Handwritten character recognition

Report of Assignment 6

SOURCE CODE: https://github.com/azhaubassar/hdw-recognition

Aruzhan Zhaubassarova
Suleyman Demirel University Kaskelen, Almaty
aruzhan.zhaubassar@gmail.com

Darkhan Suleimenov
Suleyman Demirel University Kaskelen, Almaty
darkhan.amangeldievic@gmail.com

ABSTRACT

Handwritten character recognition is one of the important issues in pattern recognition applications. In this paper, we give a detailed description of how we have accomplished with our task - Classify handwritten characters (kazakh)by using certain resources: data sets, method, features, etc. Through the whole paper, we answered questions about the project itself, the method we used, answers we obtained and analyses which we come up with. The aim of this work is to review existing methods for the handwritten kazakh character recognition problem using machine learning and deep learning algorithms and implement one of them for Android/iOS application.

KEYWORDS

Character recognition, handwritten recognition, machine learning algorithm, classification algorithm, neural networks

1. DATA/FEATURE:

1.1 *How the data is prepared for the competition?*

We collected data consisting of 412 samples(png) of characters for training and testing our model.

1.2 *What is the format for the data?*

Feature - independent input variable or variable constructed from the input variables. At the end, our code provides with the sample that predict our handwritten character on our application on iOS

2. LEARNING METHOD

2.1 *What is the goal of learning? Is the goal regression or classification?*

Our goal of learning is to predict handwritten kazakh character. The goal is more like classification.

2.2 *Methods?*

a)Support Vector Machine.

Support-vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. [1]

SVM advantages:

- SVM's are very good when we have no idea on the data.
- Works well with even unstructured and data like text, Images.
- SVM models have generalization in practice, the risk of overfitting is less in SVM.

SVM disadvantages:

- Long training time for datasets.
- Comparing with CNN model SVM give us less accuracy

d)CNN

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery. CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.[2]

CNN advantages[4]:

- Neural networks work best with more data.
- With any number of inputs and layers, CNN can train.
- Once trained, the predictions are pretty fast.

3. APPLICATION IMPLEMENTATION

3.1 *React-Native Application implementation*

The main piece of user interface it contains is a custom view object that facilitates the drawing of characters. A view is any single piece of user interface that can be shown to the user. It also contains two buttons: “Reset” and “Predict”. The clear button simply clears the drawing, and the prediction is shown to the user in input below.

3.2 Connecting to back-end and send data by POST request(image png format)

```
if (result.pathName) {
  // Create the form data object
  var data = new FormData();
  data.append('image', {
    uri: result.pathName,
    name: 'selfie.jpg',
    type: 'image/png'
  });
  // Create the config object for the POST
  const config = {
    method: 'POST',
    headers: {
      Accept: 'application/json',
      'Content-Type': 'multipart/form-data;',
    },
    body: data
  };
     fetch('https://hdw-recognition.herokuapp.com/predict',
config).then(responseData => {
    // Log the response form the server
    console.log(responseData);
    AsyncStorage.setItem('response', responseData)
  }).catch(err => { console.log(err); });
}
```

3.3 Visualize response

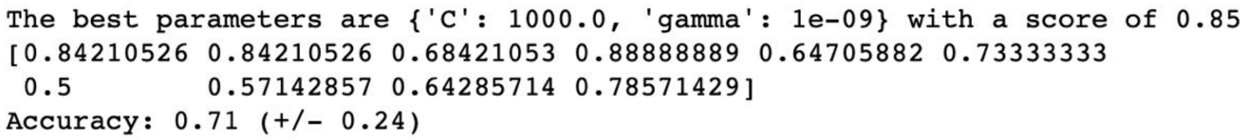
```
fetch('https://hdw-recognition.herokuapp.com/predict',
config).then(responseData => {
  // Log the response form the server
  console.log(responseData);
  AsyncStorage.setItem('response', responseData)
})
AsyncStorage.getItem('response').then((char)=>{
 this.setState({ char:char })
});
this.setState({data: base64String});
```

4. EXPERIMENTAL RESULT AND DISCUSSION

Name of algorithm	Correctly Classified Instances % (value)	Incorrectly Classified Instances % (value)	Time Taken (min)
SVM	71 %	29 %	2
CNN	98 %	2 %	40

SVM:

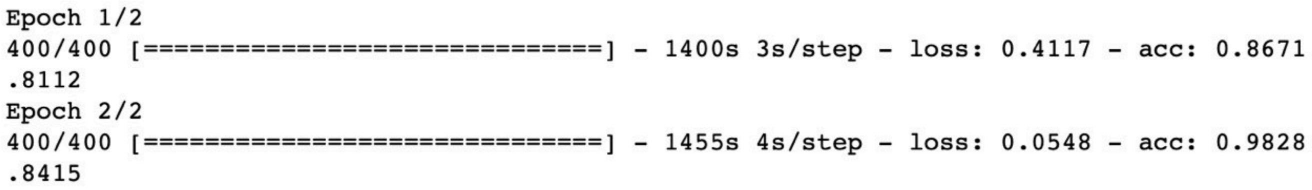
Here our accuracy of SVM implementation:



```
The best parameters are {'C': 1000.0, 'gamma': 1e-09} with a score of 0.85
[0.84210526 0.84210526 0.68421053 0.88888889 0.64705882 0.73333333
 0.5        0.57142857 0.64285714 0.78571429]
Accuracy: 0.71 (+/- 0.24)
```

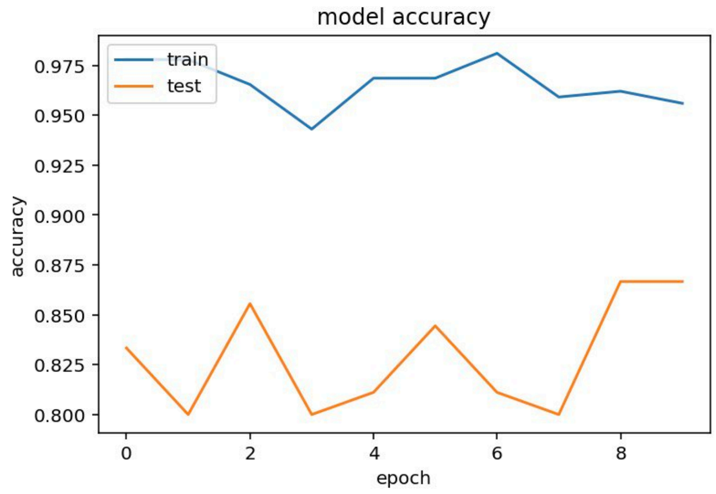
CNN:

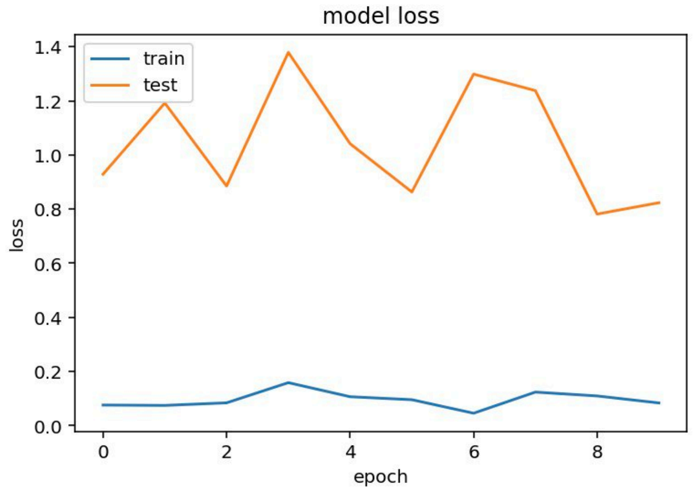
Here our accuracy of CNN implementation:



```
Epoch 1/2
400/400 [==============================] - 1400s 3s/step - loss: 0.4117 - acc: 0.8671
.8112
Epoch 2/2
400/400 [==============================] - 1455s 4s/step - loss: 0.0548 - acc: 0.9828
.8415
```

We have considered model accuracy and model loss tables and made a comparison between training and testing data.





CONSLUSION

The main objective of this investigation is to find a representation of isolated handwritten digits/ kazakh characters that allow their effective recognition. In this paper used two different machine learning algorithm for recognition. In any recognition process, the important problem is to address the feature extraction and correct classification approaches. The overall highest accuracy 98.28% is achieved in the recognition process by CNN.

FUTURE WORK

In future we would like to add more training data and make our model more wider and better. Also we are planning to make an implementation of our application by using other algorithms and methods, such as Forest Tree, KNN, XGBoost in order to get higher accuracy.

Current application's source code and data is available by the link below:

ML application https://github.com/azhaubassar/hdw-recognition

React-native application https://github.com/azhaubassar/HandwrittenRecognitionApp

REFERENCES

[1] https://en.wikipedia.org/wiki/Support-vector_machine
[2] https://en.wikipedia.org/wiki/Convolutional_neural_network
[3] http://yann.lecun.com/exdb/mnist/
[4] https://data-flair.training/blogs/convolutional-neural-networks-tutorial/