# .5 Introduction of Deep Learning

Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning [1]. Deep learning architectures can be divided into Feed-Forward-Net and Recurrent-Neural-Network. Feed-Forward-Net is an artificial neural network, which consists of an input layer, some hidden layers and an output layer. The information moves in only one direction, from the input nodes, through the hidden nodes to the output nodes [2]. The fundamental difference of the Recurrent-Neural-Network to Feed-Forward-Net is that RNN can use intern state (memory) to process the variable length of inputs. The representative architecture is long short-term memory.

In our project, all the features extracted from various paper, such as mass, dark grey proportion, are the values with a fixed length, not a sequence of inputs. The outputs are the number of paper class. According to these characters, we selected the architectures, such as Convolutional Neural Network and some pre-trained Network, which belong to Feed-Forward-Network.

## 2.5.1 Convolutional Neural Network

The convolutional neural network is one of the most popular algorithms for deep learning. Most commonly applied to analyze visual imagery [3]. Figure 1 shows the basic architecture of CNN. Like the most other Neural Network, a CNN is composed of an input layer and an output layer and many hidden layers in between. The typical character of CNN is the mathematical operation called convolution. Convolution is a specialized kind of linear operation, and it can create features from input images. In CNN this operation is executed by multiplication of matrix between the input image and a series of convolutional filters. The rectified linear unit (ReLU) can transfer the output of each convolutional layer and allow for nonlinearity. The final part in one feature learning is pooling. It can reduce the number of parameters, which should be learned by network and combine the feature with other pixels together. Commonly, these three operations are repeated over tens or hundreds of layers in management to detect different features.

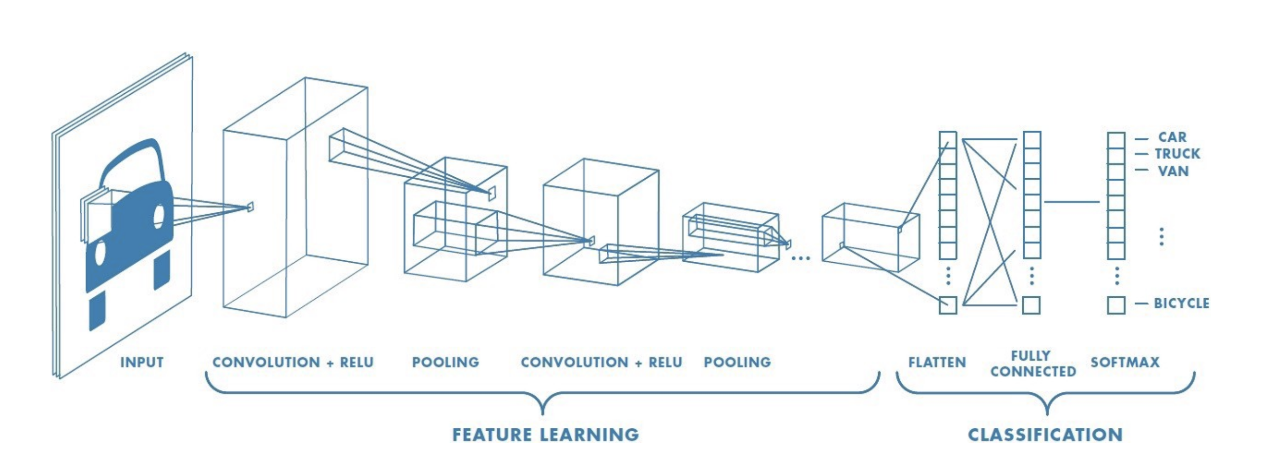


Fig.1 The basic architecture of CNN

After feature detection, the architecture shifts to fully connected layer, which is similar to the traditional multi-layer perceptron neural network (MLP). For the classification problem, there is additionally a softmax layer to calculate the probability of the input in each class.

In principle, CNN is applied in image classification, which belongs to the 2-D problem. In our project, CNN is also planned to apply in the NIR spectrum, in order to predict the content of Kaolin. However, the NIR spectrum is a 1-D spectrum. It means that directly using CNN is not possible. Chenhao Cui [4]from University College London has proposed a new method about the implementation of traditional CNN in the spectroscopic analysis by changing the input size of the input layer and build a new series of convolution filters which are suitable for 1-D input data. Malek, Melgani and Bazi [5] also explored 1D-CNN for spectroscopic regressions. In figure 2 shows the converted architecture of a CNN, which can be applied in the NIR spectrum. The concrete process included feature extraction, and pre-processing will be introduced in chapter 3.

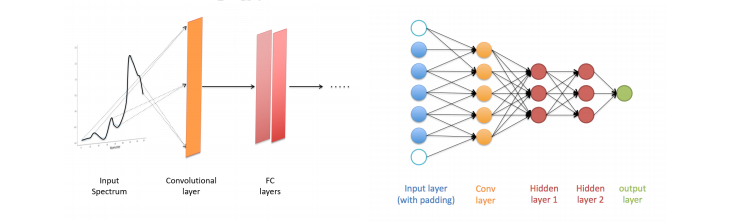


Fig.2 CNN architecture applied NIR spectroscopic analysis [4]

The most significant advantage of CNN compared to traditional machine learning is that CNN does not need to extract the feature manually. The convolutional filter can detect the feature from computer vision (end to end learning). It can create more reliable features than them from the algorithm written by a human.

## 2.5.2 Transfer learning

Although data mining and machine learning technologies have already succeeded in many knowledge engineering areas including classification, regression and clustering[6], for deep learning network, it still needs a large amount of data for training and testing, if a neural network needs to be built for a particular application. Nowadays, in some applications (such as in paper classification), data collection is still a troublesome and expensive work. In that case, Transfer learning is a desirable choice. Instead of building a new neural network, transfer learning can achieve a relatively precise result by fine-tuning based on a pre-trained network. Nowadays, there are various pre-trained architectures of neural network. Most of them can be executed in image classification problem.

Since there are totally 4000 objects can be used for paper classification, it is impossible to build a new neural network as a classifier. Therefore, it is more property to build a classification neural network by transfer learning. In this chapter, five kinds of typical architecture, which are used in this project, will be introduced. They are AlexNet, SqueezeNet, GoogLeNet, VGG16 and ResNet 18. All the following introduction of the neural network are referred to the architecture in Matlab® 2020a

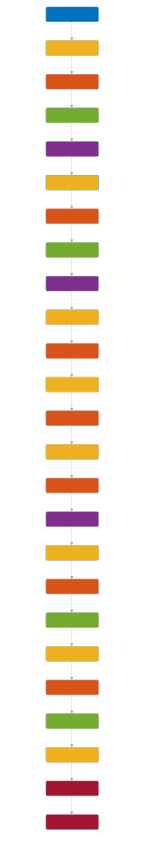
**AlexNet** is a typical convolutional neural network. Figure 3 shows the overview architecture of AlexNet. It contains eight learnable layers – five convolutional and three fully-connected.

Fig. 3 Architecture of AlexNet

The output of the last fully connected layer is fed to a 1000-way softmax which produces a distribution over the 1000 class labels [7]. The Outstanding of AlexNet compared to the other Neural Network before it is that it enlarges the learning capacity of CNNs by increasing the depth and breadth of a CNN. Additionally, AlexNet uses ReLU Nonlinearity and Local Response Normalization as depicted in Figure 4, in order to reduce the overfitting problem during the training. Therefore, Alexnet has better robustness and compared to the other popular pre-trained network such as GoogLeNet, which are also used in this project, it also has a simpler architecture.

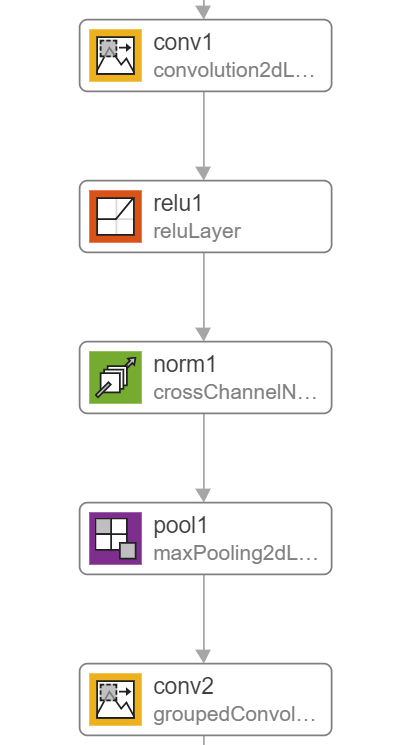
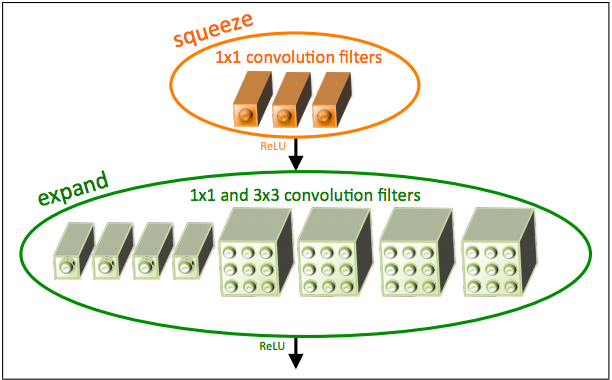


Fig.4 The first feature detection layer in AlexNet

**SqueezeNet** can be regarded as an alternative of AlexNet with fewer parameters, which means it can maintain high accuracy and at the same time, reduce the calculation time and the requiring computer memory. However, even if SqueezeNet is comparable with AlexNet, it does not mean that SqueezeNet is a “squeeze version” of AlexNet. The structure of SqueezeNet is different from AlexNet. The critical component in SqueezeNet is Fire Module[13]: A Fire module is comprised of a squeeze convolution layer (which has only 1x1 filters), feeding into an expand layer that has a mix of 1x1 and 3x3 convolution filters (Figure 5). Moreover, a global average pooling layer takes the place of the fully connected layer in SqueezeNet.

Fig 5 Organization of convolution filters in the Fire module

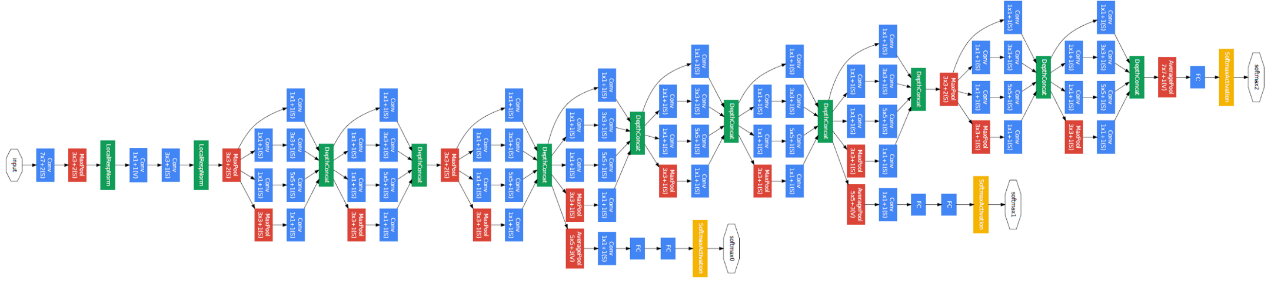
**GoogLeNet** is another convolutional neural network with 22 layers deep. Its critical architecture is called inception. Figure 5 shows an overview of the GoogLeNet.

Fig 5 Architecture of GoogLeNet

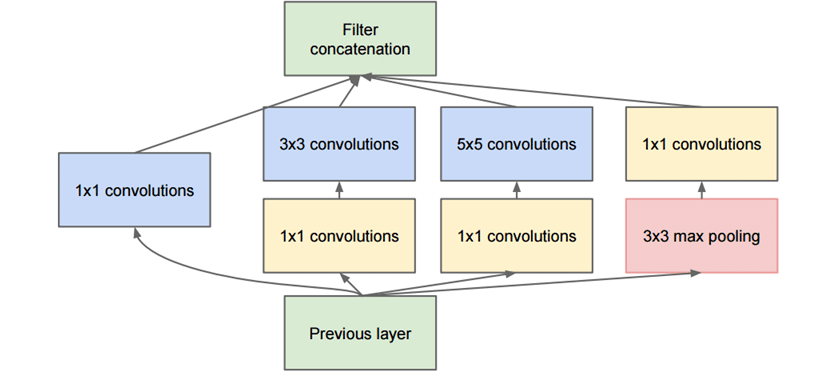
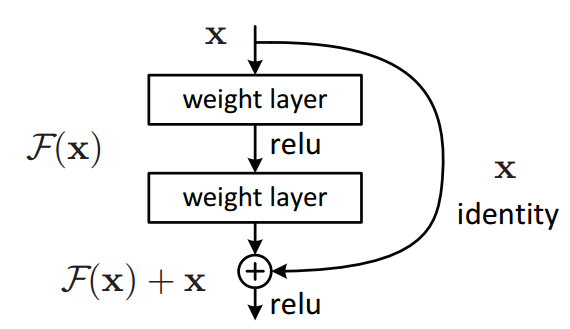
The main idea of the Inception module (Figure 6) is that it stacks 1x1 convolution for computing reductions before the expensive 3 x 3 and 5 x 5 convolutions [8]. The function is similar to the PCA, which maintains the height and width of the feature map but reduces the depth. The advantage is learning efficiency. Besides, compared to Alexnet, GoogLeNet reduced the number of parameters from 60 million to 4 million.

Fig. 6 Inception module

**ResNet** is an artificial neural network and uses residual learning function instead of the unreferenced learning function. The hallmark of ResNet is that it can solve the vanish gradient problem in the deep layer. The principle of residual learning is described in Figure 7. Instead of underlying mapping H(x), Residual mapping F(x) is as input to fit each layer. In this time, the original feature map H(x) equals to F(x) + x. This formulation can be realized by a feedforward neural network with "shortcut connection" [9].

Fig. 7 Residual learning: a building block

As mentioned before, in AlexNet, there are many 11 x 11 and 5 x 5 convolutions. They cause much kernel parameters and decrease the calculation performance. However, a large convolution kernel can enlarge the receptive field and combine more information from an image, which is benefited to increase the quality of the generated feature maps. GoogLeNet solves this problem by stacking a 1 x 1 convolution to reduce the depth of the feature map.

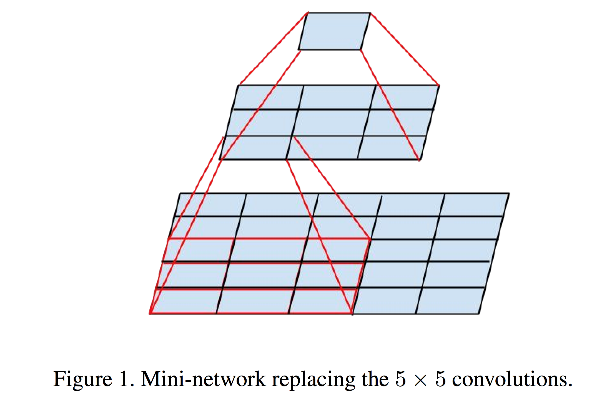
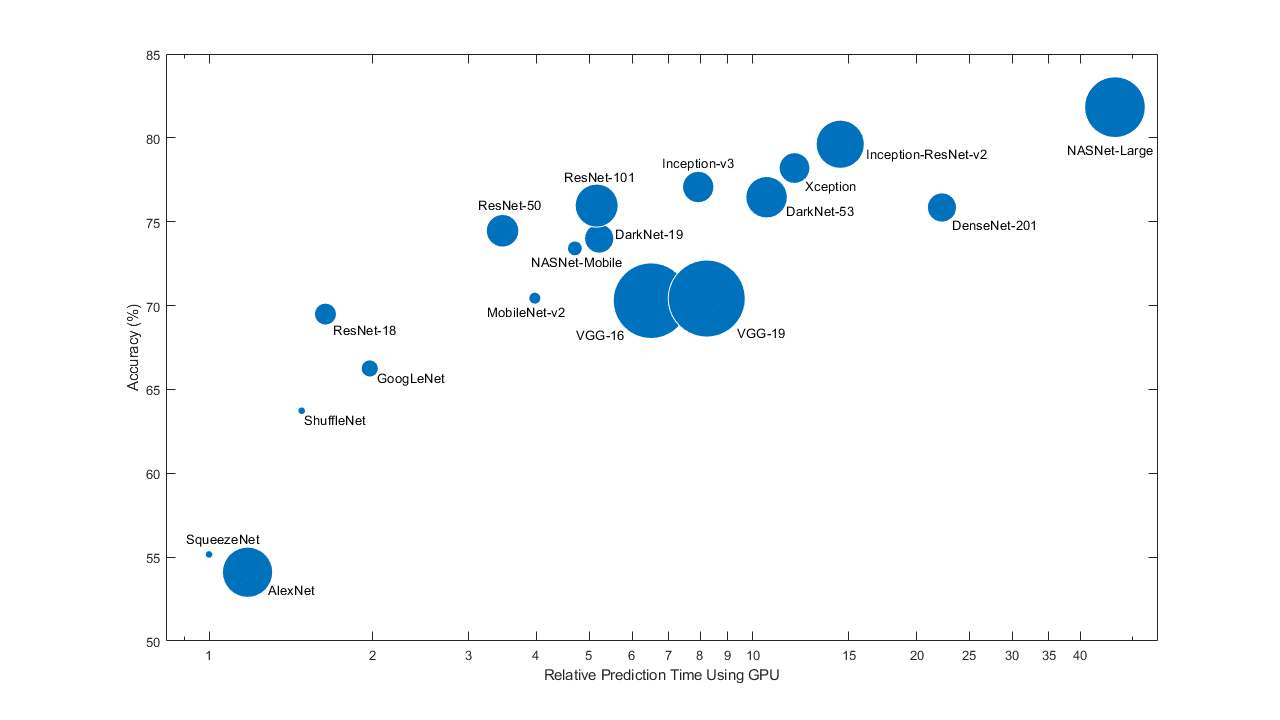
**VGGNet** is also a convolutional neural network, which can also solve this dilemma from another perspective. VGGNet uses small 3 x 3 receptive fields through the whole net [10]. As shown in Figure 8, a 5 x 5 convolution layer can be replaced by a stack of two 3 x 3 convolution layers. It is confirmed that a stack of two 3 x 3 convolution has more effective receptive field than a 5 x 5 convolutions. By analogy three such layers can achieve the similar function of a 7 x 7 convolution layer.

Fig. 10 Mini-network replacing the 5 x 5 convolutions

Figure 11 shows the performance of different training model according to the accuracy, speed and utilization of computer resource (especially in GPU), which are the most important characteristics of a model.

Fig. 10 Characteristic of the different pre-trained model

It can be concluded that the increase in accuracy of the model comes with sacrificing calculation time. Therefore, choosing a network is generally needs to make a tradeoff among these characteristics. Table 1 shows a summary of the pre-trained model, which are used in this project. From the theoretical perspective, it is still hard to specify which model is the properest. The results of using the different models will be introduced in chapter 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Network | Depth | Parameter | Size | Characteristic |
| AlexNet | 8 | 61 Million | 227 MB | Parallel Working on two GPU |
| SqueezeNet | 18 | 1.24 Million | 4.6 MB | fewer parameter and smaller size |
| GoogLeNet | 22 | 7.0 Million | 27 MB | Inception module |
| ResNet 18 | 18 | 11.7 Million | 44 MB | Residual learning function |
| VGG16 | 16 | 138 Million | 515 MB | Stacked convolutions |

Table 1 Overview of different model

# 3.1 Feature from NIR Spectrum

## 3.1.1 Hyperspectral imaging

The hyperspectral camera used in our project is an FX17 from SPECIM Imaging Ltd., which can provide totally 224 wavelengths over a range from 900nm to 1700nm [11]. This camera can scan 640 pixels in line and provides the spatial resolution of 0.94 nm. The camera can take a picture in every trigger and take totally 70 line-scans. All the information is stored in a hypercube, whose length and width represent location and depth represents the wavelength.

Since the camera can only capture the intensity of light, it still needs to be corrected, in order to get the reflection value. The dark background can absorb all the light and then even no photon can reflect back to the sensor. The intensity is as *B* recorded. The intensity of the white background can be recorded as *W* in the same way. Using these two parameters can the actual reflectance of Object with intensity of object be calculated in following equation 3-1

3-1

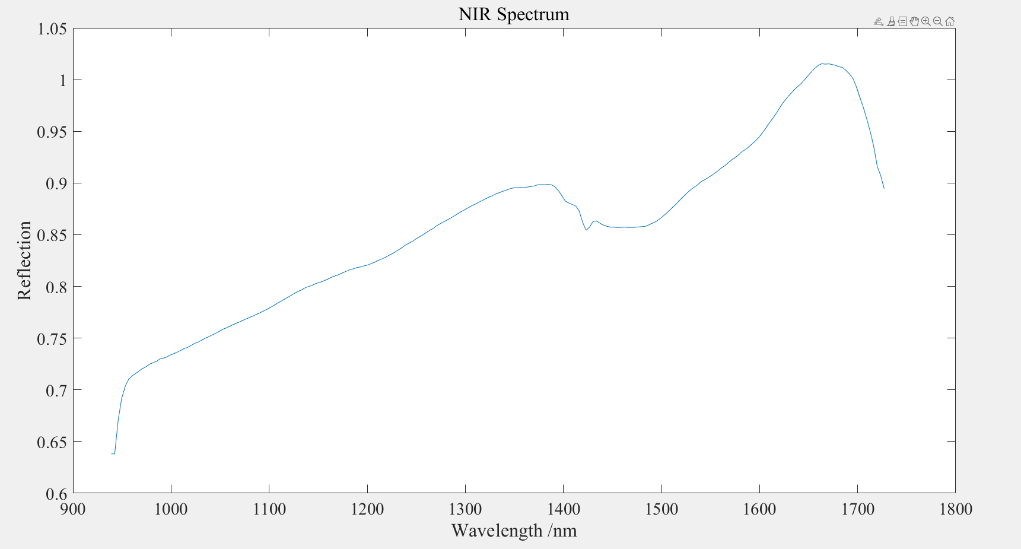
In order to decide the range of interest (ROI), the value of the 100th channel is checked to distinguish the conveyor belt and object. Those pixels, where the intensity is very low, are labelled to black. Other pixels inversely will be labelled to white. In order to avoid the influence of the object's bounding, the pixels at the bounding of the object are also labelled to black. Only the white-labelled area is used for the feature project (interest-region). Since the size of the original hypercube is too large, only the average value of each channel (wavelength) is recorded, and the whole 224 values are regarded as the final NIR spectrum for an object. A typical spectrum is shown in Figure 3-1.

Fig. 3-1 NIR Spectrum after the process of hyperspectral image

It can be detected that there is a spike between the 1400nm and 1450nm. This spike is regarded as a relevant factor to Kaolin content. The content of Kaolin varies according to the different paper class, especially in the grey carton and in the white carton. The difference between these two classes is always hard for the computer version.

## 3.1.2 NIR Spectrum pre-processing

**Select Region of Interest**

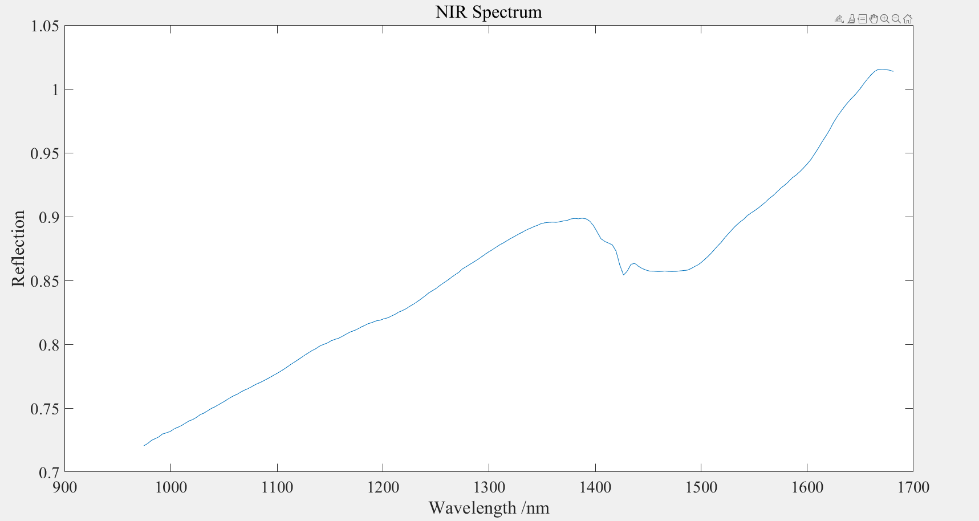
As Figure 3-1 described, the spectrum has a rapid upward trend at the beginning. In this wavelength range, the spectrum contains too much noise, and these values are also useless for the classification and regression. The spectrum at the end of the mess range is also unstable. Therefore, a range of wavelength should be decided, in order to obtain a data set with good quality. In our project, the region of interest (ROI) is from 974 nm to 1680 nm. The cropped spectrum is as shown in Figure 3-2.

Fig. 3-2 NIR Spectrum after ROI process

**Savitzky Golay Filter and Median Filter**

As mentioned in section 3.1.1, the spike appearing in the NIR spectrum is a very important feature for classification. Savitzky Golay Filter can increase the spectral resolution and remove the background. In general, Savitzky Golay Filter is often used with the second derivative. Because it can not only remove the constant and linear background but also be easier to interpretation [12]. However, the drawback of the differentiation is that it also amplifies the noise. Therefore, in our project, before Savitzky Golay Filter, a Median Filter is used to remove the noise. Besides, the moving window in Savitzky Golay Filter can also smooth the spectrum. In Matlab, the window size and the order of polynomial order in Savitzky Golay Filter function should be carefully selected. If the window size is too big, it will filter out the relevant spike. In our project, the window size is 11, and the order of the polynomial is 3. For Kaolin content regression, the first and second derivatives. For paper classification, only the first derivative is used to process the NIR spectrum. Figure 3-3 shows the result with the Median filter and SG filter in the second order.

(此处有图)

**Standard Normal Variate**

Standard Normal Variate is a common method to handle the NIR spectrum. SNV can remove multiplicative interferences of scatter and particle size [12]. The spectrum after SNV is centred to have a mean value 0 and scaled to have a standard deviation 1. The equation is depicted as 3-2

3-2

represents the mean value of variable and is the standard deviate of variable . In our project, the number of the row represents the amount of NIR spectrum, and each column contains the value of the corresponding wavelength. Therefore, SNV is executed along the row dimension. Especially for the classification, it is also necessary to execute the SNV method along every column in order to make the feature matrix more property for machine learning.

(此处有图)

**Offset Correction**

Offset Correction is an alternative method to SNV in our project. It is not a popular method and executed only when it causes singular value by using the SNV method. In our project, these phenomenon occurs in linear discriminant analysis method that will be introduced in section 3.1.3.

Offset correction is applied to correct for a parallel baseline shift (Figure 3-5). The principle of this method is arbitrary chosen value and Subtraction from each spectrum independently. In our project, the mean reflection of first five value in the spectrum is calculated and used to correct.

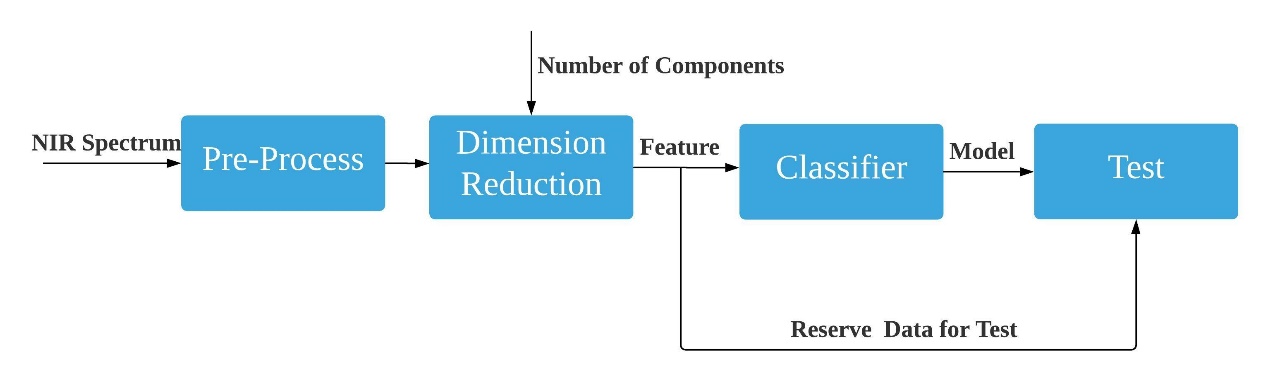
3-3

Where is the corrected spectrum, is the original spectrum and is the mean value of the first five wavelengths. The processed result is shown in Figure 3-6.

(此处有图)

## 3.1.4 Feature Extraction for paper classification

Some dimension reduction methods have been introduced in chapter 2. In this section, the particular application of these methods in feature extraction for paper classification will be introduced. Although different dimension reduction methods are based on various principles, they follow the same working flow of paper classification as Figure 3-7

Fig. 3-7 Common Process of Paper classification based on the NIR Spectrum

In pre-processing, the common pre-processing like Select of ROI, SNV and SG filter are used in PLS, PCA and Autoencoder methods. Especially in our project, if the SNV along the column direction is executed in the LDA method, it will cause singular value in the extracted feature matrix. These singular value cannot be regarded as features in machine learning application. Therefore, the alternative offset correction method is used here instead of SNV along each wavelength (column direction).

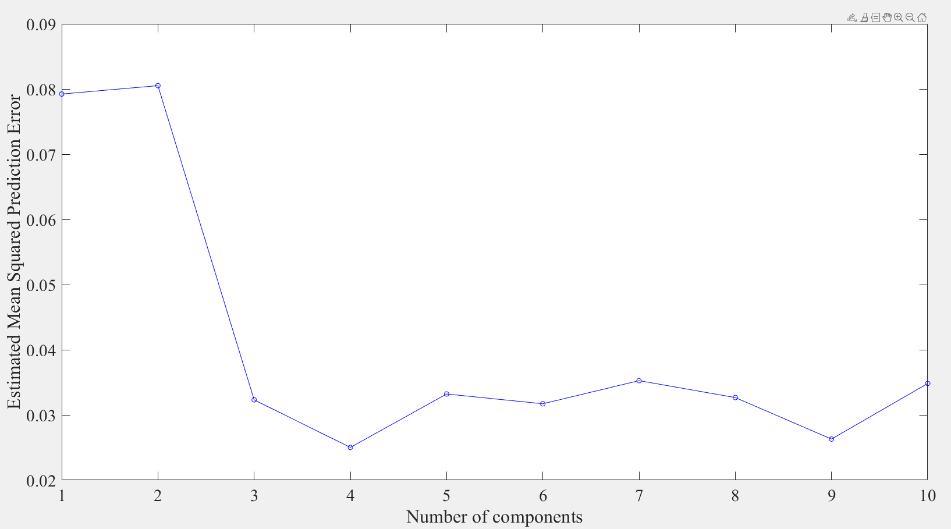
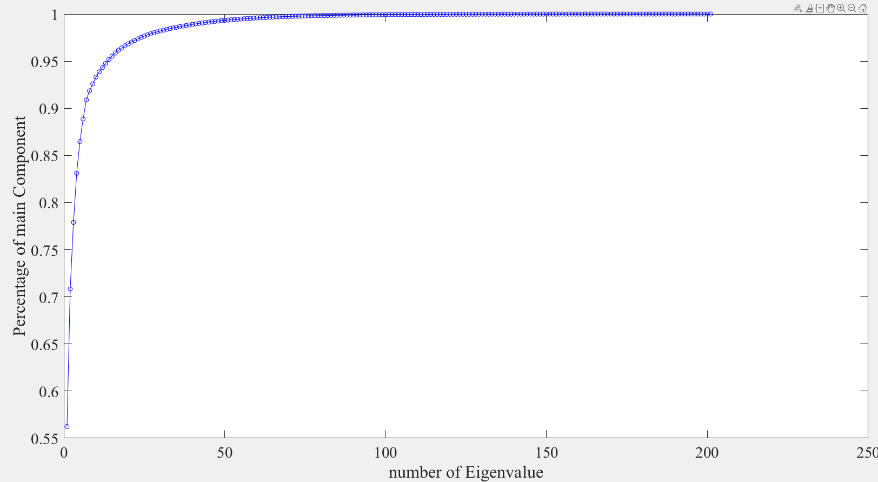
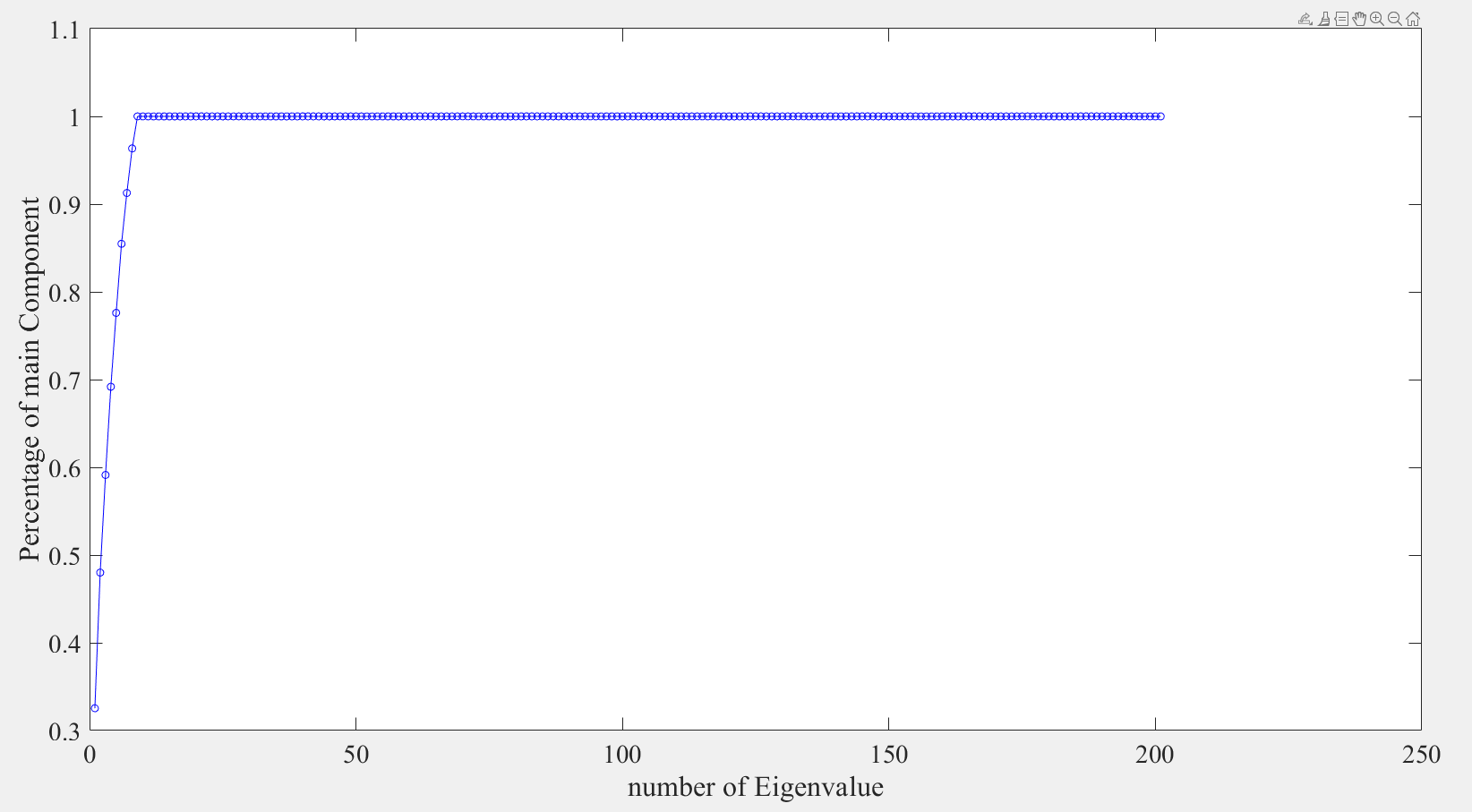
Afterwards, the number of components should be decided. The criteria of component selection are based on the principle of the methods: In PLS the MSE between the response calculated from first n components and the original label can suggest the optimal amounts of components. This value means that how completely the reduced predictors describes the observation. Usually, the optimal point in the graphic 3-9 is selected, when before this point, the mean square error declines rapidly but afterwards slowly. This point is also regarded as “Elbow Point”.

Fig 3-9 EMSE in each component of PLS

According to the principle of PCA, the nth eigenvalue of the XTX represents the data variation in the nth component. It can also be regarded as the criteria for selecting the optimal amount of PCA component. Therefore, we cumulatively sum up each eigenvalue step by step and calculate the ratio to represent the contribution of the first n components. Similarly, we also select the elbow point in the graphic. In our project, the optimal amount of principal components is 7 (Figure 3-10 left).

Fig 3-10 Cumulative Summation of Intern Variance in percentage among each component in PCA

In the LDA method, the eigenvalue of the matrix represents the between variance and within variance. According to the principle of LDA, we should find the best amounts of components, where the transformed input have the biggest between variance and smallest within variance. Therefore, similar to the process of PCA, the cumulative sum of each eigenvalue is plotted as a graphic. In the graphic (Figure 3-10), eight components can already occupy fast 95% of the total value. The later components only have a small contribution to catch information from the original spectrum. Hence we select the first eight components in LDA method and transform the original NIR spectrum in 8 dimensions.

Fig 3-11 Cumulative Summation of Eigenvalue in percentage among each component in LDA

There are no standard criteria for autoencoder to decide how many neurons are proper for dimension reduction. Therefore, based on the principle of autoencoder, we calculate the residual between the original input and reconstructed input from autoencoder network. If the residual is small enough, we consider that neurons in the hidden layer are able to represent the original NIR spectrum. There is a rule of thumb in autoencoder: The amount of neurons in the hidden layer should not be less than the amount of classes. Since there are ten classes in our project, we define ten neurons in the hidden layer and train the network. The result of the residual is in this time 0.078. It is small enough to rebuild the original spectrum. Therefore, the neurons in the hidden layer can be regarded as features for further machine learning.

In summary, the result of feature extraction based on the different methods is recorded in Table 3-2. Thanks to these dimension reduction methods, the original NIR spectrum with 224 wavelengths can be reduced in the lower dimensions and at the same time, they preserve the most information from the original spectrum. It is worth noting that in PLS method we did not select the elbow point as the amount of component, because we find that when we select the elbow point (4) in PLS method, the validation accuracy of the trained model is only 76%. The reason is although at four components the residual error is smallest, the transformed data in low dimension is so compressed that the classifier such as SVM cannot find a proper dimension, where it can define the class boundary. After trying with different components and comparing the validation accuracies, we find that seven components of PLS as features for our project have the best performance. The result of classification will be introduced in detail in chapter 4.

|  |  |
| --- | --- |
| Method | Amounts of Feature after Reduction |
| PLS | 7 |
| PCA | 7 |
| LDA | 8 |
| Autoencoder | 10 |

Table 3-2 Feature Extraction based on different methods

# 4.1 Machine Learning

## 4.1.3 The Result of Classification

In this section, the result of paper classification only based on the feature from the NIR spectrum. As mentioned in 4.1.1, The NIR spectrum in our project is easy to be influenced by some environmental factors such as lighting temperature. In order to prove this problem, we firstly used the 3000 objects as the training dataset, where each object is scanned in the order of class and 1000 objects as test dataset, where the order of scanning is random. SVM is used as the classifier to train the specific model. The results of each method mentioned in Section 3.1.4 are recorded in Table 4-1

|  |  |  |
| --- | --- | --- |
| Method(Components) | Validation Accuracy | Test Accuracy |
| PLS (7) | 91% | 90% |
| PCA (7) | 90% | 90% |
| LDA (8) | 98% | 88% |
| Autoencoder (10) | 91% | 91% |

Table 4-1 Results of Paper Classification based on SVM in different methods

The second column in the table shows validation accuracy in the principle of 10-fold-cross validation. It can be concluded that the training model of features from each reduction method can achieve high accuracy with cross-validation. Especially when we use the LDA method to extract the feature, the validation accuracy can reach even 98%. However, the exorbitant validation accuracy of a model often comes with the hidden danger of overfitting. This phenomenon occurs in test accuracy. Although according to the total accuracy of the test dataset, the LDA method seems to have a good performance of the independent test dataset. However, from confusion matrix(Figure 4-1) it can be seen that fast all the objects from tenth class white corrugated paper are misclassified and according to the property of LDA principle the false label prefer to concentrate in one class, in our project is the ninth class corrugated paper. The similar misclassification problem between these two classes also occurs in other methods. Nevertheless, no method has such a serious misclassification problem as LDA method. Since the disadvantage of the LDA is that it is easy to become overfitting and besides it cannot guarantee that NIR spectrum from our project is not influenced by environment noise, it can be concluded that LDA method is not proper to process the NIR spectrum even if it can separate the object in lower dimensions very well.

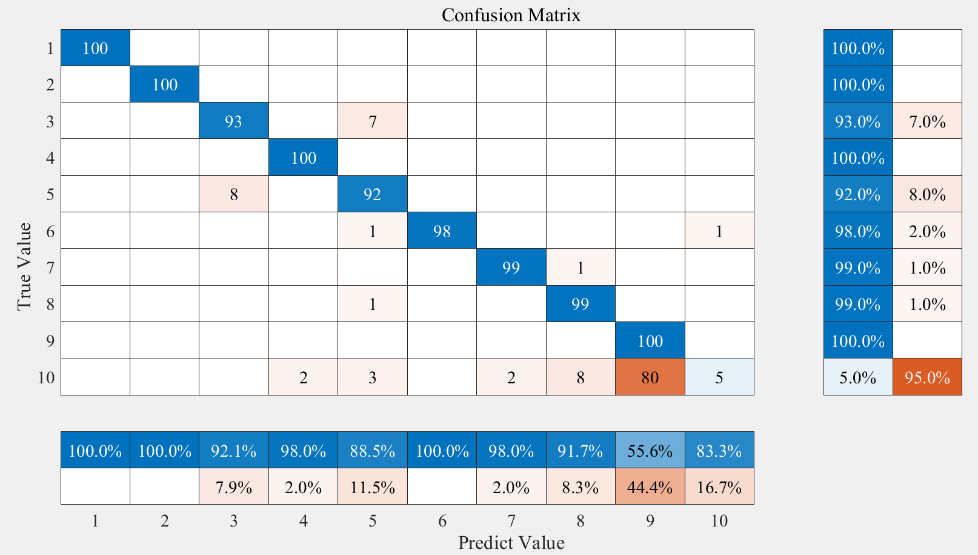


Fig. 4-1 Confusion Matrix with LDA methods on 1000 independent datasets

In order to improve this method, we put the first 50 objects in each class into the training dataset additionally, whose spectrums are scanned randomly. In this time, there are totally paper 3500 objects for training, where each class consists of 50 objects with random scanning order and 300 with sequence scanning order. The result based on using the new mix dataset is shown in table 4-2. The results of classifiers trained by feature from PCA, PLS and Autoencoder methods is no significant difference to previous results. However, at this time, the LDA classifier has much better performance both in total accuracy and confusion matrix (Figure 4-2). Although the overfitting problem still exists in the tenth class, it improves a lot.

|  |  |  |
| --- | --- | --- |
| Method(Components) | Validation Accuracy | Test Accuracy |
| PLS (7) | 91% | 91% |
| PCA (7) | 91% | 89% |
| LDA (8) | 99% | 95% |
| Autoencoder (10) | 92% | 91% |

Table 4-2 Results of Paper Classification based on SVM in different methods

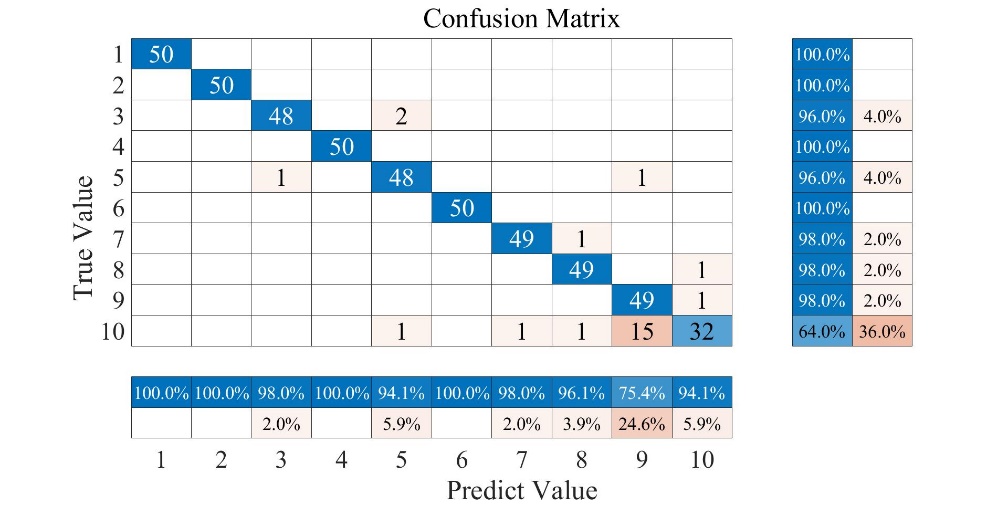


Fig. 4-2 Confusion Matrix with LDA methods on 500 independent dataset

However, it is worth noting that the improvement in LDA comes with the sacrifice of the test dataset. Moreover, only the result of LDA method is benefited from the modification in the training dataset. The other three methods (PLS, PCA and Autoencoder) already have good performance and robustness on the independent test data. Therefore, it makes no sense for these three methods to use the mix training dataset.

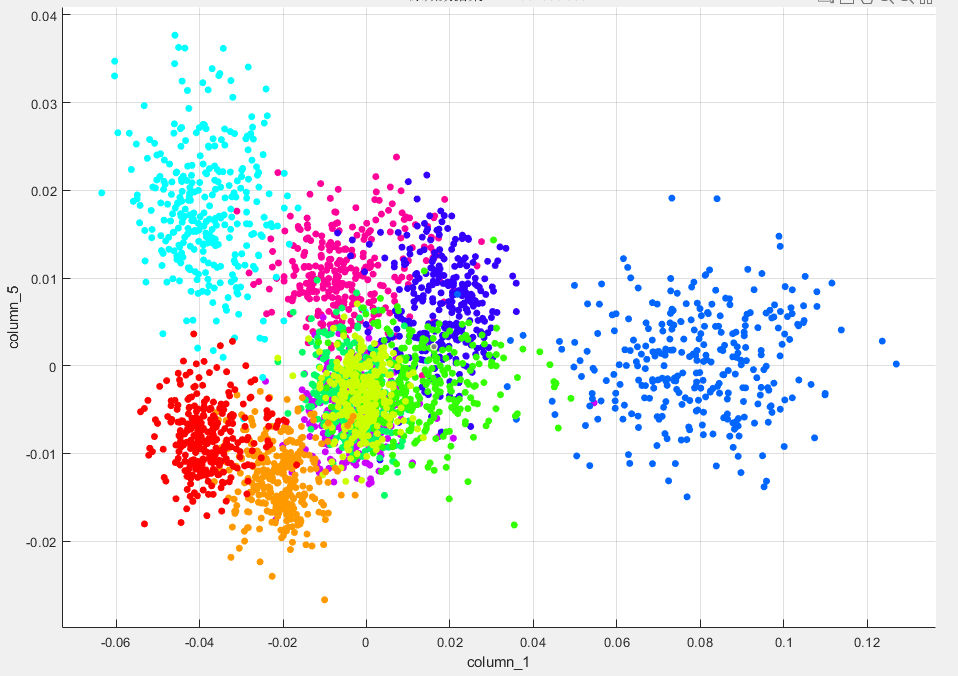
In conclusion, according to the dataset we have yet, we recommend that autoencoder and PLS are the most suitable method for feature extraction in the NIR spectrum because of robustness and relatively high accuracy. The property of the LDA method restricts its application in processing the NIR spectrum. However, the LDA methods can already get a better separation among different classes in low dimensions (Figure 4-3). Without any machine learning algorithms, it can get some class boundaries directly. Since the overfitting problem in LDA and it cannot guarantee a noise-free spectrum in our project, there is still potential to improve this method. For example, hence the new mixed training dataset in our project can improve the problem. It is desired to build a new training dataset with random scanning order, in order to remove the environment effect.

Fig. 4-3 Feature Extraction by LDA method

# 4.3 Combination of Deep learning and Machine learning

According to the above discussion, it can be concluded that both machine learning and deep learning have their own advantages in our project. In machine learning, there are various options of classifiers and a wide range of feature extraction methods for a user to choose. The flexibility is another hallmark of machine learning. We can combine different features from different methods and detect which combination has the best result. However, we should intuitively know which feature is proper to produce the result and has a good understanding of the dataset. On the contrary, we do not need to understand the principle of feature extraction in deep learning because the neural network will choose the best feature by itself. Moreover, the accuracy in some models is very high. Nevertheless, the training time and semi-black box principle should also be considered. Especially if the result of a neural network is not unsatisfied, it is hard for the user to figure out the problem and debug it.

Therefore, finally, we try to combine these two methods to see if there is any improvement. Section 4.3.1 will introduce the combination, where we select a pre-trained deep learning network as feature extractor and use SVM to train a classifier. Section 4.3.2 will introduce the other combination, where we inversely use image feature extraction method by ourselves and train a simple multi-layer perceptron.

## 4.3.1 Combine ResNet18 and SVM classifier

Fig. 4-5 Process of the combination of ResNet 18 and SVM

Figure 4-4 shows the working process of this combination. The dataset is the same as separation in other methods. We use 3500 mixed objects as training dataset and 500 as test dataset. We select the ResNet18 as feature extractor because it has the best performance and relative faster calculation speed in section 4.2. Therefore firstly we compress the image with image pre-processing for ResNet 18, as mentioned in section 4.2. Then we select the value in the latest convolutional layer”pool5” as the feature, because the image has gone through all the convolutional layer in the ResNet before this layer. We can regard the value in this layer as what computer vision has learned from the image. Afterwards, similar to the process of machine learning we have done in section 4.1, we choose SVM as a classifier and train a model to fit this feature. Finally, we apply the trained model in the test set and get the final accuracy (Table 4-3).

|  |  |  |
| --- | --- | --- |
| Method(Features) | Validation Accuracy | Test Accuracy |
| ResNet + SVM (512) | 98% | 95% |

Table 4-3 Result of the combination of ResNet 18 and SVM

The combination of two approaches can get higher accuracy than simple ResNet and this accuracy is close to the best result we can get with machine learning in section 4.1, but in this time we do not need to extract feature and combine different methods by ourselves. This end to end learning is relatively convenient and at the same time, maintain high accuracy.

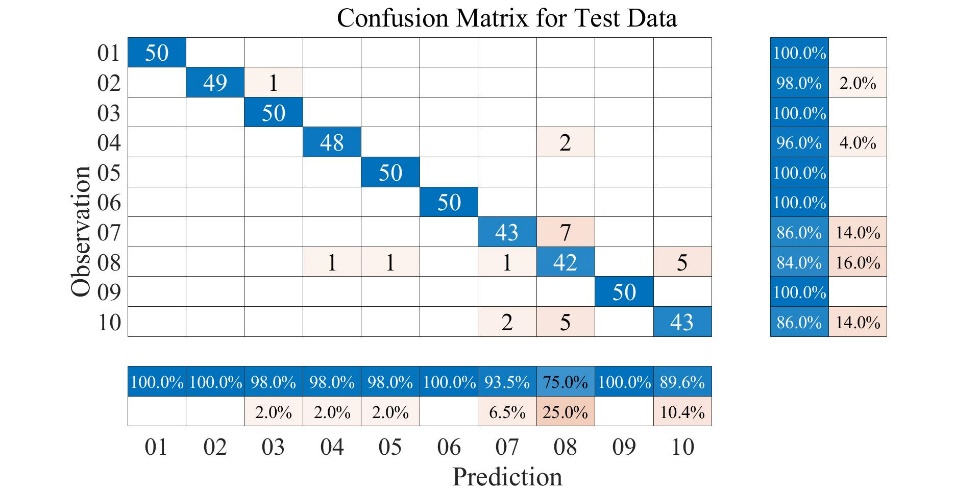


Fig. 4-6 Confusion matrix of combination approach: ResNet + SVM