EMO SENSE (Emotion Recognition)

Introduction

Facial expression recognition (FER) has emerged as a significant field within computer vision and artificial intelligence, offering applications in areas such as human-computer interaction, psychological studies, and security systems. The objective of FER is to automatically identify and classify human emotions based on facial expressions. Traditional FER systems focus on recognizing emotions from individual faces, leveraging algorithms or deep learning models trained on large datasets of facial expressions. The development of these systems involves preprocessing steps like face detection, followed by feature extraction and classification. Advanced deep learning techniques, such as convolutional neural networks (CNNs), have significantly enhanced the accuracy and robustness of these systems, making real-time emotion detection feasible.

Facial expression recognition at its core involves detecting and classifying facial emotions such as happiness, sadness, anger, surprise, fear, and disgust. The process begins with face detection, where algorithms like Haar cascades, or more recent methods using deep learning models such as YOLO (You Only Look Once), SSD (Single Shot Multibox Detector), MTCNN to identify the presence of faces within an image. Once the faces are detected, the next step involves extracting relevant features from these faces. Deep learning methods like CNNs, have revolutionized feature extraction by automatically learning hierarchical features from raw pixel data. The final stage involves classifying these features into predefined emotion categories

While individual facial expression recognition is well-established, recognizing emotions in group images presents additional challenges. Group facial expression recognition (GFER) involves analyzing multiple faces within a single image, each potentially exhibiting different emotions, and dealing with complex scenarios like occlusions, varying facial orientations, and diverse lighting conditions. GFER is crucial in social and behavioral analysis, enabling the understanding of collective emotional states in social gatherings, classrooms, or crowd events. The complexity increases with the need to accurately detect and recognize each face within a group, requiring robust face detection and alignment techniques. Moreover, GFER must handle interactions and relationships between different faces, which can provide context for more accurate emotion recognition.

Enhanced Spatial Attention (ESA):

The ESA layer emphasizes important regions of an image by applying a spatial attention mechanism. This mechanism helps the neural network concentrate on significant areas, which can lead to better performance. The network consist of average pool and max pool operations. Both pooling operations produce two single-channel feature maps from the multi-channel input. These operations help in summarizing the information across channels, highlighting different aspects of the input. The results of the average and max pooling are concatenated along the channel dimension, creating a two-channel feature map. This concatenation effectively combines different types of spatial summaries of the input features. A convolutional layer with a kernel size (typically 7x7) and a sigmoid activation function is applied to the concatenated feature map. This layer generates an attention map, a single-channel output that indicates the importance of each spatial location in the input. The attention map is element-wise multiplied with the original input feature map. This multiplication scales the original features according to their importance, as indicated by the attention map. by integrating the Enhanced spatial attention to regular convolution the performance of the network is slightly deviated and the network need more attention to increase the performance.

Convolution Block Attention Module (CBAM):

CBAM is an attention mechanism that can be integrated into Convolutional Neural Networks (CNNs) to improve performance by focusing on important features. It consists of two main components: Channel Attention Module (CAM) and Spatial Attention Module (SAM). CAM focuses on the channel-wise importance of features. It aims to highlight the significant channels and suppress less useful ones. Global average pool and global max pool operations summarize each channel's information across the spatial dimensions. Two dense layers (MLPs) are applied to the outputs of GAP and GMP. These layers share weights to reduce the number of parameters and computations. The outputs of the two MLPs (one from GAP and one from GMP) are combined using element-wise addition. A sigmoid activation function is applied to produce the channel attention map. The input feature map is multiplied by the channel attention map to scale the channels according to their importance. Furthermore, SAM focuses on the spatial importance of features. It aims to highlight significant regions and suppress less important ones. Average and max pooling operations are applied across the channel dimension to generate two spatial maps.

This structure, with integrated CBAM in the convolutional blocks, aims to improve the network's ability to learn discriminative features by focusing on important channels and spatial locations. The performance of this networks training accuracy and val accuracy with different evaluation metrics is shown below.

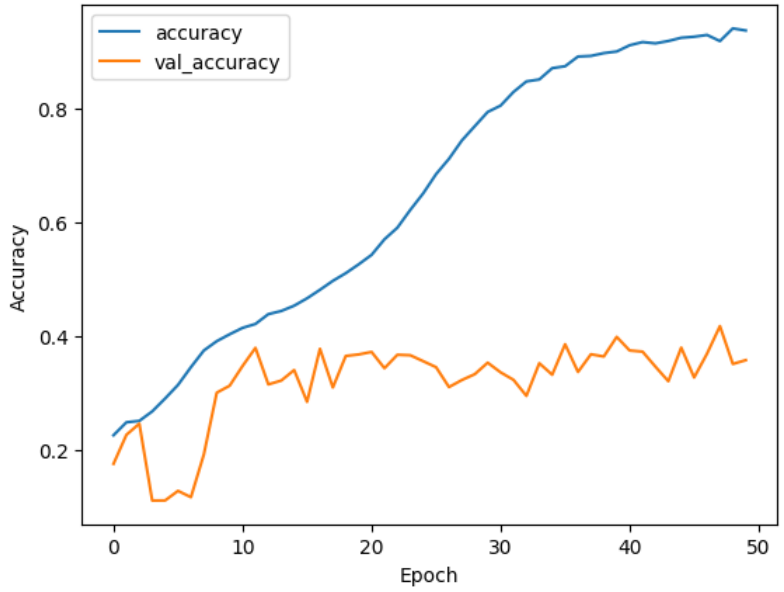

Fig.1. Training accuracy for CBAM Network

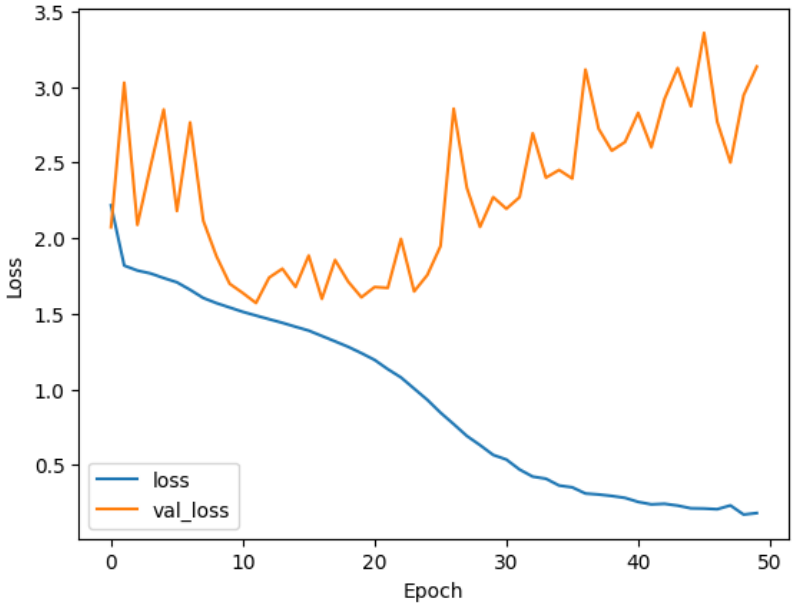

Fig.2. Training loss for CBAM Network

Table I

Evaluation Metrics for CBAM Network

Classes	Precision	Recall	F1-score
Angry	0.25	0.20	0.23
Disgust	0.26	0.12	0.16
Fear	0.25	0.20	0.22
Happy	0.57	0.59	0.58
Sad	0.24	0.41	0.30
Surprise	0.70	0.18	0.29
Neutral	0.34	0.36	0.35

Hierarchical Convolutional Block (HCB):

A convolutional block in a CNN is designed to extract hierarchical features from input images through a series of operations. Each block typically begins with a convolutional layer (Conv2D), which applies multiple filters to detect various patterns, such as edges or textures. Following this, an activation function like ReLU introduces non-linearity, allowing the network to learn complex features. Batch normalization stabilizes the learning process by normalizing the outputs of the convolutional layers, ensuring faster and more stable training. Next, a pooling layer (MaxPooling2D) reduces the spatial dimensions of the feature maps, enhancing computational efficiency and making the features invariant to small translations. Dropout layers then randomly deactivate a fraction of neurons during training to prevent overfitting, promoting the learning of robust features. By stacking these blocks, the network can progressively learn more abstract and high-level representations, which are crucial for tasks like emotion recognition from facial images. The performance of this networks training accuracy and val accuracy with different evaluation metrics is shown below.

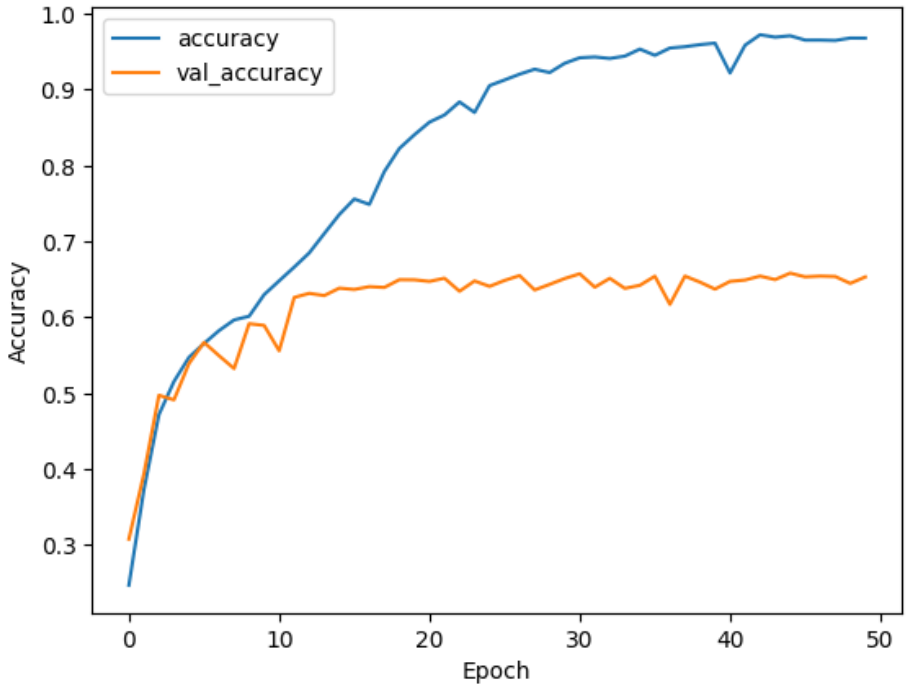

Fig.3. Training accuracy for HCB Network

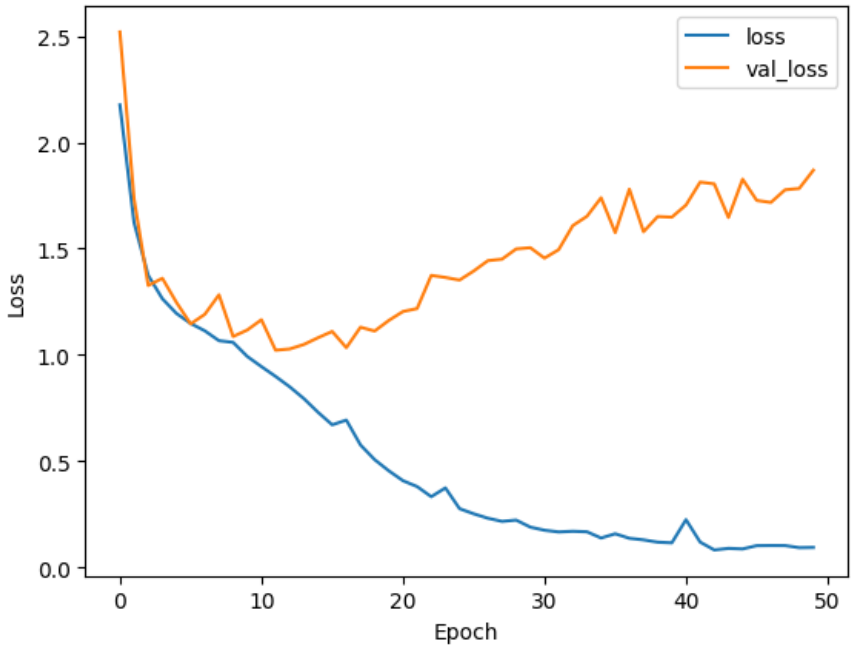

Fig.4. Training loss for HCB Network

Table II

Evaluation Metrics for HCB Network

Classes	Precision	Recall	F1-score
Angry	0.61	0.50	0.55
Disgust	0.69	0.65	0.67
Fear	0.48	0.52	0.50
Happy	0.87	0.80	0.84
Sad	0.54	0.49	0.51
Surprise	0.72	0.79	0.76
Neutral	0.56	0.67	0.61

Resnet :

The ResNet model is defined with an initial convolutional layer followed by several residual blocks. Each residual block consists of two convolutional layers with batch normalization and ReLU activation. A shortcut connection is added to bypass the convolutional layers, either directly or via a projection layer to match dimensions when necessary. This helps mitigate the vanishing gradient problem and allows for training deeper networks. After the residual blocks, global average pooling is applied to reduce the feature maps' dimensions before passing them to a fully connected layer with a softmax activation to output class probabilities. By stacking residual blocks, ResNet can learn a hierarchy of features, from low-level features like edges and textures to high-level features like shapes and objects. This hierarchical learning improves the model's ability to generalize from training data to unseen data. The use of residual connections simplifies the optimization of deep neural networks. Instead of learning a full transformation, each residual block only needs to learn the residual mapping, which is often simpler and more stable to optimize. ResNet models consistently achieve top performance on different models. The model is compiled with Adam optimizer and categorical cross-entropy loss, then trained for 30 epochs. Finally, the model's performance is evaluated, and accuracy and loss curves are plotted to visualize the training process.

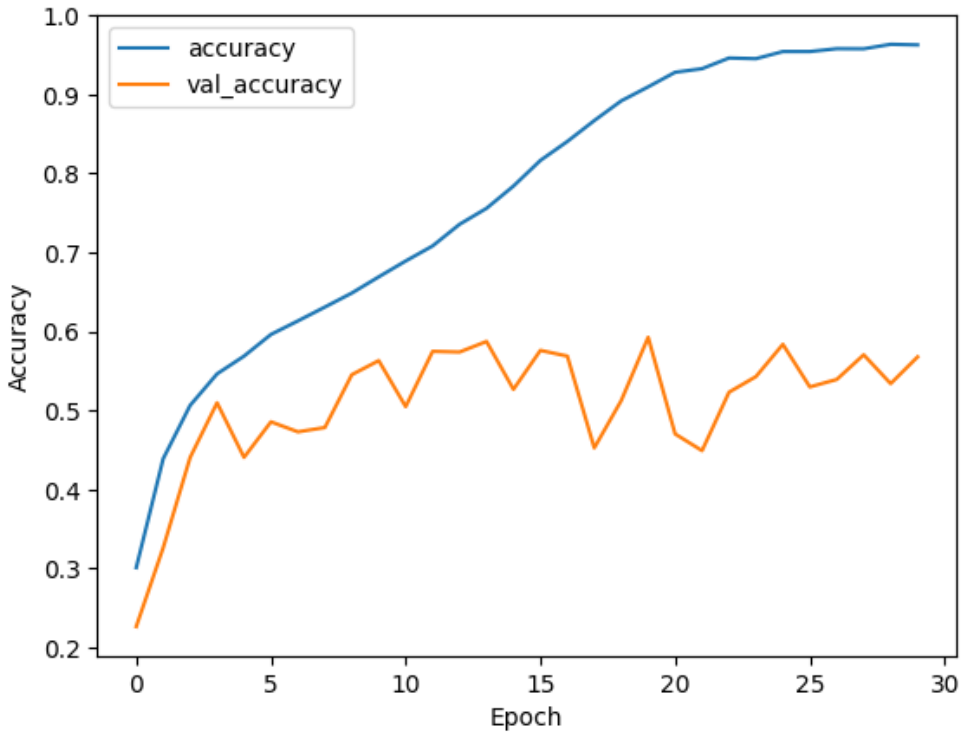

Fig.5. Training accuracy for Resnet Network

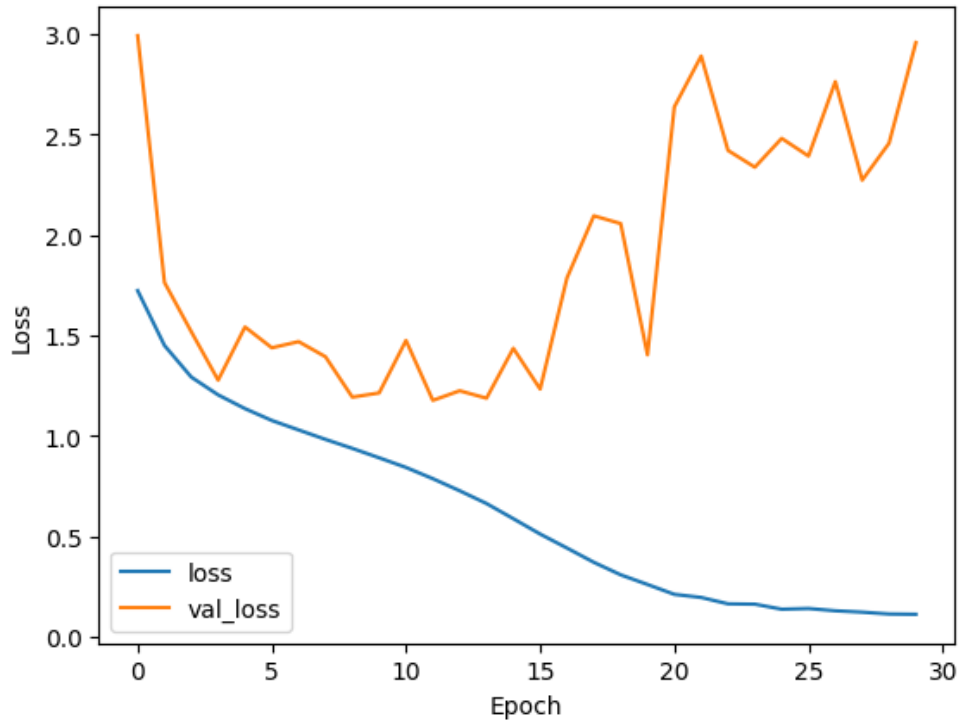

Fig.6. Training accuracy for Resnet Network

Table III

Evaluation Metrics for Resnet Network

Classes	**Precision**	**Recall**	**F1-score**
Angry	0.63	0.36	0.46
Disgust	0.88	0.27	0.42
Fear	0.48	0.45	0.46
Happy	0.81	0.77	0.79
Sad	0.69	0.13	0.22
Surprise	0.68	0.77	0.72
Neutral	0.39	0.85	0.53

IOU for Resnet :

IoU for class 0: 0.29831932773109243

IoU for class 1: 0.2641509433962264

IoU for class 2: 0.30206985769728334

IoU for class 3: 0.6541966426858513

IoU for class 4: 0.12275215011727912

IoU for class 5: 0.5683918669131238

IoU for class 6: 0.3633934535738143