KEC Conference 2024
ISSN: 3021-940X (Print) DOI: https://doi.org/10.3126/injet.v1i2.66709

Facial Emotion Recognition System Using CNN for Song Mapping

Narayan Paudel[1], Saugat Neupane[2], Smarika Shrestha[3], Suyasha Nepal[4*], Pralhad Chapagain[5]

[1]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, paudelnarayan434@gmail.com*
[2]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, saugatneupane50@gmail.com*
[3]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, smareeka.shrestha@gmail.com*
[4]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, nepalsuyasha@gmail.com*
[5]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, pralhadchapagain@kec.edu.np*

Abstract

This project introduces a novel system that integrates Facial Emotion Recognition (FER) with music mapping to enhance human-computer interaction. Leveraging a custom Convolutional Neural Network (CNN) algorithm, developed using the FER2013 dataset, we classify emotions into four categories- happy, sad, neutral, and angry- employing Haar Cascade for precise face detection and grayscale conversion for optimal CNN input. Our custom CNN demonstrates superior performance, achieving a testing accuracy of 77.23%, notably surpassing established models like VGG16 and ResNet50, which achieved 55.87% and 62.76% respectively. This system swiftly identifies emotions and recommends songs from the "278k Emotion Labeled Spotify Songs" playlist, aiming to boost user satisfaction. Through nuanced comparisons with VGG16 and ResNet50, our approach underscores its inherent strengths, suggesting promising advancements in Facial Emotion Recognition (FER) and music recommendation systems. With a focus on precision in emotion detection, our system subtly elevates user experience, contributing meaningfully to ongoing research in the field.

Keywords: Facial Emotion Recognition, Music Mapping, CNN, Haar Cascade, Grayscale Images

1. Introduction

Emotion detection through facial recognition involves analyzing facial expressions to discern and classify emotions displayed by individuals, which are crucial indicators of human feelings. Facial expressions are the vital identifiers for human feelings, because it corresponds to the emotions (Mehendale, 2020). A facial expression is a complex movement of the facial muscles that conveys the subject's feelings to others (Kumar, et al., 2021). This emotion detection process utilizes computer vision algorithms to track vital facial features and movements, such as the eyes, eyebrows, nose, and mouth, to detect emotions like happiness, sadness, and anger. Feature detectors or filters help identify various features present in the image such as edges, vertical lines, horizontal lines and bends (Athavle, et al., 2021). Convolutional Neural Networks (CNNs) are instrumental in this endeavor, extracting features from facial images for tasks like gender and emotion recognition. Integrating emotion detection into systems like the emotion-based song mapping platform enhances user experiences by tailoring music selections to match their current emotional state, marking a convergence of artificial intelligence, music psychology, and computer vision. This fusion displays more personalized and effective interactions, underscoring the significance of emotions in refining user satisfaction

**Corresponding Author*

enhances user experiences by tailoring music selections to match their current emotional state, marking a convergence of artificial intelligence, music psychology, and computer vision. This fusion displays more personalized and effective interactions, underscoring the significance of emotions in refining user satisfaction with music experiences.

2. Related Works

Emotion detection technology holds the potential to enhance user engagement, tailor services to individual emotional states, and provide valuable insights into emotional responses to products and experiences. Understanding and recognizing emotions are crucial for various applications, such as human-computer interaction and personalized user experiences. Previous projects on emotion detection have been undertaken because of the significant influence that emotions wield over human behavior, communication, and decision-making.

In a paper (Maharjan, et al., 2023), the authors conducted a comprehensive study on facial emotion recognition using the FER-2013 dataset. The dataset consists of facial expressions categorized into seven emotion classes: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. The focus of this project was to extract distinct and precise facial features to aid in classifying individual images for identification purposes. To address the issue of imbalanced classification, they employed oversampling techniques. The pre-processing pipeline first detected faces using the Haar Cascade algorithm and then transformed the detected faces into grayscale images. The feature extraction step was performed using the Local Binary Pattern (LBP) algorithm. The output obtained was fed to CNN, and the model was trained using a range of hyperparameters: 60 epochs were utilized to prevent overfitting. The Adam optimizer was chosen to update the model's weights during training, and a learning rate was set accordingly to optimize the convergence process. The test data fed to the model resulted in 65.617% of testing accuracy. Overall, the approach presented in this paper showcased a systematic and practical methodology for facial emotion recognition, combining the power of LBP feature extraction with the discriminative capabilities of a CNN architecture.

In a paper (Bhadangkar & Pujari, 2020), accessible datasets, namely CK+, JAFFE and FED dataset were utilized. In the realm of Deep Learning, Convolutional Neural Network (CNN), Deep Dense Network (DDN), and Recurrent Neural Network (RNN) were utilized, and CNN demonstrated superior accuracy compared to the other two. This result suggests that CNN is the most effective algorithm for emotion identification and classification within its respective paradigms. The study's findings highlight the potential of employing advanced Deep Learning techniques, particularly CNN, in face emotion recognition tasks and underscore the importance of considering different algorithmic approaches to achieve optimal accuracy in emotion classification tasks.

The FERC is based on a two-part convolutional neural network (CNN): The first part removes the background from the picture, and the second part concentrates on the facial feature vector extraction. In FERC model, the expressional vector (EV) is used to identify the five different types of regular facial expression (Mehendale, 2020). The FERC method finds applications in human-computer interaction, psychiatric observations, drunk driver recognition, and lie detection, with the added capability of processing both image and video inputs. Building upon the insights gained from FERC, this research aims to enhance facial emotion recognition systems, thereby enriching user experiences and extending emotion-aware computing applications across diverse domains.

The fundamental differentiating factor of our research, contrasting existing Literature Reviews, is its unique utilization of the FER2013 dataset for facial emotion recognition, seamlessly integrated with the 278k Emotion Labeled Spotify Songs dataset to facilitate personalized song mapping. While previous studies have explored various algorithms for emotion recognition, such as SVM, KNN, CNN, and LBP, and focused on different datasets, our study endeavors to bridge the gap between facial emotion recognition and music mapping, offering a novel and comprehensive approach to enriching user experiences in emotion-aware computing. Notably, through a comparative analysis with established models like VGG16 and ResNet50, our project demonstrates the superiority of our customized CNN model in emotion recognition tasks, marking a significant advancement in the field. Additionally, our project addresses the challenge of imbalanced classification by employing data augmentation techniques to enhance the robustness and generalization of the

emotion recognition model, further contributing to the progression of emotion-based technology and its potential applications in interactive entertainment, music streaming platforms, and personalized therapeutic interventions. This unique combination of datasets, pre-processing techniques, and deep learning methodologies sets our research apart from previous literature, making it a valuable and innovative contribution to emotion recognition and music mapping.

3. Methodology

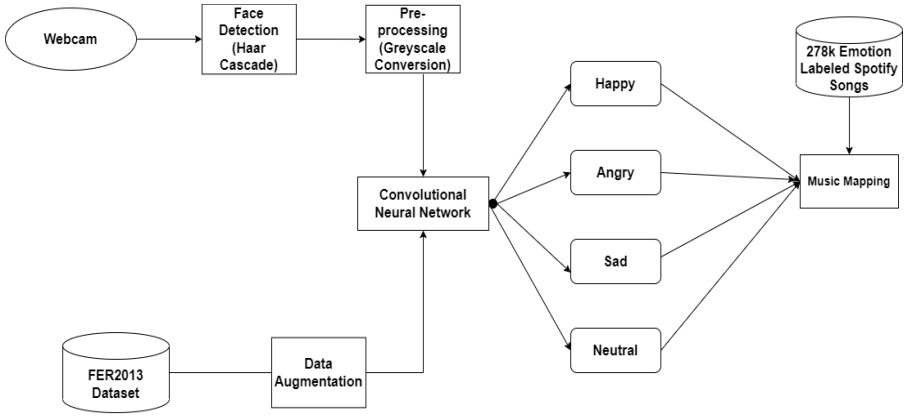

Figure 1. Working Mechanism of Facial Emotion Recognition using CNN for Song Mapping

3.1 Dataset Description

3.1.1 FER-2013

The FER-2013 dataset comprises a total of 26,217 images, each meticulously labeled to represent various emotions including anger, happiness, sadness, and neutrality. These labeled facial images collectively offer a comprehensive collection, enabling thorough exploration and analysis of emotional expressions. (ResearchGate, 2020)

Figure 2. FER-2013 data samples

This dataset is divided into training and testing datasets, with the training dataset consisting of 21,005 images and the testing dataset consisting of 5,212 images. The train and test data for each emotion class is as given:

Table 1. Train and Test Data

Data	Train	Test
Angry	3995	958
Happy	7215	1774
Neutral	4965	1233
Sad	4830	1247

Table 2. Calculation of bearing capacity by Trial-and-Error method

3.1.2 278k Emotion Labeled Spotify Songs

This dataset from Kaggle includes 278,000 Spotify tracks, each tagged with emotions like happy, sad, angry, or neutral. It is used to pair songs with the detected facial emotions, creating a music recommendation system that caters to individual moods and tastes. This rich, emotion-labeled collection is vital for research into emotion-based music recommendation systems.

3.2 Data Pre-Processing

We finalized the FER2013 dataset, selecting four emotion classes— angry, happy, neutral, and sad- relevant to our project's playlist, which comprises songs labeled with corresponding emotions. The selected classes showed an imbalance in data. Imbalanced datasets are harmful because they bias models towards majority class predictions. Imbalanced datasets also render accuracy a deceitful performance metric (Shorten & Khoshgoftaar, 2019). To address data imbalance, we performed data augmentation by applying transformations such as rotation, height and width shifting, and zooming. These transformations were applied with the following parameters: rotation range of 10 degrees, height shift range of 0.1, width shift range of 0.1, and zoom range of 0.1. Additionally, we used a fill mode of "nearest" to handle any empty pixels resulting from the augmentation process. Thus, we balanced each emotion class to twice the size of the 'happy' class.

3.3 Algorithm Description

Convolutional Neural Network

A convolutional neural network is a unique deep neural network model. It is particularly reflected in two aspects. On the one hand, the connections between its neurons are not fully connected. On the other hand, in the same layer, some nerves share connection weights between the elements. It is precisely because of these two unique properties that the number of parameters is reduced, which in turn dramatically reduces the complexity of the network model (TheClickReader, 2022).

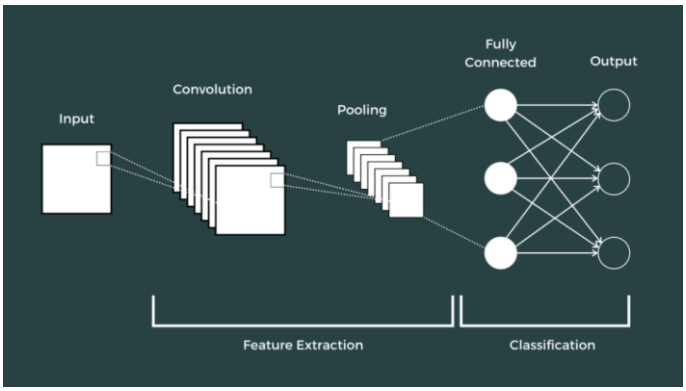

Figure 3. CNN Architecture

Convolutional neural networks usually contain the following layers:

1. Convolutional Layer

The primary function of the network's core component is to extract distinct characteristics from the input. The features extracted by shallow networks are often low-level, such as edges, lines, and corners. As the number of network layers increases, the network can extract increasingly complex features.

$$X_j^l = f\left(\sum_{i \in M_j} X_i^{l-1} * k_{ij}^l + b_j^l\right) \quad \text{(Equation 1)}$$

Equation (1) is the convolution calculation formula. l represents the current layer in the neural network; X^l_j is output of the jth feature map in layer l, capturing specific patterns; i is the index iterating over convolution regions (receptive fields); X_i^{l-1} is the input data from the previous layer; k_{ij}^l is the convolutional kernel (filter) for the jth feature map in layer l; b_j^l is the bias term associated with the jth feature map; f is the activation function applied to the convolution result; M_j is the set of convolution regions for the jth feature map.

2. Pooling Layer

After the convolutional layer, features with large dimensions were obtained. The features were divided into several regions, and the maximum or average value nine was taken to obtain new features with smaller dimensions.

3. Activation Function

Each layer of the neural network applies linear transformations exclusively, resulting in a model with insufficient expressive capability. Introducing an excitation function and incorporating nonlinear factors led to improved performance in solving complex problems.

4. Fully Connected Layer

The obtained local features were combined into global features to calculate the score of each last category.

3.4 Model Description

3.4.1 Customized CNN

A customized Convolutional Neural Network (CNN) is a tailored architecture designed to address specific tasks or datasets within computer vision. It involves fine-tuning components, such as layer configurations, activation functions, and regularization techniques to optimize performance. This customization process begins with clearly defining the task at hand, followed by data collection and pre-processing steps to ensure dataset suitability. Experimentation with different architectural choices during model design allows for adaptation to the unique characteristics of the problem domain. Through iterative training and evaluation phases, the effectiveness of the CNN is assessed and refined, ultimately leading to its deployment within production environments for real-world applications.

In constructing our CNN model, we employed a strategic layering approach, starting with two Conv2D layers with 32 and 64 filters for initial feature extraction from 48x48 grayscale images. Batch normalization was applied to stabilize learning, followed by max pooling and dropout to reduce overfitting. We then introduced two sets of Conv2D layers with reduced filters and incorporated SeparableConv2D layers to minimize parameters without compromising feature learning. The model was regularized using L2 regularization and further dropout, and concluded with a dense layer of 256 neurons and an output layer with 4 neurons, reflecting the number of classes. This architecture was iteratively refined, balancing model complexity and computational efficiency, and validated to ensure robust generalization.

Layer Name	Output Size	Kernel Size	Activation	Details
Input	48x48	-	-	Grayscale image input (48x48x1)
Conv2D_1	48x48	3x3	ReLU	32 filters, padding='same'
Conv2D_2	48x48	3x3	ReLU	64 filters, padding='same'
BatchNorm_1	48x48	-	-	Batch normalization
MaxPool2D_1	24x24	2x2	-	Max pooling (2x2)
Dropout_1	24x24	-	-	Dropout (25%)
Conv2D_3	24x24	3x3	ReLU	96 filters, padding='same'
Conv2D_4	24x24	3x3	ReLU	96 filters, padding='same'
BatchNorm_2	24x24	-	-	Batch normalization
MaxPool2D_2	12x12	2x2	-	Max pooling (2x2)
Dropout_2	12x12	-	-	Dropout (25%)
SepConv2D_1	12x12	3x3	ReLU	128 filters, padding='same'
SepConv2D_2	12x12	3x3	ReLU	128 filters, padding='same'
BatchNorm_3	12x12	-	-	Batch normalization
MaxPool2D_3	6x6	2x2	-	Max pooling (2x2)
Dropout_3	6x6	-	-	Dropout (25%)
Flatten	1x1	-	-	Flatten layer
Dense_1	256	-	ReLU	Fully connected (dense) layer
BatchNorm_4	256	-	-	Batch normalization
Dropout_4	256	-	-	Dropout (50%)
Dense_2	4	-	Softmax	Output layer (4 classes)

Figure 4. Detailed Customized CNN Architecture

Table 3. Model Summary (CNN)

Layer (type)	Output Shape	Parameter Count
conv2d (Conv2D)	(None, 48, 48, 32)	320
conv2d_1 (Conv2D)	(None, 48, 48, 64)	18496
batch_normalization (BatchNormalization)	(None, 48, 48, 64)	256
max_pooling2d (MaxPooling2D)	(None, 24, 24, 64)	0
dropout (Dropout)	(None, 24, 24, 64)	0
conv2d_2 (Conv2D)	(None, 24, 24, 96)	55392
conv2d_3 (Conv2D)	(None, 24, 24, 96)	83040
batch_normalization_1 (BatchNormalization)	(None, 24, 24, 96)	384
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 96)	0
dropout_1 (Dropout)	(None, 12, 12, 96)	0
separable_conv2d (SeparableConv2D)	(None, 12, 12, 128)	13280
separable_conv2d_1 (SeparableConv2D)	(None, 12, 12, 128)	17664
batch_normalization_2 (BatchNormalization)	(None, 12, 12, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 128)	0
dropout_2 (Dropout)	(None, 6, 6, 128)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 256)	1179904
batch_normalization_3 (BatchNormalization)	(None, 256)	1024
dropout_3 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 4)	1028

Total Parameters: 1,371,300

3.4.2 VGG16

VGG16, as its name suggests, is a 16-layer deep neural network (DataGen, n.d.). It is a specific Convolutional Neural Network (CNN) architecture that consists of 13 convolutional layers and three fully connected layers. The convolutional layers use small 3x3 filters with a stride of 1 and zero-padding, followed by max-pooling layers to reduce spatial dimensions. VGG16 is known for its uniform architecture, where convolutional layers are stacked on each other, resulting in a straightforward and easy-to-understand design (VARSHNEY, 2020).

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 5. Detailed VGG Architecture

Table 3. Model Summary (VGG16)

Layer (type)	Output Shape	Parameter Count
vgg16(Functional)	(None, 1, 1, 512)	14714688
flatten(Flatten)	(None, 512)	0
dense(Dense)	(None, 512)	262656
dense_1(Dense)	(None, 4)	2052

Total Parameters: 14,979,396

3.4.3 Resnet50

ResNet50 is a convolutional neural network architecture renowned for its depth and efficacy in computer vision tasks. Developed by Microsoft Research, it features 50 layers, including residual blocks with shortcut connections, addressing the vanishing gradient problem in deep networks. These shortcuts enable ResNet50 to learn residual mappings, facilitating the training of intense models. Leveraging its depth and skip connections, ResNet50 captures intricate features and patterns, making it a preferred choice for tasks such as image classification, object detection, and segmentation. While ResNet models with higher depth (e.g., ResNet101, ResNet150) may offer increased representational capacity, they correspondingly exhibit a higher risk of overfitting, particularly in scenarios with limited training data. Given the observed high overfitting in ResNet50, it is reasoned that models with greater depth would likely exacerbate this issue. Therefore, ResNet50 emerged as the optimal choice, striking a balance between model complexity and generalization performance within the constraints of the dataset.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 6. Detailed ResNet Architecture *(OpenGenusIQ, n.d.)*

Table 4. Model Summary (ResNet50)

Layer (type)	Output Shape	Parameter Count
resnet50 (Functional)	(None, 2, 2, 2048)	23587712
global_average_pooling_2d(GlobalAveragePooling2D)	(None, 2048)	0
dropout(Dropout)	(None, 2048)	0
dense(Dense)	(None, 4)	8196

Total Parameters: 23,595,908

3.5 Experimental Results

Table 5. Hyperparameters used

Parameter	Value
Epoch	60
Batch size	64
Optimizer	Adam
Learning rate	0.0001

3.5.1 Customized CNN

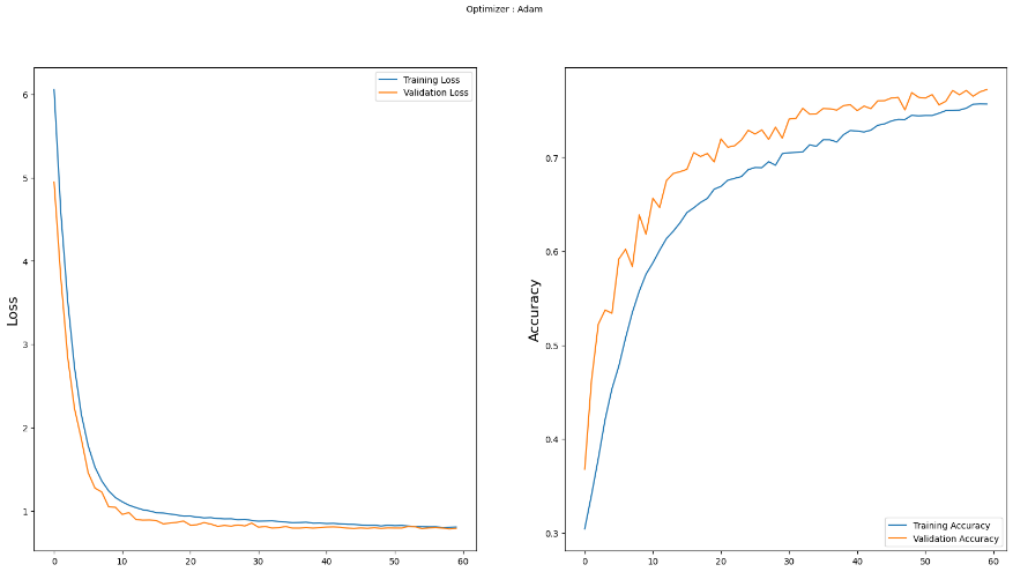

Figure 7. Loss and Accuracy (CNN)

Testing Accuracy: 0.7723
Testing Loss: 0.7900

3.5.2 VGG16

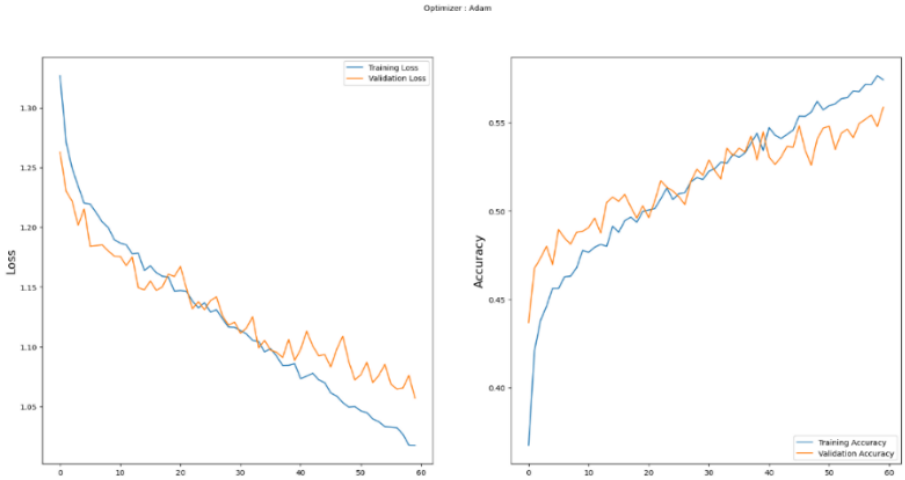

Figure 8. Loss and Accuracy (VGG16)

Testing Accuracy: 0.5587
Testing Loss: 1.0571

3.5.3 ResNet50

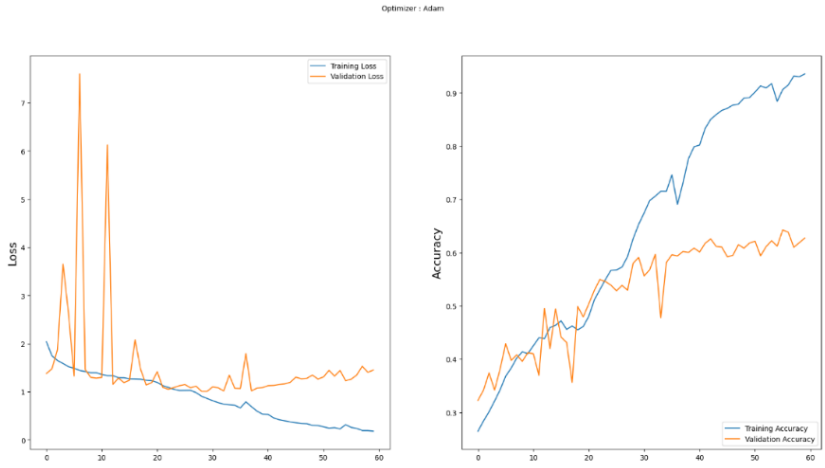

Figure 9. Loss and Accuracy (ResNet50)

Testing Accuracy: 0.6276
Testing Loss: 1.4512

3.6 Model Comparison

Table 6. Model Comparison

Model	Accuracy	precision	recall	f1-score	Total Parameters
Custom. CNN	0.7723	0.773	0.772	0.771	1,371,300
VGG16	0.5587	0.557	0.559	0.558	14,979,396
ResNet50	0.6276	0.638	0.628	0.631	23,595,908

This table compares the performance of three machine learning models: Custom CNN, VGG16, and ResNet50. The comparison is based on four key metrics: Accuracy, Precision, Recall, and F1-Score. The Custom CNN model appears to be the most effective, with all metrics approximately at 0.772. The VGG16 model has the lowest performance, with scores around 0.558 for each metric. Meanwhile, the ResNet50 model has moderate performance levels, with an Accuracy of 0.6276, Precision of 0.638, Recall of 0.628, and an F1-Score of 0.631. This data suggests that the Custom CNN model is superior to the other two in terms of these metrics.

The decision to use a lightweight custom CNN model, as opposed to larger pretrained models like VGG16 or ResNet50, is driven by several practical considerations. Firstly, a smaller model demands less computational power, which translates to cost and time savings during both training and inference phases. This makes it particularly appealing for deployment in environments with limited computational resources. Secondly, the streamlined architecture leads to faster inference times, a crucial factor for real-time applications. Thirdly, lightweight models are less susceptible to overfitting, especially when the available training data is not extensive. Additionally, they have a smaller memory footprint, making them suitable for devices where storage is at a premium. Lastly, they are more energy-efficient, which is beneficial for battery-operated devices. The overarching goal of designing a lightweight model is to achieve an optimal balance between predictive performance and resource efficiency, enabling the deployment of machine learning solutions in scenarios where larger models would be impractical due to their hefty resource requirements.

3.7 Song Mapping

In the final phase, we implemented the Song Mapping feature to enhance the user experience by providing personalized music recommendations based on the detected facial emotions. Leveraging the output labels generated by our facial emotion recognition model in real-time, we established a connection to the URI of the ’278k Emotion Labeled Spotify Songs’ dataset. This capability facilitated the dynamic retrieval of a random selection of songs that corresponded to the detected emotions. By integrating this functionality into our user interface, we enabled users to seamlessly explore and enjoy music that resonates with their current emotional state.

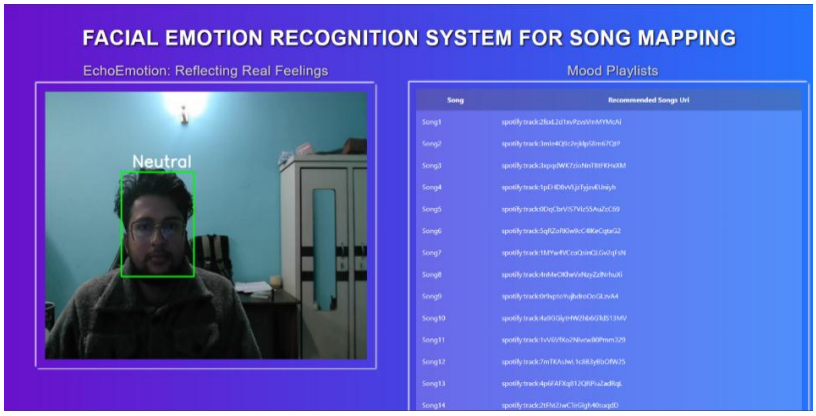

Figure 10. User Interface

5. Discussion

This study presented a novel Facial Emotion Recognition System using a custom Convolutional Neural Network (CNN) for song mapping, demonstrating a significant advancement in human-computer interaction. The custom CNN model achieved a testing accuracy of 77.23%, outperforming established models like VGG16 and ResNet50. This success can be attributed to the meticulous dataset preparation, including data augmentation to address class imbalances, and the strategic layering of the CNN.

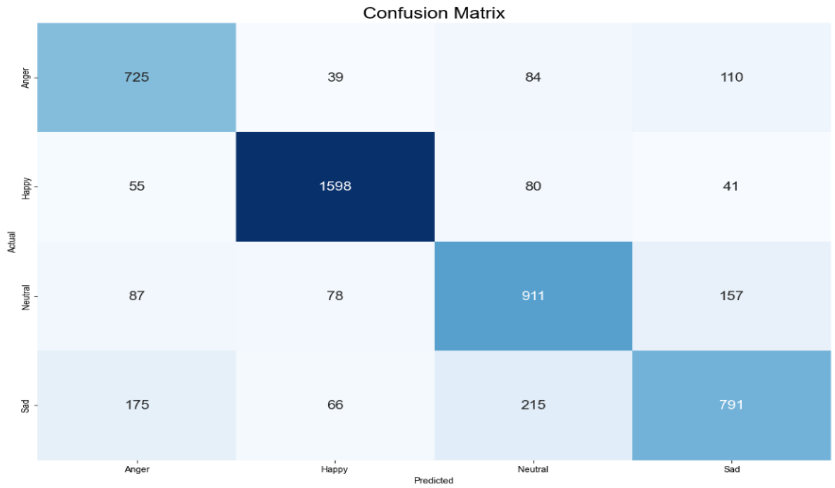

Figure 11. Confusion Matrix (Customized CNN)

The confusion matrix reveals varying frequencies of correct and incorrect predictions for each emotion category. Among the highest figures, "happy x happy" recorded the highest count with 1598 instances, indicating the model's accurate prediction of happiness when the actual emotion was also happy. Conversely, among the lowest figures, "happy x anger" had the lowest count with only 39 instances, indicating instances where the model incorrectly predicted happiness when the actual emotion was anger. These extremes highlight the model's performance across different emotion categories, offering valuable insights for further refinement and improvement.

The integration of the FER2013 dataset with the '278k Emotion Labeled Spotify Songs' dataset for personalized song mapping represents a unique convergence of emotion recognition technology and music psychology. The system's ability to recommend songs based on detected emotions offers a more immersive and satisfying user experience.

6. Conclusion and Future Enhancements

Based on the research conducted using the FER2013 dataset for facial emotion recognition, which involved the development and comparison of three models (customized CNN, VGG16, and ResNet50), several key findings have emerged. Firstly, our study revealed that the customized CNN model demonstrated superior accuracy in comparison to the pretrained VGG16 and ResNet50 models. This finding underscores the efficacy of a tailored approach to neural network architecture design, specifically customized for the task at hand, in achieving better performance outcomes. Secondly, our research emphasized the critical role of dataset characteristics in model selection and performance. Our findings indicate that for datasets such as FER2013, which exhibit limited variability and complexity, employing a customized approach enables better representation of relevant features, resulting in superior performance in facial emotion recognition tasks. It is important to acknowledge the well-established effectiveness of ResNet50 and VGG16 models in a wide range of computer vision tasks. However, our focus lies in showcasing the superiority of the customized CNN model tailored explicitly for the FER2013 dataset. This finding highlights the importance of selecting appropriate models tailored to the unique challenges of specific datasets, ultimately contributing to the advancement of facial emotion recognition technology.

Moving forward, several potential avenues for future enhancements in our project include incorporating multiple modalities such as audio, text, and facial images to enhance emotion recognition accuracy by leveraging diverse sources of information. This could involve techniques such as multimodal fusion and deep learning architectures capable of handling diverse data types. By leveraging complementary information from these modalities, we anticipate improved robustness and accuracy in emotion recognition. Additionally, integrating user feedback mechanisms to continuously refine and personalize song recommendations based on individual preferences and emotional states would further improve the accuracy and effectiveness of our facial emotion recognition system. This could entail implementing collaborative filtering algorithms and sentiment analysis techniques to interpret user preferences and emotional states, ultimately providing more personalized and relevant recommendations to users.

Acknowledgements

We extend our heartfelt gratitude to all who contributed to the "Facial Emotion Recognition System Using CNN for Song Mapping" project. Special thanks to the Department of Computer and Electronics Engineering faculty of Kantipur Engineering College, Dhapakhel, including teachers Bishal Thapa, Nishan Khanal, Pawan Acharya and staff for their support. We also appreciate the guidance of our peers and mentors, as well as the dedication of our team members. Together, we have not only achieved our project goals but have also enriched our collective learning experience. We extend our sincerest thanks to everyone involved for their indispensable contributions and unwavering support throughout this endeavor.

References

Athavle, M., Mudale, D., Shrivastav, U. & Gupta, M., 2021. Music Recommendation Based on Face Emotion Recognition. Journal of Informatics Electrical and Electronics Engineering (JIEEE), 2(2), pp. 1-11.

Bhadangkar, D. K. & Pujari, J. D., 2020. Comparative analysis of Identification and Classification of Face Emotions Using Different Machine Learning and Deep Learning Algorithms. Turkish Journal of Computer and Mathematics Education, 11(3), p. 1708–1722.

DataGen, Understanding VGG16: Concepts, Architecture, and Performance. [Online]
Available at: https://datagen.tech/guides/computer-vision/vgg16/ [Accessed 3 March 2024].

Kumar, B. K., Swaroopa, K. & Balaga, T. R., 2021. Facial Emotion Recognition and Detection Using CNN. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(14), pp. 5960-5968.

Maharjan, S., Maharjan, R., Ghimire, S. & Bhattarai, N., 2023. Emotion Recognition Sysem. International Conference on Engineering and Technology, 5(4), pp. 22-26.

Mehendale, N., 2020. Facial Emotion Recognition Using Convolutional Neural Networks (FERC). SN Applied Sciences, Volume 2, pp. 446-454.

OpenGenusIQ, Understanding ResNet50 architecture. [Online] Available at: https://iq.opengenus.org/

resnet50-architecture/?fbclid=IwAR0ipTnMTIaDZuzTXKerBmkL599NcK2pTH6Yg5pusd91Bdy2afNFiRE OK6M [Accessed 5 March 2024].

ResearchGate, S. F. o., 2020. Hybrid-Deep Learning Model for Emotion Recognition Using Facial Expressions. [Online] Available at: https://www.researchgate.net/figure/Sample-of-the-FER2013-dataset_fig1_343556711 [Accessed 3 January 2024].

Shorten, C. & Khoshgoftaar, T. M., 2019. A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1), pp. 1-48.

TheClickReader, 2022. Building a Convolutional Neural Network. [Online] Available at: https://www.theclickreader.com/building-a-convolutional-neural network/?fbclid=IwAR2VMCtAs51ZDgsI-Npe-ZDIUVFsSN8yOvFBZCwTvn9-asgiP5H1iPkUr3A [Accessed 5 March 2024].

VARSHNEY, P., 2020. VGGNet-16 Architecture: A Complete Guide. [Online] Available at: https://www.kaggle.com/code/blurredmachine/vggnet-16-architecture-a-complete-guide?fbclid=IwAR2DuxMjagksWqs0oEgC07VrzXXhEZ5vMU87rQyxZ79112ulcG7IVnAsDvY [Accessed 5 March 2024].