Movie Poster Genres Classification using Convolutional Neural Network

Jordan Tangy
3rd Year Student in Computer Science and Mathematics at Ariel University, Israel

Ron Sider
3rd Year Student in Computer Science and Mathematics at Ariel University, Israel

ABSTRACT

Over the last few decades, Artificial Neural Networks are showing great performances is many fields such as voice and face recognition, generating new Data, language translation, Image classification and much more. One of the most popular deep neural networks is the Convolutional Neural Network (CNN). The interest in having Convolution connected to a deep NN has recently begun to surpass classical methods performance in different fields; especially in pattern recognition. CNN take this name from mathematical linear operation between matrixes called convolution. They have multiple layers; including convolutional layer, non-linearity layer, pooling layer and fully-connected layer.

Among different type of models, Convolutional neural networks has been demonstrated high performance on image classification. In this project we built a Convolutional neural network for Movie Poster multi-label (genres) classification of Movies. The data set used was taken from Kaggle[1] and originally collected from IMDB website. It contains 41K different movie Posters . Additionally, there is a file containing the labels of each movie i.e. to each movie is associated the number 1 if it belongs to a specific genre, otherwise 0. In this paper we will explain how we used a Convolutional Neural Network in order to perform Movie Genre Classification based on Movie Posters and because a movie may belong to multiple genres, this is a multi-label image classification problem. In addition, we will also state the parameters that effect CNN efficiency and how CNN was implemented to obtain good results to classify the images.

KEYWORDS

Movie genre classification, movie poster, multi-label classification, Convolutional Neural Network

1 INTRODUCTION

Image attribute extraction has been widely studied in recent years, since visual attributes can boost various tasks such as image retrieval and image captioning . Most prior studies either focus on detecting or recognizing visual entities like object and scene, or extracting semantic concepts embedded in images. These attributes really provide widespread influence on multimedia retrieval and many computer vision applications. On the other hand, some visual attributes are implicit but can be easily perceived by human beings.

(a) (b) (c) (d)

Figure 1: Sample movie poster images. (a) *Captain America: The First Avenger* (Action , Adventure, Sci-Fi; (b) A Star is Born (Drama, Music , Romance); (c) *Expendables 2* (Adventure, Action, Thriller); (d) *Interstellar* (Sci-Fi , Mystery , Adventure).

When studying visual attributes for different types of images, we found that movie poster image contains different types of attributes and according to the genre, we can identify certain patterns. First, movie posters are created to attract people paying time and money to watch the corresponding movie. Information on a movie poster, therefore, should be attractive. Figure 1 shows four sample posters. In the figure caption we show movie name and the corresponding genres (in the parentheses), which are obtained from the IMDB website. Most of them present the most important imagery of the corresponding movie. For example, Figure 1(a) shows a violent scene with army aircrafts, fire in the background and a person holding a shield. This dramatic scene attracts people who like excitement or combat. Second, different movies target at different populations, and movie posters should concisely present genre information. For example, we can clearly perceive that Figure 1(a) and Figure 1(c) present action elements, Figure 1(d) seems mysterious, and Figure 1(b) describe a love story. Third, movie posters usually present important objects like gun and fire in Figure 1(c) and the shield of Captain America in Figure 1(b). Overall, we can identify different components in one image that give an indication to the people who are about to watch which kind of movie it is. Those component combined create a pattern that we want to detect in order to classify the image.

In this work, we propose to analyze movie poster images and classify them into movie genres based on a Convolutional neural network.

[1] https://www.kaggle.com/dadajonjurakuziev/movieposter/version/3

.

Estimating the genre of a movie according to its poster is actually not a new idea. There are many approach to solve this problem.

Although the approach used is this paper is a classical one, some challenges were encountered such as how many convolutional layers should be used and how many neurons should be used in the fully connected layers, how often Batch Normalization should be used, etc…

The work in [2] shows that a person's face and objects on a poster can be detected to establish correlations between objects and movie genre. These inspiring works motivate us to study implicit factors related to movie genres, and encourage us to build a computational model to do classification.

Contributions of this work are summarized as follows.

- To facilitate the proposed movie poster analysis, we collect a large movie poster dataset from the IMDB website. This dataset mainly consists of posters of movies, as well as the associated metadata like movie genres.
- We construct a computational model based on convolutional neural network to automatically classify a movie poster into genres. Note that one movie belongs to multiple genres, see Figure 1. This task is therefore a multi-label image classification problem.

The rest of this paper is organized as follows. Section 2 provides brief related word and required background. Section 3 describes the proposed model to achieve movie genre classification based on poster images. Section 4 shows experiments/simulation results Section 5 will describe our previous attempts and the last section concludes this paper.

2 RELATED WORKS AND REQUIRED BACKGROUND

There have been a few works on movie genre classification. Wei-Ta Chu and Hung-Jui Guo (Academics from National Chung Cheng University, Taiwan) proposed an interesting approach to classify movie posters. They used a large movie poster dataset and another file that is mapping the poster according to its genres. For example if the genres associated to movie X are action, comedy and adventure, then the number 1 will be assigned to those genres and to the rest will be assigned the number 0.

Let's break there approach into three parts.

They ***first*** build a Convolutional Neural Network composed of seven convolutional layers followed by a batch normalization layer. After normalizing, the feature maps are flattened as a vector to be a visual representation. Then this goes through a fully connected layer and another batch normalization layer. The ***second*** part is to input the image into the YOLO (You Only Look Once) algorithm which is an algorithm used for object recognition, i.e. that the algorithm will recognize different kind of objects on the poster e.g. cars, dogs, cats, persons, books, instruments, etc… .

The detected objects are flattened and are going through a fully connected layer. The output of the ***first*** and the ***second*** sub networks are combined and embedded to a fully connected layer, from there to a batch normalization and finally, this goes through softmax function in order to classify the results.

The main idea is to extract objects from the picture and to train the Network to understand that there are relations between certain objects and genres. For instance, if on the picture we see persons holding guns and tanks in the background, the YOLO algorithm will identify those objects. When these objects are embedded with the previous output of the convolution, the NN will start to establish a link between guns/tanks and genre War/Action. This technique goes much further, after face recognition, the gender and race of the person can be guessed and associates this characteristics to specific genres. It finds out that this method is powerful, because usually, objects on posters are a good indicator of the movie genre.

3 Project Description

3.1 Data description

Given a set of training data $D = \{X, Y\}$, where $X = \{x_1, x_2, \ldots, x_N\}$is the set of N posters images and $Y = (\mu_1, \mu_2, \ldots, \mu_N)$ is the corresponding genre information. The vector $\overrightarrow{y_i} = (y_{i,1}, y_{i,2}, \ldots, y_{i,M})$ isa binary vector , where $y_{i,j} = 1$ indicates that the ***i***th poster belongs to the ***j***th genre. Note that a movie may belong to multiple genres, i.e.$1 \leq \sum_j y_{i,j} \leq M$. Based on D, we would like to construct a computational model that outputs the probability of a given poster image x_i belonging to each movie genre. That is, the model is acting as a function such that $f(x_i) = (\tilde{y}_{i,1}, \tilde{y}_{i,2}, \ldots, \tilde{y}_{i,M})$ where $0 \leq \tilde{y}_{i,j} \leq 1$. A good model should be able to output the best probabilities, and only the three highest probabilities will be retained in order to obtain the genres prediction. As said in [1], our dataset contains a file of 41K movie posters and a csv file with all the related metadata (id, year, title, genres,..). Moreover , in contains the name of each genre (each column represents a different genre) and foreach row in the csv file (i.e. each movie) is associated the number 1 under the corresponding genre which the movie belongs to, otherwise 0.Regarding the movie poster file, it was truncated from 41K images to 30Kimages because of a lack of RAM in order to process all of the images. After importing these files to the work environment we transform the csv file into a dataframe object in order to manipulate easily.

In order to process each picture, choosing a picture resolution is critical because with a resolution too small , a lot of information are lost and the Network can won't be able to extract the necessary features .On the second hand, a too big resolution will take a lot of time to the Network to learn the features. After trial and error and also considering our limitations in memory , 224x224x3 was the optimal resolution to use.

After resizing the pictures to the desired resolution, they were inserted

in an array in order to ease their manipulation . The data was split into to parts : training set at a rate of 80% and testing set at a rate of 20%.

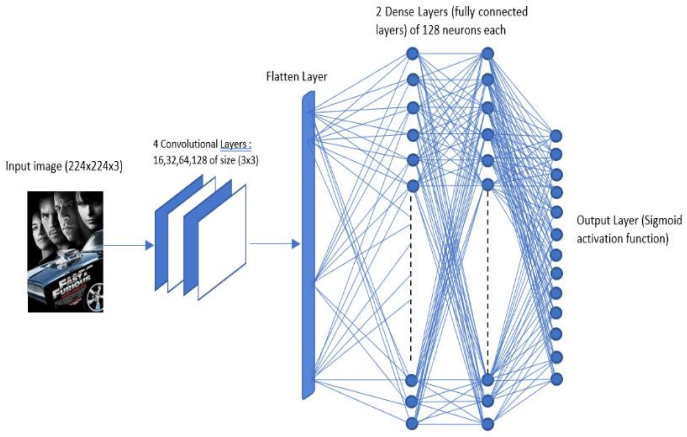
3.2 CNN Model

This Network consist of 4 Convolutional Layers, each one is followed by a Batch Normalization , MaxPooling and Dropout (range between 30% to 50%). The first Convolutional Layer contains 16 filters, the second contains 32 filters, the third contains 64 filters and the last one contains 128 filters, all of them are of a size of 3x3.The next layer is a Flatten in layer in order to feed the result to a MLP Network with two Dense Layers of 128 neurons each and followed by a Batch Normalization and a dropout of 50%.The final layer is a Dense Layer composed of 13 neurons each one representing a different genres (action, adventure, animation, comedy, crime, drama, fantasy, horror, mystery, romance, sci-fi , short,thriller).

Although it is a multi-label classification, the activation function used is Sigmoid rather than Softmax.

Indeed, Softmax makes fluctuations on the accuracy of the model and moreover, a movie can belong to multiple genres i.e. the output classes are not mutually exclusive, that's why using sigmoid was preferable in this case.

Here is a simplified illustration of our model:



The 4 Convolutional Layers will extract the features. For example, the input picture that is shown on this representation is a associated to the genres : action and adventure. The Convolutional will identify patterns (such as the car, black color in the background, the persons faces) and will associate those features to action and adventure movies and that is how the network learns progressively how to recognize patters and make the right association as long as enough example were feed into the Network. The bottom line is that the model considers that certain objects appearing at a certain frequency, the type of the characters, maybe also certain colors ,will allow to the network to classify the poster and give a probability for each genre. An additional example is romance movies. On romance movie posters, we can see frienquently the two main characters (usually a man a woman), and that is a strong metric that the movie poster is a romance movie. Although movies can't belong to all the genres, the Network still gives a very small probability to unrelated genres (e.g. for a romance movie poster, the prediction rate for sci-fi will be at a very low rate, really close to 0).Given a poster x_i , the network outputs the probability vector $y_i = (y_{i,1}, y_{i,2}, y_{i,3} \dots, y_{i,m})$ where $m = 13$ i.e. 13 plausible genres and where $y_{i,j}$ represents the probability of the movie x_i being related to the movie genre j.The sigmoid activation function scales a probability in a range from 0 to 1 where values close to 0 mean that the probability for the movie to be of genre j is less possible and where values close to 1 or at least higher enough than 0 (0.3 , 0.4) means that the movie is more likely to be of genre j. After going through the sigmoid function, the network learned a little more about movie genres , and that is in the case of the training phase. During the test phase, the output vector is sorted by indexes where each index represent a different genre : [action, adventure, animation, comedy, crime, drama, fantasy, horror, mystery, romance, sci-fi , short,thriller].The 2 highest values are returned. Those values are integers ranging from 0 to 12. For example, if the 2 highest values are at index 3 and 9 respectively , then the genres at index 3 and 9 meaning "comedy" and "romance" will be returned and that is the prediction of the model.

It is usually easier to understand the architecture of the Network after breaking it down to smaller pieces. That is what we tried to show in the table below:

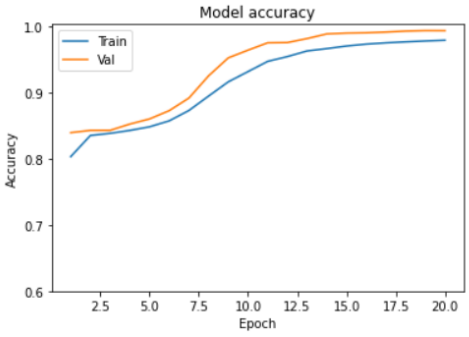
Detailed configurations of the proposed network.

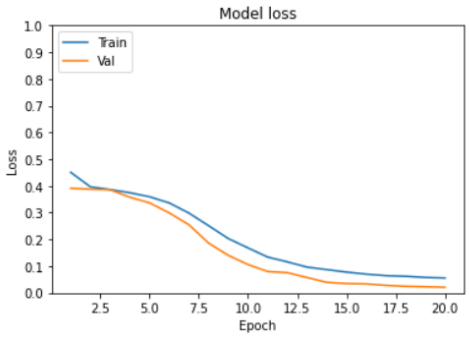
Input image (224x224x3)				
Part 1	Conv3-16 → Batch Normalization → MaxPooling2D(2x2) → Dropout (10%)	Conv3-32 → Batch Normalization → MaxPooling2D(2x2)	Conv3-64 → Batch Normalization → MaxPooling2D(2x2) → Dropout (20%)	Conv3-128 → Batch Normalization → MaxPooling2D(2x2)
Part 2	Flatten Layer → Dense Layer (128 neurons) → Dense Layer (128 neurons)			
Part 3	Output Layer (13 neurons – Sigmoid activation function)			

Con3-16 means a convolutional Layer of 16 filters, and each filter has a size of 3x3

4 Experiment/Simulation Results

4.1 Model Accuracy and Loss





The first plot shows the accuracy of the model. The blue line represents the evolution of the training data accuracy through the 20 epochs. The orange line represent the evolution of the validation data through the 20 epochs, we can see that the model accuracy reaches 97% on the training set.

The second plot shows the loss of the model. The blue line represents the evolution of the training loss through the 4 epochs. The orange line represents the evolution of the validation loss through the 20 epochs, we can see that the model loss reaches around 0.05% on the training set.

4.2 Test Output examples

4.2.1 The Network succeeded to predict the right genres

```
img = image.load_img('/content/love-romance-chocolate-movie-poster-md.jpg', target_size=(pic_width,pic_height, 3))
plt.imshow(img)
img = image.img_to_array(img)
img = img/255.0
img = img.reshape(1, pic_width, pic_height, 3)

genres = data.columns[6:]
y_prob = model.predict(img)
arr = np.array(y_prob)
top2 = np.argsort(y_prob[0])
print('The 2 genres predicted for the Movie "Love, Romance and Chocolate" are :')
for i in range(11,13):
  index = top2[i]
  print(genres[index])
print("The expected output was at least 'romance', hence the network succeeded")
```

```
The 2 genres predicted for the Movie "Love, Romance and Chocolate" are :
romance
drama
The expected output was at least 'romance', hence the network succeeded
```

4.2.2 The Network did not succeed to predict the right genres

```
img = image.load_img('/content/sci-fi-fantasy_0017_ebay_listing.jpg', target_size=(pic_width,pic_height, 3))
plt.imshow(img)
img = image.img_to_array(img)
img = img/255.0
img = img.reshape(1, pic_width, pic_height, 3)

genres = data.columns[6:]
y_prob = model.predict(img)
arr = np.array(y_prob)
top2 = np.argsort(y_prob[0])
print('The 2 genres predicted for the Movie "Avatar" are :')
for i in range(11,13):
  index = top2[i]
  print(genres[index])
print("The expected output was at least 'sci-fi', but it doesn't appear, hence the network didn't succeed to predict the right genre")
```

```
The 2 genres predicted for the Movie "Avatar" are :
thriller
horror
The expected output was at least 'sci-fi', but it doesn't appear, hence the network didn't succeed to predict the right genre
```

5 Previous Attempts

Our previous attempts were about the same subject (predicting something on a set of movies) but it was centered towards predicting movie ratings based on tags.

We mainly worked with two files. One file contains a list of movies with there ratings (the movies were rated on a scale from 0 to 5 by many users). The second file contains a list of tags (e.g. action, war, horror, fun, comedy, cold,…) , more exactly 1128 tags. For each tag there is a number, expressing how strong a tag is related to the movie. For example if the tag "action" is associated with the number 0.8, it means that the movie is related to action and that we are expecting to see "action" in the movie and if on the contrary for the same movie, the tag "horror" is associated to the number 0.001 , the meaning is that there a very low amount of elements in the movie so that we can't classify this movie as an horror movie (i.e. we won't see horror scenes in the movie)

The idea was to create a Neural Network that would be able to detect relationships between the tags and the attributed rating so that we could predict a movie rating, based on his tags. When the training phase is starting, the network is fed by 1128 numbers (tag numbers) that will get multiplied by random weights. As the training goes on, the use of gradient descent is used in order to minimize the loss and to update the weights. After training , we can now predict a the rating of a movie that the network never trained on and get an approximate rating usually close to the real one attributed by the user.

In conclusion, our previous attempts were very basic and taking a very few parameters compared to the main model presented is this paper.

6 Conclusion

We have presented a system to classify movie posters into genres. A Convolutional neural network is proposed to jointly consider visual appearance and object information. Multi-label classification is achieved by creating many convolution layers that have for mission to extract different features , create relations between those features and movie genres (back propagation helps to make this happen) and improves itself after that a consequent number of picture were presented to the Network and that the training happened on many epochs in order to reinforce the learning and adjust as much as possible. Whenever a new poster will be presented to the network, based on the training , an estimation will be given in probabilities.

ACKNOWLEDGMENTS

This work was done in the frame of the Course "Deep Learning and Natural Language Processing" with Dr. Amos Azaria. His classes and advices helped us realizing