Image Classification on Webtoon Thumbnails Using CNN/RNN Architecture and Image Search Using Locality Sensitive Hashing

Jae Park (JP)
jae.park@uga.edu
University of Georgia
Athens, Georgia, USA

ABSTRACT

We created CNN and RNN models to classify Webtoon thumbnails into the respective artist and genre. We then used image embedding and locality sensitive hashing to simulate similar image search. The classifier for the artist was not able to generalize beyond the training set; however, the genre classifier and the similar image search showed promising result.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**.

KEYWORDS

neural networks, RNN, CNN, image embedding, nearest neighbor, locality sensitive hashing, image classification

ACM Reference Format:
Jae Park (JP). 2022. Image Classification on Webtoon Thumbnails Using CNN/RNN Architecture and Image Search Using Locality Sensitive Hashing. In *CSCI 6380 Data Mining Final Project, May 09, 2022, Athens, GA.* ACM, New York, NY, USA, 4 pages.

1 INTRODUCTION

Webtoons are digital comics that originated from South Korea, and unlike manga (comics originating from Japan), Webtoons are displayed in a vertical strip format with color, which make them convenient for readers to view on their smartphones and computers. They are also typically available to the readers for free on various platforms. Thanks to their convenience and ease of access, drawing Webtoon has proven to be a thriving market in Korea, with many aspiring Webtoon artists and publishing media platforms like Naver, Daum, and Kakao. More recently, it has even gained popularity overseas with official websites like webtoons.com, which translates them from Korean to English for a wider audience.

Increase in the popularity of Webtoons has led to an increase in the competition among the Webtoon artists. Figure 1 is a screenshot of the front page of Naver, one of South Korea's largest Webtoon publishing platform. Webtoons are divided into the respective weekday in which they are weekly released and sorted by decreasing popularity by default. This means that the lesser known Webtoons will usually stay at the bottom of the page and hence have a more difficult time reaching a wider audience. Amongst the hundreds of Webtoons being released weekly, a Webtoon will need to stand out in order to start gaining attention from the readers. In particular, we believe that the thumbnails for these Webtoon are crucial for this purpose.

Each Webtoon has a thumbnail connected to it in the front page. It is the snippet of image that appeals to the readers by displaying its art, tone, and mood. The thumbnail is usually the first impression to a lot of new readers, and even for old readers, the thumbnail can serve as a useful tool to select what Webtoons to read next. For example, they might recognize an art style of a new Webtoon, which was drawn by an artist who worked on other popular Webtoons. In other cases, the reader might not appreciate the art quality and choose other popular Webtoon.

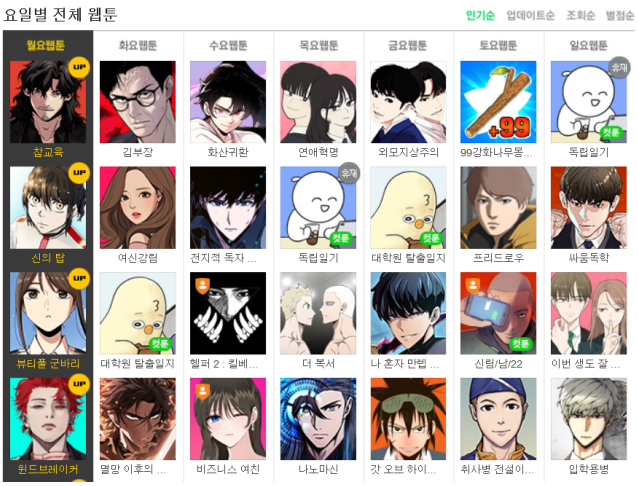

Figure 1: The front page of Naver Webtoon website divided into weekdays and sorted by popularity by default (https://comic.naver.com/webtoon/weekday)

2 PROBLEM DEFINITION

We are interested in determining whether a computer can learn to recognize an art style based on the thumbnails. Mainly, we want to examine if the thumbnail can be used to identify the artist/author of a Webtoon and classify the Webtoon to the correct genre. Lastly, we want to create a recommendation system that uses thumbnail to recommend to readers new Webtoons with similar art style.

CSCI 6380 Data Mining, May 09, 2022, Athens, GA

3 METHOD

For this project, we've decided to use Python because of the abundance of machine-learning libraries. For the neural network models, we used Keras, Tensorflow, and annoy (https://github.com/spotify/annoy).

For the dataset, since there is no existing dataset on Webtoons, we built a scraper using Python and scrapped the title, authors, genre, and thumbnail of each Webtoons on Naver Comics website (https://comic.naver.com/webtoon/creation). The thumbnails were encoded into a 3-dimensional array with shape equal to 101×125×3 (a few Webtoon with different shapes were discarded). We applied one-hot encoding on the genre attribute since there was only a few classes. For authors attribute, we applied a nominal encoding. For modeling, the dataset was split into training (65%), validation (15%), and testing set (20%).

For the first part, we made a CNN model with the architecture shown in figure 2 to identify the authors based on the thumbnail. We used the ADAM optimizer with accuracy as the metrics and sparse categorical cross-entropy for the loss metrics to be minimized. We fitted the model on the 3-dimensional encoding of the thumbnail for the input and integer encoding of the authors for the output. They were then evaluated on a testing dataset using typical classification metrics such as precision, recall, and F-measure.

For the second part, we made several CNN models to classify genre based on the thumbnail. We utilized AutoKeras, an AutoML library for Keras NN models, to create the CNN image classification models. As before, the resulting models were fitted on the 3-dimensional encoding of the thumbnail for the input, and for the output, 2-dimensional one-hot encoding matrix of the genre were provided. The resulting model was evaluated on the testing dataset with the same metrics as first part.

For the last part, we utilized the models created for first and second part to create embedding model to create a vector representation of the thumbnails for similar image search. We then added the embedding into locality sensitive hashing space using annoy library and used different distance metrics to find similar images.

4 EXPERIMENT

In the author classification, the model in 2 had the tendency of over-fitting and was unable to generalize beyond the training set. On the training set, the model scored 0.98955, 0.99261, and 0.99043 on precision, recall, and f1-score, respectively. However, on the testing set, the model scored 0.00325, 0.00664, and 0.00389. Figure 3 shows the sample output for five test cases. The thumbnails in the first row are the input thumbnails from the testing dataset. The thumbnails in the second row are one randomly chosen work from the predicted authors for the respective thumbnail. As clearly seen with the input and output in the first and the fifth column, the model does not seem to understand art style.

The model in genre classification was able to perform better than the author classification. This is somewhat expected since there are more instances for each categories for genre compared to authors. The AutoKeras generated the model architecture shown in figure 4, and interestingly, the model included an RNN architecture. On the training dataset, the model had scored 0.23910, 0.32053, and 0.19649 for precision, recall, and f1-score. On the testing dataset, the model

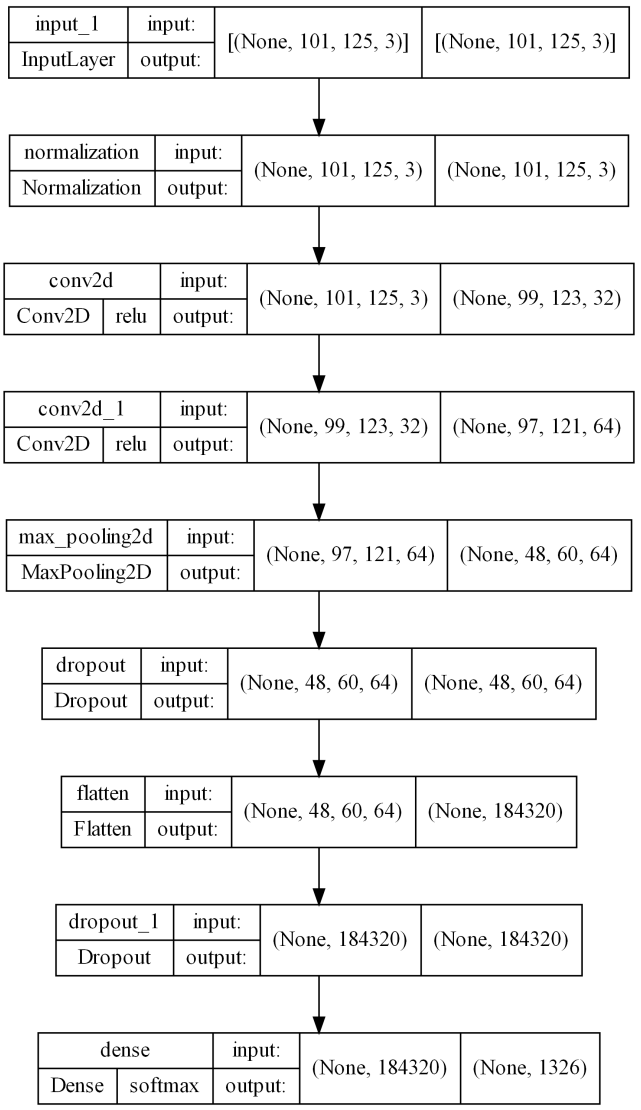

Figure 2: The CNN architecture used for classifying the authors of Webtoon thumbnails

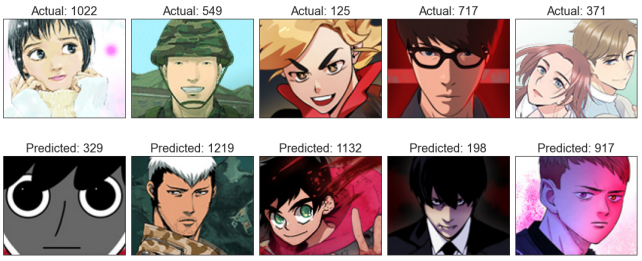

Figure 3: Visualization of the actual and predicted output for author classification

scored similarly with 0.22584, 0.28761, and 0.17386. Figure 5 shows 5 samples of the output and its actual and predicted genres.

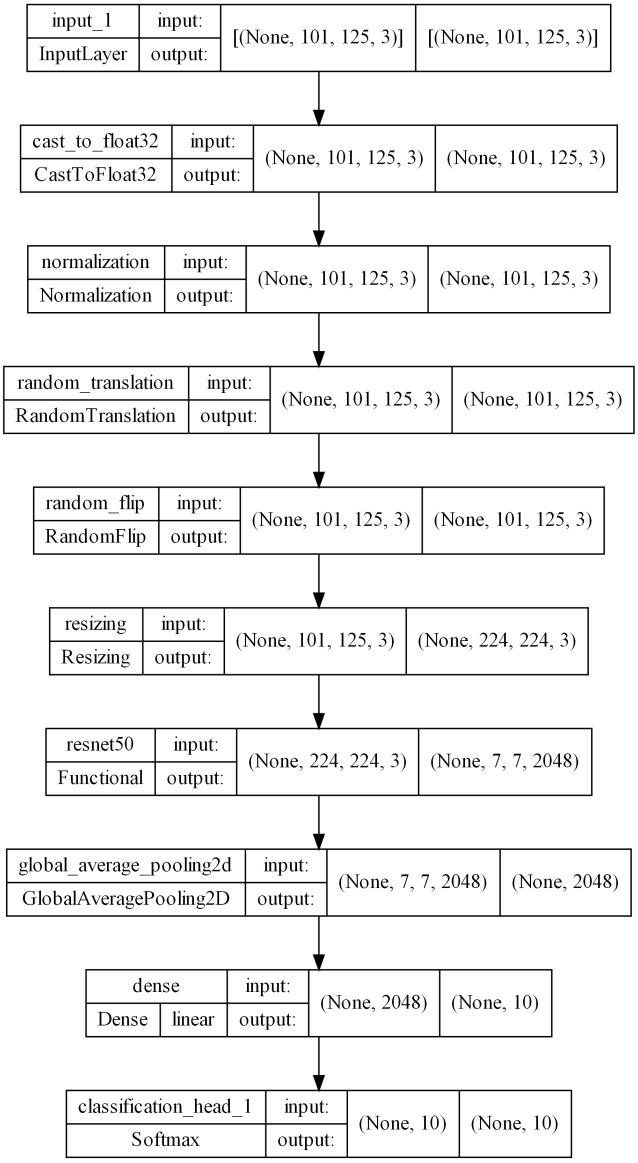

Figure 4: The CNN architecture generated by the AutoKeras used for classifying the genre based on the Webtoon thumbnails

Figure 5: Visualization of the actual and predicted output for genre classification

For the similar image search, the model created for genre classification was used to create an embedding model with the architecture

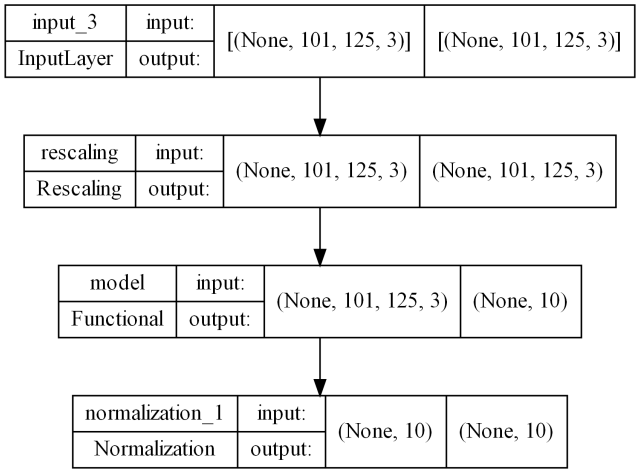

Figure 6: Embedding model created from genre classification model

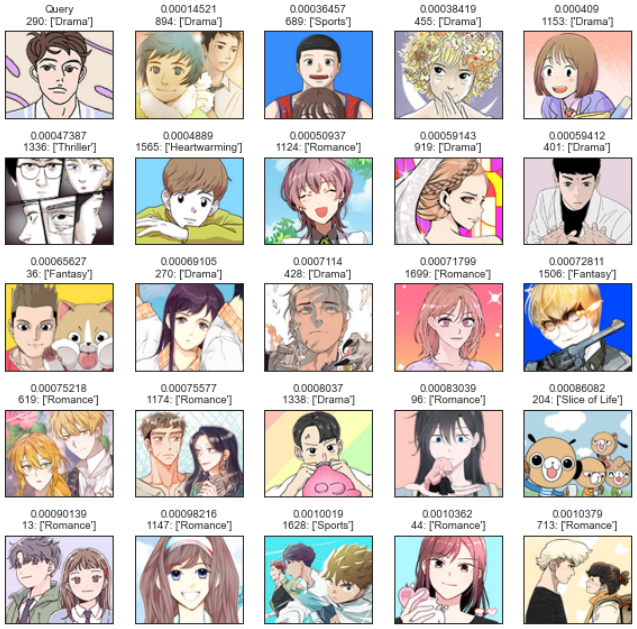

Figure 7: First example output of similar image search

shown in figure 6. The embedding model was used to generate embedding for each thumbnail in all of the dataset. The embeddings were added to a approximate nearest neighbors index module in annoy library that use locality sensitive hashing. For the distance metrics, we used a euclidean distance. For evaluation. a randomly selected thumbnail was chosen for query and 24 other thumbnails were selected based on the given query. Figure 7 and 8 demonstrate two different run of the similar image search on the Webtoon thumbnails. The first image on the top left corner is the query image, and the 24 other thumbnails are the nearest thumbnails found.

The difference in the results is clear based on the color scheme and the character portrayed.

5 CONCLUSION

There was a notable number of limitations in our project. First, we had a limited number of instances () available for training and testing. Future studies may incorporate different websites like Kakao and Daum to overcome this problem. Second, for legal reasons, we were not able to use the content of the Webtoons, which is a better representation of an author's art style. This was one of the reason we opted to use thumbnails of the Webtoons, which was a significant drop in the number of instances per author and genre. Third, due to the lack of computing resources like GPU and TPU hardware, we were not able to learn more complex models and to train the models for higher number of epochs.

Despite the result from the author classification, the result of genre classification and the image similarity, albeit subjective, show promising insight into the ability of CNN to learn artistic style and resemblance. With the growing popularity of Webtoons and similarly, manga, this technology can provide a potential benefit for the Webtoon publishing platforms and the Webtoon artists. Webtoon artists will be able to reach a wider audience with the use of recommendation system, and the Webtoon platforms can garner more audience thanks to the recommendation system that can specialize recommendation to each readers based on their taste.

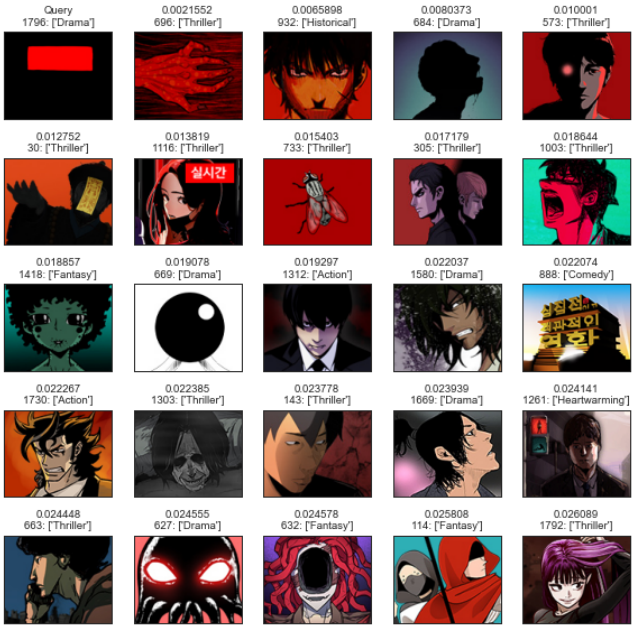

Figure 8: Second example output of similar image search