Classifying Movie Poster Genres with Deep Metric Learning

Edgar Trujillo
dept. Computer Science (of Tickle College of Engineering.)
University of Tennessee, Knoxville
Knoxville, USA
etrujil1@vols.utk.edu

***Abstract*—In recent years, deep learning techniques have revolutionized various fields, including computer vision, natural language processing, and speech recognition. One such prominent method in deep learning is deep metric learning, which focuses on learning an embedding space to facilitate similarity-based tasks. This research project proposes a novel approach to categorizing movie posters into distinct aesthetic categories using deep metric learning. Our primary goal is to develop an effective embedding model that captures the visual essence of movie posters, enabling accurate categorization based on the inherent aesthetics of an image. Furthermore, we hypothesize that this approach will provide a more nuanced understanding of the posters.' visual features and enhance the performance of classification tasks compared to traditional methods.**

***Index Terms*—image-embeddings, image-retrieval, visual-search, resnet**

I. Introduction

Movie posters play a crucial role in conveying the essence of a film and attracting audiences. A movie poster's visual contents and aesthetics can reveal critical information about the film, such as its genre, mood, and theme. This research investigates whether movie posters can be classified solely based on their visual contents and aesthetics. To tackle this problem, we propose a novel approach using deep metric learning, which aims to learn a low-dimensional embedding space where visually similar movie posters are grouped.

Our approach is to employ a base ResNet model, a deep residual neural network architecture known for achieving high accuracy on various image classification tasks, as the backbone for our embedding model. We plan to enhance the base ResNet model by training it with the proxy anchor loss. This advanced loss function encourages better intra-class compactness and inter-class separation in the learned embedding space. By optimizing the embedding model using proxy anchor loss, we expect a more discriminative representation of movie posters, enabling efficient and accurate categorization based on visual content and aesthetics.

The main contribution of our work, as opposed to previous research, lies in combining the ResNet architecture with the proxy anchor loss for the specific task of movie poster categorization based on visual aesthetics. While existing works have focused on classifying movie posters into genres or using different deep-learning techniques for aesthetic evaluation, our approach aims to provide a more comprehensive understanding of movie posters' visual features and improve classification task performance. Furthermore, our method has the potential to extend its application to other multimedia analysis domains, enhancing the overall impact of our research.

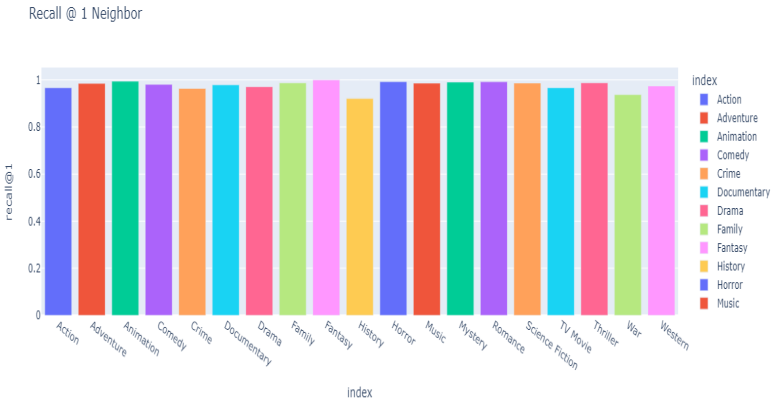

Fig. 1. Genre retrieval results for recall@1.

Fig. 2. Movie poster sample images. (a) Harry Potter and the Half-Blood Prince ['Adventure', 'Fantasy']; (b) Halloween ['Horror', 'Thriller']; (c) The Super Mario Bros. Movie ['Animation', 'Adventure', 'Family', 'Fantasy', 'Comedy']; (d) 365 Days: This Day ['Romance', 'Drama']

II. Previous Work

Our research project is inspired by several previous works that have significantly contributed to the field. These works encompass a range of domains and methods, which we will use as starting points and references for our project.

A. Convolutional Neural Networks

The ResNet architecture, proposed by [1] revolutionized the Convolutional Neural Networks (CNNs) field by introducing residual connections. This technique significantly improved the training of deeper networks. These "skip connections" allow the gradient to be backpropagated to earlier layers without getting diluted, effectively mitigating the vanishing gradient problem that hampers the performance of deep neural networks. Consequently, ResNet has become a cornerstone

in image-based use cases, providing a foundation for many subsequent works in image classification, object detection, and other computer vision tasks.

B. Movie Genre Classification

Domain-specific works such as [2] demonstrate the potential of deep learning techniques for movie poster classification based on genres, providing a starting point for our research. Further inspiration comes from [3], who explore language-guided zero-shot deep metric learning for images.

C. Metric Learning

Works like [4] have made significant strides in deep metric learning. The researchers behind this paper proposed an innovative approach that introduces a proxy anchor for each class and computes the loss based on the distances to the positive and negative proxy anchors, which has proven effective in improving the performance of deep metric learning models. This work has built upon previous explorations in the field, such as [5], which laid the groundwork for deep metric learning through the development of an embedding that was trained to separate the encoding of different faces in a Euclidean space.

[6] represents another substantial contribution to deep metric learning. The authors proposed a simple and efficient method that leverages proxies, or representative entities for each class, to simplify the distance metric learning process. This method circumvents the need for complicated pair or triplet selection strategies, thus making the learning process more straightforward and efficient. The paper resonates with the work of [7], which offered early insights into deep-ranking methods for learning fine-grained image similarity, suggesting potential parallels and applications across different areas of deep learning research.

Combining the insights and techniques from these foundational and domain-specific studies, we aim to develop a novel approach for categorizing movie posters into distinct aesthetic categories using deep metric learning. This will contribute to a more nuanced understanding of movie posters' visual features and improve classification task performance.

III. Technical Approach

Since we aim to categorize movie posters into distinct aesthetic categories based solely on their visual content, we approached this project assuming it required an embedding-based model versus a multi-label classification model. While a multi-label classification CNN model can learn to extract features(**embeddings**) from movie posters and classify them into multiple categories, it is not considered an embedding model in the same sense as deep metric learning models. Instead, the distinction between the two approaches lies in the learning objective and the structure of the output representation.

A. Multi-Label Classification

A multi-label classification CNN model is designed to predict multiple category labels for a given input image. It typically outputs probabilities for each category by a fully connected layer at the network's end, followed by a sigmoid or softmax activation function. The learning objective is to optimize the model parameters by minimizing the classification loss.

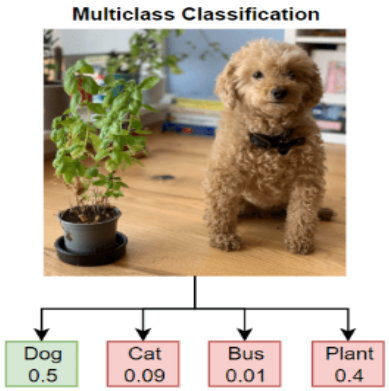

Fig. 3. Example Multi-label classification [8]

B. Deep Metric Learning

In contrast, an embedding-based approach using deep metric learning focuses on learning a low-dimensional embedding space where visually similar movie posters are *"pushed"* together and dissimilar posters are far apart. The learning objective is to optimize the model parameters by minimizing a specialized loss function(**proxy anchor loss**), which encourages the learned embeddings to capture the intrinsic relationships between movie posters based on their visual content.

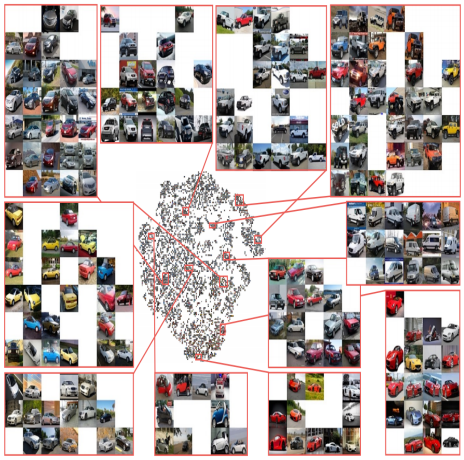

Fig. 4. Example visualization of metric learning [9]

C. Project Workflow

Subsequently, we then needed to agree upon an overall flow to train and evaluate our embedding model. The approach we took for this project involves the following key steps:

1) Data Collection and Preprocessing:

- We collected a large dataset of movie posters gathered from **themoviedb.org** labeled with genre categories.
- Once collected, we preprocessed the images by resizing them to a uniform size, normalizing pixel values, and applying data augmentation techniques such as rotation to increase the model's robustness to variations in styles and genres.

2) Network Architecture:
   - Using the **ResNet50** architecture as the backbone for our embedding model.
   - **ResNet50** is a deep residual neural network known for achieving high accuracy on various image classification tasks. It employs shortcuts(**skip connections**) to avoid the vanishing gradient problem that can occur in deep networks.
3) Learning Method:
   - Using the **Proxy Anchor Loss** as the learning objective to train the **ResNet50**-based embedding model.
   - **Proxy Anchor Loss** encourages better intra-class compactness (i.e., minimizing the distance between embeddings belonging to the same category) and inter-class separation (i.e., maximizing the distance between embeddings from different categories)
   - **Proxy Anchor Loss** code provided by [4]
4) Model Training:
   - Using **PyTorch and PyTorch Lightning** as training framework.
   - **PyTorch** is an open-source machine learning framework and **PyTorch Lightning** is an extension of PyTorch that simplifies the training and experimentation process.
5) Evaluation and Testing:
   - Evaluating with **Recall@K** metric.
   - **Recall@K** measures the ability of the model to retrieve relevant images within the top-k ranked results.
6) Similar Movie Posters Retrieval
   - Using **Approximate Nearest Neighbors** to retrieve most similar movie posters. (**ANN**)
   - **ANN** is an efficient similarity search algorithm suitable for image retrieval tasks.
   - Once a search index is built, one can perform a search by querying the index with a movie poster(**embedding**), and **ANN** will return the approximate nearest neighbors, i.e., the most similar movie posters, by **Euclidean** distance.

IV. Dataset and Implementation

The decision to create a new movie dataset by leveraging themoviedb.org's API was because all existing public movie datasets did not contain recent movie releases, as well as contained issues with acquiring the movie poster image files due to the stale dataset. Themoviedb.org API contains a https://api.themoviedb.org/3/movie/popular endpoint which accepts a page parameter and returns a response containing 100 movies.

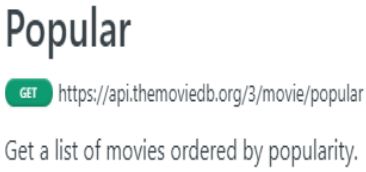

Fig. 5. API endpoint from themoviedb.org

A sequence of calls were made to query the top 100 pages of popular movies, resulting in a dataset size of **10000** movies.

TABLE I
THEMOVIEDB.ORG MOVIE RESPONSE

Field Name	Field Type
adult	boolean
backdrop_path	string
genre_ids	array of integers
id	integer
original_language	string
original_title	string
overview	string
popularity	number
poster_path	string
release_date	string
title	string
video	boolean
vote_average	number
vote_count	integer

TABLE II
THEMOVIEDB.ORG GENRES

Genre ID	Genre Name
28	Action
12	Adventure
16	Animation
35	Comedy
80	Crime
99	Documentary
18	Drama
10751	Family
14	Fantasy
36	Horror
10402	Music
9648	Mystery
10749	Romance
878	Science Fiction
10770	TV Movie
53	Thriller
10752	War
37	Western

Converting the movie genre array from integers to strings resulted in **19** unique movie genres. While it is natural and expected for movies to belong to more than a single genre,

we created a new **single_genre** column from each genre array for this research. The change attempts to have our embedding-based model learn to classify the most prominent genre for each movie poster. We expect our learned model to excel in posters that convey a single genre and potentially struggle as the number of total genres increases in a movie poster.

TABLE III
SINGLE_GENRE COLUMN CREATION

Movie	...	Genres	Single Genre
The Super Mario Bros. Movie	...	['Animation', 'Adventure', 'Family', 'Fantasy', 'Comedy']	['Animation']
...	...	...	...

In addition, we then one-hot encoded our **single_genre** column. One-hot encoding is used in CNN training for deep metric learning to convert categorical labels into a form that can be fed to the neural network. This approach creates a binary column for each category and assigns a value of 1 to the category applicable for a particular instance. One-hot encoding is helpful in deep metric learning because it allows the model to learn distinct features for each class without assuming an ordinal relationship between different categories.

Next, we adopted a stratified sampling strategy to partition the dataset, allocating **80%** for training, **10%** for validation to fine-tune the model parameters, and the remaining **10%** for testing to evaluate the generalizability of our model.

Finally, for training, utilized the following hyperparameters.

TABLE IV
MODEL TRAINING HYPERPARAMETERS

Hyperparameter	Value
Batch Size	32
Learning Rate	0.0001
Embedding Size	512
Epochs	20
Proxy-Alpha	32
Proxy-Margin	0.1
Pretrained Model	ResNet50
Freeze Pretrained	False
Embedding Normalization	True
Optimizer	AdamW

Through trial-and-error, each training iteration took around 20 minutes to train our embedding model on a PC with the technical specifications of an i7 CPU, 32GB RAM & Nvidia 3080 GPU.

V. EXPERIMENTS AND RESULTS ANALYSIS

This section presents the experimental setup, methods, and results from our deep metric learning approach to categorizing movie poster genres. We conducted experiments using separate Jupyter notebooks to train the model, extract embeddings, create an approximate nearest neighbors search index, qualitatively evaluate image retrieval, test on unseen movie posters and visualize the learned embeddings.

A. Model Training

```
  | Name      | Type         | Params
-----------------------------------------
0 | model     | ResNet       | 26.6 M
1 | criterion | Proxy_Anchor | 9.7 K
-----------------------------------------
26.6 M    Trainable params
0         Non-trainable params
26.6 M    Total params
106.463   Total estimated model params size (MB)
```

Fig. 6. Model Trainable Parameters.

The training phase of the model was implemented using a dedicated Jupyter Notebook. The training progress was monitored based on the validation-loss function value and the recall@3 value on the validation dataset. PyTorch Lightning automatically handles the creation of a model version entry with the hyperparameter & model-checkpoints saved.

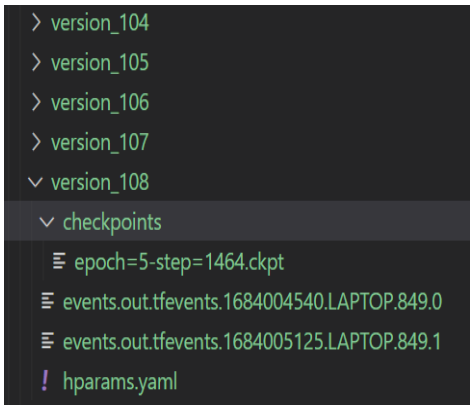

Fig. 7. PyTorch Lightning model training management.

B. Embedding Extraction

Following model training, all embeddings from the dataset were extracted using another Jupyter notebook. The trained model transformed each movie poster image in the dataset into a low-dimensional embedding. The resulting embeddings were saved for subsequent analyses and experiments.

C. Approximate Nearest Neighbors Index

An approximate nearest neighbors search index was created to enable efficient retrieval of similar movie posters based on their embeddings. This index allows a quick look-up of the most similar posters to a given query poster in the embedding space based on Euclidean distance.

D. Qualitative Evaluation of Image Retrieval

The model's performance in retrieving similar movie posters was qualitatively evaluated by retrieving the four nearest neighbors for a selection of query posters. The retrieved posters were visually inspected to assess the model's ability to capture the visual aesthetics of movie posters and group similar posters together.

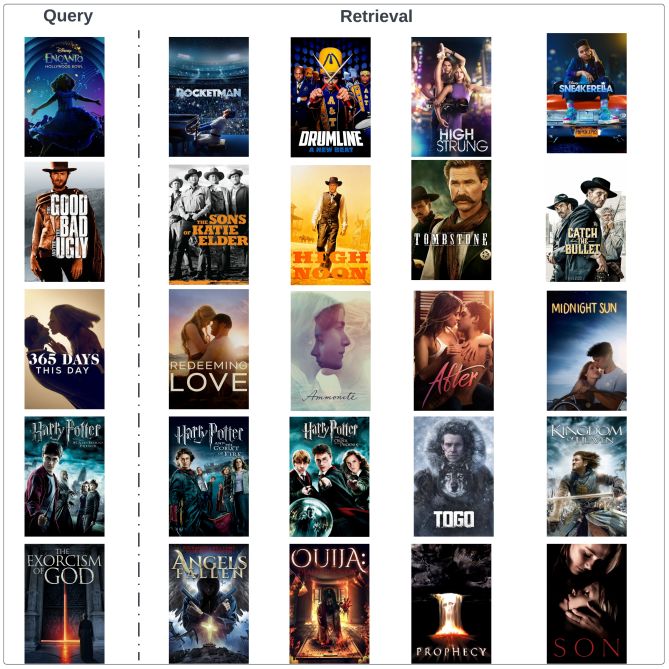

Fig. 8. Retrieval results on a set of images from the dataset using our training and search index method. Left column contains query images. The results are ranked by distance.

E. Evaluation of Unseen Movie Posters

We then assessed the generalization capabilities of the trained model. It was tested on a set of unseen movie posters from upcoming releases. The model's ability to correctly categorize these posters based on their visual contents and aesthetics was evaluated, highlighting the potential real-world applicability of our approach.

F. Visualization of Embeddings

Lastly, the learned embeddings were visualized using Uniform Manifold Approximation and Projection (UMAP), a dimensionality reduction technique. This visualization provides insights into the structure of the embedding space, the distribution of movie posters in this space, and the model's ability to separate different aesthetic categories.

G. Results and Discussion

The results obtained from our experiments reveal that the trained model could learn the visual essence of different categories well. When evaluated at **Recall@1**, our model achieved an average **Recall@1** of **97.70%**.

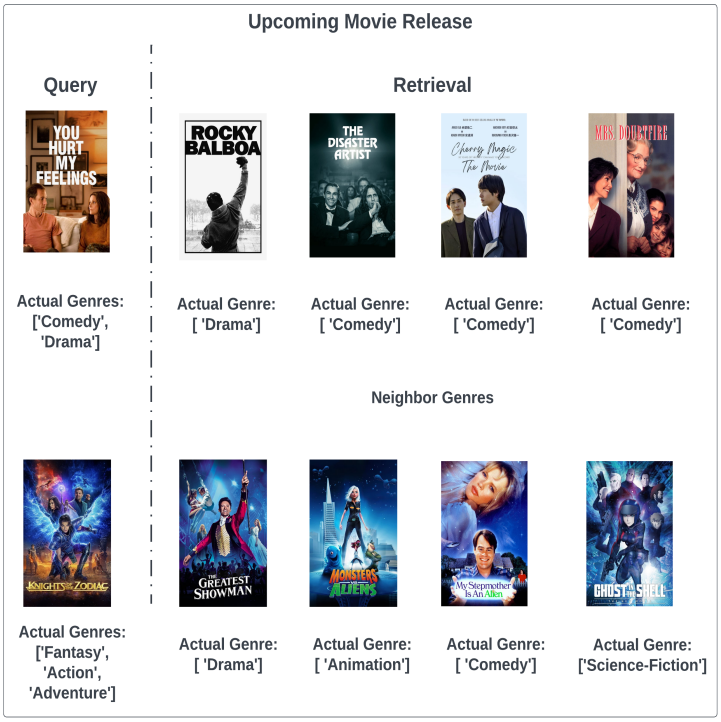

Fig. 9. Retrieval results on a set of untrained movie posters for upcoming movie releases. Left column contains query images. The results are ranked by distance.

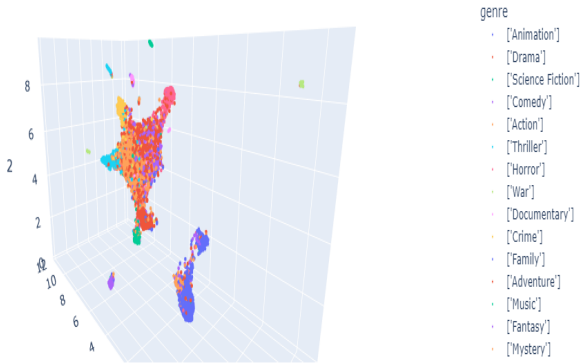

Fig. 10. Distribution of movie poster embeddings, and the model's ability to separate different aesthetic categories.

The results indicate that when the model is prompted with any image query of a selected genre, there is a **97.70%** chance that the nearest neighbor returned will be of the same genre as the image query. We settled on recall@k because it measures the ability of the model to retrieve relevant images within the top-k-ranked results. This metric helps assess the effectiveness of the embedding in capturing semantic similarities. As

TABLE V
RECALL FOR MOVIE POSTER NEIGHBORS

index	recall@1	recall@3	recall@8
Action	0.966516	0.987766	0.997424
Adventure	0.985240	0.996310	0.996310
Animation	0.994213	0.996528	0.998843
Comedy	0.981245	0.995499	0.999250
Crime	0.963889	0.975000	0.991667
Documentary	0.978873	0.978873	0.978873
Drama	0.971191	0.993906	0.998892
Family	0.987500	0.993750	0.996875
Fantasy	1.000000	1.000000	1.000000
History	0.921053	0.947368	0.973684
Horror	0.992274	0.998896	0.998896
Music	0.986111	0.986111	1.000000
Mystery	0.990826	0.990826	0.990826
Romance	0.992063	0.997354	0.997354
Science Fiction	0.986799	0.990099	0.996700
TV Movie	0.966667	0.966667	0.966667
Thriller	0.987826	0.998261	0.998261
War	0.937500	0.937500	0.950000
Western	0.974359	0.974359	0.974359

TABLE VI
MAJORITY GENRE FOR MOVIE POSTER NEIGHBORS

index	majority@1	majority@3	majority@8
Action	0.966516	0.723117	0.645203
Adventure	0.985240	0.809963	0.750923
Animation	0.994213	0.939815	0.924769
Comedy	0.981245	0.876969	0.830458
Crime	0.963889	0.777778	0.691667
Documentary	0.978873	0.802817	0.767606
Drama	0.971191	0.781717	0.695845
Family	0.987500	0.865625	0.800000
Fantasy	1.000000	0.824701	0.800797
History	0.921053	0.815789	0.500000
Horror	0.992274	0.866446	0.833333
Music	0.986111	0.847222	0.819444
Mystery	0.990826	0.871560	0.834862
Romance	0.992063	0.875661	0.838624
Science Fiction	0.986799	0.841584	0.795380
TV Movie	0.966667	0.666667	0.433333
Thriller	0.987826	0.867826	0.789565
War	0.937500	0.737500	0.737500
Western	0.974359	0.871795	0.833333

expected, our average recall@k for each genre grew as the number of nearest neighbors increased.

Fig. 11. Genre retrieval results by majority vote for movie poster with 5 neighbors.

Evaluating our model on a majority vote metric reveals that it regresses in its ability to accurately predict the most prominent genre based on a majority vote from its search neighbors. For example, for any search query requesting five nearest neighbors, our model averages a **77.58%** that the top genre from the returned neighbors will be of the same query genre.

The regression of the majority vote as the number of nearest neighbors increases supports our hypothesis that the model would excel in posters that convey a single genre and potentially struggle as the number of total genres increases in a movie poster. One possible reason can be attributed to how we extract the topmost genre into our **single_genre** column, and further experimentation that leverages the genre column in its entirety will produce a more sophisticated model that not only captures dominant genres but can effectively catch minor genre themes in movie posters.

VI. CONCLUSION

In conclusion, our research set out to investigate whether movie posters' genre categorization could be achieved based solely on their visual contents and aesthetics. Using a novel approach that combines a ResNet architecture with a proxy anchor loss function, our deep metric learning model learned a low-dimensional embedding space that grouped visually similar movie posters. Our approach showed promising results, with a high Recall@1 score of **97.70%**, demonstrating proficiency in retrieving relevant images within top-ranked results. However, the model showed limitations in situations involving multiple genres within a single poster. Despite these challenges, the potential for refining and extending our approach to other image-based domains remains promising. Future work could focus on enhancing our method to accommodate the complexity of multiple genres within a single poster, which would provide a more sophisticated understanding of movie posters' visual features and improve the overall performance of the classification task.

References

[1] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[2] Chu, W. T., Guo, H. J. (2017, October). 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 (pp. 39-45).

[3] Kobs, K., Steininger, M., Hotho, A. (2023). InDiReCT: Language-Guided Zero-Shot Deep Metric Learning for Images. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1063-1072).

[4] Kim, S., Kim, D., Cho, M., Kwak, S. (2020). Proxy anchor loss for deep metric learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3238-3247) https://github.com/tjddus9597/Proxy-Anchor-CVPR2020.

[5] Schroff, F., Kalenichenko, D., Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).

[6] Movshovitz-Attias, Y., Toshev, A., Leung, T. K., Ioffe, S., Singh, S. (2017). No fuss distance metric learning using proxies. In Proceedings of the IEEE international conference on computer vision (pp. 360-368).

[7] Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., ... Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1386-1393).

[8] GradCAM. MATLAB amp; Simulink. (n.d.). https://www.mathworks.com/help/deeplearning/ug/multilabel-image-classification-using-deep-learning.html

[9] Metric Learning. Papers With Code. (n.d.). https://paperswithcode.com/task/metric-learning

VII. Appendix

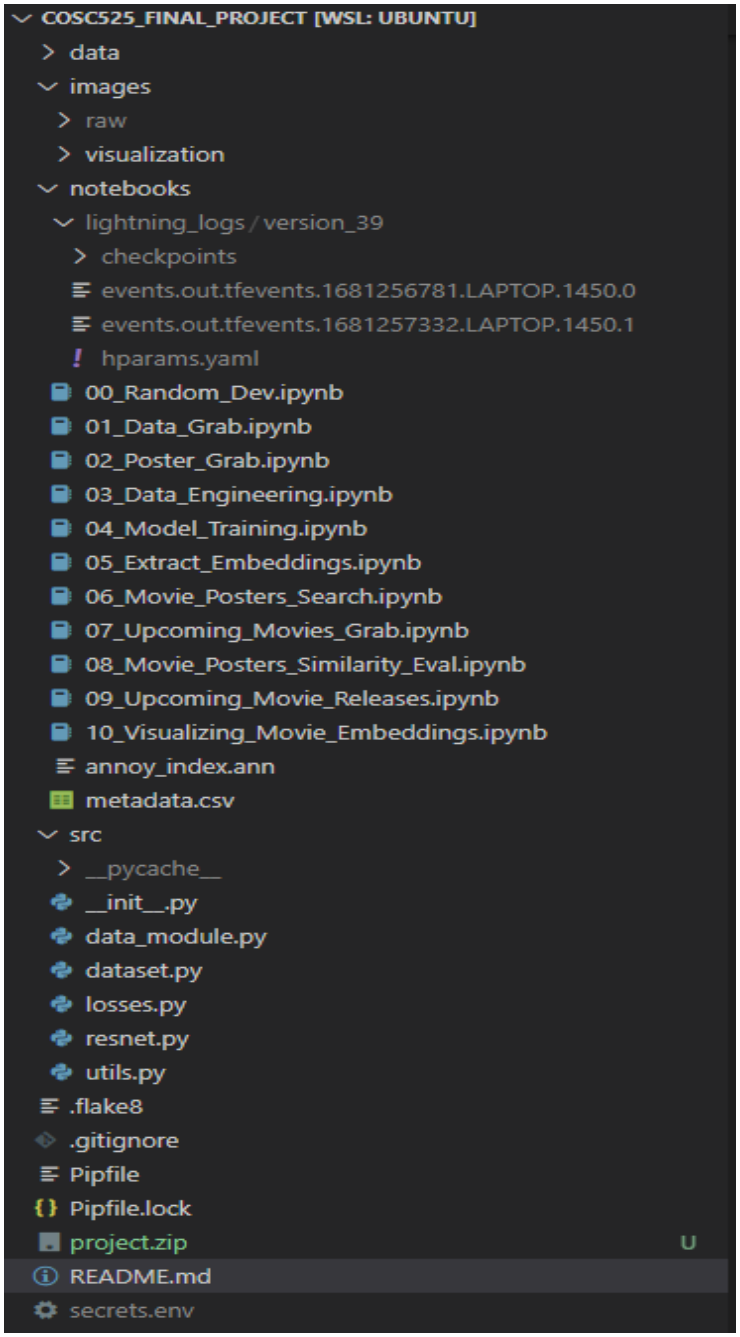

Fig. 12. Project Structure.