Anime Recommender with your Mood

Po-Yi Lee

Master of Computing and Innovation, University of Adelaide

ABSTRACT

Recommendation systems have become pervasive in recent years as many industries tend to suggest their products from a large amount of data including users' information and products themselves. In addition, given that virtual reality technology has been more frequently used, the future applications could be implemented with embedded computer vision tools such as emotion detection, gesture recognition. I used the fer2013 emotion datasets to train the models to develop an emotion detection engine with the user's gesture recognition function using Mediapipe. Since Japanese anime have received widespread support around the world, the anime dataset Anime Recommendation Database 2020 from Kaggle datasets is choson in this project. Apart from the general recommendation results using the traditional Recommendation system methods, I developed an advanced anime recommendation engine embedded with computer vision tools which can observe the user's mood in real time and give users updated recommendation lists by adding their emotional state into the consideration of the anime recommendation system.

1 INTRODUCTION AND MOTIVATION

Recommendation system plays an important role in various industries such as music applications, video platforms, food delivery and so on. After I learnt from different kinds of methods to implement recommendation systems, I consider that the limitation of these methods exist because the data and reference that they used in these methods do not include the user's mood at the moment they are watching or using the product. Therefore, in this project plan, not only understanding some of the recommendation methods, I would also like to develop a more advanced recommendation engine by combining the real-time emotion detection and gesture recognition system with these methods.

2 PROJECT NOVELTY AND SIGNIFICANCE

In this section, I will present why this project is novel and important for the future recommendation system. Traditional recommendation systems commonly look at the data from users' using experience and the features of items such as . However, what if we can know users' mood while they are watching? I believe that users' facial expressions are important to observe for the recommendation system. If we could obtain the users' emotional states and put them into the calculation of recommendation systems, this would be valuable datasets for various companies.

3 MATERIAL

3.1 Anime Recommendations Database dataset from Kaggle

The dataset for the recommendation System is called Anime Recommendations Database from Kaggle(https://www.kaggle.com/CooperUnion/anime-recommendations-database) which contains information on user preference data from 73,516 users on 12,294 anime including anime id, anime name, genre, type, the number of episodes, rating and members. There are two csv. file including Anime.csv (anime_id - myanimelist.net's unique id identifying an anime, name - full name of anime, genre - comma separated list of genres for this anime, type - movie, TV, OVA, etc., episodes - how many episodes in this show. (1 if movie), rating - average rating out of 10 for this anime, members - number of community members that are in this anime's "group") and Rating.csv(user_id - non identifiable randomly generated user id, anime_id - the anime that this user has rated, rating - rating out of 10 this user has assigned (-1 if the user watched it but didn't assign a rating)).

3.2 fer2013 dataset from Kaggle

In this project, I found the dataset called fer2013 from Kaggle which consists of 35887 grayscale, 48x48 sized face images with seven emotions - angry, disgusted, fearful, happy, neutral, sad and surprised(https://www.kaggle.com/deadskull7/fer2013). This dataset can be used to train the model to recognize different emotions. There are only two outcomes including "like" and "dislike" that we are interested in. Therefore, I categorize angry, disgusted, fearful, sad and neutral as "dislike". Happy and surprised belong to "like"(Shown in Table 1). While detecting the emotion outcome, the system will ask the users if they want to keep watching or not when five "dislike" outcomes show up.

Outcome	**Emotion**	**Amount in Training Dataset**	**Amount in Testing Dataset**	**Number**
Dislike	Angry	3995	958	0
Dislike	Disgusted	436	111	1
Dislike	Fearful	4097	1024	2
Like	Happy	7215	1774	3
Dislike	Neutral	4965	1233	4
Dislike	Sad	4830	1247	5
Like	Surprised	3171	831	6

Table 1. Categories and Distributions of Fer-2013 Dataset

4 METHODOLOGY

4.1 Recommendation System Methods

4.1.1 Popular-based approach

The popular-based approach to the recommendation system is the baseline performance and the most basic recommendation that we can see anywhere. For example, Top ten movies today is one of the popular-based recommendations. Since the users do not have too much information about them before they use products. It would be a safe method to recommend to the users but the limitation is that this implementation is not personalized.

We can have an access to the weighted rating score according to the number of votes for the item, the average rating for the item, the minimum votes required to be listed in the popular items and the average rating across in the whole dataset. The formula of the weighted rating score is:

$$Weighted\ rating = (v \div (v+m)) \times R + (m \div (v+m)) \times C$$

where:

R is the average rating for the item.

v is the number of votes for the item.

m is the minimum votes required to be listed in the popular items.

c is the average rating in the whole dataset.

In figure 1, we can obtain the top 10 popular anime result according to their weighted rating. The application would be shown in the beginning when the users turn on the application.

top10 popular anime:

	name	weighted_rating
1997	Steins;Gate	8.074834
1569	Fullmetal Alchemist: Brotherhood	7.989009
3314	Kimi no Na wa.	7.865382
2238	Hunter x Hunter (2011)	7.814615
1462	Clannad: After Story	7.783129
1269	Code Geass: Hangyaku no Lelouch R2	7.728091
1088	Tengen Toppa Gurren Lagann	7.662938
2931	Shigatsu wa Kimi no Uso	7.645272
9	Monster	7.615327
944	Code Geass: Hangyaku no Lelouch	7.609182

Figure 1. Top 10 popular anime result using the popular-based approach

4.1.2 Content-based filtering approach

In[1], the content-based filtering approach use the concept of TF-IDF (Term Frequency-Inverse Document Frequency) to categorize the items based on the similarity of the items. In order to compute the similarity between item vectors, one of the methods is called Cosine similarity method which requires the angle of cosine between the two items. This angle of cosine can be computed by finding the dot product between the two item identities in figure 2. According to [], the two items have more similarity when the angle is lesser. Therefore, the content-based anime recommendation lists can be formed by finding the angle of cosine.

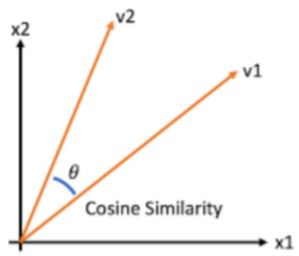


Figure 2. The angle between two vectors v1 and v2.

In this project, firstly, the important features of each anime including their name, genre and type would be extracted and then these features can be converted into a matrix of their token counts by using the method **CountVectorizer** from the sklearn library. Once we obtain the count matix of these anime, their cosine similarity would be computed in a matrix form by using **cosine_similarity** method from the sklearn library. Finally, the list of the most similar anime can be ranked based on their cosine similarity value (refers to cosine theta). The cosine theta ranges from 0 to 1. The example result of this method is shown in figure 3.

```
The 10 most recommended anime to Kimi no Na wa. are:

              name                 consine similarity
1 16     ['Shigatsu wa Kimi no Uso'] 0.6030226891555273
2 1494   ['Harmonie'] 0.5962847939999438
3 60     ['Hotarubi no Mori e'] 0.5892556509887895
4 1959   ['Air Movie'] 0.5892556509887895
5 1111   ['Aura: Maryuuin Kouga Saigo no Tatakai'] 0.5773502691896258
6 2103   ['Clannad Movie'] 0.5555555555555556
7 5796   ['Taifuu no Noruda'] 0.5555555555555556
8 894    ['Momo e no Tegami'] 0.5443310539518174
9 6119   ['Shisha no Sho'] 0.5443310539518174
10 986   ['Shakugan no Shana'] 0.5270462766947299
```

Figure 3. Top 10 content-based anime result using the cosine similarity approach

This recommendation system approach would be implemented in the application after the users select their first anime and do not like this one after detecting their face expression.

4.1.3 Collaborative filtering using k-Nearest Neighbors approach

Compared to content-based recommendation approach which only considers the item itself, collaborative filtering approach collaboratively filters items from a large set of the other options between users preferences. According to [2], if two users A and B liked the same product, they will also have similar interests in the future.

Therefore, I merged the users' ratings on each anime into a table using the method **pivot_table** in Pandas library and the method **csr_matrix** in scipy library. Then, we can find the distance between the selected anime and the other anime by implementing k-Nearest Neighbors method and ranking the anime based on these distance values.

```
Collaborative filtering Recommendations using k-Nearest Neighbors for 1001 Nights:

   name                          distance
1: Doujouji  0.6951378907399183
2: Tori no Uta  0.6994727644692178
3: Kataku  0.7351508534365385
4: Bavel no Hon  0.7358126894960499
5: Aru Tabibito no Nikki Specials  0.7358977060716129
```

Figure 4. Top 5 anime result using Collaborative filtering using k-Nearest Neighbors

4.2 Emotion classification and detection

4.2.1 CNN model

In terms of image recognition and classification, in recent years, deep learning methods are considered to be the popular methods on facial emotion recognition and Convolutional neural networks (CNNs) are one of the methods specialized in this area [3]. To train the model which can be used in the webcam feed, firstly, the haar cascade method is used to detect the user's face in each frame of the video process. Secondly, the region of face image is resized to 48x48 and is put as input into the CNN to predict one of seven emotions. The structure of CNN I use consists of 4 Conv2D layers with "ReLu" function as an activation function and 3 2x2 MaxPooling layers. In the last layer, the CNN outputs a list of softmax scores for all the labels of emotions and the emotion label with the highest score will be selected as the predicted emotion of the user.

4.2.2 Different models and evaluation metrics

In addition to the CNN model, I also implemented three keras API including vgg16, resnet50 and mobilenet in 5 epochs on the image classification. The table of each result is shown in table 2.

	accuracy	**f1 score**
vgg16	0.7031	0.0479
Resnet50	0.6427	0.0000e+00
mobilenet	0.6511	0.0029

Table 2. Accuracy and f1 score on different keras API on Anime Recommendations Database

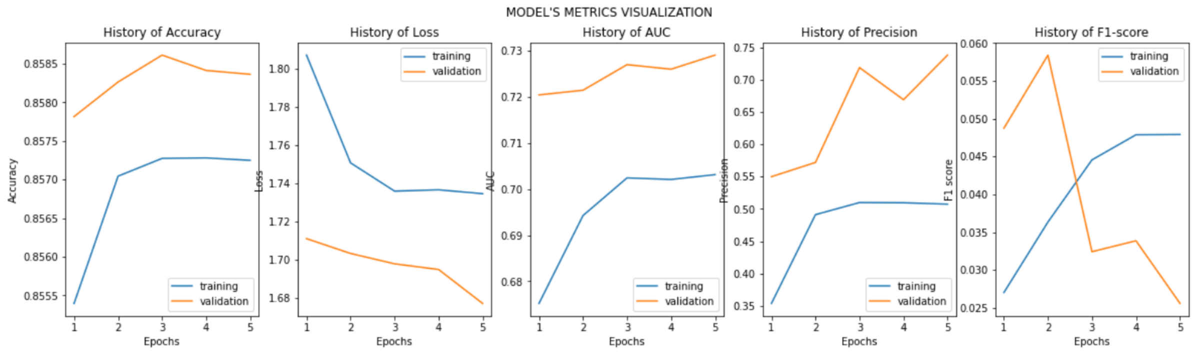


Figure 5. Vgg16 model

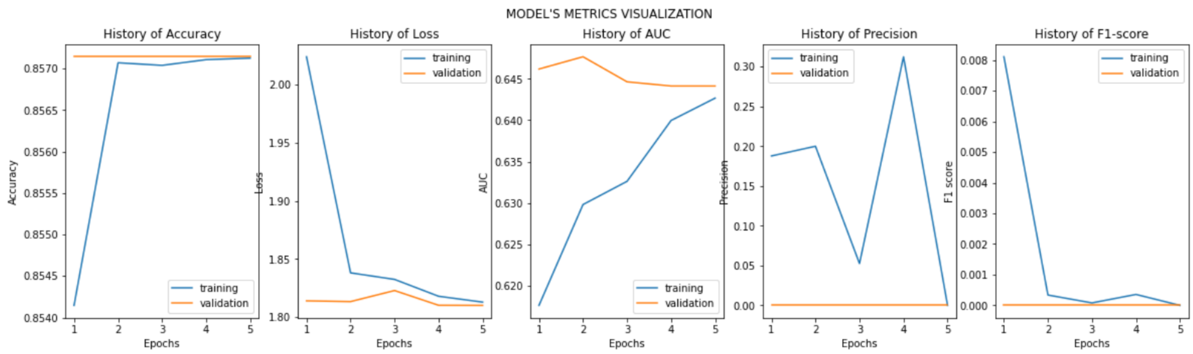


Figure 6. Resnet50 model

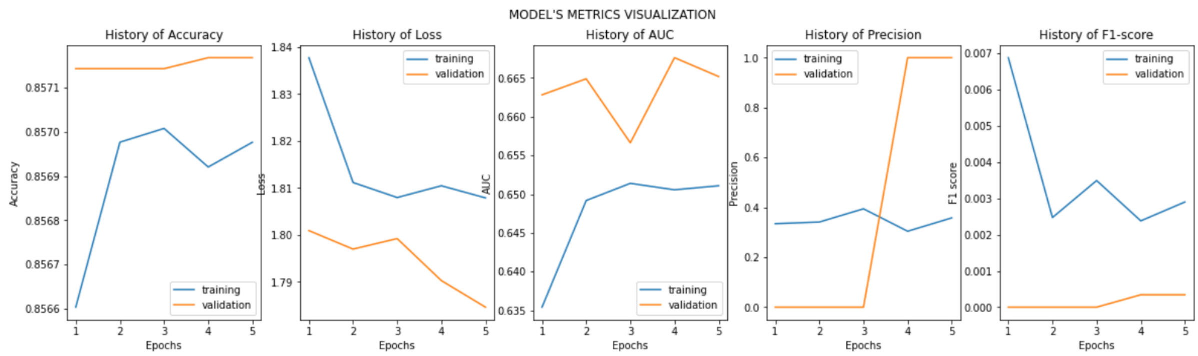
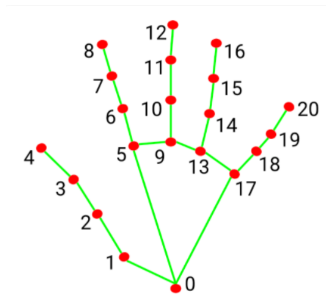


Figure 7. Mobilenet model

4.3 User gesture recognition

After determining what emotion the user has, the system will ask the user if he or she likes this anime or not. For the convenience of user experience, I add a gesture recognition function into the application to identify the user's gesture including thumbs-up and thumbs-down. In order to achieve this, I need to train the ML system to classify the different gestures. In the beginning, I found a dataset called Gesture Recognition from Kaggle(https://www.kaggle.com/imsparsh/gesture-recognition) in order to train a model able to predict the thumbs-up and thumbs-down gesture using CNN method and a tool called MediaPipe Hands which can employ machine learning to infer 21 3D landmarks of a hand from just a single frame. I use the MediaPipe Hands to track hand and finger movements and find 3D landmark locations of hands because the way of finding the position of hands is more accurate than CNN. I have experimented with different formulas to distinguish two gestures. Finally, thumbs-up can be predicted as "like"(Figure 9) if the position of index_finder_mcp(Landmark 5 in Figure 8) wrist is higher than thumbs_tip(Landmark 4 in Figure 8) and the position of thumbs_tip(Landmark 4 in Figure 8) is higher than wrist(Landmark 0 in Figure 8). On the other hand, thumbs-down can be predicted as "dislike" if the position of the wrist(Landmark 0 in Figure 8) is higher than thumbs_tip(Landmark 4 in Figure 8).



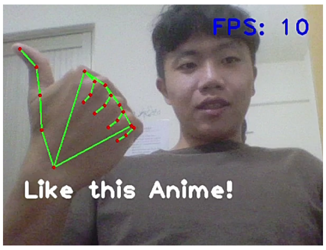


Figure 8.: Hand landmarks in MediaPipe Figure 9. The visualized prediction outcome of "like"

5 SOFTWARE ARCHITECTURE

In this section, I will present the software architecture of this project (shown in figure 9). When the users turn on the application, the popular-based recommended list will be displayed based on the anime recommendation database. Assume that the users have choson one anime, the camera will start to detect their facial expression with a trained model which was trained with fer2013 dataset. If the prediction of the emotion detector has accumulated five times "dislike", the gesture recognition system will start to ask the users if they like the anime which the users are watching. If the recognized outcome is "like", then the process will go into the emotion detecting section. Otherwise, the content-based recommended list will remove the anime which they are watching and the updated recommended list will be displayed to the users.

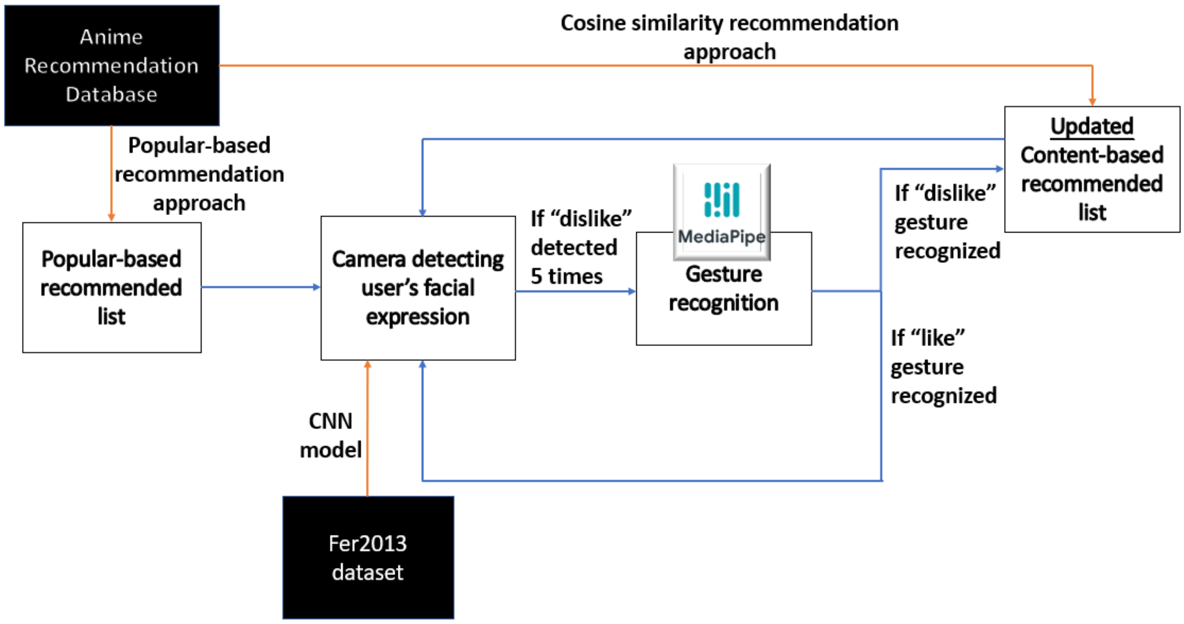


Figure 10. Software architecture of Anime Recommender with your Mood

6 KNOWLEDGE LEARNT FROM THIS COURSE

In this section, I will conclude what I learnt from this course.

Firstly, I was very interested in the recommendation system but had no chance to dive into this machine learning field. Since I took this course, I have learnt various recommendation system methods to implement such as popular-based, content-based approach, collaborative-filtering approach and so on. Due to the time limitation and the requirement of my software system, I only implemented some of the methods.

Secondly, I have also learnt how to implement the image classification using the different models such as CNN, vgg16, resnet50, mobilenet etc. This field is interesting, deep and large to research. Also I understood how to combine these training models with web cameras.

Thirdly, I have learnt a new tool called Mediapipe which is fun to be applied to many projects in the future. In addition to the hand detection, there are also upper body detection and the whole body detection.

Last but not least, I improved my presentation skills in these pitch presentations throughout the semester. I did not have too many chances to present the things I learnt and was nervous to do the presentation in public. However, in this course, I experienced how to prepare my pitch and talk about what project I was doing in front of other people.

7 CONCLUSIONS

An advanced recommendation software has been developed in combination with the traditional recommendation systems and emotion recognition. In addition, an interactive function of gesture recognition is also added in this application. In the future, I will add more recommendation system approaches into this system such as collaborative filtering approach which is not used so far.

REFERENCES

[1] Cui, B-B 2017, ‘Design and Implementation of Movie Recommendation System Based on Knn Collaborative Filtering Algorithm’, ITM Web of Conferences, vol. 12, p. 4008–.
[2] Thorat, P.B., Goudar, R.M. and Barve, S., 2015. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. International Journal of Computer Applications, 110(4), pp.31-36.
[3] Giannopoulos, Panagiotis, Isidoros Perikos, and Ioannis Hatzilygeroudis. "Deep learning approaches for facial emotion recognition: A case study on FER-2013." Advances in hybridization of intelligent methods. Springer, Cham, 2018. 1-16.