Machine Learning Week5 group project report

Zhipeng Fu
Trinity College Dublin
22309070

Jiyuan Liu
Trinity College Dublin
22305480

Shuo Jia
Trinity College Dublin
22301057

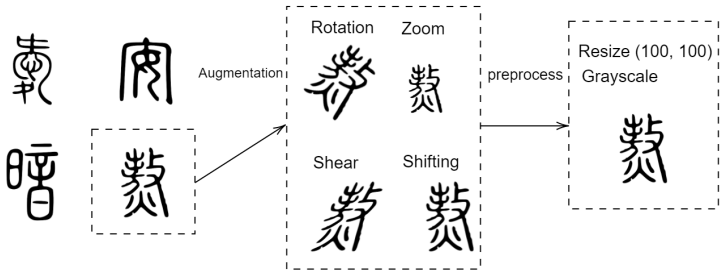
Date: December 3, 2022

1 Introduction

In museums or some ancient books, we often see many ancient Chinese characters. But as we all know, it is impossible to type ancient Chinese characters by our mobile or PC and it is also a hard thing for most of the Chinese people to distinguish them because they look very different from simplified Chinese. So even though we are curious about those ancient characters, we have no method to search them. In the development history of Chinese characters, seal script is a very meaningful and widely circulated font. The main task of the project is to match seal script Chinese characters with simplified Chinese characters through image recognition, so as to help people gain a better understanding of ancient Chinese texts.

The input to our algorithm is 100 * 100 resolution and 1-channel grayscale color images of 50 ancient Chinese characters in seal script with their labels. We then use AlexNet model of CNN and SVM as the baseline on high-performance computer to perform image 50-classification.

2 Dataset and Features



The dataset we use contains 2107 images of seal script ancient Chinese characters belonging to 50 classes. Dataset derived from

- Website(approximate 50*12): Wiktionary,Chinese dictionary website.
- Handwrite(50*6): photographs of handwritten seal script characters.

- Image augmentation(1207): including rotation, shifting, zoom, and shear, as image shown above. The image augmentation techniques generated similar but different training samples after a series of random changes to the training images. It improved the generalization ability of the model by reducing the dependence of the model on certain attributes

We preprocess the image to resize to 100*100 and 227*227, which is the input shape of Alexnet and transform to grayscale.

3 Methods

Alexnet

The AlexNet network structure using 8 layers of convolutional neural network, the first 5 layers are convolutional layers, and the remaining 3 layers are fully connected layers. Convolutional layer and maxpooling layer and full connect layer are the classic elements in CNN, and were taught in lecture, but this structure also use batch normalization in convolutional layers, here is the detail of this layer:

- Pull the data back to a normal distribution to avoid the problem of gradient disappearance
- After normalization, it can effectively avoid the problem of excessive parameter changes caused by different data distributions

$$y_i^{(b)} = BN_{(x_i)}^{(b)} = \gamma(\frac{x_i^{(b)} - \mu(x_i)}{\sqrt{\sigma(x_i)^2 + \varepsilon}}) + \beta \tag{1}$$

In this equation,x_i represent the vector$[x_i^1, x_i^3, x_i^3, x_i^m]$,$m$ represent the batch size,add ε to avoid the case of dividing zero, μ represent the average of this vector and σ means standard deviation. use γ, β to control average and standard deviation of y_i.

SVM with SIFT

SVM is mainly used for binary classification, for multi-classification, SVM generally uses the ovo or ovr method, and ovo is one-to-one. In the k-class data set, SVM is designed separately for each of the two types of samples for classification. This method must design k(k-1)/2 classifiers, and finally select by voting.

In the feature extraction part of the image, we chose the scale invariant feature transform method (SIFT). It is to find key points in different scale spaces and calculate the direction of key points. The key points found by SIFT are some very prominent points that will not change due to factors such as lighting, affine transformation, and noise, such as corner points, edge points, bright spots in dark areas, and dark points in bright areas. The SIFT feature remains invariant to rotation, scale scaling, brightness changes, etc., and is a very stable local feature. Because our images are handwritten, and many images are generated with different brightness and shooting angles, it is very suitable to use the SIFT method for feature extraction on our own data set.

It's hard to explain the whole theory behind SIFT here, so we only describe the main idea of SIFT: $D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) + I(x, y)$,$D(\hat{x}) = D + \frac{1}{2}\frac{\partial D^T}{\partial x}\hat{x}$. G represnt Gaussian blur, I

represent data distribution on image, we according to the value of $D(\hat{x})$ to choose feature point.

Categorical-cross

We implement a muti-classification task, so we choose to use Categorical-cross as our loss function.

$$L = \frac{1}{N}\sum_{i} L_i = -\frac{1}{N}\sum_{i}\sum_{c=1}^{M} y_{ic} log(p_{ic}) \tag{2}$$

In this equation,M represent the number of Classes,y_{ic}represent the vector like [0,0,0,1] 1 represent the true label, p_{ic} represent the probability of i belong to c.

4 Experiments

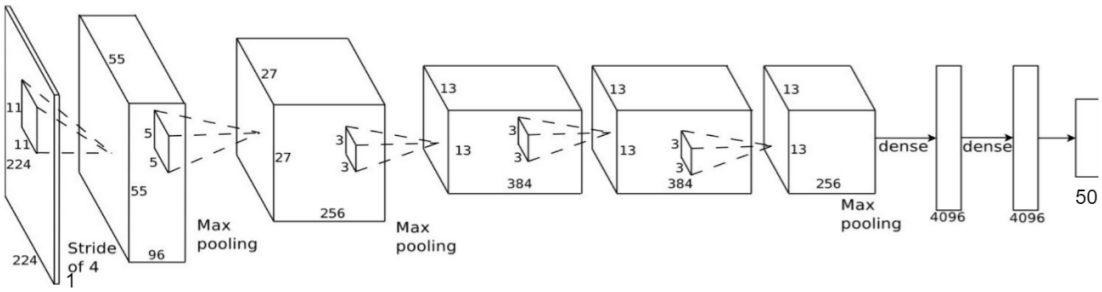
Dataset: Total 2107 images are used in this program. We choose 1600 images as trainset and 507 as test dataset. In the process of every epoch, we choose 20% trainset as validation set, which consisit of 320 images.

Evaluation:primary metrics are Accuracy ,Confusion Matrix and AUC. And because it is a 50 classification task, it's hard to present all the indicator for every class, so we will use macro average method to compare the performance.

4.1 Alexnet

Change structure of Alexnet: Because our task is to identify character in 50 classification, color infomation is useless so we choose to use image in grayscale to decease computing resource consuming. So when we use Alexnet structure, we need to change the $inputshape$ to one channel and final Dense layer to 50 classification.

Choice of hyperparameters:In this task we use classic Alexnet model, so we don't change the parameters like strides, kernel size, num of channels, num of layers, activation function etc. Because this is a deep learning task so it's very time consuming to tuning hyperparameters using grid search, so we only change the value of L1 to handle over-fit problem.



As Fig1 shown, the validation accuracy and loss of Alexnet($L1 = 0$) model is oscillating up and down, represent over-fit, this is because of the small scale of dataset, total 2100 images.

To deal with this, we have two strategies:

- ***Save best model:***Using $tf.keras.callbacks.ModelCheckpoint$ we monitor the validation accuracy to get the best model, in which $Accuracy = 86\%$, AUC and confusion matrix like Fig1,Figure 3 shows. With macro AUC=0.99, is a good model to implement task
- ***Use L1 regularizer:***The default Alexnet don't have regularizer, because batch norm layer and dropout between FC layer could handle the problem of over-fitting, but in this case, we need add L1 to fix it caused by too small dataset. And the result as Fig2 shows, by implementing regularizer L1=0.001, the over-fit problem get better, as we can see that loss of validation set keep the pace with train set and confusion matrix looks better too, no bright spots except for the diagonal

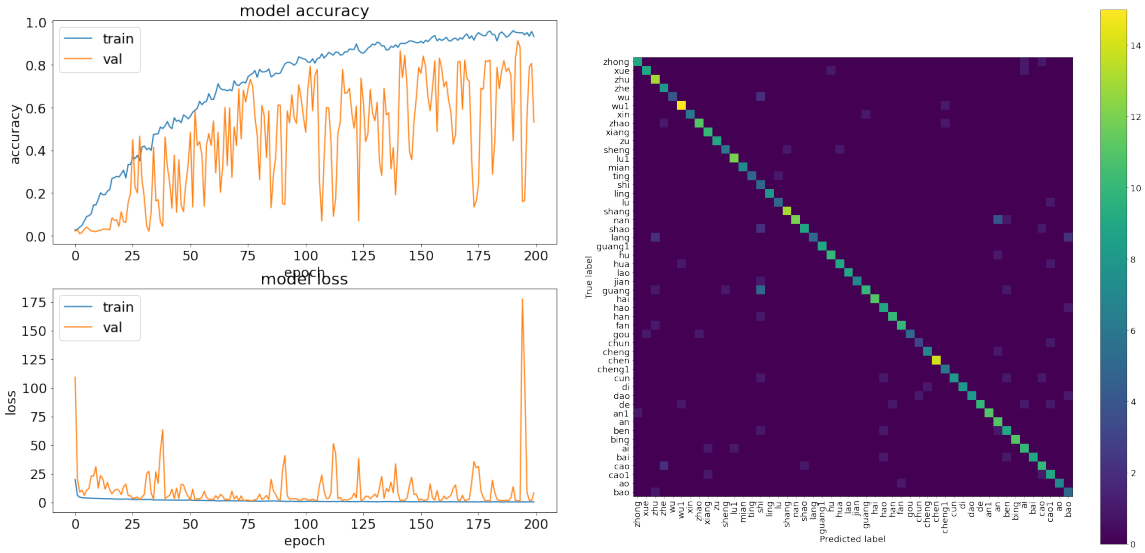


Figure 1: validation accuracy and confusion matrix of Alexnet L1=0

4.2 SVM

Gridsearch in SVM:

We use the grid search method for parameter adjustment, which is to exhaustively enumerate the parameter list to find the optimal parameter combination. And in this experiments, the best parameter set is C=0.1,gamma=1,kernel=linear.

Code explaination:

When reading the picture data, because the resolution of the pictures is inconsistent, we resize all the pictures to a resolution of 100*100, and then take the grayscale image for SIFT feature extraction. In the code, we write these steps into the feature extraction function, and the parameter of the function is the image path data. We used the pre-prepared labeled image paths as input to perform feature extraction on all image data.

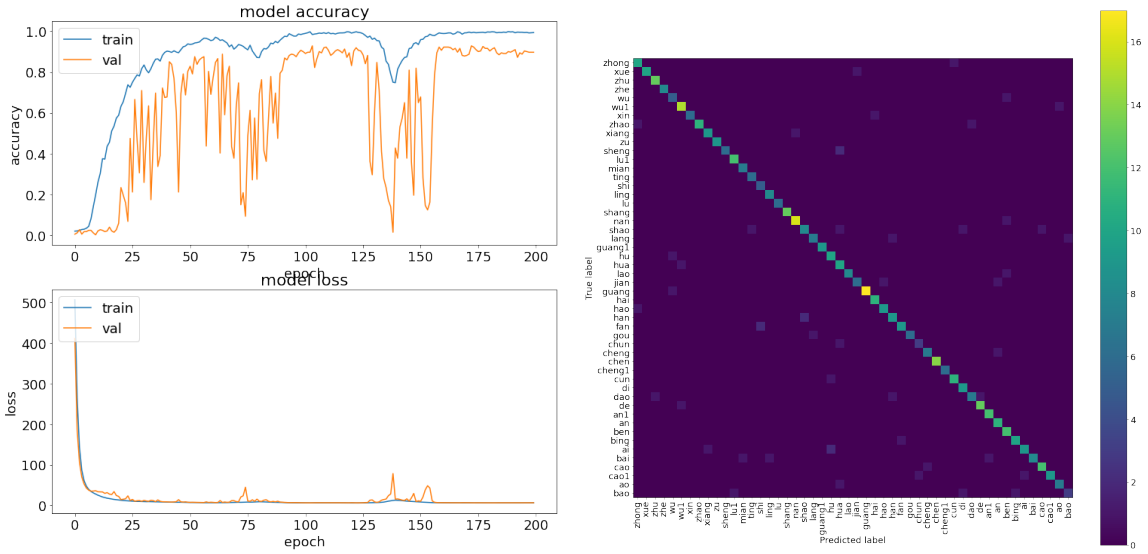


Figure 2: validation accuracy and confusion matrix of Alexnet L1=0.001

Because the feature dimensions extracted on each image using SIFT are not necessarily the same, we pad the image to keep all feature dimensions to the maximum dimension, and then use PCA to reduce the feature dimension.

Then we put the feature and label as x and y into the SVM model for parameter adjustment. After adjusting the key parameters of kernel, C and gamma, the accuracy of our model can reach 0.504739336492891.

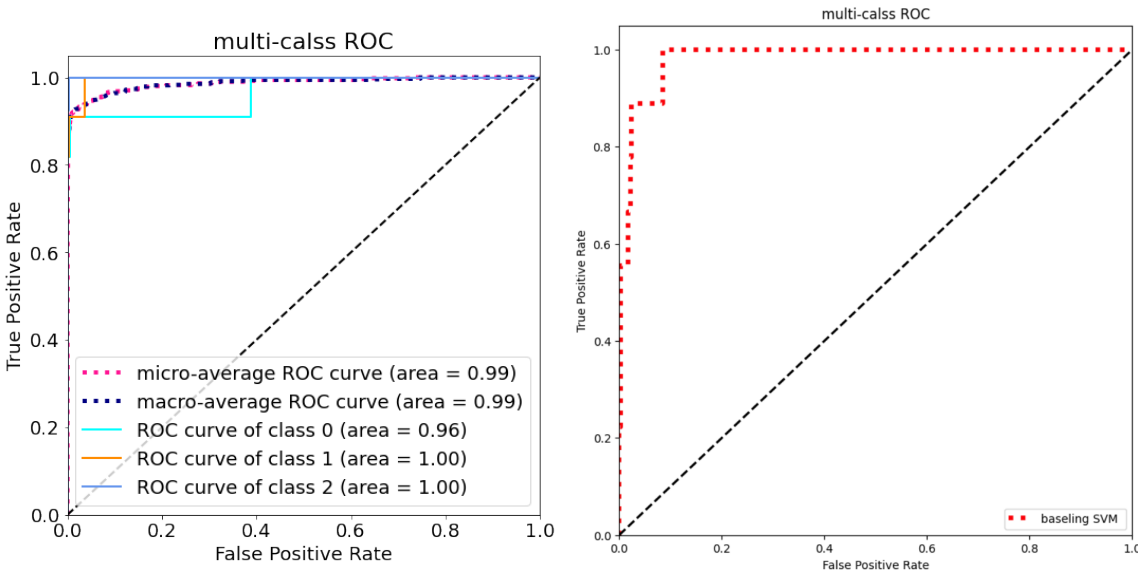


Figure 3: AUC of Alexnet and SVM

Table 1: Alexnet and SVM model performance table

model	*hyperparameters*	*accuracy*	*macro precision*	*macro recall*	*macro F1*
Alexnet	L1=0	86%	0.89	0.87	0.86
Alexnet	L1=0.001	90%	0.90	0.90	0.89
SVM	C=0.1,gamma=1,kernel=linear	50%	0.54	0.52	0.50

5 Summary

By comparison, we found that the performance of Alexnet is significantly better than SVM, according to $Accuracy$ Alexnet with L1(90%)>Alexnet(86%)>SVM(50%)

6 Contributions

github link :

https://github.com/KIERANLJY/ML_Group_Project.git

Dataset: Derive raw data from Wiktionary – Shuo Jia

Create handwritten images – Jiyuan Liu, Zhipeng Fu, Shuo Jia

Implement data augmentation – Jiyuan Liu, Zhipeng Fu

Organize images in dataset – Jiyuan

Code: Read data and data preprocess – Jiyuan Liu

SVM and SIFT – Shuo Jia

AlexNet – Zhipeng Fu

Report: Introduction, Dataset and Features – Jiyuan Liu

Methods, Experiments/Results/Discussions – Zhipeng Fu, Shuo Jia

Summary, Contributions – Jiyuan Liu, Zhipeng Fu, Shuo Jia