Predicting Movie Popularity and Ratings with Visual Features

Farshad B. Moghaddam
Karlsruhe Institute of Technology
Karlsruhe, Germany
farshad.moghaddam@student.kit.edu

Mehdi Elahi
Free University of Bozen-Bolzano
Bozen - Bolzano, Italy
meelahi@unibz.it

Reza Hosseini
Vaillant Business Services
Remscheid, Germany
reza.hosseini82@gmail.com

Christoph Trattner
University of Bergen
Bergen, Norway
christoph.trattner@uib.no

Marko Tkalčič
Free University of Bozen-Bolzano
Bozen - Bolzano, Italy
marko.tkalcic@unibz.it

***Abstract*—In Movie Recommender Systems, when a new user registers to the system and she has not yet provided any information about her, the system may not be able to generate personalized recommendations for that user. In such a *Cold Start* situation, many real-world recommender systems suggest popular movies to the new user. Such movies are very likely to be interesting to the new users. A very common approach for measuring the movie popularity is based on counting the number of ratings (as user votes) provided by a community of the existing users. However, in certain cases, we cannot properly measure the popularity of the movies with this common approach.**

This paper proposes a novel method for predicting the popularity of movies. The method is based on hybrid visually-driven features, representative of the movie content, which can be used to effectively predict not only the movie popularity but also the average rating of the movie. Our extensive experiments on a large dataset of more than 13'000 movies trailers show that the proposed hybrid approach achieves promising results by exploiting visual *Attractiveness* features of movies in comparison to the other baseline features.

I. Introduction

One of the main challenges in Movie Recommender Systems research is the so-called *Cold Start* Problem [5]. This problem happens when a new user registers to the system and requests recommendations before providing any rating (*New User* problem), or when a new item is added to the catalogue of the system and has not received any rating from the users (*New Item* problem) [15]. One of the typical solutions for the new user problem is to present the popular items to them, allowing the system to obtain some initial information about the user that can be used then to generate more personalized recommendations [13]. Popular movies are very likely to be interesting to the majority of users and hence to serve them in the initial stage of their interaction with a recommender system.

The movie popularity can be computed in different ways. The most common method in recommender systems community, is to use the number of ratings (or the number of user votes) provided by users for movies and consider it as an indicator of the popularity of movies. While this could be a powerful indicator of the popularity, it may fail to function when the movies are new. Indeed, this indicator requires the system to access a large network of users who have already provided considerable number of ratings to the movies.

In addition to the research area of recommender systems, the task of *popularity prediction*, has been also studied in the other research areas and in the various contexts [6], including social media analysis [1], online news applications [22], and multimedia retrieval [17]. Each of these related works have attempted to predict the popularity based on the various forms of available data. However, they mostly rely on the social media data in resolving the task [24].

Our work substantially differs from the related work. First of all, none of these methods are able to predict the popularity of a movie that has not been even screened. However, our method is capable of predicting the popularity of the movies even before they are publicly shown. Moreover, in contrast to these methods, we do not rely on any social media activity of users, or we do not even investigate *high-level* features such as star actors in a movie. Instead, we investigate the *low-level* features based on internal (visual) characteristics of the movies without relying on external data from social networks. Moreover, in contrast to our approach, they have conducted experiments on small datasets (e.g., 312 movies in [14]), which may not necessarily result in conclusive outcomes.

In particular, this paper proposes a novel technique for analyzing the visual aspects of movies and building predictive models capable of estimating the popularity of the movies. Our technique is based on a recently proposed feature extraction and engineering mechanism that have been shown as a promising descriptive power of images [4]. Our technique demonstrate that extracting the visual features and providing them as an input to the state-of-the-art machine learning algorithm (i.e., Gradient Boosting) can result in considerable quality of popularity prediction task.

978-1-7281-3634-9/19$31.00 © 2019 IEEE

Accordingly, we have formulated the following research questions:

- **RQ1:** To what extent are visual appeal features capable of predicting movie popularity?
- **RQ2:** To what extent are they able to predict the average rating of a movie?

We have extracted visual features from a large dataset of more than 13'000 movie trailers [1], according to the methodology proposed in [3], [26], [27]. These movie trailers have shown to be visually similar to their corresponding full-length movies [3]. We have conducted a comprehensive set of experiments (including the exploratory study and predictive analysis) with several visual features adopting the state-of-the-art machine learning algorithms. The results have shown the substantial power of visual features in predicting the popularity of movies, as well as their average ratings, provided by users.

II. Related Work

Various works have been conducted in the context of studying the effect of popularity of digital content for different applications. [2], [7], [9], [10], [14], [19], [20], [23], [25]. Szabo in [21] proposed predictive models, e.g., based on linear regression to predict the popularity of a video based on the number of previous views in YouTube. Pinto in [17] proposed a Multivariate Linear model that extends the [21] model by sampling the number of views at regular time slots. They showed that their model outperforms [21] model.

Beside the noted works, others attempted to study the financial success of movies in terms of relation of popularity and the box-office revenue. Authors in [14] investigated the relation of logged activities of Wikipedia users on a movie page with the box-office success. They showed that by analyzing *#views, #users,* and *#edits* as well as *Collaborative rigor* [12], it is possible to make an estimation of the box-office revenue. Sharda in [19] used neural networks in order to predict the box-office. They divided the movies to several categories (ranging from *flop* to *blockbuster*) and converted box-office prediction problem to a classification problem. They used features such as genre and showed that such features can be used to effectively predict the box-office.

Very limited works (e.g., [7], [25]) have also exploited visual features together with social networks activities to predict the number of views of an online video from YouTube and Facebook. They used models such as Support Vector Regression and showed that these type of models can predict the video popularity.

In contrast to these works, our proposed approach differs in the following aspects: (i) First, the majority of prior works attempt to predict the popularity (in terms of #views) of a user-generated content in video sharing web applications such as YouTube. Hence their proposed methods highly rely on availability of data from the social activities of users as well as social connections of the movie publisher (e.g., friend list or follower network). In contrast, in this work, we do not make such an assumption and instead analyze only the existing video content, automatically. (ii) Second, most of the prior works require a movie to be published publicly. However, although our method is also applicable in that scenario, we mainly focus on cinema industry (e.g., Hollywood scale). The cinema movies (may also be called *feature films*) are produced with multi-million dollar budgets and a failure could cause huge financial damages, compared to low-cost user-generated videos for social networks. Accordingly, in our considered scenario, the popularity prediction has to be made before a movie is screened, allowing the movie production companies to get an estimation on the target factors such as popularity, user rating and box-office. This will allow them to take proactive decisions, i.e., launching advertisement campaigns. (iii) Finally, almost all of these works have evaluated their methods on very small datasets of few hundred movies (e.g., 312 movies [14]), while we have used a large dataset of more than 13'000 movie trailers.

[1] the dataset will soon be published publicly, as a supplementary material of this paper.

III. Methodology

We have used a large dataset, containing 13053 movie trailers [3] that had their titles available in the Movielens dataset [8]. Prior work has reported large values of similarity, based on visual features, extracted from movie trailers and their corresponding full-length movies [3]. For each movie, we also collected the meta-data such as #ratings, average rating, genre, and the year of production from IMDB. Every movie can have one or multiple genre label(s) out of 30 possible genres (e.g., drama, comedy, romance, etc.). The following list represents the entire methodology (see Figure 1):

1) **Splitting Movies**: every movie is split into shots, i.e., sequences of consecutive frames captured without interruption of the movie camera;
2) **Identifying Key-Frames**: within every shot a frame is selected as representative of the shot (called Key-frame);
3) **Extracting Features**: every key-frame is analyzed and the visual features are extracted;
4) **Aggregating Features**: the features are then aggregated over the whole movie to build an individual feature vector descriptive of the movie;
5) **Training and Predicting**: the visual feature vectors are exploited in order to train the prediction algorithms.

Splitting Movies & Identifying Key-frame. We have exploited a technique based on *Color Histogram Distance*, which splits the movies into shots, i.e., sequences of consecutive frames captured without interruption of the movie camera (see Figure 1). In fact, the transition between shots of a movie is typically very large. By comparing the color histogram of each frame, the histogram intersection is measured in order to compare the activities. Lets denote h_t and h_{t+1} as histograms of successive frames, then intersection is calculated based on the following:

$$\mathbf{s}(h_t, h_{t+1}) = \sum_b \min(h_t(b), h_{t+1}(b)) \qquad (1)$$

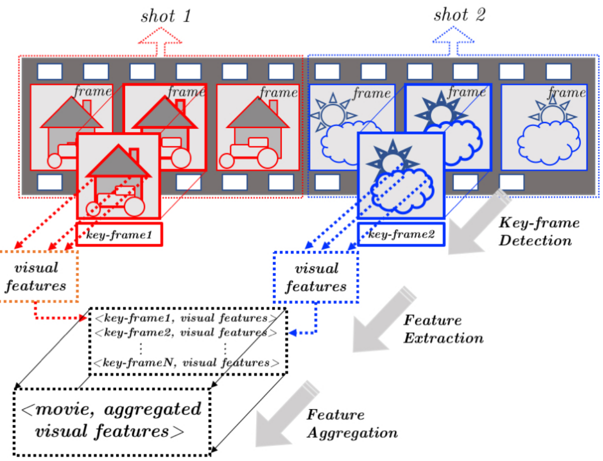

Fig. 1. The method of feature extraction and aggregation

where b is the index of the histogram bin. By comparing s with a threshold, we can indeed split the movies into the building blocks of shots. Within each, a frame is selected as a representative frame (key-frame).

Visual Feature Extraction. We have extracted a set of visual features capable of effectively capturing the *attractiveness* of every (key) frame of the movies. San Pedro and Siersdorfer [18] used a similar set features to predict popularity of Flicker images. In detail, the features are the following:

- ***f1:Sharpness*** measures the clarity and level of details within the elements of a frame. This feature is related to the brightness contrast of edges in a frame. The algorithm utilizes the *image* Laplacian, divided by the average luminance (μ_{xy}) around pixel (x,y):

$$frame\_sharpness = \sum_{x,y} \frac{L(x,y)}{\mu_{xy}}, \text{ with } L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (2)$$

- ***f2:Sharpness Variation*** is calculated via the standard deviation of all pixel sharpness values.
- ***f3:Contrast*** measures the relative difference in brightness or color of local features in a frame. The root mean square contrast (RMS-contrast) is often used to compare frames [18]:

$$frame\_contrast = \frac{1}{N} \sum_{x,y} (I_{xy} - \overline{I}) \quad (3)$$

where I_{xy} is the intensity of a pixel, $\overline{I}$ represents the arithmetic mean of the pixel intensity and N is the number of pixels.
- ***f4:RGB Contrast*** is almost identical to the basic contrast feature, explained before. However, it is extended to the three-dimensional RGB color space.
- ***f5:Saturation*** measures the colorfulness of the frame relative to the brightness. In the HSV color space the saturation estimation can be calculated via the RGB approximation of

$$frame\_saturation = \frac{1}{N} \sum_{x,y} S_{xy}, \text{ with} \quad (4)$$

$$S_{xy} = max(R_{xy}, G_{xy}, B_{xy}) - min(R_{xy}, G_{xy}, B_{xy})$$

where N is the amount of pixels in a frame and R_{xy}, G_{xy} and B_{xy} are the coordinates of the color of the pixel in sRGB space.
- ***f6:Saturation variation*** measures the variation in saturation via the sample standard deviation of all pixel saturation in a frame.
- ***f7:Brightness*** measures the average brightness of a frame; It uses a standard luminance algorithm

$$frame\_brightness = \frac{1}{N} \sum_{x,y} Y_{xy}, \text{ with} \quad (5)$$

$$Y_{xy} = (0.299 * R_{xy} + 0.587 * G_{xy} + 0.114 * B_{xy})$$

where Y_{xy} denotes the luminance value and N is the amount of pixels in a frame. R_{xy}, G_{xy} and B_{xy} are the three RGB color space channels of pixel(x,y).
- ***f8:Colorfulness*** measures the individual color distance of the pixels in a frame. Therefore, the frame needs to be transferred in to sRGB color space using $rg = R - G$ and $yb = 1/2\ (R + G) - B$ and subsequently, colorfulness can be measured as

$$frame\_colorfulness = \sigma_{rgyb} + 0.3 \cdot \mu_{rgyb}, \text{ with} \quad (6)$$

$$\sigma_{rgyb} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2},\ \mu_{rgyb} = \sqrt{\mu_{rg}^2 + \mu_{yb}^2}$$

where R, G and B are the color channels of the pixels and σ is the standard deviation, respectively μ the arithmetic mean.
- ***f9:Entropy*** of a frame is often used to determine how much information needs to be encoded by a compression algorithm. As an example, a frame with illustrating the moon craters has a very high edge contrast, which leads to a high entropy. This means that the frame cannot be compressed very well which suggests that it can be used to measure the frame's texture. We used *Shannon Entropy* as follows: we converted the frame to grey scale, where each pixel has only an intensity value. Then, we count the occurrences of each distinct value. Finally, we apply the following formula:

$$frame\_entropy = - \sum_{x \in [0..255]} p(x) \cdot log_2(x) \quad (7)$$

where p_x is the probability of finding the gray-scale value x among all the pixels in the frame.
- ***f10:Naturalness*** measures the difference (or similarity) between a frame and the human visual perception of the real world, with respect to colorfulness and dynamic range. Although subjective, it is an important visual quality metric when it comes to design [18]. We transfer the frame color space, if not already, to HSL. Then we use only pixels within the thresholds $20 \le L \le 80$ and $S \ge 0.1$. In the next step, pixels are grouped in to one of the three sets 'Skin','Grass' or 'Sky', based on their H coordinate (hue). In order to calculate the naturalness of

each set, the average saturation value of the group (μ_S) is used:

$$N_{Skin} = e^{-0.5(\frac{\mu_S^{Skin}-0.76}{0.52})^2},\ if\ 25 \leqslant \text{hue} \leqslant 70 \quad (8)$$

$$N_{Grass} = e^{-0.5(\frac{\mu_S^{Grass}-0.81}{0.53})^2},\ if\ 95 \leqslant \text{hue} \leqslant 135 \quad (9)$$

$$N_{Sky} = e^{-0.5(\frac{\mu_S^{Sky}-0.43}{0.22})^2},\ if\ 185 \leqslant \text{hue} \leqslant 260 \quad (10)$$

In the final step, the naturalness index can be calculated using

$$frame\_naturalness = \sum_i \omega_i N_i,\ i \in \{\text{'Skin'},\text{'Grass'},\text{'Sky'}\} \quad (11)$$

where ω_i represents the fraction of pixels of the specific group in the whole frame. Naturalness ranges from 0 (a unnatural frame) to 1 (a natural frame).

Feature Aggregation. To form the feature vector description of a movie, we have aggregated the visual features extracted from its key-frames. We have performed various aggregation functions. First of all, in order to model the ordinal nature of the subsequent key-frames of a movie, we have fitted *5th degree polynomial* to each of the visual features and used the coefficients as aggregated features. We have also computed *minimum, maximum, mean, standard deviation, median,* as well as, *1st & 3rd quartiles* of each visual feature, across all the key-frames of a movie. The last aggregated movie feature is the *number of key-frames*, within every movie. This process has resulted in a vector of the length of *121 aggregated features* per movie. These are features that are all extracted automatically and hence we refer to them as *automatic*. There are features that need *manual* labeling (e.g., by experts). We considered genre and year of production, since we may know them before the movies are screened.

In the data extracted from IMDB, genre comes in the form of comma-separated list of string for every movie. Each movie has up to three genres, with majority of them (1263) having three genres. For converting them to features, we used vectorization. i.e. we have 28 features for all the genres and when a movie belong to a genre, the corresponding feature is 1 otherwise it is 0. Finally, we merged both types of automatic and manual features and formed *hybridized* visual features. We refer to this extended vector as *hybrid* visual features.

Learning Algorithms & Evaluation Protocol. As learning method, we used the state-of-the-art machine learning algorithm *Gradient Boosting* (Microsoft LightGBM [11]) and set the number of trees to 300. We have also repeated the experiments with further machine learning algorithms but obtained similar results. The extension of our results together with the extracted dataset and the implementation details will be provided as a supplementary material, in the form of a technical report. These additional algorithms were *Random Forest* (with 300 trees), *Logistic Regression* and *Regression Tree*[2]. Furthermore, we implemented additional basic baselines, i.e., prediction based on (i) *distribution* of the train set, (ii) *mean* of the train set, and (iii) randomness.

[2]we adopted an implementation of these algorithms in *Scikit-learn* package [16].

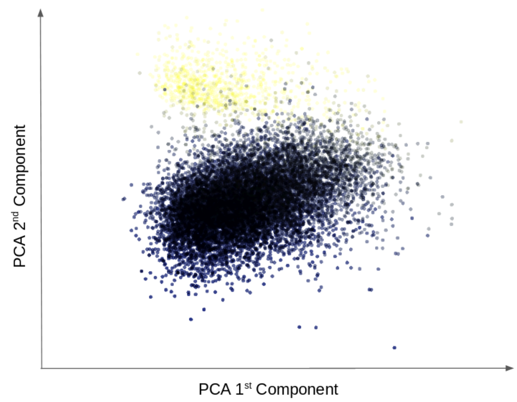

Fig. 2. Visualization of the Principle Component Analysis (PCA) applied on the most important visual features.

As evaluation protocol we employed *k-fold cross validation* by randomly splitting the dataset into 7 non-overlapping subsets where in every iteration, $\frac{6}{7}$ of instances are used as training the models and the rest $\frac{1}{7}$ for testing. We have measured different evaluation metrics, i.e., *Accuracy*, *F1-score*, *RMSE*, and *MAE*. We have also measured Precision and Recall metrics. But due to the space limitation and conformity of results with F1-score, we have not reported them in this paper.

Moreover, we have categorized the number of ratings, provided by IMDB users, into 3 classes: *Popular* (#ratings above 200'000, total of 1050 movies), *Mid-popular* (#ratings 25'000[3] - 200'000, 1220 movies), *Unpopular* (#ratings below 25'000, 10900 movies) movies. This is a more challenging classification task compared to considering only 2 classes (popular vs unpopular).

IV. Results

A. Experiment A: Exploratory Analysis

We initially performed exploratory analysis in order to grasp a better understanding of the data. The observed results are presented in this section (experiment A).

Principle Component Analysis (PCA). In the first experiment, we were interested in visualizing the data by reducing the large feature dimensions and making a 2D plot of the data. We have used a well-known dimensionality reduction method called Principle Component Analysis (PCA). Figure 2 represents the results. As it can be seen, the data can be reduced and represented by 2 main principle components (x-axis and y-axis in the figure). Accordingly, we could visually identify two clusters of the movies, plotted with yellow and black colors. We conjecture that these clusters may represent the older (perhaps less popular) movies and the newer (perhaps more popular) movies. Some manual checks gave indication of our conjecture. However, further in-depth analysis is necessary

[3]as suggested here: https://www.imdb.com/chart/top

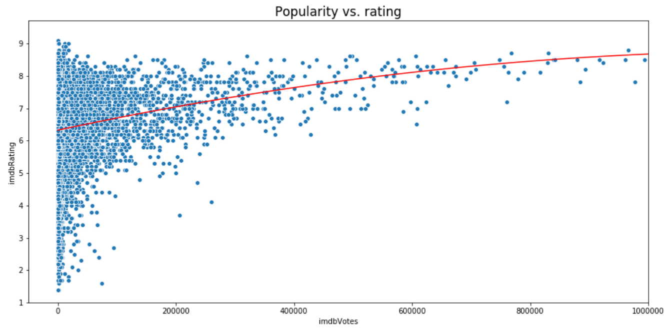

Fig. 3. Popularity of the movies vs the average ratings in IMDB.

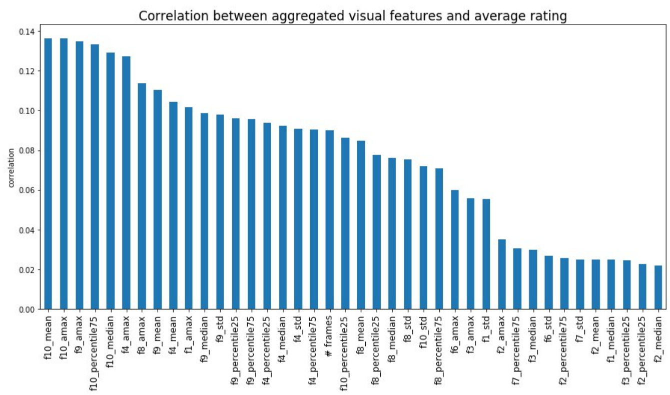

Fig. 4. Pearson Correlation between visual features and average ratings of movies. Please note that we focused on the magnitude and hence plotted the absolute values.

for confirming our early observation. This is indeed planned as a future work.

Popularity vs Average Rating. In this paper, we targeted two important variable originated from different characteristics of the movies, i.e., popularity and average ratings. Hence, we were interested in exploring the potential correlations between these two variables. Figure 3 illustrates the correlations between popularity and average ratings in IMDB. As it can be seen, there is a positive correlation between these two target variables. Accordingly, as the popularity of a movie is boosted and the movie receives higher number of ratings, the average ratings provided by the users increases. This is an expected phenomenon as the users typically rate what they like. However, it is also observed that there are considerable number of movies with high average rating that have not received large number of ratings.

Correlation of Visual Features with Average Rating. We have also computed the correlation between the visual features and the average ratings of movies. We were more interested in the magnitude (regardless of the sign) and hence we focused on the absolute values. The results are plotted in Figure 4. As it can be expected, there are features with higher correlations and features with lower correlations. The results show that the most correlated and hence the most important feature is f_{10} (Naturalness) where its mean, maximum, 3rd percentile, and median are among top features. The next correlated features are f_9 (Entropy of the frame) and f_4 (RGB Contrast), maximum of both being also among top features.

TABLE I
COMPARISON OF DIFFERENT FEATURES FOR PREDICTING THE IMDB NUMBER OF RATINGS BASED ON VISUAL FEATURES.

Features	Feature Extraction	F1	Accuracy
Hybrid Visual	*hybrid*	**0.460**	0.568
Visual	*automatic*	0.448	**0.569**
Genre	*manual*	0.455	0.563
Year	*manual*	0.404	0.492
Distribution	-	0.421	0.419
Random	-	0.338	0.358

We believe that the top visual features can be better predictive of the average rating values of the movies. Furthermore, a feature engineering can be very beneficial in selecting the best features for the task of average rating prediction. Although the adopted predictive model (Gradient Boosting) implements a feature selection mechanism, however, again, a more in-depth analysis can result in improvement of our understanding of the data.

B. Experiment B: Predictive Analysis

In the second experiment, we built a predictive model based on the state-of-the-art machine learning algorithm, i.e., Gradient Boosting, in order to address the formulated research questions (i.e., RQ1 and RQ2).

RQ1: Predicting Movie Popularity. Table I presents the results of using different features for predicting different classes of the number of IMDB ratings. As shown, in terms of F1 metric, the best results were achieved by using hybrid visual features. Accordingly, training the classifier on the hybrid visual features obtained F1 score of 0.460. In terms of accuracy metric, visual and hybrid visual features achieved very similar results, i.e., the accuracy of 0.568 and 0.569, respectively.

RQ2: Predicting the Movie's Average Rating. Table II presents the results obtained by training the classification algorithm on different types of features in order to predict IMDB average rating. As the results show, hybrid visual features overtake all the other features by obtaining the lowest RMSE and MAE values 0.938 and 0.716, respectively. The genre features can also achieve a very good results, i.e. RMSE and MAE values 0.958 and 0.733, which is interesting. For this specific task, the visual features alone do not perform as good as hybrid visual features.

Overall, the results of these experiments confirm that using a state-of-the-art classification model, trained on hybrid visual features, can be used to predict the popularity and ratings of the movies even before the movies are publicly screened. This is a promising outcome and it presents the fantastic potential of the visual features in the movie industry.

TABLE II
COMPARISON OF DIFFERENT ML ALGORITHMS FOR PREDICTING THE IMDB AVERAGE RATING BASED ON VISUAL FEATURES.

Feature	**Feature Extraction**	**RMSE**	**MAE**
Hybrid Visual	*hybrid*	**0.938**	**0.716**
Visual	*automatic*	1.074	0.829
Genre	*manual*	0.958	0.733
Year	*manual*	1.090	0.845
Mean	-	1.092	0.846
Random	-	2.760	2.280

V. CONCLUSION

In this paper we addressed two research questions and investigated the utility of visual features to predict the popularity (RQ1), and average ratings (RQ2). We have evaluated our proposed method by conducting a set of extensive experiments on a large dataset of more than 13'000 movie trailers. The results of our experiments have successfully answered the research questions and our *hybrid* visual features are capable of predicting the popularity, and average rating of the movies better than other baseline methods.

While the presented results are preliminary, but still, as a proof of concept they are promising and hence we are working on an extension of this work. First, we plan to investigate different kinds of methods for feature aggregation and fusion. Moreover, we will employ deep learning methods to extract image embeddings (e.g., ResNet) as they have proven to be very reliable in preforming the image classification. Finally, we will attempt to exploit these features for a different task, i.e., content-based recommendation, adopting a framework [4] that has shown to be well-compatible with visual features.

REFERENCES

[1] Peng Bao, Huawei Shen, Junming Huang, and Xueqi Cheng, "Popularity prediction in microblogging network: a case study on sina weibo", 22nd International World Wide Web Conference, WWW 13, Rio de Janeiro, Brazil, May 13-17,2013, Companion Volume. 177178. https://doi.org/10.1145/2487788.2487877

[2] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems", IEEE/ACM Transactions on networking 17, 5 (2009), 13571370.

[3] Yashar Deldjoo, Mehdi Elahi, Paolo Cremonesi, Franca Garzotto, Pietro Piazzolla, and Massimo Quadrana. "Content-based video recommendation system based on stylistic visual features", Journal on Data Semantics 5, 2 (2016), 99113

[4] Vicente Dominguez, Pablo Messina, Denis Parra, Domingo Mery, Christoph Trattner, and Alvaro Soto. "Comparing Neural and Attractiveness-based Visual Features for Artwork Recommendation", Proc. of the DLRS. ACM, 5559.

[5] Mehdi Elahi, Matthias Braunhofer, Tural Gurbanov, and Francesco Ricci, "User Preference Elicitation, Rating Sparsity and Cold Start", Chapter Chapter 8, 253294.

[6] Flavio Figueiredo. "On the prediction of popularity of trends and hits for user generated videos", Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013. 741746. https://doi.org/10.1145/2433396.2433489

[7] Giulia Fontanini, Marco Bertini, and Alberto Del Bimbo,"Web Video Popularity Prediction using Sentiment and Content Visual Features", ICMR, John R. Kender, John R. Smith, Jiebo Luo, Susanne Boll, and Winston H. Hsu (Eds.).

[8] F Maxwell Harper and Joseph A Konstan, "The movielens datasets: History and context", ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4 (2016)

[9] F Maxwell Harper, Funing Xu, Harmanpreet Kaur, Kyle Condiff, Shuo Chang, and Loren Terveen, "Putting users in control of their recommendations", Proceedings of the 9th ACM Conference on Recommender Systems. ACM, 310

[10] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma, "Correcting Popularity Bias by Enhancing Recommendation Neutrality", RecSys 2014 Posters.

[11] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree", Advances in Neural Information Processing Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 31463154.

[12] Royce Kimmons. "Understanding collaboration in Wikipedia", First Monday 16, 12 (2011).

[13] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. "Facing the cold start problem in recommender systems", Expert Syst. Appl. 41, 4 (2014), 20652073.

[14] Marton Mestyan, Taha Yasseri, and Janos Kertesz. "Early Prediction of Movie Box Office Success based on Wikipedia Activity Big Data", CoRR (2012)

[15] Farshad Bakhshandegan Moghaddam and Mehdi Elahi, "Cold Start Solutions For Recommendation Systems", Chapter 2, book title: Big Data Recommender System

[16] Fabian Pedregosa, Gal Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. "Scikit-learn: Machine learning in Python", Journal of machine learning research 12, Oct (2011), 28252830.

[17] Henrique Pinto, Jussara M. Almeida, and Marcos Andr Gonalves. "Using early view patterns to predict the popularity of youtube videos", Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013. 365374. https://doi.org/10.1145/2433396.2433443

[18] Jose San Pedro and Stefan Siersdorfer. "Ranking and Classifying Attractiveness of Photos in Folksonomies", Proc. of WWW09. ACM, NY, USA.

[19] Ramesh Sharda and Dursun Delen, "Predicting box-office success of motion pictures with neural networks", Expert Systems with Applications 30, 2 (2006).

[20] Harald Steck, "Item popularity and recommendation accuracy", Proceedings of the fifth ACM conference on Recommender systems, 2011. ACM, 125132.

[21] G. Szabo and B.A. Huberman, "Predicting the popularity of online content", Commun. ACM 53, 8 (2010), 8088.

[22] Alexandru-Florin Tatar, Panayotis Antoniadis, Marcelo Dias de Amorim, and Serge Fdida. "From popularity prediction to ranking online news", Social Netw. Analys. Mining 4, 1 (2014), 174. https://doi.org/10.1007/s13278-014-0174-8

[23] Alexandru Tatar, Marcelo Dias de Amorim, Serge Fdida, and Panayotis Antoniadis, "A survey on predicting the popularity of web content", Journal of Internet Services and Applications 5, 1 (2014), 8

[24] Christoph Trattner, Dominik Moesslang, and David Elsweiler, "On the predictability of the popularity of online recipes", EPJ Data Sci. 7, 1 (2018), 20. https://doi.org/10.1140/epjds/s13688-018-0149-5

[25] Tomasz Trzcinski and Przemyslaw Rokita. "Predicting popularity of online videos using Support Vector Regression", CoRR abs/1510.06223 (2015).

[26] Deldjoo, Yashar and Elahi, Mehdi and Quadrana, Massimo and Cremonesi, Paolo, Using visual features based on MPEG-7 and deep learning for movie recommendation, International Journal of Multimedia Information Retrieval, volume 7, number 4, pages 207–219, 2018, Springer

[27] Elahi, Mehdi and Deldjoo, Yashar and Bakhshandegan Moghaddam, Farshad and Cella, Leonardo and Cereda, Stefano and Cremonesi, Paolo, Exploring the semantic gap for movie recommendations, Proceedings of the Eleventh ACM Conference on Recommender Systems, 326–330, 2017 ACM

[4] *Hi-Rec* framework, available at: https://fmoghaddam.github.io/Hi-Rec/