iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: IV Month of publication: April 2021

DOI: https://doi.org/10.22214/ijraset.2021.33947

www.ijraset.com

Call: ✆08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429*
*Volume 9 Issue IV Apr 2021- Available at www.ijraset.com*

# Movie Recommendation System through Movie Poster using Deep Learning Technique

Harshali Desai[1], Shiwani Gupta[2]
*[1]M.E. Second Year in Computer Engineering, [2]Assistant Professor, Thakur College of Engineering and Technology, Mumbai, India*

***Abstract:** Movie recommendation system plays an important role for all media service providers like Netflix, Amazon Prime etc. to increase their business by providing user with appropriate list of movies from very broader lists. By helping user to choose movies of their interest, media service providers thus attract huge traffic which indeed helps them to increase revenue. Very often target audience opt to watch movie by just looking at the movie poster. Thus, users can immediately gain an idea about a particular movie from its movie posters. So, from movie posters one can easily guess the genres of the movie. In this paper we have proposed movie recommendation system which recommends the movies based on its genres which are predicted through its movie posters. Firstly, the movie posters are obtained from IMDB website by performing web scraping. These movie posters are then used to train the Convolution Neural Networks. Convolution Neural Networks then predicts the genres of the movie and then based on predicted genres movies are recommended to the end user based on their cosine similarity. The proposed model is than compared with the existing baseline model and found that our proposed model performs slightly better in terms of accuracy, f1-score and precision.*

***Keywords:** Movie Recommendation, Convolutional Neural Network, Movie posters, Cosine similarity, Web scraping, Movie genres*

## I. INTRODUCTION

There is tremendous rise in usage of recommender systems in various E-commerce sites, movie Web Services, music Web Services etc. from last few decades. Recommender system have therefore become a crucial part of daily online activities. In very simple term, recommender system aims to find or suggest relevant items to the user based on user's behaviour, past history etc. Almost all application uses recommendation system so as to increase their revenue by meeting the user's expectations and giving them best user experience. Specifically, in the field of entertainment, various movie applications have millions of movies for the user to browse and watch. But for users to browse movies by their own and then select movies becomes a time-consuming activity. At this place, movie recommender systems come into the picture which helps the user to find the right movie by minimizing the options. Movie Recommendation Systems aims at suggesting relevant movies to the users. The basic algorithms for recommendation systems are:

### *A. Collaborative filtering*

Recommends based on similar users. This method works by finding similar group of users from a large set of users, which is similar to the targeted user by studying their similar taste.

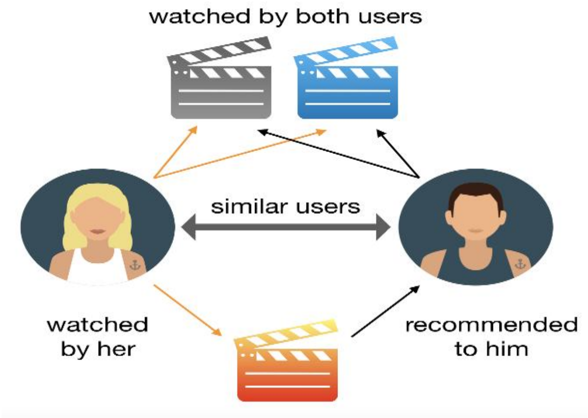


Fig. 1 Collaborative Filtering [1]

### *B. Content-Based Filtering*

Recommends based on user's history itself. In this method, similarity between the items of same user is found. Also, user's history is also considered for finding similar items for him.

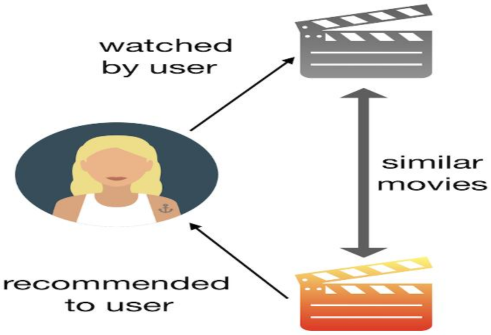


Fig. 2 Content Based Filtering [2]

### *C. Hybrid Method: Hybrid of Above two Approaches.*

Apart from the above basic algorithms there are many other approaches developed from last few decades for recommending movies which we will study in literature survey section. Recommendation System may also use various similarity measures in order to obtain similarity between two contents or items. There are many ways to calculate similarity between two items, but our proposed methodology uses cosine similarity to calculate similarity between actual and predicted genres.

Based on literature survey, a proposed methodology for movie recommendation system using the movie posters will be explained in further section.

## II. LITERATURE REVIEW

The authors of paper [3] have proposed a novel algorithm where Recommendation system deals with analysing the behaviour of user over the passage of time and predict the item ratings. According to authors, a lot of work is done to improve recommendation system. But still most model fails due to sparsity and also suffer from high computational complexity due to usage of heavy deep learning and machine learning approaches. To deal with this issue, authors have proposed a novel Time Fly algorithm for building simplified recommendation system. The authors have carried this experiment on various versions of MovieLens dataset and compared their model with another algorithms.

In paper [4] the authors have proposed a movie rating prediction algorithm named 'MCBF-SVD', which is based on explicit data and implicit data from movie datasets to predict the future ratings of movies from users with a certain degree of activity. In RF algorithm, they firstly, removed the users who rated less than 50 movies and their rating records, then calculates the difference value between the average rating of each user and the average rating of all users, after that deletes the users with the larger difference value. Finally, they kept the remaining users' rating data for carrying out further process. The novel Rating filtering (RF) algorithm proposed by the authors removes users with excessive rating differences to increase the MAE and RMSE. So, the main idea of the RF is to delete the users with large differences from the average rating of all users. Further, authors have proposed MCBF-SVD to alter rating according to the movie categories. Based on this, they used SVD to predict the future rating of movies from users. Their experimental results proved that the MCBF-SVD could effectively reduce errors of rating prediction models. In addition, this method can assist to increase the variety of recommended movies and alleviate the cold-start issue in theory. The authors used 2 famous movie datasets: hetrec2011, movielens-2k-v2 (an extension version of MovieLens -10M) and ml-latest.

The authors of paper [5] have used Natural Language Processing technique to generate more consistent version of Tag Genome, which is a side information that is associated with each movie in the Movie Lens 20M dataset. Also, they have proposed a 3-layer autoencoder so as to create more compact representation of tags which can improve the accuracy and computational complexity. Finally, the authors have combined the proposed representation with matrix factorization technique so as to develop a unified framework that outperforms state-of-art models by at least 2.87% RMSE and 3.36% MAE. The authors have firstly tried to reduce the total number of tags by combining the similar genome tags together. To eliminate the effect of freely user-created tags, the authors proposed to apply a mapping process: original tags which share the common context are grouped into a new tag associated with a composite score. Then a natural language processing technique named word2vec is used to cluster the same meaning tags. So, authors have used spaCy library for implementing pre-processing step and for calculating the similarity score between two tags.

They have kept a fixed threshold value of 0.65, which indicates that if the similarity score of two tags exceeds this value than the 2 tags is consider to have same meaning. The similar tags are then clustered, which gives reduced representation of tags. Finally, a composite score is assigned to the new tag.

The above step slightly improves the accuracy, but authors have further used autoencoders to discover a latent feature embedded in raw data. So, to keep reducing the dimension of genome tags and learn hidden structures they attempt to apply an autoencoder to newly created tags in the previous step. Then the proposed model is integrated with Matrix Factorization techniques, SVD and SVD++.

The authors in this paper [6] have proposed movie recommendation model based on word vector feature. The authors have used Doc2Vec model to extract the semantics, grammar and word order of the sentence, then transform it into a fixed dimension vector, then calculate the similarity of the vector and finally apply it to the collaborative filtering recommendation algorithm.

In the paper [7], authors have proposed a method for effectively recommending preferable movies for each user by using community user's movie rating information and movie metadata information with deep learning technology. A simple and effective item recommendation model is used based on Word2Vec algorithm with metadata. The proposed method uses various metadata of movie, such as movie director, actor, production year, production cost, movie tag etc. The values of these metadata are embedded as vector and are used as input and output of proposed Word2Vec network. Movie embedding is also used as input and output with meta data embedding. This input output data is obtained from user’s viewing history and purchase history. The inputs are initialized with pretrained embedding using the Word2vec algorithm. Two methods are used for obtaining pretrained metadata vector. The input embedding is generated by concatenating those of pretrained metadata embedding and movie embedding.

## III.PROPOSED METHODOLOGY

### *A. Dataset*

The proposed work initially uses the dataset taken from Kaggle [8]. The dataset contains various files like rating.csv, keywords.csv, links.csv, credits.csv etc. But we’ll be only interested in movies_metadata.csv file. This file contains information of 45,466 movies which is also featured in Full Movie Lens dataset. Various features of movies are included in this file like genres, IMDB Id, budget, revenue, release dates, languages, production countries, companies etc. The dataset consists of movies released on or before July 2017. Since the proposed work focuses on predicting genres from movie posters, we have downloaded various movie posters from IMDB website by performing Web Scraping on initial dataset. A Python Framework, BeautifulSoup [10] is used for web scraping process. To perform web scraping process, we need to obtain the IMDB website link for various movies. For this we have created a IMDB link. The IMDB link is created based on the ImdbId from the dataset. To create IMDB link for particular movie we concatenate IMDB website [9] home page link with the ImdbId for particular movie from the dataset. Since the structure of all movie pages on IMDB website are same. So, through web scraping we can easily retrieve poster link of each movie by simply going to its IMDB page and taking the content of the ‘src’ HTML attribute corresponding to the poster. Once we have got all poster links, we add them to our dataset by creating a new column. So, this web scraping process gives as the IMDB poster link from where we can now download our movie posters using these poster links

Below diagram show the implementation flow for obtaining movie posters from IMDB website.

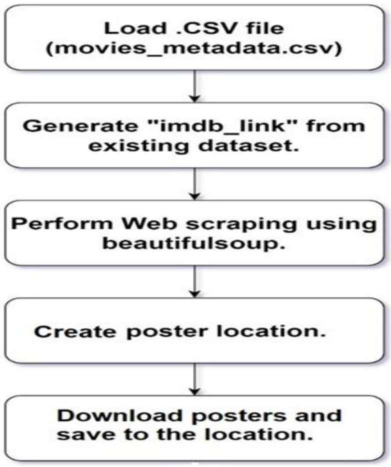

Load .CSV file
(movies_metadata.csv)
Generate "imdb_link" from existing dataset.
Perform Web scraping using beautifulsoup.
Create poster location.
Download posters and save to the location.

Fig. 3 Implementation flow of movie poster collection

Fig. 4 Movie Poster Dataset

### B. *Data Pre-processing and Analysis*

Initially we use a dataset which is available on kaggle. From this dataset we require only ImdId, genres, title column. The rest of the columns are eliminated as they are of no use. Now we need to clean our dataset so that we can achieve proper accuracy. We have cleaned our dataset by eliminating all those rows which contains null ImdbId and genres, also we have eliminated all those rows which have empty genres. We have also drop unwanted N/A genres from the dataset.

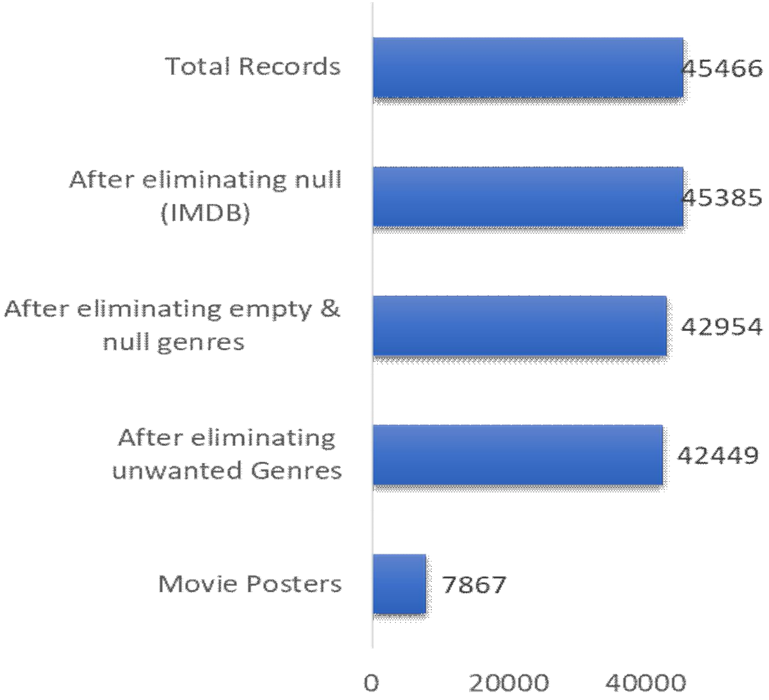


Fig. 4 Dataset Summarization

From the below Word Cloud and Histogram, we can say that the Drama, Romance, Comedy, Adventure and Crime are most frequently occurred genres in the dataset. Whereas, the least occurred genres are Western, Short, Animation, War, News, Reality-Tv and Documentary. There are in total 24 Genres used in our proposed model.

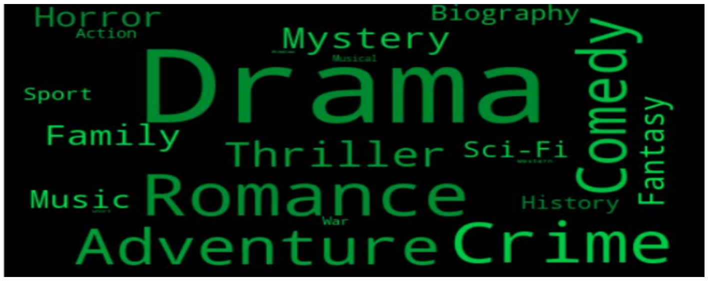


Fig. 5 Word Cloud for genres

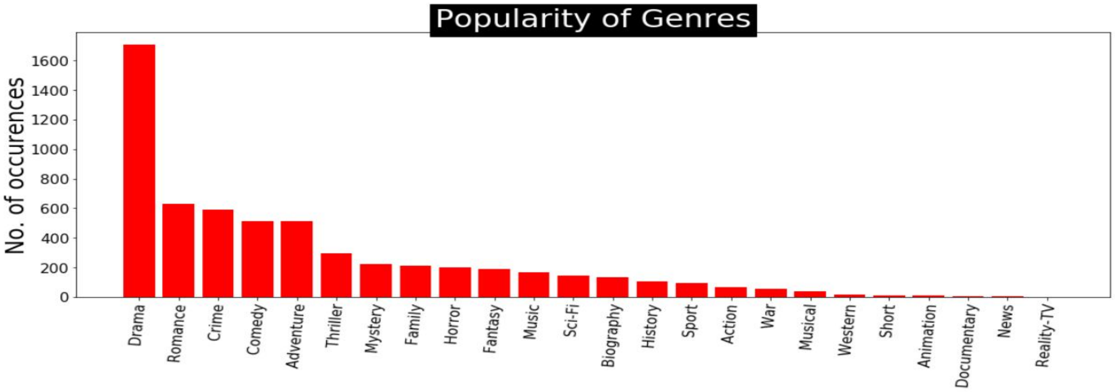


Fig. 6 Histogram for genres

### *C. Proposed Model*

For prediction of genres, we are building our Customized Convolutional Neural Network by using Keras framework which allow to build Deep Learning model. We will be using Sequential Model, which is the easiest way to build a model, since it allows to build a model layer by layer. So, we'll stack sequential layers on top of each other. Once the model is built, we train it with help of 80-20% training and validation dataset respectively. And finally, 10% dataset are used to predict the test instance and evaluate the result.

Let's first understand some terms used to build CNN model.

*1)* Convolution Layer: They are important layer which is used to apply filter to extract features from original image.

*2)* Pooling Layer: Pooling Layer: It is used to reduce the dimensionality. They are of 3 types- Max pooling, Min pooling and Average pooling. Max pooling will take the maximum pixel value from the part of image matrix. Min pooling will take the minimum pixel value from the part of image matrix. Average pooling takes the average pixel value from the part of image matrix.

*3)* Dense Layer: This is also called as fully connected layer. Fully connected layers are placed before the classification output of a CNN and are used to flatten the results before classification [10]

The customized CNN model is constructed so that we achieve maximum accuracy. The block architecture for proposed CNN model is shown below.

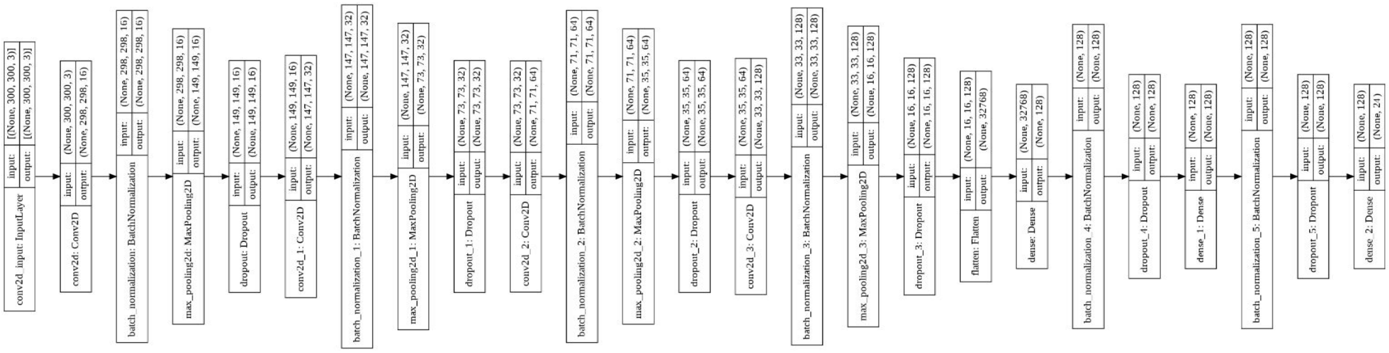

conv2d_input: InputLayer | input: [(None, 300, 300, 3)] | output: [(None, 300, 300, 3)]
conv2d: Conv2D | input: (None, 300, 300, 3) | output: (None, 298, 298, 16)
batch_normalization: BatchNormalization | input: (None, 298, 298, 16) | output: (None, 298, 298, 16)
max_pooling2d: MaxPooling2D | input: (None, 298, 298, 16) | output: (None, 149, 149, 16)
dropout: Dropout | input: (None, 149, 149, 16) | output: (None, 149, 149, 16)
conv2d_1: Conv2D | input: (None, 149, 149, 16) | output: (None, 147, 147, 32)
batch_normalization_1: BatchNormalization | input: (None, 147, 147, 32) | output: (None, 147, 147, 32)
max_pooling2d_1: MaxPooling2D | input: (None, 147, 147, 32) | output: (None, 73, 73, 32)
dropout_1: Dropout | input: (None, 73, 73, 32) | output: (None, 73, 73, 32)
conv2d_2: Conv2D | input: (None, 73, 73, 32) | output: (None, 71, 71, 64)
batch_normalization_2: BatchNormalization | input: (None, 71, 71, 64) | output: (None, 71, 71, 64)
max_pooling2d_2: MaxPooling2D | input: (None, 71, 71, 64) | output: (None, 35, 35, 64)
dropout_2: Dropout | input: (None, 35, 35, 64) | output: (None, 35, 35, 64)
conv2d_3: Conv2D | input: (None, 35, 35, 64) | output: (None, 33, 33, 128)
batch_normalization_3: BatchNormalization | input: (None, 33, 33, 128) | output: (None, 33, 33, 128)
max_pooling2d_3: MaxPooling2D | input: (None, 33, 33, 128) | output: (None, 16, 16, 128)
dropout_3: Dropout | input: (None, 16, 16, 128) | output: (None, 16, 16, 128)
flatten: Flatten | input: (None, 16, 16, 128) | output: (None, 32768)
dense: Dense | input: (None, 32768) | output: (None, 128)
batch_normalization_4: BatchNormalization | input: (None, 128) | output: (None, 128)
dropout_4: Dropout | input: (None, 128) | output: (None, 128)
dense_1: Dense | input: (None, 128) | output: (None, 128)
batch_normalization_5: BatchNormalization | input: (None, 128) | output: (None, 128)
dropout_5: Dropout | input: (None, 128) | output: (None, 128)
dense_2: Dense | input: (None, 128) | output: (None, 24)

Fig. 7 Block Architecture for proposed CNN model

Once the genres are predicted, we use cosine similarity to obtain top 10 recommendation. Using cosine similarity, we have found a relationship between the predicted and actual genres.

Cosine similarity gives a value between 0 to 1, where +1 indicates exactly similar and 0 indicates exactly different. The below diagram depicts how the proposed model works.

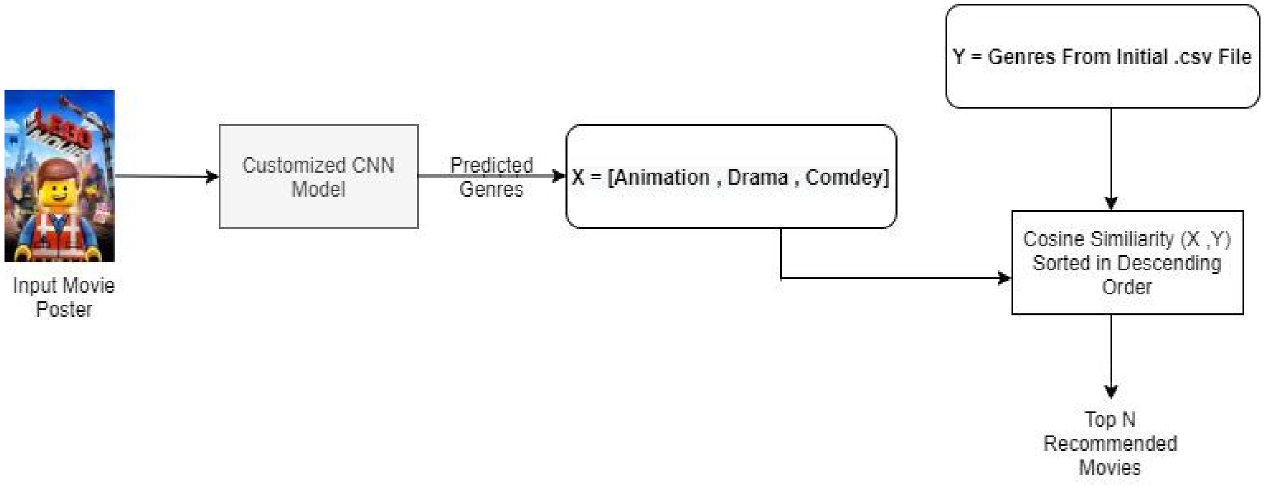

Y = Genres From Initial .csv File
Customized CNN Model
Predicted Genres
X = [Animation , Drama , Comdey]
Cosine Similiarity (X ,Y) Sorted in Descending Order
Input Movie Poster
Top N Recommended Movies

Fig. 8 Proposed Model

### D. *Evaluation Metrics*

Since prediction of genres falls under Multilabel classification, the evaluation is done differently than the multiclass classification. Evaluation metrics for multilabel classification is classified as Sample Based and Label Based [11].

*1)* Sample based (Sb) computes average difference between the actual labels and the predicted labels for each training sample, averaged over all the training samples in the dataset.

$$\text{Accuracy}_{\text{(Sb)}} = \frac{1}{n} \sum_{i=1}^{n} \frac{|y(i) \cap \hat{y}(i)|}{|y(i) \cup \hat{y}(i)|} \qquad (1)$$

$$\text{Precision}_{\text{(Sb)}} = \frac{1}{n} \sum_{i=1}^{n} \frac{|y(i) \cap \hat{y}(i)|}{|\hat{y}(i)|} \qquad (2)$$

where, y(i) is true label in ith sample and ŷ(i) is predicted label in ith sample.

*2)* Label based (Lb) is opposite of sample based, it computes each label separately and then average is taken over all labels.

$$\text{Accuracy}_{\text{(Lb)}} = \frac{1}{k} \sum_{l-1}^{k} \frac{\sum_{i-1}^{n} |y^{l}(i) \cap \hat{y}^{l}(i)|}{\sum_{i=1}^{n} |y^{l}(i) \cup \hat{y}^{l}(i)|} \qquad (3)$$

$$\text{Precision}_{\text{(Lb)}} = \frac{1}{k} \sum_{l-1}^{k} \frac{\sum_{i-1}^{n} |y^{l}(i) \cap \hat{y}^{l}(i)|}{\sum_{i=1}^{n} |\hat{y}^{l}(i)|} \qquad (4)$$

where, y(i) is true label in ith sample and lth label and ŷ(i) is predicted label in ith sample and lth label.

*3)* Hamming Loss is incorrectly predicted labels to the total number of labels.

## IV. RESULTS AND ANALYSIS

The below table shows the comparison of existing and proposed model evaluation metrics.

TABLE I
Comparison of evaluation metrics of existing model and our proposed model

| Model | Accuracy $_{\text{(Sb)}}$ | Precision $_{\text{(Sb)}}$ | Accuracy $_{\text{(Lb)}}$ | Precision $_{\text{(Lb)}}$ |
|---|---|---|---|---|
| Proposed Model | 0.4619 | 0.5623 | 0.3258 | 0.5692 |
| Baseline Model [12] | 0.4532 | 0.5398 | 0.3167 | 0.5645 |

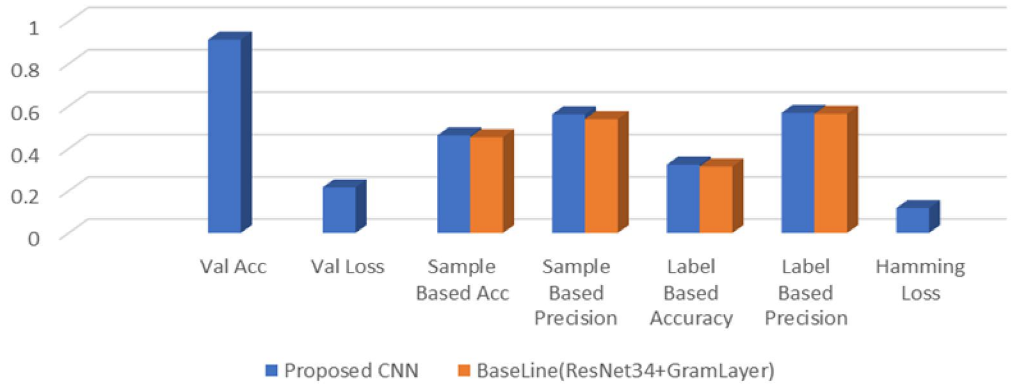


Fig. 9 Evaluation metrics comparison

The precision, recall and f1-score for each label is shown in below figure.

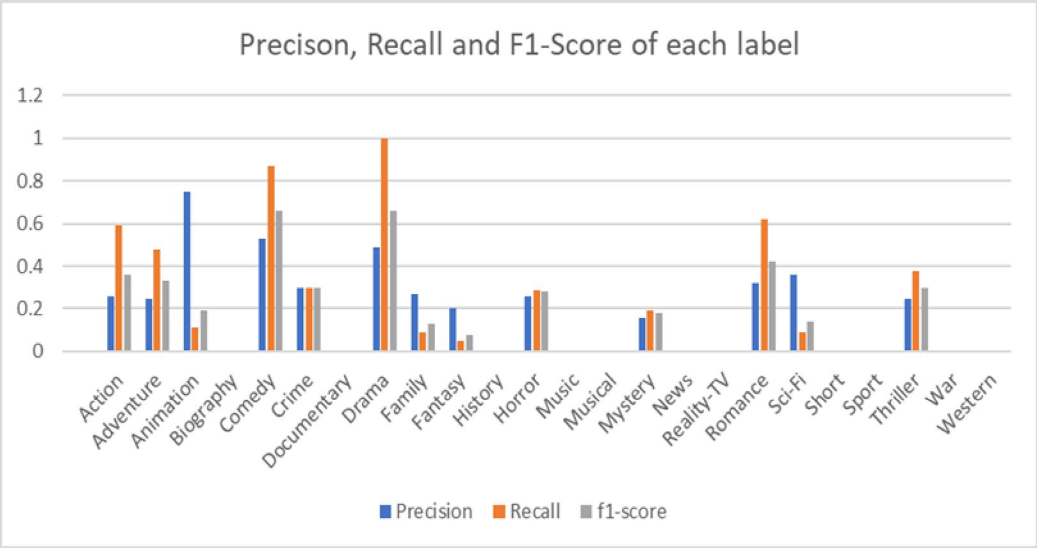


Fig. 10 Individual labels precision, recall and f1

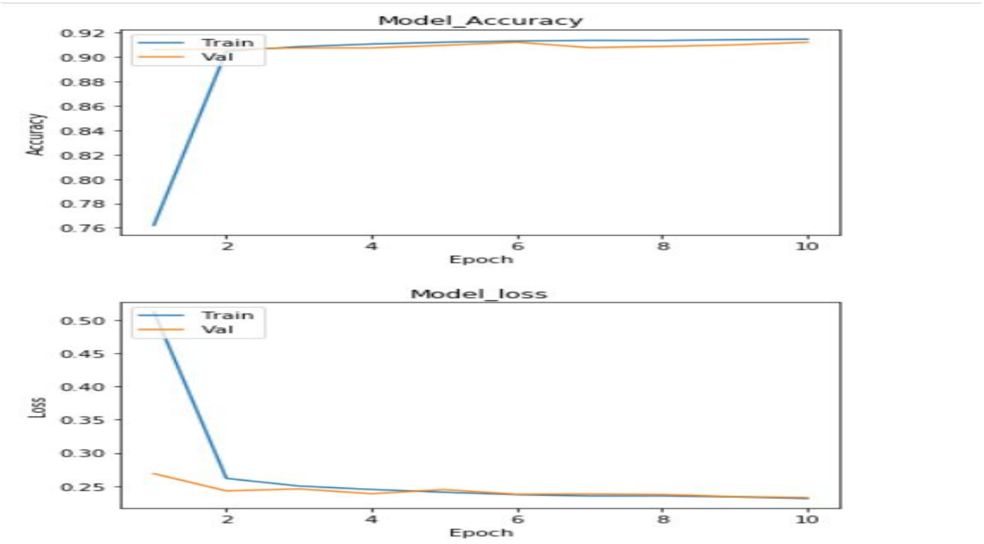
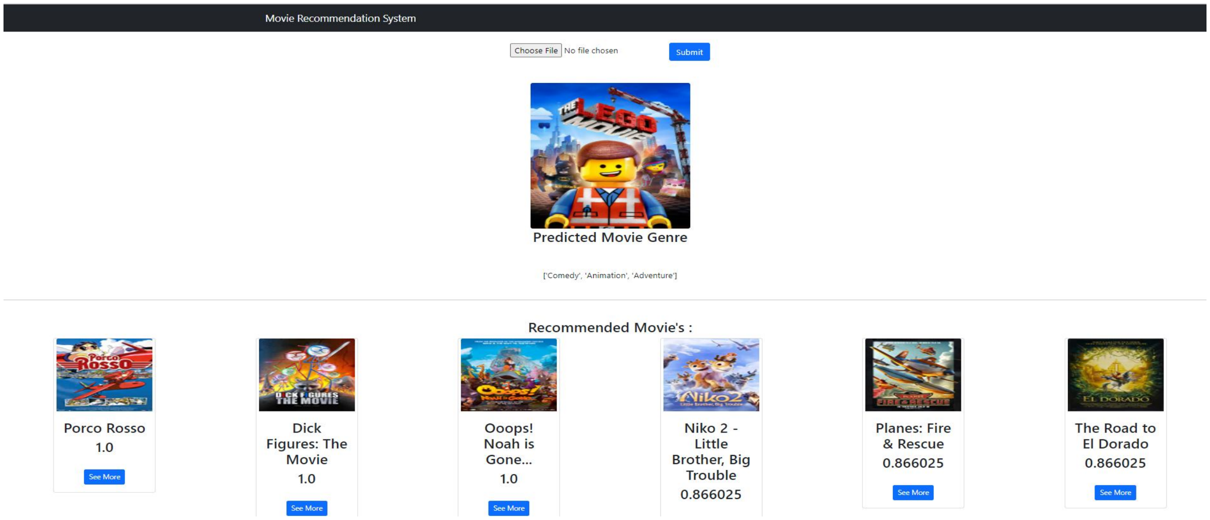


Fig. 11 Training and Validation accuracy and loss of proposed CNN model.



Movie Recommendation System

Choose File No file chosen

Predicted Movie Genre

['Comedy', 'Animation', 'Adventure']

Recommended Movie's :

Porco Rosso 1.0

Dick Figures: The Movie 1.0

Ooops! Noah is Gone... 1.0

Niko 2 - Little Brother, Big Trouble 0.866025

Planes: Fire & Rescue 0.866025

The Road to El Dorado 0.866025

Fig. 12 Final output screen of proposed model

## V. CONCLUSIONS

Movie posters gives an idea about movie content and its genres. Based on colors, objects, expressions of actors etc. one can easily determine the genre of movie. Humans are more or less able to predict genre of a movie only by looking at its poster. Therefore, we can say that the poster contains some characteristics which can be utilized in deep learning algorithms to predict its genre. In this paper, Deep Neural Network (Convolutional Neural Network) is built to classify a given movie poster image into its genres. Finally, based on these predicted genres, cosine similarity is calculated between actual and predicted genres to obtain list of recommended movies from dataset. Our proposed model is than compared with the baseline model and found that our proposed model performs 0.9% better than the baseline model in terms of accuracy and 0.4% in terms of precision.

## VI. FUTURE SCOPE

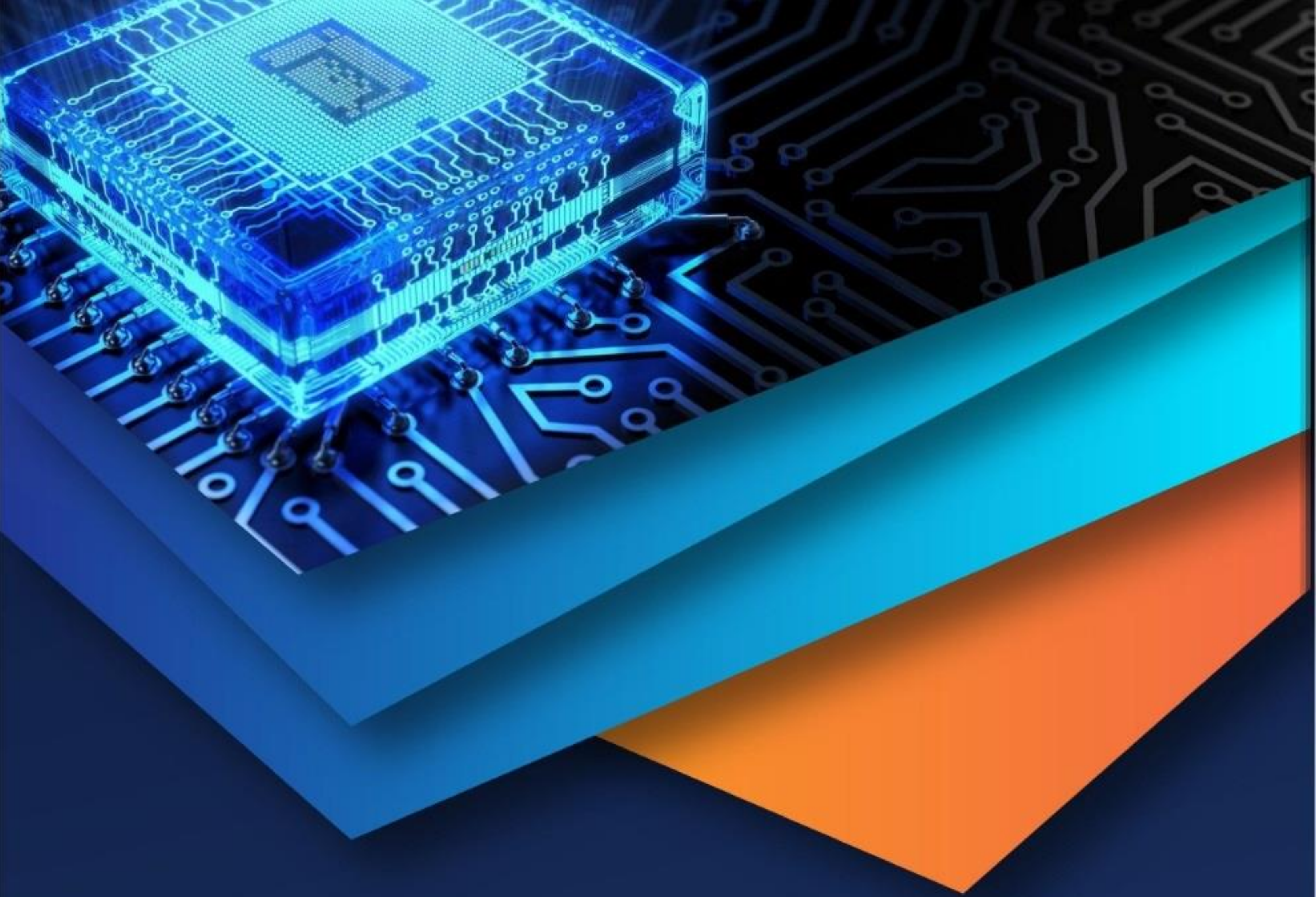
The movie posters are taken from IMDB websites using the IMDB id and IMDB link. But there are many movies which are very old and not present on the website. Due to this the dataset became imbalance which can be handle by using various resampling techniques. This model can also be tried on some good balance datasets. One can also use Matrix factorization techniques like SVD for recommendation model.

## VII. ACKNOWLEDGMENT

We acknowledge the industry mentor, Mrs. Kirti Sharma for the guidance and suggestions provided during the entire research process.

## REFERENCES

[1] https://towardsdatascience.com/how-to-build from-scratch-a-content-based-movie recommender-with-natural-language processing-25ad400eb243
[2] https://towardsdatascience.com/how-to-build-from-scratch-a-content-based-movie recommender-with-natural-language-processing-25ad400eb243
[3] Bam Bahadur Sinha, R. Dhanalakshmi, Ramchandra Regmi (2020), "TimeFly algorithm: a novel behavior-inspired movie recommendation paradigm", Pattern Analysis and Applications, https://doi.org/10.1007/s10044-020-00883-8
[4] Xiaoyue Li, Haonan Zhao, Zhuo Wang and Zhezhou Yu (2020), "Research on Movie Rating Prediction Algorithms", 2020 5th IEEE International Conference on Big Data Analytics, 978-1-7281-4111-4/20/$31.00
[5] Tan nghia duong, Tuan anh vuong, Duc minh nguyen, Quang hieu dang (2020)," Utilizing an Autoencoder-Generated Item Representation in Hybrid Recommendation System", 10.1109/ACCESS.2020.2989408
[6] Gaojun Liu, Xingyu Wu1 (2019), "Using Collaborative Filtering Algorithms Combined with Doc2Vec for Movie Recommendation", 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC 2019).
[7] Yeo Chan Yoon, Jun Woo Lee (2018), "Movie Recommendation using Metadata based Word2Vec Algorithm", International Conference on Platform Technology and Service
[8] https://www.kaggle.com/rounakbanik/the-movies-dataset?select=links.csv
[9] https://www.imdb.com/
[10] https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac#:~:text=There%20are%20three%20types%20of,task%20on%20the%20input%20data.
[11] https://medium.datadriveninvestor.com/a-survey-of-evaluation-metrics-for-multilabel-classification-bb16e8cd41cd
[12] Jeong A. Wi, Soojin Jang, Youngbin Kim(2020),"Poster-Based Multiple Movie Genre Classification Using Inter-Channel Features", IEEE Access, vol 8, 10.1109/ACCESS.2020.2986055



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)