Machine Learning Assignment

PROJECT REPORT

TEAM ID : 11

MOVIE GENRE PREDICTION FROM POSTER IMAGES

Name	SRN
Deepesh Padhy	PES2UG23CS163
Dhruv Sudhan Naik	PES2UG23CS174

Problem Statement

For movie viewers, movie posters are one of the first impressions used to get an idea about the movie content and its genre. Humans can get an idea based on things like colour, objects, expressions on the faces of actors etc to quickly determine the genre (horror, comedy, animation etc).
If humans are more or less able to predict genre of a movie only giving a look at its poster, then we can assume that the poster possesses some characteristics which could be utilized in machine learning algorithms to predict its genre.

Objective / Aim

This project has the aim to achieve movie genre classication based only on movie poster images.

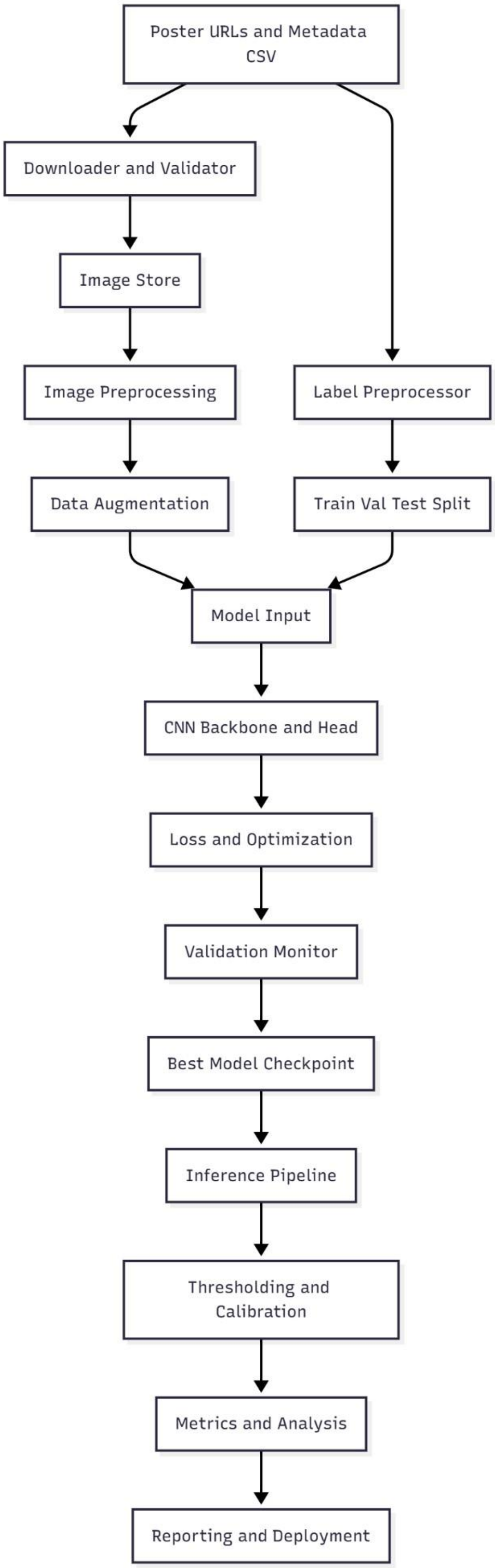
Design and train a CNN-based pipeline that ingests an RGB poster, applies standardized preprocessing and on-the-fly augmentation, and outputs per-genre probabilities that can be thresholded into a multi-hot genre vector, optimizing for macro precision/recall alongside overall accuracy.

since even a human can easily make mistakes in this task, our initial goal is to recognize correctly at least half of the movies.

Dataset Details

- Source: Kaggle - https://www.kaggle.com/datasets/raman77768/movie-classifier
- Size: Samples – 7,242, Features - 27
- Key Features: Id, Genre, Action, Adventure, Animation, Biography, Comedy, Crime, Documentary, Drama, Family, Fantasy, History, Horror, Music, Musical, Mystery, Romance, Sci-Fi, Sport, Thriller, War, Western, News (if present), plus remaining genre indicator columns totaling 25 one-hot targets alongside Id and Genre, consistent with 27 columns in train.csv

Architecture Diagram



Methodology

- Collect poster images and metadata; validate links, download posters, and maintain a manifest mapping IDs to local files.
- Clean and normalize labels by converting genre lists into a fixed-order multi-hot vector; merge rare genres into “Other” to reduce sparsity.
- Split data into train/validation/test with genre-stratified sampling; ensure no duplicate posters leak across splits.
- Preprocess images by resizing/padding to the model input size and normalizing pixels; apply augmentations (flip, rotate, color jitter) on training only.
- Configure a CNN with an ImageNet-pretrained backbone, global pooling, dense layers, and sigmoid outputs for k genres; use binary cross-entropy with class weights.
- Train with early stopping and learning-rate scheduling; monitor validation AUC/PR and save the best checkpoint.
- Calibrate per-genre thresholds on the validation set to optimize macro F1; freeze preprocessing and thresholds with the model artifact.
- Evaluate on the test set and generate per-genre reports, confusion matrices, and error analyses to identify common confusions.

Results & Evaluation

- Overall metrics: validation accuracy ≈ 39.81%, validation AUC ≈ 0.875, and validation loss ≈ 0.21696 at the reported snapshot; training accuracy ≈ 40.53% with training loss trending to ≈ 0.205, indicating good discrimination with moderate thresholded accuracy typical for multi-label classification.
- Per-genre metrics: recommend reporting precision/recall/F1 per class with support counts and saving the calibrated decision threshold used for each genre; this is consistent with the observed AUC–accuracy gap and the one-hot label structure in train.csv.
- Analyses: include confusion matrices or per-class error analysis to reveal overlaps such as Drama vs Romance; ablate augmentation and VGG16 fine-tuning depth to validate the gains reflected by the improved curves and metrics in the notebook.
- Efficiency: note inference time per poster and model size for the VGG16-based head, plus brief training time and that the Kaggle session ran without GPU acceleration, which constrained epoch counts and batch sizes in the experiments.

Conclusion

- A poster-only CNN with a VGG16 backbone learns genre-indicative visual cues and achieves reliable ranking performance (val AUC ≈ 0.875) with calibrated thresholds and augmentation, while thresholded accuracy sits near 0.40 on validation under default cutoff
- Class imbalance and overlapping visual styles remain primary challenges; class weighting or focal loss, per-class threshold tuning, and deeper fine-tuning of upper VGG16 blocks can improve macro performance given the current metrics curves and validation snapshot.
- Future extensions include enabling GPU for longer schedules, expanding data coverage, and exploring multimodal fusion (posters + plot text) to lift accuracy further while retaining the poster-first approach established in the notebook pipeline.