IMAGE TEXTURE CLASSIFICATION ALGORITHM

LEI XIA

MSc of Advanced Computer Science

An MSc Dissertation

Under the Supervision of Dr. Xianfang Sun

School of Computer Science and Information,

Cardiff University November 2021

Abstract

As a basic attribute of object surface, texture is widely used in image classification, cause it can represent images preferably. Texture classification is a technology that uses images processing technology to extract important surface gray information from images and classify texture features. Image texture classification is an important basically issue in the field of version of computer and recognition of pattern. The aim of this project is to study several of normal algorithms in image texture classification which could solve the issues of image texture classification. In addition, the commonly used image classification methods are reviewed, including K- nearest neighbor method, K- means clustering, decision tree, boosting algorithm and support vector machine. For normal image texture classification algorithms are selected deeply research and implementation, namely Gray Level Co-occurrence Matrix algorithms, Local Binary Patterns and ResNet. It could be introduced in detail by these algorithms respectively, and their specific implementation process is given through an example. In the end, the work which is done in this dissertation is summarized, and the further research and application of texture classification is prospected.

Acknowledgements

In the process of creating this dissertation, i would like to gratefully thank my supervisor, Dr. Xianfang Sun, who consistently supported me throughout the project. His guidance and kindness had largely encouraged me to finish the project better.

Table of contents

1. Introduction..... 5

2.Aim and Objectives..... 7

3.Background..... 7

3.1 Deep Learning..... 7

3.2 Gray Level Co-occurrence Matrix..... 7

3.3 Local Binary Patterns..... 8

3.4 Convolutional Neural Network..... 9

4.Principles..... 9

4.1 Gray Level Co-occurrence Matrix 9

4.1.1 Definition..... 9

4.1.2 Definition of Spatial Relationship..... 9

4.1.3 The Process of Generation in GLCM..... 10

4.2 Local Binary Patterns 9

4.2.1 LBP Algorithm Idea..... 10

4.2.2 Principle of Standard LBP Algorithm..... 10

4.3 The Principle of Convolutional Neural Networks 9

4.3.1 input layer..... 11

4.3.2 CONV layer..... 12

4.3.3 ReLU layer..... 12

4.3.4 pooling layer..... 13

4.3.5 FC layer..... 13

5.Related Work..... 14

5.1 GLCM..... 14

5.1.1 More detailed statistics..... 14

5.1.2 The algorithms of feature extraction in GLCM..... 15

5.2 LBP..... 16

5.2.1 Classical LBP apply in this project..... 16

5.2.2 Several special LBP algorithms..... 17

6.The implementation of our method..... 18

6.1 Feature extraction based on ResNet..... 18

6.2 Detailed introduction of ResNet..... 22

6.2.1 The architecture of model in ResNet..... 22

6.2.2 The principle of ResNet..... 23

6.2.3 ResNet 50..... 24

7. Experiment..... 24

8. Conclusion..... 32

8.1 Benefits and Deficiencies..... 33

8.2 Future Work..... 33

9.Reflections of learning..... 34

Reference..... 36

1.Introduction

In modern field of computer science, version of computer and recognition of pattern are faced with a significant problem is image texture classification (Chincheng and Yihua 2019) In tradition, texture is a macro representation of a local repetition pattern of eigenvalue intensity, and local pattern repetition and stability which are dominating characteristics in image. Additionally, image texture classification is to assign a predefined texture category to the image region to be classified based on its content. Therefore, it would be used image texture classification algorithms, such as Gray Level Co-occurrence Matrix algorithms, Local Binary Patterns and ResNet to achieve. However, when these algorithms to implement, there are many difficulties need to solve. Firstly, in example level, for a single texture image, the apparent characteristics of texture change greatly due to the illumination conditions, shooting angle, shooting distance and non rigid shape change of shooting surface during image acquisition, in which brings difficulty to the image texture classification algorithms. Moreover, in category level, there are large intra class differences firstly, that is , the apparent features of textures belonging to the same class are quite different. On the one hand, the reasons include the changes of various example levels which is mentioned above. On the other hand, including the differences of different examples in the class more importantly. The second difficulty is the inter class fuzziness, which is the texture instances of different categories have certain similarity. Finally, the noise interference is the difficulty which need to be faced. In actual scene, fuzzy and random noise and the influence of rain, fog and snow is likely to appear (Jiawang et al. 2017) . These difficulties improves the accuracy of identification.

In addition, there are signal-based processing methods to extract and classify images. This method can transform the image into domain of frequency through Fourier transform. The distribution of power spectrum is obtained by counting the local spectrum energy, and the size and characteristics of the texture can be further known, and then extracting the texture features.

Thirdly, there are model-based method and structure- based method to extract texture features. Firstly, Gibbs- MRF random field is based on the equivalence of distribution between MRF and Gibbs . The texture is modeled by determining the total energy of the mesh (Pietikainen 2000). Because natural textures have similarity at different scales, these models are applied to texture analysis. The most commonly used model is fractal Brown motion, FBM. More importantly, texture of sentence analysis and mathematical morphological analysis can be used for texture extraction. Such methods commonly belong structure- based methods. However, this kind of method is relatively simple, but the efficiency is poor. Therefore, it is rarely concerned by

scholars.

Besides, this dissertation introduces the process of achieving these algorithms. It further explains the conclusion and reflections in these algorithms. At the rest of this dissertation, it would give summary and compare for these algorithms and comment the advantages and disadvantages of them. In final, the dissertation comes to an end with conclusions and reflections on learning included.

2.Aim and Objectives

The goal of this project is that implement and compare some image texture classification algorithms, and try to improve the existing algorithms. Due to the these algorithms are difficult to solve in this process, student needs to study some knowledge which relevant with advanced mathematics and deep learning, hence, it could be picked up the Gray Level Co-occurrence Matrix algorithms, Local Binary Patterns and ResNet to research.

Aims of project, as following:

* use programming language Python to achieve these three algorithms.
*use these algorithms to extract the features of images, and using SVM and Xgboost to classify these images.
* Analyze and compare these three algorithms which is mentioned above.
* Find the advantages and disadvantages of those algorithms and try to improve them.
* The best goal that student could improve these algorithms

3.Background

Except color and shape, texture, as a natural attribute of object surface, is an important basis for human visual system to distinguish natural objects. Texture feature is used to describe the roughness, regularity and directivity of object surface. It is the information that can not be ignored in building a vision system of machine. Generally, texture is divided into three types: natural texture, artificial texture and mixed texture.

Although using Python to achieve the image texture classification algorithms to extract and distinguish images is complicated, there are model of algorithms could be applied which would be found online in recent. Therefore, it could decrease a lot of time to program these complex algorithms.

3.1 Deep Learning

Deep learning is a new research direction in the field of machine learning. It is

introduced into machine learning to close to the original goal - Artificial Intelligence. Deep learning is the internal rule and representative level of learning sample data. The information which is obtained in the learning process is helpful to the interpretation of data, such as text, image and sound. Its ultimate goal is to make the machine have the ability of analysis and learning like human beings. Deep learning is a complicated machine learning algorithm, which has achieved more results in speech and image recognition than previous related technologies

3.2 Gray Level Co-occurrence Matrix

This algorithms is a common method to describe texture by studying the spatial correlation characteristics of gray level. Because the texture is formed by the repeated occurrence of gray distribution in the spatial position, there will be a certain gray relationship between two pixels separated by a determinate distance in the image space, that is the spatial correlation of gray in the image. This method would be presented in 1973 by Dr. Haralick to describe the texture of image and classify it(Yongsheng et al. 2017).

According to the content which is mentioned above, gray level co- occurrence matrix is used to extract textural features, which is similar to ResNet module in deep learning. After extracting the features, still it can not be directly used for classification. Therefore, some statistics in gray feature matrix are used for classification.

In the past 50 years, many scholars have studied texture analysis and tried to classify extraction methods of texture features (Pietikainen 2000). There are different classification methods in different historical stages, but at present, the most popular are four classification methods: structural method, statistical method, spectrum method and model method. In this project, the gray level co-occurrence matrix and LBP algorithm in statistical method are mainly used.

In this project, the files to extract the image texture which is used by gray level co-occurrence, as well as classifying the images by .ipynb files. Specially, .ipynb files is the results of SVM and Xgboost. Moreover, the .csv file is the preprocessing data file.

3.3 Local Binary Patterns

LBP (local binary patterns) is an algorithm used to describe the local texture features of images. It has remarkable advantages such as rotation invariance and gray invariance. It was firstly proposed by Dr. Harwood in 1994 for texture feature extraction (Hong et al 2012). Moreover, the extracted feature in images is only the local texture features of the images.

LBP have emerged as one of the most prominent and widely studied local texture descriptors. It is similar with the gray level co-occurrence matrix, the characteristic of Local Binary Patterns is that extracting the module of texture in images. After extracting, it would be need Xgboost and SVM to classify the images. The reason that LBP would not classify the image texture is that it do not have corresponding statistics. In this project, it would be classified by histogram directly.

Furthermore, LBP is usually used in face recognition and target detection. In OpenCV, there is an interface for face recognition using LBP features and a method for training target detection classifier with features of LBP(Thomas and Hugues 2020). OpenCV achieves the calculation of LBP features, but it does not provide a separate interface for calculating characteristic of LBP. In other words, LBP algorithm is used in OpenCV, but it do not provide interface of function.

3.4 Convolutional Neural Network

ResNet is the abbreviation of residual network, which is one of the CNN. This series of networks are widely used in the fields of target classification and as a part of the classical neural network, which is the tasks of computer vision. The typical networks are ResNet 50, Resnet 101. in this project, ResNet 50 would be chose to extract and classify the textures of images.

Applying ResNet 50 is a bit different from other two methods which is mentioned above. ResNet 50 could classify the images, after it complete to extract. However, this method need to process deep learning by a mass of data. Therefore, the process of training is not more convenient than other two methods.

4.Principles

In this project, it has introduced three methods of texture analysis. More importantly, to understand the mathematical principles of these method, it could more effectively implement algorithms in the next step.

4.1 Gray Level Co-occurrence Matrix

4.1.1 Definition

Considering the second- order statistics, the pixel pairs with spatial relationship are studied:

$$P(i, j| d, \theta)$$

Represents that when the distance d and direction in a given space θ, the gray level starts i, and producing the probability of gray level j. It constitutes the elements of gray level co- occurrence matrix.

4.1.2 Definition of Spatial Relationship

Let the spatial position relationship of a pair of pixels be expressed as α= (a,b),(a,b).

* The values of a and b should be selected according to the characteristics of periodic distribution of texture. For Finer texture, it could be selected as (1,0),(1,1), etc.
* If the values of a and b are small, they correspond to the image of texture with slow change, and the values on the diagonal of the gray level co-occurrence matrix is large.
* The faster textures change, the smaller the value on the diagonal and the larger values on both sides of the diagonal (Pietikainen 2000).

4.1.3 The Process of Generation in GLCM

* In the image, any point(x,y) and the point that deviating from it (x + a, y+b) form a point pair. Let the gray value of the point pair be (f1, f2). Assuming that the maximum gray level of the image is L, there are L*L species in combinations of f1 and f2.
* for the whole image, count the number of occurrences of each(f1,f2) value, and then arranging it into a square matrix.
* Then, the total number of occurrences of (f1, f2) is normalized to obtain the probability P(f1, f2). Therefore, it could produce the GLCM.

4.2 Local Binary Patterns

4.2.1 LBP Algorithm Idea

The original LBP operator is defined in the domain of pixels 3* 3. Taking the pixel of domain center as the threshold, the gray value of 8 adjacent pixels is compared with the pixel value of the domain center.
If the surrounding pixels are greater than the value of central pixel, the position of the pixel is marked as 1, otherwise it is 0. In this way, 8 points in the 3* 3 domain can be compared to generate 8- bit binary numbers. These 8- bit binary numbers are arranged in turn to form a binary number. The decimal representation of this binary number is the LBP value of the central pixel, there are 28 possible LBP values. Therefore, there are 256 possible LBP values. The LBP value of the central pixel reflects the texture information of the area around the pixel.

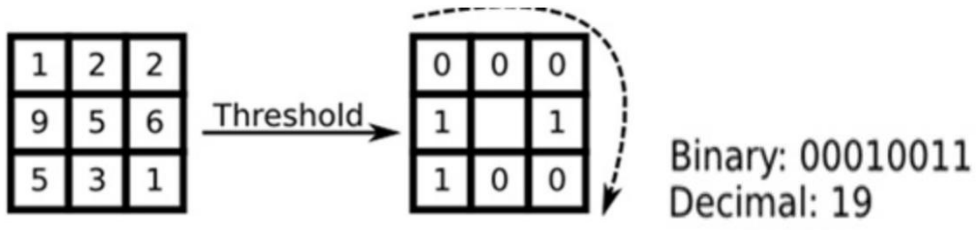

Figure1: the step of LBP algorithm

4.2.2 Principle of Standard LBP Algorithm

The LBP feature uses the joint distribution T of the local domain of the image to describe the texture feature of the image. If the number of pixels in the local domain is assumed to be P (P >1), the joint distribution T of the texture feature can be expressed as:

$$T = t(g_c, g_0, \ldots, g_{p-1}),$$

Where, p= 0, ..., P- 1

g_c represents the gray value of the central pixel of the corresponding local domain,

g_p represents the gray value of the pixel on the circle with the center of the central pixel and radius of R.

Assuming that the central pixel and the local domain pixel are independent of each other, the above definition can be written as follows:

$$T = t(g_c, g_0 - g_c, \ldots, g_{p-1} - g_c) \approx t(g_c)t(g_0 - g_c, \ldots, g_{p-1} - g_c)$$

And $t(g_c)$ determines the overall brightness of the local area, which can be ignored for texture features.
Therefore, it could get the formula:

$$T \approx t(g_0 - g_c, \ldots, g_{p-1} - g_c).$$

All in all, the definition of standard LBP is $LBP_{P,R} = \sum_{p=0}^{p-1} s(g_p - g_c)2^p$.

4.3 The Principle of Convolutional Neural Networks

CNN normally consists of 5 layers, including input layer, CONV layer, ReLU layer, Pooling layer and FC layer.

4.3.1 input layer

This layer is mainly to preprocess the original data, including: all dimensions of the input data are centered to 0; Normalization and using PCA to reduce dimension. The example could be shown figure below:

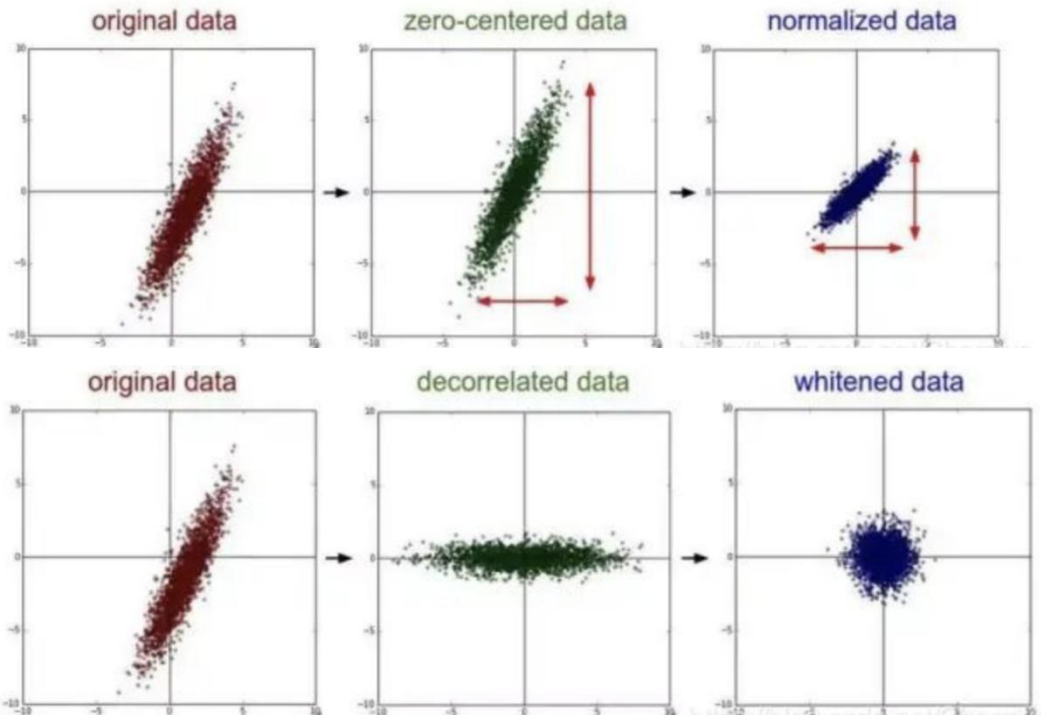

Figure2: the process by using PAC to reduce

When the input layer process the image texture, the input layer normally represents three- dimensional pixel matrix of image. These three dimensions are the width, length and depth of the image. Depth is also named the number of channels.

4.3.2 CONV layer

Convolution layer is the most important part of CNN structure. It obtains the local area information of the image through convolution operation. By using a filter, the convolution operation transforms the matrix of the sub node on the current layer of neural network to a node matrix on the next layer neural network, and the resulting matrix is called feature map（Hongchang 2021）.

4.3.3 ReLU layer

It could use the results which come from CONV layer to non- liner mapping. This map could show the process.

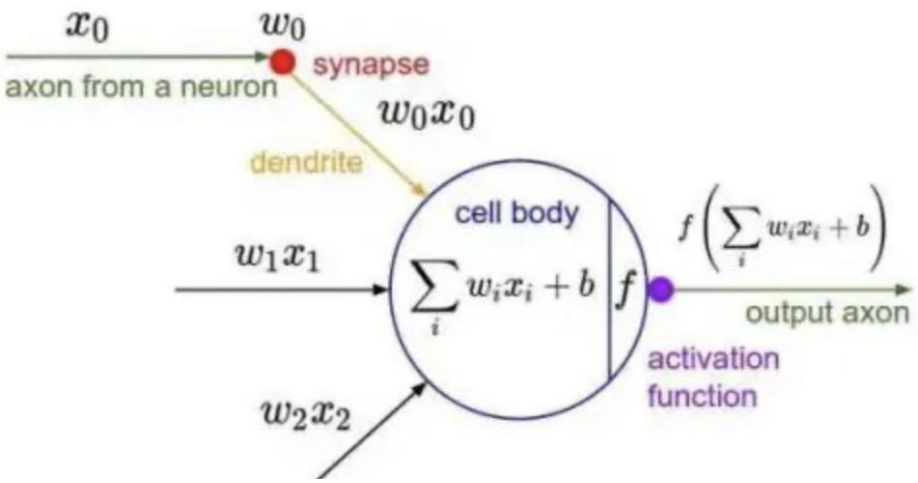

Figure3: the process of non- liner mapping

4.3.4 pooling layer

It could compress the amount of data and parameters and reduce over fitting. In other word, if the input is an image, the main function of this layer is to compress the image （ Yang et al. 2016）. Therefore, the pooling layer could compress the information of image, remove redundant information in images, and prevent the over fitting.

The methods in pooling layer are max pooling and average pooling. Regularly, people use max pooling is more frequently. And the principle is easy to understand, that is, extract the max figure in a matrix.

4.3.5 FC layer

All neurons have weight connections between the two layers (Pietikainen 2000). FC layer is usually at the tail of the CNN. That is, the connection mode of neurons is the same as that of traditional neural network. The example diagram could be shown below:

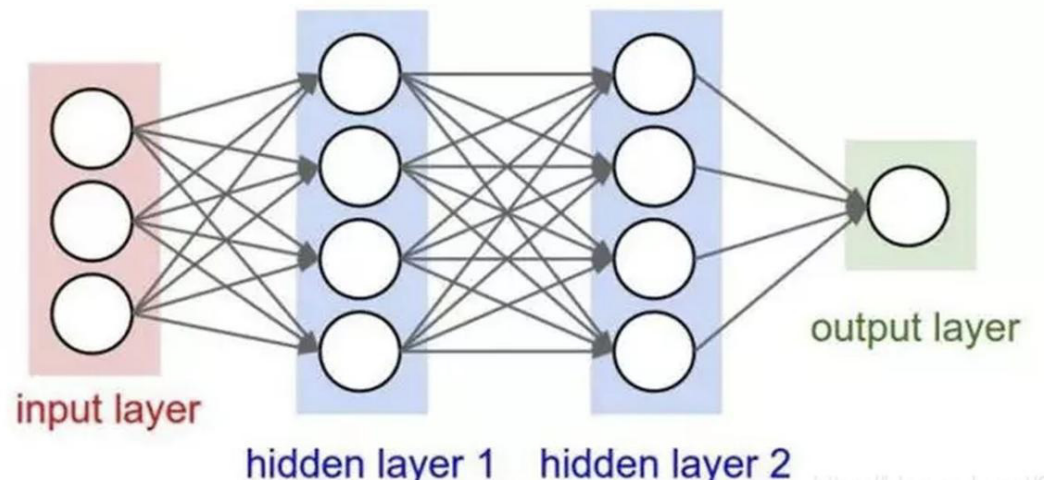

Figure4: the traditional neural network

5.Related Work

In this prat, it could introduce that the method which is applied in extracting images. Here are two method which could be mentioned above: GLCM and LBP. These two methods would be applied by others before, hence, it could have a comparison between the method which is used later and related work of these two methods.

5.1 GLCM

In the previous chapter, this dissertation mentioned the definition and calculation process of GLCM. However, in this chapter, it will introduce the previously related work on GLCM and how GLCM extracts texture of image.

5.1.1 More detailed statistics

Because GLCM can not be directly used to describe the texture features of images, predecessors have defined vectors of feature texture, such as second- order moment, entropy, contrast and correlation to reflect the features of texture.

***ASM**

This feature reflects the image gray distribution and texture thickness. The value of ASM depends on the distribution of elements in the co-occurrence matrix. If all elements are close to the same, the value of ASM is smaller. Conversely, the ASM is larger. In addition, a large ASM value means that the texture patterns are evenly distributed and change regularly.

$$ASM = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} P(i,j)^2$$

Figure5: the formula of ASM

***Contrast**

$$CON=\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}(i-j)^2 P(i,j)$$

Contrast can reflect the clarity of the image and the depth of the texture, that is, the bigger the value of the element far from the diagonal in GLCM, the bigger the contrast.

***Correlation**

COR= $\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\frac{i,jP(i,j)-\mu_1\mu_2}{\sigma_1\sigma_2}$

And the $\mu_1, \mu_2, \sigma_1, \sigma_2$ represent respectively:

$$\mu_1 = \sum_{i=0}^{L-1} i \sum_{j=0}^{L-1} P(i,j)$$

$$\mu_2 = \sum_{i=0}^{L-1} j \sum_{j=0}^{L-1} P(i,j)$$

$$\sigma_1=\sum_{i=0}^{L-1}(i-\mu_1)^2\sum_{j=0}^{L-1}P(i,j)$$

$$\sigma_2=\sum_{i=0}^{L-1}(j-\mu_2)^2\sum_{j=0}^{L-1}P(i,j)$$

Correlation reflects the similarity between domain pixels in horizontal and vertical directions. If there are a large number of horizontal textures in the image, the correlation of GLCM in the horizontal direction is greater than that in other directions.

***ENT(entropy)**

Entropy is a significant concept in theory of information. In the image, it represents the amount of information contained in the image. As a part of image information, texture is a measure of randomness. When the distribution of elements in the co-occurrence matrix is more random and all elements with little difference, the entropy is larger. This eigenvalue is used to measure the complexity of image texture.

$$\text{ENT}= -\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}P(i,j)\lg P(i,j)$$

5.1.2 The algorithms of feature extraction in GLCM

The transformation of gray level in image. If it input an RGB image, it need to convert to a image with gray level.

Gray level quantization. In order to saving the time of computing, the gray level is reduced to an appropriate value.

Calculate the co- occurrence matrix in four directions. Take the distance as d = 1, and the directions as 0, $\frac{\pi}{4}$, $\frac{\pi}{2}$,$\frac{3\pi}{4}$.

(4) The normalized co-occurrence matrix. It usually select Gaussian function as the normalized function.

(5) Calculate the values of features of statistics respectively by each co- occurrence matrix.

(6) calculate the mean value and standard deviation in each values of features. Then the 8- dimensional vector is the final vectors of feature texture.

In this project, the part of GLCM to extract features of texture would be show below:

```
class_label = [i[0] for i in sio.loadmat('./dtd/imdb/imdb.mat')['meta']['classes'][0][0][0]]
if os.path.exists('gray.csv'):
    df = pd.read_csv('gray.csv')
else :
    data_images = sio.loadmat('./dtd/imdb/imdb.mat')['images']
    dir_path = './dtd/images'
    bins = np.linspace(0, 255, 64)
    db = []
    for i in range(len(data_images['name'][0][0][0])):
        image_path = os.path.join(dir_path, data_images['name'][0][0][0][i][0])
        img = np.array(Image.open(image_path).convert('L'))
        # compress_gray = np.digitize(img, bins)
        # gray = np.uint8(compress_gray)
        glcm = greycomatrix(img, [2, 4, 8, 16],[0, np.pi / 4, np.pi / 2, np.pi * 3 / 4],256, symmetric = True, normed = T

        # Angular Second Moment
        gray_ASM = greycoprops(glcm,'ASM').flatten()
        # Contrast

        gray_contrast = greycoprops(glcm,'contrast').flatten()
        # Inverse Differential Moment
        gray_IDM = greycoprops(glcm,'homogeneity').flatten()

        temp = {'id':i,'class':data_images['class'][0][0][0][i]}
        for j in range(gray_ASM.shape[0]):
            temp['ASM_'+str(j)] = gray_ASM[j]
        for j in range(gray_contrast.shape[0]):
            temp['contrast_'+str(j)] = gray_contrast[j]
        for j in range(gray_IDM.shape[0]):
            temp['IDM_'+str(j)] = gray_IDM[j]
        db.append(temp)
    df = pd.DataFrame(db)
    df.to_csv('gray.csv')
```

Figure6: the code of GLCM to extract images texture

5.2 LBP

5.2.1 Classical LBP apply in this project

The classical algorithm and idea of LBP have been mentioned in the previous chapter 4.2. in this project, it use a classical LBP algorithm to realize the texture image after LBP processing. The result would be shown below:

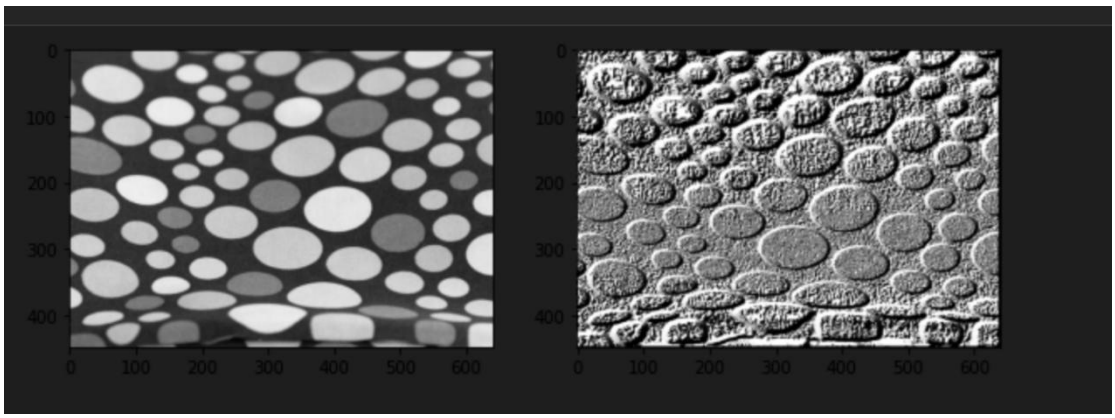

Figure7: image is processed by LBP

Moreover, In the specific application of LBP, such as texture classification, face recognition. Generally, LBP map is not used as vector of feature for classification and recognition, but the statistical histogram of LBP feature map is used as feature vector for classification and recognition. This is mainly because the features represented by LBP map are closely related to location information. Similarly, in this project, the program ran out of the histogram to show this:

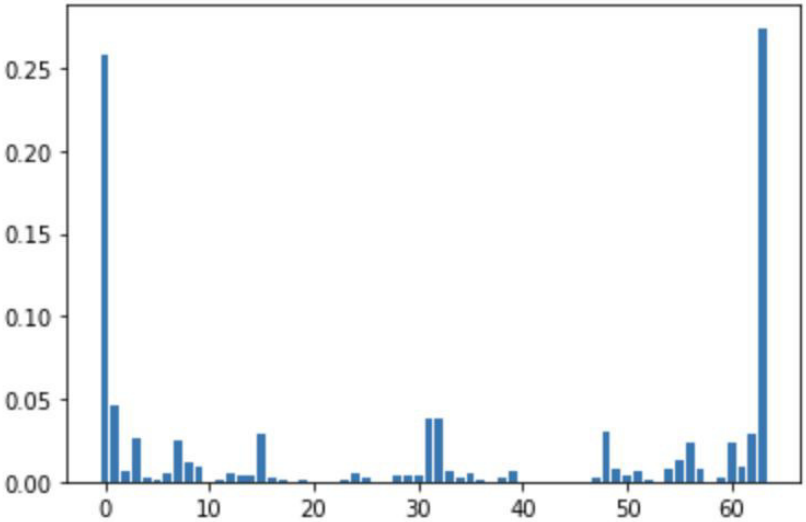

Figure8: the histogram of results of LBP

5.2.2 Several special LBP algorithms

Because the classical LBP algorithm is developed extremely early, there are still many deficiencies. Therefore, many programmers developed many deformed LBP algorithms to amend for these problems later.

***circular LBP**

The classical LBP has a defect that the radius of its coverage area is fixed. This means that it could not extract texture features of different sizes. In order to solving this problem, Ojala and other scientists have made some improvements to the classic LBP.

The obtained circular LBP Operator does not limit the number of pixels in the circular region with radius R, and then becoming an LBP Operator with P pixels in the circular region with radius R.Its definition as below:

$$LBP_{P,R} = \sum_{p=0}^{p-1} s(g_p - g_c)2^p$$

$$S(x)= \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

***Uniform Patterns**

Its mean is that a binary sequence changes from 1 to 0 or from 0 to 1 no more than twice. For a more detailed description, the mathematical expression of the uniform patterns LBP is given below:

$$U(LBP_{P,R})= |s(g_{p-1} - g_c)- s(g_0 - g_c)|+ \sum_{p=1}^{p-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|.$$

6.The implementation of our method

6.1 Feature extraction based on ResNet

With the development of convolution neural network, it is found that the deeper the convolution layer, the better the model effect. Therefore, after the success of VGG model, scholars began to try a deeper convolution network, but experiments show that the increase of network depth will cause gradient explosion and disappearance, resulting in the failure of system convergence.

The network structure of CNN has been described in detail above. In this section, it will firstly introduce how to use ResNet to extraction of features in this project.

First of all, it should initialize dataset, and producing dataloader. Then, it could program to check the dataset whether process or not. Nowadays, there are many texture classification datasets to choose. Including MINC, KTHTIPS, DTD, VisTex. In this project, the dataset should be suitable for the experiment. Therefore, VisTex must firstly exclude ,which is appropriate for outdoor environment, and the number of

images in MINC is too large and the number of images in KTHTIPS is too small. Finally, DTD data set is selected for this project, the number of images is 5640 and the image classification is 47.

```python
dtd = DTD()
dtd_loader = DataLoader(dataset = dtd,batch_size = 128, shuffle=False)
train_dataset, test_dataset = random_split(dataset=dtd,lengths=[4512,1128],generator=torch.Generator().manual_seed(512))
train_loader = DataLoader(dataset = train_dataset,batch_size = 128,shuffle = False)
test_loader = DataLoader(dataset = test_dataset,batch_size = 32,shuffle = False)
```

Figure9: the process of initialize dataset

```python
for step, (b_x, b_y) in enumerate(train_loader):
    if step > 0:
        break
print(b_x.shape,b_y.shape)

plt.figure()
for i in range(4):
    plt.subplot(4,2,2*i+1)
    plt.imshow(b_x.data.numpy()[2*i,0,:,:], cmap="gray")
    plt.axis("off")
    plt.subplot(4,2,2*i+2)
    plt.imshow(b_x.data.numpy()[2*i+1,0,:,:], cmap="gray")
    plt.axis("off")
plt.show()
print(b_y)
```

Figure10: check the situation of dataset processing

Moreover, it should introduce an important part of the code which could achieve the ResNet in deep learning. In this project, it use ResNet class to define the type of network and self feature could extract the features which is used the official model of Pytorch. Finally, self classification is made of full connection networks, it could apply for classification of feature. And the detailed code would be shown below:

```python
class ResNet(nn.Module):
    def __init__(self):
        super(ResNet,self).__init__()
        self.features = torch.hub.load('pytorch/vision:v0.10.0','resnet50',pretrained=True)
        self.classification = nn.Sequential(
            nn.Linear(1000,512),
            nn.ReLU(),
            nn.Linear(512,256),
            nn.ReLU(),
            nn.Linear(256,128),
            nn.ReLU(),
            nn.Dropout(p=0.5),
            nn.Linear(128,47),
            nn.LogSoftmax(dim=1)
        )
        # for para in self.features.parameters():
        #     para.requires_grad = False
    def forward(self,x):
        x = self.features(x)
        x = self.classification(x)
        return x
```

Figure11: the code of using ResNet class

Further, it will introduce how to use the function to train the network. In this part, reading the files of model is the first step need to add. Then, in this part, adding the code of optimizer and the function of loss in deep learning to improve the accuracy of the results of training. And in function of loss, it could use the "function of For" to reflect the process of lterative.

```python
def train_model(net,start_epochs,num_epochs,lr,train_loss_all,train_acc_all,test_loss_all,test_acc_all,frozen_para = True,
    if model_path != None:
        net.load_state_dict(torch.load(model_path),strict=True)

    if frozen_para :
        for para in net.features.parameters():
            para.requires_grad = False
        optimizer = torch.optim.Adam(filter(lambda p:p.requires_grad,net.parameters()), lr=lr)
    else :
        # for para in net.features.parameters():
        #     para.requires_grad = True
        optimizer = torch.optim.Adam(net.parameters(), lr=lr)

    max_test_acc = 0
```

Figure12: the training network of function

```python
criterion = nn.CrossEntropyLoss().cuda()
for epoch in range(start_epochs, num_epochs+start_epochs):
    print('Epoch {}/{}'.format(epoch, num_epochs+start_epochs-1))

    net.eval()
    corrects = 0
    test_num = 0
    test_loss_epoch = 0
    for step,(b_x,b_y) in enumerate(test_loader):
        b_x = b_x.cuda()
        b_y = b_y.cuda()
        output = net(b_x)
        pre_lab = torch.argmax(output,1)
        loss = criterion(output, b_y.long())
        test_loss_epoch += loss.item() * b_x.size(0)
        corrects += torch.sum(pre_lab == b_y.data)
        test_num += b_x.size(0)
    test_loss_all.append(test_loss_epoch / test_num)
    test_acc_all.append(corrects.double().item() / test_num)
    print('{} Test Loss: {:.4f}  Test Acc: {:.4f}'.format(epoch,test_loss_all[-1],test_acc_all[-1]))
```

Figure13: the code of loss function

These all the part of this project which use the ResNet 50 to extract the image texture. And how to improve the accuracy in deep learning. And in the tail of this code, it add the selection of verification and selection of 3D TSNE. These two selection, the first selection is that verify the result of training and the second selection is that demonstrate the ability of generalization and the ability of feature extraction in ResNet model. The detailed codes could be found below:

```
def val_dtd(net,dtd_loader,model_path):
    net.load_state_dict(torch.load(model_path),strict=True)
    net.eval()

    target_lab_all = torch.LongTensor()
    pre_lab_all = torch.LongTensor()
    for step,(b_x,b_y) in enumerate(dtd_loader):
        target_lab_all = torch.cat((target_lab_all,b_y))
        b_x = b_x.cuda()
        b_y = b_y.cuda()
        output = net(b_x)
        pre_lab = torch.argmax(output,1)
        pre_lab_all = torch.cat((pre_lab_all,pre_lab.cpu()))

    plt.figure(figsize=(30,30))
    acc = accuracy_score(target_lab_all,pre_lab_all)
    print("Acc: {:.4f}".format(acc))
    class_label = dtd.class_name
    conf_mat = confusion_matrix(target_lab_all,pre_lab_all)
    df_cm = pd.DataFrame(conf_mat, index=class_label, columns=class_label)
    heatmap = sns.heatmap(df_cm,annot=True,fmt="d",cmap="YlGnBu")

    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.show()
    return target_lab_all,pre_lab_all
```

Figure14: the process of verifying results

```python
def draw_tsne(net,model_path):
    net.load_state_dict(torch.load(model_path),strict=True)
    net.eval()
    X_list = []
    Y_list = []
    for step, (b_x, b_y) in enumerate(train_loader):
        if step == 5:
            break
        output = net(b_x.cuda()).cpu().detach().numpy()
        b_y = b_y.data.numpy()
        X_list.append(output)
        Y_list.append(b_y)

    # print(X[0].shape,Y[0].shape)
    X = np.concatenate(X_list,axis=0)
    Y = np.concatenate(Y_list,axis=0)
    print(X.shape,Y.shape)

    # tsne = manifold.TSNE(n_components=2, init='pca', random_state=512)
    tsne = manifold.TSNE(n_components=3, init='pca', random_state=512)
    X_tsne = tsne.fit_transform(X)
    print("Org data dimension is {}. Embedded data dimension is {}".format(X.shape[-1], X_tsne.shape[-1]))

    x_min, x_max = X_tsne.min(0), X_tsne.max(0)
    X_norm = (X_tsne - x_min) / (x_max - x_min)
    fig = plt.figure(figsize=(24, 16))
    ax1 = Axes3D(fig)
    print(X_norm.shape)
    for i in range(X_norm.shape[0]):
        if Y[i]<10:
            # plt.text(X_norm[i, 0], X_norm[i, 1],str(Y[i]),fontsize=13,bbox=dict(boxstyle="round",facecolor=plt.cm.Set1(Y
            ax1.text(X_norm[i, 0], X_norm[i, 1],X_norm[i,2], str(Y[i]),fontsize=8,bbox=dict(boxstyle="round",facecolor=plt
```

Figure15: the code of 3D TSNE map

All in all, there are all work in our method: Resnet 50 to extract and classify images. And the result of analysis and the comparison above LBP and GLCM would be purposed in the next chapter.

6.2 Detailed introduction of ResNet

With the development of CNN, there are more and more convolution layers, in order to obtaining deep-seated features. At first, Le net had only 5 layers, then AlexNet had 8 layers. Later, VggNet included 19 layers and GoogleNet had 22 layers.

Although this dissertation has introduced the principle of CNN(Convontional Neural Network), and the principle of ResNet and the process of achieving it would not narrate now. Therefore, the purpose of this selection is that comprehend the ResNet thoroughly.

6.2.1 The architecture of model in ResNet

Before the emergence of ResNet model, it was extremely difficult to train a deep neural network. To solve this problem, a deep residual network ResNet is proposed, which allows the network to deepen as much as possible, but it directly sends a part of the data in the input layer to the output layer without going through the convolutional network, and it could retain a part of information. This structure effectively prevents the gradient dispersion problem in reverse transmission. Thus, the depth of network could reach hundreds of layers.

In ResNet network, adding convolutional layer could improve the efficient of training. There is the residual block structure plays a key role. And its structure could be shown below:

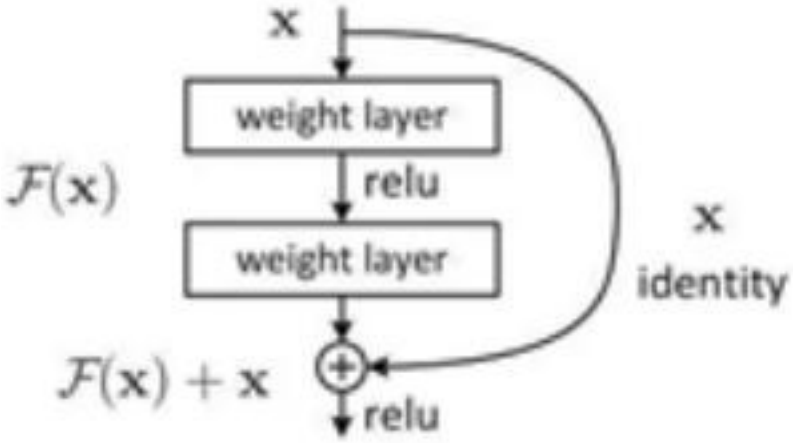

Figure16: residual block structure

6.2.2 The principle of ResNet

Assuming that a shallow network has reached saturation accuracy. Then, several identity mapping layers (y = x, output equals input) are added behind it, which increases the depth of the network, but the error will not increase, that is, a deeper network will not increase the error of the training set.

The calculation formula of residual structure could be shown below:

$$y_1 = h(x_l) + F(x_l, W_l)$$

$$x_{l+1} = f(y_l)$$

And in this formula, x_l and $x_{l=1}$ respectively represents the input and output of the L residual unit, where each residual unit contains a multi- layer structure. F is the residual function, which represents the residual of learning. H(X)= X_l represents the identity mapping, and f is the ReLU activation function.

And it could have a formula with the learning characteristics from shallow l to deep L could be obtained:

$$X_L = X_l + \sum_{i=l}^{L-1} F(x_i, W_i)$$

6.2.3 ResNet 50

ResNet series has many kinds of networks, such as ResNet 18, ResNet 34, ResNet 101, and ResNet 152. These suffix numbers represent networks of different depth. According to the experimental results and the amount of calculation, in this project, it would select ResNet 50.

The ResNet 50 network consists of 49 convolutional layers and a full connection layer, which is also the origin of the name ResNet 50. The ResNet 50 network structure could be divided into seven parts. The first part does not contain residual blocks. It mainly calculates the input convolution, regularization, activation function and maximum pooling. The second, third, fourth and fifth structures all contain residual blocks. They would not change the size of the residual block, whereas only change the dimension of the residual block. The network input is 224* 224* 3. After the convolution calculation, the output is 7* 7* 2048, the pooling layer will convert it into a vector of feature. Finally, the classifier will calculate the vector of vector and output the probability of category.

And the ResNet 50 network architecture map would be shown below

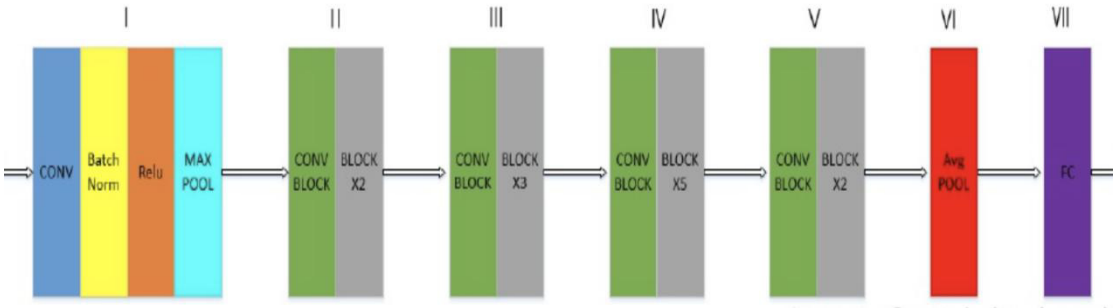

Figure17: ResNet 50 network architecture map

7.Experiment

According to the content which it could be introduced above. In this dissertation, it could know that use the normal algorithms: Local Binary Patterns and Gray Level Co-occurrence Matrix to achieve the extract texture of image, then, this project use our

method ResNet 50 to extract and classify the images. In the end, it should have a comparison above ResNet 50 and LBP, GLCM. In this chapter, the content should analyze the result of ResNet, which is apply for extract texture of images and classify images firstly. Moreover, it could analyze the result of LBP and GLCM in the process of extracting texture of images. Finally, A comparison in this three algorithms should be shown, and it would show the advantage and disadvantage in these algorithms.

7.1 The experimental result and analysis of ResNet

In this last chapter, the ResNet 50 would be applied in this project, and give an introduction of this algorithm. However, How is this accuracy of ResNet 50? and how is the relationship between loss of test and accuracy of test? In this section, it would be solved step by step.

Firstly, it should show that how to calculate the accuracy. As the content which is mentioned above. In this dissertation, an iterative function is introduced to repeat the training many times, so as to obtain many results of accuracy of training and the loss of training. It can be seen from the figure, this training has been iterated 25 times

```
3 Test Loss: 1.7702  Test Acc: 0.5248
3 Train Loss: 1.7840  Train Acc: 0.5060
Epoch 4/25
4 Test Loss: 1.7102  Test Acc: 0.5470
4 Train Loss: 1.6264  Train Acc: 0.5565
Epoch 5/25
5 Test Loss: 1.6661  Test Acc: 0.5638
5 Train Loss: 1.4370  Train Acc: 0.5997
Epoch 6/25
6 Test Loss: 1.6496  Test Acc: 0.5621
6 Train Loss: 1.4013  Train Acc: 0.6148
Epoch 7/25
7 Test Loss: 1.6158  Test Acc: 0.5887
7 Train Loss: 1.3157  Train Acc: 0.6277
Epoch 8/25
8 Test Loss: 1.6832  Test Acc: 0.5762
8 Train Loss: 1.2794  Train Acc: 0.6407
Epoch 9/25

show more (open the raw output data in a text editor) ...

24 Test Loss: 1.8725  Test Acc: 0.6108
24 Train Loss: 0.7526  Train Acc: 0.7926
Epoch 25/25
25 Test Loss: 2.1912  Test Acc: 0.5931
25 Train Loss: 0.8197  Train Acc: 0.7835
```

Figure18: the results of iteration of ResNet 50

According to these 25 results, it could pick up the highest accuracy of test, when the process iterates the 13th. The highest test accuracy is 0.6223 (62.23%), which is the final accuracy in this part. And the accuracy of full dataset could be use the code which is mentioned above to verify, it is a process of verification. The accuracy of full dataset is 0.7369 (73.69%) Similarly, it could find the best model in weight_ 13. pth, which show in the attachment .ZIP files of model files.

Furthermore, it could use a heatmap to show the predicted results. On the diagonals of the picture, the data which is predicted are correct, while the other marked data in picture are wrong data.

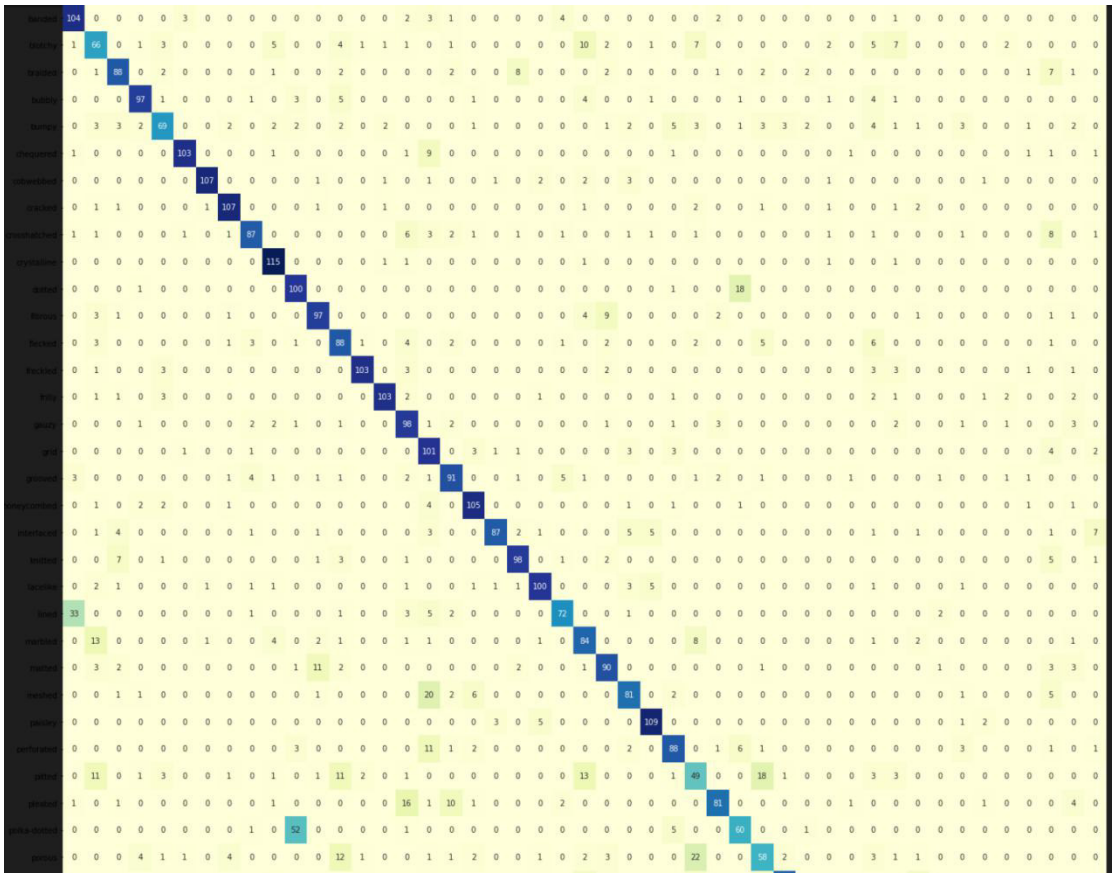

Figure19: the heatmap of results of ResNet 50

Some research have proposed that using the ResNet 50 to process the training of model, the loss of test is decreased continuously and the accuracy of test is increased

gradually. Therefore, it should draw a figure to analyze this rule, which appears in ResNet 50. And the code of drawing picture could be shown below:

```
def draw_train_process(train_loss_all,train_acc_all,test_loss_all,test_acc_all):
    plt.figure(figsize=(18,6))
    plt.subplot(1,2,1)
    plt.plot(list(range(1,26)),train_loss_all,"r.-",label="Train loss")
    plt.plot(list(range(1,26)),test_loss_all,"bs-",label="Test loss")
    plt.legend()
    plt.xlabel("Epoch number",size = 13)
    plt.ylabel("Loss value",size=13)

    plt.subplot(1,2,2)
    plt.plot(list(range(1,26)),train_acc_all,"r.-",label="Train acc")
    plt.plot(list(range(1,26)),test_acc_all,"bs-",label="Test acc")
    plt.legend()
    plt.xlabel("Epoch number",size = 13)
    plt.ylabel("Acc",size=13)
    plt.show()
```

Figure20: the code of drawing the changing

Then, the pictures show the relationship between the test loss and test accuracy. Besides, it could see the train loss and train accuracy.

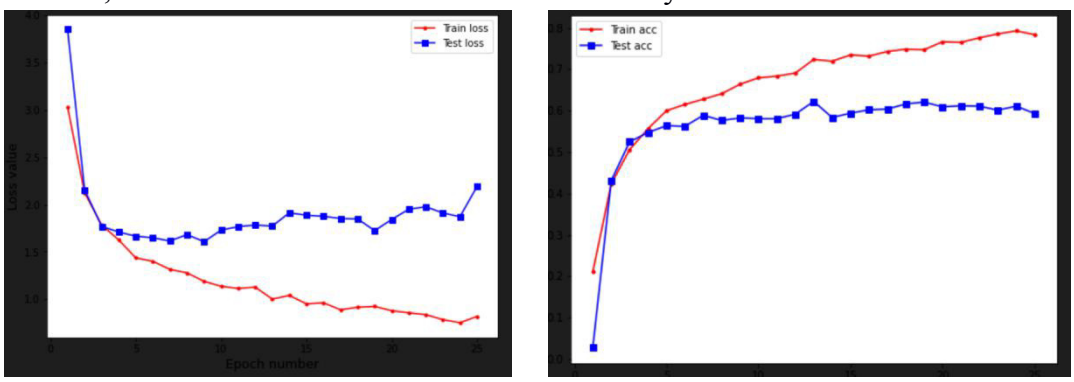

Figure21: the curve between train loss and test loss

Finally, it could discover the rule and relationship between these two in these pictures that when the beginning of training, the test loss and train loss are rapidly decreased, and the train accuracy and test accuracy are improved quickly. After third training or fourth training, The loss of test and accuracy of test tend to be stable. The train loss still reduce and the train accuracy are increased, however, the decreasing tendency and the trend of improving are slowly.

7.2 The experimental result of GLCM and LBP

In previous chapters, they has introduced how GLCM and LBP algorithms extract image texture. It also know that these two methods only extract the texture of the image. If people want to classify images, they have to introduce xgboost and SVM. These two algorithms are two classifiers, which can classify data and predict the accuracy of data. Then, people could use the results given by the classifier for analysis.

Besides, since xgboost and SVM have not been introduced in detail in the previous chapters, it would firstly introduce the application of these two classifiers in data processing.

First of all, Xgboost is improved based on GBDT (gradient boosting decision tree). Compared with the traditional GBDT, XGBoost would supports linear classifier (Yongsheng et al. 2017). It is equivalent to linear regression with two regularizations. Moreover, traditional GBDT only uses the first derivative information in optimization, while XGBoost carries out the second-order Taylor expansion of the cost function, and uses the first and second derivatives at the same time. Besides, XGBoost adds regularized items to the cost function to control the complexity of the model. For samples with missing feature values, XGBoost can automatically learn its splitting direction.

The theoretical basis of SVM is statistical learning. As a supervised learning algorithm, the goal of SVM is not only to classify the different categories of samples, but also to ensure the maximum interval of samples after classification. General SVM can be divided into three categories: linear separable SVM, linear non- separable SVM and non- linear SVM. Linear SVM can be realized by traditional algorithms, but the third SVM needs to choose the appropriate function. Therefore, it is still an academic problem to choose the function. Because there is not optimal criterion to teach people how to choose functions, they only rely on experience to discover the most efficient function. Classical kernel functions have the following categories: linear kernel function, polynomial kernel function and Gaussian kernel function.

7.2.1 The steps of XGBoost algorithm and SVM algorithm

Firstly, it could show the steps of XGBoost algorithm. The first step is that build the decision- tree function. Then, it should ensure the loss function and adapt the MSE. By using the regression tree to lterate and optimize the loss function in the third step. In the end, it could obtain the final XGBoost objective function to process training （Yongsheng and Jinwen 2011）.

Furthermore, determine the training sample set $X_i, i = 1, 2, \ldots, l$ and the labels of

sample is $Y_i \in \{+1, -1\}$ is the first step of SVM algorithm. Then, choosing the optimal kernel function $K(X_i * X_j) = \varphi(X_i * \varphi(X_j))$ and relative parameters. The third step is to calculate the maximum result of $W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$ under the constraint condition: $\sum_{i=1}^{l} y_i \alpha_i = 0$ and $0 \leq \alpha_i \leq C$, i =1, 2, ..., l. Then, calculating $\omega^* = \sum_{i=1}^{l} \alpha_i^* y_i \varphi(x_i)$, and the first component of ω^* is optimum. Finally, it could calculate and obtain the classification of x.

Then, it will be divided into two parts to deeply analyze the classification results of these two algorithms using XGBoost and SVM.

7.2.2 The experimental result of GLCM

In this part, it could be shown that how to import the XGBoost algorithms in training, and the accuracy of dataset.

```
def evaluation(target_lab_all,pre_lab_all):
    plt.figure(figsize=(30,30))
    acc = accuracy_score(target_lab_all,pre_lab_all)
    print("Acc: {:.4f}".format(acc))

    conf_mat = confusion_matrix(target_lab_all,pre_lab_all)
    df_cm = pd.DataFrame(conf_mat, index=class_label, columns=class_label)
    heatmap = sns.heatmap(df_cm,annot=True,fmt="d",cmap="YlGnBu")

    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.show()
```

+ Code + Markdown

```
from sklearn.multiclass import OneVsRestClassifier
from xgboost import XGBClassifier
clf = OneVsRestClassifier(XGBClassifier(eval_metric=['logloss','auc','error']))
clf.fit(X_train, y_train)

ypred = clf.predict(X_test)
evaluation(y_test,ypred)
```

```
C:\Users\Administrator\AppData\Local\Programs\Python\Python38\lib\site-packages\xgb
use of label encoder in XGBClassifier is deprecated and will be removed in a futur
following: 1) Pass option use_label_encoder=False when constructing XGBClassifier
integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

Acc: 0.1968
```

Figure22: the accuracy of using XGBoost in GLCM

According to the result, it could know the accuracy of the dataset is 0.1968(19.68%) This result is far lower than the result of ResNet 50, which show the accuracy of full dataset is 0.7369 (73.69%) therefore, it could discover that the efficiency of classification by XGBoost is lower than that of ResNet 50.

Then, it should move the eyesight on the result of SVM. In this project, the linear SVM would be applied for classification and calculate the result of accuracy in detaset. And the final result could be seen is 0.1534(15.34%). This result is lower than the result of XGBoost, which is demonstrated before.

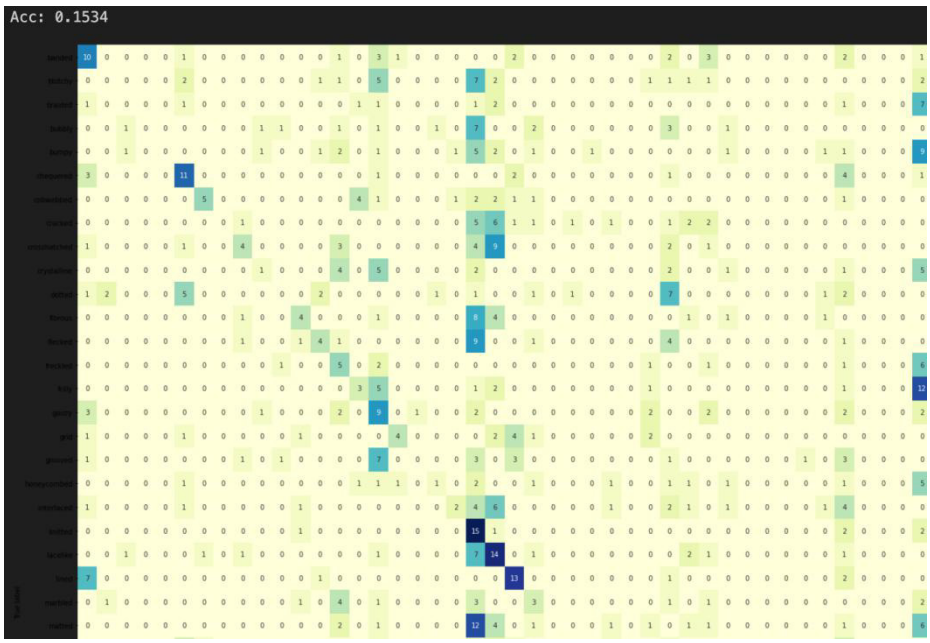

Figure23: the accuracy of using SVM in GLCM

As the content is mentioned above, on this heatmap, people could obtain the correct predicted data in training at the diagonal. However, in the picture generated by the result of SVM, people nearly can not see a complete diagonal. Therefore, the efficiency of linear SVM in classification is still low.

7.2.3 The experimental result of LBP

In the same way, XGBoost and SVM could be applied in the LBP algorithms. And there are two result of accuracy, after using these two algorithms to classify the texture of images. According to the picture is shown below:

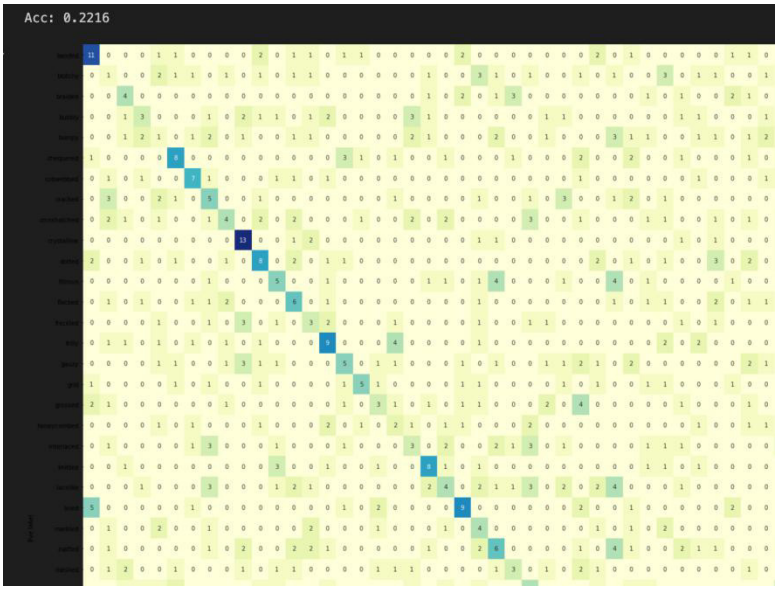

Figure24: the accuracy of using XGBoost in LBP

In this graph, it could see that the accuracy of dataset is 0.2216(22.16%), which is applied the XGBoost. And this result is few higher than the results of GLCM, which is used the XGBoost and SVM. However, this result is far lower than the result of ResNet 50. it could easily find that diagonal which shows the correct predicted data could not form in the heatmap. Moreover, the SVM applies in the LBP also appear a inadequate result. The accuracy only is 0.1383 (13.83%), which is shown the picture below. And in the heatmap, it also could not show a complete diagonal to show the correct predicted data.

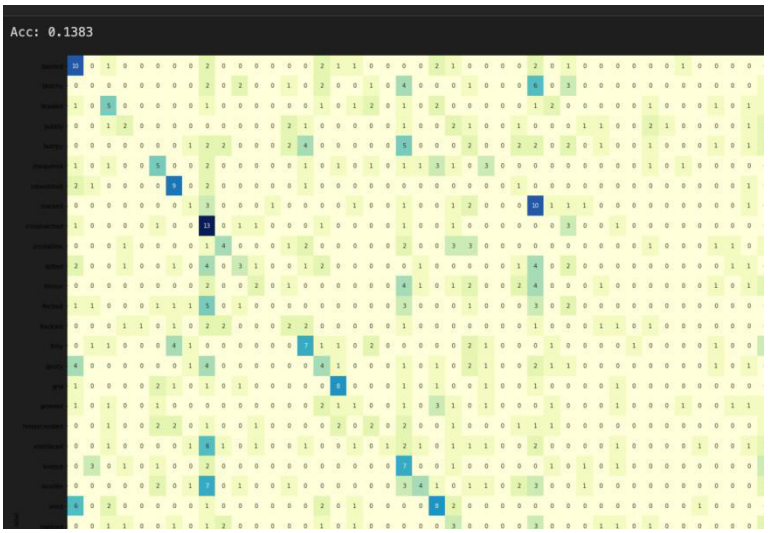

Figure25: the accuracy of using SVM in LBP

According to the result of LBP, which is mentioned above. It could discover that the accuracy of dataset after the LBP apply the XGBoost and SVM algorithms is still low.

In the other word, the XGBoost and SVM algorithms in LBP could not show the best efficiency of training.

7.3 The analysis of the experimental results

In this project, it mainly analyzes the experimental results of ResNet50, LBP and CLCM. From the above results, it can be seen that the accuracy of ResNet 50 is much higher than that of the other two algorithms. Before this project, empirically, the training results of ResNet50 should also be predicted to be better than other algorithms. After precisely adjusting the parameters, the accuracy of the neural network model has reached more than 70%. Therefore, it can be seen that the accuracy of ResNet in image texture classification is high, and it should be used in image texture classification. However, in the experiment, the test accuracy was only more than 60%. The reason may be that there are too many kinds of pictures and the volume of data is small . There are only 100 pictures in a category. However,t he accuracy of 62.23% is better than others.

Moreover, in the implementation of LBP and GLCM algorithms, XGBoost classifierand SVM classifier are used for image texture classification. When using these two kinds of classifiers for training, the accuracy is significantly reduced. Especially when using SVM classifier, the accuracy is much lower than that using XGBoost. Therefore, it can be guessed that linear SVM classifier is not suitable for image texture classification because of its low accuracy.
Finally, the result is that RESNET 50 is efficient in image texture classification. Why do XGBoost and SVM make the efficiency of LBP algorithm and GLCM algorithm extremely low? The reason should be that these two kinds of extraction of features methods are outdated. Secondly, these two methods still have advantages in secondary classification. In this project, the main task is multi- classification of data. Besides, SVM is limited to the level of feature extraction, because image features generally have non- linear separability. Moreover, there are too few data in this project, and the classification advantages of XGBoost can not be reflected. Because the accuracy of xgboost algorithm will be improved when the amount of data is large in training.

8.Conclusion

In conclusion, this project implements three methods (LBP and GLCM would be applied by precursor) to extract the features of images, and the classifying the images. By using the XGBboost and SVM to calculate the accuracy of training, the camparing

the results above three algorithms. This dissertation obtains the importance of image texture classification for computer vision from the application of three different algorithms in image texture classification. According to the results which is mentioned in the chapter 7, ResNet 50 is undoubtedly the best method to extract the image texture and classify the images. Although some aspects in this project need to be improved, the most parts of this project are successful. It could pick up the best method to solve the problems which is mentioned in aims and objectives. Nearly all aims are achieved in the end.

8.1 Benefits and Deficiencies

ResNet 50 applies in the images texture classification has been one of the major successes throughout the project. ResNet is divided into two parts, one is used to extract information and the other is used to classify information. High precision and simplicity make RESNET more and more popular in modern field of computer.

RESNET designs a residual block to increase the network depth. For deeper networks, there is the main issue that the disappear of gradient. Therefore, for layers farther away from loss, the gradient is much likely to disappear. With the increase of the number of layers, this phenomenon becomes more serious. Residual blocks are linked across layers to avoid this problem. Secondly, the maximum pooling and dropout structure are cancelled. Finally, the convolution kernel of the convolution layer is all 3 * 3, and the convolution layer with step size of 2 is used to replace the pooling layer （Parker 2011）.

However, most of things in this world are multifaceted, and RESNET also has disadvantages. First of all, as a method of deep learning training, when the data is too small, the accuracy of ResNet will also decline. Compared with the advanced Googlenet, accuracy of training in ResNet is significantly lower than that of Googlenet in the case of small data. In addition, in this project, it can be found that the training time of ResNet is greater than that of the other two algorithms. obviously, the family of ResNet is expanding. More and more layers are added to the neural network of ResNet, and its disadvantages are constantly optimized. This dissertation selects ResNet50. It can be guessed that if other types of ResNet neural networks are selected, it may have a better effect.

8.2 Future Work

This dissertation focuses on the research of image texture classification algorithms. Although some research progress has been made, there are still many problems that have not been solved. In particular, how to improve the accuracy of texture classification algorithm in the case of small and few data. For the work of this dissertation, the next step of research can be carried out from the following aspects:

8.2.1 Find the optimal parameters of SVM and choose kernel function

Although SVM classification has a good performance in solving most of issues, the performance of SVM is still poor in this project. It can be guessed that the optimal parameters may not be selected. Therefore, an efficient parameter optimization algorithm is a point of breakthrough to improve the classification performance of SVM(Rik 2020). In addition, different kernel functions have a great impact on the classification performance of SVM. As mentioned above, there is not correct criterion to guarantee the most accurate kernel function. Therefore, choosing the appropriate kernel function is still an unsolved problem in SVM.

8.2.2 Invariant of texture features

Texture invariance includes scale- invariance and rotation- invariance（Yongsheng et al. 2015）. If invariant texture features can be extracted, some restrictions on the image can be removed, such as shooting angle, position and size. In this project, statistical method is used to extract texture features. However, research in the future can focus on extracting texture invariance by spectral method. Through the research on the invariance of texture features, images can be classified and recognized without the specified framework.

8.2.3 Feature fusion

A single type of feature can not fully reflect the information of texture. Therefore, by fusing different types of features, including shape and color, it could achieve better results. Moreover, the accuracy of classification can be improved by fusing different classification algorithms, which also belongs to the idea of data fusion.

8.2.4 Multi- scale texture analysis

The texture features are different in different scales of observation. If a method can extract texture features at different scales, it can comprehensively represent features of texture . In the future, wavelet transform can be used to achieve the multi- scale texture analysis.

9.Reflections of learning

During the whole period of the dissertation, it was indicated that it full of pressure and difficulties, whereas also is memorable. At the beginning of this project, I had the basic knowledge of programming in Python, and my knowledge of advanced mathematics was relatively shallow. I felt disoriented and frustrated. Even I wanted to change the title of my dissertation with my supervisor some time. Reading all kinds of

literature made me realize that to better complete this project, I must have unremitting confidence. This will have a great impact on the success of the project. This project has taught me to have goals when doing academic research, set small goals for myself at each stage, and then complete them step by step. Although the scale of this project is still very small, and the main goal is only to complete the implementation of three image texture classification algorithms, it still gives me a lot of valuable academic research experience. Learn how a small project is achieved, which will lead to greater success in another larger task.

The process of this project was never smooth as expected. I am not a student who major in mathematics relatively specialty, even could not understand some complicated mathematical symbol in some articles, which include the content of principle of algorithms. Therefore, i spent a lot of time on studying the advanced mathematical symbol and formulas. However, in the end period of dissertation, i began to realize that achievement of code is the important part of this project, and the comprehend of mathematical definition is the significant part of the theory. Therefore, it could give a new understanding of the academic research for me.

However, because the constraint of time and the restricted knowledge, i understand that to achieve this requirement it was almost impossible, which i could find a balance between practice and theory. Within two months it was unrealistic to understand the every part of the image texture classification algorithms. Therefore, it should pick up the most crucial part in this project to study. Finding the most appropriate algorithms and making adjustments according to the results of algorithms, which has been applied in academic field. Obviously, the mathematical theory is the significant part of this project, but it is only understand by theoretically. In the end, i find the optimal solution to solve this project.

In the process of writing dissertation, there were many difficulties full of the process of writing. For example, i do not know how to achieve a English academic dissertation before. Therefore, i read a lot of article and paper to study the format of English paper. Actually, it is difficult for me to study the format of English paper, but i have tried my best to make perfect nearly all the content of this project. Besides, in the part of related work, i add the part of method, which is almost equal to methodology in English paper. Methodology is few application in Non- English paper, therefore, i ignore to write a detailed methodology in that part, which makes me frustrated.

In conclusion, all the parts of this project was going to hard, but in the end ,the correct results of analysis come out successfully. Therefore, i understand that the process of academic research is hard and boring， if people could overcome these in the process of discovering, it could obtain a lot of satisfaction.

Reference

Chincheng. H and Yihua. L. 2019. Image Texture Analysis Foundations, Models and Algorithms. Cham: Springer international publishing.

Pietikainen. M.K. 2000.Texture analysis in machine vision. Singapore

Yongsheng. D et al. 2017. Multiscale Sampling Based Texture Image Classification. IEEE signal processing letters 24 (5),p.614-618.
DOI:10.1109/LSP.2017.2670026

Hong. L et al. 2012. Fast weighted K-view-voting algorithm for image texture classification. Optical engineering. 51 (2), p.027004-027004. DOI: 10.1117/1.OE.51.2.027004

Thomas. L and Hugues. F. 2020. Modal features for image texture classification. Pattern recognition letters 135, p.249-255. DOI: 10.1016/j.patrec.2020.04.036

Rik. D. 2020. Content-based image classification : efficient machine learning using robust feature extraction techniques.Boca Raton, FL.

Yongsheng. D et al. 2015. Nonnegative Multiresolution Representation-Based Texture Image Classification. ACM transactions on intelligent systems and technology 7 (1), p.1-21. DOI: 10.1145/2738050

Parker, J. R. 2011. Algorithms for image processing and computer vision. Indianapolis, Ind.

Yongsheng. D and Jinwen. M.2011. Wavelet-Based Image Texture Classification Using Local Energy Histograms. IEEE signal processing letters 18 (4), p.247-250. DOI: 10.1109/LSP.2011.2111369

Yang. S et al. 2016. Texture image classification with discriminative neural networks. Computational visual media (Beijing) 2 (4), p.367-377. DOI: 10.1007/s41095-016-0060-6

Hongchang. W. 2021. Texture Image Classification Method of Porcelain Fragments Based on Convolutional Neural Network. Computational intelligence and neuroscience, 2021, p.1-10. DOI: 10.1155/2021/1823930

Jiawang. F et al. 2017 Structural difference histogram representation for texture image classification. IET image processing 11 (2), p.118-125. DOI: 10.1049/iet-ipr.2016.0495.