Genre Detection from Movie Posters

Shivam Sood
193050023

Srijon Sarkar
193050038

Soumyadeep Thakur
193050033

Nipun Mittal
193050005

November 24, 2019

CS 725 Project

Dept. of Computer Science and Engineering
Indian Institute of Technology, Bombay

Contents

1 Objective

Movie posters are universally used to advertise movies and a good poster must convey important qualities of a film such as theme and genre to make the movie seem appealing to audience. Posters represent one way in which people use media to influence human behavior(2). In this work, we predict the genre of a movie by looking at it's poster. It is beneficial to both the movie industry(so that advertisers can better and precisely impact their audiences) and to the researchers(by identifying the prominent feature aspects which impact the audiences).

2 Literature Review

There have been some previous works on poster based genre detection. Wei-Ta Chu and Hung-Jui Guo (1) presented a system to classify movie poster images into genres. A deep neural network is proposed to jointly consider visual appearance and object information, and a classier is constructed to estimate the probabilities of a poster belonging to different genres. They have achieved Multi-label classification by thresholding the estimate probabilities, with the thresholds adaptively determined by a grid search scheme. Gabriel Barney and Kris Kaya (2) have founded that the ResNet network and the Custom Architecture, despite performing only slightly better in pure accuracy when compared to ML-kNN, performed well in other important evaluation metrics such as F1 Score, Recall and Top K Categorical Accuracy. Their model is biased towards the labels(genres) which are frequent in the given dataset. Leslie Tu, Petra Grutzik and Kate Park (3) have founded that judging movie quality is a difficult problem because of its subjective nature. Simonyan, Karen, and Andrew Zisserman (4) evaluated very deep convolutional networks (up to 19 weight layers) for large-scale image classification. It was demonstrated that the representation depth is beneficial for the classification accuracy, and that state-of-the-art performance on the ImageNet challenge dataset canbe achieved using a conventional ConvNet architecture with substantially increased depth.

Multi-Label classification does not have any go-to accuracy measures. There are many measures which are be used but none of them alone can give a clear picture of the predictions. This is one major point which makes a multi-label classification difficult.

Another difficulty in multi-label classification is the skewness in the data. There maybe some classes which have a lot of data, while others may be starved. We don't directly deal with this problem in our project.

3 Set of Approaches Used

In this project we have used Convolution Neural Network based models for predicting genres from posters. Since one poster can be classified into mutiple genres, the problem at hand is a multi-label classification problem.

We used 3 different models for genre prediction as descibed in Section 3. For **Model 1** we used a CNN network having 9 convolutional layers and 2 dense layers. For

Model 2, we use the VGG19 (4) network pretrained on the ImageNet dataset. The output of the VGG19 network is passed through 2 dense layers. For both the models, the output layer has M units and a $sigmoid$ activation, where M is the number of genres.

For **Model 3** we combine both the previous 2 approaches. We use the output of the second last dense layer of the pre-trained VGG model and concatenate with the output of a 6-layer CNN network. This concatenated input is then fed to 2 dense layers, followed by an output layer of M units (with sigmoid activation).

As an illustration, the architecture of the CNN network used in **Model 3** for our experiment is shown in Figure 1

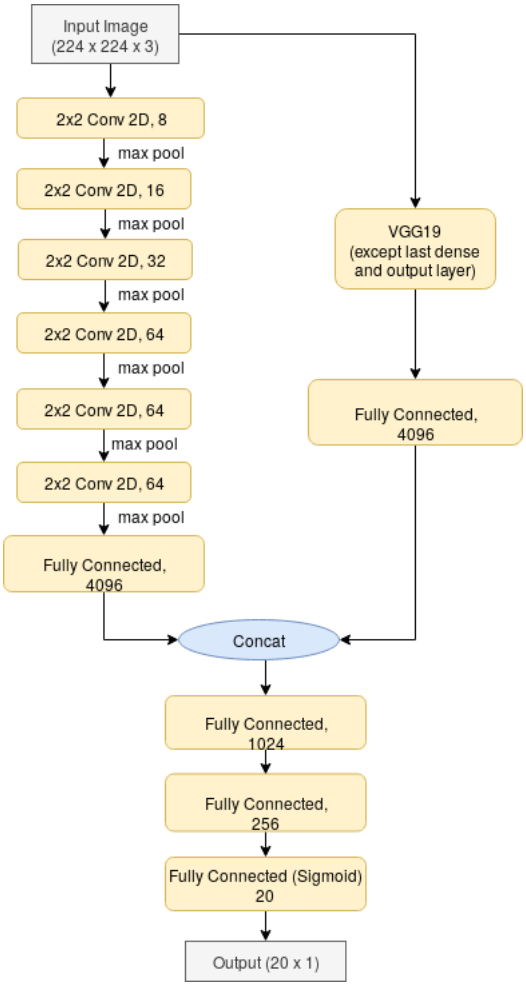


Figure 1: Architecture of Model 3

4 Experiments / Methodology

In this project we used The Movies Dataset downloaded from `https://www.kaggle.com/rounakbanik/the-movies-dataset`. The dataset downloaded has data for 45570 movies from last 4-5 decades.

4.1 Code

All the code presented is written by us. No code is taken from anywhere. Keras (Tensorflow - Python) is used for creating nueral networks. Colab is used for training. On Colab we used TPU as runtime, with 35 GB RAM and 50 GB Disk. Our project can be found here: github.com/imagine5am/CS725-genre-prediction.

4.2 Preprocessing

The dataset has IMDB ids for each movie. All movies which were not in English were removed from our dataset. Also, we considered only those movies that were released after 1990. This is because the poster images were then downloaded from the official IMDB website using web scraping techiques. Also, modern movie posters look different from the ones from previous generations, and the inherent features will also be different. We obtain 18938 images in this way. The movies are classified into a total of 20 genres. Entire web scraping and preprocessing code is written in python.

For **Model 1** and **Model 2**, the train-test split ratio is $85 : 15$. 20% of train data is further used for validation. For **Model 3**, the data was split into train and test data in the ratio of $2 : 1$. All splits are done using iterative stratification algorithm (5) which is generally done for multi-label data. We used a python library *skmultilearn* for this.

4.3 Training

All three models **Model 1**, **Model 2** and **Model 3** were trained using binary cross-entropy loss between the predicted genres and the true genres, i.e. we calculated the loss as cross-entropy loss over all labels. The code for models and training were written using *keras* with *tensorflow* as backend.

Assuming total number of class labels to be M, for the i^{th} sample the binary cross-entropy loss is defined as:

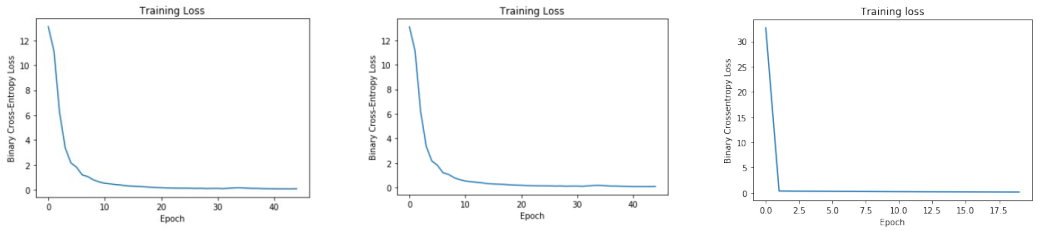
$$BinaryCrossEntropy = -\sum_{k=1}^{M} y_i^k \log o_i^k + (1 - y_i^k) \log(1 - o_i^k)$$

where y_i is the vector of true labels for sample i and o_i is the vector of predicted labels.

The cross-entropy loss was minimized using *Adam Optimizer* (6) The learning rate was initialized with 0.001.

The plots training loss against epochs for the different models are shown in Figures 2.

4.4 Results



(a) Model 1 (b) Model 2 (c) Model 3

Figure 2: Training Loss vs Epoch for different models

We used subset accuracy for evaluation of both the models. We define 2 variants of subset accuracy:

Top 3-Subset Accuracy: Let O_i' be the set of 3 genres with highest scores for sample i, and let Y_i' be the set of true genres for that sample. Let there be total N samples in the dataset, then subset accuracy is defined as:

$$Top3SubsetAccuracy = \frac{1}{N} \times \sum_{i=1}^{N} \frac{|O_i' \cap Y_i'|}{|O_i' \cup Y_i'|}$$

Truncated Subset Accuracy: Let O_i'' be the set of predicted genres by our model for sample i, for which the value of the output is more than 0.5. Let Y_i'' be the set of true genres for that sample. Let there be total N samples in the dataset, then subset accuracy is defined as:

$$TruncatedSubsetAccuracy = \frac{1}{N} \times \sum_{i=1}^{N} \frac{|O_i'' \cap Y_i''|}{|O_i'' \cup Y_i''|}$$

The subset accuracies obtained by the models are shown in Table 1

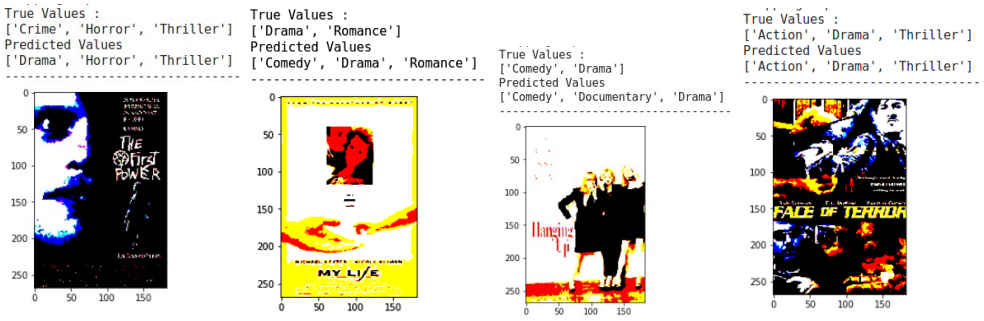
Table 1: Subset Accuracy obtained by our models

Model	Truncated Subset Accuracy	Top-3 Subset Accuracy
Model 1	21.50	21.96
Model 2	**34.36**	**30.50**
Model 3	31.16	29.97

Model 2 clearly outperforms model 1 and model 3. The reason for this is not clear. Given more time we can improve model 3 and its accuracy can be increased. The accuracy achieved (i.e. around 30%) is quite more than the one achieved in (1) which was 18%.

Although the accuracy achieved may appear subpar but it is a good number given it is a multi-label classification problem. To get an taste of how good the predictions are notice the predicted labels in Figure 4. Although the correct genre is 'Comedy', the classifier predicted an additional label 'Romance' which clears fits the context.

Figure 3 contains the examples of some good predictions. And Figure 5 contains some mis-classified samples. But please notice these posters and try to predict the genre yourself. Is your guess correct?



(a) Example1 (b) Example2 (c) Example3 (d) Example4

Figure 3: Above are some good predictions by the trained model

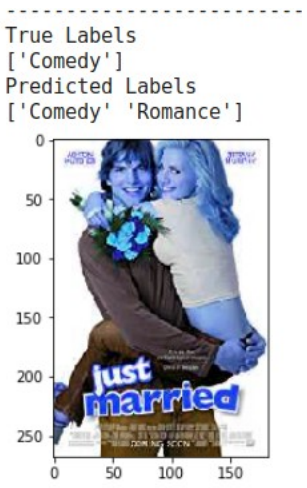
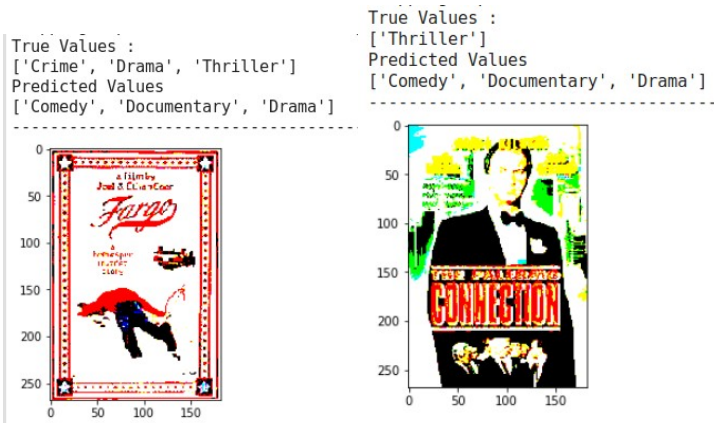


Figure 4: One of our best prediction



(a) Example1 (b) Example2

Figure 5: Above are some bad predictions by the trained model

5 Efforts and Miscellaneous

We had on the whole around 3 weeks to implement this project from scratch. This included doing a literature survey, checking multiple data sets, and experimenting with different models and finding the ideal accuracy metric for our classifier.

Around 30% of the time was spent downloading the posters and doing literature survey. Majority time(60%) was spent creating different models to classify the posters, and hyper parameter tuning in an effort to get the best results out of the models. The remaining 10% of the time went behind documentation.

Combining the output from 2 different models in order to create our **Model 3** was the most challenging part, not only from a coding perspective but also to get it to train. Had we had more time to work in this project, Model 3 would definitely outperform the others.

The web scraping part was handled by Soumyadeep and Nipun. Pre-processing was done by Soumyadeep, Shivam and Srijon. Model 1 was developed by Shivam. Model 2 was created by Srijon, and Model 3 was created by Srijon and Soumyadeep jointly. The metrics chosen to test the models were implemented by Srijon and Shivam. Project report was prepared by Soumyadeep, Nipun and Srijon. Final edit was done by Shivam and Srijon. Presentation was prepared by Nipun and Srijon.

References

[1] Chu, Wei-Ta and Guo, Hung-Jui. "Movie Genre Classification Based on Poster Images with Deep Neural Networks", 2017.

[2] Gabriel Barney (barneyga) and Kris Kaya (kkaya23). "Predicting Genre from Movie Posters." Department of Computer Science, Stanford University.

[3] Leslie Tu, Petra Grutzik and Kate Park. "Deep Tomato." Department of Computer Science, Stanford University.

[4] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint* arXiv:1409.1556 (2014).

[5] Sechidis, Konstantinos, Grigorios Tsoumakas, and Ioannis Vlahavas. "On the stratification of multi-label data." *In Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 145-158. Springer, Berlin, Heidelberg, 2011.

[6] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint* arXiv:1412.6980 (2014).