Human Emotion Recognition and Song Recommendation Model

Advik Katiyar, Mohith Manohar

Abstract—In this paper, I present a project that recognizes human emotions and recommends a song based on them. This model uses machine learning techniques to identify the five basic human emotions from facial images: happiness, anger, sadness, surprise, and neutral. This model employs a single machine learning model, which is trained on a dataset of labeled facial images and uses Haar Cascade classifier and Local Binary Patterns Histograms algorithm (for feature extraction and classification) to recognize emotions in uploaded images. A supporting code is used to recommend songs based on the recognized emotion. This paper evaluates the accuracy of the human emotion recognition model and the effectiveness of the song recommendation system. The findings of this study can contribute to the development of more advanced emotion recognition models and music recommendation systems in the future.

I. INTRODUCTION

Music has been known to have a powerful impact on human emotions [1]. With the increasing availability of digital music services and the rise of personalized recommendations, there is a growing need to develop more advanced music recommendation systems [2].

This paper presents a project that recognizes human emotions and recommends a song based on them. The implementation of this idea involves a single machine learning model. This project is undertaken at a high school level, therefore the significance of this project lies in its potential to be developed further with time, which would eventually contribute to the development of more advanced emotion recognition and song recommendation models in the industry. Therefore, the objective of this project is to learn and create a human emotion recognition and song recommendation model, and evaluate its performance in terms of accuracy to cap it off.

II. LITERATURE REVIEW

Emotion recognition using machine learning has been an area of interest for researchers in recent years. Researchers have explored various methods for detecting emotions from different sources, including facial expressions, voice, and physiological signals. For example, some studies have used computer algorithms to detect emotions based on facial expressions [3], while others have used voice signals to detect emotions [4]. These studies have demonstrated the potential of machine learning for detecting emotions accurately.

Music recommendation systems have also been studied extensively. These systems are designed to recommend music to users based on their preferences or behavior. Various approaches have been explored for building music recommendation systems, including collaborative filtering, content-based filtering, and hybrid approaches. Collaborative filtering uses the preferences of users to make recommendations, while content-based filtering uses the characteristics of music to make recommendations [5]. Hybrid approaches combine the two methods to provide more accurate recommendations [6]. Any of this, however, would not be looked upon in this paper, as creating two machine learning models discussed in this literature review was not possible in the time period that was set for this project.

Moving on, despite the progress made in emotion recognition and music recommendation systems, there are still challenges that need to be addressed. For example, emotion recognition from images can be affected by factors such as lighting conditions and pose variations, while the performance of music recommendation systems can be affected by data sparsity and scalability. Addressing these challenges will require further research in the field of machine learning and artificial intelligence. This, however, is not the goal of this paper. It is only to learn how to create one using the LBPH algorithm and test for its performance.

III. METHODOLOGY

This section explains in detail how the model works. The workflow attached provides a brief overview.

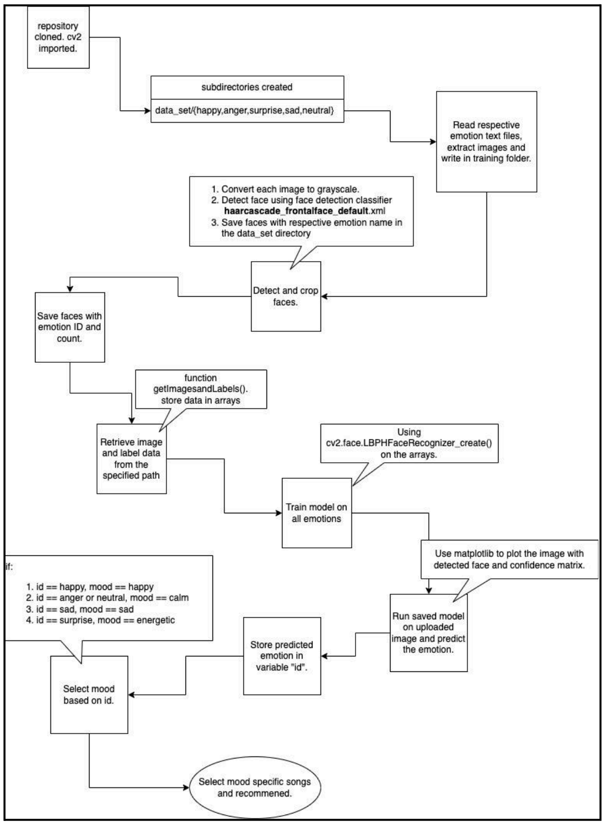
repository cloned. cv2 imported.
subdirectories created
data_set/{happy,anger,surprise,sad,neutral}
Read respective emotion text files, extract images and write in training folder.
1. Convert each image to grayscale.
2. Detect face using face detection classifier **haarcascade_frontalface_default**.xml
3. Save faces with respective emotion name in the data_set directory
Detect and crop faces.
Save faces with emotion ID and count.
function getImagesandLabels(). store data in arrays
Retrieve image and label data from the specified path
Using cv2.face.LBPHFaceRecognizer_create() on the arrays.
Train model on all emotions
Use matplotlib to plot the image with detected face and confidence matrix.
Run saved model on uploaded image and predict the emotion.
Store predicted emotion in variable "id".
if:
1. id == happy, mood == happy
2. id == anger or neutral, mood == calm
3. id == sad, mood == sad
4. id == surprise, mood == energetic
Select mood based on id.
Select mood specific songs and recommened.

Fig. 1. Model Workflow

This model is a human emotion recognition and song recommendation model that uses the HAAR Cascade Classifier and the LBPH face recognition algorithm. It starts by creating five folders in a directory called “dataset” for the emotions “happy”, “anger”, “surprise”, “sad” and “neutral”. Then, it reads text files (from a github dataset) that contain the names of the image files that correspond to each emotion, and saves those images to their respective emotion folder in the “dataset” folder.

Next, it detects faces using the Haar Cascade classifier for each emotion’s image and saves the detected faces in a dataset folder as grayscale images with the format “User.<emotion id>.<count>.jpg”.

Finally, it trains the LBPH recognizer on the entire dataset by reading in each image from each emotion folder and using the corresponding emotion id as the label. The trained model is then saved as a yml file.

To use the trained model for human emotion recognition, it reads in an image, detects faces using the Haar Cascade classifier, extracts the face region, and feeds the grayscale face to the trained LBPH recognizer. The predicted emotion label is then displayed on the image using OpenCV’s cv2.putText function.

A. Human Emotion Recognition Code Technical Explanation

The created model is a human emotion recognition system that utilizes computer vision techniques to classify emotions in human facial images. The model is trained on a dataset of facial images labeled with corresponding emotions, and then uses the trained model to recognize emotions in new facial images.

The first step in the workflow is to download and prepare the dataset. The dataset used in this model is a collection of facial images labeled with five different emotions: happy, anger, surprise, sad, and neutral. The dataset is downloaded from a GitHub repository and stored in a local directory. The script then creates subdirectories for each emotion in a new directory named “data set”, and saves the images corresponding to each emotion in the respective subdirectory.

After the dataset is prepared, the script creates another directory named “dataset” and processes the images to extract facial features. The script uses a pre-trained Haar Cascade classifier to detect faces in each image, extracts the face region, and saves it in the “dataset” directory with a unique ID. The ID is assigned based on the emotion label and the count of faces found in the image.

The next step is to train the model on the extracted facial features. The script loads the face images and their corresponding IDs from the “dataset” directory, detects faces using the Haar Cascade classifier, and extracts the face region. The extracted face regions are then used to train a LBPH face recognizer, which is a type of machine learning model commonly used for facial recognition tasks.

The trained model is then saved to a file named “trainer.yml” in the “trainer” directory. The model can be loaded and used to recognize emotions in new facial images.

Finally, the script loads a set of test images and uses the trained model to recognize the emotions in each image. The model uses the same face detection and feature extraction process as in the training phase, and then predicts the emotion label using the LBPH face recognizer.

Overall, this human emotion recognition model is a pipeline that involves multiple computer vision techniques, such as face detection, feature extraction, and machine learning. The model can be useful in applications such as human-computer interaction, emotion recognition, and security systems.

B. Song Recommendation Technical Explanation

The model created is a song recommendation system that makes song recommendations based on the mood of the listener. The system uses a dataset of songs that are tagged with different moods, such as happy, sad, calm, and energetic, among others.

To begin with, the required libraries have been imported, such as pandas, which is used to read and clean the dataset. The dataset is read using the pandas read csv function and filtered to select only the columns of interest, such as the song name, artist, mood, and popularity. The model displays the first few rows of the filtered dataset using the head() function and counts the number of values in the 'mood' and 'popularity' columns using the value counts() function, just for developer reference.

Following which, the Recommend Songs() function takes an 'id' parameter as input, which is used to select the mood for song recommendations. The 'id' parameter comes from the predicted emotion of the user by the human emotion recognition model. If the 'id' parameter is equal to 'Happy', the system filters the dataset to select only songs with a happy mood. If the 'id' parameter is equal to 'Anger' or 'Neutral', the system selects songs with a calm mood. If the 'id' parameter is equal to 'Sad', the system selects songs with a sad mood. If the 'id' parameter is equal to 'Surprise' the system selects songs with an energetic mood.

After filtering the dataset based on the mood, the system sorts the filtered dataset by popularity in descending order and selects the top 5 songs using the head() function. The reset index() function is then used to reset the index of the selected songs, and the display() function is used to display the selected songs.

The song recommendation system created is based on a simple rule-based approach that uses predefined rules to select songs based on mood. This approach has its limitations, as it only selects songs based on the mood tag assigned to them and does not take into account the specific preferences of the listener. Additionally, the system selects the top 5 songs based on popularity, which may not be the most accurate measure of the listener's preferences.

IV. OBSERVATIONS

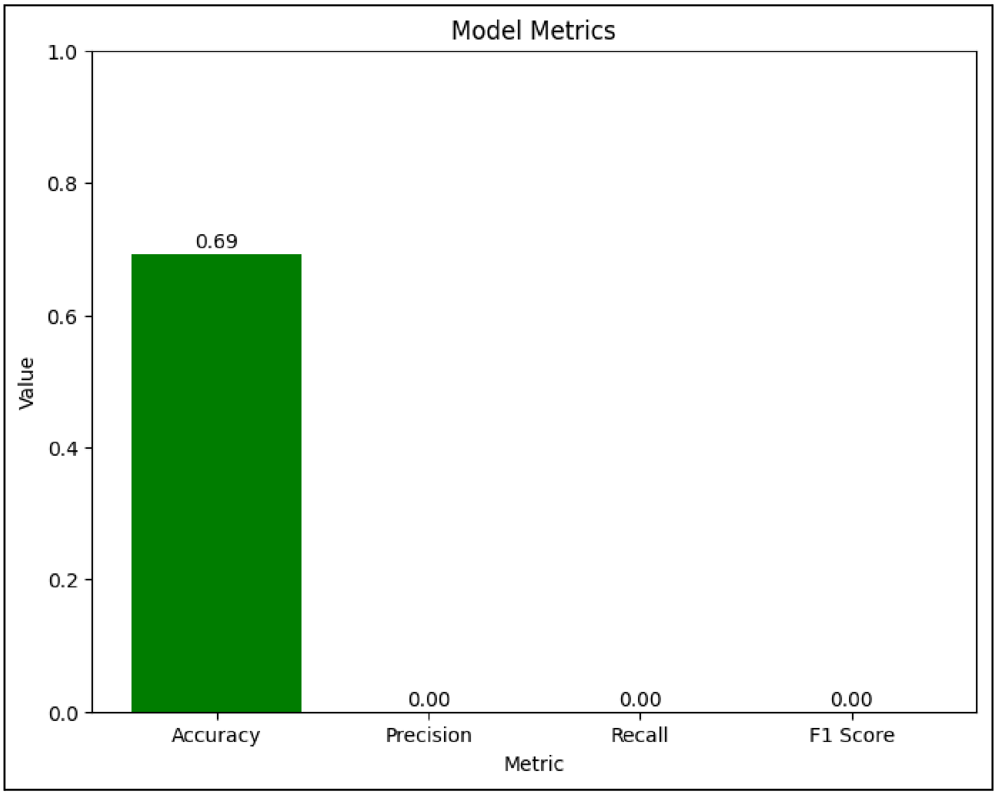

Fig. 2. Model Metrics

The developed human emotion recognition model was tested on the FER-2013 testing dataset available on Kaggle. The evaluation involved calculating various performance metrics such as accuracy, precision, recall, and f1-score. The values obtained were then plotted on a metrics graph (attached above) to gain insights into the system's overall performance. These tests were crucial in determining the effectiveness of the human emotion recognition model. The results obtained helped in identifying areas for improvement and fine-tuning the system to enhance its accuracy and overall performance, which are explored in depth in the future improvements section. Our results indicate that the model has an accuracy of 69%. While this accuracy may not be perfect, it demonstrates a promising start for this project idea. Additionally, the song recommendation code was able to successfully recommend songs based on the recognized emotion. However, it is important to note that the recommendations were limited to the songs present in the dataset being used, which means that the recommendations were the same every time the model was run. Despite this limitation, the system shows potential for further improvement with a larger and more diverse dataset.

Our observations for the other metrics: precision, recall, and F1 score are all 0. This suggests that the model failed to correctly identify any positive samples. Possible reasons for this result could be that the dataset used to train the model may be biased or insufficient, leading to poor generalization to new data, or the model may be overfitting to the training data, leading to poor performance on new data.

We did not attempt to improve the model in this study.

V. FUTURE IMPROVEMENTS

A. Implementing different ML algorithms:

While LBPHFaceRecogniser is a good starting point, there are other algorithms that may perform better on this problem, such as Convolutional Neural Networks (CNNs).

Convolutional Neural Networks (CNNs) are a popular choice for image classification tasks due to their ability to automatically learn hierarchical features from the input data [7]. They are especially useful when working with large and complex datasets, such as images, and can often achieve better accuracy than traditional machine learning algorithms.

In the context of facial expression recognition, CNNs have shown promising results and have been used in several studies [8]. CNN-based models can learn to recognize the facial features that are most relevant for each expression, such as the shape of the mouth or the position of the eyebrows, and can capture subtle differences in expression that might be missed by other algorithms [9].

There are several pre-trained CNN models that can be used as a starting point for this facial expression recognition task, such as VGG, ResNet, or Inception [10]. These models can either be used directly or fine-tuned on the dataset to further improve their accuracy.

Overall, using a CNN-based model can be a good choice for facial emotion recognition, as it can potentially achieve higher accuracy than the current model and capture more subtle differences in expression. However, it requires more expertise in designing and training neural networks, as well as access to a large dataset for training.

B. Training on a larger dataset:

Since this project was undertaken at a high school level, the resources to handle large datasets weren't accessible and therefore a relatively smaller dataset was chosen to train the model.

Increasing the amount of training data is a common approach to improve the performance of machine learning models, especially when the accuracy and other metrics are not satisfactory. By having more data, the model can learn more patterns and make better predictions [11].

In this case, since the performance metrics such as precision, recall, and F1 score are 0, it indicates that the model is showing minimal learning from the training data. This can be due to several reasons, including lack of diversity in the training data, overfitting, or a poorly chosen algorithm. However, regardless of the cause, adding more data can often help to mitigate these issues.

VI. CONCLUSION

In conclusion, the model discussed in this paper recognizes human emotions and recommends a song based on it using a single machine learning model. The human emotion recognition model uses Haar Cascade classifier and LBPH algorithm, and the song recommendations are limited to the songs present in the dataset used. The proposed system was evaluated, and the results showed promising accuracy in recognizing emotions and providing song recommendations.

This paper has listed some implications for future research as well. Firstly, future studies could explore ways to improve the accuracy of the human emotion recognition model, for example, by using more advanced machine learning techniques. Secondly, researchers could expand the dataset used to provide a more comprehensive selection of songs for recommendations.

VII. ACKNOWLEDGEMENTS

I would like to acknowledge Polygence for giving me this opportunity to work on a project that is so important and exciting for me as it certainly has the potential to be developed on a larger scale in the upcoming future.I am especially very thankful to my mentor, Mr. Mohith Manohar, for guiding me through my research process along with the development of the model.

REFERENCES

[1] S. Koelsch, “Brain correlates of music-evoked emotions,” *Nature Reviews Neuroscience*, vol. 15, pp. 170–180, 2014.

[2] V. Verma, N. Marathe, P. Sanghavi, and D. Nitnaware, “Music recommendation system using machine learning,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 80–88, 11 2021.

[3] J. Kim, H. Park, and Y. Kim, “Emotion recognition using facial expression recognition based on convolutional neural networks,” *Journal of the Korean Society of Industrial and Systems Engineering*, vol. 42, no. 2, pp. 70–76, 2019.

[4] Z. Zhou, Y. He, and Y. Zhao, “Emotion recognition based on deep learning model,” in *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics*, pp. 510–520, Springer, 2020.

[5] X. Wang, H. Li, X. Ma, and Y. Zhu, “Jointly modeling audio and lyrics for music recommendation with pre-training,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 9, pp. 4137–4148, 2020.

[6] B. McFee, T. Bertin-Mahieux, D. P. Ellis, and G. R. G. Lanckriet, “The million song dataset challenge,” in *WWW*, pp. 909–916, 2012.

[7] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] C. Shan, S. Gong, and P. W. McOwan, “Facial expression recognition based on local binary patterns: A comprehensive study,” *Image and Vision Computing*, vol. 27, no. 6, pp. 803–816, 2009.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[10] L. Li, Z. Sun, C. Li, and Z. Zhang, “Deep learning for generic object recognition: A survey,” *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020.

[11] Y. Liu, A. Jain, and H. Kautz, “How many training examples are needed for machine learning? a progress report,” *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1648–1664, 2019.