AWSome

Karthikeyan Rathore E19CSE119,
Asfahan Shah E19CSE116

Introduction

In today’s world, Music streaming services like Spotify, YouTube etc. have become really popular. Due to large number of songs and various streaming services it is very hard for a user to select a particular genre of music. Because of this various platform have an inbuild machine learning based recommendation systems. Besides this there are other features like playlists which can help the user to select and discover the music. But usually, a user selects the music based upon the mood, but it is quite common for the user to not properly express their emotion in form of words.

Though there are services mentioned above, available in these platforms that can help a user to select the music but still these are quite basic in nature and does not help in the main problem which is helping the user to select music by the user current mood. The recommendation system which is usually present in these platforms is useful but still the primary criteria of selection i.e., mood of the user is not taken into account. This is where our project AWSome comes into the play. In our project we are trying to eliminate one of the problems of the recommendation system commonly used in these platforms that is these recommendation systems usually take listening history into the account but do not take the user mood into the account. Besides that, if these platform have mood search option then there is again a problem that sometimes it is difficult for the users to explain the mood or even know it properly. We have utilized advance computer vision algorithm, machine learning algorithms and leverage the power of AWS cloud to make our project global with high availability, reliability and more security. As it is said, “**An image is equivalent to a thousand words**” therefore we have used user image as a main criterion for mood detection.

AWSome determines the mood of the user based upon the user image and automatically determines the suitable playlist of songs. From this, project the main outcome is to provide a robust song recommendation system based upon the user current mood. Another outcome of the project is to reduce the complexity and hassle faced by the user by providing a simple UI based platform. The project also aims to provide user privacy and data security by leveraging the concepts and services of AWS cloud.

Motivation and Related Works

As we are mainly working on improving the recommendation systems, the works which are related to our project can be found in various real life application and software. Some of the platforms like Spotify, YouTube etc, have really sophisticated and really complex recommendation architecture systems. These help in recommending the user appropriate songs using a variety of factors such as rating, trending, listening history and so on. These are really helpful but again, the main factor that can influence the music choice of the user is the mood. This can be real problem in the recommendation system available in the market as sometimes it can be really difficult for the user to determine the genera by himself. Moreover it can be quite difficult and long for the user to describe his moods in words.

Besides the industrial uses various researchers have also explored this topic. In a study [1] conducted by M. Gori et.al proposed a random walk based approach for recommendation system. Kunal Shah et.al [2] conducted a review of various recommendation systems which are used. They showed that the majority of the systems which are used are of two types :- content based and collaborative based. Content Based recommendation system is based upon the user own interests while the collaborative system is based on the interest of the other past users.

Various other platforms like Netflix, YouTube etc, also have these kinds of complex recommendation systems based on NLP algorithms (Natural language processing) and machine learnings. The problem that we discussed above does not become a huge issue incase of Netflix or YouTube videos due to the nature of the content (mainly long videos) but for platform like Spotify which has mainly songs and that to in mp3 format the problem amplifies. That’s why, this weakness becomes a prime motivator for us to build the project. Our project try’s to eliminate this weakness by using computer vision algorithm and machine learning model to capture the mood of the user with the help of image.

Tech Utilized

We have utilized various technologies in our project these technologies can be mainly divided into two groups:- Frontend and backend.

Front-End Specification

We have used Bootstrap, CSS and HTML5 for the frontend. As mentioned earlier our fronted is simple and intuitive which any user can easily use. We have made home page, login page, register page and music recommendation page.

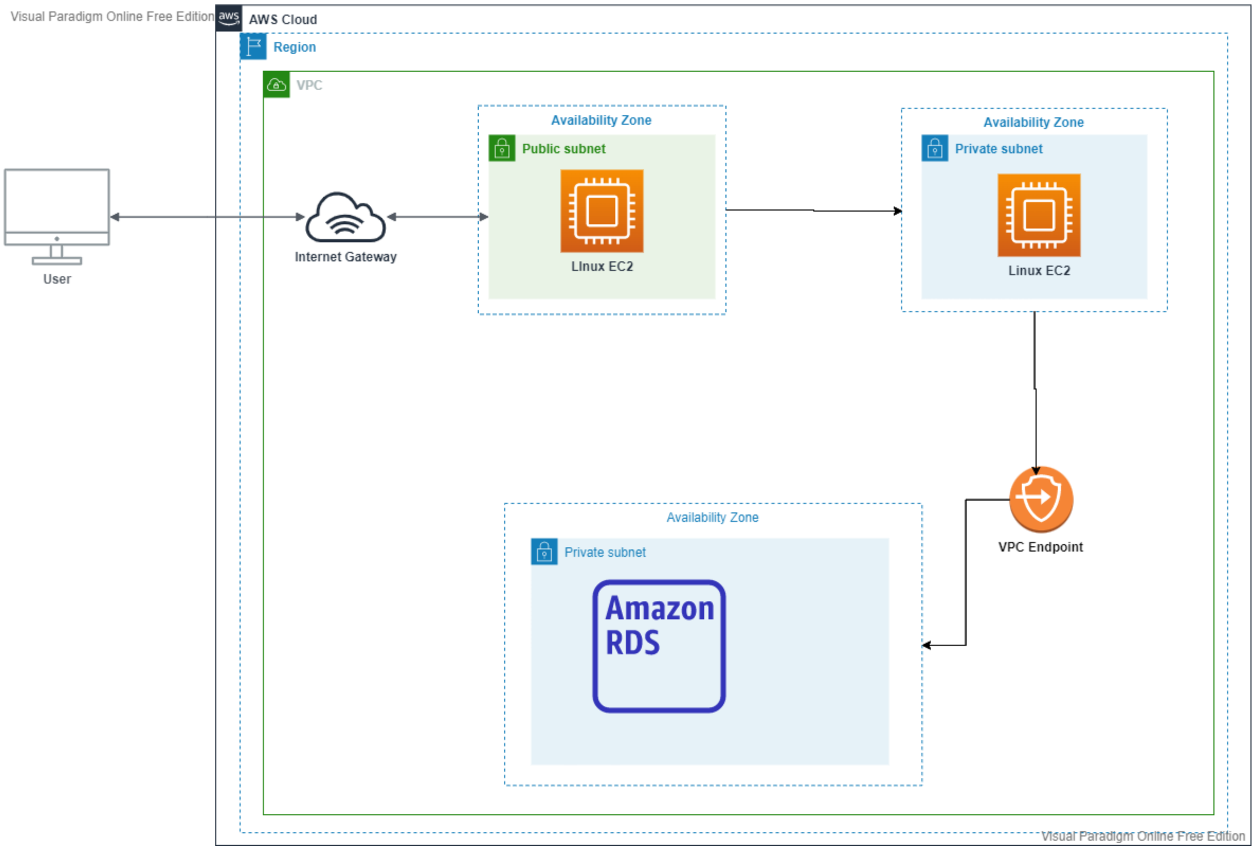
Back-End Specification

We have used Python programming language for our backend. Our project is a web application for which we have used Flask framework. We have used MYSQL database (which is attached to AWS RDS service) for storing user information. We have applied hashing techniques for securing user data. We have used Spotify API for song retrieval based upon the mood. Our emotion detection model is made from python and TensorFlow and can detect up to three emotions based upon the user image. For the deployment of the project, we have employed AWS cloud services. We have made a custom VPC where we have created public and private subnets. The private subnet consists of all the sensitive data like user information while the public subnet consists of the non-sensitive information. This has been done in order to maintain user privacy and data security as private subnet cannot be accessed globally. To update the database in the private subnet we have employed amazon VPC endpoint service. For the database we have used amazon RDS service. RDS service provides auto-scaling and backup features for increased performance and efficiency. We have created Linux-based ec2 instance for the deployment of our project.

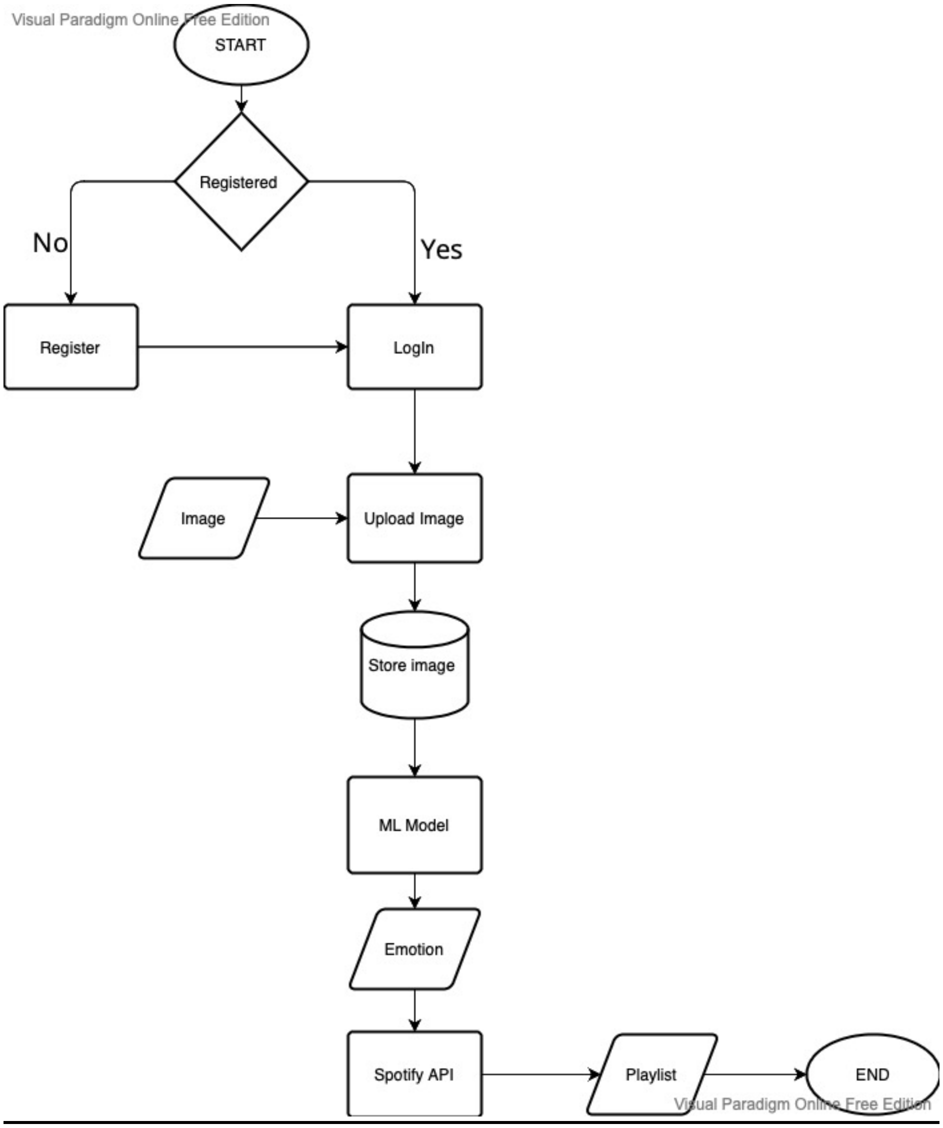
Overall, the tech utilized in both the frontend and the backend can be listed below:-

Bootstrap, CSS, HTML5, MySQL database, spotify api, python, tensorflow, opencv, flask, encryption techniques, Amazon RDS service, Amazon VPC service, Amazon EC2 instance(Linux), Internet gateway, git. We have also used various sub services in AWS like subnet creation, route table creation, VPC endpoint creation , subnet group etc.

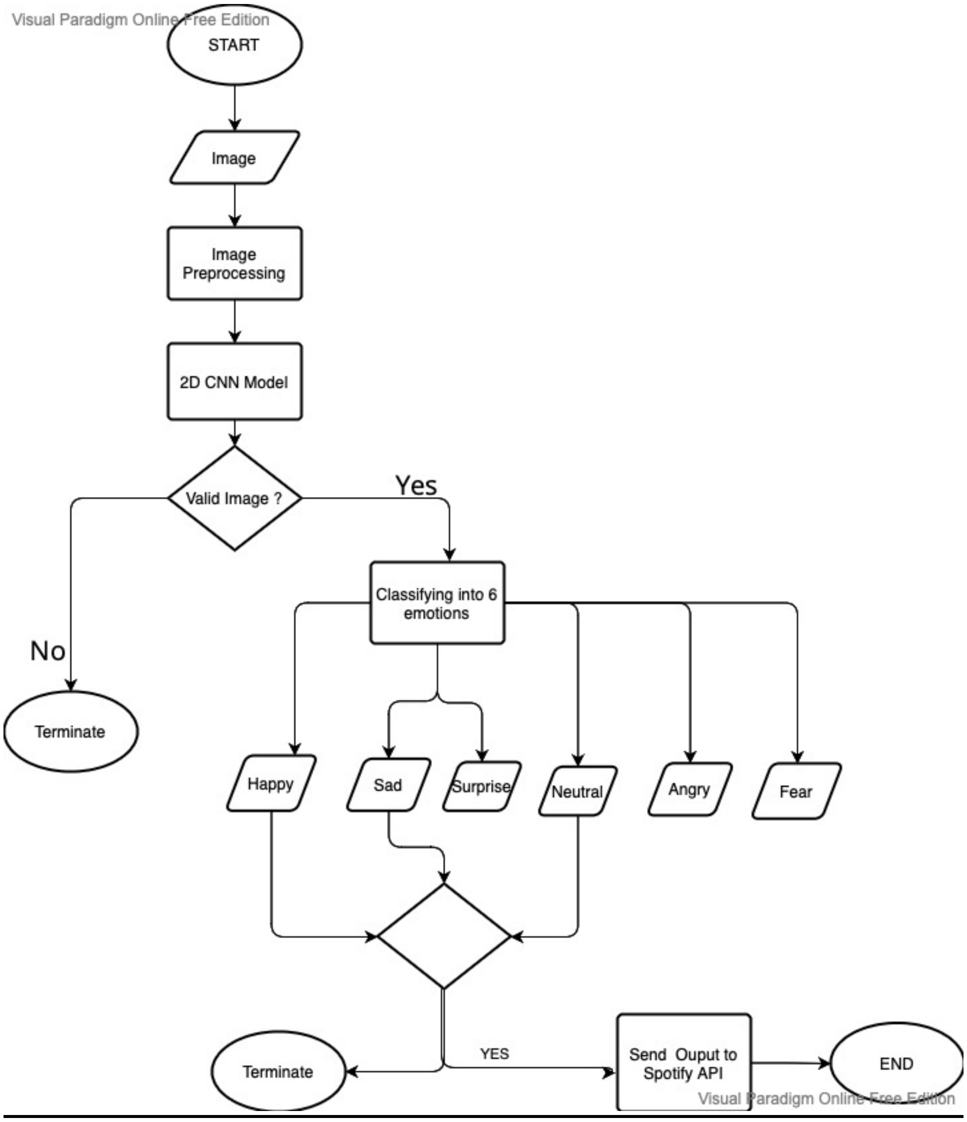
Diagrams (Architectural Diagram)



Diagrams (Overall Flow Chart Diagram)



Diagrams (Machine Learning Model Flow Chart Diagram)


Visual Paradigm Online Free Edition
START
Image
Image Preprocessing
2D CNN Model
Valid Image ?
Yes
No
Terminate
Classifying into 6 emotions
Happy
Sad
Surprise
Neutral
Angry
Fear
YES
Terminate
Send Ouput to Spotify API
END
Visual Paradigm Online Free Edition

Implementation

The Implementation / documentation of the project is quite long and hence we have summarized the main steps below.

1. Create VPC.
2. Create two private and 1 public subnet in two different availability zones.
3. Attach Internet Gateway to VPC.
4. Launch Public Ec2 and Private Ec2 instance (created using image of Public Ec2).
5. Create a subnet group and RDS for MySQL database.
6. Create vpc-endpoint for RDS.
7. Ssh into Public ec2 instance.
8. Upgrade all the packages of the Linux operating system.
9. Install all the dependencies of the project.
10. Git clone the repo: https://github.com/karthikeyanrathore/AWSome. (Currently Private repo).
11. Ssh into private ec2 instance.
12. Connect MySQL server of private ec2 instance to RDS.
13. Created schema inside the MySQL server.
14. Exit from the private ec2 instance.
15. Run the flask server.

Conclusion and Future Works.

From this project we have concluded the following things,

1. First of all we have tried to improve the overall quality of the music recommendation system by implementing computer vision based recommendation system.

2. We have implemented a 2D CNN model and the accuracy of the model is 67%.

3. The data set was obtained using the Kaggle.

4. The main goal which was of improving the recommendation system was obtained by combining this model (used for emotion detection) and Spotify API.

5. In the future the first we would improve is on the accuracy of the facial emotion recognition model.

6. The next thing that we would improve is on trying to reduce the expenditure which currently is spend on the cloud architecture by using service such as AWS lambda.

7. We will also improve the user experience and user interface of the website.

8. Currently, the dependence of our project is on Spotify API, we would also try to make our own directory of different genera of songs.

References

1. M. Gori and A. Pucci, "Research Paper Recommender Systems: A Random-Walk Based Approach," 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06), 2006, pp. 778-781, doi: 10.1109/WI.2006.149.

2. K. Shah, A. Salunke, S. Dongare and K. Antala, "Recommender systems: An overview of different approaches to recommendations," 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 2017, pp. 1-4, doi: 10.1109/ICIIECS.2017.8276172.

3. https://docs.aws.amazon.com/rds/index.html

4. https://docs.aws.amazon.com/vpc/index.html

5. https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Subnets.html

6. https://docs.aws.amazon.com/ec2/index.html

7. https://www.tensorflow.org/api_docs

8. https://flask.palletsprojects.com/en/2.0.x/

9. https://dev.mysql.com/doc/

10. https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html

11. https://developer.spotify.com/documentation/

12. https://getbootstrap.com/docs/4.1/getting-started/introduction/