Poster Time Machine: Predicting Movie Release Order from Posters

Rick Hanish
Department of Electrical and Computer Engineering
Rice University
Houston, Texas
rh103@rice.edu

***Abstract*—It is qualitatively clear that movie posters have evolved over time as general design trends have changed. To evaluate the rate at which such trends have developed in movie posters, a deep convolutional network was trained to predict which of two input movie posters corresponds to a movie released later. After gathering a dataset from IMDb, a pre-trained ResNet model extracted features from the movie posters, the feature vectors of both posters were concatenated, then a small linear network decoded those features into its prediction. After training and testing, patterns in accuracy between two years' movie posters were analyzed as a proxy for the amount of development in design trends between those years. This work is entirely novel, as no other literature attempting the same problem was found during a literature review, although other work has attempted dating photographs instead of movie posters, and plenty of work has been done to attempt to classify a movie's genre based on its poster. Results showed that the model achieves 86.3% accuracy on the test set, with accuracy decreasing as the two years get closer together, as expected. If the release of two movies is at least 18 years separated, then the model classifies the order in which they were released with over 95% accuracy, and with over 90% accuracy when separated by just 6 years. Landmark improvements in technologies associated with creating the posters did not result in noticeable trends within the model's results. Future work may improve on these results, most notably by using more computational power.**

I. Introduction

Movies have existed for over a century, and with them come movie posters. Although artistic styles have changed throughout history, the premise and point of movie posters have remained the same: To advertise the associated movie. Therefore, it is reasonable to compare movie posters from different time periods. We can see in Fig. 1 that movie posters have changed drastically between decades. For example, it is highly unlikely that the above poster for *Psycho* (1960) would appear for a movie released today due to a difference in popular artistic styles and methods between 1960 and 2024.

After introducing the problem, prior work, and the data set in this section, all models attempted and the final model used will be discussed in Section II. Results from the model are included in Section III, and the results are analyzed and discussed in Section IV. Finally, Section **??** includes a link to the code for this project.

A. Problem Statement and Motivation

It can be hypothesized that the style of movie posters at any given time provides insight into popular art styles as a whole from that time. This work aims to better understand the uniqueness of styles from across various time periods by contrasting movie posters from different years. Deep learning is a prime candidate for this analysis because it allows for complicated features and patterns to be learned that are predictive of the movie's release year. Then, those features were used to indirectly predict when a poster's respective film was released.

By analyzing the model's accuracy across several years, we hope to pick up on trends that may have significantly impacted the movie poster creation process. Although specific patterns and features the model uses for its prediction will not be interpretable due to the nature of deep convolutional models, these differences may be qualitatively and manually analyzed. Analyzing how movie posters have evolved over time may be useful to media historians, graphic designers, and advertising agencies, who each need to know how to identify subtle patterns in typography, color palettes, and layouts corresponding to different eras.

B. Background

I could not find any literature on methods to predict a movie's release date based only on its poster, which is the problem in this work. Therefore, this section briefly reviews the literature on related topics.

The most studied area of research related to movie posters is genre classification. The first appearance of this being done with deep learning was by Chu [1], where the authors fine tuned a pre-trained deep convolutional network. The network analyzes the movie poster and some related metadata, and it outputs a prediction of which genre the movie falls into. This work also released a data set containing IMDb movie poster images, discussed in Section I-C. Similar work is done by Barney [2] and Kundalia [3] using different novel techniques for genre classification. A method using transfer learning to extract text from movie posters, including the movie's title, tagline, and names of the actors, director, and production house with high accuracy was proposed by Ghosh [4], where the paper also claims that such text may be used to identify the movie's genre.

Özkan [5] evaluated movie posters for their ability to predict other metadata about a movie, namely their box-office revenue.

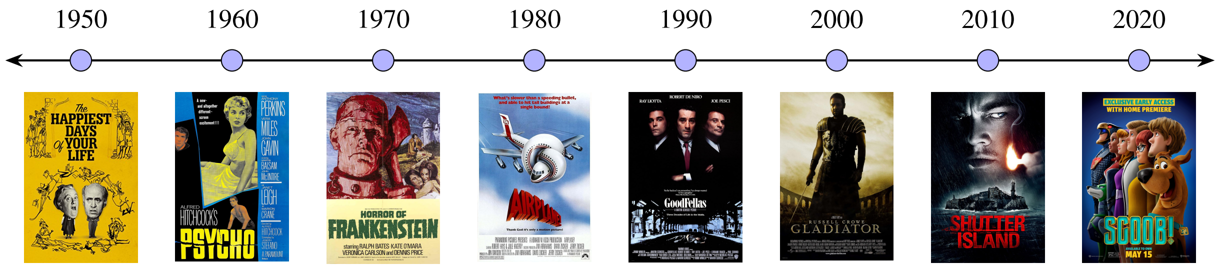


Fig. 1: A sampling of movie posters obtained from IMDb.

This work used deep convolutional networks to achieve state-of-the-art performance on this task.

Outside of the computational realm, analysis of movie posters has been done qualitatively by Rhodes [6] and Chen [7], including discussions about how the style of movie posters has changed over time.

Finally, estimating the date that an image was captured has been studied in the past, notably when Müller used deep convolutional networks [8]. However, the data used for this work was natural images instead of movie posters.

Given this prior work, the study conducted here to estimate relative release dates between two movie posters is novel.

C. Dataset

Several publicly available movie poster datasets exist. The most promising of these for this work is [1] available on Kaggle, which contains one poster image per movie for a set of Hollywood movies released between 1983 and 2015, comprising almost 8,000 total images. While a good starting point, this dataset's date range is smaller than ideal because it does not allow for analysis of movie posters across a large enough temporal scale.

IMDb is a website that documents thousands of movies and TV shows throughout history, including movie posters. The IMDbPY Python library allows the user to query a movie poster from the website, given the IMDb ID of the movie, which is a unique identifier for each entry on the website. To increase the date range of the Movie Poster Dataset, the Full IMDb Dataset [9] on Kaggle was used. This dataset contains several IMDb IDs for thousands of movies released before 1983, and it was used to query additional movie posters for older movies. The final result was a dataset containing over 100 movies per year between 1920 and 2015, with 10,752 total movie poster images.

However, this combined dataset included some form of bias between the two data sources with some defining features highly predictive of which dataset it came from. It was unclear to the human eye what the difference is between the posters in the Kaggle dataset and those obtained from IMDb; it may be that the Kaggle dataset consists only of Hollywood movie posters while the IMDb dataset contains movie posters from movies worldwide. Regardless, the model, which is trained to input a pair of movie posters and predict the relative release order of the associated two movies, could easily differentiate between the two sources, eventually learning that posters with the features related to the Kaggle dataset were all post-1983 and those with features related to IMDb were all pre-1983. This resulted in high accuracy when comparing one poster from each data source but significantly lower accuracy when comparing two posters from the same source.

Because of this, a new dataset, the "Century Poster Dataset," was collected entirely from IMDb. This dataset consists of exactly 100 movie posters per year for each year from 1925 to 2024, comprising a uniformly-distributed century's-worth of 10,000 movie posters. The bias in the previous dataset resulting from two data sources was eliminated by collecting every poster from the same source in this dataset.

Instead of inputting a single movie poster at a time to the model, each input is a pair of movie posters corresponding to movies released in different years, as will be discussed further in Section II. Therefore, for the Century Poster Dataset (or any other dataset with a constant number of posters per year), the number of unique samples available to train the model with is

$$\begin{aligned} N_{\text{pairs}} &= \binom{P}{2} - \frac{P \cdot (Y-1)}{2} \\ &= \binom{10000}{2} - \frac{10000 \cdot 99}{2} \\ &= 49,500,000 \end{aligned}$$

where N_{pairs} is the number of unique pairs to use to train the model, P is the number of posters available (10,000 for the Century Poster Dataset), and Y is the number of unique years represented by the posters (100 for the Century Poster Dataset). Note that some possible pairs are omitted due to being released in the same year. With $P = 10,000$ as shown above, this results in 49.5 million unique pairs. However, the poster dataset is divided into a 70/30 train-test split stratified by release year before finding possible pairs so that we do not train and test on the same movie posters, even if they are paired differently between the two sets. Therefore, we calculate the number of pairs with $P = 7,000$ for the training set, resulting in over 24 million unique pairs, and again with $P = 3,000$ for the testing set, resulting in just under 4.5 million pairs. Due to computational limitations to be discussed in Section IV, this dataset was too large to use, so both the training and testing sets were randomly down-sampled by a factor of 1,000, resulting in 24,255 training pairs and 4,455 testing pairs.

II. Model

This section will cover the various model architectures and hyperparameter adjustments attempted, as well as the final architecture chosen.

It is expected that the visual appearance of movie posters does not change suddenly but rather gradually as long-term trends slowly develop and die off. Therefore, we conclude that movie posters contain similar visual features in consecutive years. In other words, the relationship of a given visual feature to the years in which it is popular is a one-to-many relationship. Because of this, training a model to predict a precise release year would likely result in disproportionately poor performance. To avoid this difficult task, the model used instead was a binary classifier that takes two movie posters as input and predicts which of the two corresponds to the movie released later (i.e., more recently). In this way, the absolute year is abstracted away from the model, and it instead learns relative dates between features and combinations of features.

Before being sent to the model, each movie poster was resized to a constant size of (3, 224, 224) to eliminate variation in poster shapes and sizes and normalized with $\mu = [0.485, 0.456, 0.406]$ and $\sigma = [0.229, 0.224, 0.225]$ for the three color channels, respectively.

Within the model, a pre-trained deep convolutional model first extracted features from each resized and normalized movie poster. The feature extractor model was pre-trained on an image classification task, such as ImageNet, and the feature vector for each poster was 512-dimensional. Then, the feature vectors were concatenated, and the resulting combined feature vector was put through a small linear decoder with Rectified Linear Unit (ReLU) activations to formulate the model's prediction. The output vector is of length 2, where each element in the vector corresponds to the model's predicted probability that the corresponding movie was released later.

A UNet and a ResNet model were each attempted for the feature extractor, each with the last layer removed, and the ResNet-18 achieved the best performance by a small margin. For the decoder, the best balance of performance and speed was achieved with just two linear layers containing 128 and 2 neurons, respectively. Fig. 2 shows the final model architecture. Using dropout in the linear decoder decreased testing accuracy.

Gradient descent was performed to train the model using the Adam optimizer with a learning rate of 1e-5 and a batch size of 64. The cross-entropy binary loss function was used. The optimizer's default parameters resulted in the best performance, and a learning rate scheduler decreasing the learning rate by a factor of 0.5 each epoch slightly improved performance.

III. Results

The model was trained for 10 epochs, taking about 14 hours. It achieved a peak testing accuracy of 86.3% and peak training accuracy of >99%.

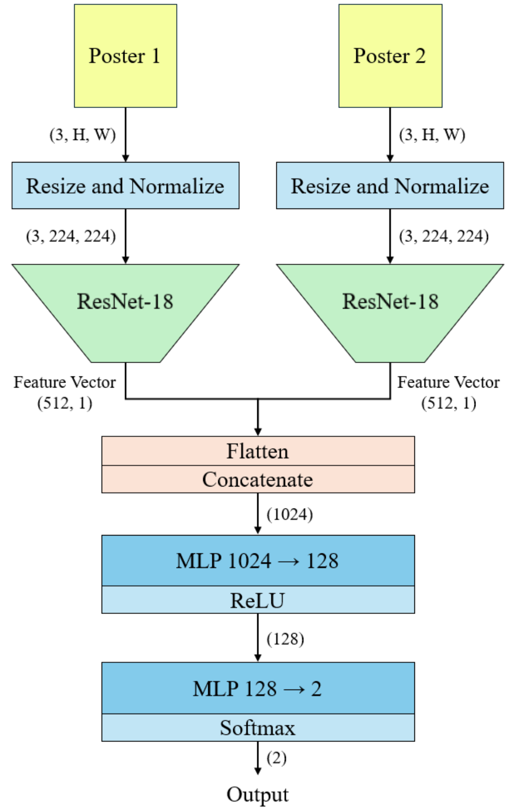


Fig. 2: Model architecture using ResNet-18 feature extractor and linear decoder

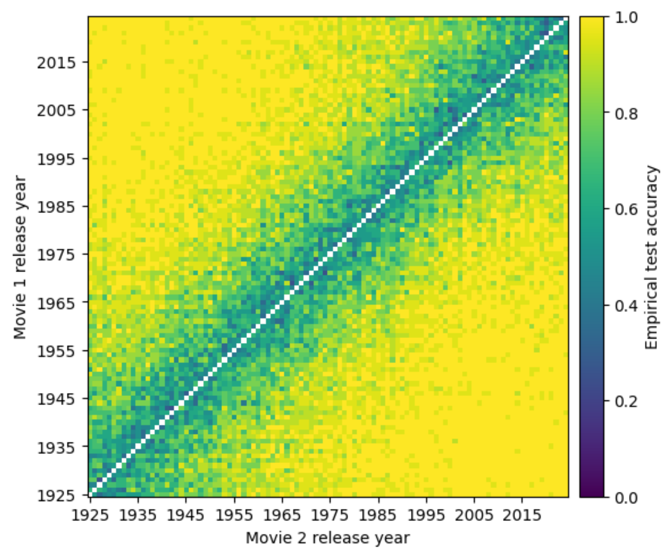


Fig. 3: Confusion matrix

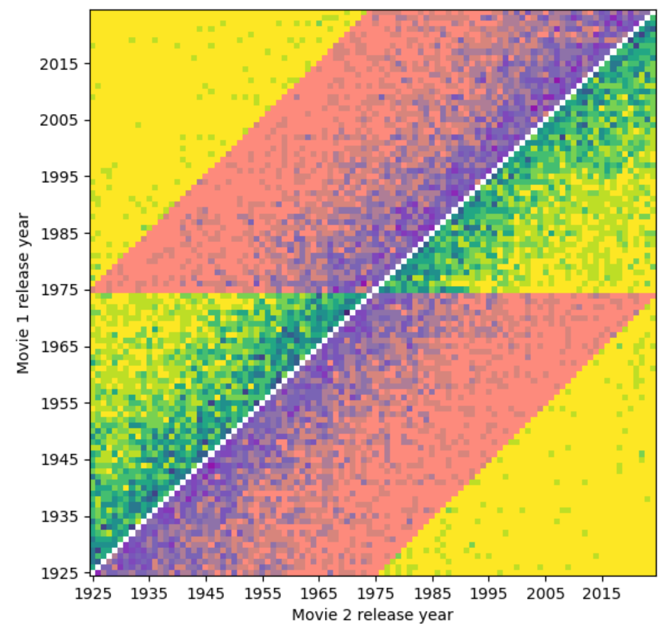
A confusion matrix illustrating the years between which the model has good and poor performance, respectively, is shown in Fig. 3.

Because the confusion matrix appears symmetrical over the diagonal, we can average accuracy scores row-wise in the highlighted region of Fig. 4a to calculate a reasonable proxy for each year's overall accuracy, shown in Fig. 4b. We only average over the highlighted pixels in each row to ensure that the average accuracy calculated at each year contains the same amount of information from an equal time range.

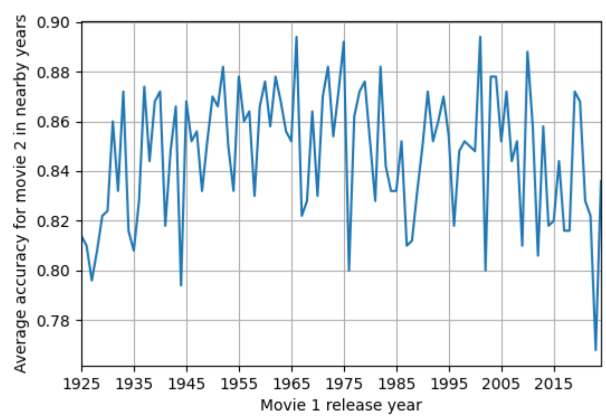
We cannot simply average across the entire row. To illustrate this, let us consider the following example. In 1930, we only have results from comparing with movies within the previous 5 years, but no further back, even though we would extrapolate that the envelope of poor accuracy in the confusion matrix would continue if extended down that far. This is, we would expect relatively poor accuracy if we compared movies in 1930 with movies in, say, 1924 – a year for which we do not have data. However, we have comparisons from movies of 1930 with movies well into the future, which results in lots of high-accuracy predictions. Contrastingly, in 1975, we had plenty of poor accuracy comparisons on both sides of the diagonal, but we lacked the far-out high accuracy comparisons we had in 1930. For this reason, the average accuracy across the entire 1930 row would be significantly higher than in 1975 despite both rows following the same pattern. This is why we use the shifting window averaging.

As mentioned above, there is a clear trend that comparisons made between movies of nearby years have relatively low accuracy. This trend causes the diagonal envelope with lower accuracy scores that we see in the confusion matrix. Fig. 5 encapsulates this trend. In this figure, the blue points represent the model's accuracy for movies that are at most the number of years apart corresponding to each blue point's X value. This is, given a particular integer X value representing some number of years, the blue point at that X value shows the average accuracy of the predictions for all movie pairs released at most X years apart in the test set. Likewise, the orange point at that X value shows the average accuracy for all movie pairs released more than X years apart, and the green point for all movies released exactly X years apart. We can think of this as drawing two parallel diagonal lines in the confusion matrix equidistant from the secondary (white) diagonal. The horizontal (or, equivalently, vertical) distance from these diagonal lines to the white diagonal corresponds to our X value. The average accuracy score for all elements of the confusion matrix between the two diagonal lines is the value of our blue point, outside the lines is our orange point, and directly on the lines is our green point. Fig. 6 illustrates two examples of this, respectively for X values of 25 and 50, where an average is taken across each colored region to determine the value of the corresponding colored points on the Fig. 5 graph.

Fig. 5 also includes a horizontal line in red representing 50% accuracy, which is the chance that we would randomly guess the correct order of a pair of movies, and one in cyan at 86.3% accuracy, which is the model's overall accuracy across all movie pairs in the test set. As we would expect, the orange points approach the cyan line as the year gap approaches 0 because the subset of samples >0 years is equal to the entire test set (i.e., no samples are $\leq$0 years apart). The blue points approach the cyan line as the year gap approaches 100 for the inverse reason – that no samples are >100 years apart.



(a) Sliding window used to calculate per-year average accuracy, superimposed on confusion matrix



(b) Average accuracy per year for comparisons within sliding window

Fig. 4

These results are novel not because they outperform a previous model but because the application itself is novel.

IV. Discussion

In this project, we trained a deep learning model to predict the chronological order of the release of an arbitrary pair of movies based exclusively on their posters. In this section, we will analyze the figures presented in Section III, discussing how the results compare with our hypotheses and acknowledging potential implications. Then, we will summarize the connection between the model and the results and conclude by presenting ideas for future work.

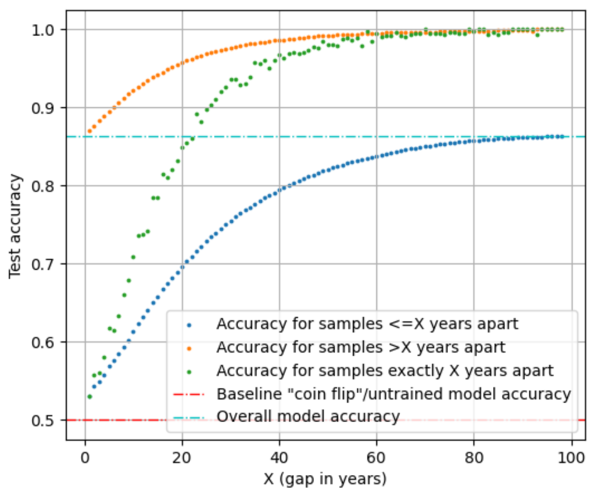
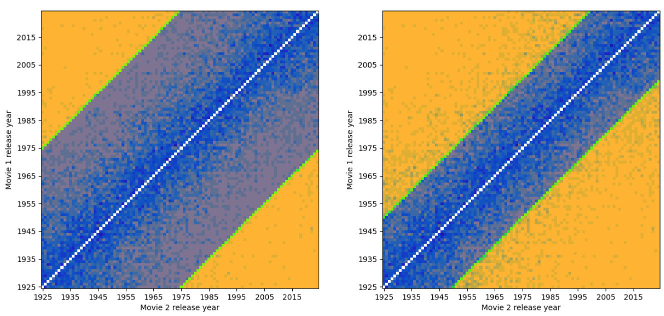


Fig. 5: Analysis of testing results showing accuracy scores for comparisons of various year intervals.



(a) Interpretation of Fig. 5 for X=50.

(b) Interpretation of Fig. 5 for X=25.

Fig. 6

Fig. 3 shows us that when the two movies were from nearby years, the predictive accuracy of their release order was significantly lower. This makes sense for two reasons: first, the absolute difference in release year is lower, so the model requires more fidelity to predict which movie was released later accurately. Second, we expect that movie posters from nearby years do not have substantial visual differences because visual trends take a long time to develop and disappear, and technology to facilitate the creation of posters would not have changed significantly between the two nearby years. Along the same lines, the model had much higher accuracy when comparing movie posters separated by a long time. This also makes sense because enough time passed between the two movies' releases for technologies and visual trends to develop, adding to significant visual differences between the two posters. Although the reason that the model makes its decisions is not interpretable due to the nature of deep convolutional networks, we expect that these differences in visual features are why it achieved high accuracy in this scenario. These two trends are illustrated by the confusion matrix being dark in the bottom left and top right corners but much lighter in the other two corners, and both align with the hypotheses posed before starting the project.

We see the same trend in Figure 5. However, this representation quantifies the trend, making it more interpretable. We see from the orange line in this graph that to be at least 95% confident that the model will correctly predict the order of two movies, we must know that their releases were separated by at least 18 years. For 90% confidence, this minimum gap shrinks to just 6 years.

Finally, Fig. 4b attempts to represent accuracy per year. This graph aimed to show clear inflection points with the hypothesis that a particular year or period with particularly high accuracy would mean that trends were developing quickly around that year, causing its posters to be visually unique and easily and accurately classified by the model. Likewise, a year or period with relatively low accuracy would imply that poster creation trends were static, so a particular year's posters could not easily be differentiated from its neighbors'. The hypothesis made before beginning the work was that large technological developments aiding in poster creation (i.e., the invention and improvement of design software on computers or advances in printing technology) would be met with coinciding periods of high model test accuracy. However, this hypothesis does not appear true, as the curve in the graph apparently does not have any interpretable trends, and any signal present is corrupted by noise of much higher amplitude. This could mean that adopting these technological trends has happened slowly, with any effect on the model's prediction accuracy spread over several years, with no particular year's posters looking staunchly different from those in surrounding years to make a noticeable impact on accuracy. It could also mean that such innovation is happening so rapidly that every year's posters look different from its neighbors, but no single year's innovation is significantly more groundbreaking than others'.

Note that Fig. 4b would have been the primary result if the model had been trained only to predict a movie's precise release year instead of classifying the relative order of two movies. By using the methods we did, we were able to develop significantly more analysis on the development of movie posters over time.

The model choice of a ResNet model as a feature extractor was apparently successful in capturing both low-level and high-level features in the movie posters while using a pre-trained model leveraged visual feature representation capabilities learned on a large-scale dataset, saving time and computational cost. The task's success relied heavily on the ResNet's ability to extract meaningful features that reflect the evolving visual trends in movie posters. Because the ResNet was pre-trained on natural images, which may not perfectly align with the domain of movie posters, the quality of extracted features may have suffered. However, training a model from scratch was not feasible and would require a larger data set. After feature extraction, concatenating the feature vectors created a joint representation that included information about both posters to allow the decoder to compare them directly. Different comparison strategies, such as cosine similarity, were not attempted because concatenation increased the information available to the decoder. The decoder was kept small and lightweight to reduce computational requirements while reducing the risk of overfitting.

A. Future Work

The largest downfall of this work was the lack of computational power available to train the model. A larger decoder architecture with more layers and neurons would likely result in stronger performance, as would the inclusion of more training data from the dataset for the model. However, because the model was trained locally without access to a GPU, this was not feasible in this work. Still, because the application itself is novel, this work may be viewed as a proof-of-concept, and future work, including more computational resources, may build upon it to improve results and introduce a new state-of-the-art model.

V. Code Availability

The code used for this project is available at https://github.com/RickHanish/Movie_Poster_Dating.

References

[1] Wei-Ta Chu and Hung-Jui Guo. "Movie genre classification based on poster images with deep neural networks". In: *proceedings of the workshop on multimodal understanding of social, affective and subjective attributes*. 2017, pp. 39–45.

[2] Gabriel Barney and Kris Kaya. "Predicting genre from movie posters". In: *Stanford CS 229: Machine Learning* (2019).

[3] Kaushil Kundalia, Yash Patel, and Manan Shah. "Multi-label movie genre detection from a movie poster using knowledge transfer learning". In: *Augmented Human Research* 5 (2020), pp. 1–9.

[4] Mridul Ghosh et al. "Understanding movie poster: transfer-deep learning approach for graphic-rich text recognition". In: *The Visual Computer* (2022), pp. 1–20.

[5] Kemal Özkan, Osman Nuri Atak, and Şahin Işik. "Using movie posters for prediction of box-office revenue with deep learning approach". In: *2018 26th Signal Processing and Communications Applications Conference (SIU)*. 2018, pp. 1–4. DOI: 10.1109/SIU.2018.8404649.

[6] Gary D Rhodes and Robert Singer. *Film by Design: The Art of the Movie Poster*. Univ. Press of Mississippi, 2024.

[7] Yunru Chen and Xiaofang Gao. "Interpretation of movie posters from the perspective of multimodal discourse analysis". In: *GSTF Journal on Education (JEd)* 1.1 (2014).

[8] Eric Müller, Matthias Springstein, and Ralph Ewerth. ""When Was This Picture Taken?" – Image Date Estimation in the Wild". In: *Advances in Information Retrieval*. Ed. by Joemon M Jose et al. Cham: Springer International Publishing, 2017, pp. 619–625.

[9] OctopusTeam. *Full IMDb Dataset (1M+)*. 2024. URL: https://www.kaggle.com/datasets/octopusteam/full-imdb-dataset (visited on 11/19/2024).