Final Project Report

Comparison of SVM MLP and LR classifier

BOYU WANG

4/2/2022

1. Abstract

This project is mainly to compare the accuracy of SVM MLP and LR Classifier in traffic sign classification. The effect of initial image processing on classification accuracy is also studied. I found that sharpening the image significantly improved the accuracy of image classification. Based on the results of various situations and cross-validation, I found that SVM was the classifier with the best classification effect in this project. This result is beneficial to the research and development of autonomous driving.

2. Introduction

Now autonomous driving is a very hot topic, and the identification of traffic signs is very important for autonomous driving. Therefore, I want to study the accuracy of the three classifiers of SVM MLP LR classifier for traffic sign classification and preliminarily discuss how to improve the accuracy of classifier by data processing, and finally select the optimal classifier for traffic sign recognition.

2.2 Logistic regression classifier

Logistic Regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome.

2.3 Support Vector Machine

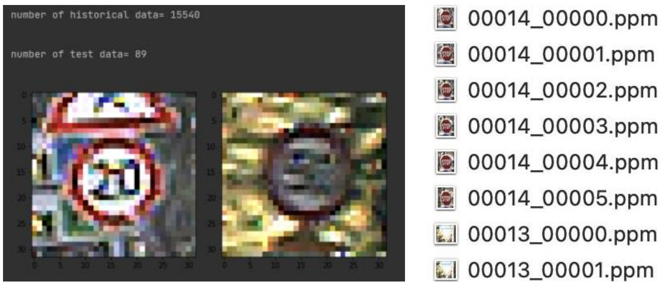
Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.

2.4 Multi-layer Perceptron

Multi-layer Perceptron (MLP) is a supervised learning algorithm that learns a function f(·):Rm→Ro by training on a dataset, where m is the number of dimensions for input and o is the number of dimensions for output.

3. Method

3.1 Data selection



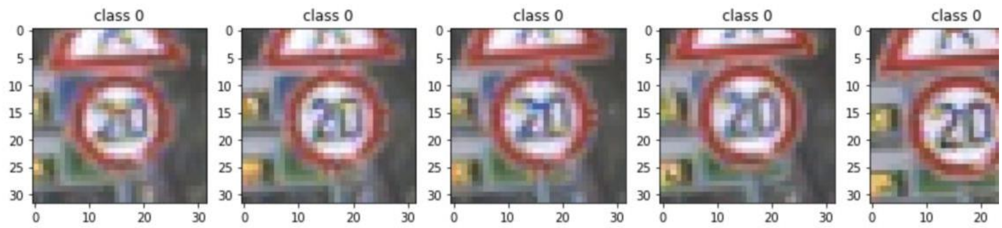
I use 15540 historical data for the training set and 89 images for the test set. About my data I use PPM format. The PPM format is a lowest common denominator color image file format. These images are divided into 15 classes.

From class0 to class14 Here I just show you 6 classes of images.
We just need to classify the training data at the beginning so that the data of the same class are in the same folder and have the same label.
When designing the input for model, the original data and their labels are corresponded well, and then when training the classifier, the corresponded data and labels are given to the classifier, and the classifier will automatically process multiple classes, and after the training is completed, the classifier will also classify the same number of classes for the new test data.

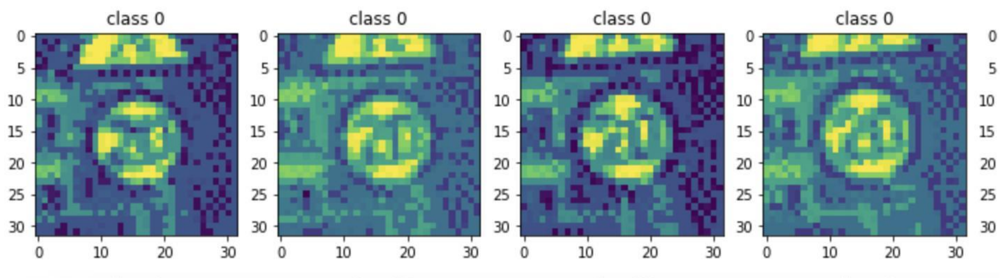
3.2 Data preprocessing

At the beginning, I tried to use the convert function to convert the images into more distinct patterns, such as binary maps and gray maps, but I found that this was not conducive to training the classifier, and the classification results were not even as good as the original images, as shown in the figure below.

```
0.9784205101777892
0.875
```

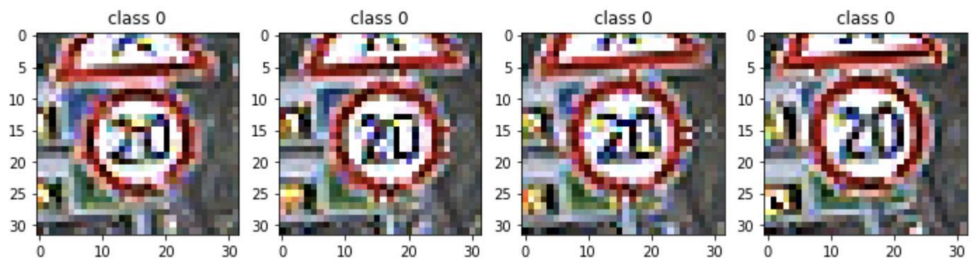



```
0.9493043030146869
0.75
```



Later I found a very efficient way to sharpen the image by importing the opencv function package as shown in the figure below.

```
0.9928497809842824
0.9375
```



3.3 Classifier’s parameter set

a. LogisticRegression (C=1, solver="sag")

The first classifier I used was a logistic regression classifier. The regularization strength is the default of 1. And because our datasets contains a large amount of historical data, the solver I use is ‘sag’, because ‘sag’ are faster for large datasets.

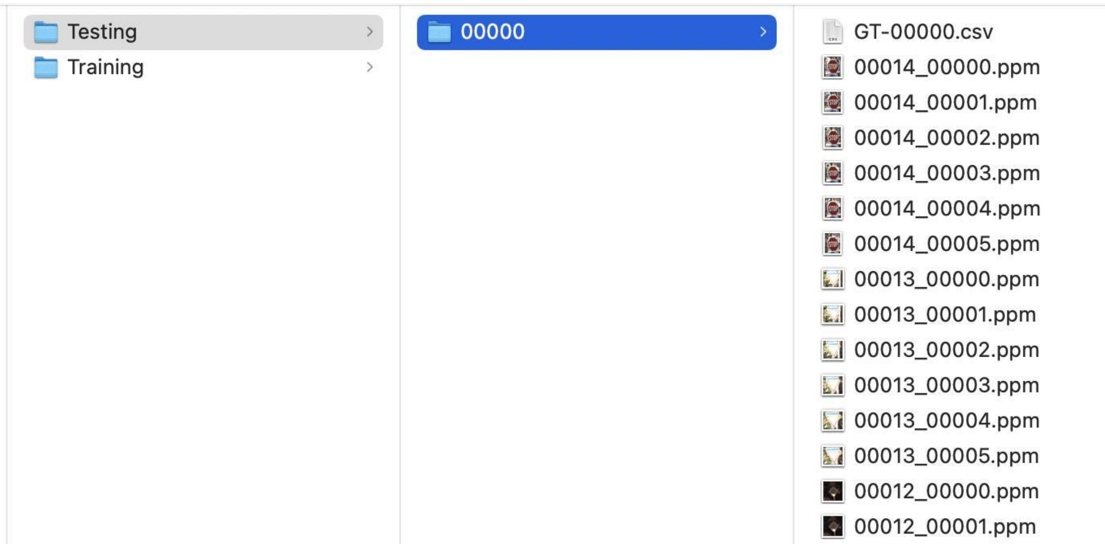
b. **lin_clf = svm.LinearSVC (C=1)**
The second classifier I use is the linear SVM, which is the one that I think gives the best results. parameter is the default.

c. **mlp = MLPClassifier (hidden_layer_sizes=(50,), max_iter=10, alpha=1e-4, solver='sgd', verbose=10, random_state=1, learning_rate_init=.1)**
The third classifier is MLP. In my comparison experiments, the MLP of 1 hidden layer with 50 neurons gave the best results.

3.4 Cross-validation

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.001, random_state=0)
```

When training the classifier, I use the ‘train_test_split’ function to divide the data into a training set and a validation set, and the test data will be selected from 0.1% of the training set for initial testing.



In addition, I created a folder named testing, which contains a total of more than 80 images of traffic signs. In the final test classification stage, I will read the images in the folder, then use the classifier to classify them, and finally print out the classification results and the images. This will help us to better see if the results are accurate, rather than just looking at a probability.

```python
# cross validation
from sklearn.model_selection import cross_val_score
# Compute three score vectores
scores1 = cross_val_score(logreg, X_train, Y_train, cv=5)
scores2 = cross_val_score(lin_clf, X_train, Y_train, cv=5)
scores3 = cross_val_score(mlp, X_train, Y_train, cv=5)

# Compute two mean scores and print out
print("Logistic Accuracy: %0.2f (+/- %0.2f)" % (scores1.mean(), scores1.std() * 2))
print("SVM Accuracy: %0.2f (+/- %0.2f)" % (scores2.mean(), scores2.std() * 2))
print("MLP Accuracy: %0.2f (+/- %0.2f)" % (scores3.mean(), scores3.std() * 2))
```

In the cross-validation stage, the cross-validation results of the three classifiers are printed out using the cross_val_score function. Cross-validation is a statistical method used to estimate the skill of machine learning models. I use it to verify that my results are accurate.

4. Result

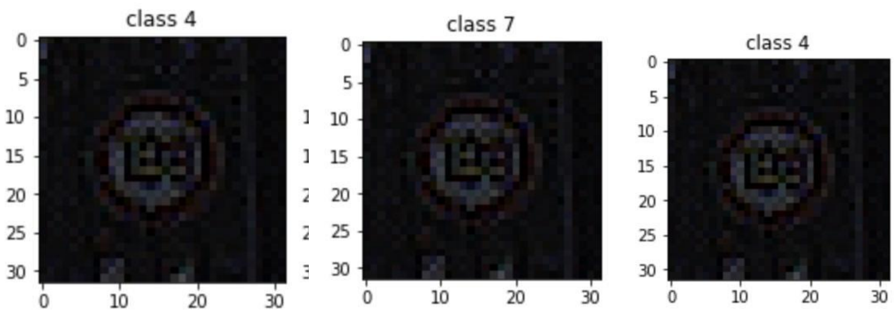
4.1 Performance of the three classifiers

```
Logistic result:  0.9929786137593404
Logistic testsets result:  0.9375

SVM result:  0.9996135016748261
SVM testsets result:  0.875

MLP Training set score: 0.989
MLP Test set score: 0.938
```

As seen from the above figure, SVM is the best performing classifier, followed by logistic regression and finally MLP.



From left to right, the recognition results of logistic regression, SVM, and MLP. Both logistic regression and MLP classify it as class4, and only the result of SVM is correct, and this image belongs to class7. In addition, for the more than 80 images used for testing, the accuracy of SVM is 0.9887, while the accuracy of the other two is not as good as that of SVM.

```
          Logistic final Classification Accuracy:

Out[77]:  0.9775280898876404
```

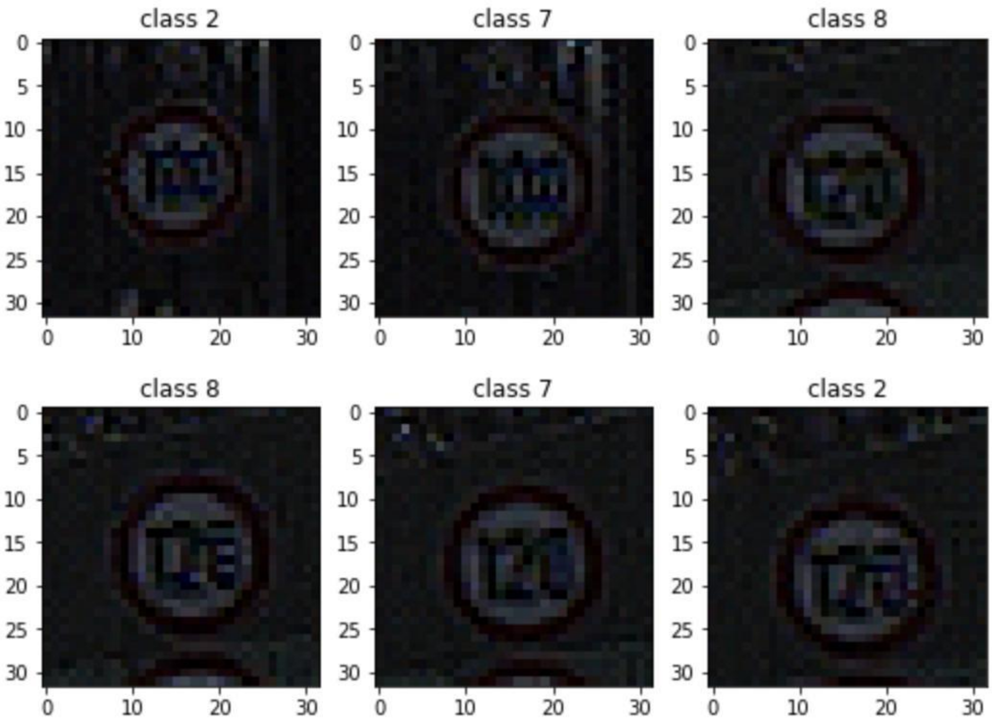
```
          SVM final Classification Accuracy:

Out[81]:  0.9887640449438202
```

```
          MLP final Classification Accuracy:

Out[79]:  0.9550561797752809
```

4.2 Images that are hard to be classified



label7 and label8 are two classes that are difficult to distinguish because there are many speed-limited labels, and they are two of the more similar classes. When they have low brightness and low clarity, it is difficult for the classifier to classify them, and logistic regression and MLP always make errors in classifying these two labels.

5. Discussion

```
Logistic Accuracy: 0.94 (+/- 0.01)
SVM Accuracy: 0.92 (+/- 0.01)
MLP Accuracy: 0.71 (+/- 0.33)
```

The above is the cross-validation result. Although the cross-validation result is that logistic
regression performs the best, SVM always produces stable and accurate test results during my testing, and although their results are relatively similar, I still think SVM is the best performing classifier.

It is interesting to note that without sharpening the images, SVM still performs well, already with about 98% correct, while logistic regression and MLP give much worse accuracy than SVM. After sharpening, the differences in the images are further amplified and logistic regression also has a very good correct rate (99.2%), only a little worse than SVM (99.8%). And the accuracy of MLP is also close to 99%. So I think the performance of logistic regression can be well represented for data with greater differences, but SVM performs well in both cases, which is one of the reasons why I choose SVM. However, MLP performs very average in either case, and after the image is processed differently, we need to do another processing on the input X value to make the loss lower, which is a very inconvenient point for MLP in my opinion.

6. Conclusion

Combining multiple situations and results, I think SVM is the best classifier in this project. From the project, I found that image brightness would have an impact on the accuracy of classifier and sharpened images would be easier to identify. I hope the results of my project can be of little help to the research in the autonomous driving field.

7. Reference

Medium. n.d. *The Perfect Recipe for Classification Using Logistic Regression*. [online] Available at: <https://towardsdatascience.com/the-perfect-recipe-for-classification-using-logistic-regression-f8648e267592> [Accessed 5 February 2022].

scikit-learn. n.d. *1.17. Neural network models (supervised)*. [online] Available at: <https://scikit-learn.org/stable/modules/neural_networks_supervised.html> [Accessed 5 February 2022].

scikit-learn. n.d. *1.4. Support Vector Machines*. [online] Available at: <https://scikit-learn.org/stable/modules/svm.html> [Accessed 5 February 2022].

Docs.opencv.org. n.d. *OpenCV: OpenCV-Python Tutorials*. [online] Available at: <https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html> [Accessed 5 February 2022].

Analytics Vidhya. n.d. *Sharpening An Image using OpenCV Library in Python*. [online] Available at: <https://www.analyticsvidhya.com/blog/2021/08/sharpening-an-image-using-opencv-library-in-python/> [Accessed 5 February 2022].

Brownlee, J., 2018. *A Gentle Introduction to k-fold Cross-Validation*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/k-fold-cross-validation/> [Accessed 5 February 2022].