Image Classification: Simpsons Character Recognition

Yanan Gao (913556858), Yu Zhu (914480813)

1 Introduction

The Simpsons is a beloved American TV series, which has been broadcasted in more than 90 countries all over the world. It is easy to notice that almost all of the characters in Simpsons have yellow skin, because Simpsons creator Matt Groening made the characters yellow to grab the attention of channel surfers. He also designed the Simpson family to have distinctive hairstyles and head-shapes. Therefore, image classification of Simpsons characters can be both fun and intuitive. In this report, several analyzing models are implemented such as K-NN, Random Forest, SVM with RBF kernel and CNN.

2 Data Preparation

2.1. Data Cleaning

First of all, It can be seen from figure 1 that some images have only one character, while a few of them have more than one characters.

Figure 1. Examples of Each Image Data

Secondly, by examining the raw data, we found out that there are over 20,000 images in training set with 47 different Simpsons characters, while around 1000 images in testing set with over 20 characters, and each character has about 50 test data. We plot a histogram (figure 2) of the number of images for each character in training set, found out this number is very imbalance. Therefore, we decided to only use those characters with reasonably large images for the machine to learn. Similar work has been done to test set.

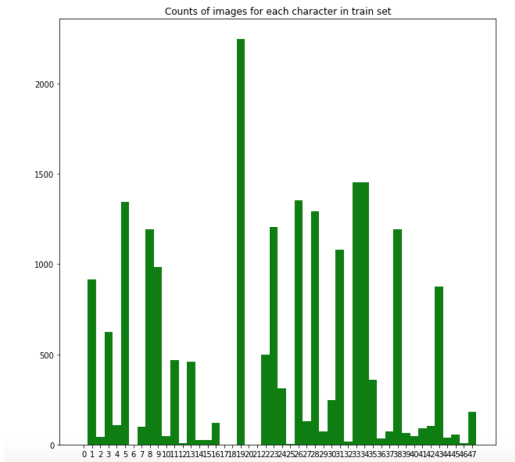

Figure 2. Counts of Images for Each Character in Train Set

After the cleaning of raw data, we now have 13 characters with the largest amount of training images in both train and test set. They are shown in table 1.

Table 1. 13 Characters Picked For Further Use

Label	Characters	Counts(train set)
0	abraham grampa simpson	913
1	bart simpson	1342
2	charles montgomery burns	1193
3	chief wiggum	986
4	homer simpson	2246
5	krusty the clown	1206
6	lisa simpson	1354
7	marge simpson	1291
8	milhouse van houten	1079
9	moe szyslak	1452
10	ned flanders	1454
11	principal skinner	1194
12	sideshow bob	877

2.2. Further Data Processing

The maximum number of images we use to train our model for each character is 1000 (for example, 1354 images have been found for Lisa Simpson, we only use the first

1000 of them), to ensure our training data is balanced. The following jobs are done to the data in this part :

•The images are loaded as size of 64*64.

•Normalize the data by dividing 255 (reduce the effect of illumination).

•Reshape the images into 3D (height = 64, width = 64, canal = 3).

•Save another set of 1D data (64*64*3) for k-NN, RF and SVM.

•Encode labels to hot vectors as response variable for CNN.

3 Model Implementation

3.1. K-Nearest Neighbors (k-NN)

We firstly implement k-NN model because it has several advantages. Clearly, it is simple to implement and requires short training time. However, the accuracy and testing time are not desirable in most of the cases. No matter how, we can still use this result as a base line. To find the best parameter of the number of neighbors (denoted as K), we perform 5-fold Cross Validation on each choice of K (from 1 to 20). We pick k = 19 according to CV score since from figure 3, we can see CV score is nearly converged. The accuracy is 47.58%.

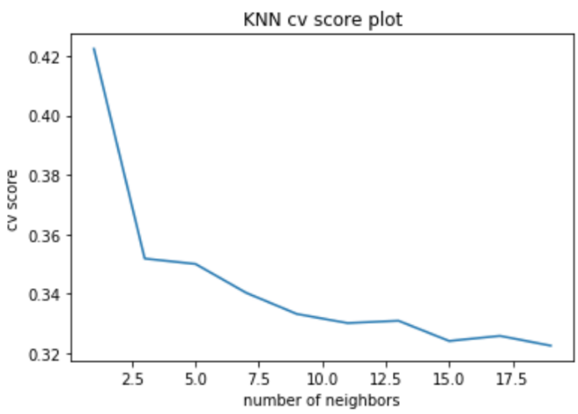

Figure 3. CV Score Plot for k-NN

3.2. Random Forest (RF)

Random forest is another method could be used in image classification. Therefore, we fit the Random Forest model using default values in "sklearn" package to see what the result will be. The accuracy is 88.9%.

3.3. Supported Vector Machine with RBF Kernel

For multi-class classifications with SVM, we need to find out the most suitable kernel to optimize, and it depends on dataset condition (Han and Michelin, 2006). From previous studies, SVM with RBF kernel function performed better and was recommended to be implemented.

We want to optimize the classification accuracy through tuning the parameters of the RBF kernel (C and γ) and minimize the over-fitting. Since default parameters performs poorly(accuracy is about 50%), we decide to use the grid search and cross validation to select the best combination of hyper-parameters. So the process of parameter evaluation is shown in figure 4.

Figure 4. Flow chart of SVM-RBF hyper-parameters selection

The beginning range we set for C and γ is: $C = \{0.1, 0.5, 1, 5, 10, 50\}$, $gamma = \{10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}, 0.1, 0.5, 1, 5\}$. So there are 6 selections for C and 8 selections for γ, the total number of combinations for parameters is 48. For each parameter combinations, we do 3-fold cross validation. Therefore, we have 144 fits for the SVM model. And the best combination of parameters we obtain is C = 5 and $\gamma = 0.05$. The accuracy for this RBF-SVM model is about 88.77%

3.4. Convolutional Neural Network (CNN)

3.4.1. Optimizer setting

We pick the "categorical crossentropy" as our loss function to define how bad our model behaves. The choosing of optimizer is also very important. Here, we use Stochastic Gradient Descent optimizer. Although SGD has been consider "outdated" because of it's slow convergence. However, SGD has better generalization than adaptive optimizers (Wilson, Ashia C., et al).

3.4.2. Number of layers tuning

For the first model (denoted as model1), with relative small training data size, we use 2 convolutional layers (with 32 filters for each, ReLU is used to introduce non-linearity), followed by one pooling layer (MaxPooling2D) and two fully connected lays. In this model, we have also set the parameter of dropping out for a relative large number 0.5 to lower the risk of over-fitting. We have achieved an accuracy of over 95%. (sadly, we have lost the piece of code for this model)

In order to solve this problem and increase the accuracy, we searched CNN with deeper structures (denoted as model 2). The set up of model2 is shown in figure 5. The model accuracy has now increased to 97.66% for 40 epochs.

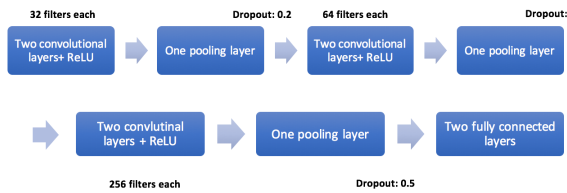

Figure 5. Flow Chart of CNN Model 2

3.4.3. Data augmentation

To further increase accuracy and avoid over-fitting, we introduce data augmentation into previous model 2, the new model is denoted as model 3. Our data is changed by randomly rotating or shifting images to reproduce different motions of Simpson characters. By doing augmentation (we also increase the epochs to 50), the accuracy now is 98.13%.

4 Model Evaluation

4.1. K-Nearest Neighbor Method

The result of 47.58% is not satisfying, as we have expected. However, compared with random guessing (with a probability of $\frac{1}{13}$ to get the right classification), it is already much better. This shall be used as the baseline of the accuracy of all the models.

4.2. Random Forest

We are surprised to see that the accuracy is relatively high using default values: 88.9%. Moreover, confusion matrix (figure 6) also tells us that there is no severe misclassification for any character.

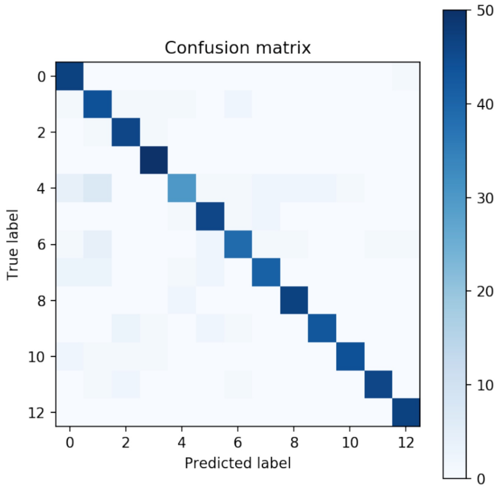

Figure 6. Confusion Matrix for Random Forest Model

4.3. Supported Vector Machine with RBF Kernel

It's good to see this RBF-SVM model achieves an accuracy of 88.77%. From the confusion matrix (figure 7), misclassification only happens when classifying homer_simpsons. It can be confused with almost all other characters. It might because the features of homer_simpsons is relative less obvious than others, or the image quality is worse.

For further research, we might shrink the range of parameter set and increase the precision of the grid search based on the current research in order to achieve better result.

4.4. Convolutional Neural Network

Normally speaking as the number of epochs increase, more information will be learned, thus resulting in higher accuracy. From figure 8 and figure 9, we can see that the accuracy and loss have already converged. Therefore 40 epochs for model 2 and 50 epochs for model 3 are enough.

Compare model 2 and model 3, although model 3 takes a longer time to converge, we get a very high accuracy in a reasonable time period.

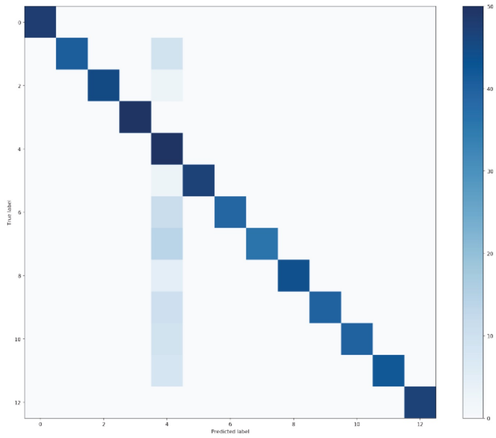

Figure 7. Confusion Matrix for SVM-RBF Model

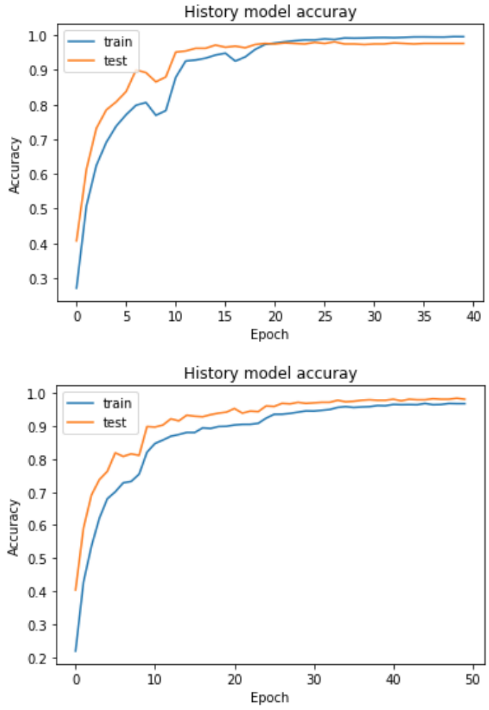

Figure 8. Accuracy Comparison between Model 2 and Model 3

4.5. Model Comparison

• Model accuracy: From table 2, we can see that CNN models achieve the highest accuracy, especially CNN model 3. SVM-RBF and Random forest also give a reasonably good result. However k-NN performs poorly, is not even comparable.

Table 2. Accuracy Results of Three Models

Model	Accuracy	Precision	Recall	F1
KNN	47.58%	0.52	0.48	0.46
Random Forest	88.9%	0.89	0.89	0.89
SVM-RBF	88.77%	0.95	0.89	0.91
CNN model2	97.66%	0.98	0.98	0.98
CNN model3	98.13%	0.98	0.98	0.98

• Computation time: CNN models offer reasonably good speed. The situation for SVM-RBF is not optimistic. In order to achieve an accuracy comparable to CNN, the grid

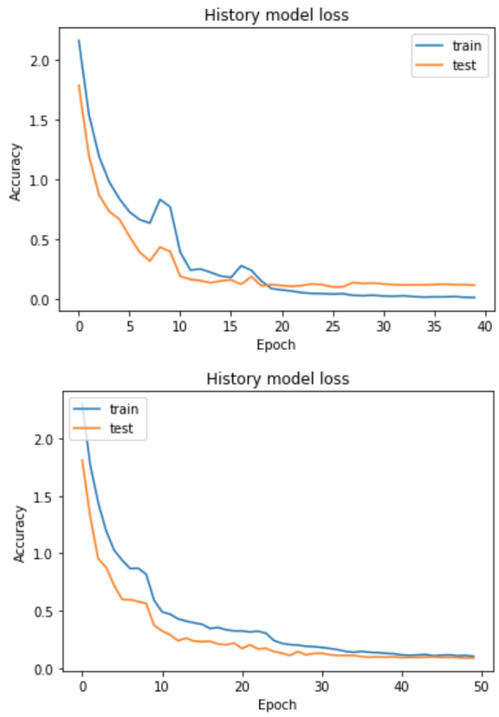

Figure 9. Loss Comparison between Model 2 and Model 3

search of parameter tuning has to be done which is highly computationally expensive (It takes several days to finish on CPU).

• Generality: In CNN model 3, we have intentionally changed data by distorting training images, created "weird"-looking images not existing in real life. However, the accuracy is still very high. This might indicate that CNN model has certain generality.

5 Conclusion and Future Work

From previous study, we have seen the superiority of Convolutional Neural Network towards image classification. However, further improvement can be done to this dataset.

From dataset point of view, more images can be collected, especially for those characters with very few images in training set. Moreover, the quality of images can also be improved.

On the other hand, we have noticed that some images contain more than one character. Our CNN model only recognizes one character from each image. Therefore, other method such as Recurrent Neural Network (RNN) should also be considered.

6 References

[1] Han, J. and K. Michelin, 2006. Data Mining: Concepts and Techniques. 2nd Edn., Elsevier Inc, San Fransisco.

[2] Wilson, Ashia C., et al. "The marginal value of adaptive gradient methods in machine learning." *Advances in Neural Information Processing Systems*,2017