Transfer Learning-based CNN Classification for Simpsons Characters

Yueheng Li
Department of Computer Science
Stanford University
yueheng@stanford.edu

Abstract

The Simpsons is the one of the most popular show in the history. It's important for the animation industry to classify certain characters. Convolutional neural network (CNN) shows potentials for variance visual applications. However, CNN can be difficult to train for simpsons character classification due to lack of images and low resolution of image quality. Instead, transfer learning can gain knowledge from a different but related problem. In this project, I classified 10 simpsons characters by using CNN with transfer learning. I explored different pre-trained models, optimizations and hyperparameters, and got the final results with pre-trained VGG16 model and Adam optimization. The model achieved 76% precision, 73% recall and 74% f1 score on average for test data.

1 Introduction

Since its debut on December 17, 1989, 653 episodes [1] of The Simpsons have been broadcast. It is the longest-running [2] American sitcom, and the longest-running American scripted primetime television series in terms of seasons and number of episodes [3]. The animation industry and a lot of fans around the world are interested in classify characters.

As one of deep neural network, Convolutional neural network (CNN) [4] is widely and commonly used for analyzing visual imagery. However training CNN can be difficult since it takes huge amount of data, time and experience to get ideal results. Transfer learning [5] is a technique where a model trained on one task is re-purposed on a second related task and can help to train a CNN with less data in less time.

In the project, I classified simpsons characters. The input of the algorithm was 10 simpsons characters (10 classes) images, and I used transfer learning with VGG16 to output the classification of the simpsons characters.

2 Related work

CNN with transfer learning has been explored in different applications. For instance, digital mammographic tumor classification using transfer learning from deep convolutional neural networks [6], using Deep Learning for Image-Based Plant Disease Detection [7], and etc.

The simpsons character classification has mostly been attempted by simpsons fans on Kaggle [8]. Methods used including training CNN from scratch [8], which didn't take advantage of transfer learning and the accuracy was relatively low; training using support vector machine, which has simple

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

implementation but seems to not work very well [9]. Other methods used were logistic regression, k neighbors classifier, decision tree classifier and etc.

3 Dataset and Features

The simpsons dataset used in this project consisted of 10 simpsons characters (10 classes). Each character had around 1000 to 2000 images. All the images were directly taken and labled from TV show episodes and downloaded from Kaggle [11]. Characters include: Bart Simpson, Charles Montgomery Burns, Homer Simpson, Krusty The Clown, Lisa Simpson, Marge Simpson, Milhouse Van Houten, Moe Szyslak, Ned Flanders, Principal Skinner.

The images for each character were distributed to train/dev/test data data with percentage 70/20/10. There were 10816 training images, 1999 dev images and 996 test images. Image resolution of 64x64 and 128x128 and 256x256 were explored. The final results presented in this report used 128x128 for accuracy and velocity balance.

Data were preprocessed: 1) Characters with less than 1000 images were eliminated from the original dataset and each character has 1000 to 2000 images. 2) Data augmentation was used for optimization, including flipping, cropping, rotating, and etc.

Figure 1. Sample image of Bart Simpson, Homer Simpson and Marge Simpson from left to right

4 Methods

There are several models pre-trained on imagenet [12] applicable to images classification. In this project, ResNet50 [13], VGG19 [14] and VGG16 were explored. I'll present the result of using VGG16 with transfer learning.

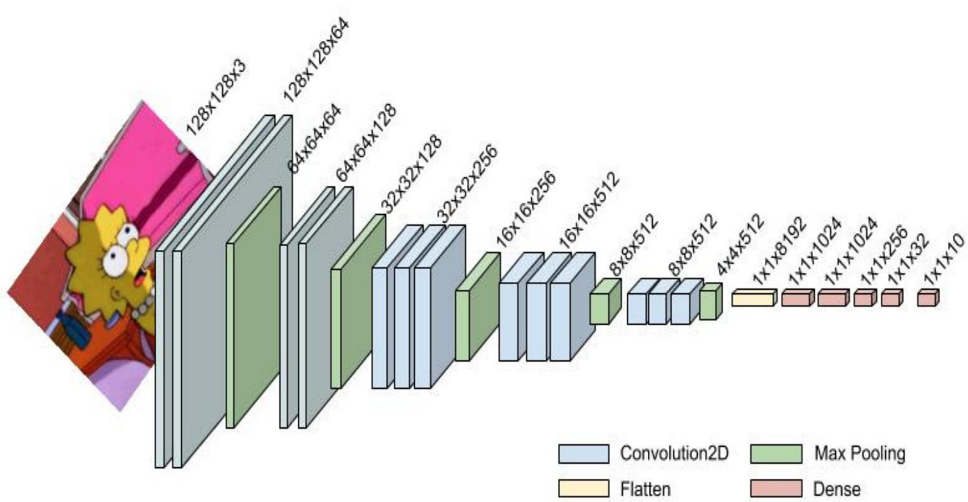

Figure 2. Model architecture

As shown above, the model used in the project consisted of pretrained VGG16 convolution and max pooling layers, and 5 fully connected layers of neural size 1024, 1024, 256, 32 and 10 were attached after the flatten layer of VGG16. The size of the each layer is shown in figure 2.

Cross entropy was used as the loss function.

$$L(y, \hat{y}) = -\sum_{i=1}^{\mathrm{N}_c} y_i \log(\hat{y}_i) \tag{1}$$

where $y \in \mathbb{R}^{10}$ is a one-hot label vector, and N_c is the number of classes and is 10 in this project.

Accuracy of dev dataset was used to evaluate the training as the evaluation metric.

5 Experiments/Results/Discussion

To determine the best model to used to classify simpsons characters, I iteratively improved the results by 1) investigating different architectures 2) tuning hyperparameters

I explored different pre-trained model, including ResNet50, VGG19 and VGG16, and found for the simpsons dataset, VGG16 produced the best accuracy.

Different hyperparameters were explored using random search [15]. The model was trained for 80 epochs, Adam optimization was used with learning rate 0.0001 and learning rate decay rate 0.000001, number of fully connected layers was 5, number of neurons in each fully connected layer were 1024, 1024, 256, 32, 10, batch size was 4, dropout rate of each fully connected layer was 0.5.

In the beginning, the model was suffering from low and first increasing then decreasing dev accuracy, I added more dropout, used data augmentation and higher resolution images, then both the train accuracy and dev accuracy converged to around 50%.

After adding more fully connected layers and more neurons in each layer, the model achieved 52% train accuracy, train loss was 1.5245 and 76.35% dev accuracy, dev loss was 0.8494. Dev accuracy was higher than train accuracy because dropout was used, and dropout randomly switches off particular neurons during training, but during dev phase it was disabled,

As shown below, for test data, the model achieved 76% precision, 73% recall and 74% f1 score on average. Marge simpson had the highest precision 100% and Homer simpson had the lowest presision 52%. The confusion matrix showed the model performs well in general for all 10 characters.

character	precision	recall	f1-score	support
bart simpson	0.87	0.78	0.82	100
charles montgomery burns	0.84	0.75	0.79	100
homer simpson	0.52	0.86	0.65	100
krusty the clown	0.82	0.77	0.79	97
lisa simpson	0.79	0.71	0.75	100
marge simpson	1.00	0.64	0.78	100
milhouse van houten	0.76	0.83	0.79	99
moe szyslak	0.67	0.68	0.68	100
ned flanders	0.63	0.68	0.65	100
principal skinner	0.69	0.62	0.65	100
avg / total	0.76	0.73	0.74	996

Table 1. Testing result of each character

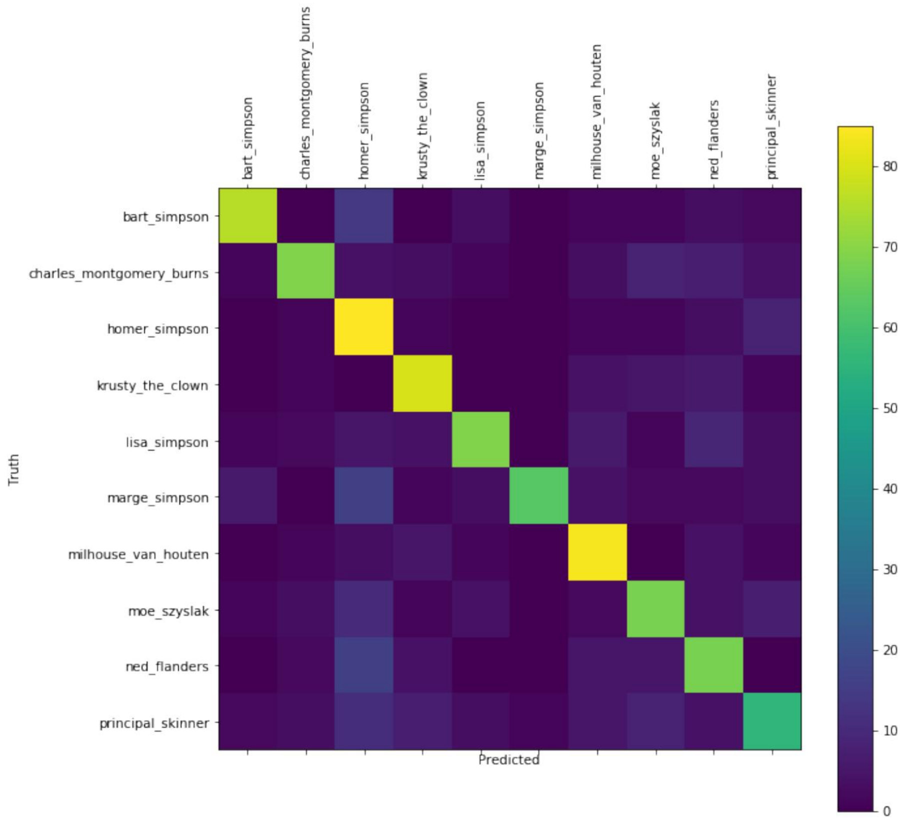

Figure 3. Confusion matrix of the testing result

6 Conclusion/Future Work

It's shown in this project that CNN with transfer learning can produce good classification results. Given that a relatively simpler pre trained model VGG16 actually achieved better results compared with ResNet50 and VGG19, there may be an opportunity to implement a new simpler architecture.

The train accuracy is lower than the dev accuracy as presented in the report. My next step would be trying different dropout rates, and increase the model complexity to increase the training accuracy.

In this project, all characters with less than 1000 images were eliminated. My next step would be reusing architecture used in this project or exploring other architecture to train more characters including characters with very few images.

7 Contributions

This project is a solo project.

8 Code

https://github.com/YuehengLi/CS230/blob/master/simpsons.ipynb

References

[1] List of The Simpsons episodes. https://en.wikipedia.org/wiki/List_of_The_Simpsons_episodes

[2] List of longest-running U.S. primetime television series. https://en.wikipedia.org/wiki/List_of_longest-running_U.S._primetime_television_series

[3] The Simpsons. https://en.wikipedia.org/wiki/The_Simpsons

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. (2017) ImageNet classification with deep convolutional neural networks. *Commun. ACM 60, 6 (May 2017), 84-90.* DOI: https://doi.org/10.1145/3065386

[5] Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson. (2014) How transferable are features in deep neural networks? *CoRR abs/1411.1792*

[6] Huynh, B. Q., Li, H., Giger, M. L. (2016). Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. *Journal of medical imaging (Bellingham, Wash.), 3(3), 034501.*

[7] Mohanty SP, Hughes DP and Salathé M (2016) Using Deep Learning for Image-Based Plant Disease Detection. *Front. Plant Sci. 7:1419.* doi: 10.3389/fpls.2016.01419

[8] Simpson Characters Classification. https://www.kaggle.com/cla0709/simpson-characters-classification

[9] Simpsons ID challange. https://www.kaggle.com/kaggletoby/simpsons-id-challange

[10] Simpsons character classification. https://www.kaggle.com/alexattia/the-simpsons-characters-dataset/kernels

[11] The Simpsons Characters Data. https://www.kaggle.com/alexattia/the-simpsons-characters-dataset

[12] ImageNet. http://image-net.org/

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. (2015) Deep Residual Learning for Image Recognition. *CoRR abs/1512.03385*

[14] Karen Simonyan, Andrew Zisserman. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR abs/1409.1556*

[15] James Bergstra and Yoshua Bengio. (2012) Random search for hyper-parameter optimization. *J. Mach. Learn. Res. 13 (February 2012), 281-305*