

Graphical user interface, application

Description automatically generated

**PROJECT TITLE**

**MOVIE GENRE DETECTION SYSTEM**

**By**

**Amjad Abdullah Alqahtani, 437804250**

**Atheer Awad Alqadi, 437804251**

**Fawziah Tariq Misfer, 437804434**

**Ruba Ahmed Almohya, 437805795**

**Supervised by:**

**Prof. Talal Saeed Saleh Kaid**

**And**

**Dr. Abeer Abdulqawi Raweh Othman**

*Computer Science Department*

**This Project Documentation is submitted in partial fulfillment for the requirement for the Degree in *Computer Science***

**Academic Year 2020-2021/1441 (Semester-II (**

# COMMITTEE REPORT

# ABSTRACT

AI assists in every section of our lives. How not when the AI is concerned with making machine think and act like humans. the amount of data that is generated by humans and machines its huge to human’s ability to understand, interpret, and make complex decisions based on that data. So, the Artificial intelligence is important in every side of our life, like learning, health, industry and fun.

AI had a great role in the entertainment, games, movies because a machine is able to outperform humans in some "intellectual competencies", like knowing various information about the movies. In this aspect, the idea of our project generated simply, A new method appears to make it easier for us to know the movie that is we want information about it by take only shoots of movie poster, meaning it only shoots it and get everything about it.

The name of our project is "movie genre detection system" by using deep learning. We will use algorithms that is allowed to take shoot poster of the film then analysis and determine what type of the film and some information.

# DECLARATIONS

We hereby declare that the project entitled Movie Genre Detection System is entirely us

own work. We have not copied from any other sources except where due the references or

acknowledgement is made explicitly in the text, no has any part been written for us by another

person.

|  |  |
| --- | --- |
| A picture containing shape  Description automatically generatedName: Fawziah  Signature:  Date: 28 / 11 /2020 | Name: Amjad  Signature:  Date: 28 / 11 /2020 |
| A picture containing diagram  Description automatically generatedName: Ruba  Signature:  Date: 28 / 11 /2020 | Text, letter  Description automatically generatedText, letter  Description automatically generatedName: Atheer  Signature:  Date: 28 / 11 /2020 |

# ACKNOWLEDGEMENTS

Here we are approaching the end of the road, as we are approaching the achievement of our dreams that we have always dreamed of.

We would not have gotten here without those who encouraged us all the time and who lightened up our paths, to the most important men in our lives, our fathers.

To those who we rely on, those with generous hearts, who helped us along the way, our mothers.

To those who made a considerable effort to help us and were the best support for us, our brothers, sisters, and friends.

To everyone who contributed to our education throughout our lifetime.

To our country, which has always been our supporter, that always embraces us.

Thank you, Dr. Talal and Dr. Abeer, for your enormous help and effort.

CONTENTS

[COMMITTEE REPORT i](#_Toc68706006)

[ABSTRACT ii](#_Toc68706007)

[DECLARATIONS iii](#_Toc68706008)

[ACKNOWLEDGEMENTS iv](#_Toc68706009)

[LIST OF FIGURES viii](#_Toc68706010)

[LIST OF TABLES ix](#_Toc68706011)

[CHAPTER ONE: INTRODUCTION 1](#_Toc68706012)

[1.1 INTRODUCTION 1](#_Toc68706013)

[1.2 MOTIVATION 2](#_Toc68706014)

[1.3 PROBLEM STATEMENT 2](#_Toc68706015)

[1.4 PROJECT OBJECTIVES 2](#_Toc68706016)

[1.5 PROJECT SCOPE 2](#_Toc68706017)

[1.6 SUGGESTED SOLUTION 3](#_Toc68706018)

[1.7 PROJECT CONTRIBUTION 3](#_Toc68706019)

[1.8 PROJECT LIMITATION 3](#_Toc68706020)

[1.9 THESIS ORGANIZATION 3](#_Toc68706021)

[CHAPTER TWO: BACKGROUND AND LITERATURE REVIEW 4](#_Toc68706022)

[2.1 OVERVIEW 4](#_Toc68706023)

[2.2 MOVIE GENRE DETECTION SYSTEM 4](#_Toc68706024)

[2.2.1 Machine learning 4](#_Toc68706025)

[2.2.2 Deep learning 5](#_Toc68706026)

[2.3 EXISTING SYSTEM 5](#_Toc68706027)

[2.3.1 Shazam 5](#_Toc68706028)

[2.3.2 NatureID 6](#_Toc68706029)

[2.3.3 Foodvisor 7](#_Toc68706030)

[2.4 COMPARATION WITH RELEVANT SYSTEMS 7](#_Toc68706031)

[CHAPTER THREE: SYSTEM ANALYSIS & DESIGN 8](#_Toc68706032)

[3.1 OVERVIEW 8](#_Toc68706033)

[3.2 PROJECT METHODOLOGY 8](#_Toc68706034)

[3.2.1 Working of Deep Learning 8](#_Toc68706035)

[3.3 SYSTEM REQUIREMENTS 9](#_Toc68706036)

[3.3.1 Functional requirements 9](#_Toc68706037)

[3.3.2 Non-Functional requirements 9](#_Toc68706038)

[3.4 SOFTWARE AND HARDWARE REQUIREMENTS 10](#_Toc68706039)

[3.4.1 Software requirements 10](#_Toc68706040)

[3.4.2 Hardware requirements 10](#_Toc68706041)

[3.5 DEVELOPMENT METHODOLOGY 10](#_Toc68706042)

[3.5.1 Use Case Diagram 10](#_Toc68706043)

[3.5.2 Class Diagram 13](#_Toc68706044)

[3.5.3 Sequence diagrams 14](#_Toc68706045)

[1.5.4 Activity diagram 17](#_Toc68706046)

[1.6 INTERFACE DESIGN 18](#_Toc68706047)

[CHAPTER FOUR: IMPLEMENTATION 21](#_Toc68706048)

[4.1 OVERVIEW 21](#_Toc68706049)

[4.2 DATASET 21](#_Toc68706050)

[4.3 VIRTUAL ENVIRONMENT ON KAGGLE 23](#_Toc68706051)

[4.4 APPLICATION IMPLEMNTATION 24](#_Toc68706052)

[4.4.1 Main page 24](#_Toc68706053)

[4.4.2 Training page 25](#_Toc68706054)

[4.4.3 Testing page 26](#_Toc68706055)

[4.4.4 Predicting Page 27](#_Toc68706056)

[CHAPTER FIVE: TESTING 28](#_Toc68706057)

[5.1 SOFTWARE TESTING 28](#_Toc68706058)

[5.2 TEST CASE SPECIFICATION 28](#_Toc68706059)

[5.2.1 Training 28](#_Toc68706060)

[5.2.2 Testing 28](#_Toc68706061)

[5.2.3 predicting 28](#_Toc68706062)

[5.3 VALIDATION 29](#_Toc68706063)

[CHAPTER 6: CONCLUSION AND FUTURE WORK 32](#_Toc68706064)

[6.1 CONCLUSION 32](#_Toc68706065)

[6.2 FUTURE WORK 32](#_Toc68706066)

[APPENDIX I: PROJECT PLAN 33](#_Toc68706067)

[GANTT CHAR 34](#_Toc68706068)

[PROJECT ORGANIZATION 35](#_Toc68706069)

[Internal Structure 35](#_Toc68706070)

[Rolls and Responsibilities 35](#_Toc68706071)

[Appendix II: Coding 36](#_Toc68706072)

[REFERENCE 52](#_Toc68706073)

# LIST OF FIGURES

[Figure ‎1‑1 Deep learning structure 2](file:///C:\Users\DELL\Desktop\project\project1.docx#_Toc57232799)

[Figure 2- 1 shazam 5](file:///C:\Users\DELL\Desktop\project\project1.docx#_Toc57232796)

[Figure 2- 2 NatureID 6](file:///C:\Users\DELL\Desktop\project\project1.docx#_Toc57232797)

[Figure 2- 3 Foodvisor 7](file:///C:\Users\DELL\Desktop\project\project1.docx#_Toc57232798)

[Figure 3- 1 Deep Learning 8](file:///C:\Users\DELL\Desktop\project1.docx#_Toc58440988)

[Figure 3- 2 Use case 11](#_Toc58440989)

[Figure 3- 3 class diagram 14](file:///C:\Users\DELL\Desktop\project1.docx#_Toc58440990)

[Figure 3- 4 Sequence diagrams for training 15](file:///C:\Users\DELL\Desktop\project1.docx#_Toc58440991)

[Figure 3- 5 Sequence diagrams for testing 16](#_Toc58440992)

[Figure 3- 6 Sequence diagrams for predicating 16](#_Toc58440993)

[Figure 3- 7 activity diagram for training 17](file:///C:\Users\DELL\Desktop\project1.docx#_Toc58440994)

[Figure 3- 8 activity diagram for testing 18](#_Toc58440995)

[Figure 3- 9 activity diagram for predicating 18](#_Toc58440996)

[Figure 3- 10 main interface 18](#_Toc58440997)

[Figure 3- 11 training interface 19](#_Toc58440998)

[Figure 3- 12 testing interface 19](#_Toc58440999)

[Figure 3- 13 predicating interface 20](#_Toc58441000)

[Figure 4- 1 image folder 21](file:///C:\Users\DELL\Desktop\project\لقيتها%20سطح%20مكتب\project1%20(4).docx#_Toc68459376)

[Figure 4- 2 data folder 22](file:///C:\Users\DELL\Desktop\project\لقيتها%20سطح%20مكتب\project1%20(4).docx#_Toc68459377)

[Figure 4- 3 training dataset 22](file:///C:\Users\DELL\Desktop\project\لقيتها%20سطح%20مكتب\project1%20(4).docx#_Toc68459378)

[Figure 4- 4 test dataset 23](file:///C:\Users\DELL\Desktop\project\لقيتها%20سطح%20مكتب\project1%20(4).docx#_Toc68459379)

[Figure 4- 5 Virtual environment on Kaggle 23](#_Toc68459380)

[Figure 4- 6 Execution of the code 24](#_Toc68459381)

[Figure 4- 7 main page UI 24](#_Toc68459382)

[Figure 4- 8 training UI 25](#_Toc68459383)

[Figure 4- 9 testing UI 26](#_Toc68459384)

[Figure 4- 10 predicting UI 27](#_Toc68459385)

[Figure 5- 1 accuracy of training in CNN **Error! Bookmark not defined.**](file:///C:\Users\DELL\Desktop\project\لقيتها%20سطح%20مكتب\project1%20(4).docx#_Toc68459326)

[Figure 5- 2 Complete testing 30](file:///C:\Users\DELL\Desktop\project\لقيتها%20سطح%20مكتب\project1%20(4).docx#_Toc68459327)

[Figure 5- 3 Complete predicting 31](file:///C:\Users\DELL\Desktop\project\لقيتها%20سطح%20مكتب\project1%20(4).docx#_Toc68459329)

# LIST OF TABLES

[Table 1 comparison 7](#_Toc68385655)

[Table 2 Software requirements 10](#_Toc68385656)

[Table 3 Hardware requirements 10](#_Toc68385657)

[Table 4 Read images UC Description 12](#_Toc68385658)

[Table 5 chose technique UC Description 12](#_Toc68385659)

[Table 6 training UC Description 12](#_Toc68385660)

[Table 7 training UC Description 12](#_Toc68385661)

[Table 8 load file UC Description 12](#_Toc68385662)

[Table 9 load model UC Description 13](#_Toc68385663)

[Table 10 testing UC Description 13](#_Toc68385664)

[Table 11 load image UC Description 13](#_Toc68385665)

[Table 12 predicting 13](#_Toc68385666)

[Table 13 training Case Specification 28](#_Toc68385667)

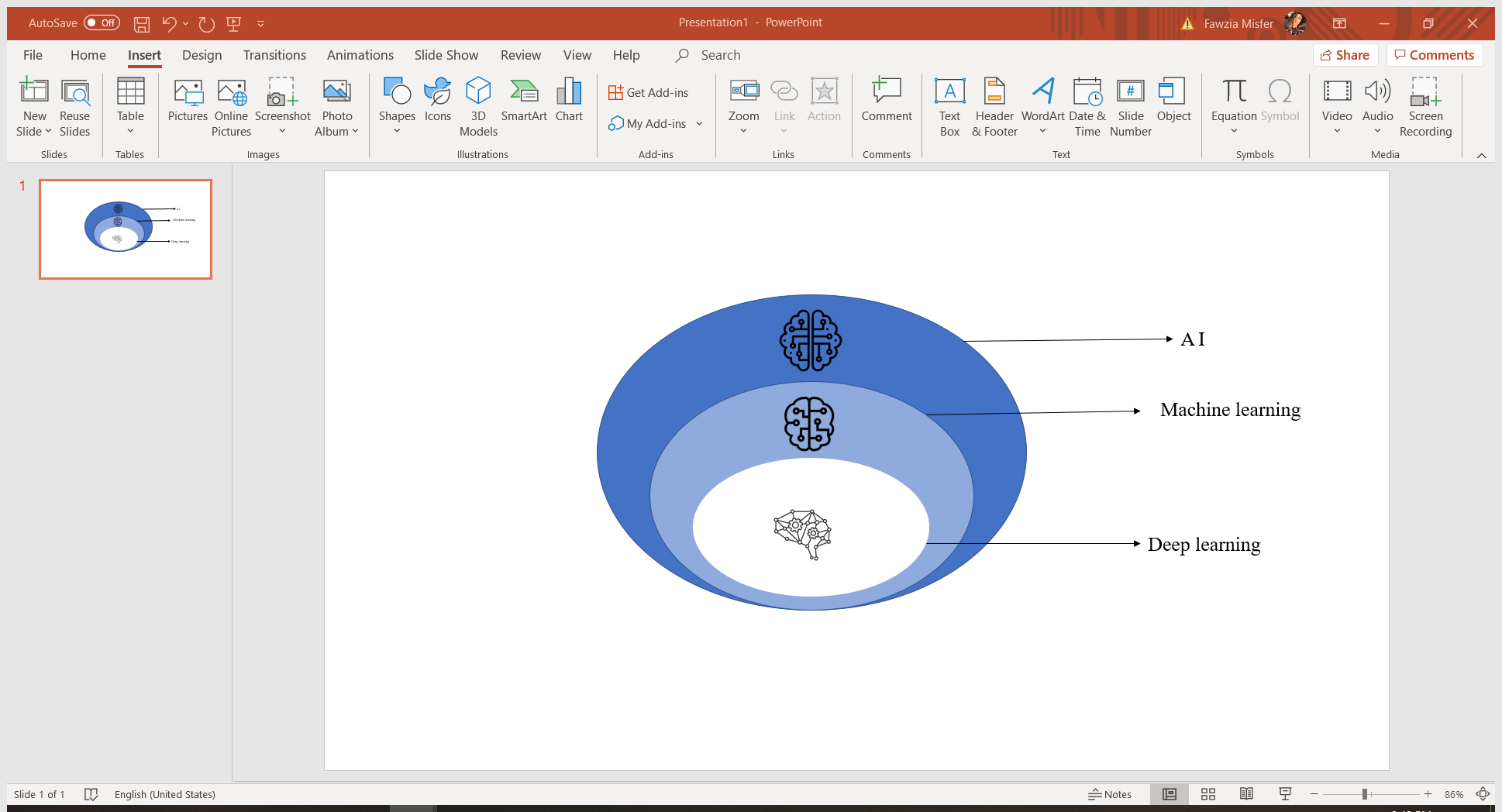
[Table 14 test Case Specification 28](#_Toc68385668)

[Table 15 predict Case Specification 28](#_Toc68385669)

# CHAPTER ONE: INTRODUCTION

## INTRODUCTION

John McCarthy coined the concept Artificial Intelligence in 1956. He described it as the science and engineering of making intelligent machines.AI is the computer science branch that deals with the study and design of smart agents that perceive their environment and take actions that maximize their chances of success. AI can be described as the ability to keep two different ideas in mind at the same time and remain the ability to function. AI, however, needs to include learning from past experience, reasoning for decision-making, power of inference, and rapid response. It must also be capable of making choices based on priorities and resolving complexity and ambiguity. Machines programmed to perform tasks that would require intelligence to be performed by humans are said to possess artificial intelligence. The scientific objective of AI is to understand intelligence by designing computer programs that use symbolic inference or logic within the system to exhibit intelligent behavior. The concept of AI is not independent of time. By holding time in mind, it gives the decision of any method. [1] In this project we will use the term deep learning - shows in figure 1- : It is a technique for machine learning that aims to solve problems by constructing layer-wise models with many levels of increasing abstraction. That is, it helps multi-layer computer models to learn datasets with multiple hidden layers. These techniques have significantly enhanced the recognition of state-of-the-art expression, the recognition of visual artifacts, object identification, and many other areas such as drug discovery and genomics. By using the backpropagation algorithm to show how a computer can alter its internal parameters that are used to compute the representation in and layer from the representation in the previous layer, deep learning discovers intricate structure in large data sets. In the processing of images, video, voice, and audio, deep convolutional networks have brought breakthroughs, while recurrent networks have shone a light on sequential data such as text and speech. [2]



**Figure 1‑1 Deep learning structure**

## MOTIVATION

As we know that watching movies is part of human life and that technology has become necessary in all things, we decided to develop a system that allows the user to know the appropriate classification of films by analyzing images. And to enable the user to the quick search process and ease of access.

## PROBLEM STATEMENT

Many people are attracted to a poster, but when he finishes watching the movie, he puts a bad rating because he does not have enough information about the type of film or the appropriate age. This is very important information before choosing a movie in the movie theater or when purchasing a movie.

## PROJECT OBJECTIVES

* Design a method that simulates artificial intelligence to reveal movie information.
* Add a new search method for movies.
* Motivate people with technology and artificial intelligence to help them facilitate standard research methods into smart methods.

## PROJECT SCOPE

The scope will be in movies’ world. When the project is completed, it will be able to provide an accurate, modern, and easy way to search for films.

## SUGGESTED SOLUTION

In this project, the movie genre detection system we will doing a system to Identify posters films, getting information, and offering by application for users. We propose a method supported by a CNN architecture to classify movie image according to their genres, CNNs are the new method for image classification, using concepts from image processing to ensure some degree of position, scale, and distortion invariance. Our contributions in this project are to make available a movies image dataset, which includes more than 8000 images that belong to one of the following genres: action, comedy, horror, or drama ...etc.

To construct this system, the software python is used to write the code and MYSQL database to build the images of our system.

## PROJECT CONTRIBUTION

1. We proposed a system that can detect the film from photo.
2. The system will not be limited and focus only on the photo, we plan to add more objects like age rate and kind of the film.
3. Taking in consideration the new films, so there will be a permanent update of the system.
4. The possibility of developing this project in the future and using it with developments around us.

## PROJECT LIMITATION

* We will focus on the images, and how to make the machine recognize it.
* We got data that contains 8000 images for different movies.
* We will focus on the posters of movies just.

## THESIS ORGANIZATION

**Chapter 1:** This chapter is an introduction to our project idea.

**Chapter 2:** This chapter talks about the background and literature review, and the basic technology used in our project.

**Chapter 3:** This chapter talks about System Specifications and Analysis.

**Chapter 4**: In this chapter we will show the implementation of our project.

**Chapter 5:** This chapter reviews the project to ensure that the application works correctly and that the purpose of this project is fulfilled.

**Chapter 6:** This chapter is the project's conclusion, and what we plan to focus on in the future.

# CHAPTER TWO: BACKGROUND AND LITERATURE REVIEW

## 2.1 OVERVIEW

This section provided background details about the domain of our project. We will get an overview of Movie Genre Detection and a simplified concept of deep learning and machine Learning. And we will discuss about existing systems similar in some feature of our system. Then last section is comparation with the relevant system.

## 2.2 MOVIE GENRE DETECTION SYSTEM

In Artificial Intelligence, systems are designed to perform specific tasks that are used for a wide variety of activities, including the identification of film genres, which we will talk about in our project. We propose to achieve movie genre detection analysis based simply on movie poster images, which means classify a given movie poster image into genres. Because a movie may associate with multiple genres, this is a multi-label image classification problem. So, we collect a large movie poster dataset. Based on this dataset, A deep neural network (NN) is constructed to describe object information. And the movie genre detection systems are using machine learning and deep learning techniques.

### Machine learning

Machine learning algorithms are built to predict computers and achieve tasks faster and with very high quality using a group of pictures.

There are four ways of classification of Machine Learning:

* + 1. Binary: have two class labels.
    2. Multi-Class: have more than two class labels.
    3. Multi-Label: have two or more class labels, where one or more class labels may be predicted for each example. Therefore, we will use this Classification task in our project.

Consider the example of poster of movie, where a given photo may have multiple objects in the scene and a model may predict the presence of multiple known objects in the photo, such as “house” “tree,” “person,” etc.

* + 1. Imbalanced: the number of examples in each class is unequally distributed.

### 2.2.2 Deep learning

Recent advances in machine learning underpin a collection of algorithms with an impressive ability to decipher the content of images. These deep learning algorithms are being applied on many sides.

We can train convolutional neural networks from scratch or use pre-trained networks to quickly learn new tasks or create new deep networks for image classification.

Fine-tuning a pre-trained image classification network with transfer learning is typically much faster and easier than training from the beginning.

Using pre-trained deep networks enable to quickly learn new tasks without defining and training a new network, having millions of images, or having a powerful GPU.

After defining the network architecture, we must define training parameters using the training options function. We can then train the network using train network. Use the trained network to predict class labels or numeric responses images and are transforming the analysis and interpretation of imaging data.

We can train a convolutional neural network on a CPU, a GPU, multiple CPUs, or GPUs, or in parallel on a cluster or in the cloud. Training on a GPU or in parallel requires Parallel Computing Toolbox™. Using a GPU requires a CUDA® enabled NVIDIA® GPU with compute capability 3.0 or higher. Specify the execution environment using the training Options function.[2]

## 2.3 EXISTING SYSTEM

Many concepts for detection systems have appeared in the literature in recent years that use methods of image processing and machine and deep learning techniques to predict kind items.

Some systems or approaches below note that the characteristics in most of them may be close to what we want to build in our system:

### 2.3.1 Shazam Shazam - Music Discovery, Charts & Song Lyrics

**Figure 2- 1 shazam**

An American application that is owned by Apple Inc. The application may detect songs, movies, ads, and television shows based on a brief sample being played and using the microphone on the computer. Android, MacOS, WatchOS, iOS, and Windows are compatible with this program. Shazam recognizes the songs based on the audio fingerprint and also based on the time-frequency graph that defined a spectrogram.

It uses a smartphone or computer's built-in microphone to capture a brief sample of the audio being played. A list of audio fingerprints is recorded by Shazam in a database. The user checks a song for 10 sec and the Shazam application makes an audio fingerprint. Shazam works by analyzing the recorded sound in a database of millions of songs and finding a match based on an acoustic fingerprint. If a match is identified, details such as the title of the single, artist and album will be sent back to users. [3]

### 2.3.2 NatureID

 This app will recognize the breed of the dog or cat you have protected by NatureID and provide you with some useful information about them.

**Figure 2- 2 NatureID**

So, if you are not sure of a dog or cat's breed, just concentrate your camera on the item you are interested in and take a picture.

Using the smart plant and animal identifier app to explore the wonderful world of nature This app has some advanced features:

* Identify with up to 95 percent accuracy over 10,000 species.
* Recognize over 108 breeds of cat and over 487 races of dogs.
* To get the most accurate plant and pet identification ever, our recognition algorithm has been dramatically enhanced.
* Get an overview of every dog, cat, and plant.
* Take images from your computer or upload pictures.
* Diagnose a plant: take photos of the symptoms of the plant and get a complete explanation of the disease, as well as recommendations for proper care.
* Name search-by entering their names, find a plant species or a pet breed easily.
* Keep track of the plants and pets you are adding to your favorites.
* Section Insights-read entertaining and helpful articles on plants and pets.[4]

### 2.3.3 Foodvisor

 A company has developed a mobile app that helps you to record what you eat in order to lose weight, follow a diet, or get healthier. You can upload files by taking a picture of your plate before you eat. The firm uses deep learning to allow image recognition to detect what you are about to eat. In addition to identifying the kind of food, the app tries to estimate the weight of each item. Using camera autofocus info, Foodvisor seeks to determine the distance between your plate and your screen. It then calculates each food item's area. Depending on the form of food, the business then attempts to extrapolate the volume of each object. And, if you have anything wrong with Foodvisor, you can manually fix it before logging your meal, you can manually correct it. Since it is too demanding, many individuals give up on diet trackers. The Foodvisor technology is all about making the process of data entry as smooth as possible. After that, for what you just ate, you get a rundown of nutrition facts, calories, proteins, carbohydrates, fats, fibers, and so on.[5]

**Figure 2- 3 Foodvisor**

## 2.4 COMPARATION WITH RELEVANT SYSTEMS

The following table 1 shows a comparison between the relevant system and the proposed system:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Factors | Our project | Shazam | NatureID | Foodvisor |
| Domain | Movies | Sound | Plants, cats, dogs | Healthy eats |
| The Technique Used | Deep learning | ML | Deep learning | Deep learning |
| Accuracy  Results | 92% | 95.71% | 95% | 91.5% |

**Table 1 comparison**

# CHAPTER THREE: SYSTEM ANALYSIS & DESIGN

## 3.1 OVERVIEW

In this chapter, the project methodology to be followed in this project, Functional and non-functional system specifications, Implementation Methodology, will be decided (use case diagram, class diagram, sequence diagram, and activity diagram). Also, interfaces.

## 3.2 PROJECT METHODOLOGY

Deep learning is the technique of this project that will be followed during implementation.

### 3.2.1 Working of Deep Learning

Deep learning using a neural network .in deep learning we have three phases know as layer:

Input, hidden, output.

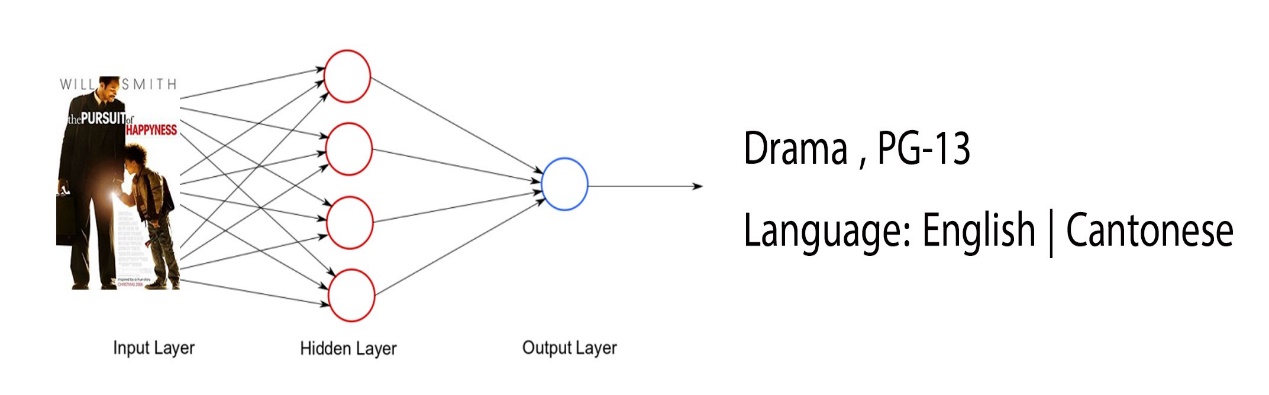
* Input means input data is processed in our project data is poster film

And also input layer has one neuron for each poster.

* Get information from input layer, apply mathematical computations, that is the mean of hidden.

Deciding the number of hidden layers and the number of neurons for each layer is one of the difficulties in constructing neural networks.

* Output means return output data and in our project is genre. [6]



**Figure 3- 1 Deep Learning**

**Advantages of Deep learning:**

* + From its own memory it can get a new image.
  + It can adjust automatically to all data such as image, voice, etc.
  + It gets its results more quickly.
  + It can learn over time through billions of images and recognize patterns.
  + It can manage large amounts of data for small networks with a much lower learning cost.
  + It can be for one system or more than one (any kind of systems), such as just a face recognition, or for an image reconstruction.

**Disadvantages of Deep learning:**

* + It is not efficient 100% and it will have some hard problems.
  + It does not give us accurate data. It is just approximate statistics.
  + It needs huge data sets to train.
  + It is very hard to understand and to describe and is not completely understood.
  + It is requiring a large amount of memory and computational resources because computationally very expensive.
  + It takes time because it requires to train the model to learn about deep structures, a process which requires billions of hours of computation.
  + It tends to be more costly.

## 3.3 SYSTEM REQUIREMENTS

### 3.3.1 Functional requirements

1. Read image correctly.
2. Extracting the most important features that help us in the process of discrimination.
3. Learning using deep learning.
4. Testing.

### 3.3.2 Non-Functional requirements

1. Efficiency: The system must response very fast, this will show high performance of the movie genre detection system.
2. Scalability: is how well the system deals with increasing use or size of data. This means that as much as data and new users entered in the system, its performance must stay the same.
3. Availability: The system availability and connecting must be all the time.
4. Maintainability: This is the simplicity with which we can modify the system. The harder it is to make a modification, the lower the maintenance power.

## 3.4 SOFTWARE AND HARDWARE REQUIREMENTS

### Software requirements

* Language: Python language.
* Virtual Environment: Kaggle.

|  |  |
| --- | --- |
| Software | Usage |
| Python | The Python language is used to build the system, and it was chosen because it is simple to program and offers many useful built-in functions. |
| Virtual  Environment on  Kaggle | The Kaggle site allows users to run programs in a simulated world and offers a range of helpful services to run the software faster, such as the GPU. |

**Table 2 Software requirements**

### 3.4.2 Hardware requirements

|  |  |
| --- | --- |
| Hardware | Specification |
| Computer | Processor: core i5 or more  Processor speed: 2GHz  RAM: 8GB or more  Hard disk: 250GB or more  GPU: NVIDIA |

**Table 3 Hardware requirements**

## DEVELOPMENT METHODOLOGY

### Use Case Diagram

The basic form of all requirements for any system or software for a program considered new and not developed is the use case.

We use two behaviors in it:

1- Expectant behavior (what?)

2- Unexpected behavior (how?)

Display relation between actor, function, and system. It, in turn, ignores showing order performance. It is designed from an end-user perspective. [7]

**Advantages of Use Case Diagram:** [7]

* It is easy to be understood by stakeholders because it is made of narrative text.
* It is a technique to know the system’s functional needs.
* It is used to get the basis like estimating, planning, and ensuring that the results are correct.
* It describes all software systems in a single example.

**Disadvantages of Use Case Diagram:** [7]

* The support of software tools that are required is limited.
* It is difficult to describe the geometric and temporal details.
* Not easy to know the non-functional specifications.

Diagram

Description automatically generated

**Figure 3- 2 Use case**

|  |  |
| --- | --- |
| Use case | Read images |
| Actor | User |
| Pre-condition | Dataset is available |
| Flow of event | The user will insert the poster of film for training process |

**Table 4 Read images UC Description**

|  |  |
| --- | --- |
| Use case | Chose technique |
| Actor | User |
| Pre-condition | Posters were inserted by user |
| Flow of event | The user will select one of the deep learning techniques (CNN, transfer learning) to do the training process and get the accuracy |

**Table 5 chose technique UC Description**

|  |  |
| --- | --- |
| Use case | Training |
| Actor | User |
| Pre-condition | User had already inserted posters and select techniques |
| Flow of event | On the given dataset, the user will begin the training phase |

**Table 6 training UC Description**

|  |  |
| --- | --- |
| Use case | Training |
| Actor | User |
| Pre-condition | The phase of training is completed |
| Flow of event | The user will save the model for the testing and the prediction processes |

**Table 7 training UC Description**

|  |  |
| --- | --- |
| Use case | Load file |
| Actor | User |
| Pre-condition | User already have posters for testing |
| Flow of event | The user will select file for testing |

**Table 8 load file UC Description**

|  |  |
| --- | --- |
| Use case | Load Model |
| Actor | User |
| Pre-condition | The model has saved |
| Flow of event | The user will select Model for testing |

**Table 9 load model UC Description**

|  |  |
| --- | --- |
| Use case | Testing |
| Actor | User |
| Pre-condition | The file and model already selected |
| Flow of event | The user will start the testing process to provide the accuracy |

**Table 10 testing UC Description**

**Table 11 load image UC Description**

|  |  |
| --- | --- |
| Use case | Load image |
| Actor | User |
| Pre-condition | User already have posters for predicting |
| Flow of event | The user will select image for predicting |

|  |  |
| --- | --- |
| Use case | Predicting |
| Actor | User |
| Pre-condition | User selected image |
| Flow of event | The user will start the predicting process to predict the type of film on the given image by using the selected model with the accuracy |

**Table 12 predicting**

### Class Diagram

Class diagrams are used when developing a system model.

It shows the classes in this system model and the relationships and associations between these classes.[7]

**Advantages of Class Diagram: [8]**

1. Helps to understand easier and clarify data models for an overview of plans and operations, whether simple or complex.
2. A better and easier understanding of the implementation plan.

**Disadvantages of Class Diagram: [8]**

1. Class diagrams take longer to build and maintain, which is annoying.
2. **Chart, diagram, box and whisker chart

   Description automatically generated**Lack of clarity to understand the beneficiary of the overly complex class diagram is also a defect.

**Figure 3- 3 class diagram**

### Sequence diagrams

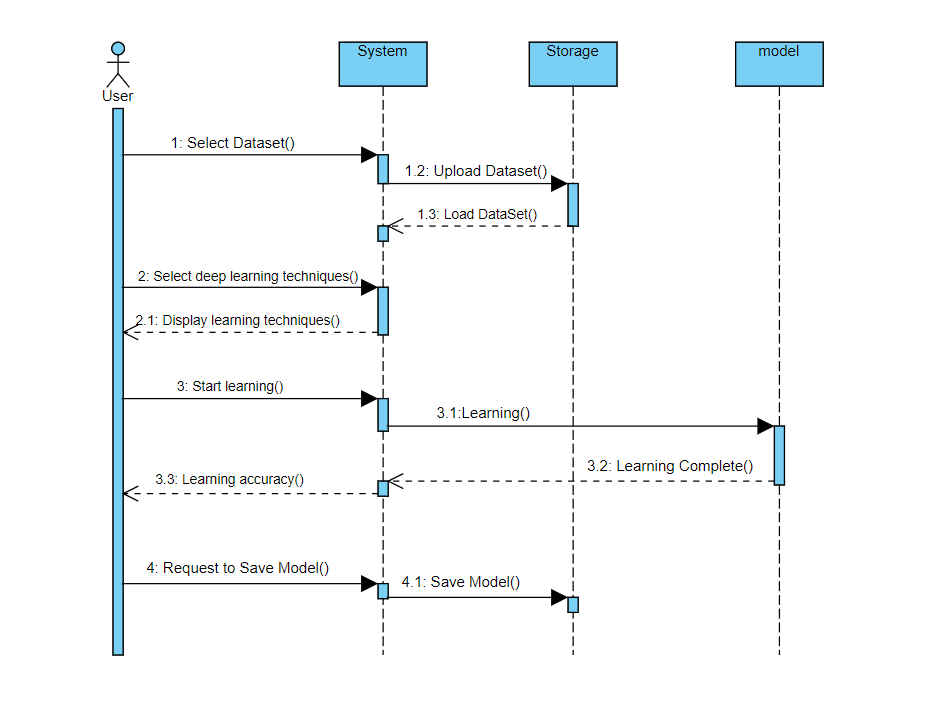
Fundamentally in the UML, to model the interactions and relationships between actors and also the system and its components, we use sequence diagrams.

And in turn, it shows the sequence of these interactions that can occur in the use case or use case instance. [7]

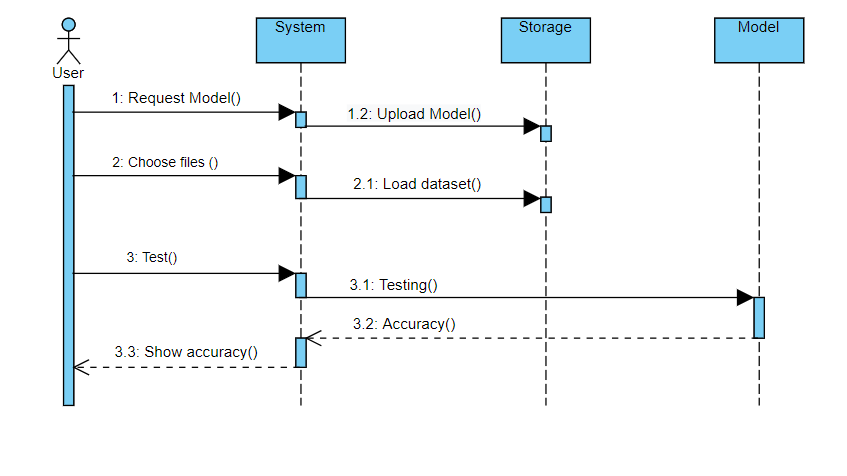
**Advantages of Sequence Diagram:[9]**

1. Represents the details in the use case.
2. Clarifies and assists in understanding and planning the detailed functions of a current or future scenario.
3. It shows us how the interaction between objects and components is as they interact with each other to complete the process.

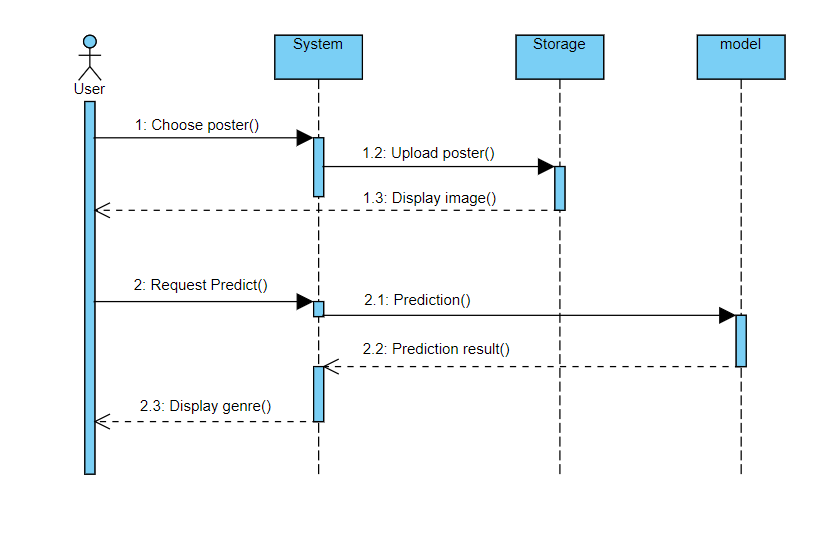
**Disadvantages of Sequence Diagram: [10]**

1. A sequence diagram is a snapshot between two-time frames. to some extent an "incomplete" diagram. It does not mean that a complete process or process is displayed, as shown in the activity diagram or use case, from beginning to end. So, these diagrams cannot be used for complete process documentation.

**Figure 3- 4 Sequence diagrams for training**



**Figure 3- 5 Sequence diagrams for testing**



**Figure 3- 6 Sequence diagrams for predicating**

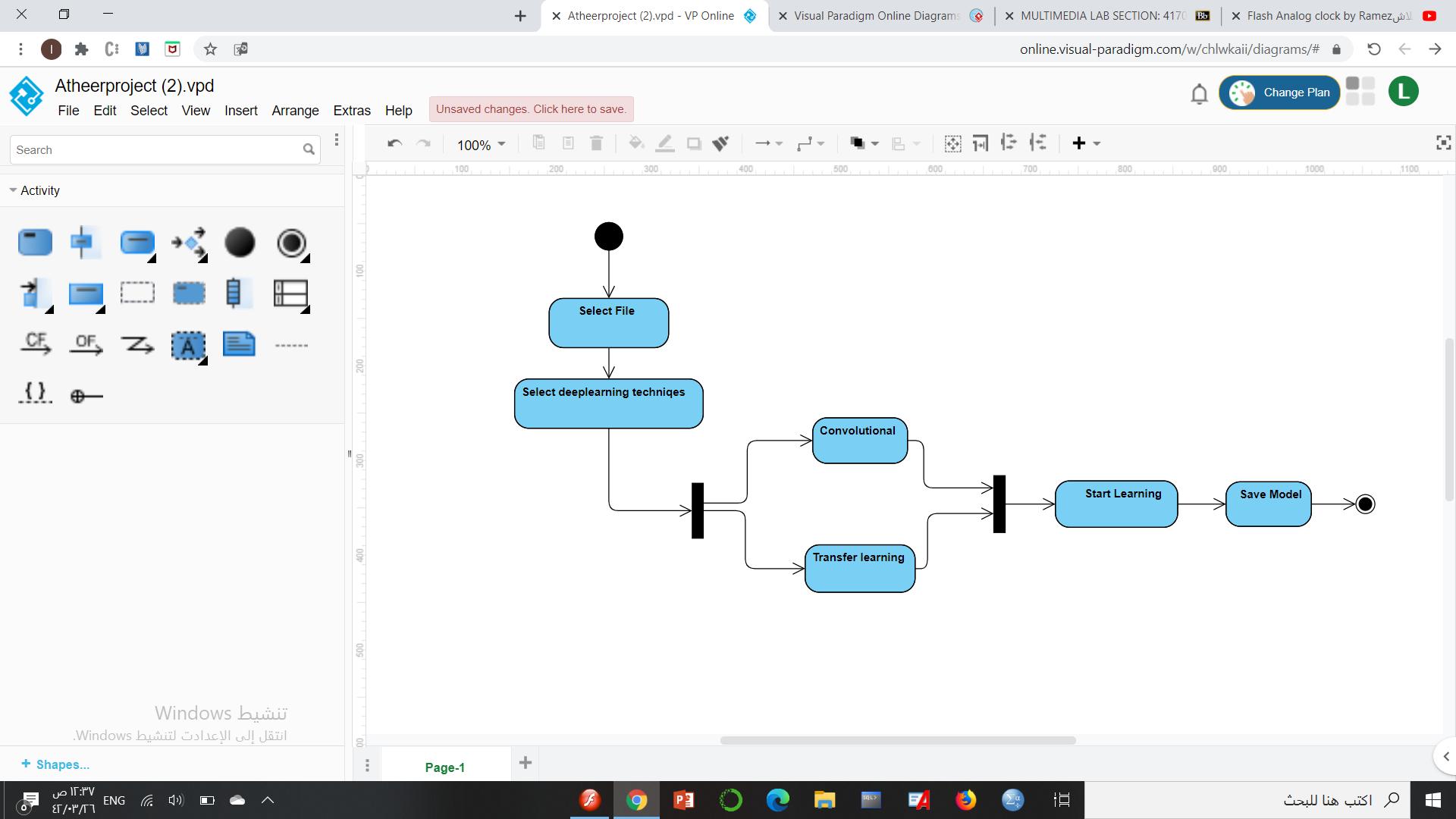
### Activity diagram

Explains the system's dynamic aspects. The flow from an activity to another activity is represented. It is possible to describe the activity as a system operation. [7]

**Activity diagrams’ advantages:** [7]

1. Normally easily comprehensible for both analysts and stakeholders.
2. for analyst’s activity diagrams allow show various conditions and actors within a workflow through the use of swim lanes. Swim lanes are optional as a single condition or actor is normally displayed without them.
3. Describing a complicated sequential algorithm.
4. Dealing with multithreaded applications.

**Activity diagrams’ disadvantages:** [7]

1. Activity diagram: does not give details about the object behavior or the way objects that collaborate.
2. If the project large it will be complex to draw a single activity diagram.

**Figure 3- 7 activity diagram for training**

Diagram

Description automatically generated

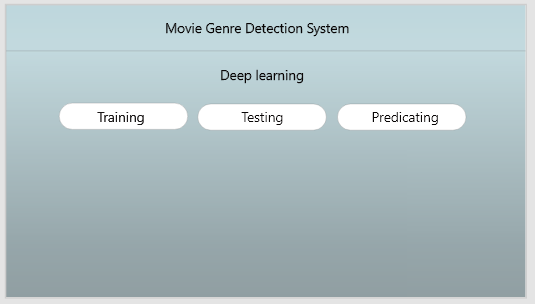
**Figure 3- 8 activity diagram for testing**

Diagram

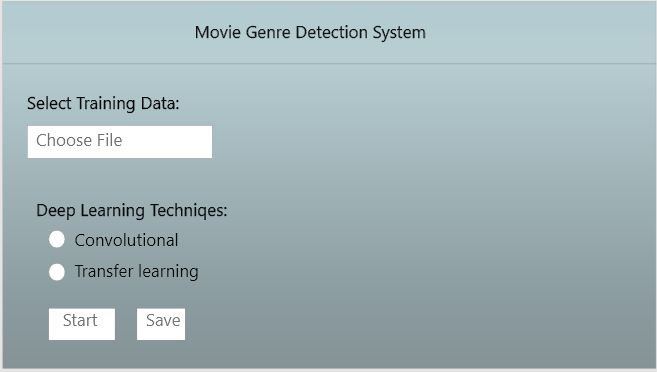
Description automatically generated

**Figure 3- 9 activity diagram for predicating**

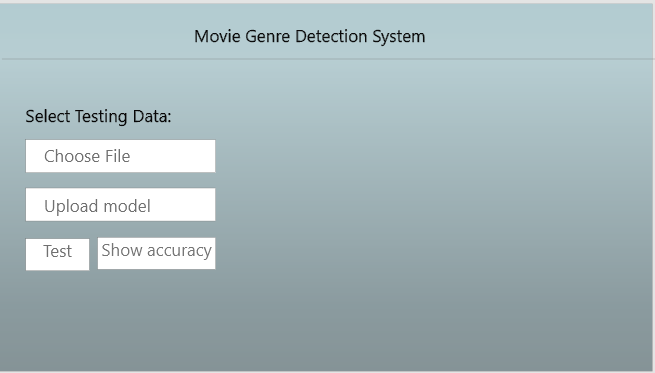
## INTERFACE DESIGN



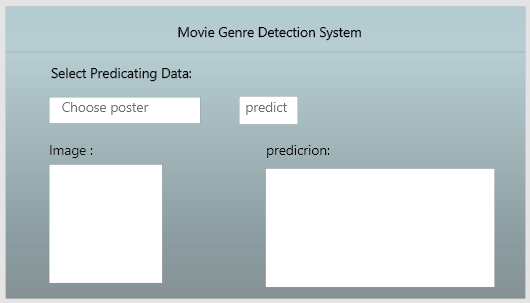
**Figure 3- 10 main interface**



**Figure 3- 11 training interface**



**Figure 3- 12 testing interface**



**Figure 3- 13 predicating interface**

# CHAPTER FOUR: IMPLEMENTATION

## 4.1 OVERVIEW

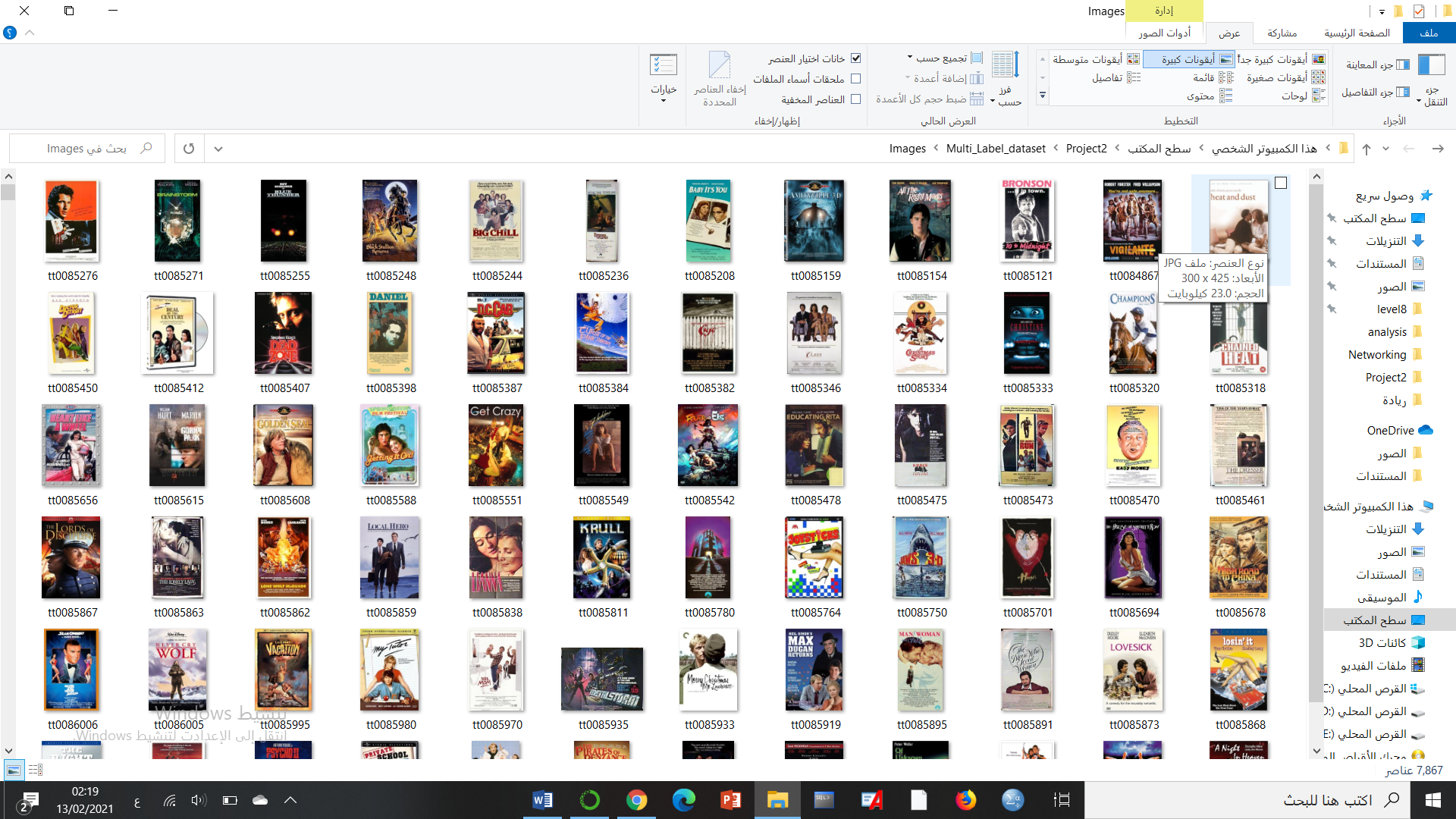
At the implementation stage, we will start creating the models to make it learning on our dataset.

We will create the interface of our system by using python language.

## 4.2 DATASET

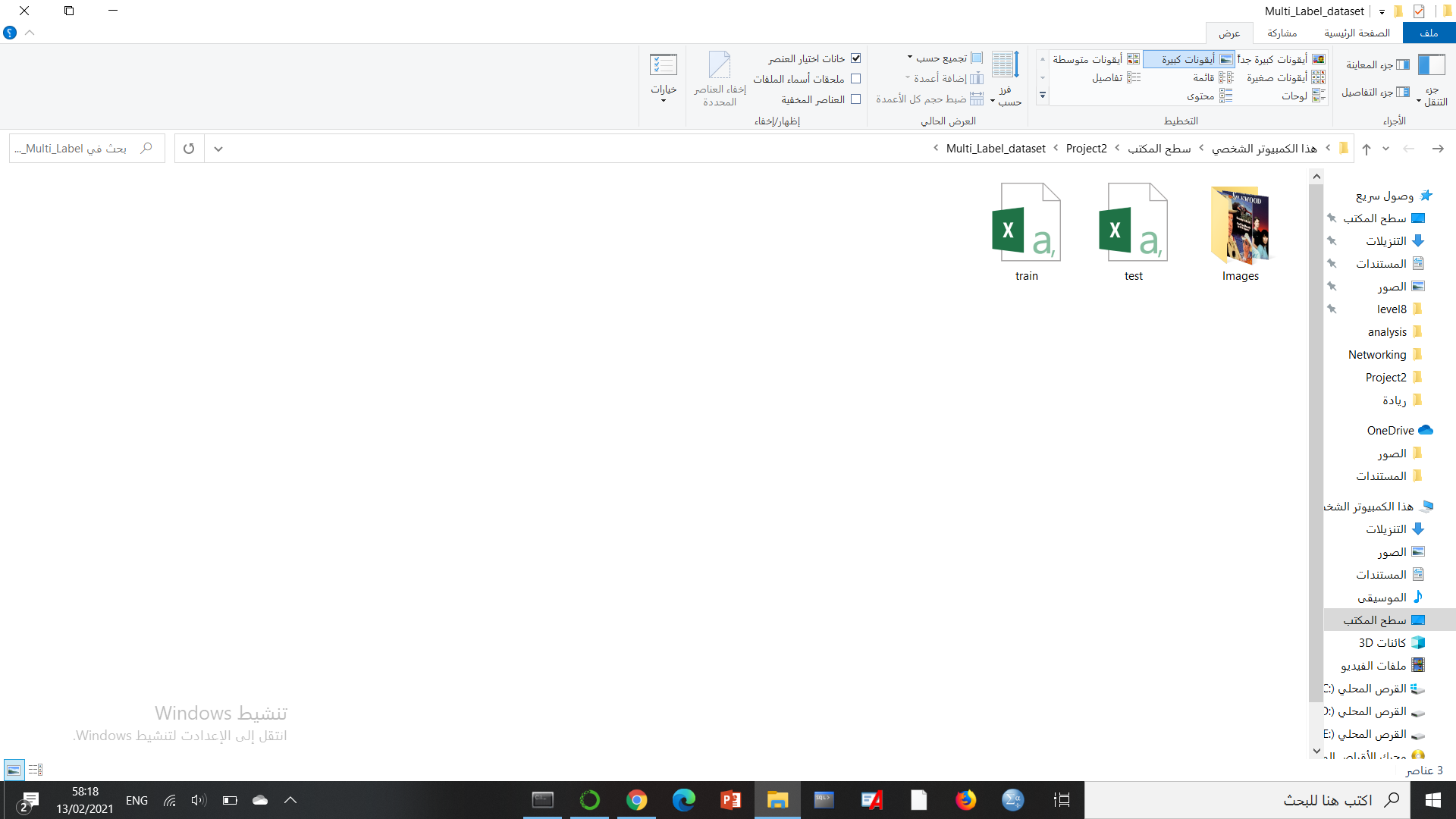
To evaluate and train our model we need a dataset that contains images.

Our dataset contains 7867 images and its notion with 25 classifications, and we get this dataset from Kaggle.

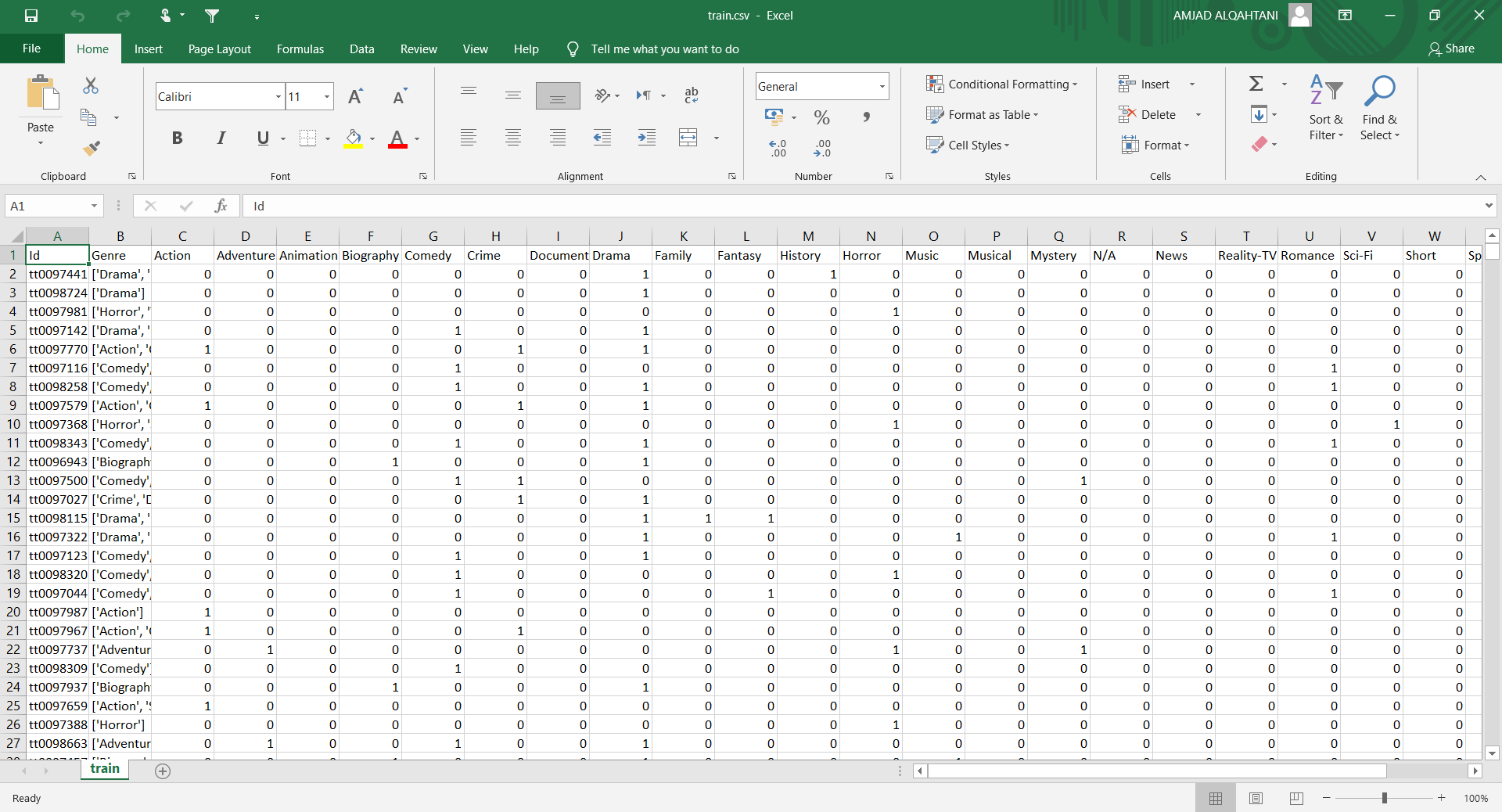
Here we have samples of the image posters:

**Figure 4- 1 image folder**

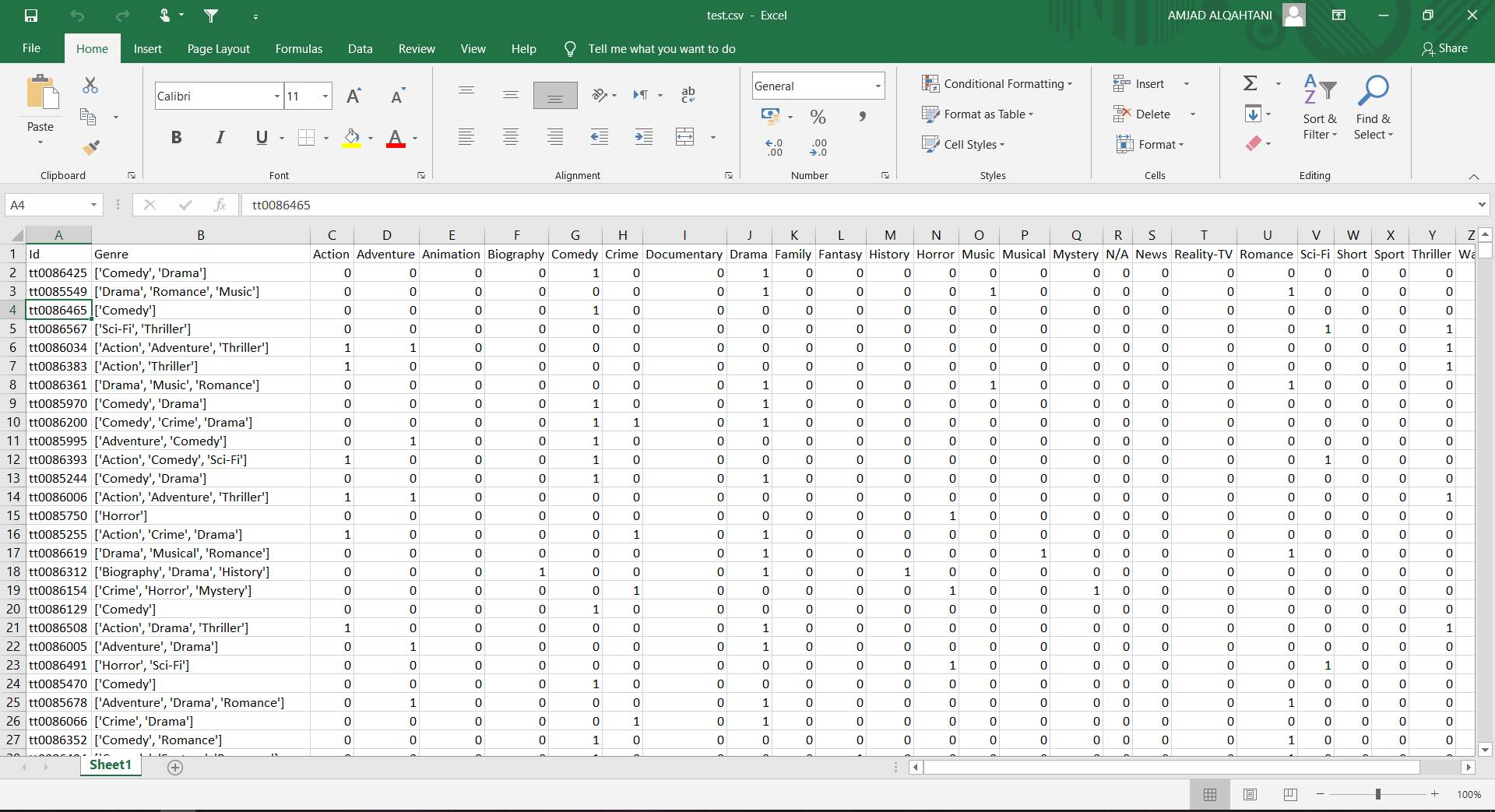
We will use 90% for the training process to make the model learning on our dataset and 10% for a test and prediction. The dataset for test and prediction is not seen.



**Figure 4- 2 data folder**



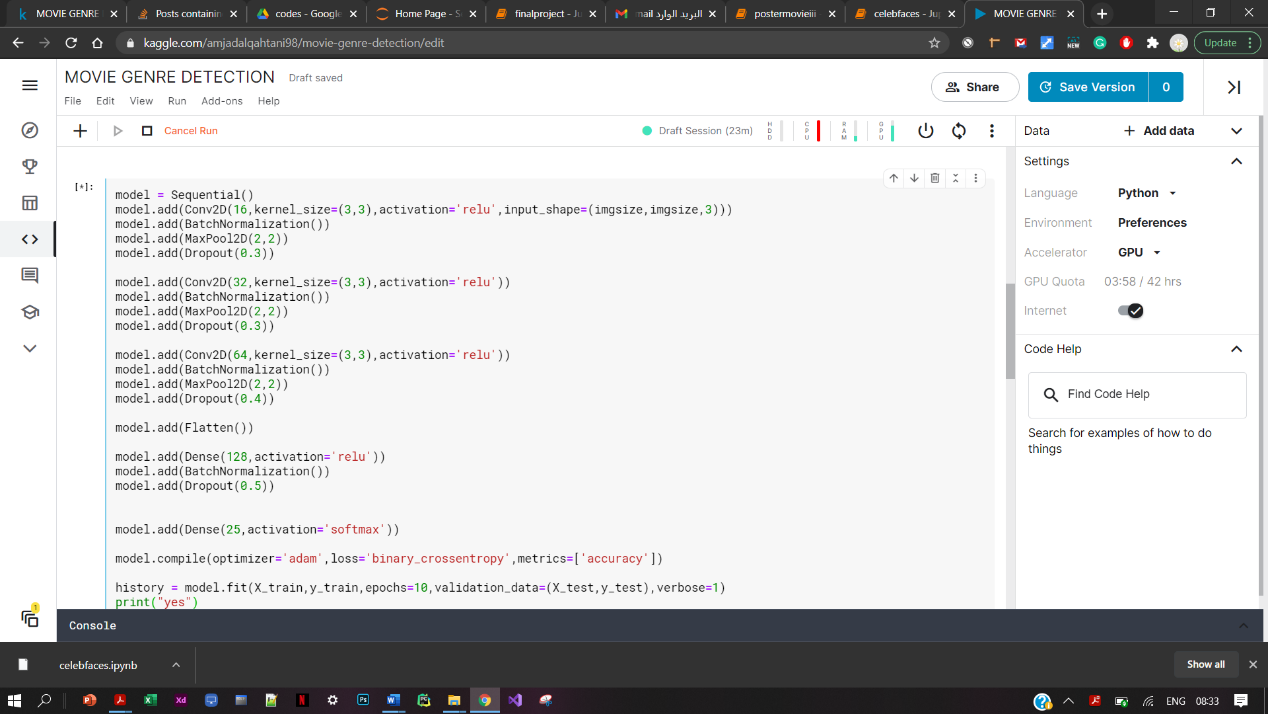
**Figure 4- 3 training dataset**



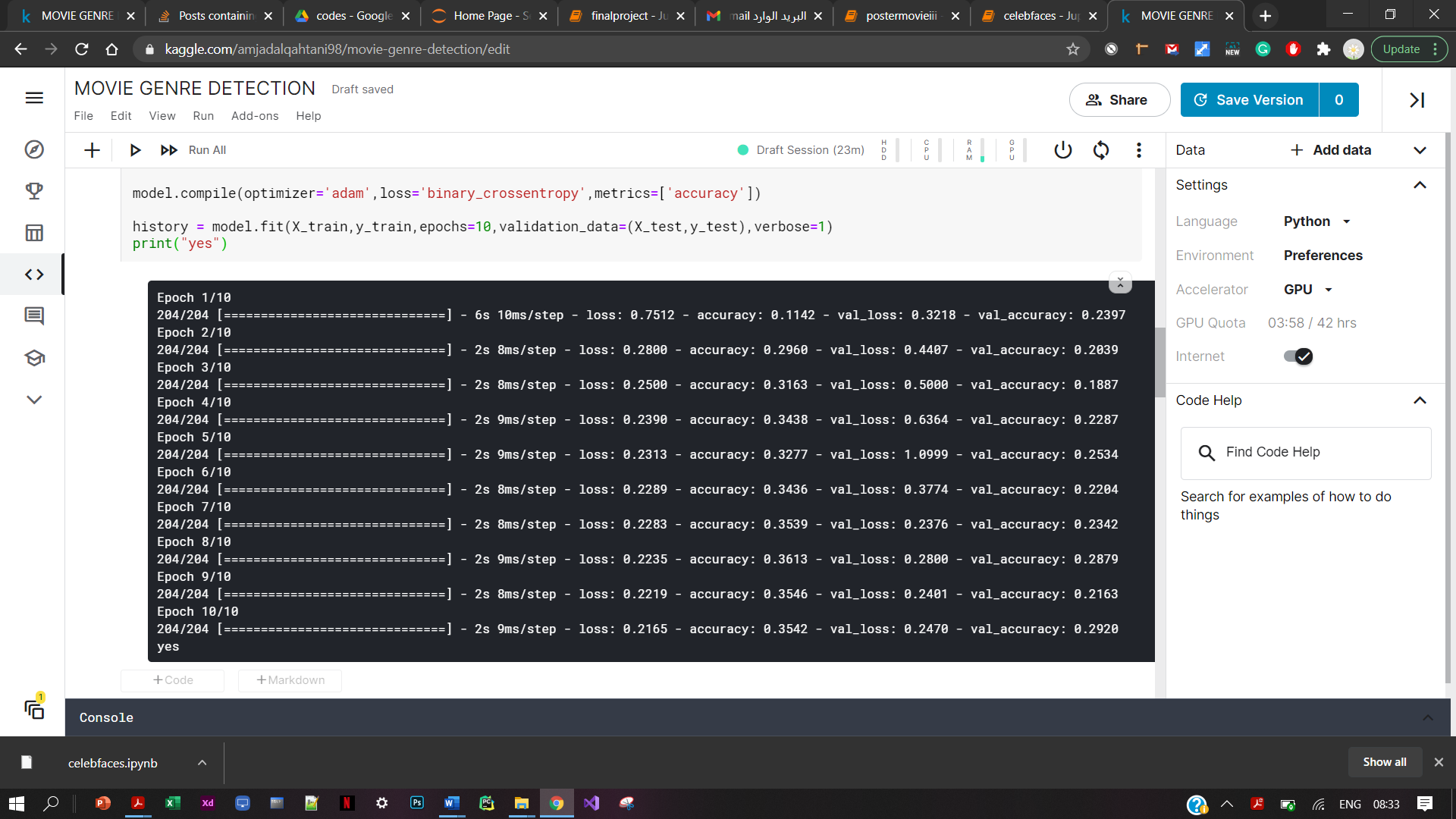
**Figure 4- 4 test dataset**

## 4.3 VIRTUAL ENVIRONMENT ON KAGGLE

To execute our code, we use a virtual environment on Kaggle because machine and deep learning programs need hours in the learning process. Kaggle provide GPU and make the excuse of our code faster than excuse the code on the CPU of a computer.



**Figure 4- 5 Virtual environment on Kaggle**



**Figure 4- 6 Execution of the code**

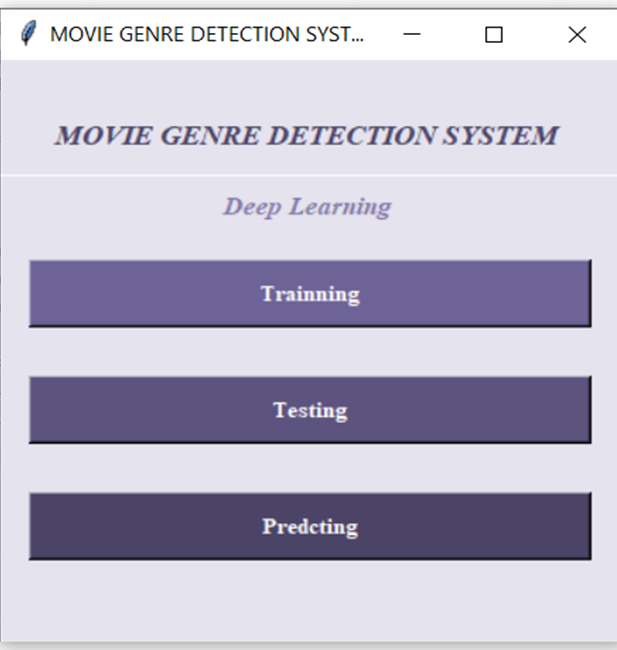
## 4.4 APPLICATION IMPLEMNTATION

The following we show how we create the screens of our system.

### 4.4.1 Main page

This figure 4-7 shows the main page, through this page the user can select a page to visit

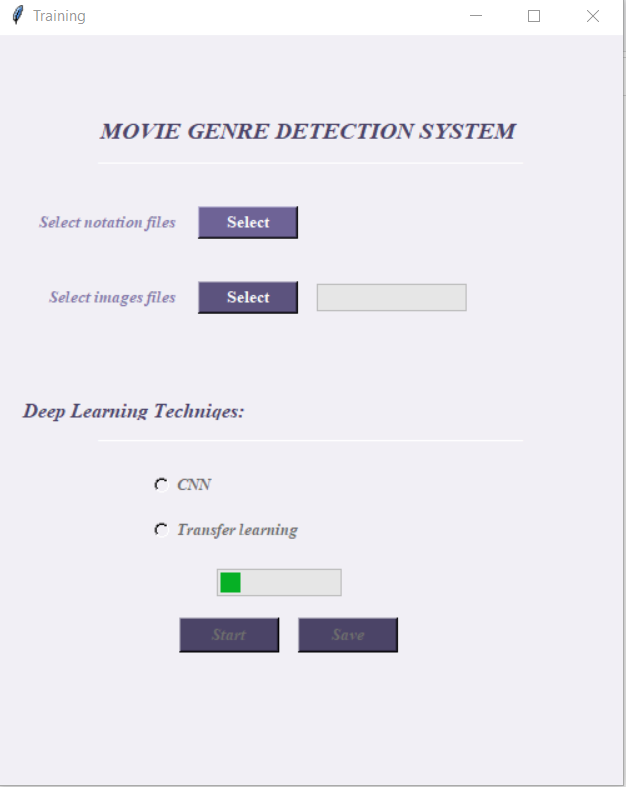
(Training, Testing, Predicting)



**Figure 4- 7 main page UI**

### 4.4.2 Training page

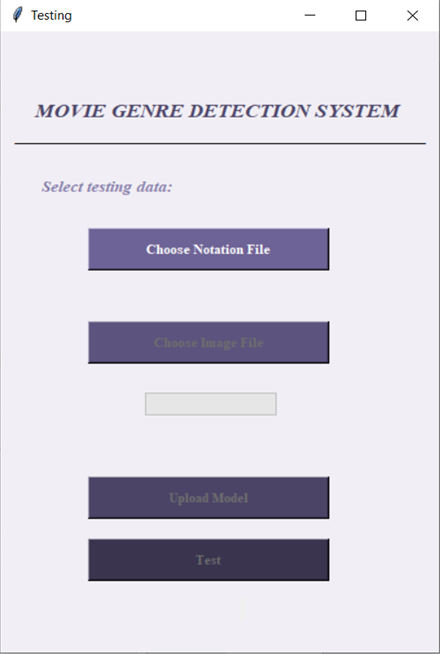
In figure 4-8, through this page the user will select the notation files (Train.csv) and select the image files (image) the files when it done loaded it will appear on the progress bar then from Deep Learning Techniques will choose one of CNN or Transfer learning as shown below, when the user press starts, the system will start learning on the selected files and then press save buttons save the model.



**Figure 4- 8 training UI**

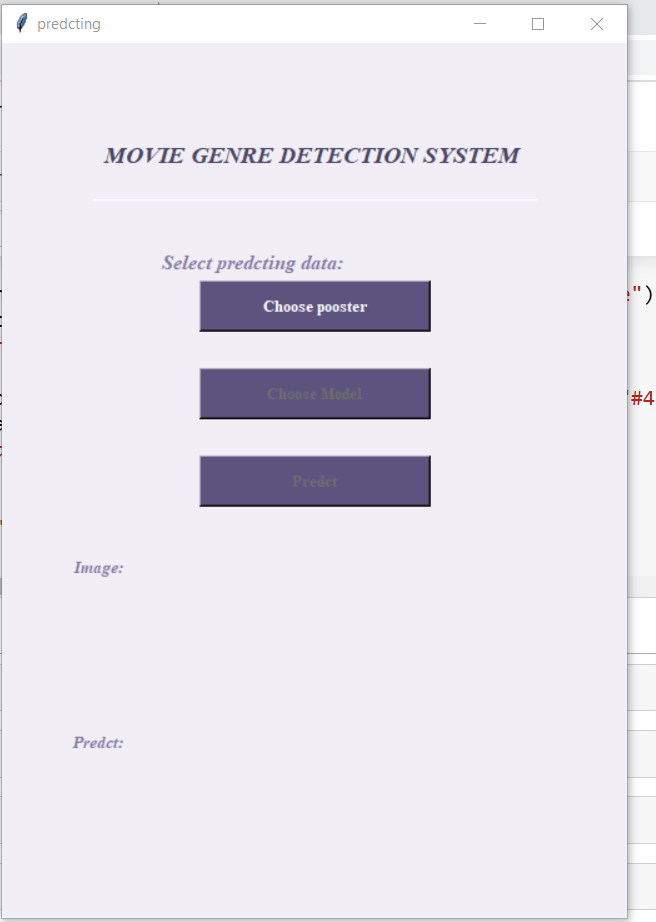
### 4.4.3 Testing page

The figure 4-9 shows the Testing page, through this page the user will choose the notation files (Test.csv) then select the image files (image) when the files it done loaded it will appear on the progress bar, then we will choose the upload model button to choose the model need to test it. then press test button to testing these files and show accuracy.



**Figure 4- 9 testing UI**

### 4.4.4 Predicting Page

The figure 4-10 shows the Predicting page, finally through this page the user will choose one poster to classification predicting and choose the upload model button to upload model, then press the predict button the system will display the classifications of the selected image either action or comedy or etc.

**Figure 4- 10 predicting UI**

# CHAPTER FIVE: TESTING

## 5.1 SOFTWARE TESTING

Software testing is defined as the process of evaluating the software application and verifying that the program is doing what it is supposed to do to determine if the software meets the specific requirements and discover the flaws of this program before using it.

## 5.2 TEST CASE SPECIFICATION

The testing cases divided here according to testing, training and predict. And for each of them we tested the functional requirements to ensure that they work correctly.

### 5.2.1 Training

|  |  |
| --- | --- |
| Requirements | Results |
| Select notation file | Successfully done. |
| Select image file | Successfully done. |
| Save | Successfully done. |
| Start | Successfully done. |

**Table 13 training Case Specification**

### 5.2.2 Testing

|  |  |
| --- | --- |
| Requirements | Results |
| Choose notation file | Successfully done. |
| Choose image file | Successfully done. |
| Upload Model | Successfully done. |
| Test | Successfully done. |

**Table 14**  **test Case Specification**

### 5.2.3 predicting

|  |  |
| --- | --- |
| Requirements | Results |
| Choose Poster | Successfully done. |
| Predict | Successfully done. |

**Table 15 predict Case Specification**

## 5.3 VALIDATION

Some of entry fields in the interface have a validation rule to test if the entry is

correct or not.

1. **Complete training**

After completing the training process a progress bar will be finished, then it will

show up a message box indicating the accuracy of the model.

Graphical user interface, application

Description automatically generated

**Figure 5- 1 accuracy of training in CNN**

1. **Complete testing**

Graphical user interface, application

Description automatically generatedThe message box will show up and indicating the accuracy and error of the testing process.

**Figure 5- 2 Complete testing**

1. **Complete predicting**

Graphical user interface, application

Description automatically generatedHere we come to the main purpose of our project the predicting of the poster. We will choose the poster that we want to show its classification. The poster will be printed on the screen. After pressed the predict button the classifications will show up on the screen (Figure 5-3).

**Figure 5- 3 Complete predicting**

# CHAPTER 6: CONCLUSION AND FUTURE WORK

## 6.1 CONCLUSION

In our project, Movie Genre Detection System is proposed, the aim for this project is extract and select the main features of a movie based on its poster only, to recognize and detect the movie genre, for instance: drama, horror, action,.etc. by using artificial intelligence methods like the deep-learning technique. This project providing high accuracy of extracting the genre.

This project consists of six chapters, the first chapter contains the problem definition, suggested solution, and what our project contributes.

The second chapter talks about the background and literature review. So, we talk about the machine and deep learning techniques that are used to predict the genre of a movie by extracting features from the image. The chapter also talks about some existing system that has some similar features of our system. Chapter three contains our project methodology, System requirements, hardware and software requirements, UML diagrams, and interface design. In chapter four we are implementing the whole system.

And in chapter five, we testing the project to ensure that the system is working correctly.

## 6.2 FUTURE WORK

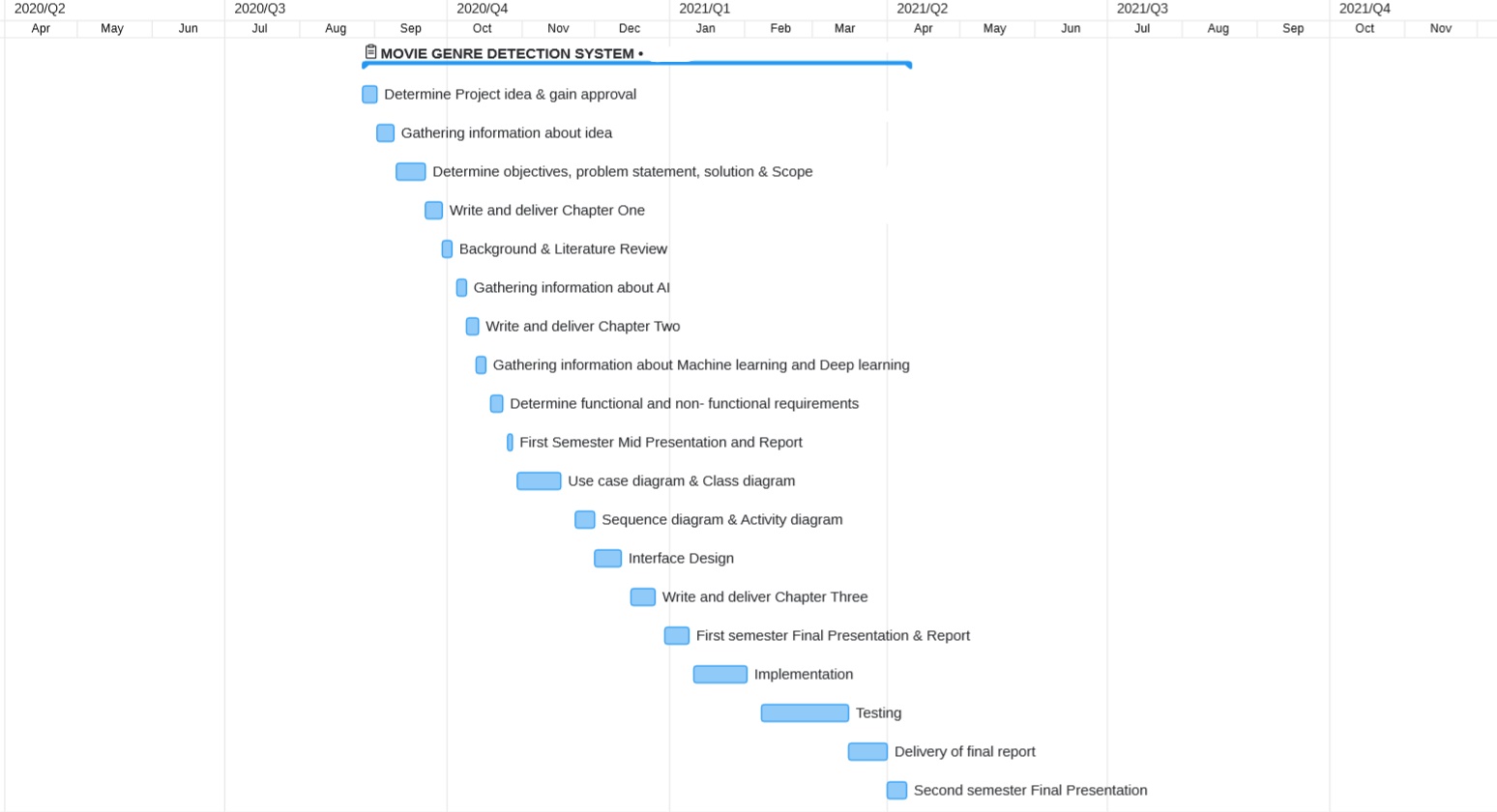
In the future we can upgrade our system, the improvements of the system

can be as following:

* Improve the system to be applying on a larger dataset.
* Improve the accuracy by using different methods.
* We can make our system detect more than the genre of movie

# APPENDIX I: PROJECT PLAN

## GANTT CHAR



## PROJECT ORGANIZATION

### Internal Structure

We are working as a team to building our project, and our structure to performing this project as following:

* 1. Project Coordinator
  2. Project Supervisor
  3. Project Leader
  4. Project Members

### Rolls and Responsibilities

***Project coordinator***: Dr. Akilah

***The scope of responsibility****:* Distribute projects and supervisors between groups and coordinate the graduation project exam.

***Project supervisor:*** Dr. Tala Qaid, Dr. Abeer Abdulqawi

***The scope of authority:*** Full authority over the project team.

***The scope of responsibility:*** Monitoring the performance all processes of the project and evaluate the work for each step.

***Project Leader:*** Fawziah Tariq Misfer.

***The scope authority:*** Full authority over the student’s team.

***The scope responsibility:*** Distribution work between the members in the team and support them by references to get benefit and successes the project.

***Project members:*** Amjad Abdullah, Atheer Alqadi, Ruba Ahmed.

***The scope authority***: Full authority over their tasks.

***The scope of responsibility:*** Definition the problem, collect data, analysis, design, and implementation. That each member will perform the specified mission for him by the project leader

# Appendix II: Coding

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import cv2

import tensorflow as tf

from tensorflow import keras

from keras import layers

from keras.layers import \*

from keras.models import \*

from tqdm import tqdm

from sklearn.model\_selection import train\_test\_split

from keras import callbacks

from tensorflow.keras import regularizers

from keras.applications import \*

from keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras import Sequential

from tensorflow.keras.layers import Conv2D,MaxPool2D,Dense,Flatten,BatchNormalization,Dropout

from tensorflow.keras.optimizers import Adam

import glob

import tkinter as tk

from tkinter import filedialog

from tkinter import simpledialog

from tkinter import ttk

from tkinter import \*

import os

import time

from datetime import datetime

import shutil

import threading

import json

import h5py

from tensorflow.python.keras.models import model\_from\_json

#from keras.models import Model,model\_from\_json

from keras.layers import Input

#from keras.layers import Dense

from keras.models import load\_model

from PIL import Image

from PIL import ImageTk

root=tk.Tk()

root.configure(background="#e5e3ed")

global im\_pre

im\_pre=[]

global imgsize

imgsize=120

* **Training Part:**

def save\_model():

e=tk.simpledialog.askstring(title="Save", prompt="What's your Model Name?:")

try:

json\_model = model.to\_json()

with open('Model1/'+e+'.json', 'w') as json\_file:

json\_file.write(json\_model)

json\_file.close()

model.save\_weights('Model1/'+e+'.h5')

except:

print("can't Save the model")

return

def fireTransferLearning():

from tensorflow.keras.models import Model

global model

datagen = ImageDataGenerator()

from tensorflow.keras.applications.vgg16 import VGG16

baseModel = VGG16 (weights= "imagenet", include\_top=False,input\_shape=(imgsize,imgsize,3))

headModel = baseModel.output

headModel = Flatten(name='flatten')(headModel)

headModel = Dense(128, activation='relu',kernel\_regularizer=regularizers.l1\_l2(l1=1e-5, l2=1e-4))(headModel)

headModel = Dropout(0.5)(headModel)

headModel = Dense(25, activation='sigmoid')(headModel)

model = Model(inputs = baseModel.input, outputs= headModel)

for layer in baseModel.layers:

layer.trainable = False

model.compile(loss='binary\_crossentropy', optimizer= 'adam', metrics=['accuracy'])

history = model.fit(datagen.flow(X\_train, y\_train, batch\_size=32),validation\_data=(X\_test, y\_test),epochs=10,verbose=0)

w="Max Training Accuracy: "+str(round(max(history.history['accuracy']),2))+" \nMax Validation Accuracy: "+str(round(max(history.history['val\_accuracy']),2))+"\nMin Training Loss: "+str(round(min(history.history['loss']),2))+"\nMin Validation Loss: "+str(round(min(history.history['val\_loss']),2))

messagebox.showinfo("Result", w)

def fireCNN():

global model

model = Sequential()

model.add(Conv2D(16,kernel\_size=(3,3),activation='relu',input\_shape=(imgsize,imgsize,3)))

model.add(BatchNormalization())

model.add(MaxPool2D(2,2))

model.add(Dropout(0.3))

model.add(Conv2D(32,kernel\_size=(3,3),activation='relu'))

model.add(BatchNormalization())

model.add(MaxPool2D(2,2))

model.add(Dropout(0.3))

model.add(Conv2D(64,kernel\_size=(3,3),activation='relu'))

model.add(BatchNormalization())

model.add(MaxPool2D(2,2))

model.add(Dropout(0.4))

model.add(Flatten())

model.add(Dense(128,activation='relu'))

model.add(BatchNormalization())

model.add(Dropout(0.5))

model.add(Dense(25,activation='sigmoid'))

model.compile(optimizer='adam',loss='binary\_crossentropy',metrics=['accuracy'])

history=model.fit(X\_train,y\_train,epochs=10,validation\_data=(X\_test,y\_test),verbose=0)

w="Max Training Accuracy: "+str(round(max(history.history['accuracy']),2))+" \nMax Validation Accuracy: "+str(round(max(history.history['val\_accuracy']),2))+"\nMin Training Loss: "+str(round(min(history.history['loss']),2))+"\nMin Validation Loss: "+str(round(min(history.history['val\_loss']),2))

messagebox.showinfo("Result", w)

def Check():

time.sleep(5)

if var.get() == 1:

fireCNN()

elif var.get() == 2:

fireTransferLearning();

else:

return

save['state']=NORMAL

def start\_check\_thread(event):

global submit\_thread

submit\_thread = threading.Thread(target=Check)

submit\_thread.daemon = True

progressbar.start()

submit\_thread.start()

trainning.after(20, check\_submit\_thread)

def check\_submit\_thread():

if submit\_thread.is\_alive():

trainning.after(20, check\_submit\_thread)

else:

progressbar.stop()

def open\_cv():

filename = filedialog.askopenfilename(initialdir="C:/", title="select file",filetypes=(("CSV Files","\*.csv"), ("all files", "\*.\*")))

global df

try:

df = pd.read\_csv(filename)

except:

print("can't open the file")

return

selectbut2['state']=NORMAL

def open\_file\_training():

choose=filedialog.askdirectory()

global imgsize

imgsize = 120

global X

X = []

global y

global path

global img

#tqdm

textvar.set("starting...")

length = (range(df.shape[0]))

old = 0

new = 0

try:

for i in length:

new = (i \* 100) / int(df.shape[0])

if int(new) > int(old):

old = new;

progress['value'] +=1

path = choose+'/'+df['Id'][i]+'.jpg'

img = cv2.imread(path)

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

img= cv2.resize(img,(imgsize,imgsize))

X.append(img)

trainning.update\_idletasks()

except:

print("Error in index or something in loop ")

return

X = np.array(X)

y = df.drop(['Id','Genre'],axis=1)

y = y.to\_numpy()

global X\_train, X\_test, y\_train, y\_test

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.1)

textvar.set("finished.")

rad1['state']=NORMAL

rad2['state']=NORMAL

Start['state']=NORMAL

def trainning():

global trainning

global selectbut2

global rad1

global rad2

global Start,save

trainning=Toplevel(root)

trainning.geometry("500x600")

trainning.title("Training")

trainning.configure(background="#f1eff5")

l=Label(trainning,text="MOVIE GENRE DETECTION SYSTEM ",fg="#4b4467", bg="#f1eff5")

l.place(relx = 0.5, rely = 0.13, anchor=CENTER)

l.config(font=("Times", 14, "bold italic"))

l=Label(trainning,text="\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_",fg="white", bg="#f1eff5")

l.place(relx = 0.5, rely = 0.16, anchor=CENTER)

selectlabel=Label(trainning, text="Select notation file",fg="#8379a7", bg="#f1eff5")

selectlabel.place(relx = 0.29, rely = 0.25, anchor=E)

selectlabel.config(font=("Times", 10, "bold italic"))

selectbut=Button(trainning,text="Select", command=open\_cv,width=10,height=1,fg="white", bg="#6e6396")

selectbut.place(relx = 0.48, rely = 0.25, anchor=E)

selectbut.config(font=("Times", 10, "bold roman"))

selectlabel=Label(trainning, text="Select images folder",fg="#8379a7", bg="#f1eff5")

selectlabel.place(relx = 0.29, rely = 0.35, anchor=E)

selectlabel.config(font=("Times", 10, "bold italic"))

selectbut2=Button(trainning,text="Select", command=open\_file\_training,width=10,state=DISABLED,height=1,fg="white", bg="#5c537e")

selectbut2.place(relx = 0.48, rely = 0.35, anchor=E)

selectbut2.config(font=("Times", 10, "bold roman"))

global progress

progress=ttk.Progressbar(trainning,orient=HORIZONTAL,length=120, mode='determinate')

progress.place(relx = 0.75, rely=0.35, anchor=E)

global textvar

textvar = StringVar();

textlable = Label(trainning, textvariable=textvar, fg="#4b4467", bg="#f1eff5")

textlable.place(relx=0.92, rely=0.35, anchor=E)

textlable.config(font=("Times", 10, "bold italic"))

techlabel=Label(trainning, text="Deep Learning Techniqes:",fg="#4b4467", bg="#f1eff5")

l=Label(trainning,text="\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_",fg="white", bg="#f1eff5")

l.place(relx = 0.5, rely = 0.53, anchor=CENTER)

techlabel.place(relx = 0.4, rely = 0.5, anchor=E)

techlabel.config(font=("Times", 12, "bold italic"))

global var

var = tk.IntVar()

rad1=Radiobutton(trainning,text="CNN",variable=var,state=DISABLED,value=1,fg="#8379a7", bg="#f1eff5")

rad2=Radiobutton(trainning,text="Transfer learning",variable=var,state=DISABLED,value=2,fg="#8379a7", bg="#f1eff5")

rad1.place(relx = 0.35, rely = 0.6, anchor=E)

rad2.place(relx = 0.49, rely = 0.66, anchor=E)

rad1.config(font=("Times", 10, "bold italic"))

rad2.config(font=("Times", 10, "bold italic"))

save=Button(trainning, text="Save", command=save\_model,state=DISABLED, width=10,height=1,fg="white", bg="#4b4467")

Start=Button(trainning, text="Start", command=lambda:start\_check\_thread(None),state=DISABLED ,width=10,height=1,fg="white", bg="#4b4467")

Start.place(relx = 0.45, rely = 0.8, anchor=E)

save.place(relx = 0.64, rely = 0.8, anchor=E)

Start.config(font=("Times", 10, "bold italic"))

save.config(font=("Times", 10, "bold italic"))

global progressbar

progressbar = ttk.Progressbar(trainning, mode='indeterminate')

progressbar.place(relx=0.55, rely=0.73, anchor=E)

* **Testing Part:**

def open\_cv\_test():

filename = filedialog.askopenfilename(initialdir="C:/", title="select file",filetypes=(("CSV Files","\*.csv"), ("all files", "\*.\*")))

global df

try:

df = pd.read\_csv(filename)

except:

print("can't open the file")

return

choose1['state']=NORMAL

def open\_file\_test():

choose=filedialog.askdirectory()

global imgsize

imgsize = 120

global X

X = []

global y

global path

global img

#tqdm

textvar.set("starting...")

length = (range(df.shape[0]))

old = 0

new = 0

try:

for i in length:

new = (i \* 100) / int(df.shape[0])

if int(new) > int(old):

old = new;

progress['value'] +=1

path = choose+'/'+df['Id'][i]+'.jpg'

img = cv2.imread(path)

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

img= cv2.resize(img,(imgsize,imgsize))

X.append(img)

test.update\_idletasks()

except:

print("Error in index or something in loop ")

return

X = np.array(X)

y = df.drop(['Id','Genre'],axis=1)

y = y.to\_numpy()

global X\_train, X\_test, y\_train, y\_test

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.1)

textvar.set("finished.")

load['state']=NORMAL

def upload\_model():

global model\_test

try:

pathTest=filedialog.askopenfilename(initialdir="C:/", title="select file",filetypes=(("JSON Files","\*.json"),("all files", "\*.\*")))

json\_file=open(pathTest, 'r')

json\_savedModel= json\_file.read()

json\_file.close()

model\_test=model\_from\_json(json\_savedModel)

model\_test.load\_weights(pathTest.replace('json','h5'))

testbut['state']=NORMAL

except:

print("can't upload model ..")

return

def Test\_Model():

t.set("Test is starting ...")

global er,ac

model\_test.compile(optimizer='adam',loss='binary\_crossentropy',metrics=['accuracy'])

er,ac=model\_test.evaluate(X\_test,y\_test,verbose=0)

t.set("Test is finsh")

w="Accuracy : "+str(round(ac,2))+"\nError: "+str(round(er,2))

messagebox.showinfo("Result", w)

def testing():

global test,choose1

global load,testbut

test=Toplevel(root)

test.geometry("400x600")

test.title("Testing")

test.configure(background="#f1eff5")

l=Label(test,text="MOVIE GENRE DETECTION SYSTEM ",fg="#4b4467", bg="#f1eff5")

l.place(relx = 0.5, rely = 0.13, anchor=CENTER)

l.config(font=("Times", 14, "bold italic"))

l=Label(test,text="\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_",fg="black", bg="#f1eff5")

l.place(relx = 0.5, rely = 0.17, anchor=CENTER)

selectlabel=Label(test, text="Select testing data:",fg="#8379a7", bg="#f1eff5")

selectlabel.place(relx = 0.4, rely = 0.25, anchor=E)

selectlabel.config(font=("Times", 11, "bold italic"))

choose=Button(test, text="Choose Notation File",command= open\_cv\_test,width=30,height=2 ,fg="white",bg="#6e6396")

choose.place(relx = 0.75, rely = 0.35, anchor=E)

choose.config(font=("Times", 10, "bold roman"))

choose1=Button(test, text="Choose Images folder",command= open\_file\_test,state=DISABLED,width=30,height=2,fg="white", bg="#5c537e")

choose1.place(relx = 0.75, rely = 0.50, anchor=E)

choose1.config(font=("Times", 10, "bold roman"))

global progress

progress=ttk.Progressbar(test,orient=HORIZONTAL,length=120, mode='determinate')

progress.place(relx = 0.63, rely=0.60, anchor=E)

global textvar

textvar = StringVar();

textlable = Label(test, textvariable=textvar, fg="#4b4467", bg="#f1eff5")

textlable.place(relx=0.55, rely=0.66, anchor=E)

textlable.config(font=("Times", 9, "bold roman"))

load=Button(test, text="Upload Model",command=upload\_model,state=DISABLED,width=30,height=2,fg="white", bg="#4b4467")

load.place(relx = 0.75, rely = 0.75, anchor=E)

load.config(font=("Times", 10, "bold roman"))

testbut=Button(test, text="Test",width=30,height=2,command=Test\_Model,state=DISABLED,fg="white",bg="#3a344f")

testbut.place(relx = 0.75, rely = 0.85, anchor=E)

testbut.config(font=("Times", 9, "bold roman"))

global t,l0

t=StringVar();

l0=Label(test, textvariable=t,fg="#4b4467")

l0.place(relx = 0.56, rely = 0.93, anchor=E)

l0.config(font=("Times", 9, "bold italic"))

* **Predicting Part:**

def predicting():

global predict

global predctbut ,choose2

predict=Toplevel(root)

predict.geometry("500x700")

predict.title(" predicting")

predict.configure(background="#f1eff5")

l=Label(predict,text="MOVIE GENRE DETECTION SYSTEM ",fg="#4b4467", bg="#f1eff5")

l.place(relx = 0.5, rely = 0.13, anchor=CENTER)

l.config(font=("Times", 14, "bold italic"))

l=Label(predict,text="\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_",fg="white", bg="#f1eff5")

l.place(relx = 0.5, rely = 0.17, anchor=CENTER)

selectlabel=Label(predict, text="Select predcting data:",fg="#8379a7", bg="#f1eff5")

selectlabel.place(relx = 0.4, rely = 0.25, anchor=CENTER)

selectlabel.config(font=("Times", 12, "bold italic"))

choose=Button(predict, text="Choose pooster",width=25,height=2,fg="white", bg="#5c537e",command=open\_image)

choose.place(relx = 0.5, rely = 0.3, anchor=CENTER)

choose.config(font=("Times", 10, "bold roman"))

choose2=Button(predict, text="Choose Model",state=DISABLED,width=25,height=2,fg="white", bg="#5c537e",command=upload\_model\_p)

choose2.place(relx = 0.5, rely = 0.4, anchor=CENTER)

choose2.config(font=("Times", 10, "bold roman"))

predctbut=Button(predict, text="Predict",state=DISABLED,width=25,height=2,fg="white", bg="#5c537e",command=predict\_image)

predctbut.place(relx = 0.5, rely = 0.5, anchor=CENTER)

predctbut.config(font=("Times", 10, "bold roman"))

img=Label(predict, text="Image:",fg="#8379a7", bg="#f1eff5")

img.place(relx = 0.2, rely = 0.6, anchor=E)

img.config(font=("Times", 10, "bold italic"))

pre=Label(predict, text="Predict:",fg="#8379a7", bg="#f1eff5")

pre.place(relx = 0.2, rely = 0.8, anchor=E)

pre.config(font=("Times", 10, "bold italic"))

def upload\_model\_p():

global model\_pre

try:

pathPre=filedialog.askopenfilename(initialdir="C:/", title="select file",filetypes=(("JSON Files","\*.json"),("all files", "\*.\*")))

json\_file=open(pathPre, 'r')

json\_savedModel= json\_file.read()

json\_file.close()

model\_pre=model\_from\_json(json\_savedModel)

model\_pre.load\_weights(pathPre.replace('json','h5'))

predctbut['state']=NORMAL

except:

print("Can't upload model")

return

def open\_image():

global im\_pre

im\_pre=[]

imgsize=120

try:

filename = filedialog.askopenfilename(initialdir="C:/", title="select file",filetypes=(("Image","\*.jpg"), ("all files", "\*.\*")))

im = cv2.imread(filename)

im = cv2.cvtColor(im, cv2.COLOR\_BGR2RGB)

im = cv2.resize(im,(imgsize,imgsize))

im\_pre.append(im)

im\_pre=np.array(im\_pre)

im = Image.open(filename)

im = im.resize((imgsize, imgsize), Image.ANTIALIAS)

im = ImageTk.PhotoImage(im)

panel = Label(predict,image=im,borderwidth=2, relief="groove")

panel.image = im

panel.place(relx = 0.55, rely = 0.7,anchor =E)

choose2['state']=NORMAL

except:

print("Can't open image ..")

return

def predict\_image():

global i

tx=tk.Frame(predict)

tx.place(relx = 0.6, rely = 0.9, anchor=E)

scrollbar = Scrollbar(tx)

f=["Action","Adventure","Animation","Biography","Comedy","Crime","Documentary","Drama","Family","Fantasy","History","Horror","Music","Musical","Mystery","N/A","News","Reality-TV","Romance","Sci-Fi","Short","Sport","Thriller","War","Western"]

f=np.array(f)

x=im\_pre.reshape(1,imgsize,imgsize,3)

p=model\_pre.predict(x)

p=p[0]

T = tk.Text(tx, height=6, width=20, yscrollcommand=scrollbar.set)

T.config(font=("Times", 9, "bold roman"))

for i in range(len(p)):

l=f[i]+'\t'+str(int(p[i]\*100))+'%\n'

T.insert(tk.END, l)

scrollbar.config(command=T.yview)

scrollbar.pack(side=RIGHT, fill=Y)

T.pack(side="left")

* **Main Part:**

l=Label(root,text="MOVIE GENRE DETECTION SYSTEM ",fg="#4b4467",bg="#e5e3ed")

l.place(relx = 0.5, rely = 0.13, anchor=CENTER)

l.config(font=("Times", 13, "bold italic"))

l2=Label(root,text="\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_",fg="white", bg="#e5e3ed")

l2.place(relx = 0.5, rely = 0.18, anchor=CENTER)

l3=Label(root,text="Deep Learning ",fg="#8379a7",bg="#e5e3ed")

l3.place(relx = 0.5, rely = 0.25, anchor=CENTER)

l3.config(font=("Times", 11, "bold italic"))

trainning=Button(root,text="Training", command=trainning,width=44,height=2,fg="white",bg="#6e6396")

trainning.place(relx = 0.5, rely = 0.4, anchor=CENTER)

trainning.config(font=("Times", 10, "bold roman"))

testing=Button(root,text="Testing",command=testing, width=44,height=2,fg="white", bg="#5c537e")

testing.place(relx = 0.5, rely = 0.6, anchor = CENTER)

testing.config(font=("Times", 10, "bold roman"))

predicting=Button(root,text="predicting",command= predicting , width=44,height=2,fg="white", bg="#4b4467")

predicting.place(relx = 0.5, rely = 0.8, anchor = CENTER)

predicting.config(font=("Times", 10, "bold roman"))

root.geometry("350x350")

root.title("MOVIE GENRE DETECTION SYSTEM")

root.mainloop()

# REFERENCE

[1] An overview of AI by Dheeraj Sagar [<https://www.researchgate.net/publication/236346414_AN_OVERVIEW_OF_ARTIFICIAL_INTELLIGENCE>]

[2] Bengio, Yoshua; LeCun, Yann; Hinton, Geoffrey (2015). "Deep Learning" [<https://www.researchgate.net/publication/277411157_Deep_Learning>]

[3] About us, Shazam [ <https://www.shazam.com/company> ]

[4] About us, foodvisor [<https://www.foodvisor.io/en/>]

[5] About us, NatureID [<https://www.natureid.no/>]

[6] Want to know how Deep Learning works by Radu Raicea [<https://www.freecodecamp.org/news/want-to-know-how-deep-learning-works-heres-a-quick-guide-for-everyone-1aedeca88076/>]

[7] SOFTWARE ENGINEERING Ninth Edition Ian Sommerville

[8] <https://www.educba.com/class-diagram/>

[9] <https://www.lucidchart.com/pages/uml-sequence-diagram>

[10] Disadvantage of sequence diagram [<https://www.programmersought.com/article/4869113487/>]