AI Chess Master

1. Introduction

The problem of Chess Position comes under the multiclass image classification problem. Since the dataset is very large, and deals with images, one efficient and most used method would be using CNN. By using open-cv, such image classification problems become less complicated and interesting. However, SVM is also a good choice if we extract important features using principle component analysis. And the model is also efficient to deal with multiclass classification problems by using kernels. To solve this problem, I chose the approaches of CNN as well as SVM.

2. Methodology

2.1 Dataset Description

The dataset Chess Position contains 100000 images of randomly generated chess positions. The images are generated using 28 styles of chess boards and 32 styles of chess pieces. All the images are 400X400 pixels. Train set contains 80000 images and Test set contains 20000 images. The labels are given in FEN notation. FEN is a standard notation of describing a chess board positions. Generally a chess board contains 32 items. There are 13 unique labels including empty space to occupy.

2.2 Data pre-processing for models

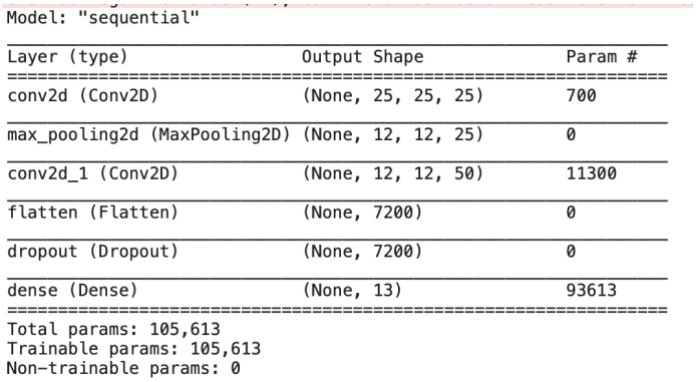
Steps for preparing dataset

1. Split the whole chess board into 64 individual cells of 50X50 pixels. And rescale the image into 25X25 pixels
   split_chess_board(board) method is used for this purpose.
2. To extract labels of each 25X25 image, decode FEN and separate labels.
   split_labels(y) method is used for this purpose
3. Create two arrays with images in one array and labels in another.
4. Labels are encoded using *LabelEncoder* from *sklearn.*

2.3 Tested Models

2.3.1 CNN(Convolutional Neural Networks)

Since the dataset is huge, a subset of 3000 train images and 1000 test images are chosen.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 25, 25, 25)        700
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 12, 12, 25)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 12, 12, 50)        11300
_________________________________________________________________
flatten (Flatten)            (None, 7200)              0
_________________________________________________________________
dropout (Dropout)            (None, 7200)              0
_________________________________________________________________
dense (Dense)                (None, 13)                93613
=================================================================
Total params: 105,613
Trainable params: 105,613
Non-trainable params: 0
```

Figure 1. Architecture of CNN

The input shape of the image is (25,25,3) since I rescale the image into that 25X25, the value 3 indicates the colour channels. Since I did not convert the image into grayscale there are 3 channels for R,G and B. I decided to design a simple model because I am using only 3000 images to train.

- A non-linear activation function “relu” is used in convolution layers. which will reduce the computational complexity in neural network.
- A kernel initializer “he_uniform” is used to initialize kernels by drawing samples from a normal distribution.
- To avoid overfitting a dropoutlayer of 0.3 is used.
- Output Dense layer has 13 neurons. Which indicates the 13 unique labels to predict.
- Loss function “sparse_categorical_crossentropy” is used since there are 13 distinct mutually exclusive output classes. Generally it is used when there are more than 2 classes.
- Optimizer “adam” is used. It performs well with sparse data

2.3.2 SVM and PCA

SVM is a robust model for multiclass classification problems. For the better performance of SVM, it is necessary that the dataset is balanced. From the EDA, I found that dataset is not balanced because when we split the chess board into 64 sub images, more number of sub images are of empty spaces in chess board. It makes the dataset imbalanced.

- Dataset is balanced using resampling method. After resampling the train set contains only 2379 images. But for SVM it was enough for training.
- Normalization is done using *StandardScaler* from *sklearn.*
- PCA is applied before training with SVM in order to reduce the dimensionality and taking only important features.
- I chose *n_components=150* for selecting 150 important features.
- SVM with *rbf* kernel is used because with high dimensionality data rbf is a good choice.
- A model pipeline is created for automation of the prediction process.

3. Results

3.1 CNN

- To evaluate the model, matrix “Accuracy ” is used. With only 2 epochs, the model was shows a performance of training accuracy 99.9% and validation accuracy 100%.

- This is due to the fact that only limited size data is given to the model to train. Also the model is too simple to train.
- When tested with unseen images of board, it was giving correct labels.

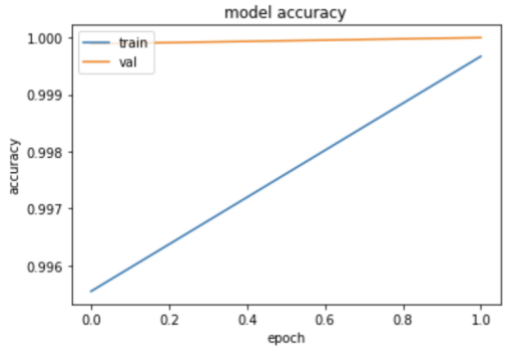


Figure 2. Performance of CNN

3.2 SVM and PCA

- Sklearn *accuracy_score* is used to compute accuracy of multiclass classification problem
- Training accuracy of 96.6 % is obtained and with test accuracy of 96.6 % is obtained.

4. Conclusion

The problem was quite interesting and require lot of attention to understand.

Both the choices of modelling was efficient when it comes to performance. With more computational power and resources, CNN will be a good choice. SVM is a good choice with more explainability. The choice depends solely on the size of dataset.