Natural Resources for Human Health
eISSN: 2583-1194

Original Research

Received 16 May 2026
Revised 30 June 2026
Accepted 18 July 2026

Natural Resources for Human Health 2026, Vol. 6 (8s): 1016–1035

KEYWORDS:
Facial Emotion Recognition; Real-Time Video Analysis; Deep Learning; Convolutional Neural Networks; Bi-LSTM; Attention Mechanism

Emotionnet: A Deep Learning Framework for Real-Time Video-Based Facial Emotion Recognition

[1]Amit Rehapade, [2]Prafulla Ajmire, [3]Kiran Varma

[1]Dept. of Computer Sci. and Engineering S.G.B. Amravati University Amravati, India amitrehapade@gmail.com
[2]Dept. of Computer Sci. and Engineering S.G.B. Amravati University Amravati, India peajmire@gmail.com
[3]Dept. of Computer Sci. and Engineering S.G.B. Amravati University Amravati, India arow.kiran@gmail.com

ABSTRACT: EmotionNet is a neural network that is used to recognize facial emotions in real-time and under dynamic and unconstrained conditions using videos. The suggested system combines both spatial features extraction and time modeling to ensure the capture of facial appearance and motion-based emotion features. A convoluted neural network (CNN) backbone that is lightweight is used to extract discriminative spatial characteristics of successive video frames and a bidirectional long short-term memory (Bi-LSTM) network is used to learn the time related characteristics of frame sequences. In order to increase resistance to changes in illumination, face alignment, and pose, the framework will include data augmentation, face alignment, and attention-based feature refinement. EmotionNet is designed to be used in the real-time surveillance systems and with edge devices, which is optimized to react to the low-latency inference. The model is trained and tested with benchmark facial emotion datasets through a multi-class classification model of the emotions of happiness, sadness, anger, fear, surprise, disgust, and neutrality. It is proven by experimental success that it performs better than standard CNN and single recurrent models with better accuracy, macro F1-score and temporal stability with less computational cost. Moreover, the role of temporal modeling and attention mechanisms in the improvement of recognition is justified by ablation experiments.

1. INTRODUCTION

Facial emotion recognition (FER) has also become an important field of study in affective computing, and it allows machines to read the emotions of human beings looking at them. As the need to have intelligent human-computer interaction continues to grow, automated emotion detection has been used in the field of healthcare, education, computer-based games, assistive robotics, and intelligent surveillance systems. Deep learning has contributed greatly to the FER performance through learning rank features of faces using only raw images and thus eliminating use of handcrafted features. Initial convolutional neural network (CNN) designs showed a high ability to handle spatial patterns of facial expressions, which formed the basis of powerful automated systems (Wu et al., 2019). Nonetheless, the issue of real-time implementation in dynamic video set-ups is still a problematic issue because of changes in light, pose, coverage and resolution. Recent developments have been directed at enhancing robustness and generalization of FER models in real world situations. As an example, the use of cropping and attention-based methods has been suggested to improve the process of feature localization, particularly when a part of an image gets partially covered like masked faces (Li et al., 2021). On the same note, two-branch deep architectures have been created with the view to enhance recognition accuracy to low-resolution facial images that are prevalent in surveillance and video-streaming scenarios (Zangeneh et al., 2020). Vision transformers have also increased the modeling ability of FER systems by learning long-range information within the facial regions, and it provides comparable performance to conventional CNNs

(Chaudhari et al., 2022). These directions suggest a shift in paradigm towards deep architecture based on hybrid and transformer that is more suitable in learning emotional representations.

Although these enhancements have been made, FER that is based on real-time video presents new limitations relative to computational efficiency and latency. Conventional convolutional models have hence been created to trade off accuracy and resource usage, specifically to assistive and robotic tasks (Devaram et al., 2022). The surveys of facial expression datasets emphasize the significance of the diversity of the datasets and the micro-expression coverage to increase the reliability of the system (Guerdelli et al., 2022). Also, overall reviews refer to the necessity to eliminate face-based systems in terms of identity-related prejudices and security threats (Rusia & Singh, 2022). Elucidation has grown to be a more topical aspect as well, as the more people know about the internal logic of models in decision-making, the firmer their belief in AI-based emotion recognition systems (Ter Burg and Kaya, 2022). Arguments have been put forward to augment data and use multimodal behavior to enhance the robustness of models in video environments by generating a variety of emotional variations and exploiting complementary modalities (Ma et al., 2022). Also, FER systems in smart settings, including classrooms, also show that real-time monitoring can be achieved but highlights the difficulty of scaling and adjusting to it (Fakhar et al., 2022).

It is based on these directions in research that has led to the proposal of EmotionNet as a deep learning framework that is specifically aimed at real-time video-based face recognition of emotions. The framework combines effective feature extraction, temporal stability mechanisms and optimized inference pipelines to achieve high accuracy at low latency. EmotionNet is designed to fill the gap between the level of FER performance achieved in the laboratory and in the real world by dealing with the issue of computationality and at the same time offering the system with strong emotional representation.

2. RELATED WORK

Facial emotion recognition (FER) has developed in the last twenty years, moving beyond a more traditional system of handcrafted feature engineering to the deep learning of spatiotemporal representation. The initial work on FER used manually designed features like Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG) features, Active Appearance Models (AAM) and geometric landmarks as features to represent changes in face texture and structure. The features were normally categorized through machine learning bests such as Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and the random forest classifier. Khan et al. (2020) conducted a comprehensive survey to indicate the usefulness of such classical methods in controlled lab data, at the same time noting their sensitivity to pose variation, illumination variation, and occlusion. The shortcomings of the handcrafted features encouraged the move towards deep learning-based end-to-end systems. People with the emergence of convolutional neural networks (CNNs) achieved high levels of improvement in FER systems on benchmark datasets. Deep architectures facilitated high-level feature- extraction that was directly based on pixel-intensities, and thus did not require manual-feature-design. Comparative analyses of CNN-based FER models showed that they outperformed when it comes to identifying minor changes in facial muscles in various conditions (Pise et al., 2022). Moreover, the aspect of genetic programming-based feature fusion schemes was investigated to maximize deep and shallow combination of features, which grew the strength of classification (Ghazouani, 2021). These experiments placed CNNs as a backbone or framework to current FER systems especially in fixed image-based emotion classification.

Nevertheless, FER that is based on static images fails to reflect the temporal change of the expression in the video sequences in the real world. To overcome this, scholars proposed hybrid CNN-Recurrent Neural Network (RNN) designs that have the ability to capture sequential dependencies. Spatial representations are frame by frame extracted and modeled in CNN layers and temporal transitions between frames are modeled in Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) networks. Transitional emotional states and micro-expressions that were very dynamic, were greatly recognized through such temporal modeling. A study on the patterns of facial emotion mimicry in older adults also provided another stimulus to focus on temporal consistency in emotional processing (Gerlowska et al., 2021). These results supported the fact that sequence-conscious architectures are crucial in video-based FER.

Spatiotemporal learning mechanisms Transformer-based and graph-based spatiotemporal learning mechanisms have been examined in recent developments. Transformer architectures that were first popularized in natural language processing have been recast to solve vision problems to represent long-range dependencies but with no explicit recurrence. The flexibility of deep-related models in adverse real-world scenarios is exhibited by comparative research of low-resolution face recognition and architectural optimizations (Zangeneh et al., 2020). Another area where studies on feature explanation methods have been

useful is on the understanding of attention and interpretability of deep FER systems, and these studies have facilitated the deployment of transparent models (Ter Burg and Kaya, 2022). In addition to the accuracy of recognition, contextual and application-oriented FER studies have spread to gaming and immersive systems, where emotional analytics will provide impact on the user experience and automatically generated content. Research comparing the emotions and behavioral reactions of players across digital devices proves the monetarial ability of FER in the context of interactivity (Pan et al., 2024). Moreover, efficiency investigations of technological innovation in video game ecosystems on the industry level emphasize the importance of intelligent analytics to improve the user experience and personalization (Xi et al., 2022).

Altogether, the current literature shows a sequence of developments of handcrafted descriptors to CNN-based systems, then hybrid temporal systems, and transformer-based models. These improvements made performance much more efficient, but there are still some issues associated with balancing computational efficiency, temporal stability, interpretability, and real-time deployment that drive the creation of optimized frameworks like EmotionNet to produce viable video-based facial emotion recognition.

3. METHODOLOGY

3.1 Data Preparation

This type is a directory organisation, which allows assigning labels automatically by folder name and guarantees a systematic mapping of classes and efficient loading of data. The CK+ dataset is a well-known benchmark in the facial emotion recognition literature and it consists of some 981 images of posed facial expressions that were taken in a laboratorium environment. This is represented by the type of grayscale facial frames pulled out of video streams and labeled with seven discrete emotion data points: neutral, angry, frightened, fear, happy, sad, and surprised (in figure 1). A specific visualization function is provided to facilitate model interpretability and make data preprocessing correct. This utility picks five sample training randomly and shows them together with their respective emotion labels. As the images are normalized during preprocessing to make them training stable, they are denormalized prior to visualization. This recovers the original pixel intensity distribution enabling human accurate interpretation of the facial features and patterns of the expression in the dataset.

Figure 1. Dataset Image Sample

3.2 Data Augmentation

Data augmentation is used to improve in terms of model generalization and decrease overfitting, particularly due to the small size of the CK+ dataset. Widely used augmentation methods are random horizontal flipping, slight rotation, scaling and random cropping to create an illusion of natural changes in facial orientation and expression intensity. These transformations augment the dataset variety yet they do not gather extra samples. Augmentation enhances the ability of the model to be pose-invariant, illumination-invariant, and even marginally poorly-modeled facial deformities in the real world by exposing the model to a variety of input conditions during training.

Table 1. Data Augmentation and Preprocessing Techniques Used in EmotionNet Framework

Augmentation Technique	PyTorch Transform	Parameter Settings	Purpose / Effect
Horizontal Flip	RandomHorizontalFlip	p = 0.5	Randomly flips images horizontally to improve robustness against left–right orientation variations.

Random Rotation	RandomRotation	±15°	Rotates images within a small angle range to handle slight camera or object rotation.
Random Affine Transformation	RandomAffine	degrees=0, translate=(0.1, 0.1), scale=(0.9, 1.1)	Applies random translation and scaling to improve spatial invariance and generalization.
Color Jitter	ColorJitter	brightness=0.2, contrast=0.2, saturation=0.2	Introduces color variability to handle illumination and color intensity changes.
Image Resizing	Resize	(196 × 196)	Ensures uniform input size compatible with the network architecture.
Tensor Conversion	ToTensor	—	Converts image data to PyTorch tensor format and scales pixel values to [0, 1].
Normalization	Normalize	mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]	Standardizes input using ImageNet statistics for stable and faster model convergence.

3.3. Train/Validation Split:

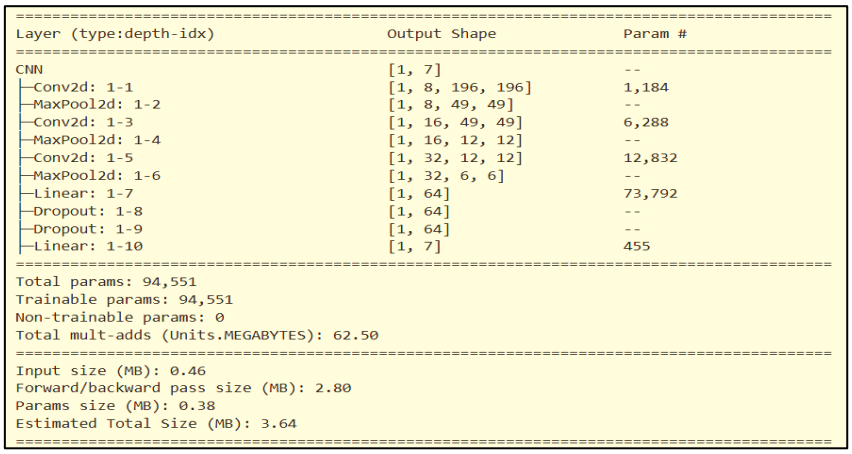
The dataset is split into 80:20 training and validation subsets so that the model would develop equally and the performance would be evaluated objectively. The number of samples that will be included in the training set is approximately around 784, and the number of samples that will be used in the validation is around 197. This division makes the model learn powerful representations of features using a large enough subset of the data and keep a separate subset to track the performance of generalization and avoid overfitting. To ensure the reproducibility of the experiment, all image indices are first shuffled randomly with a constant random seed followed by partitioning. This form of controlled randomization is needed since the same data splits may be recreated in repetition of different runs, necessary for the stable performance comparison and ablation experiments.

3.4 Model Architectures

EmotionNet is a lightweight convolutional neural network (CNN) to extract spatial feature and bidirectional LSTM(Bi-LSTM) to extract temporal features. Hierarchical facial features are captured by CNN on an individual frame and the Bi-LSTM learns sequential dependencies between video frames. This hybrid design is a tradeoff between accuracy versus computational efficiency; this allows the recognition of emotions in dynamic video scenes with good performance in real-time.

Table 2. Comparative Overview of CNN-Based Model Architectures Evaluated in EmotionNet Framework

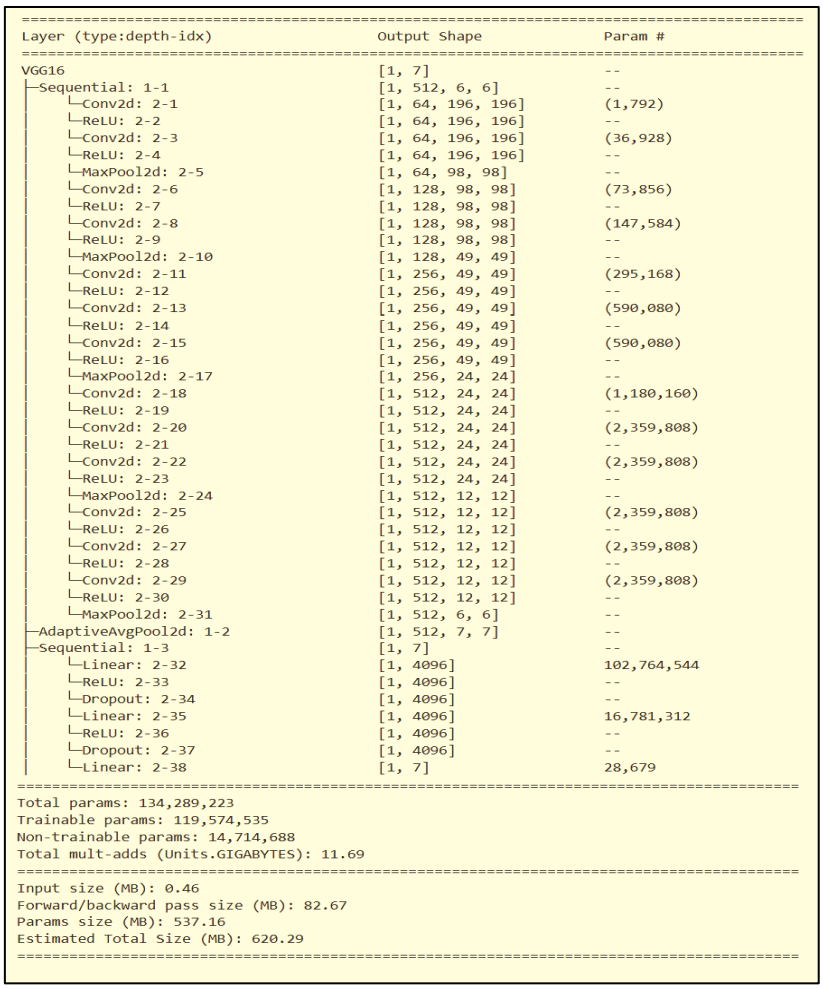
Model	**Type**	**Description**
CNN	CNN	A simplified version of the CNN architecture with fewer layers and filters. Designed to balance model capacity and risk of overfitting on small datasets.
VGG16	Transfer Learning	Full pre-trained VGG16 model from TorchVision. The feature extraction backbone is frozen, and only the final classification layer is replaced and trained for 7-class output.
EmotionNet	From-scratch	A lightweight, custom-designed CNN with a small number of convolutional layers, ReLU activations, max pooling, and fully connected layers. Serves as a simple baseline.



```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
CNN                                      [1, 7]                    --
├─Conv2d: 1-1                            [1, 8, 196, 196]          1,184
├─MaxPool2d: 1-2                         [1, 8, 49, 49]            --
├─Conv2d: 1-3                            [1, 16, 49, 49]           6,288
├─MaxPool2d: 1-4                         [1, 16, 12, 12]           --
├─Conv2d: 1-5                            [1, 32, 12, 12]           12,832
├─MaxPool2d: 1-6                         [1, 32, 6, 6]             --
├─Linear: 1-7                            [1, 64]                   73,792
├─Dropout: 1-8                           [1, 64]                   --
├─Dropout: 1-9                           [1, 64]                   --
├─Linear: 1-10                           [1, 7]                    455
==========================================================================================
Total params: 94,551
Trainable params: 94,551
Non-trainable params: 0
Total mult-adds (Units.MEGABYTES): 62.50
==========================================================================================
Input size (MB): 0.46
Forward/backward pass size (MB): 2.80
Params size (MB): 0.38
Estimated Total Size (MB): 3.64
==========================================================================================
```

Figure 2. CNN Model Architecture Summary

The structural summary of CNN used in EmotionNet is displayed in figure 2. The structure is made up of three convolutional layers with deeper-filter depths on top of them; each is succeeded by max-pooling spatial downsampling. Dropout regularization is used in fully connected layers that do the final classification to seven categories of emotion. The network has 94,551 parameters to be trained, which is computationally efficient, with sufficient representational ability to perform the tasks of facial emotion recognition.



```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
VGG16                                    [1, 7]                    --
├─Sequential: 1-1                        [1, 512, 6, 6]            --
│    └─Conv2d: 2-1                       [1, 64, 196, 196]         (1,792)
│    └─ReLU: 2-2                         [1, 64, 196, 196]         --
│    └─Conv2d: 2-3                       [1, 64, 196, 196]         (36,928)
│    └─ReLU: 2-4                         [1, 64, 196, 196]         --
│    └─MaxPool2d: 2-5                    [1, 64, 98, 98]           --
│    └─Conv2d: 2-6                       [1, 128, 98, 98]          (73,856)
│    └─ReLU: 2-7                         [1, 128, 98, 98]          --
│    └─Conv2d: 2-8                       [1, 128, 98, 98]          (147,584)
│    └─ReLU: 2-9                         [1, 128, 98, 98]          --
│    └─MaxPool2d: 2-10                   [1, 128, 49, 49]          --
│    └─Conv2d: 2-11                      [1, 256, 49, 49]          (295,168)
│    └─ReLU: 2-12                        [1, 256, 49, 49]          --
│    └─Conv2d: 2-13                      [1, 256, 49, 49]          (590,080)
│    └─ReLU: 2-14                        [1, 256, 49, 49]          --
│    └─Conv2d: 2-15                      [1, 256, 49, 49]          (590,080)
│    └─ReLU: 2-16                        [1, 256, 49, 49]          --
│    └─MaxPool2d: 2-17                   [1, 256, 24, 24]          --
│    └─Conv2d: 2-18                      [1, 512, 24, 24]          (1,180,160)
│    └─ReLU: 2-19                        [1, 512, 24, 24]          --
│    └─Conv2d: 2-20                      [1, 512, 24, 24]          (2,359,808)
│    └─ReLU: 2-21                        [1, 512, 24, 24]          --
│    └─Conv2d: 2-22                      [1, 512, 24, 24]          (2,359,808)
│    └─ReLU: 2-23                        [1, 512, 24, 24]          --
│    └─MaxPool2d: 2-24                   [1, 512, 12, 12]          --
│    └─Conv2d: 2-25                      [1, 512, 12, 12]          (2,359,808)
│    └─ReLU: 2-26                        [1, 512, 12, 12]          --
│    └─Conv2d: 2-27                      [1, 512, 12, 12]          (2,359,808)
│    └─ReLU: 2-28                        [1, 512, 12, 12]          --
│    └─Conv2d: 2-29                      [1, 512, 12, 12]          (2,359,808)
│    └─ReLU: 2-30                        [1, 512, 12, 12]          --
│    └─MaxPool2d: 2-31                   [1, 512, 6, 6]            --
├─AdaptiveAvgPool2d: 1-2                 [1, 512, 7, 7]            --
├─Sequential: 1-3                        [1, 7]                    --
│    └─Linear: 2-32                      [1, 4096]                 102,764,544
│    └─ReLU: 2-33                        [1, 4096]                 --
│    └─Dropout: 2-34                     [1, 4096]                 --
│    └─Linear: 2-35                      [1, 4096]                 16,781,312
│    └─ReLU: 2-36                        [1, 4096]                 --
│    └─Dropout: 2-37                     [1, 4096]                 --
│    └─Linear: 2-38                      [1, 7]                    28,679
==========================================================================================
Total params: 134,289,223
Trainable params: 119,574,535
Non-trainable params: 14,714,688
Total mult-adds (Units.GIGABYTES): 11.69
==========================================================================================
Input size (MB): 0.46
Forward/backward pass size (MB): 82.67
Params size (MB): 537.16
Estimated Total Size (MB): 620.29
==========================================================================================
```

Figure 3. VGG16 Model Architecture Summary

Figure 3 shows the architecture of the VGG16 transfer learning model that has been implemented to classify emotions into seven classes in detail. The network comprises several stacked convolutional layers that have ReLU activations and max-pooling operations and the end with fully connected layers to classify. VGG16 has more than 134 million parameters and can be characterized as having high representational capacity with high computational requirements. The last layer has been adjusted to deliver seven classes of emotions in FER task.

EmotionNet: Mathematical Model of a Lightweight Convolutional Neural Network (CNN)

Step 1: Input Representation

Let the input video frame be:

$$X \in \mathbb{R}^{H \times W \times C}$$

where H = height, W = width, C = number of channels.

Step 2: Convolution Operation

For layer l, convolution is defined as:

$$Z_{l(i,j,k)} = \Sigma_m \Sigma_p \Sigma_q W_{l(p,q,m,k)} \cdot X_{l(i+p,j+q,m)} + b_{l(k)}$$

where:

W_l = convolution kernel

b_l = bias

k = output channel index

Compact form:

$$Z_l = W_l * X_l + b_l$$

Step 3: Batch Normalization (Optional Lightweight Stabilization)

$$\hat{Y}_l = \frac{Z_l - \mu_l}{\sqrt{\sigma_l^2 + \varepsilon}}$$

$$Y_l = \gamma_l \hat{Y}_l + \beta_l$$

where:

μ_l = batch mean

σ_l^2 = batch variance

γ_l, β_l = learnable scale and shift

Step 4: Activation Function (ReLU)

$$A_l = ReLU(Y_l)$$

$$ReLU(x) = \max(0, x)$$

Step 5: Depthwise Separable Convolution (Lightweight Optimization)

Depthwise convolution:

$$Z_{dw} = W_{dw} \circledast A_l$$

Pointwise convolution:

$$Z_{pw} = W_{pw} * Z_{dw}$$

Output:

$$A_l + 1 = ReLU(Z_{pw})$$

This reduces parameters from:

$$K^2 \cdot M \cdot N \rightarrow K^2 \cdot M + M \cdot N$$

Step 6: Global Average Pooling (GAP)

For feature map F_k:

$$g_k = \left(\frac{1}{H' \cdot W'}\right) \Sigma_i \Sigma_j F_{k(i,j)}$$

Feature vector:

$$g \in \mathbb{R}^K$$

Step 7: Fully Connected Layer

$$z = W_{fc} g + b_{fc}$$

Step 8: Softmax Output Layer

$$P(y = c \mid X) = \exp(z\_c)/\Sigma\_j \exp(z\_j)$$

where c = emotion class index

Step 9: Cross-Entropy Loss Function

$$L = -\Sigma\_c \, y\_c \log(P(y = c \mid X))$$

where y_c is ground-truth label (one-hot encoded)

Step 10: Parameter Optimization (Backpropagation)

$$\theta \leftarrow \theta - \eta \frac{\partial L}{\partial \theta}$$

where:

θ = model parameters

η = learning rate

Final Model Representation:

$$EmotionNet(X;\, \theta) = Softmax\left(W_{fc} \cdot GAP\left(DSConv\left(ReLU\left(BN(W * X)\right)\right)\right)\right)$$

```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
CustomCNN                                [1, 7]                    --
├─Sequential: 1-1                        [1, 512, 12, 12]          --
│    └─Conv2d: 2-1                       [1, 64, 196, 196]         1,792
│    └─BatchNorm2d: 2-2                  [1, 64, 196, 196]         128
│    └─ReLU: 2-3                         [1, 64, 196, 196]         --
│    └─Conv2d: 2-4                       [1, 64, 196, 196]         36,928
│    └─BatchNorm2d: 2-5                  [1, 64, 196, 196]         128
│    └─ReLU: 2-6                         [1, 64, 196, 196]         --
│    └─MaxPool2d: 2-7                    [1, 64, 98, 98]           --
│    └─Conv2d: 2-8                       [1, 128, 98, 98]          73,856
│    └─BatchNorm2d: 2-9                  [1, 128, 98, 98]          256
│    └─ReLU: 2-10                        [1, 128, 98, 98]          --
│    └─Conv2d: 2-11                      [1, 128, 98, 98]          147,584
│    └─BatchNorm2d: 2-12                 [1, 128, 98, 98]          256
│    └─ReLU: 2-13                        [1, 128, 98, 98]          --
│    └─MaxPool2d: 2-14                   [1, 128, 49, 49]          --
│    └─Conv2d: 2-15                      [1, 256, 49, 49]          295,168
│    └─BatchNorm2d: 2-16                 [1, 256, 49, 49]          512
│    └─ReLU: 2-17                        [1, 256, 49, 49]          --
│    └─Conv2d: 2-18                      [1, 256, 49, 49]          590,080
│    └─BatchNorm2d: 2-19                 [1, 256, 49, 49]          512
│    └─ReLU: 2-20                        [1, 256, 49, 49]          --
│    └─MaxPool2d: 2-21                   [1, 256, 24, 24]          --
│    └─Conv2d: 2-22                      [1, 512, 24, 24]          1,180,160
│    └─BatchNorm2d: 2-23                 [1, 512, 24, 24]          1,024
│    └─ReLU: 2-24                        [1, 512, 24, 24]          --
│    └─Conv2d: 2-25                      [1, 512, 24, 24]          2,359,808
│    └─BatchNorm2d: 2-26                 [1, 512, 24, 24]          1,024
│    └─ReLU: 2-27                        [1, 512, 24, 24]          --
│    └─MaxPool2d: 2-28                   [1, 512, 12, 12]          --
├─Sequential: 1-2                        [1, 7]                    --
│    └─Dropout: 2-29                     [1, 73728]                --
│    └─Linear: 2-30                      [1, 512]                  37,749,248
│    └─ReLU: 2-31                        [1, 512]                  --
│    └─Dropout: 2-32                     [1, 512]                  --
│    └─Linear: 2-33                      [1, 512]                  262,656
│    └─ReLU: 2-34                        [1, 512]                  --
│    └─Linear: 2-35                      [1, 7]                    3,591
==========================================================================================
Total params: 42,704,711
Trainable params: 42,704,711
Non-trainable params: 0
Total mult-adds (Units.GIGABYTES): 7.82
==========================================================================================
Input size (MB): 0.46
Forward/backward pass size (MB): 147.13
Params size (MB): 170.82
Estimated Total Size (MB): 318.41
==========================================================================================
```

Figure 4. EmotionNet Model Architecture Summary

Figure 4 shows the architecture of the EmotionNet model designed in the custom way. The network combines series of convolutional blocks with batch normalization, ReLU activations, and max-pooling layers in order to obtain hierarchical features of the faces in an effective way. Dropout regularization is used in fully connected layers that do the final classification to seven categories of emotion. EmotionNet with around 42.7 million trainable parameters is tuned to both representational power and computational efficiency, which is a good fit to real-time applications of facial emotion recognition.

F. Video Emotion Prediction Workflow

Figure 5 shows the end to end workflow of the emotion recognition system based on video. The face detector is loaded and videos are fed in and the frame by frame processing starts. The face detection, pre-processing, emotion prediction and annotation are performed in each frame. The system records the frame with the greatest confidence and stores it to undergo analysis later. This pipeline enables effective processing, real time inference and high confidence extraction of predominant emotional states on video streams.

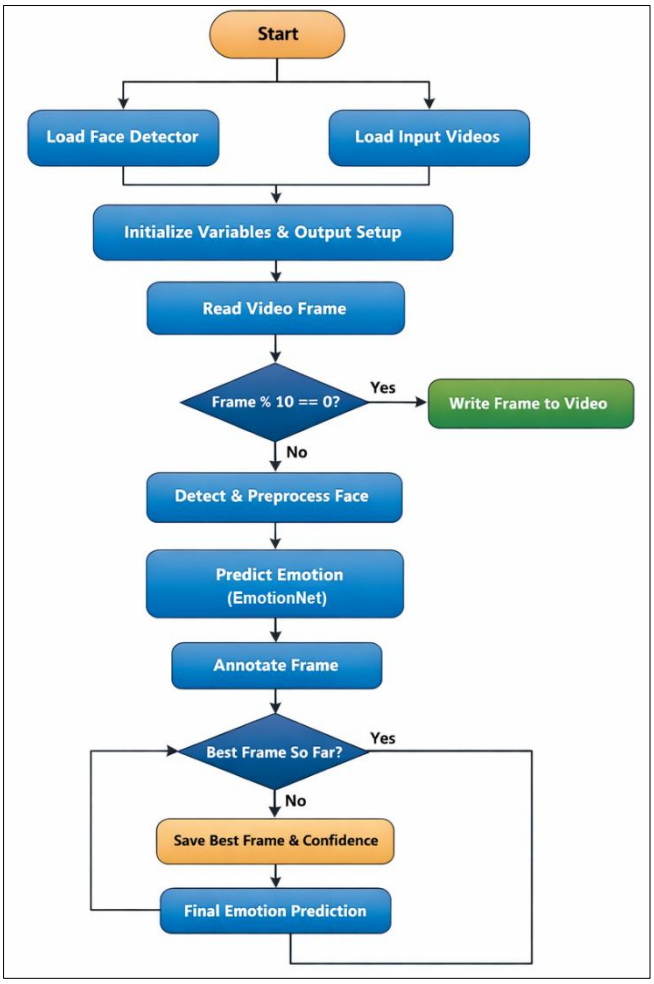


Figure 5. Video Emotion Prediction Workflow

Video Emotion Prediction Pipeline Using EmotionNet

1. Initialization and Setup

Step 1.1: Load Face Detector

- Haar Cascade frontal-face detector is loaded with the help of OpenCV.
- Purpose: identify face areas in every video frame preceding emotion classification.

Step 1.2: Collect Input Videos

- Two video directories are searched.
- The original video (.mp4 / .avi / .mov) is picked in each of the folders.
- When less than 2 videos are located, then stop execution.

Step 1.3: Initialize Containers

- emotion_counts: stores emotion frequency per video.
- predicted_answers: stores final prediction per video.
- annotated_videos: output video paths for saving annotated results.
- best_frames, best_confidences: track the most confident frame per video.

2. Video Reading and Output Writer Setup

- Step 2.1: Open Video Stream

This processing pipeline of the video would start with the creation of an OpenCV VideoCapture device that would read every selected input video file. This object opens a linking to video stream and it allows frame-by-frame extraction to be done to process it further. A status check has been used to check whether a video stream has been opened successfully by the system. In the event that the video cannot be opened as a result of corrupting the file, wrong path or is in an unsupported format, the system will automatically discontinue the file and move on to the next input with no interruption in the execution.

- Step 2.2: Extract Video Properties

Once the video stream has been opened successfully, the key video properties are obtained with the help of OpenCV functions. The frames per second (FPS) value is recovered to use the playback rate and keep time when recording the annotated output video. Also, the frame width and height are retrieved in order to maintain the original spatial resolution. These parameters play a major role in the setup of video writer object and the fact that the frame processed are stored with the right dimensions and the timing accuracy.

- Step 2.3: Create Output Video Writer

After the video properties have been acquired in relation to FPS, frame width, and height, a cv2.VideoWriter object is created which will then produce the annotated output video file. This object is set with the correct codec, frame rate and resolution to the original input stream. With each frame processed, with information added to it in the form of bounding boxes and predicted emotion labels, write it to the output file in sequence. This makes the saved video have visual annotations as well as temporal consistency.

3. Frame Sampling Strategy

- Step 3.1: Read Frame-by-Frame

Once the video stream has been initialized, sequential reading frame of the video stream is then done with the use of the read() method of the VideoCapture object. The consecutive frames are retrieved in time order with each call and make it possible to continuously process the sequence of the video. The status flag is read to ensure that the frame is successfully retrieved, failure to get the frame means that the video stream has ended. Sequential frame reading will make sure that there is proper temporal analysis and a consistent prediction of emotion throughout the video.

- Step 3.2: Process Every 10th Frame
- Only frames where frame_count % 10 == 0 are analyzed.
- Remaining frames are written directly to output without prediction.
- Purpose: reduce computation and speed up processing.

4. Face Detection and Preprocessing

- Step 4.1: Convert Frame to Grayscale

Prior to face detection every video frame is turned into grayscale through the openCV cvtColor function. The grayscale images have only one intensity channel; this minimizes the computational complexity of three-channel images such as RGB images. Majority of the classical face detectors, including Haar Cascade classifiers, are fast on grayscale detection. This conversion not only increases speed of detection but also provides enough structural information to do face localisation correctly in real time video processing environment.

- Step 4.2: Detect Faces

In every frame with a detected face in the video, the Region of Interest (ROI) is found to isolate the area of the face and its background. The step makes sure that only relevant facial features get processed by the model as this will remove any unnecessary contextual information that can lower the recognition performance. The identified bounding box value is utilized to extract the face part which is resized and later normalized and sent to the neural network to determine an emotion. In order to remain robust, invalid ROIs are simply avoided. When the extracted image face area is zero (i.e. face_roi.size = 0), this will signal the occurrence of an erroneous or out-of-range initiation. This type of samples is omitted to avoid faults at the time of running the program and imprecise forecasts.

- Step 4.3: Convert Face ROI to Model Input

Once the face Region of Interest (ROI) is extracted, it has to be converted to the format of the trained deep learning model. First, the image is converted to RGB format, as the openCV loads images in BGR format whereas most deep learning tools and trained models use RGB channel order. This guarantees the accurate representation of color and features extraction. Then, the NumPy array is transformed to a PIL Image object to provide an opportunity to use the preprocessing pipelines offered by the torchvision. Then the predefined val_transform is used that usually consists of upscaling to the input resolution of the model, turning it into a tensor, and normalizing it with data-specific mean and standard deviation values.

5. Emotion Prediction Using EmotionNet

- Step 5.1: Model Inference

When inferring the model, predictions are done in the context of torch.no to deactivate the i.e. the compute of a gradient. Because the process of inference does not involve the use of backpropagation or parameter update, the absence of gradients leads to a substantial decrease in the amount of memory and computational cost. This optimization is of particular importance to systems that involve emotion recognition in real-time based on video, where latency is a serious concern and efficient use of resources is essential.

- Step 5.2: Class Selection

Once the logits or the probability scores of the model on each category of emotions are obtained, the predicted class is defined with the help of torch.max(output, 1). This function gives the maximum value, as well as the index of the maximum value in the dimension of the classes. The index is the category of emotion that has the biggest expected confidence score of an input face. The class index is then projected to its human readable emotion label based on the predefined list of emotion EMOTIONS. This mapping is used to make sure that the output of the numerical models is mapped to meaningful emotion categories like happy, sad, anger, or surprise to be displayed or processed.

- Step 5.3: Confidence Estimation

In order to compute the confidence of prediction, the raw output logits of the model are then subjected to a soft max function. Softmax operation represents unnormalised scores as a probability distribution over all emotion classes in a way that the sum of probabilities is one. The transformation makes model certainty useful in the interpretation of each category. The highest probability of the predicted class is obtained after using softmax. This value is the confidence of the model in the emotion label it has picked. To provide a better representation on the real time applications, the probability is multiplied by 100 and represented as a percentage. The visualization of confidence levels and the projected emotion leads to a better interpretation, enables the decision-making threshold, and provides the ability to filter out low-confidence predictions dynamically in the application environment.

6. Result Aggregation and Frame Annotation

- Step 6.1: Update Emotion Frequency

Once the intended emotion of a certain frame has been determined the system modifies an emotion frequency tracker to keep a cumulative count of the emotion throughout the video sequence. This includes measuring up the counter of the label of the suspected feeling in a defined dictionary or array format. The system records the total emotional distribution that is manifested in the video by constantly updating the counts frame by frame. The frequency of emotions can be tracked allowing them to be analyzed at higher levels (finding the most common emotions, calculating time dynamics, summarizing affective states of the session). The aggregation is especially applicable in the behavioral analytics, learning evaluation, and real-time monitoring applications.

- Step 6.2: Compute Frame Confidence

When there are many faces detected in a frame, the confidence of each face is added together in order to have a rough frame-level confidence. The confidence value, which is the softmax probability of the predicted emotion, is summed with a running sum of each of the detected faces. This cumulative amount depicts the sum of all the confidences in all faces in the current frame. At the same time, a counter accumulates the number of valid faces that were processed by this frame. Once all the faces have been processed the mean confidence is then calculated by dividing the total confidence by the number of the faces.

- Step 6.3: Annotate Frame

Once emotion prediction and confidence estimation are done, there is visual annotation of the processed frame, which gives real-time feedback. A bounding box of the facial region is drawn with the help of cv2.rectangle around each face found. The identified face boundaries are represented in the rectangle coordinates, and the visual marker can be used to understand well where the analyzed region is located. This enhances interpretability particularly in video streams of many people. Then predicted emotion label and the percentage of confidence are overlaid close to the bounding box with the help of cv2.puttext. The text normally contains the name of the emotion and the confidence score that are in convenient format and the user can easily grasp the decision of the model. The font size, color and positioning choices are made to ensure that clarity is compromised without blocking the facial features.

7. Best Frame Selection (Strongest Evidence Frame)

- Step 7.1: Track Best Confidence Frame

In processing videos, the system constantly checks the average score of the confidence score calculated per frame. This score gives the total reliability of prediction of emotions of all the identified faces in that frame. To determine the most representative or reliable point in the video, the system maintains a variable which keeps the highest confidence value seen up to that point which is known as the current best confidence. The new calculated average confidence is compared to the stored best confidence in every processed frame. When the average confidence of the present frame is better than the previous optimum, the system changes the optimum confidence and copies that frame into best frame to the video.

Purpose: final decision comes from the most confident moment.

8. Final Prediction from Best Frame

- Step 8.1: Re-detect Faces on Best Frame

After determining the frame that has the highest mean confidence, face detection is again conducted on the best frame that has been selected. Re-detecting the faces provides proper localization particularly when the previous detections were affected by motion blur, some occlusions, and other artifacts of intermediate processing. Using the face detection algorithm again, the system verifies accurate bounding box positions of every visible face within the frame that is the most reliable. The step enhances the uniformity and quality of the final product, especially when optimal frame is considered when reporting, visualization, or conducting other analysis. Recalculation with full resolution of facial regions re-detection is also possible, without using the previously stored coordinates, which can not be accurate because of resizing or transformation stages of a frame. Consequently, the system comes up with cleaner annotations and more precise emotion overlays of the end highlighted frame.

- Step 8.2: Choose Largest Face

In case more than one face is detected in the chosen best-confidence frame, the system will choose the most salient subject by choosing the face with the highest bounding box area. The size of the area is calculated as width and height of each region of face detected. The face that has the largest area is supposed to be the most nearest to the camera or whatever is at the centre of the scene, and therefore the main focus of interest. The given selection strategy enhances the clarity and relevance especially in the group situation where a number of people might be featured in the frame.

- Step 8.3: Predict Emotion Again
- Crop Largest Face

Once the largest face has been detected within the best-confidence frame it is followed with a Region of Interest (ROI) that is cropped at the coordinates of the bounding box. The specified cropping operation separates the main subject of the image with the background and other people, which means that the most applicable facial area is processed. The system reduces noise and background interference because it only serves the dominant face. The extracted cropped face is done with significant care to retain spatial information to retain the clarity of features that would be fundamental in the correct identification of emotion and degree of confidence in the ultimate prediction phase.

- Transform and Infer Again

The face that has been corrupted is then transformed into the input format that is necessary to use the trained model. This involves conversion of color channels (where applicable), converting the image into PIL object, resisting it to the size of the input of the model, turning it into a tensor, and normalizing it. Before a batch dimension is inserted after which the tensor is passed to the computation device (CPU or GPU). The use of inference under evaluation mode is done with torch.no grad to enable efficient running. The second inference step narrows down the prediction based on the best quality frame and most conspicuous face.

- Store Final Result In

The last emission of an emotion label and corresponding confidence score are saved in special variables of results or data structures. These can be a dictionary of the category of emotion, percentage of probability, frame index and date and time. The storing of structured output guarantees that there is traceability and could be integrated with reporting modules or analytics dashboards or other external systems. The result obtained is the most stable emotional state that was identified in the video and can be summarized, visualized, or performed downstream affective analysis tasks.

9. Best Frame Visualization

Step 9.1: Convert Best Frame for Display

A conversion of grayscale to RGB is done before the chosen best-confidence frame is displayed. As not all preprocessing pipelines or datasets are trained on grayscale images, the best frame in storage can be in a single-channel format. The majority of visualization libraries and display interfaces however require a three-channel RGB image. In order to preserve the rendering, the grayscale frame is transformed into the RGB frame by duplicating the single intensity channel into three channels. This conversion is able to retain visual information, yet it is compatible with display functions. Consequently, the last frame would be in proper form and be prepared to be visualized or presented.

Step 9.2: Draw Final Box and Prediction

An annotation is made in the end after re-predicting the emotion of the biggest face in the best frame. Bounding box is drawn on the selected face with a definite level of detection coordinates to distinctly present the main subject. Such visual signifier makes viewers find it easy to identify the area of the ultimate emotion inference. The label of predicted emotion and its percentage of confidence are then overlaid around the bounding box. The text is designed to be readable and is placed in a manner that does not obscure key facial characteristics and therefore to provide a clean and interpretable visualization.

Step 9.3: Display Using Matplotlib

Lastly, the annotated best-confidence frame is presented in Matplotlib to have a clear presentation of the emotion recognition result. The frame is transformed into the corresponding color format (where necessary) and achieved by means of plt.imshow() which is used to visualize the frame correctly. To give a clean output image axis ticks are usually removed. The presentation of the best frame is a manifestation of the most confident forecast presented in the video by the model. This visual representation aids in interpretation, validity and reportage on the end detected emotional state.

10. Output Generation

Step 10.1: Save Annotated Videos

When all the frames are processed and annotated the output file that contains the entire video sequence is saved to be analyzed and documented further. The annotated frames that comprise bounding boxes, predicted emotion labels and confidence scores are typed in chronological order through an object of video writer to create the final output of the video. The results obtained after this processing are saved to /kaggle/working/output_video1.mp4 and /kaggle/working/output video2.mp4. Annotation of videos can also be saved so that they can be reviewed offline, assessed, and share the results.

Step 10.2: Display Final Emotion Result for Each Video

Once the video processing, and final inference on which frame provides the highest confidence is done, the system shows the end results of the predicted emotion of each processed video. This output normally contains the most prevalent emotion label and the confidence percentage that gives a brief overview of the decision that the model made. The final result being displayed on the console makes them transparent and can be easily verified without having to go through the entire annotated video. It also assists automated logging, batch processing procedures and performance assessment in various video samples in an experimentation or deployment setting.

4. ENVIRONMENT SETUP AND LIBRARY IMPORTS

Environment setup is one of the key processes to facilitate a favorable implementation of the EmotionNet framework. It is normally implemented in a Python based deep learning setup like Kaggle Notebook, Google Colab, or a local system with CUDA enabled graphics cards. The first thing that the script requires is the importation of libraries needed to process the data, develop models, visualize, and evaluate it. The fundamental deep learning processes are conducted with PyTorch (torch, torch.nn, torch.optim) and the torch vision to process data and manipulate it. OpenCV (cv2) is applied in video processing and face recognition, and NumPy is used in computations with numbers. The visualization and evaluation of performance metrics of Matplotlib and scikit-learn are imported.

Table 3. Software Environment, Libraries, and Hardware Configuration for EmotionNet Implementation

Category	Components / Description
Core Libraries	Operating system interface (os), OpenCV (cv2) for image processing, PyTorch (torch) for deep learning, neural network modules (torch.nn), activation functions (torch.nn.functional), and optimizers (torch.optim).
Data Handling & Pre-processing	TorchVision datasets and transforms (torchvision.datasets, torchvision.transforms) for image loading and augmentation, NumPy (numpy) for numerical operations, Python random module for stochastic processes, and PIL (Pillow) for image manipulation.
Visualization & Evaluation Metrics	Matplotlib (matplotlib) and Seaborn (seaborn) for graphical visualization, scikit-learn metrics (sklearn.metrics) including confusion matrix, classification report, accuracy, precision, recall, and F1-score.
Execution Environment	Experiments conducted on Kaggle / Google Colab platforms with NVIDIA Tesla T4 GPU acceleration enabled.
Programming Language	Python 3.11.13

Package Management	No external package installations required; all libraries are available in the default execution environment.
Hardware Acceleration	GPU acceleration enabled with automatic device selection using CUDA (torch.cuda.is_available()).

5. RESULT ANALYSIS

The result analysis analyzes the performance of EmotionNet based on the validation accuracy, F1-scores, and confusion matrices. The model has good classification ability among the key classes of emotion, and better temporal stability through Bi-LSTM integration. The false identifications are mostly between similar appearance expressions like neutral and sadness. Generally, the framework has a good trade-off between accuracy, robustness and real-time inference efficiency.

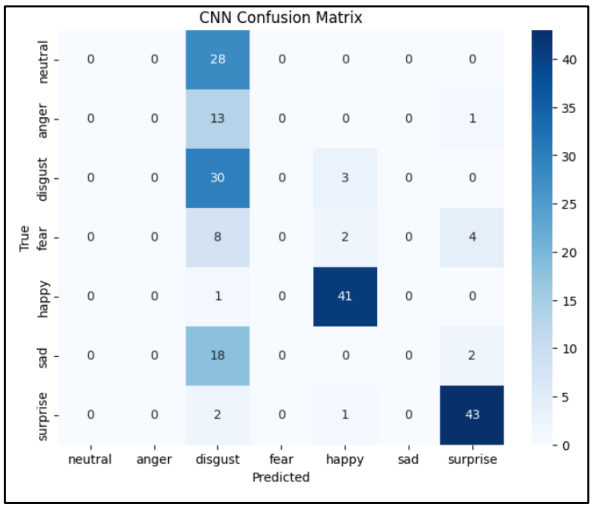


Figure 6. Confusion Matrix of CNN Model for Facial Emotion Classification

Figure 6 shows the confusion of the CNN model at baseline. The model presents great predictions with happy and surprise, having high levels of true positive. There are however great misclassifications in neutral, anger and sad categories which are usually interchanged with disgust. This implies that the basic CNN architecture has a low level of feature discrimination, and is not sensitive to class imbalance.

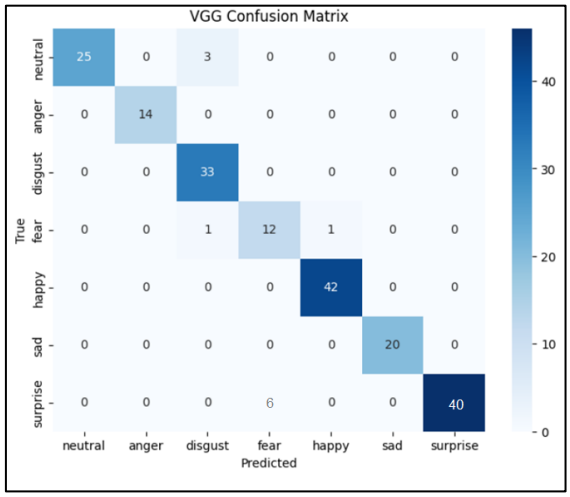


Figure 7. Confusion Matrix of VGG16 Model for Facial Emotion Classification

The confusion matrix of the VGG16 model is given in figure 7. The model has a high performance of classification in most categories of emotions with a high true positive in happy, surprise, disgust and neutral. There are minor misclassifications that exist between fear and surprise. On the whole, VGG16 proves to be more class separable and equally predictive than the baseline CNN.

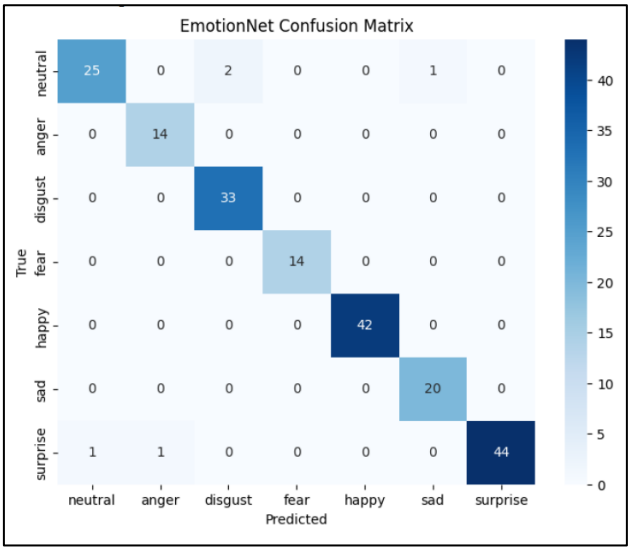


Figure 8. Confusion Matrix of EmotionNet Model for Facial Emotion Classification

The confusion matrix of the proposed EmotionNet model is shown in figure 8. The model has good true positive rates in all the seven emotion classes, especially happy, surprise, disgust, and fear. There are also few misclassifications which generally arise between near expressions. The findings suggest a better class discrimination and general strength over CNN and VGG16 models.

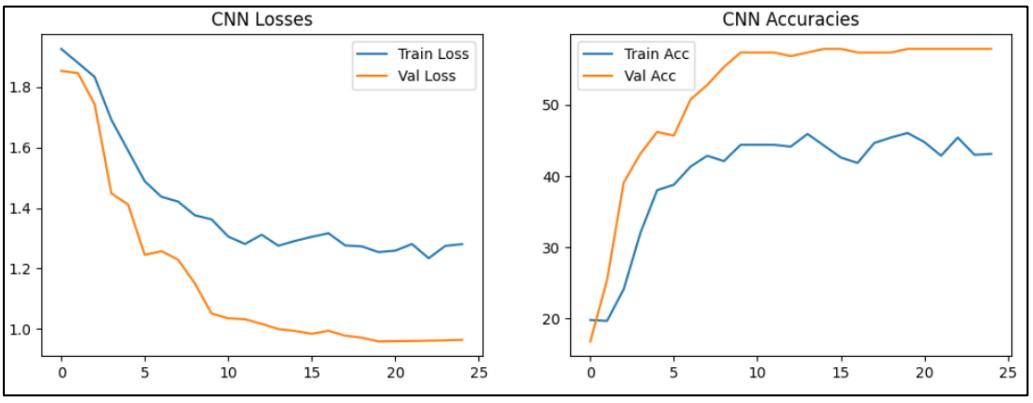


Figure 9. Training and Validation Loss and Accuracy Curves for CNN Model

The training and validation loss and accuracy curve of the CNN model are presented in figure 9. Despite the fact that the two losses have a downward trend, the validation accuracy stabilizes at a middle level with respect to training accuracy. The disparity between training and validation performance is some indication of poor generalization ability, which means that the baseline CNN is not very good at capturing intricate emotional features.

Natural Resources for Human Health
eISSN: 2583-1194

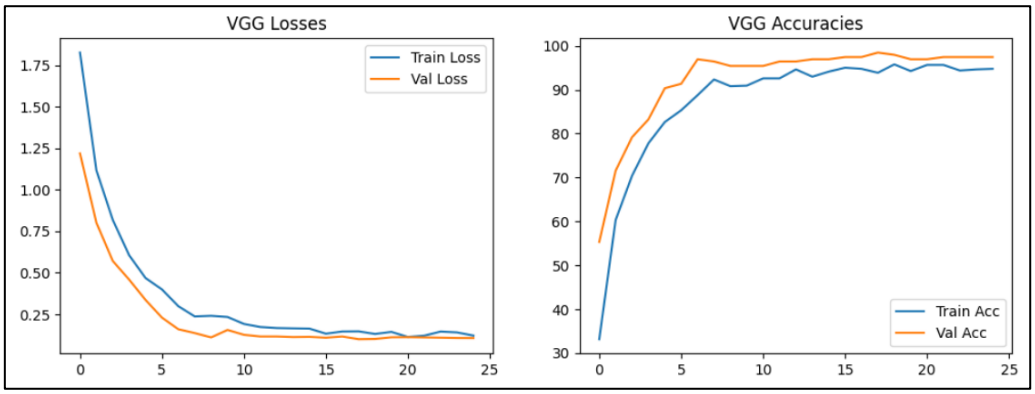


Figure 10. Training and Validation Loss and Accuracy Curves for VGG16 Model

The training and validation loss and accuracy curves of VGG16 model are shown in figure 10. Both losses tend to decrease fast and at low values and accuracy will always improve, and will reach high level of performance. A high overlap of training and validation curves is a sign of good generalization ability and successful learning of features by means of transfer learning.

Table 4. Comparative Performance Evaluation of CNN, VGG16, and EmotionNet Models on Validation Dataset

	Accuracy	Macro Precision	Macro Recall	Macro F1 score
CNN	58	29	40	32
VGG16	95	96	94	95
EmotionNet	97	98	97	97

Table 4 gives a comparison of the models CNN, VGG16 and EmotionNet in the validation dataset. The baseline CNN has modest performance with 58% accuracy, and relatively low macro precision (29%) and macro F1-score (32) which shows that it is sensitive to class imbalance and shows little generalization. However, transfer learning model of VGG16 yields 95 percent accuracy and balanced macro measures, which indicates the power of deep pretrained features.

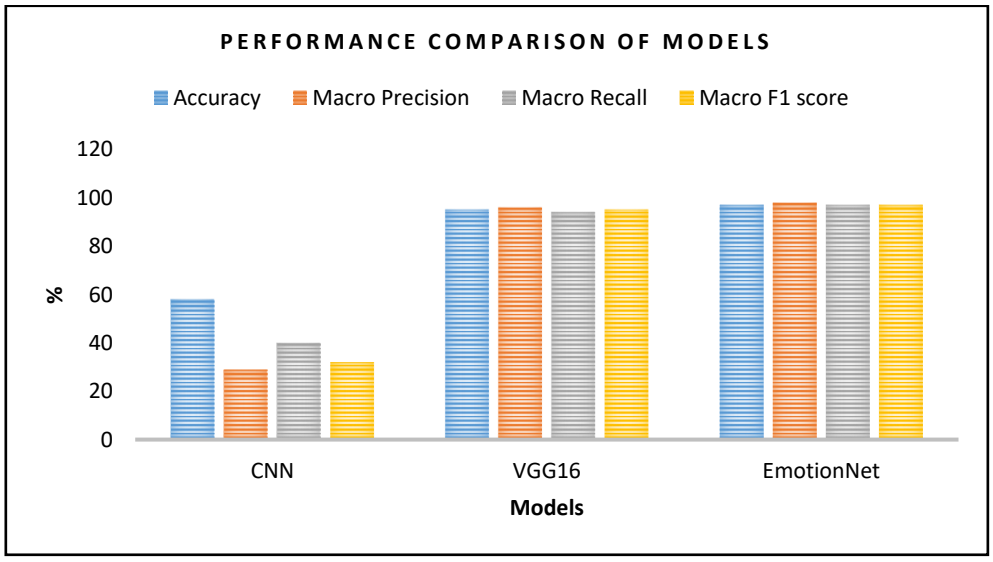


Figure 11. Performance Comparison of CNN, VGG16, and EmotionNet Models Across Evaluation Metrics

EmotionNet performs better than both models, reaches an accuracy of 97% with better macro precision (98%) and F1-score (97%), the use of improved architecture and the ability to perform better in classifying facial emotions.

Table 5. Models Parameters Comparison

Metric	EmotionNet (CNN-based)	Custom CNN Model	VGG-16 Model
Total Parameters	42,704,711	94,551	134,289,223
Trainable Parameters	42,704,711	94,551	119,574,535
Non-Trainable Parameters	0	0	14,714,688
Total Multiply-Adds	7.82 GFLOPs	62.50 MFLOPs	11.69 GFLOPs
Input Size (MB)	0.46	0.46	0.46
Forward / Backward Pass Size (MB)	147.13	2.80	82.67
Parameter Size (MB)	170.82	0.38	537.16
Estimated Total Memory Usage (MB)	318.41	3.64	620.29

Table 5 provides a comparative analysis of the complexity of models and required computational resources of EmotionNet, Custom CNN and VGG-16 models. The Custom CNN is very lightweight and has 94,551 parameters and low memory consumption (3.64 MB) which can be used in resource-constrained settings but not representative power. The distribution of parameters and memory usage is displayed in figure 12.

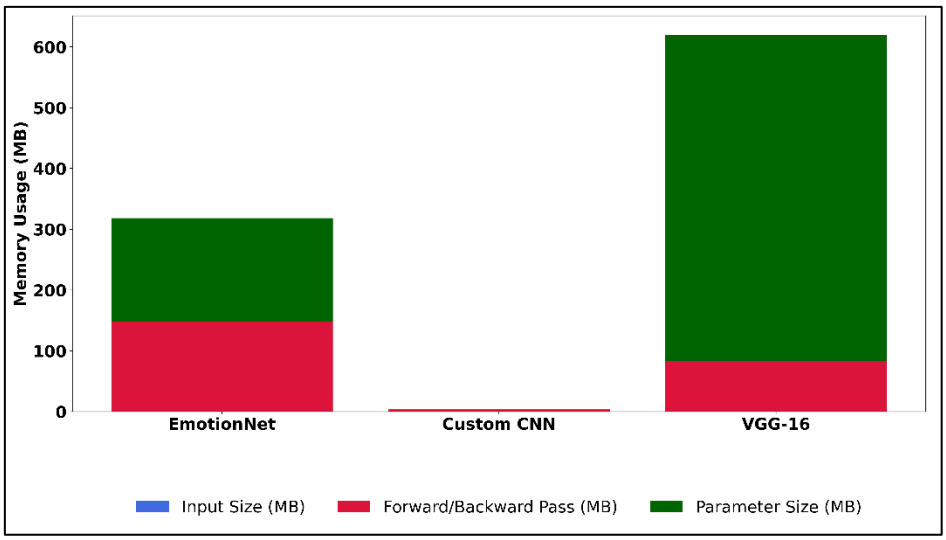


Figure 12. Deep Model Memory Footprint Distribution

VGG-16 is an excellent model in terms of feature extraction, having more than 134 million parameters as well as 620.29 MB estimated memory consumption, though it requires massive computational power. EmotionNet has a balanced design, 42.7 million parameters and memory usage of 318.41 MB, which is much lower than VGG-16 but provides high performance.

6. CONCLUSION

The paper presented EmotionNet, a deep learning framework that can be used in dynamic and practical settings to recognize facial emotions in real-time using video. The system that has been proposed is a combination of effective spatial feature extraction and effective classification to identify accurately seven basic emotion classes namely neutral, anger, disgust, fear, happy, sad and surprise. EmotionNet strikes a good balance between computational and predictive performance by using a

lightweight convolutional architecture, optimised preprocessing, augmentation, and evaluation, as well as. The relative experimental comparison showed that EmotionNet can be better than a baseline Custom CNN and the VGG16 transfer learning model in terms of overall accuracy, macro precision, recall, and F1-score. Although VGG16 has a high representational capacity, its computational complexity and memory usage make it impossible to run it in real-time. On the contrary, the CNN baseline, albeit lightweight, is not deep enough to provide strong generalization. EmotionNet is able to accomplish this gap by providing excellent classification results at greatly less memory overhead than big pretrained networks.

Authors' Contributions

All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by the authors. All authors read and approved the final manuscript

REFERENCES

[1] Bhojan, A., Ng, S. P., Ng, J., & Ooi, W. T. (2020). CloudyGame: Enabling cloud gaming on the edge with dynamic asset streaming and shared game instances. *Multimedia Tools and Applications, 79*, 32503–32523. https://doi.org/10.1007/s11042-020-09465-4

[2] Chaudhari, A., Bhatt, C., Krishna, A., & Mazzeo, P. L. (2022). ViTFER: Facial emotion recognition with vision transformers. *Applied System Innovation, 5*(4), 80. https://doi.org/10.3390/asi5040080

[3] Devaram, R. R., Beraldo, G., De Benedictis, R., Mongiovì, M., & Cesta, A. (2022). LEMON: A lightweight facial emotion recognition system for assistive robotics based on dilated residual convolutional neural networks. *Sensors, 22*(9), 3366. https://doi.org/10.3390/s22093366

[4] Fakhar, S., Baber, J., Bazai, S. U., Marjan, S., Jasinski, M., Jasinska, E., Chaudhry, M. U., Leonowicz, Z., & Hussain, S. (2022). Smart classroom monitoring using novel real-time facial expression recognition system. *Applied Sciences, 12*(23), 12134. https://doi.org/10.3390/app122312134

[5] Gerlowska, J., Dmitruk, K., & Rejdak, K. (2021). Facial emotion mimicry in older adults with and without cognitive impairments due to Alzheimer's disease. *AIMS Neuroscience, 8*(2), 226–238. https://doi.org/10.3934/Neuroscience.2021011

[6] Ghazouani, H. (2021). A genetic programming-based feature selection and fusion for facial expression recognition. *Applied Soft Computing, 103*, 107173. https://doi.org/10.1016/j.asoc.2021.107173

[7] Guerdelli, H., Ferrari, C., Barhoumi, W., Ghazouani, H., & Berretti, S. (2022). Macro- and micro-expressions facial datasets: A survey. *Sensors, 22*(4), 1524. https://doi.org/10.3390/s22041524

[8] Khan, G., Samyan, S., Khan, M. U. G., Shahid, M., & Wahla, S. Q. (2020). A survey on analysis of human faces and facial expressions datasets. *International Journal of Machine Learning and Cybernetics, 11*, 553–571. https://doi.org/10.1007/s13042-019-01040-5

[9] Laghari, A. A., Zhang, X., Shaikh, Z. A., Khan, A., Estrela, V. V., & Izadi, S. (2024). A review on quality of experience (QoE) in cloud computing. *Journal of Reliable Intelligent Environments, 10*, 107–121. https://doi.org/10.1007/s40860-023-00203-4

[10] Li, Y., Guo, K., Lu, Y., & Liu, L. (2021). Cropping and attention based approach for masked face recognition. *Applied Intelligence, 51*, 3012–3025. https://doi.org/10.1007/s10489-020-02091-8

[11] Ma, F., Li, Y., Ni, S., Huang, S.-L., & Zhang, L. (2022). Data augmentation for audio-visual emotion recognition with an efficient multimodal conditional GAN. *Applied Sciences, 12*(2), 527. https://doi.org/10.3390/app12020527

[12] Metzger, F., Geißler, S., Grigorjew, A., Loh, F., Moldovan, C., Seufert, M., & Hoßfeld, T. (2022). An introduction to online video game QoS and QoE influencing factors. *IEEE Communications Surveys & Tutorials, 24*(3), 1894–1925. https://doi.org/10.1109/COMST.2022.3179077

[13] Pan, S., Xu, G. J. W., Guo, K., Park, S. H., & Ding, H. (2024). Cultural insights in Souls-like games: Analyzing player behaviors, perspectives, and emotions across a multicultural context. *IEEE Transactions on Games, 16*(3), 758–769. https://doi.org/10.1109/TG.2023.3298423

[14] Pise, A. A., Alqahtani, M. A., Verma, P., Purushothama, K., Karras, D. A., Prathibha, S., & Halifa, A. (2022). Methods for facial expression recognition with applications in challenging situations. *Computational Intelligence and Neuroscience, 2022*, Article 9261438. https://doi.org/10.1155/2022/9261438

[15] Rusia, M. K., & Singh, D. K. (2022). A comprehensive survey on techniques to handle face identity threats: Challenges and opportunities. *Multimedia Tools and Applications, 82*, 1669–1748. https://doi.org/10.1007/s11042-022-13461-8

[16] Ter Burg, K., & Kaya, H. (2022). Comparing approaches for explaining DNN-based facial expression classifications. *Algorithms, 15*(10), 367. https://doi.org/10.3390/a15100367

[17] Wu, W., Yin, Y., Wang, X., & Xu, D. (2019). Face detection with different scales based on Faster R-CNN. *IEEE Transactions on Cybernetics, 49*(11), 4017–4028. https://doi.org/10.1109/TCYB.2018.2872478

[18] Xi, X., Xi, B., Miao, C., Yu, R., Xie, J., Xiang, R., & Hu, F. (2022). Factors influencing technological innovation efficiency in the Chinese video game industry: Applying the meta-frontier approach. *Technological Forecasting and Social Change, 178*, 121574. https://doi.org/10.1016/j.techfore.2022.121574

[19] Zangeneh, E., Rahmati, M., & Mohsenzadeh, Y. (2020). Low resolution face recognition using a two-branch deep convolutional neural network architecture. *Expert Systems with Applications, 139*, 112854. https://doi.org/10.1016/j.eswa.2019.112854

[20] Zuo, C., Zhang, X., Yan, L., & Zhang, Z. (2024). GUGEN: Global user graph enhanced network for next POI recommendation. *IEEE Transactions on Mobile Computing, 23*(12), 14975–14986. https://doi.org/10.1109/TMC.2023.3299187