brings a new traffic portal to the platform's product sales, and the results returned by the method are more in line with the users' needs for product retrieval than the textual retrieval method, as the images themselves contain more information than the textual descriptions, and the method does not require manual description of the images but only It is also possible to automate the feature extraction of images, regardless of the volume of data in the platform, as the method does not require a manual description of the image, but only a defined feature extraction method for the image. Of course, there are drawbacks to the existing image retrieval methods in the platform, such as the user inputting the image to be retrieved and the image in the image library of the platform are originally inter-domain, so the retrieval may be inaccurate due to reasons such as background, shooting angle, etc., or if the display of the clothing described in the image to be retrieved is seriously distorted or obscured, the retrieval results cannot be accurately returned to the user.

As one of the most diverse and difficult to retrieve products sold on e-commerce platforms, the exploration of retrieval methods for clothing and apparel will provide guidance and solutions for other areas within the platform. The image based retrieval method in the platform, also known as "image search", has the following feature vector extraction methods, taking the early clothing feature extraction method as an example, which extracts low-level visual features such as texture, colour and geometry of clothing in the image to obtain clothing descriptions, but the robustness of the features extracted in this way is generally not strong and its susceptible to garment occlusion, deformation or different illumination, such as feature extraction using one or more of Harris [1], SIFT [2] and HOG [3]. Current methods for apparel feature extraction have adopted deep learning based on convolutional neural networks, as deep convolutional networks have powerful feature extraction capabilities for low-level visual features as well as high-level semantic features of images, and the method is also widely used in other areas of society.

The application of deep learning technology in the field of clothing retrieval can greatly improve user satisfaction and reduce the time spent shopping for clothing, but relatively speaking the

current deep learning technology is not only useful in the field of clothing image retrieval, its feature extraction of video data and related technology can also contribute to the platform sales and improve the user experience. In the face of the current rise of short videos and live e-commerce, shopping recommendations based on short videos and live broadcasts will provide a new traffic portal for clothing and apparel products, and its application scenarios include the platform recommending the clothing on the anchor in real time when the user is watching a live clothing shopping broadcast, recommending the clothing of the person in the video when the user is watching a short video, and the user can upload a video containing a clothing display for retrieval on the platform.

This paper will therefore focus on the "apparel-based" approach. This paper will therefore go a step further than 'image search' and will explore the problem of recommending clothing and apparel items based on short videos or live streaming. Compared to a single image, the video shows the clothing in different angles and forms, so the information contained in the video is more comprehensive than a single image, thus giving us a new way to solve the problem of deformation, obscuration and partial display of the clothing itself. The main objective of this paper will be to extract the characteristics of the garments in the videos to help users retrieve their target garments more accurately within the platform.

Even though this paper studies video-based clothing retrieval technology, as video is composed of a series of video frames (pictures), both video-based and image-based retrieval technology should consider feature extraction by video frame. Therefore, the processing method of this paper is to decompose the video into a group of video frames, and then use the traditional image retrieval processing method for clothing detection, and then recombine the marked video frames into a complete video.

2 Related Work

This paper investigates deep learning-based clothing retrieval, and deep convolutional neural networks are the basis of the model in this paper. The various stages involved in garment retrieval, including garment region detection, garment semantic label classification, and image feature extraction, are all based on deep convolutional neural networks. This chapter will review the fundamentals of deep learning algorithms, including simple neural network models, convolutional neural networks, back propagation algorithms, and common models of deep neural networks.

2.1 Neural Network

A neural network, in this paper referred to as an artificial neural network, is a model of computing that mimics the network of neurons in the human brain. In this section will briefly introduce neural network models, give definitions and computational expressions for neuronal models, and introduce both single-hidden layer neural network and multi-layer neural network models.

2.1.1 Neuron Model

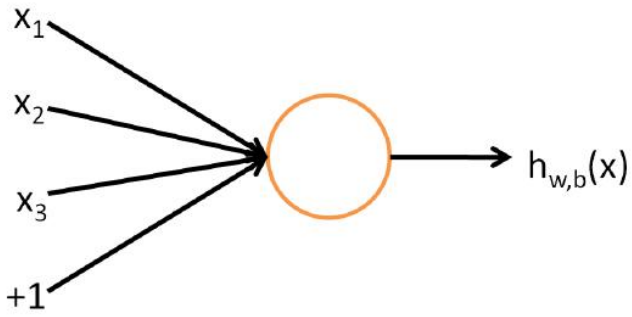

Figure 1 Single neuron model

The single neuron model is shown in Figure 1, where x denotes the input signal and the figure shows a three-dimensional vector ($x = [x_1, x_2, x_3]^T$), with the input vector plus the bias term (+1) as the input to the neuron, $h_{w,b}(x)$ denotes the output, and the computational The calculation equation is as follows.

$$h_{w,b}(x) = f(w^T x + b) = f\left(\sum_{i=1}^{3} w_i x_i + b\right) \#(1)$$

where $f(.)$ is called the excitation function.

For the supervised learning problem, we have a set of data $\left(x^{(i)}, y^{(i)}\right)$ and we wish to assume that the model $h_{w,b}(x)$ can approximate the real data label y. The sigmoid function is usually chosen as the excitation function, as shown in the following equation.

$$f(z) = \frac{1}{1 + exp(-z)} \#(2)$$

In addition to the sigmoid function, there are many other activation functions, such as the tanh function, ReLU (Rectified Linear Units) and so on

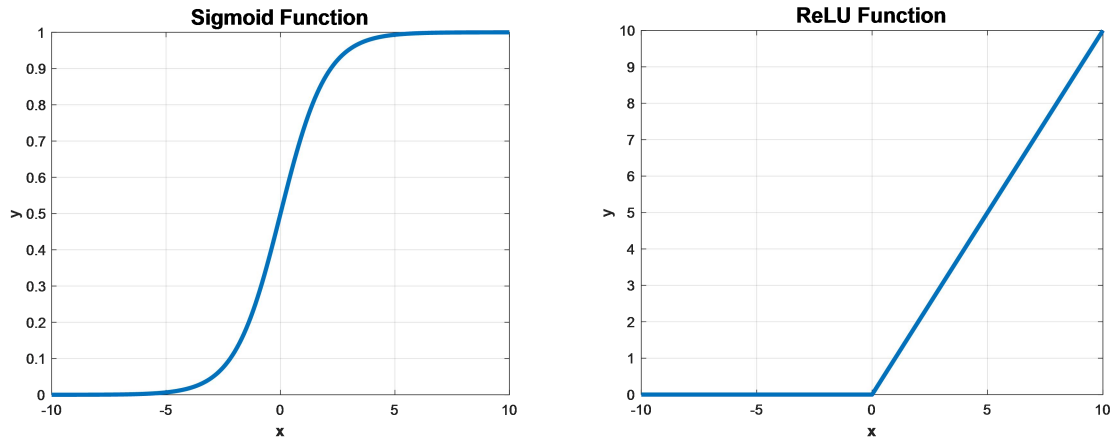

Figure 2 Two kinds of activation function (Sigmoid and ReLU)

Functions of sigmoid have the following properties.

(1)smooth, continuous differentiable functions.

(2)map the entire real number space between 0 and 1 and can represent probability values.

(3)The derivative is easy to find, $f'(z) = f(z)(1 - f(z))$.

2.1.2 Single Hidden Layer Model

Based on the single neuron model, we can "stack" multiple neurons to form a neural network, so that the output of some neurons is used as input to others. A simple single hidden layer neural network is shown in Figure 3.

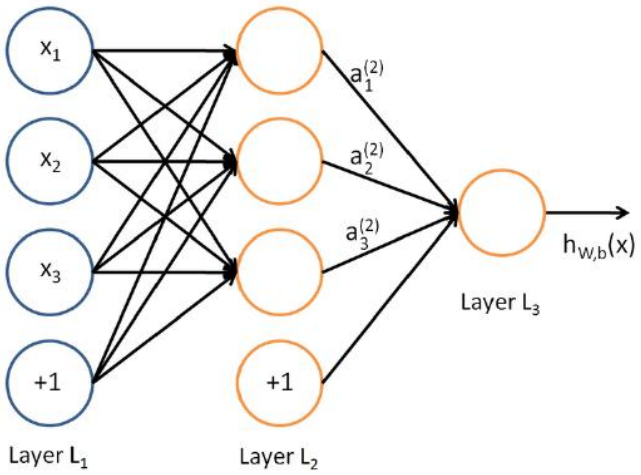

Figure 3 Single hidden layer neural network

The leftmost layer in the network is called the input layer, the rightmost layer is called the output layer, and the middle layer is the hidden layer. The example in Figure 3 has three neurons in the input layer (excluding the bias term), three neurons in the middle layer, and only one neuron in the output layer. We use $W^{(l)}$to denote the parameter (weight) matrix of the lth layer neuron to the $lth+1$ layer neuron $W_{ij}^{(l)}$denotes the weight of node j at layer l on node i at layer $l+1$. $b^{(l)}$denotes the vector of bias term parameters at layer $l+1$, and $b_i^{(l)}$denotes the bias term parameter of the i-th node at the $lth+1$ level. We also use $z^{(l)}$ to denote the weighted sum vector at layer l. and $a^{(l)}$ to denote the output vector of the lth layer weighted and after the excitation function. For the single hidden layer model in Figure 3, the network parameters are (W the network parameters are$(W,b)=\left(W^{(1)},b^{(1)},W^{(2)},b^{(2)}\right)$ and the following expression can be obtained.

$$z^{(2)}=W^{(1)}x+b^{(1)}$$

$$a^{(2)}=f\left(z^{(2)}\right)\#(3)$$

$$z^{(3)}=W^{(2)}a+b^{(2)}$$

$$h_{W,b}(x)=a^{(3)}=f\left(z^{(3)}\right)$$

2.1.3 Multi-Hidden Layer Model

The previous section described a single hidden layer neural network model, i.e., there is only one intermediate hidden layer in addition to the input and output layers. Artificial neural networks are designed to simulate the human brain to build computational models, which has roughly tens of billions of neurons, and thus neural network models in practice often have more than one layer. A neural network model with two hidden layers is shown in Figure 4.

Similarly, multi-hidden layer neural networks have greater than or equal to two hidden layers. The neuron nodes in each layer can be calculated according to the forward propagation equations 3 are calculated layer by layer. In general, the deeper the layers, the more complex the function that can be characterised by the neural network The higher the layers, the more abstract the features are compared to the lower layers.

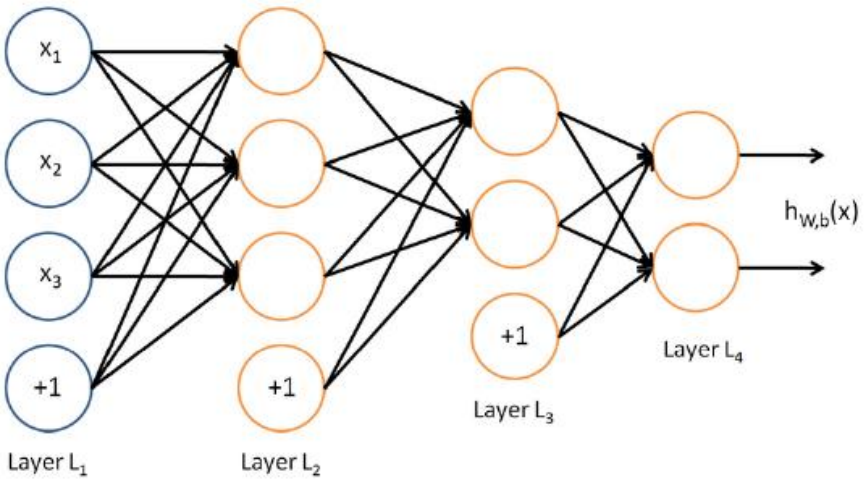

Figure 4 Double hidden layer neural network

2.2 Convolutional Neural Networks

Both the single hidden layer model presented in 2.1.2 and the multi-hidden layer model presented in 2.1.3 are fully connected networks, i.e., all neuron nodes in the upper layer contribute weights to all neuron nodes in the lower layer. Such a fully connected network has the following drawbacks: (1) Many weight parameters. A fully connected network with N neurons in the upper layer and M neurons in the lower layer will yield $M * N$ neurons. The more parameters there are, the slower the training and the more likely it is to overfit; (2) Fully connected networks have no translational invariance and are sensitive to local distortions

In contrast to fully connected networks, convolutional neural networks ensure some degree of translational invariance and deformation invariance.

(1) Localized sensory fields.

(2) Weight sharing.

(3) Spatial down sampling

Taking the input data as an image, a typical convolutional network is shown in Figure 5, where the green matrix is a three-channel image, the blue matrix is multiple convolutional kernels, and the last matrix is the output of the image after the convolutional kernels. Assuming that the size of the input image is $im\_h * im\_w$, the size of the convolution kernel is $k\_h * k\_w$, the number of convolution kernels is k_d, and the step size of the convolutional translation is s, the features obtained after the input image has gone through the convolution kernel are also a matrix, whose size is calculated as in Equation 4.

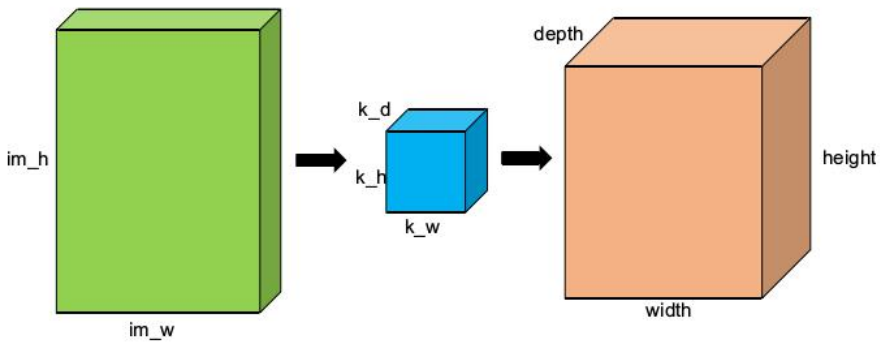

Figure 5 Convolution neural network

$$height = \left[\frac{im\_h - k\_h}{s}\right] + 1$$

$$width = \left[\frac{im_w - k_w}{s}\right] + 1 \#(4)$$

$$depth = k\_d$$

The size of the convolutional kernel corresponds to the size of the perceptual field, and the convolutional translation step controls the extent of down sampling, the larger the step, the greater the extent of down sampling. The number of convolution kernels is equal to the number of output feature maps. Convolution mimics the concept of the perceptual field of the human eye

and has the property of shared weights, hence translation invariance and local aberration invariance, and fewer parameters than fully connected networks.

2.3 Backward Propagation Algorithm

Whether it is a fully connected neural network as described earlier or a convolutional neural network, once the network parameters are determined, then given the input we can use the forward propagation algorithm to find the output layer by layer. The determination of the parameters requires the use of a back propagation algorithm, where the parameters are continuously updated on the training set until they finally converge.

Suppose our training set has m samples, i.e., {(x (1), y (1)), (x (2), y (2)), . . ., (x(m), y(m))}, one can Define the cost function for a single training sample as

$$J = (W, b; x, y) = \frac{1}{2}\left\|h_{W,b}(x) - y\right\|^2 \#(5)$$

For m training samples, the total cost function is

$$J(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m} J\left(W, b; x^{(i)}, y^{(i)}\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}\left(W_{ij}^{(l)}\right)^2$$

$$= \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{W,b}\left(x^{(i)}\right) - y^{(i)}\right\|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}\left(W_{ij}^{(l)}\right)^2 \#(6)$$

where n_l denotes the number of layers in the network and s_l denotes the number of neurons in the lth layer. The first term on the right-hand side of the equation is the mean square error term and the second term is the regularization term (or weight decay term, used to prevent overfitting). The parameter λ is used to control the relative importance of these two terms. The objective is to find suitable parameters W and b to minimise the aggregate cost function J (W, b). It is common practice to initialise the parameters W and b with a reasonable initialisation (e.g., random initialisation) and then iteratively update the parameters using an optimisation algorithm (e.g., batch gradient descent). The formula for one update iteration of the gradient descent method is as follows.

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b)$$

$$b_i^{(l)} = b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(W, b) \#(7)$$

where α is the learning rate. How to find the bias is the key, and the back propagation algorithm is an efficient algorithm for finding the bias, see Algorithm 2-1. $\delta^{(l)}$ is called the deviation term and is used to measure the 'contribution' of the nodes in the layer to the final output error. The symbol ⊗ denotes the Hadamard product, i.e., the corresponding elements are multiplied by each other. Assuming that the excitation function $f'(.)$ is a sigmoid function, $f'\left(z^{(l)}\right)$ can be obtained from $a^{(l)} \otimes \left(1 - a^{(l)}\right)$

Algorithm of back propagation

1: Using the forward propagation algorithm, calculate L2, L 3 until the output layer Ln

2: To the output layer, $\delta^{(n_l)} \longleftarrow \left(y - a^{(n_l)}\right) \otimes f'\left(z^{(n_l)}\right)$

3: **for** $l = n_l - 1 \to 1$ **do**

4: $\nabla_{W^{(l)}} J(W, b; x, y) \leftarrow \delta^{(l+1)} \left(a^{(l)}\right)^T$

5: $\nabla_{b^{(l)}} J(W, b; x, y) \leftarrow \delta^{(l+1)}$

6: $\delta^{(l)} \leftarrow \left(W^{(l)}\right)^T \delta^{(l+1)} \otimes f'\left(z^{(l)}\right)$

7: **end for**

2.4 Deep Neural Network

The previous sections introduced fully connected neural networks and convolutional neural networks, which form the basis of deep neural networks, as well as the most fundamental parameter iterative update algorithm for neural networks, the backpropagation algorithm. Deep neural networks generally refer to deep convolutional neural networks, where depth refers to the number of layers in the network. Deep networks fuse low, medium and high-level features by layering the network, and research has shown that the depth of the network is crucial, and that success in areas such as classification and recognition is largely due to deep models. This section will introduce the general structure of deep neural networks, namely deep convolutional neural networks, and the two common techniques of pooling and dropout.

2.4.1 Deep Convolutional Neural Networks

The general model of a deep convolutional neural network is shown in Figure 6. The structure of this network is essentially the same as Alex-Net, where the green cells represent the output of the convolutional layer, the red cells represent the pooling layer, and the blue cells represent the output after an excitation function (e.g., ReLU).

Figure 6 shows five convolutional layers (Layer 1 to Layer 5) and two fully connected layers (Layer 6, Layer 7). This network structure achieved first place in the ILSVRC 2012, and since then deep learning, especially deep convolutional neural networks, has been developing rapidly in various fields. Many famous network models including ZF-Net and VGG-Net [4] have similar multi-layer convolutional layers plus two fully connected layers. Since convolutional layers have features such as weight sharing and translation invariance compared to fully connected layers, GoogleNet [5] removed the fully connected layers and replaced them with pooling layers. Kaiming He et al [6] proposed a deep residual network and built a 152-layer deep convolutional neural network, which won the first place in several projects at ILSVRC 2015. Deep convolutional neural networks are an important model for deep learning.

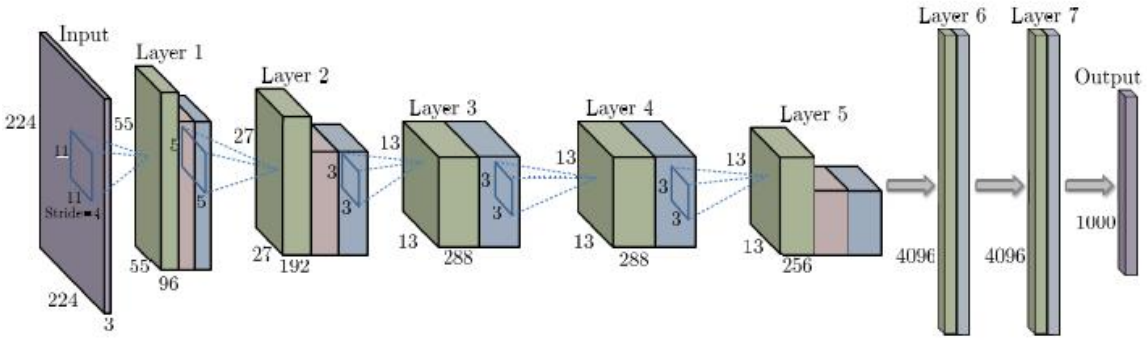

Figure 6 Deep Convolution Neural Network

2.4.2 Pooling

The pooling operation, mentioned in the previous section, is a common one in deep neural networks and is usually followed by a convolutional layer. Pooling is the process of maximising or averaging features over a certain region. Convolution can be used to obtain image features because of the 'statics' property of images, i.e., features that are useful in one local area may also be useful in another area. Similarly, in order to describe an image with a large image size, it is

natural to think of describing features in terms of aggregation statistics in different regions. Pooling is such a pooling statistic, where either max-pooling or average-pooling can be used to count features in a region of the image. In general, the aggregated statistics result in features that are less dimensional and less prone to overfitting. In addition, pooling is also translation invariant, e.g., when there is a small translation of the image, it does not affect the maximised pooled features.

To summarise pooling has the following characteristics.

(1) being translation invariant.

(2) Reducing the size of the next layer of inputs, reducing the amount of computation and the number of parameters.

(3) Prevention of overfitting.

(4) It is possible to obtain a fixed-length output [3].

2.4.3 Dropout

For deep neural networks, which are prone to overfitting with small amounts of data due to the large number of network parameters, dropout is another effective way to prevent overfitting, apart from adding a regularisation constraint (weight decay) term to the cost function, as proposed by Nitish Srivastava et al [7].

Dropout is a more technical approach. During training, several neurons and the forward and backward connections associated with these neurons are randomly dropped from the neural network model. The forward propagation algorithm is updated during the training phase of the network from Equation 8 below.

$$r_j^{(l)} \sim Beronoulli(p)$$

$$\tilde{a}^{(l)} = r^{(l)} \otimes y^{(l)}$$

$$z^{(l+1)} = W^{(l+1)}\tilde{a}^{(l)} + b^{(l+1)} \#(8)$$

$$a^{(l+1)} = f\left(z^{(l+1)}\right)$$

Where p is the dropout parameter, i.e., the percentage of randomly dropped neurons for a given layer, which can be determined by cross-checking or empirically by simply setting it to 0.5. In the testing phase after training, each W_{ij} needs to be updated to pW_{ij}

dropout works similarly to model fusion, but dropout is easier to implement than model fusion. The dropout technique has been shown to be effective in preventing overfitting and improving the performance of the model. dropout has been widely adopted by various deep neural network models since its introduction.

3 Method

3.1 Articulated Pose Estimation

In our study of clothing recognition in webcasts for online sales, the photos or videos are often taken or recorded by professionals, so the backgrounds are often not overly complex, and the quality of the images is clear. Even so, there is still a large amount of background-related content in these images, and we need less clothing-related data. Automating the visual analysis of garments through machine learning models is a more effective method than manual tagging.

In previous research, Yannis [8] advocates an approach that is divided into two main steps: 1) querying the category of clothing present in the video through classification detection and 2) using image retrieval techniques to find clothing items that are visually similar to the target, followed by categorisation. For example, as shown in Figure 1, the image on the left depicts a brightly coloured fashion model in a complex background, while the image on the right is a series of similar product images, i.e., the clothes worn by the model are presented on a white background. The product photos are often accompanied by a commentary on the category of clothing, making it easier for the customer to select.

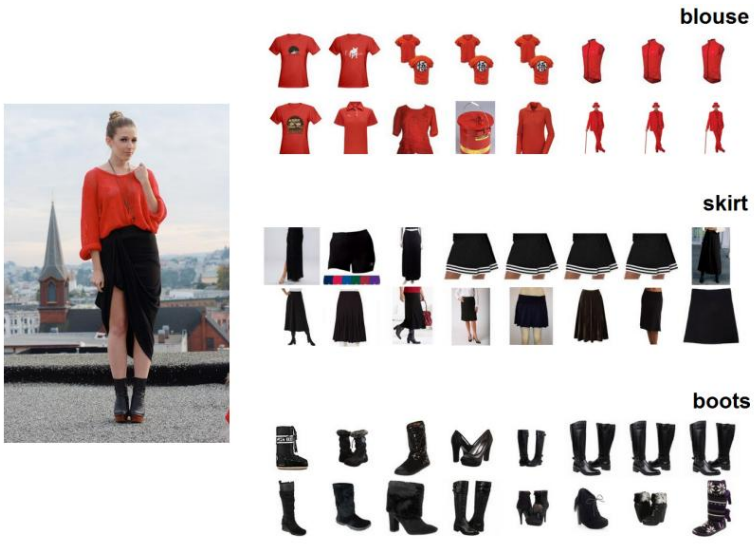

Figure 7 Image of model

Yannis' paper focuses on the then-popular idea of starting garment recognition with human pose estimation. When image I is successfully detected as a human pose, the system names a group of 26 parts of the body and displays them in a square area in the image space. The set of images is named Np (Np=26) and each part is represented as $P=p_1, p_2 \ldots, p_{Np}$. Each detected body part is

arranged in the same order, for example, the human head is often numbered p_1, p_2 . In order to easily distinguish each body part as well as the numbering, different coloured boxes are used to depict the pose estimates. They then used the Fashionista dataset to create a prior probability map of clothing, and after quantifying and normalising the pose estimation body parts, the most confident detection data was returned by keeping the examples with a prior probability greater than 0.5. The images were then segmented using Felzenswalb's popular segmentation method as well as selected approximate Gaussian mixture (AGM) clustering.

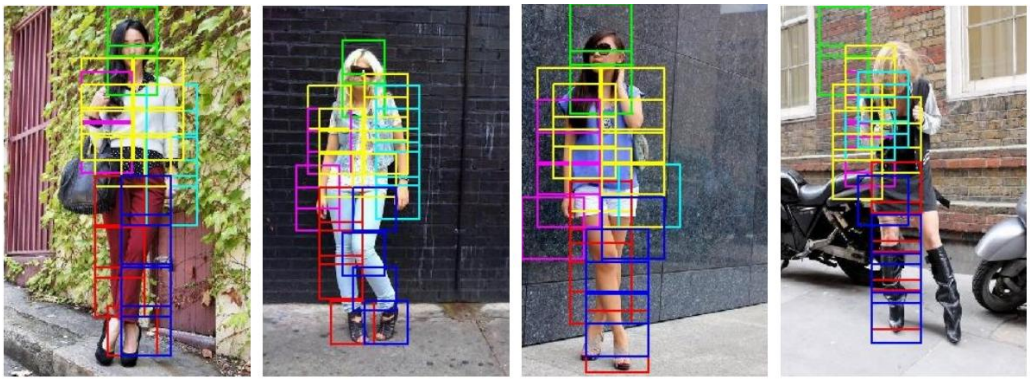

Figure 8 Pose estimation examples

3.2 Clothing prospect testing

As the first step in the garment image retrieval task, the task is to obtain the garment image region and the garment category in the image, which consists of two subtasks: garment classification of the image and target object location. The reason for foreground detection of clothing images is to remove the influence of the background part of the image on the image retrieval accuracy, as clothing images may be taken in different scenes and environments, so there may be some complex background.

Techniques such as garment key point detection and image segmentation are able to detect foreground areas of garments, but in this paper, we consider the simplicity and relative maturity of the corresponding techniques for target detection. Most of the early algorithms for detecting garment regions in images were based on various operators constructed by hand. Since most of the manually constructed features are low-level visual features and lack high-level semantic features, one can only design various complex features to ensure the robustness of the image representation, among which the typical algorithms include: HOG Detector [3], Viola Jones Detectors [9], Deformable Part-based Model [10] and other models.

For existing solutions, target detectors are generally classified into two categories, firstly, single-stage target detectors, which model the whole target detection task as a regression problem, resulting in an end-to-end model, represented by SSD [11], YOLO [12] series, etc. Then, two-stage target detectors, which are based on the idea of classification, firstly, based on the feature map in the detection process is of course accompanied by a little bit of regression, represented by the models of the RCNN [13-15] series.

Because single-stage based target detectors use a regression-based approach directly in the modelling of the model, the target object bounding box and its class are predicted in an end-to-end manner. The accuracy of the one-stage detector is relatively lower. Since YOLO's model is end-to-end with full convolution, it is very fast but slightly less accurate. The main improvements proposed by YOLO V2 are a stronger feature extraction network and the introduction of the Anchor concept in the Faster RCNN, but the size of the Anchor in YOLO V2 is determined by clustering. YOLO V3 adopted some of the more useful modules and techniques in target detection algorithms at the time, such as the FPN module and multiscale prediction of Anchor settings. The main contribution of YOLO V4 was the use of various effective data enhancements in processing model inputs and the use of various new modules in the backbone network, such as CSP module in CSPNet [16], and the Mish activation function, among others.

In the target detection task, the main task is to solve the two problems of what and where the target is, and thus the corresponding task can be described as classifying the objects in the image and obtaining the four coordinates of the target frame containing the target region in the image. For example, the low-level features extracted from the backbone network proposed in SSD can be used to detect small objects in an image because their field of perception is small and the information needed to be contained in a single pixel is still relatively small, while the high-level features extracted from the backbone network can be used to detect small objects in an image because their field of perception is large and because the network is non-linear. The higher-level features extracted from the backbone network can be used to detect large objects because of their large receptive field and the non-linearity of the network, and therefore the features are high-level semantic features, so they can be used to detect large objects with good results, while the lower-level features in the SSD have weaker information, which leads to the model being less effective in detecting small objects than expected. HyperNet [17], on the other hand, for the output of each layer of the network, where the feature maps of the front layers are generally

larger and therefore need to be pooled and reduced, while the feature maps of the back layers are smaller and therefore need to be up-sampled and expanded for the smaller features, followed by feature map fusion. This fusion approach allows for the simultaneous presence of low-level visual information as well as high-level semantic features, but it relatively loses some information in both spatial as well as semantic dimensions. The feature pyramid structure proposed in FPN, on the other hand Firstly, the features at each level have both low-level visual features and high-level semantic features, and then the features obtained at each level are used to make predictions independently, so that the information can compensate for each other to a certain extent. The second lies in the use of a stronger underlying network, i.e. by enhancing the feature extraction capability of the network through various methods, such as ResNet based on residual blocks, which to some extent solves the problem of neural networks being too deep to train, thus making the network deeper and better modelled, while DCN [18] gets a better underlying network from another perspective, which is invariant to the structure of the convolutional kernel itself The proposed solution is to give an offset to each one-pixel point in the feature map, which is learned through an additional convolutional structure, so that the whole network can learn information about the target offset, rotation and other deformations in the graph.

3.3 Dataset

The selection of the dataset for this paper is extensive, choosing DeepFashion, which is used by many scholars in global, and its updated version DeepFashion2, and discussing the advantages and disadvantages between the datasets

3.3.1 Fashion-MNIST

For the initial selection of the dataset, I chose the Fashion-MNIST dataset, which is widely used in the field of clothing inspection, in order to get a quick start on the operations and experiments in the field of clothing inspection. It is provided by the research arm of Zalando (a German fashion technology company). The Fashion-MNIST dataset is very similar to the MNIST dataset in that it is 60,000 trained and 10,000 tested, and the size of the images is also 28*28. However, MNIST is a handwritten number 0-9 classification, while Fashion-MNIST is a classification of clothing and contains images from 10 categories The images are t-shirt, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot.

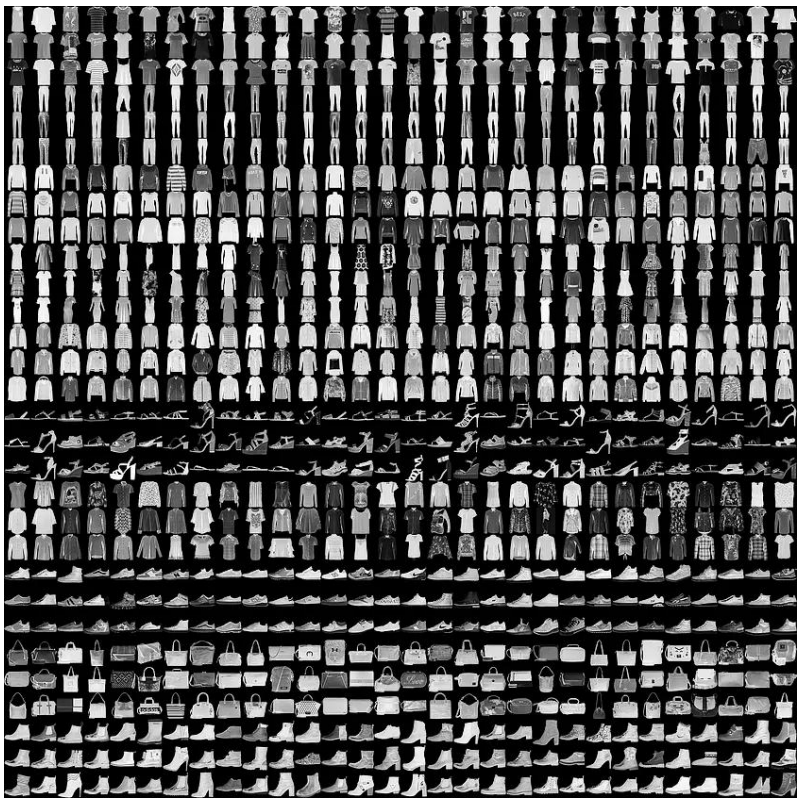

Figure 9 Image of Fashion-MNIST

3.3.2 DeepFashion

As Fashion - MNIST is only a 28*28 grayscale image, it is only an introductory test dataset and its number of images, image size and colours do not meet the requirements of this research. This dataset was created by the Multimedia Laboratory of the Chinese University of Hong Kong, which provides a large database of clothing, and the DeepFashion database, which has several properties that have attracted scholars to study it:

(1) Firstly, DeepFashion contains over 800,000 diverse fashion images, from well-placed shop images to unconstrained consumer photos, constituting the largest visual fashion analysis database.

(2) Secondly, DeepFashion is richly tagged with information about clothing. Each image in this dataset is labelled with 50 categories, 1000 descriptive attributes, bounding boxes and clothing landmarks.

(3) DeepFashion contains over 300,000 cross-pose, cross-domain image pairs.

(4) Four benchmarks were developed using DeepFashion data, including attribute prediction, consumer-to-store clothing retrieval, in-store clothing retrieval and landmark detection. The data and annotations from these benchmarks can also be used as training and test sets for the following computer vision services (e.g., clothing detection, clothing recognition and image retrieval).

3.3.3 DeepFashion2

However, in a paper published in cs.CV in 2019, Yuying Ge, Ruimao Zhang and other academics from the Chinese University of Hong Kong released an upgraded version of DeepFashion. Even though DeepFashion has been the most popular standard dataset among image detection scholars worldwide since its release, and is often used in several studies and papers, there are still some problems with the treadmill, such as only one type of garment being labelled in one image, no pixel-level garment labelling, and using the same key point description for each garment type. improves the above problems and further increases the dataset and implements four typical applications of clothing target detection, pose estimation, image segmentation and retrieval based on Mask R-CNN as a baseline on this dataset.

DeepFashion2[19] includes 491K images of 801K garment instances (two instances in one image, e.g., tops and trousers) in 13 categories. Each image description contains proportions (percentage of the figure), shading or not, size (out of bounds or not), perspective (yes/no human wear, positive/measured face), border, finer landmark, and pixel-level mask; the same cut, pattern, and style ones are divided into groups, and the data is divided into 43.8K groups. Images include user images and online shop product images, those pointing to the same product are considered a pair, and the dataset contains 873K pairs, 3.5 times more than DeepFashion. The effect of DeepFashion2 compared to DeepFashion is shown in Figure 10.

Figure 10 The comparison between 1 and 2

DeepFashion2 defines different landmarks for different categories, with 23 key points per landmark, higher than DeepFashion's 4-8 key points.

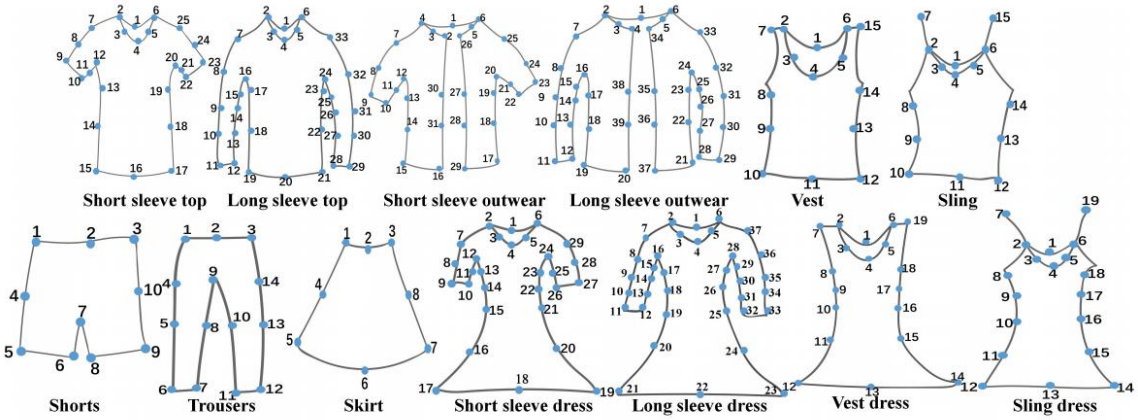

Figure 11 The pointed sample

Building the dataset

DeepFashion2 data was sourced from DeepFashion and the web. All product/user pair images in DeepFashion were added and all other images were removed; more product/user pair images were crawled from the web. Manual annotation removed images that were heavily obscured, low resolution, and too small. The richness of the images is shown below: Scale refers to the proportion of garments in the image, Occlusion refers to occlusion, Zoom-in shows whether the edges are out, and Viewpoint shows the angle.

Data annotation

DeepFashion has 50 categories, but more than half of them account for less than five thousandths of the total, and some are dichotomous. DeepFashion has organized them into 13 categories with no dichotomies and roughly balanced examples, as shown below.

As the different categories have different forms of distortion, for each category its relative pose is defined, i.e., the interaction of landmark silhouette and skeleton (e.g., the effect of the improved description of 'sling'). The mask is built in a semi-automatic way: it is first labelled according to the contours using an algorithm and then adjusted manually.

Building the model

The model consists of three parts: the FN for extracting features, the perceptual network PN and the similarity network MN. the same network is used to solve the four types of problems mentioned above. The model inputs are two images, I1 and I2. In the first stage, the two images are fed into the FN network separately to extract features. The underlying model of FN is ResNet50, which extracts the FPN features, and then uses Roi Align to extract the FPNs of each

region of interest, and extracts features at different levels of the pyramid for different targets at a later stage.

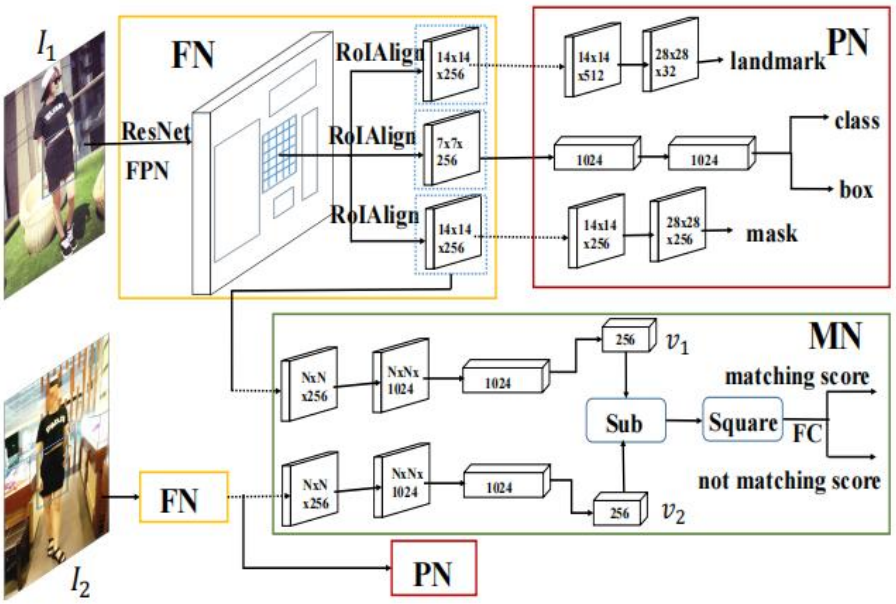

Figure 12 Match R-CNN

In the second stage, the features are fed into the perceptual network, the three branches of the perceptual network implement landmark key point annotation, locating the box where the garment is located to classify the content in the box class, and pixel-level image segmentation mask. In the third stage, the extracted features are fed into the similarity network MN, the feature extraction layer is trained to have good characterization ability for the garment image, the MN network consists of substract, square and a fully connected layer (to calculate the distance) for comparing two images to see if they match.

Objective functions

The objective function for the optimization is a combination of the above objectives.

$$min_{\Theta}\mathcal{L} = \lambda_1\mathcal{L}_{cls} + \lambda_2\mathcal{L}_{box} + \lambda_3\mathcal{L}_{pose} + \lambda_4\mathcal{L}_{mask} + \lambda_5\mathcal{L}_{pair} \#(9)$$

The weight of each error is controlled through the coefficient λ, where the similarity error is calculated as follows:

$$\mathcal{L}_{pair} = -\frac{1}{n}\sum_{i}^{n}[y_i log(\hat{y}_i) + (1-y_i)log(1-\hat{y}_i)] \#(10)$$

Y_i is 1 when two identical images belong to the same commodity, otherwise it is 0. The conditions for selecting RoI vary for different objectives. During retrieval, the clothing region with the highest confidence level is taken as the instance to be queried.

3.4 Related Theories

Target detection theory for images

Target detection is responsible for answering the questions of what the target object is and where it is in computer vision tasks. In recent years, technologies related to target detection have become widely used in various areas of society, for example, industry uses target detection to screen products for defects in order to increase their product yield, while pedestrian detection can be used to assist in automatic vehicle driving or for public security systems. For the apparel retrieval domain, foreground detection is required to get the area of apparel in the image. This is due to the fact that in clothing retrieval tasks, the image to be retrieved and the image library shooting scene is cross-domain, so it will cause the same clothing in different pictures on a variety of display, mainly including: a variety of background, shooting angle, the location of the clothing in the picture and the intensity of light, etc., so before the clothing retrieval need to use the target detection technology to remove the influence of factors such as image background. The foreground detection of clothing can be used to classify and locate the target object using currently used target detection methods. To date, deep learning-based target detection techniques include single-stage detection models such as the YOLO series, SSD, etc., and two-stage detection models such as the RCNN series.

3.5 YOLO V3

Single stage-based target detection models

A single-stage target detection model is an end-to-end neural network that is based on the idea of regression to classify targets and localise target areas, so that the network does not need to generate alternative frames when making inferences. SSD and YOLO, for example, are both classical end-to-end single-stage target detection algorithms. SSD, with its proposed feature pyramid structure and the corresponding full convolutional neural network structure, achieves higher accuracy and faster performance than other networks under the same conditions. The YOLO series, and the new YOLO network from The YOLO family has evolved from YOLO V1

to YOLO V5, and the performance of the corresponding target detection models has continued to improve as the model versions have been iterated.

YOLO V3[20] is one of the more effective target detection algorithms, as shown in Figure 12, which uses DarkNet53 as its backbone network and uses some of the leading-edge structures in target detection, such as anchor frames and FPN structures. YOLO V3 inherits the features of V1 and V2 and develops its own strengths. The structure of the fully convolutional network makes YOLO V3 fast inferring, and since there is no pooling layer in the network, YOLO V3 controls the size of the feature map output by adjusting the convolution step of the convolution kernel. In addition, the residual module of the ResNet network is borrowed from the backbone network Darknet53 to enhance the feature extraction capability of the network.

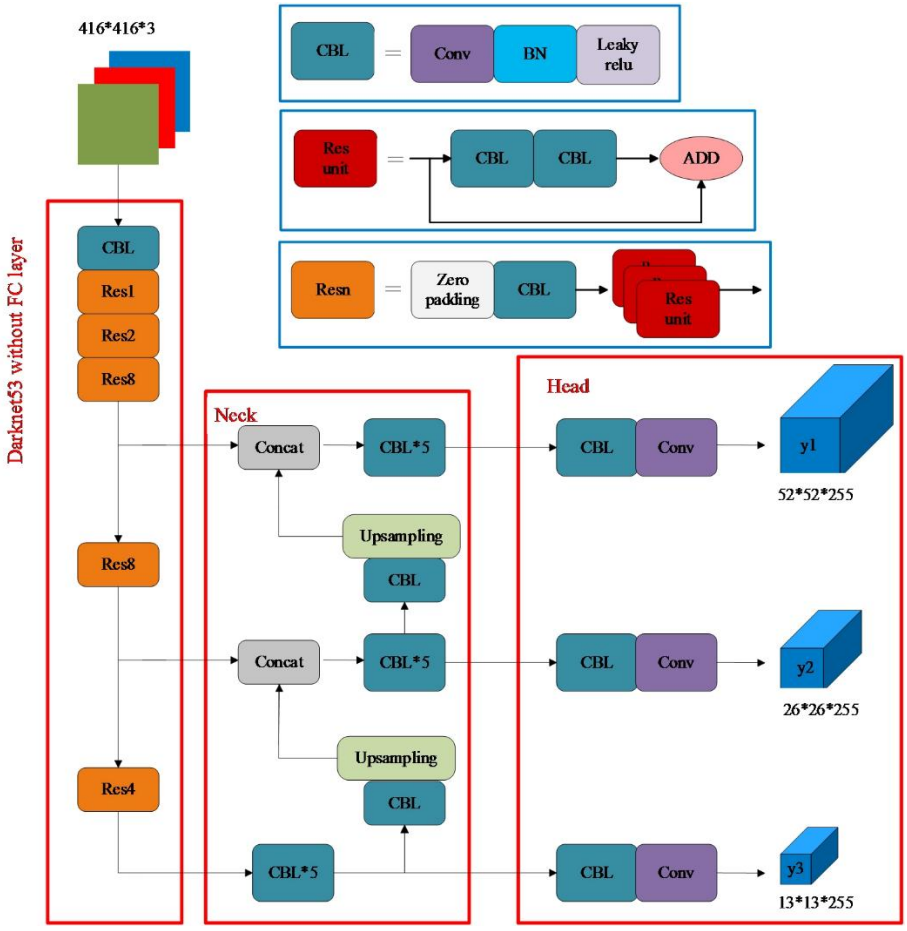

Figure 13 The structure of YOLOV3

In the FPN module of YOLO V3, the features extracted by the backbone network at different stages are used and the information from different layers is passed to each other so that the features extracted from different layers are more informative to predict targets of different sizes. In the prediction part, the output of YOLO V3 can be described as $[3 \times (4 + 1 + 80)] \times N \times N$, which means that for an image the feature map output is $N \times N$ in length and width, while for each pixel point on the feature map 3 Anchor boxes are used for detection, where each Anchor

box has a different size, and for each Anchor box 4 positions are output, 1 target The final result is 80 category probabilities.

The 80 is the number of categories of targets in the coco dataset, as shown in Figure 12. The three feature maps output by YOLO V3 are down sampled by multiples of 32, 16 and 8. As shown in Figure 2-1, features of different granularity are integrated by concatenation, i.e., the feature maps are stitched together by channel dimension, or sampled first if the other dimensions of the map are not consistent.

The key reason for the success of the YOLO family as an end-to-end neural network is the setting of the loss function. Although the loss function may be slightly different in different versions of YOLO, the focus is the same, including the loss of the predicted target class, the loss of the target box position and the loss of the predicted region as a target. The total loss is the sum of the losses of the 3 feature maps. As shown in Equation 11, where λ is the weight parameter that controls the weight between the target frame localisation, target category prediction and predicted target losses, and noobj indicates the absence of a target in the target frame.

$$L = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (2 - w_i \times h_i) [(x_i - \hat{w}_i)^2 + (y_i - \hat{y}_i)^2]$$

$$+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (2 - w_i \times h_i) \left[(w_i - \hat{w}_i)^2 + \left(h_i - \hat{h}_i\right)^2\right]$$

$$- \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(\hat{C}_i log(C_i) + \left(1 - \hat{C}_i\right) log(1 - C_i)\right)^2 \#(11)$$

$$- \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} \left[\hat{C}_i log(C_i) + \left(1 - \hat{C}_i\right) log(1 - C_i)\right]$$

$$- \sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c \in classes} \left[\hat{p}_i(c) log\left(p_i(c)\right) + \left(1 - \hat{p}_i(c)\right) log\left(1 - p_i(c)\right)\right]$$

4 Experimentation and Result

In the previous chapter we discussed the main algorithm designed in this paper, the network model and the dataset used. In this chapter, experiments will be designed to verify the effectiveness of the methods proposed in this paper. The experimental data, the experimental protocol, followed by the experimental results and an analysis of the results will be given in detail.

4.1 Dataset Exploration and Reanalysis

The dataset we have used, DeepFashion2[19], is a dataset that contains a larger number of images and types of clothing. The following table gives a comparison of the data from DeepFashion2 with several previous datasets.

	DARN	WTBI	DeepFashion	FashionAI	DeepFashion2
YEAR	2015	2015	2016	2018	2019
Images	183K	425K	800K	357K	491K
Categories	20	11	50	41	13
Bboxes	7K	39K	None	None	801K
Landmarks	None	None	120K	100K	801K
Masks	None	None	None	None	801K
Pairs	91K	39K	251K	None	873K

Table 1 Comparisons between the DeepFashion2 and previous datasets

4.2 Training with the YOLOv3 model

Firstly, we downloaded the dataset from the web, unpacked the data and obtained the train and validation folders. The annos folder is the annotation file and the image file contains the original image files. The DeepFashion2 annotation format was then converted to COCO format.

After completing the download of the YOLOv3 model, this implementation contains some training tricks, such as Cosine scheduler learning rate, Mosaic, CutMix, label smoothing, CIoU, etc. The just generated data label files train.txt and valid.txt were placed in the root of the project and the code was used to carve out a percentage of the training set as the validation set. In this paper, we used 15,000 garment images as the data set and the remaining 5,000 as the test data, so the ratio of the training set to the test set is 3:1.

4.3 Result

The system's ability to recognise single garments and multiple garments in everyday life after training with the YOLOv3 model will be demonstrated. Most of the images shown are taken from the author's life photos and were chosen to match the 13 clothing types specified by DeepFashion2 as far as possible. In order to distinguish between the different clothing types, the results are displayed with different coloured boxes circled and with the name of the clothing tested and the degree of certainty attached. To save time and improve efficiency, judgements with a confidence level of 0.5 or less are not shown automatically.

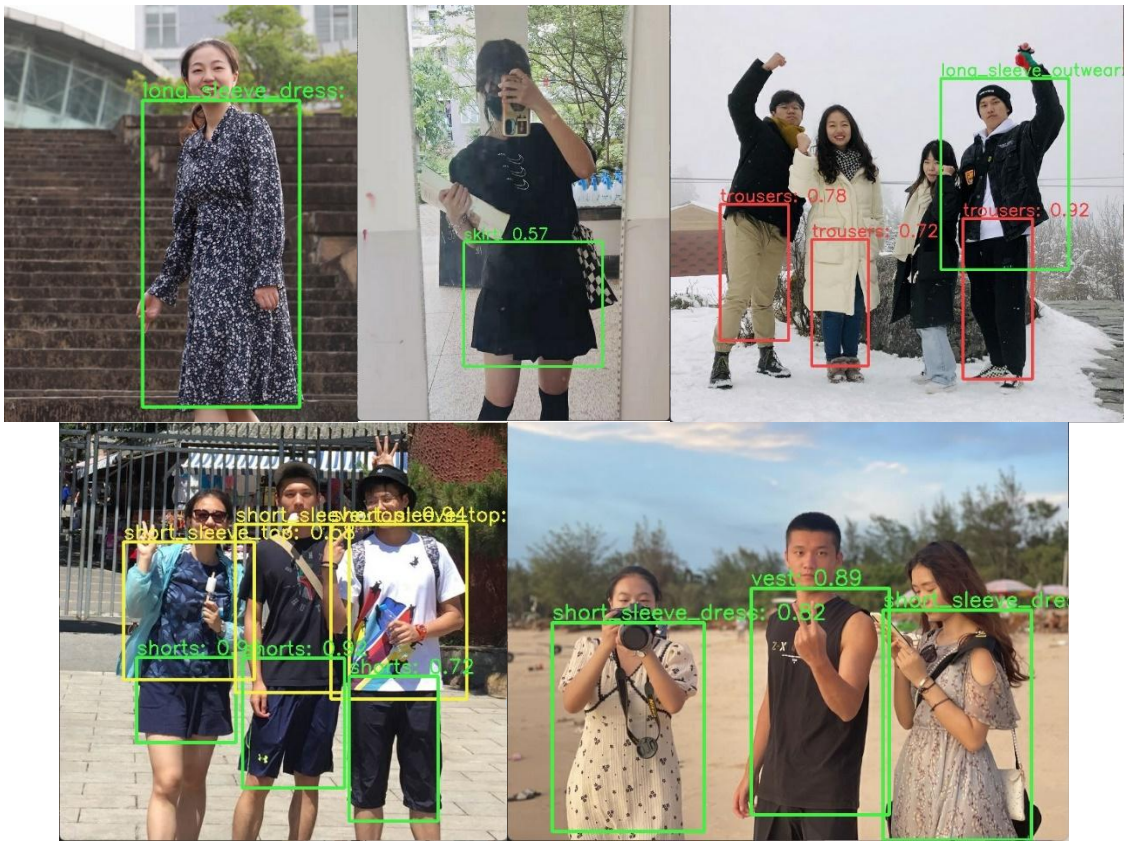

Figure 14 The result of clothing classification

In the illustration shown above, we can conclude that the system is clearly less capable of multi-target detection than single-target detection. When a picture involves more than one person, factors such as the background, the pose of the person, and the obscuration of the person are all important in influencing the final result of the experiment. We can also conclude from the AP values in the next chapter that the system is better at summer clothing (i.e., short-sleeved clothing) than long-sleeved, and we can see from Figure 2 that skirt is indeed a more difficult type of clothing to detect. In Figure 3 we can see four people wearing winter clothing with minimal interference from the background factors within the picture, but three people's tops are

still not recognised. This is due to the fact that the people are obscured and the long down jacket category is not added to the training.

Figure 15 Comparison between the samples with same figure

In the two images above, Figure 1 is not detected by the system, while Figure 2 is a cropped sample of Figure 1, which is immediately recognised by the system and has high confidence in both the trousers and the long-sleeved top. We can conclude from this that the system has strict requirements for the size of the people in the images, and it is difficult to recognise small clothing objects like the one in Figure 1.

4.4 VOC mAP

This chapter describes the equations used to judge the merits of the project's results Average Precision.

4.3.1 IOU

Intersection Over Union (IOU), also known as the intersection ratio, is used to measure the degree of overlap between two boxes. For a real box and a predicted box, their IOUs are calculated as follows.

$$IOU = \frac{S1 \cap S1}{S1 \cup S2} \#(12)$$

S1 and S2 represent the two rectangular bounding boxes, i.e., the real and predicted garment areas, $S1 \cap S1$ represents the area of the overlap between the two areas, and $S1 \cup S2$ represents the total area occupied by the two areas. The maximum value of IOU is 1 and the minimum value is 0. When IOU is 1, it means that the two regions overlap completely, and when IOU is 0, it means that the two areas do not overlap at all. Therefore, the larger the IOU, the higher the quality of the predicted garment area.

4.3.2 Precision-Recall Curve

precision means how many samples with a prediction of 1 are correct

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{all\ detections} \#(13)$$

recall means how many of the samples that would have been 1 were predicted correctly

$$Precision = \frac{TP}{TP + FN} = \frac{TP}{all\ ground\ truths} \#(14)$$

PR curve use: measure the performance of the model when the confidence threshold is changed

4.3.3 Average Precision

11-point interpolation: before VOC2010, the Precision maximum was selected when Recall >= 0, 0.1, 0.2, ..., 1., 1 is the maximum value of Precision at 11 points, and AP is the average of these 11 Precision, when only 11 points are used to approximate the area under the PR curve.

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} \rho\ interp(r)$$

$$\rho interp(r) = \max_{\tilde{r}:\tilde{r} \geq r} \rho(\tilde{r}) \#(15)$$

$\rho(\tilde{r})$ is the precision from recall = $(\tilde{r})$

In the Figure 15, we can see the mean Average - Precision for each garment type after testing

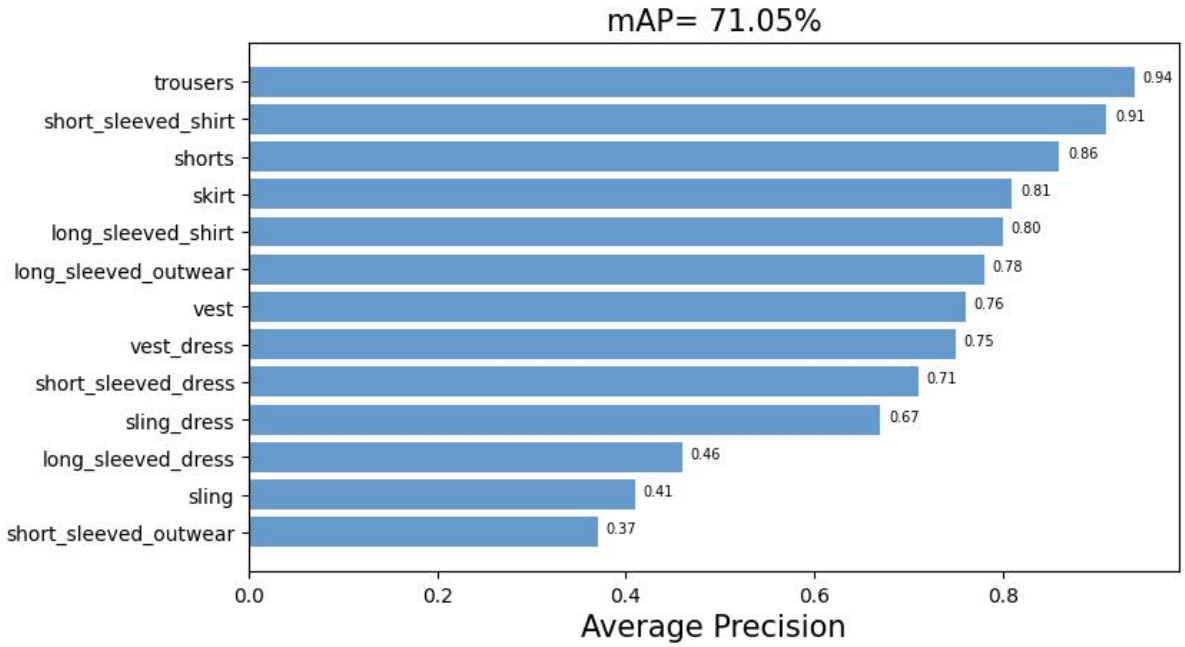

Figure 16 Average Precision in 13 categories

From the line graph presented above, we can conclude that the Average-Precision of trousers is the largest figure with 0.94 in that of thirteen categories. At the same time, the short-sleeved outwear is the most difficult clothing type to detection, the figure of that only gets 0.37. Overall, most clothing types occupy AP values above 0.5, with the exception of long-sleeved dress, sling, short sleeved outwear

4.5 Result of Output in Video

After debugging the model, the final process is completed by applying it to a video decomposition system. The code will first perform the decomposition of the video into frames, then perform garment recognition on each frame, and finally combine them into a complete video. Figure 17 shows the experimental results.

Figure 17 The result of clothing detection in video

In the picture above we can conclude that clothing recognition is well represented in dealing with complex environments. In the video frames we can observe that there are still some images that are too blurred to be recognised, but in the video, production results this is only a one frame error to be recognised by the human eye. We should try the results of the multiplayer video again

5 Discussion

This chapter mainly conducts experiments and performance analysis of each module of video-based garment detection scheme. Firstly, the experimental effect of clothing foreground retrieval is analyzed, and then the effect of image-based clothing detection algorithm is displayed and analyzed. Then, based on the above results, the effect of adding the module based on key frame screening is displayed and the corresponding retrieval effect is compared. Finally, the effect of feature fusion based on feature attention mechanism is analyzed and compared. To sum up, this paper discusses a video-based clothing retrieval framework and its optimization scheme, so that the framework can be applied to real-time detection scenes and offline detection fields Scene.

In the whole experiment, we can see that the results are still satisfactory. Apart from the indicators, data set DeepFashion2 has trained a model with good generalization ability. The image examples listed in this article are not comprehensive and do not even include all clothing categories, because considering that 13 categories require at least dozens of images to meet the requirements, there is no need to place so many images in this article. The title of our project is: Deep Clothes Wearing Classification in video, but what we have been talking about is image-based clothing recognition. We also mentioned this reason at the beginning of this paper. All clothing recognition is based on video or picture, and it is the ultimate basis or image target detection, which is also the core of our research. For the method of video, this paper uses the system to pre-process the video into a series of pictures of 30 frames in advance, and then the system detects and marks each picture. The tagged images are then stored in a target folder, and the final step of the system is to synthesize all the images in the target folder into a video. Experimental results show that the output video effect is clear, and the labelled image tracking effect is good.

6 Conclusion

To review this paper, we have done the following: learning the basics of deep learning algorithms, searching and reviewing related materials and previous approaches to garment detection, learning about the YOLOV3 model, building a model for training tests, and summarising the experimental results.

From the experimental data, although the overall mAP value is 71.05%, the AP values of the thirteen garment types are different and disparate. We can conclude that the overall detection effect of the system is good and can accurately detect most of the formulated garment types, while a few garment types (long sleeved dress, sling, short sleeved outwear) have a low detection effect. The reason for this is probably the uneven distribution of the results due to the lack of training samples as a result of the small number of garment types accommodated in the random sampling of the dataset.

The goal of the experiment is video-based clothing detection, but this paper takes a traditional image target detection approach, decomposing the video and then composing it. When looking at the data samples a large number of similar frames were found and a large number of frame samples were blurred. The blurred samples do not affect the imaging effect of the final output video even if they cannot be identified. However, such an operation wastes a lot of computational power and makes the computer recognise too many similar and useless frames. In future research, there should be a key frame extraction and detection step for video and the target can be tracked to avoid repetitive operations.

In this paper the YOLOV3 model is used, the reason for using it is that in reviewing the literature and related code content, it was found to be the most used and widely used. In future research, more models should be looked at to evaluate the strengths and weaknesses of each model in terms of efficiency, computational power and time spent on garment detection.

References

[1]. C. G. Harris, M. Stephens. A combined corner and edge detector[C]. Alvey vision conference, Manchester, 1988, 1-6.

[2]. D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110.

[3]. N. Dalal, B. Triggs. Histograms of oriented gradients for human detection[C]. IEEE computer society conference on computer vision and pattern recognition, San Diego, 2005, 886-893.

[4]. Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. arXiv preprint arXiv:1409.1556, 2014.

[5]. Christian Szegedy, Wei Liu, Yangqing Jia et al. “Going deeper with convolutions”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015:1–9.

[6]. Kaiming He, Xiangyu Zhang, Shaoqing Ren et al. “Deep Residual Learning for Image Recognition”. arXiv preprint arXiv:1512.03385, 2015.

[7]. Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky et al. “Dropout: a simple way to prevent neural networks from overfitting.” Journal of Machine Learning Research, 2014,15(1): 1929–1958.

[8]. Yannis Kalantidis, Lyndon Kennedy & Li-Jia Li. (2013) "Getting the Look: Clothing Recognition and Segmentation for Automatic Product Suggestions in Everyday Photos".

[9]. P. Viola, M. J. Jones. Robust real-time face detection[J]. International journal of computer vision, 2004, 57(2): 137-154.

[10]. P. F. Felzenszwalb, R. B. Girshick, D. McAllester, et al. Object detection with discriminatively trained part-based models[J]. IEEE transactions on pattern analysis and machine intelligence, 2009, 32(9): 1627-1645.

[11]. W. Liu, D. Anguelov, D. Erhan, et al. Ssd: Single shot multibox detector[C]. European conference on computer vision, Amsterdam, 2016, 21-37.

[12]. A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao. YoLov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv: 2004.10934, 2020.

[13]. R. Girshick, J. Donahue, T. Darrell, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]. IEEE conference on computer vision and pattern recognition, Columbus, 2014, 580-587.

[14]. R. Girshick. Fast r-cnn[C]. IEEE international conference on computer vision, Boston, 2015, 1440-1448.

[15]. S. Ren, K. He, R. Girshick, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]. International Conference on Neural Information Processing Systems, Montreal, 2015, 91-99.

[16]. W. Liu, Y. Wen, Z. Yu, et al. Large-margin softmax loss for convolutional neural networks[C]. International Conference on Machine Learning, New York, 2016, 507-516.

[17]. T. Kong, A. Yao, Y. Chen, et al. Hypernet: Towards accurate region proposal generation and joint object detection[C]. IEEE conference on computer vision and pattern recognition, Las Vegas, 2016, 845-853.

[18]. J. Dai, H. Qi, Y. Xiong, et al. Deformable convolutional networks[C]. IEEE international conference on computer vision, Venice,2017, 764-773.

[19]. Yuying Ge, Ruimao Zhang, Lingyun Wu, Xiaogang Wang, Xiaoou Tang & Ping Luo. (2019)"DeepFashion2: A Versatile Benchmark for Detection, Pose Estimation, Segmentation and ReIdentification of Clothing Images".

[20]. Joseph Redmon & Ali Farhadi. (2018) "YOLOv3: An Incremental Improvement".