MULTI LABEL GENRE DETECTION OF MOVIEs USING MOVIE POSTERS

Jayakumar Rameshbabu|Jayasurya Bhaskar|Yogalakshmi Saravanan
Jayakumar.r@Knights.ucf.edu|jayasurya@knights.ucf.edu|yogalakshmi@knights.ucf.edu

Abstract

Image detection and classification is an important topic of conversation in any recent computer vision arena. Among the multiple applications that can be derived from detection and classification of images, genre detection is a relatively novel and unique application. This project utilizes image processing and convolutional neural networks to create a system for multi label image classification of the genre of a movie by inputting the movie poster. We have proposed five different models towards tackling this problem and performed an in depth analysis for each model using metrics and visualization to determine the best solution.

1.Introduction

The realm of films and movies has become such an important part of entertainment, and a domain that is greatly valued by people of all age groups and sectors. Which is why work related to entertainment has become prominent in fields like machine learning, computer vision and artificial intelligence. In particular the database for films and movies has expanded so much that there is a necessity to automate the handling of such massive databases and classifying them. The genre of films has become an important feature in many systems pertaining to entertainment, for example recommendation systems that are a trending topic in recent times. Movies have multiple genres associated with them and having to manually watch and classify these movies is a manual job that can be easily automated. This is why we propose a system that can solve this issue.

PROBLEM STATEMENT: To create an automated and optimized multilabel genre prediction system for movies based on movie posters

2.Method

2.1 Related Work:

Classification of Movie Posters to Movie Genres - Samuel Sung and Rahul Chokshi(2017)[1] - the dataset used in this paper is similar to the dataset we are handling and this served as groundwork reference to handle the dataset. The authors have implemented RESNET and DENSENET to perform classification with the suggestion that this model can be improved through CNN

Automatic Movie Posters Classification into Genres - Marina Ivasic-Kos et al (2016)[2] - this paper implemented machine learning models such as KNN and naive bayes to predict the genre of movies using the movie dataset. Using this paper as basis we implemented our own models to improve accuracy and performance

2.2 Our solution:

We propose a system that can predict the genre of a poster by simply using the movie poster as input.Our system aims to detect the genre of the movies by identifying and classifying the text, color scheme and other visual features present in the poster by creating a custom built neural network. To gauge the better

performing models we have taken five different models and performed analysis to determine the best solution to this problem.

3.Experiments

3.1 Dataset:

The dataset we have utilized in this project is the comprehensive and widely utilized Movie Posters dataset from Kaggle[3]

Dataset specifications:

- The dataset contains a total of 7876 images of movie posters across various genres.
- This dataset was collected from the IMDB website.
- Each poster image is associated with a movie as well as some metadata like ID, genres, director, box office and multiple other features. The ID of each image is set as its file name.

From this comprehensive set of features, only the genre attributes and ID have been taken for our problem statement

LINK TO THE DATASET: https://www.kaggle.com/datasets/raman77768/movie-classifier

3.2 Evaluation metrics:

The primary two metrics used for evaluation are:

- Accuracy
- Loss - cross entropy

$$Accuracy = \frac{\text{No of correct predictions}}{\text{Total no of predictions}}$$

$$\text{Cross-entropy} = -\sum_{i=1}^{n}\sum_{j=1}^{m} y_{i,j}\log(p_{i,j})$$

3.3 Model architecture and implementation:

The five models we have custom built and implemented are :

- Model 1 - a neural network based model with three convolutional layers having relu activation functions. It makes use of batch normalization and max pooling functions and custom set parameters. It uses sigmoid activation function for dense layers.
- Model 2 - an improvement of model one by decreasing the convolutional layers to two layers using the relu activation function. It utilizes softmax activation for the dense layers
- Model 3 - a vgg model from scratch with a very deep network of 13 convolutional layers. A Keras convolutional neural network model using the VGG19 architecture for classification with 25 classes. The model is created from scratch, without using any pre-trained weights
- Model 4 - inception model executed from scratch
- Model 5 - mobilenet model modified to suit our requirements by removing top layer and Add additional layers on top of the MobileNet model.

Each of these models was run for 20 epochs and the results from each model were pitted against each other based on our evaluation metrics.

IMPLEMENTATION PROCESS:

- Loading the dataset - 2300 poster images from the movie dataset
- Preprocessing - converting the images to numpy array, reshaping and splitting into test and train dataset
- Model implementations - formulating and implementing the five models using the training dataset
- Testing - running all five models with the test dataset
- Visualization and analysis- comparison of the performances of the five models by visual exploration

ARCHITECTURE

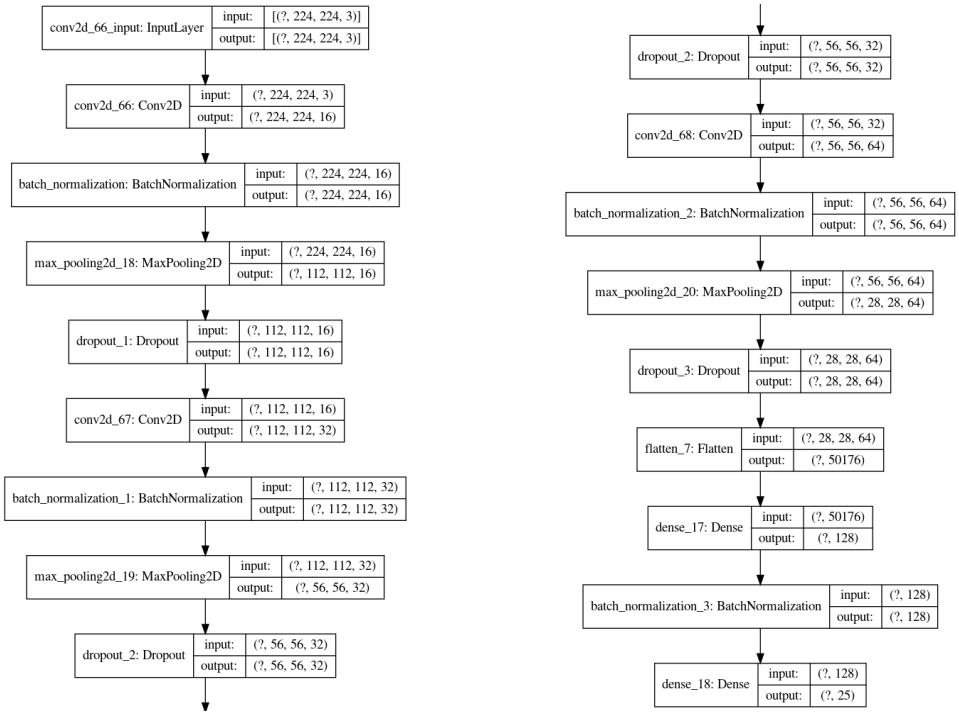


Fig 1 - Architecture of model 1

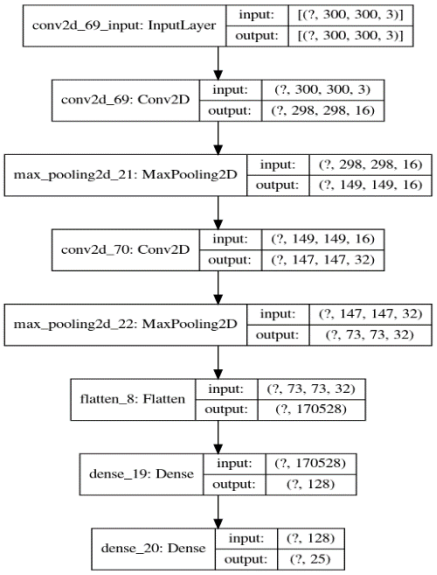


Fig 2 - Architecture of model 2

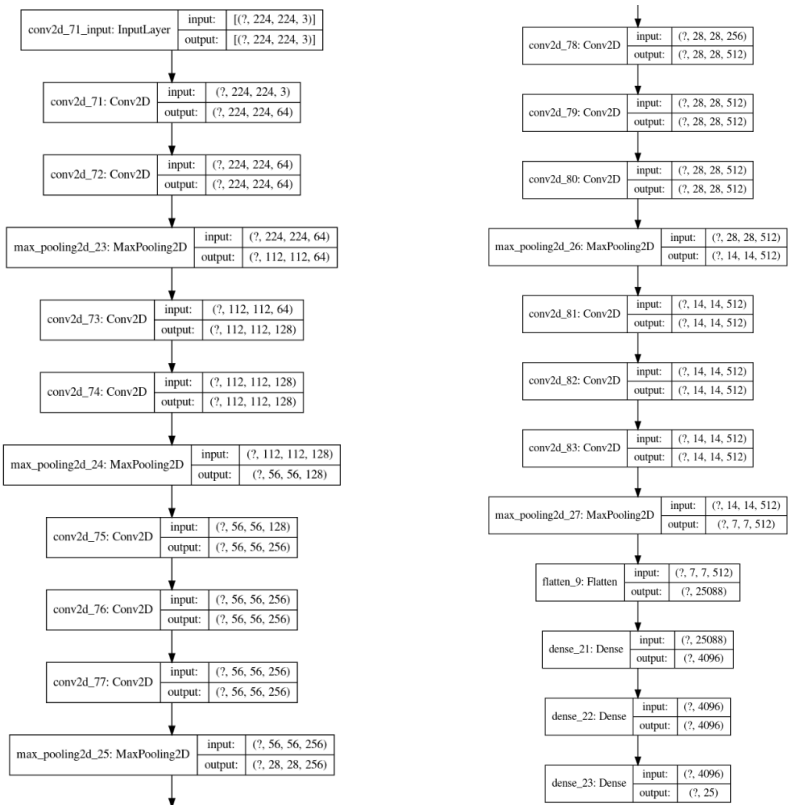


Fig. 3 - Architecture of model 3

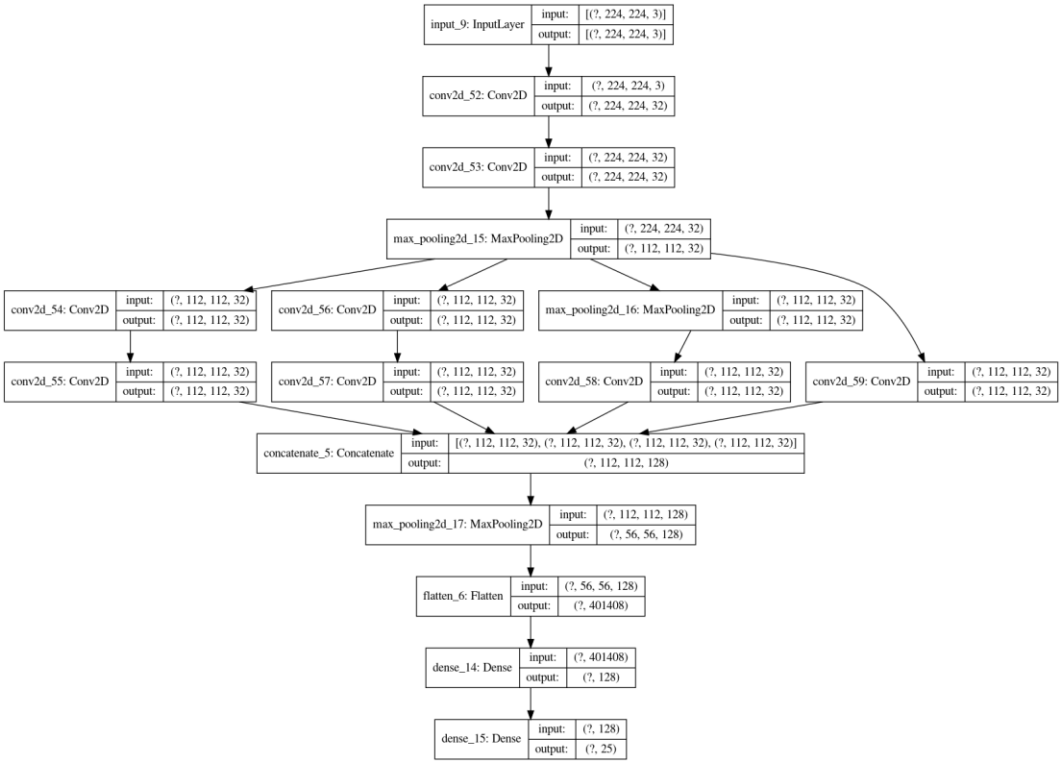


Fig.4 - Architecture of model 4

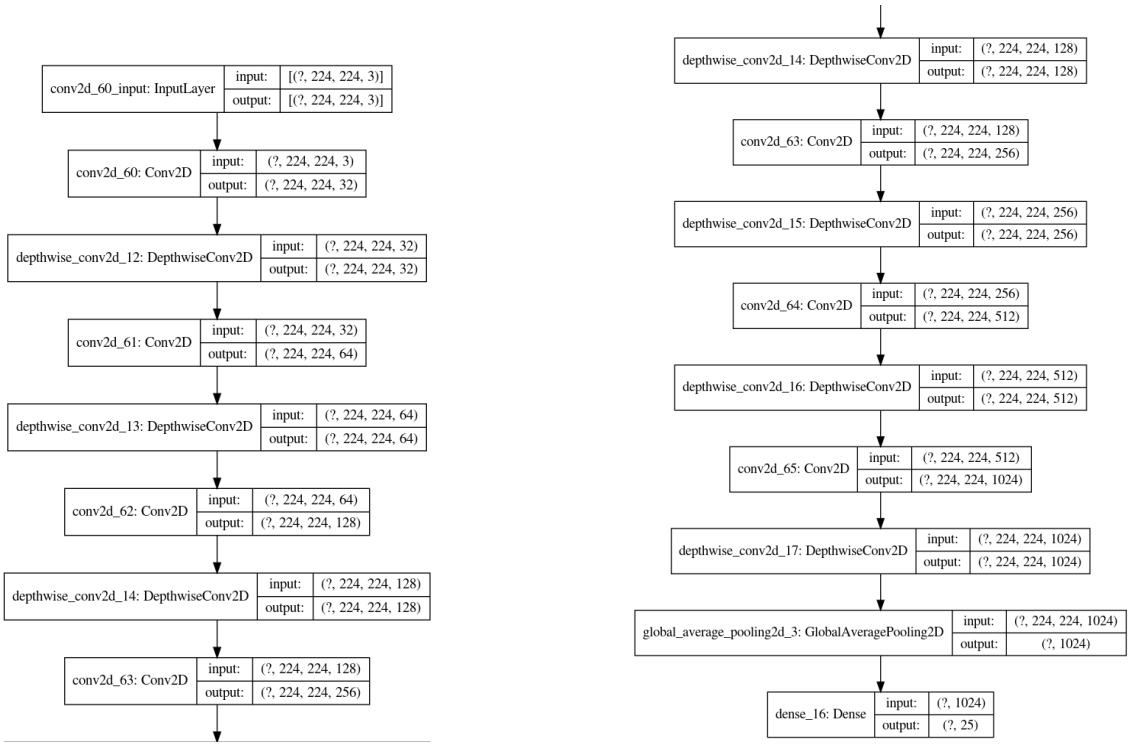


Fig.5 - Architecture of model 5

3.4 Results:

We were successfully able to create and implement all five models that were able to produce results of various accuracies. An in-depth visual exploration of our findings has been discussed in the next section

The output predictions from all five models are displayed in fig.6(a) and fig.6(b). The actual genre classification of the movies has been displayed to check the validity of the models. From the multiple experiments we performed on various inputs two outputs have been randomly chosen to demonstrate the results of our project.

To summarize our results:

- Importing and related image preprocessing of a subset of the movie dataset has been carried out.
- Two neural network-based models have been custom built and implemented
- VGG and inception model has been built from scratch
- A modification of MobileNet model to suit our purposes has been built
- All five models have been successfully implemented and results have been documented

Accuracies and val_accuracies obtained of all five models:
M1 - 0.9749567, 0.9069566
M2 - 0.9876578, 0.9116522
M3 - 0.9419516, 0.9151305
M4 - 0.9156517, 0.91130435
M5 - 0.99972945, 0.9153043

```
Actual classes:
Comedy
Drama
Romance

Model 1
Predicted classes:
Comedy
Sci-Fi

Model 2
Predicted classes:
Drama
Comedy

Model 3
Predicted classes:
Comedy
Western

Model 4
Predicted classes:
Comedy
Drama

Model 5
Predicted classes:
Drama
Comedy
```

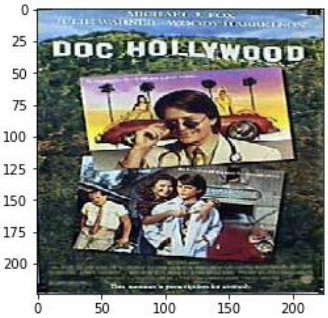


fig.6(a) - sample output 1

```
Image : 2
Actual classes:
Comedy
Drama

Model 1
Predicted classes:
Comedy
Adventure

Model 2
Predicted classes:
Comedy
Drama

Model 3
Predicted classes:
Comedy
Western

Model 4
Predicted classes:
Comedy
Adventure

Model 5
Predicted classes:
Drama
Comedy
```

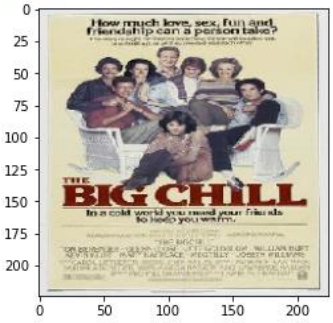


fig.6(b) - sample output 2

3.5 Analysis and discussions:

Fig.7 shows a comparison between our models : model 1 and model 2 based on the loss encountered over the duration of 20 epochs.

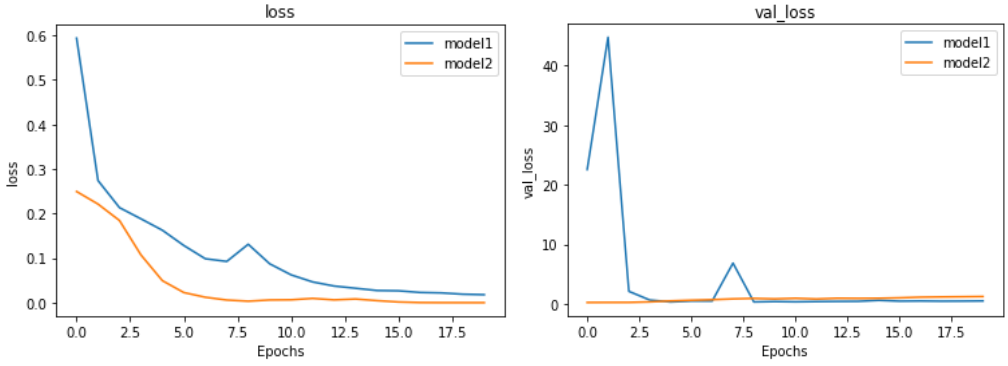


Fig.7 - loss and val_loss shown for model 1 and model 2 for clearer comparison

Fig.8 shows the comparison between model 1 and two on the basis of accuracy over the duration of 20 epochs.

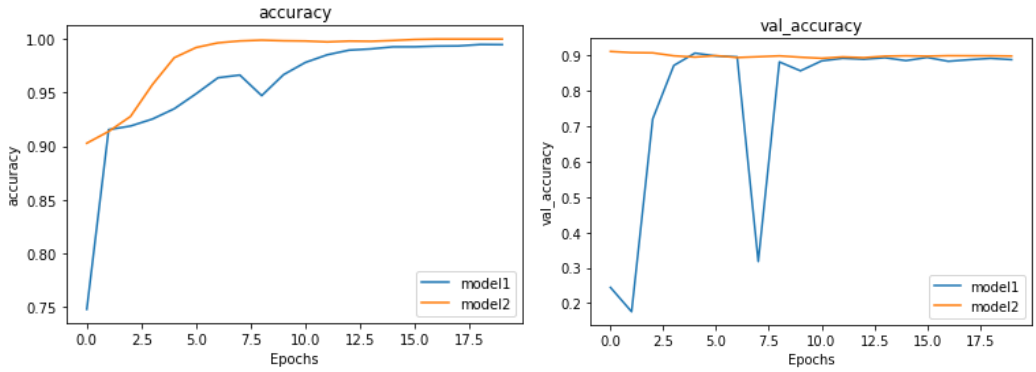


Fig.8 - accuracy and val_accuracy shown for model 1 and model 2 for clearer comparison

Fig. 9 shows the comparative loss encountered by all 5 models over the 20 epochs.

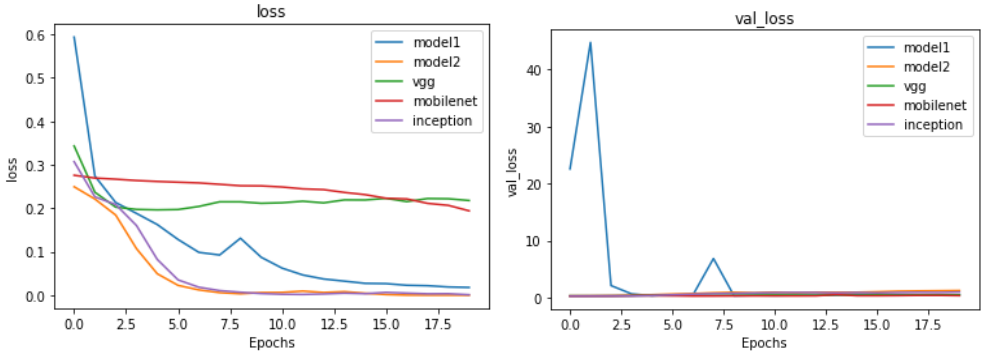


Fig. 9 - visual representation of loss and val_loss for all models

Fig.10 shows the accuracies of the five models among which once again inception and model 2 have performed well

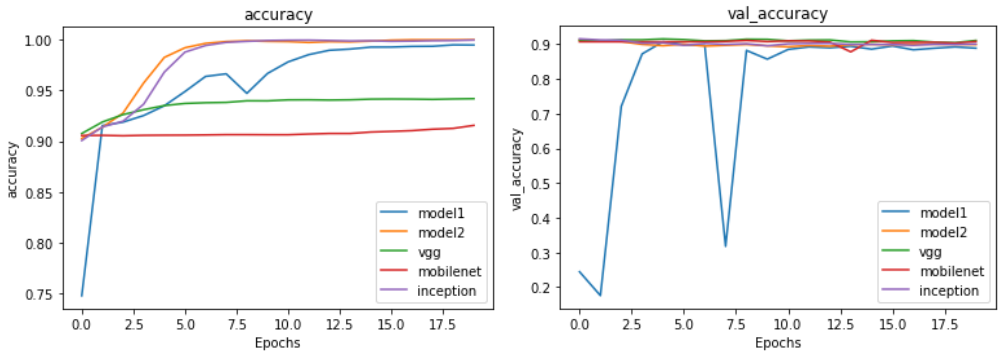


Fig.10 - visual representation of accuracy and val_accuracy for all models

DISCUSSIONS:

- In the comparison between our models, model 2 an improvement on model 1 has outperformed the latter in terms of both accuracy and loss
- Comparing our models to the other 5 we see that model 2 and inception models have the lowest loss and highest accuracy. We observe that our improved model is almost on par with the inception model having near equal scores
- Model 2 has a strong future scope and can be developed on to surpass inception since it is much easier and computationally inexpensive when compared to inception model

4. Limitations:

The primary limitation we encountered were hardware constraints. Having to process a large dataset of 7000+ images led to multiple system crashes even with GPU. We performed our code experimentation on Kaggle kernels using the free GPU access provided. Out of the entire dataset only 2300 images were taken by us to train and validate our models owing to this constraint. Higher power processors are required to completely perform computations on the entire dataset

5.Conclusion:

To quickly and easily assign multiple genre labels to movies using their movie posters we proposed, formulated and implemented a CNN based system. Five models were successfully implemented and evaluated using metrics. From that we have deduced two models(model 2 and inception) were clearly superior to the other three models. The results showed high accuracies around 97-99%. From the available resources a subset of images was trained and validated. This project holds value in movie recommendation systems in particular as it is able to completely automate the process of classifying movies on the basis of genre. The limitations and future scope that have been elaborated in the previous sections are a something to be noted and improved upon

6.Individual Contribution:

The designing and training of 2 neural network models. Specifically model 2 and the model based on the vgg-16 architecture. Given below in Fig.11 shows the model 2 summary in detail and fig 11 shows the model_vgg.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 16)	448
max_pooling2d (MaxPooling2D)	(None, 111, 111, 16)	0
conv2d_1 (Conv2D)	(None, 109, 109, 32)	4640
max_pooling2d_1 (MaxPooling2	(None, 54, 54, 32)	0
flatten (Flatten)	(None, 93312)	0
dense (Dense)	(None, 128)	11944064
dense_1 (Dense)	(None, 25)	3225

Total params: 11,952,377
Trainable params: 11,952,377
Non-trainable params: 0

Fig.11- Architecture summary of model 2

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 224, 224, 64)	1792
conv2d_3 (Conv2D)	(None, 224, 224, 64)	36928
max_pooling2d_2 (MaxPooling2	(None, 112, 112, 64)	0
conv2d_4 (Conv2D)	(None, 112, 112, 128)	73856
conv2d_5 (Conv2D)	(None, 112, 112, 128)	147584
max_pooling2d_3 (MaxPooling2	(None, 56, 56, 128)	0
conv2d_6 (Conv2D)	(None, 56, 56, 256)	295168
conv2d_7 (Conv2D)	(None, 56, 56, 256)	590080
conv2d_8 (Conv2D)	(None, 56, 56, 256)	590080
max_pooling2d_4 (MaxPooling2	(None, 28, 28, 256)	0
conv2d_9 (Conv2D)	(None, 28, 28, 512)	1180160
conv2d_10 (Conv2D)	(None, 28, 28, 512)	2359808
conv2d_11 (Conv2D)	(None, 28, 28, 512)	2359808
max_pooling2d_5 (MaxPooling2	(None, 14, 14, 512)	0
conv2d_12 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_13 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_14 (Conv2D)	(None, 14, 14, 512)	2359808
max_pooling2d_6 (MaxPooling2	(None, 7, 7, 512)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_2 (Dense)	(None, 4096)	102764544
dense_3 (Dense)	(None, 4096)	16781312
dense_4 (Dense)	(None, 25)	102425

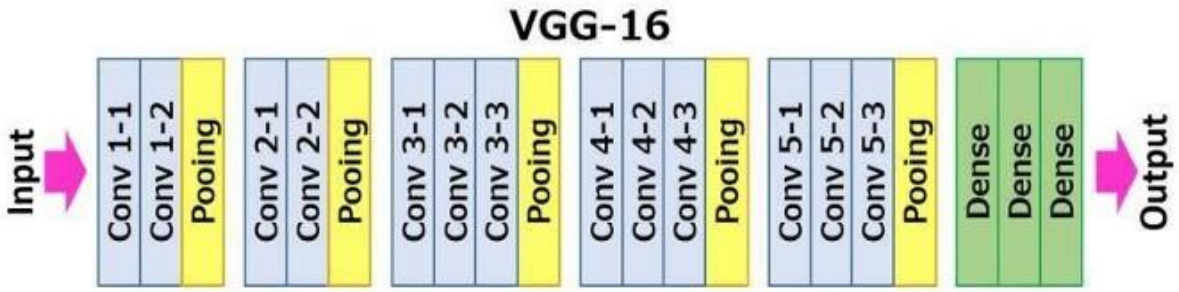
Total params: 134,362,969
Trainable params: 134,362,969
Non-trainable params: 0

Fig.12- Architecture summary of model_vgg (model 3)

I created the model 2 by just using a basic neural network structure consisting of 2 convolutional layers with each having a maxpooling layer after them and using ReLU activcation functuion. The dense layers have 128 units and the final output layer uses softmax. I trained it for 20 epochs and achieved an accuracy of 98.7%

For the VGG based model first I used the inbuild keras vgg model and tried training it . After learning about the architecture and how it works in detail, I went on to create a model from scratch that simulated the architecture of vgg16[5]. The model architecture diagram can be seen in detail in Fig.3.

It ended up with a total of 134,362,969 trainable parameters. This was the model which required the most time to train in comparison to all the 5 models that we have trained in this project.It consists of short consists of 16 layers, i.e.,13 convolution layers, 3 Fully connected layer, 5 Maxpool layers and 1 SoftMax layer. The final accuracy obtained was 94.1%.



I also solicited all the codes done by my teammates and compiled and run the training on my device to get the necessary results and outputs. We have uploaded the python notebook and the link to the dataset along with this report.

7. References

[1] Sung, Samuel, and Rahul Chokshi. "Classification of movie posters to movie genres." In *Proceedings of the Workshop on Multimodal Understanding of Social, Affective and Subjective Attributes*. 2017.

[2]Ivasic-Kos, Marina, Miran Pobar, and Ivo Ipsic. "Automatic movie posters classification into genres." In *International Conference on ICT Innovations*, pp. 319-328. Springer, Cham, 2014.

[3]Raman, "Movie posters," *Kaggle*, 06-May-2020. [Online]. Available: https://www.kaggle.com/datasets/raman77768/movie-classifier. [Accessed: 06-Dec-2022].

[4]Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014)

[5]Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

[6]Chu, Wei-Ta, 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*, pp. 39-45. 2017.