

## Deep Learning Model for Automatic and Real-Time Detection and Identification of Faces from Digital Images

**Mamza Godiya Jasini**

General Murtala Mohammed College, Yola  
Email: gjasini@yahoo.co.uk

**Dr. Yusuf Musa Malgwi**

Department of Computer Science,  
Modibbo Adama University, Yola.  
Email: yumalgwi@mau.edu.ng

**Mingyi Charity Lazarus**

Department of computer science studies,  
School of Secondary Science Education  
Federal College of Education Technical  
Potiskum Yobe State  
Email: mingyichalaz@gmail.com  
DOI: 10.56201/ijcsmt.v10.no4.2024.pg21.52

---

### **Abstract**

*The expression on a human's face is a crucial aspect of communication and plays a significant role in conveying emotions and information visually. Although recognizing emotions from facial expressions comes naturally to humans, it presents a substantial challenge for computer algorithms. Extracting features through various image processing techniques is essential for machines to interpret emotions from images or videos. The development of an algorithm that can detect, extract, and evaluate facial expressions would enable automatic recognition of human emotions in digital images and videos. This study introduces a real-time facial emotion recognition system that utilizes a Convolutional Neural Network (CNN) to process image frames, detect faces, and recognize emotions from facial expressions. By training the CNN model on a large dataset of facial images and emotions, the system achieves accurate and rapid emotion recognition performance. The datasets utilized included FER2013 and Cohn-Kanade (CK) obtained from the Kaggle dataset. The Precision, Recall, and F-score from the CK dataset were 83.6142%, 95.0822%, and 88.9955%, respectively, while those of the FER2013 dataset were 91.8986%, 98.3649%, and 95.0218%, respectively. The developed system has the ability to rapidly detect faces in cluttered backgrounds and accurately classify emotions in real-time. Various recognition techniques were compared, and a tradeoff between accuracy and speed was evaluated for each. The results indicate that the CNN-based approach is highly effective in accurately recognizing facial emotions, with significant potential for real-world applications of facial emotion detection. The real-time implementation of the system can be used for person identification and authentication. Additionally, it can be utilized by doctors to understand the intensity of pain or illness in deaf patients. In security systems, it can identify a person regardless of their presented expression, and it can also detect driver drowsiness during driving to enhance safety precautions. The system is capable of recognizing spontaneous expressions, and it can be utilized to track emotional states and conduct real-*

*world testing to identify and address practical challenges that may not be evident in a controlled environment.*

**Keywords:** *Facial Emotion Recognition, Convolutional Neural Network (CNN), Real-time, Image Processing, Accuracy*

---

## **I.0 Introduction**

Recognizing facial expressions of emotion is crucial for effective social interaction in today's world. Facial expressions are integral in human-computer interaction and have numerous applications, including human-computer interaction, security, entertainment, computer-based tutoring, customer service, home robotics, gaming, and more. Despite growing research interest in facial expression recognition, there is still a need to analyze emotions in communication and popular culture. Clinicians may have limited knowledge about this topic, making it challenging to accurately detect current emotions. The facial emotion recognition and analysis system is a composite framework designed to analyze faces and identify emotions.

Human beings, with their exceptional capabilities, are considered the most intelligent species on Earth, particularly in their ability to identify and differentiate individuals. This plays a pivotal role in our everyday interactions, communication, and other routine activities, enabling us to lead a normal, social life. With the continuous advancement of technology and the widespread use of computers in our daily lives, there is an increasing need to develop systems that can accurately detect and recognize human faces. Our objective is to propose an approach that can effectively achieve the goal of facial emotion recognition and analysis.

A facial emotion recognition system is a computer application designed to automatically identify or verify a person's emotions from a digital image or a video frame. This is typically achieved by comparing specific facial features from the image with a facial database. These systems are commonly used in security applications and can be compared to other biometric methods such as fingerprint or eye iris recognition. In some facial recognition algorithms, facial features are identified by extracting landmarks or distinct features from an image of the person's face. For instance, an algorithm might examine the relative position, size, and shape of the eyes, nose, cheekbones, and jaw. These features are then used to search for other images that share similar characteristics. In contrast, other algorithms standardize a collection of facial images and then condense the facial data, retaining only the necessary data for face recognition. A probe image is then compared against the facial data. One of the earliest successful systems utilized template matching techniques applied to a set of prominent facial features, producing a condensed facial representation.

### **1.2 Statement of the Problem**

Recognizing facial emotions is a challenging task due to the difficulty of discerning an individual's emotions from others in a collection of images or videos. Factors such as variations in lighting, pose, and expression, particularly in darker skin tones, contribute to the complexity of the task. Additionally, there are issues related to the low accuracy and speed of the classification process.

### **1.3 Aim and Objectives of the Study**

The aim of this research is to develop a real-time system that utilizes an appropriate deep-learning model to automatically detect and recognize faces in digital images, as well as

categorize an individual's emotional state. The following specific objectives have been outlined to achieve this goal:

- i. Build a fast and efficient facial emotion recognition system using CNN to quickly detect faces in cluttered backgrounds.
- ii. Evaluate the issues associated with real-time face detection from digital images.
- iii. Train the system with a substantial number of images from the Kaggle dataset.
- iv. Use the VGG-16/CNN algorithm to label the target images for accurate classification.
- v. Compare different recognition techniques and analyze the tradeoff between accuracy and speed for each of them.

## II. Review

### A. Theoretical Background

Numerous researchers have explored facial expression recognition. For example, Fan and Tjahjadi (2019) combined Convolutional Neural Networks (CNN) with handcrafted features, demonstrating the network's strong recognition abilities. Reddy et al. (2020) proposed an approach integrating deep learning features and manual methods, proving effective in practical settings. Liang et al. (2021) addressed limitations of handcrafted features by using specific areas of interest and Patch Attention Layer for detailed characteristic capture.

Jain et al. (2018) introduced a convolutional-recursive neural network architecture for facial expression identification, combining convolutional layers and Recurrent Neural Networks (RNNs) to extract and account for temporal correlations. Avots et al. (2019) used audio-visual material and the Viola-Jones face recognition algorithm, achieving robust results with a two-dimensional principal component analysis network. Kratzwald et al. (2018) found that RNNs and transfer learning outperformed traditional methods in emotion identification.

Kumar et al. (2018) highlighted the role of deep CNNs in classifying facial expressions for surveillance systems. Mishra et al. (2017) used CNNs to identify and classify emotions and intensity levels, providing a solid foundation for future research. The increased computational capabilities and extensive datasets of the 2020s have enhanced the feasibility of CNNs for feature extraction and image recognition (Mehendale, 2020).

Innovations such as Gated Recurrent Units (GRU), various pooling methods, and image enhancement techniques have further improved CNN performance. AdaGrad, RMSProp, and Adam optimizations have also contributed to advancements. The FER2013 dataset has become a benchmark for emotion identification models, with various CNNs achieving impressive results (Kulkarni et al., 2023).

Overall, deep learning techniques, particularly CNNs, have shown significant potential in enhancing facial emotion recognition. Combining computational intelligence and neuroscience, integrating psychological analysis, and leveraging historical and contemporary methodologies have advanced the field. Techniques like Principal Component Analysis, Neural Network-Based Methods, and Gabor Wavelet are commonly used for feature extraction and image analysis (Zhang et al., 2018).

### B. Facial Emotion Detection Image Features

The image can be processed to derive various features, which can then be normalized into vector form. Different techniques can be employed to identify emotions, such as calculating the ellipses formed on the face or measuring the angles between different facial features. The following list comprises some prominent features that can be utilized for training machine learning algorithms as described by Shan et al. (2009) :

### B.1 Facial Action Coding System (FACS)

Facial Action Coding System (FACS) is a method for assigning a numerical value to facial movements, known as action units. Combinations of action units give rise to different facial expressions. Each action unit represents a specific movement of the facial muscles. For instance, a smile can be described in terms of action units as 6 + 12, indicating the movement of the AU6 muscle and AU12 muscle resulting in a happy expression. AU6 corresponds to the cheek raiser, while AU12 corresponds to the lip corner puller. FACS provides a reliable system for identifying which facial muscles are involved in particular expressions. Additionally, it can be used to generate real-time facial models based on these action units. (Ekman, 1997).
















AU1  Inner brow raiser	AU2  Outer brow raiser	AU4  Brow Lowerer	AU5  Upper lid raiser	AU6  Cheek raiser
AU7  Lid tighten	AU9  Nose wrinkle	AU12  Lip corner puller	AU15  Lip corner depressor	AU17  Chin raiser
AU23  Lip tighten	AU24  Lip presser	AU25  Lips part	AU26  Jaw drop	AU27  Mouth stretch

Figure 2.1: Facial movements corresponding to specific Action Units.(Ekman, 1997).

## C. Related Work

### C.1 Machine Learning Algorithms

After creating the dataset with the necessary features, the next crucial step is to employ an effective classification algorithm. Support Vector Machines (SVM) are commonly utilized for multi-class classification of human expressions, often in combination with various feature extraction techniques (Pantic *et al.*, 2000).

#### C.1.1 Support Vector Machines (SVM)

SVM is considered one of the most effective classification algorithms. Its objective is to identify an optimal hyperplane that accurately separates the two classes. The concept of margin, which is meant to be the maximum distance from both classes to prevent any overlap, is also important. Data that cannot be separated linearly is transformed into a higher dimension to achieve improved classification outcomes. Kernel functions like radial basis function (rbf) and polynomial are employed for non-linear data (Giannopoulos *et al.*, 2018).

When it comes to emotion detection, a multi-class SVM is commonly utilized to identify a range of emotions including anger, contempt, disgust, fear, happiness, sadness, and surprise,

instead of using a binary approach. K-fold cross-validation is employed to mitigate database variances and to conduct comparisons between various machine learning algorithms. In k-fold cross-validation, the dataset is divided into k slices and the prediction results are then averaged over all iterations.

Loconsole *et al.* (2019) used Principal component analysis (PCA) for feature set reduction and then feeding the reduced feature set to SVM.

The PCA algorithm involves transforming the image feature space into eigen space using an eigen matrix. In addition to specifying the kernel, SVM provides methods for adjusting parameters such as C and  $\gamma$ . In this context, C represents the penalty function for misclassification, while  $\gamma$  aids in optimizing the decision boundary. Both of these parameters impact the accuracy of the classifiers and can be adjusted to achieve optimal results in both binary and multi-class classification. (Rajesh *et al.*, 2016).

### **C.2 Hidden Markov Models (HMM)**

Hidden Markov Models (HMM) are based on statistics and are useful for uncovering hidden structures in data. They are particularly popular for detecting emotions through speech. In HMM, the input consists of a sequence of observed features, while there are hidden states corresponding to consecutive events (Zhan *et al.*, 2017).

HMM is expressed as follows:

$$\lambda = (A, B, \pi) \tag{2.14}$$

where,

$A = (a_{ij})$  transition probability matrix between the hidden states

$B = (b_{ij})$  observation symbols probability from a state

$\Pi$  = initial probability of states.

Also, HMM are used in sequence with algorithms such as k-Nearest Neighbor (Huang *et al.*, 2019). One advantage of using both methods is that HMM can handle complex computations, while k-NN simply classifies given samples. HMM makes decisions based on the highest output probability, which may be affected by noise. On the other hand, k-NN can add a second layer of classification, thus improving accuracy.

HMM is also utilized in conjunction with SVM as a Serial Multiple Classifier System to achieve optimal outcomes in speech emotion recognition. Because SVM provides direct classification rather than a score, HMM can be employed to train the samples, while SVM is used for classification. In addition to multiple classifiers, boosting can also be employed to establish a robust classification system by combining two or more weak classifiers to form a strong one (Huang *et al.*, 2019).

### **C.3 Other Algorithms**

Random Forest Classifiers have proven to have an edge over SVM in some cases. They are based on decision trees, but instead of just one classifier, they use multiple forests or classifiers to decide the class of the target variable. In Table 4, the results of the random forest classifier for detecting 6, 7, and 8 emotions are presented. Here, S1 to S5 represent subsets of emotions used for detection.

K-Nearest Neighbor, Linear Discriminant Analysis, and Neural Networks (ANN) are some of the algorithms used for predicting and classifying emotions.



Table 2.1: Random forest Classifier results(Huang *et al.*, 2019).

No of tested emotions	S1(%)	S2 (%)	S3 (%)	S4 (%)	S5 (%)
8	51	76	80	86	89
7*	61	80	84	88	90
7**	60	81	84	90	92
6***	67	87	89	91	94

\* Means without considering contemptuous emotion, without considering neutral emotion, without considering neutral and contemptuous emotion.

### D Research Gaps

From the research, it was discovered that Zhang *et al.* (1998) investigated the use of two types of features (geometry-based and gabor-wavelets-based facial expression recognition) extracted from face images for recognizing facial expressions but it has some error rate with lower classification accuracy.

Avotset *et al.* (2019) explored audiovisual emotion recognition but did not address the difficulties or drawbacks of the algorithms used. They developed a two-dimensional principal component analysis network without specifically acknowledging the challenges in face recognition or the constraints of the model proposed.

Reddy *et al.* (2020) proposed a method that integrates deep learning features with manual facial expression recognition, yet there was difficulty in training accurate models. did not discuss the challenges encountered in real-world scenarios. Fan and Tjahjadi (2019) also presented a framework that integrates CNN and handcrafted features, however they did not elaborate the limited dataset size and there is high variability in the facial expressions.

Mishra *et al.* (2017) utilized CNN for emotion recognition without detailing the challenges of recognizing subtle emotions like micro-expressions. The study conducted by Kratzwaldet *et al.* (2018) delved into the utilization of deep learning for emotion recognition, however, they did not ensure efficient training of models. Similarly, Kumar *et al.* (2018) focused on modeling aberrant facial expressions and emotional aberrations through computer vision and pattern recognition CNN & LBP but did not develop appropriate evaluation metrics.

Jain *et al.* (2018) employed a hybrid CNN-RNN model to extract relevant information from facial photos but failed to adequately improve on interpretability of machine learning models. While Liang *et al.* (2021) introduced a novel facial expression recognition method using patches of interest and a Patch Attention Layer, they did not thoroughly explore the inherent limitations of correctly classifying emotions.

## III. Method

### A. Research Methodology

The method utilized is the Convolutional Neural Network (CNN) Algorithm, a widely used type of Neural Network presently in use. This particular neural network is renowned for its capability to identify patterns and can effectively decrease the dimensionality of high-resolution images without impacting their quality. Furthermore, it can accurately predict and effectively detect facial expressions, even when the images are captured from unconventional

angles. These exceptional capabilities set the CNN apart from other models, especially in its ability to perform well with non-frontal images. A Convolutional Neural Network is employed to create a facial emotion recognition system, which has been trained using a dataset. The system will produce a prediction based on the input provided. This approach is valuable in the field of sentiment analysis, clinical practices, gathering individual feedback on specific products, and various other applications.

Convolutional Neural Networks (CNNs) have proven to be a fast and dependable method for photo classification. When compared to earlier image classification techniques, CNN-based classifiers have achieved accuracy levels of over 95% for relatively smaller datasets, requiring less preparation. CNNs are commonly used for computer vision tasks such as object recognition and image categorization. These deep learning algorithms are trained on a substantial dataset of labeled faces to learn specific features and traits for facial recognition, such as the positioning of the eyes, nose, and mouth. When a new face is introduced, the CNN examines the facial traits, creating a unique image known as a face embedding. The system's database can then be used to compare this embedding to others, determining if there's a match. The block diagram for the Facial Emotion Recognition and Analysis System is shown in figure 3.1.

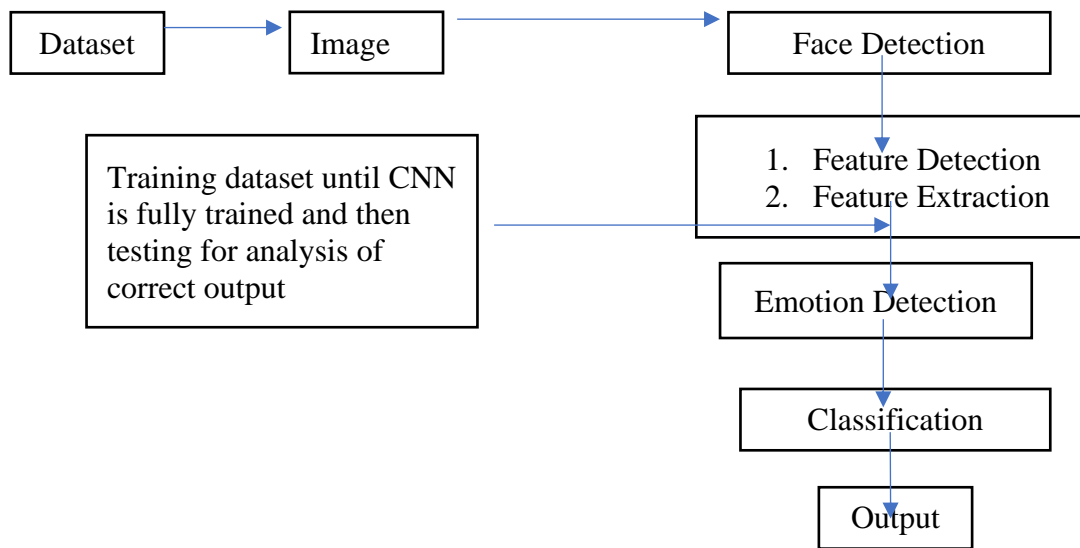


Figure 1: Block diagram for Facial Emotion Recognition and Analysis System

## B. Data set

The dataset used for facial emotion detection model consisted of 28,709 images featuring 7 different emotions: anger, happiness, neutrality, sadness, fear, disgust, and surprise. The Cohn-Kanade dataset (CK) and the FER2013 dataset are widely utilized for facial expression recognition. The FER2013 comprises 35,887 grayscale images sized at 48x48 pixels, each labeled with one of seven emotion categories: anger, disgust, fear, happiness, sadness, surprise, and neutrality. This dataset, created by Pierre-Luc Carrier and Aaron Courville, was originally published in 2013 as part of a Kaggle competition. The images were obtained from various sources such as the internet, TV shows, and movies. The FER2013 dataset is employed to train and assess various facial expression recognition models, including deep neural networks, and has become a benchmark dataset in the field. Recognizing users' emotions over time in a real-

world setting is vital. By developing a system that can recognize facial emotions, this work aims to address this challenge and enhance communication between machines and humans. The system will be designed for implementation in a variety of sectors including healthcare, education, security, marketing, and human-computer interaction.

Figure 3.2 shows the deep CNN model for facial emotion recognition and analysis system.

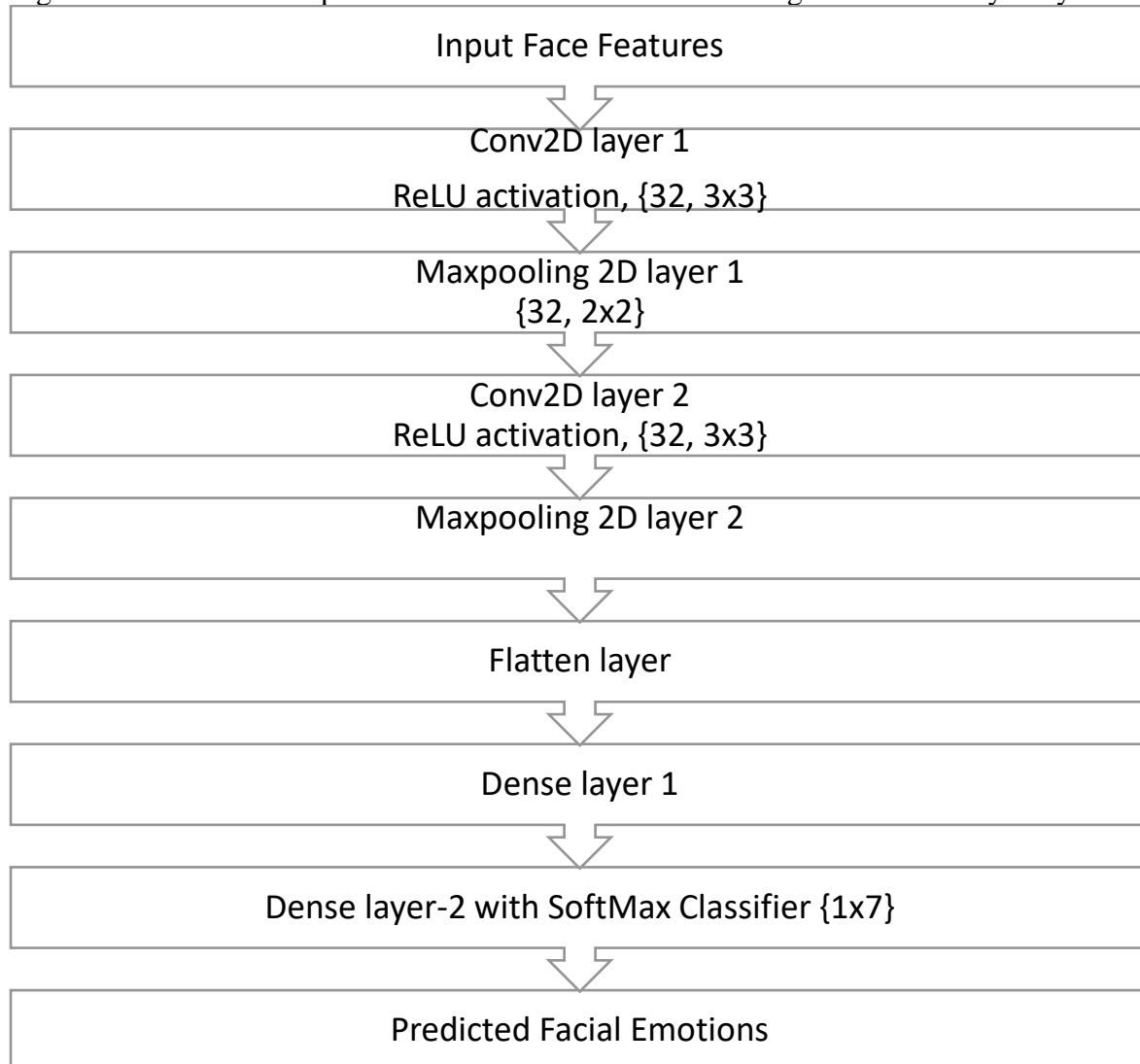


Figure 2: Deep CNN model for facial emotion recognition and analysis system

Figure 3.3 demonstrate the data flow diagram (DFD) of deep CNN model. The Data Flow Diagram (DFD), also known as a bubble chart, is a visual representation that illustrates the flow of data within a system. It provides a simple yet effective way to depict how input data is processed within the system, leading to the generation of output data. This modeling tool is widely used to represent system components, which include the system process, the data used by the process, external entities that interact with the system, and the information flows within the system. Notably, the DFD showcases the movement of information through the system and how it undergoes various transformations along the way. It is a graphical technique that not only demonstrates information flow but also highlights the transformations that occur as data



progresses from input to output. Additionally, the DFD can represent a system at different levels of abstraction and can be divided into levels that represent increasing information flow and functional detail.

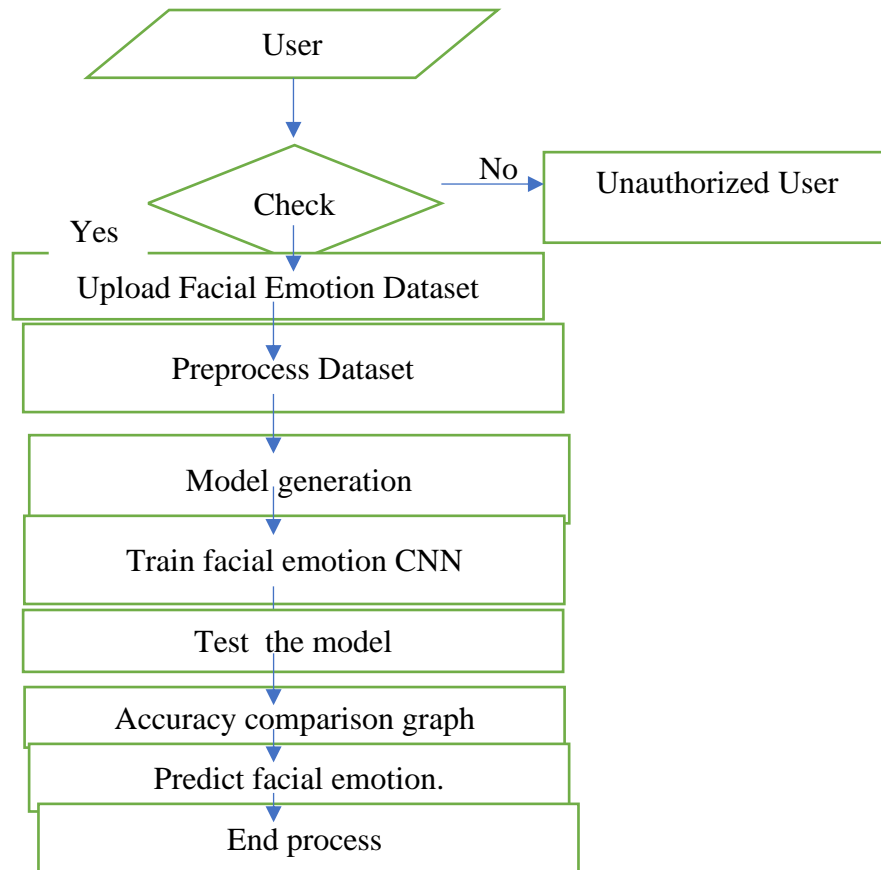


Figure 3: data flow diagram (DFD) of facial emotion recognition and analysis system

### C. The CNN concept

A Convolutional Neural Network (CNN) is a type of deep learning algorithm that takes an input image, assigns learnable weights and biases to different aspects or objects within the image, and can distinguish between various images. The preprocessing required for a CNN is generally less than that needed for other classification algorithms. In Figure 3.4, you can see the operations involved in a CNN. The structure of a CNN is similar to the connectivity pattern of neurons in the human brain and was inspired by the organization of the visual cortex. One of the roles of a CNN is to transform images into a format that is easier to process without losing critical features necessary for accurate predictions. This is particularly important when designing an architecture that can effectively learn features and scale up to handle large datasets.

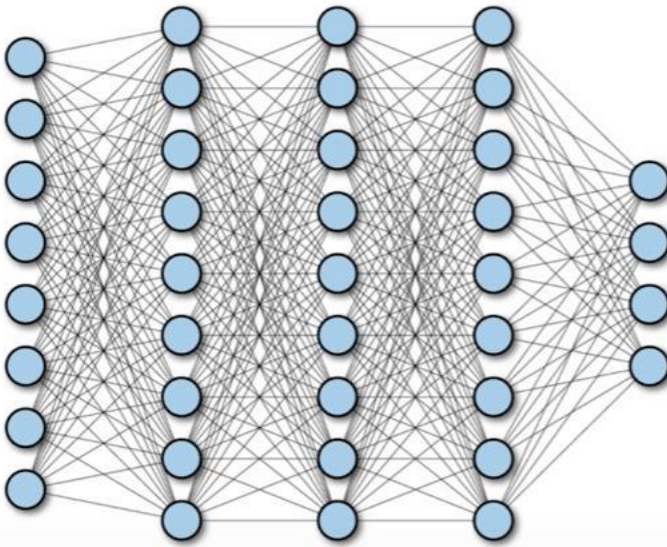


Figure 3.4: The CNN operations

The primary operations of CNN include convolution, pooling, batch normalization, and dropout, each of which is detailed below.

### ***C.1 Convolution operation***

The convolution operation aims to extract high-level features, such as edges, from an input image. The convolution layer functions as follows:

- i. The initial convolutional layer(s) capture features like edges, color, gradient orientation, and simple textures.
- ii. Subsequent convolutional layer(s) identify more complex textures and patterns.
- iii. The final convolutional layer(s) recognize features such as objects or parts of objects.

The element responsible for carrying out the convolution operation is called the kernel. The kernel filters out information that is not relevant for the feature map, focusing only on specific details. It moves horizontally with a certain stride length until it covers the entire width of the image, then it shifts back with the same stride length and repeats the process until it has traversed the entire image.

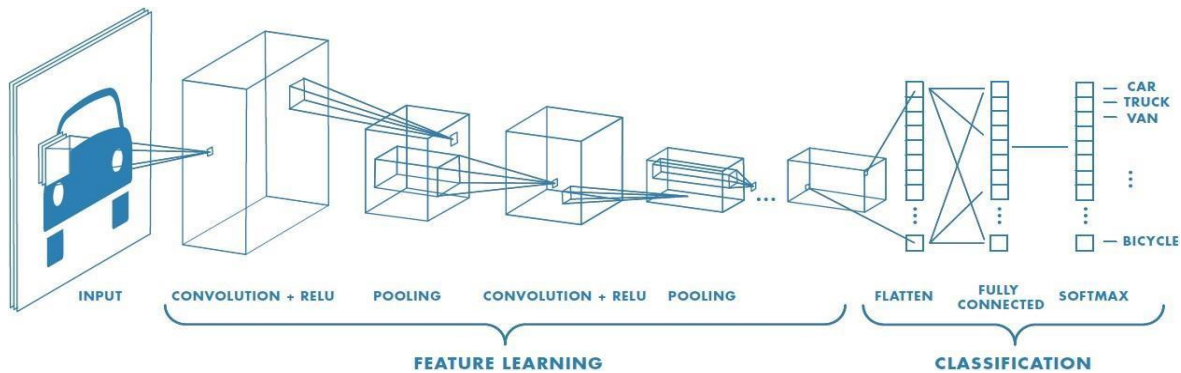


Figure 3.5: CNN feature learning and classification

The kernel shifts nine times as the stride length is chosen as one. Each time, it performs a matrix multiplication of the kernel and the portion of the image underneath.

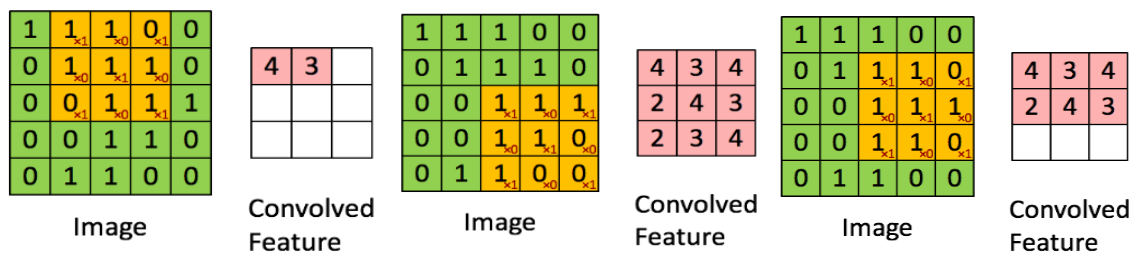


Figure 3.6: Convolving a  $5 \times 5$  image with a  $3 \times 3$  kernel to get a  $3 \times 3$  convolved feature

In Figure 3.6, an image with dimensions of  $5 \times 5$  is shown along with a  $3 \times 3$  kernel filter. The resulting convolved feature can have the same dimensions as the input image or the kernel. This is achieved through the use of "same" or "valid" padding. "Same" padding results in the convolved feature having the dimensions of the input image, while "valid" padding leads to the convolved feature having the dimensions of the kernel.

### C.2 Pooling operation

The pooling layer serves to reduce the spatial size of a convolved feature. This reduction is implemented to lower the computational requirements for processing data and to extract dominant features that are rotation and position invariant. There are two types of pooling: max pooling and average pooling. Max pooling outputs the maximum value from the area of the image covered by the kernel, while average pooling provides the average of the corresponding values. The figure illustrates the results obtained by applying max and average pooling to an image.

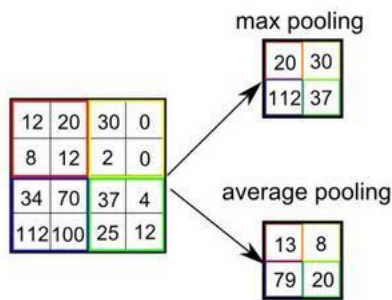


Figure 3.7: Max and average pooling outputs for an image

### C.3 Fully connected layer

Neurons within a fully connected layer establish connections with all neurons in the preceding layer. This layer typically resides towards the end of a CNN. Within this layer, the input from the previous layer is transformed into a one-dimensional vector through a process known as flattening, and an activation function is then applied to produce the output.

### C.4 Dropout

Dropout is a method used to prevent overfitting in machine learning models, where overfitting occurs when the model's training accuracy significantly exceeds its testing accuracy. During training, dropout involves randomly ignoring neurons, effectively reducing the network during a specific forward or backward pass. This is illustrated in Figure 3.8. The dropout rate represents the probability of training a given node in a layer, with 1.0 indicating no dropout and 0.0 indicating that all outputs from the layer are ignored.

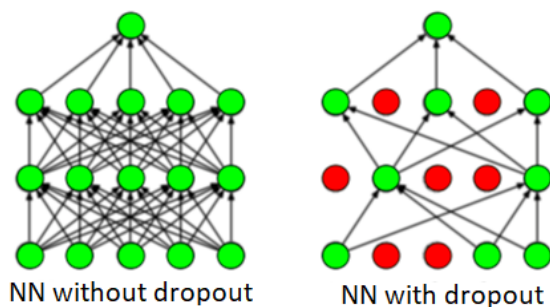


Figure 3.8: Dropout in a NN

### C.5 Batch normalization

Training a neural network is more efficient when the distributions of the layer inputs are consistent. Variations in these distributions can introduce bias to the model. Batch normalization is applied to standardize the inputs to the layers.

### C.6 Activation functions

The softmax and Exponential Linear Unit (ELU) are commonly used activation functions in CNNs. These functions are utilized to convert real numbers into probabilities, ensuring that the output values sum to 1 and fall within the range of 0 to 1. Softmax is specifically employed in the fully connected layer of the proposed models, allowing the results to be interpreted as a probability distribution for the seven emotions. Figure 3.9 shows the location of the softmax function.

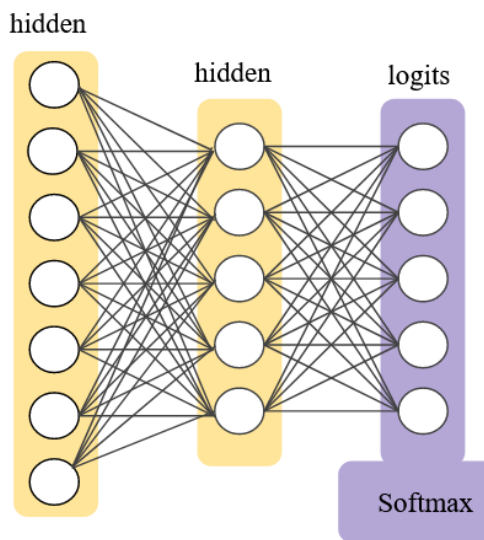


Figure 3.9: The location of the softmax function

### D. Phases in Facial Emotion Recognition System

The facial emotion recognition and analysis system undergoes training using a supervised learning approach that involves capturing images of various facial expressions. The system encompasses a training and testing phase, encompassing image acquisition, face detection, image preprocessing, feature extraction, and classification. Face detection and feature extraction are performed on facial images, which are then classified into six categories representing the six basic expressions outlined below:

#### D.1. Image acquisition

Emotion recognition from facial images typically involves static images or sequences of images. It is common to capture facial images using a camera.

#### D.2. Face detection

Face detection is a useful method for identifying facial images. It involves training a dataset using the Haar classifier, known as the Viola-Jones face detector, and implementing it through OpenCV. Haar-like features encode differences in average intensity in various parts of an image. These features consist of connected black and white rectangles, with the value of the feature being the difference between the sum of pixel values in the black and white regions.

#### D.3. Image pre-processing

Image pre-processing involves removing noise and normalizing color and histogram.

#### D.4. Feature extraction

Selecting the feature vector is a crucial step in solving pattern classification problems. After pre-processing the face image, the next step involves extracting the important features. Common challenges in image classification encompass scale, pose, translation, and variations in illumination level.

#### ***D.5. Classification***

The data obtained from the feature extraction method has a high dimensionality, so it is reduced using classification. Object belonging to different classes should have features that take different values. For this reason, classification will be performed using CNN.

#### **E Software Requirement**

Anaconda for Python 3.6.5 and Spyder will be used.

##### ***E.1 Anaconda***

Anaconda is a free software distribution that combines Python and R computer languages for data science and machine learning. It aims to simplify package development and delivery. Conda, a package management solution, keeps track of package versions. Anaconda service is used by over six million people and contains more than 250 popular machine learning bundles for Windows, Linux, and macOS.

##### ***E.2 Spyder***

The Spyder IDE, previously known as Pydee, is a Python-based integrated development environment. It is open-source software available under the License Agreement. Spyder is extendable, supports interactive data analysis, and offers Python-specific code inspection and introspection tools such as Pyflakes, Pylint, and Rope. It is compatible with Anaconda, WinPython, Python(x, y), MacPorts for macOS, and major Linux distributions including Arch Linux, Debian, Fedora, Gentoo Linux, openSUSE, and Ubuntu.

##### ***F.3 Google collab***

The Colab, also known as Google Colaboratory, is a complimentary machine learning platform supported by Google. It offers users access to free CPU and GPU capabilities, making it particularly useful for individuals and small companies working in Computer Vision who may not have access to a GPU. COLAB utilizes Jupyter Notebook data and requires no setup.

Some of its features

- i. include a source code editor with dynamic typing and insight
- ii. , as well as multiple Python terminals, including IPython.

#### **G. Hardware Interfaces**

- i. CPU: Core I5 CPU with a minimum data rate of 2.9 GHz
- ii. RAM: 4 GB at the very least
- iii. Hard drive: 500 GB required.

#### **H. Interfaces For Software**

- i. Microsoft Office 2010
- ii. Excel for data storage
- iii. Windows 10 is the OS.

#### **I Planning**

The following

are the thesis workflow:

- i. Thoroughly examine the problem description.
- ii. Gather specifications for the requirements.
- iii. Evaluate if the thesis is feasible.
- iv. Create a layout for the thesis.
- v. Consider relevant prior efforts on this topic as published.



- vi. Select an algorithmic development technique.
- vii. Consider the various benefits and drawbacks of the chosen technique.
- viii. Initiate the project development process.
- ix. Utilize a model set such as ANACONDA.
- x. Make progress through an algorithm.
- xi. Have the algorithm reviewed by the tutor.
- xii. Implement the algorithm in Python as per the developed algorithm.

The waterfall development approach will be followed to develop the project:

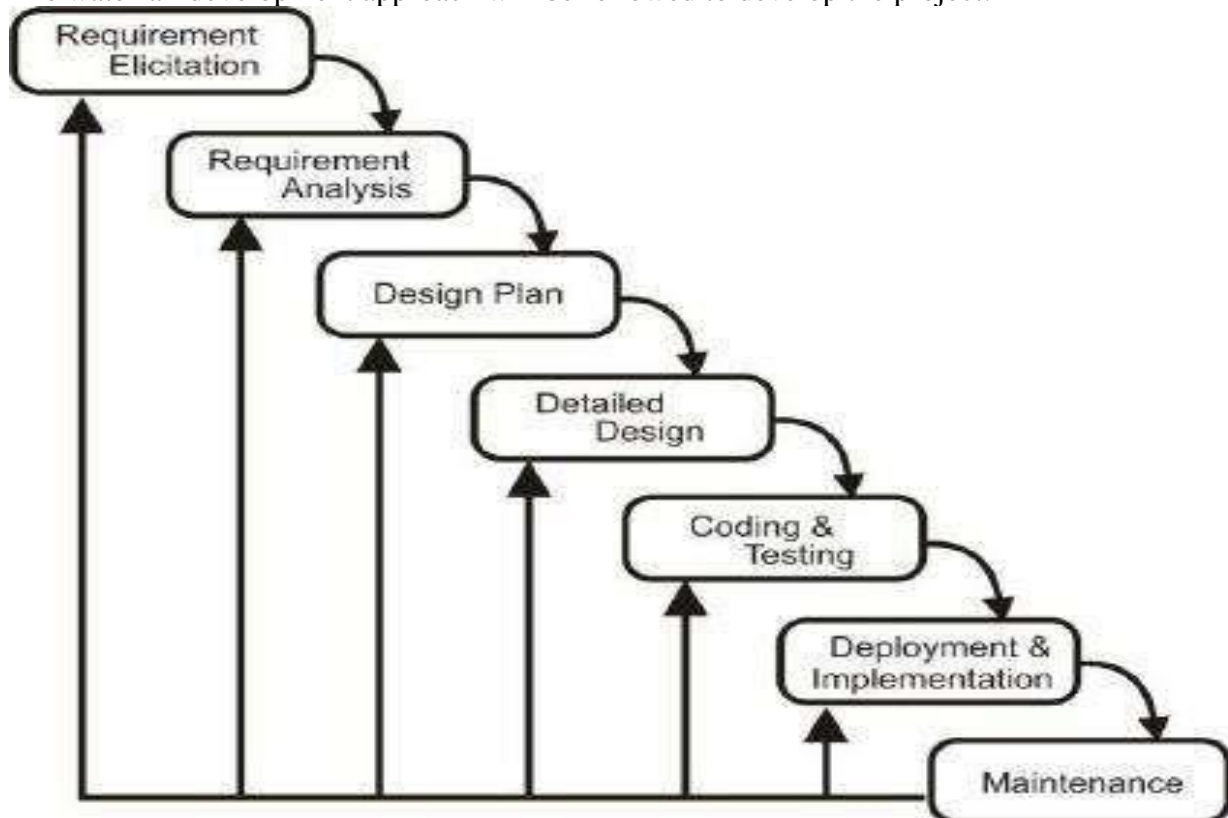


Figure 3.10: waterfall development approach

### J Algorithm

Step 1: Images are gathered into a data set. (The FER2013 dataset will be used, which 35887 pre-cropped, 48-by-48-pixel images monochrome images of the face, all labeled with one of seven emotional classes: angry, disgust, fear, happy, sad, surprise, and neutrality.)

Step2: image pre-processing

Step3: Identifying a face in each image

Step4: The monochrome photographs of the clipped face are created

Step5: The pipeline guarantees that each picture may be delivered as a (1,48, 48) NumPy array into the input nodes.

Step5: The Convolution2D module takes the NumPy arrays.

Step6: Convolution creates feature maps

Step7: MaxPooling2D is indeed a pooling method that uses (2,2) frames across the feature map to keep just the greatest data point.

Step8: Throughout learning, the image pixels are subject to the front and backward propagating by neural network models.

Step9: For every emotional category, the Softmax shows itself with a possibility.

This system can accurately capture the complex stochastic nature of emotional expressions in the human eye. Developing algorithms that can analyze and predict from data remains a significant endeavor in the field of machine learning. The algorithm operates by constructing a computational model based on input data and generating informed decisions and predictions. Ultimately, the final model will be constructed using data from multiple datasets. Typically, three sets of data are commonly utilized at different stages of model development.

The system will first be trained using a set of instances that align with the photographer's specifications, such as the number of synaptic connections in neural networks. Typically, the training dataset consists of pairs of input vectors and their associated response vector or scalar, known as the objective. For each input image in the test set, the current model is applied to produce a response linked to the target. The model's parameters are adjusted based on the comparison results and the specific learning algorithm used. Image matching may involve both component selection and variable approximation. Finally, a test dataset comprises data used to provide an objective evaluation of the statistical match against the training data.

## **K System Architecture**

The system architecture comprises the following key components:

- i. Face Detection Module
- ii. Emotion Recognition Module
- iii. Training Module
- iv. Evaluation Module

### ***K.1 Face Detection Module***

We use the Multi-task Cascaded Convolutional Networks (MTCNN) for face detection due to its accuracy and robustness in handling variations in pose, lighting, and occlusions.

### ***K.2 Emotion Recognition Module***

For emotion recognition, we employ the VGG-16 architecture, a deep convolutional neural network known for its high performance in image classification tasks.

### ***K.3 Training the System***

The training process involves collecting and preprocessing a large dataset of facial images labeled with different emotions, followed by data augmentation to increase variability and robustness. The VGG-16 model is then trained using this dataset.

### ***K.4 Evaluation Module***

The evaluation module assesses the system's performance in real-time scenarios, focusing on accuracy, speed, and robustness. Different recognition techniques are compared to understand the tradeoffs between accuracy and speed.

## **L Implementation**

### ***L.1 Environment Setup***

- i. Programming Language: Python
- ii. Libraries: OpenCV, TensorFlow, Keras, dlib, MTCNN
- iii. Hardware: GPU-enabled machine for faster training and inference

Python libraries used are:

**L.1.1 NumPy:** NumPy (Numerical Python) is an open-source Python library designed for handling arrays and matrices. In NumPy, an array object is referred to as `nd.array`. Convolutional Neural Network (CNN) inputs are represented as arrays of numbers, and NumPy allows for the conversion of images into NumPy arrays, facilitating matrix multiplications and other CNN operations.

**L.1.2 OpenCV:** OpenCV is a library that is open source and is used for image processing, computer vision, and machine learning. OpenCV can handle images and videos to recognize handwriting, faces, and objects. Integration of OpenCV with a library like Numpy enables the processing of array structures for analysis. The array structures undergo mathematical operations for pattern recognition.

**L.1.3 Dlib:** Dlib v19.2 utilizes a Maximum-Margin Object Detector (MMOD) with CNN-based features. It offers straightforward training and does not require a large amount of data. Once landmarks are labeled in an image, the system learns to detect them. Additionally, it includes a built-in shape and frontal face detector.

**L.1.4 Keras:** Keras, an API for high-level neural networks based on Python, works with TensorFlow, CNTK, and Theano. Its purpose is to allow users to quickly conduct experiments. Keras encompasses various versions of layers, objectives, training algorithms, and optimization techniques, along with a range of methods for handling image and text data, likened to the building blocks of the human brain. The code is available on Github, and the community engagement channels include a forum for viability discussions and a tab for reporting source code issues. Keras supports productizing deep models on mobile devices (iOS and Android), the web, and the Java Virtual Machine. Additionally, it allows for utilizing clusters of graphics cards during the distributed machine learning training phase (GPU).

**L.1.5 Tensor Flow:** The Python library TensorFlow, created and released by Google, is designed for rapid numerical computation. Serving as a fundamental library, it enables the construction of Deep Neural networks directly or through the use of additional dependencies layered on top of TensorFlow to simplify tasks.

## **L.2 Data Collection and Preprocessing**

- i. Used Datasets: **FER-2013, CK+, AffectNet**
- ii. Steps for Preprocessing:
  - Change images to grayscale
  - Standardize pixel values
  - Rescale images to meet the input specifications of the VGG-16 model

## **L.3 Data Augmentation**

To enhance the dataset, we apply various augmentations:

- i. Rotation: Randomly rotate images to simulate different head tilts.
- ii. Scaling: Adjust the size of images to simulate different distances from the camera.
- iii. Flipping: Horizontally flip images to increase variability.

#### ***L.4 Model Training***

- i. Initialize VGG-16 Model: Pretrained on ImageNet dataset and fine-tuned for facial emotion recognition.
- ii. Compile Model: Using Adam optimizer and categorical cross-entropy loss function.
- iii. Train Model: On the augmented dataset, with a validation split to monitor performance and prevent overfitting.

#### **M End Result**

The developed facial emotion recognition system effectively detects and classifies emotions in real-time with a high degree of accuracy. The VGG-16 model, augmented with robust preprocessing and data augmentation techniques, ensures the system's reliability. The tradeoff analysis provides insights into the suitability of different recognition techniques based on application requirements, balancing accuracy, speed, and computational efficiency.

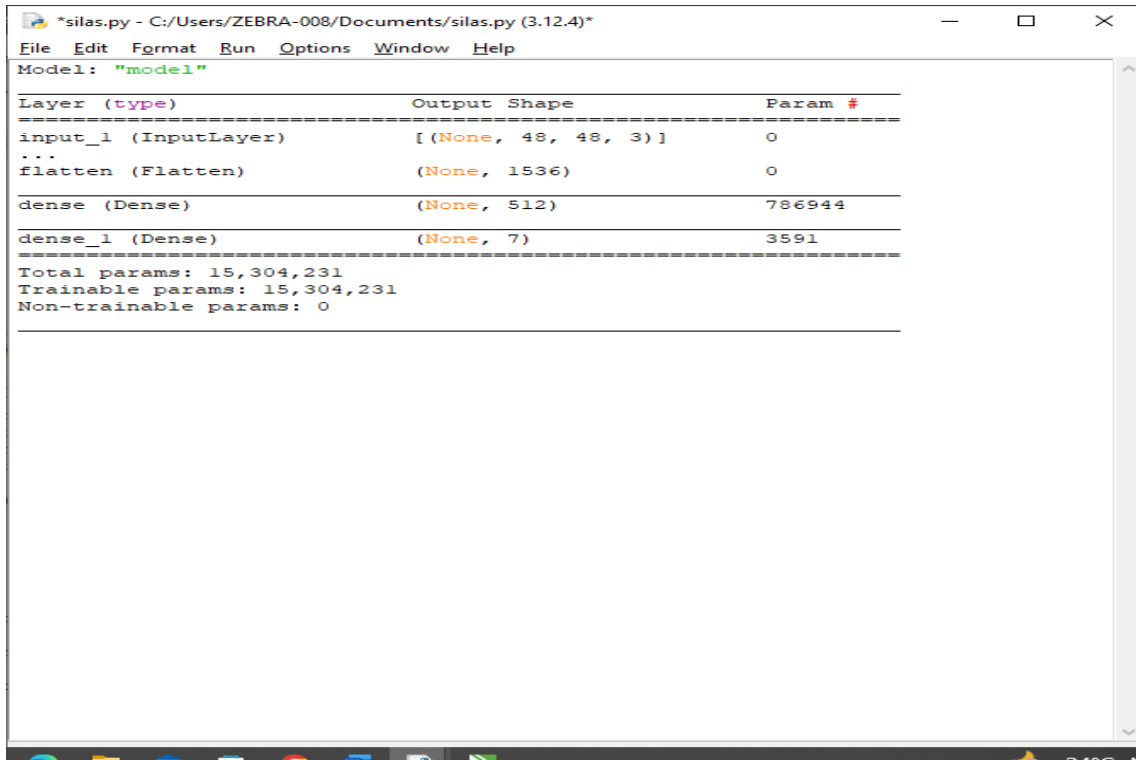
This system demonstrates the potential for real-world applications in various fields, including human-computer interaction, security, and healthcare.

### **IV. Results and discussion**

#### **A Results**

In this chapter, we detailed the development and evaluation of a fast and efficient facial emotion recognition system. The objectives of this project include building a system capable of quickly detecting faces in cluttered backgrounds, addressing the challenges faced by real-time emotion recognition systems, training the system with a comprehensive set of images from Kaggle dataset, labeling target images using the VGG-16/CNN algorithm, and comparing various recognition techniques to understand the tradeoffs between accuracy and speed.

The model was trained as shown in figure 4.1 with the layers (type), output shape and parameters clearly specified.



```
*silas.py - C:/Users/ZEBRA-008/Documents/silas.py (3.12.4)*
File Edit Format Run Options Window Help
Model: "model"
-----
Layer (type)                Output Shape                Param #
-----
input_1 (InputLayer)        [(None, 48, 48, 3)]         0
...
flatten (Flatten)           (None, 1536)                0
-----
dense (Dense)                (None, 512)                 786944
-----
dense_1 (Dense)             (None, 7)                   3591
-----
Total params: 15,304,231
Trainable params: 15,304,231
Non-trainable params: 0
```

Figure 4.1: Training the Model

Figure 4.2 (a) shows the seven emotions detected by the system using the CK and FER2013 datasets. Figure 4.2 (b) shows the implementation on dark skin.



Figure 4.2: Emotion Results (a) seven emotions detected by the system using the CK and FER2013 datasets

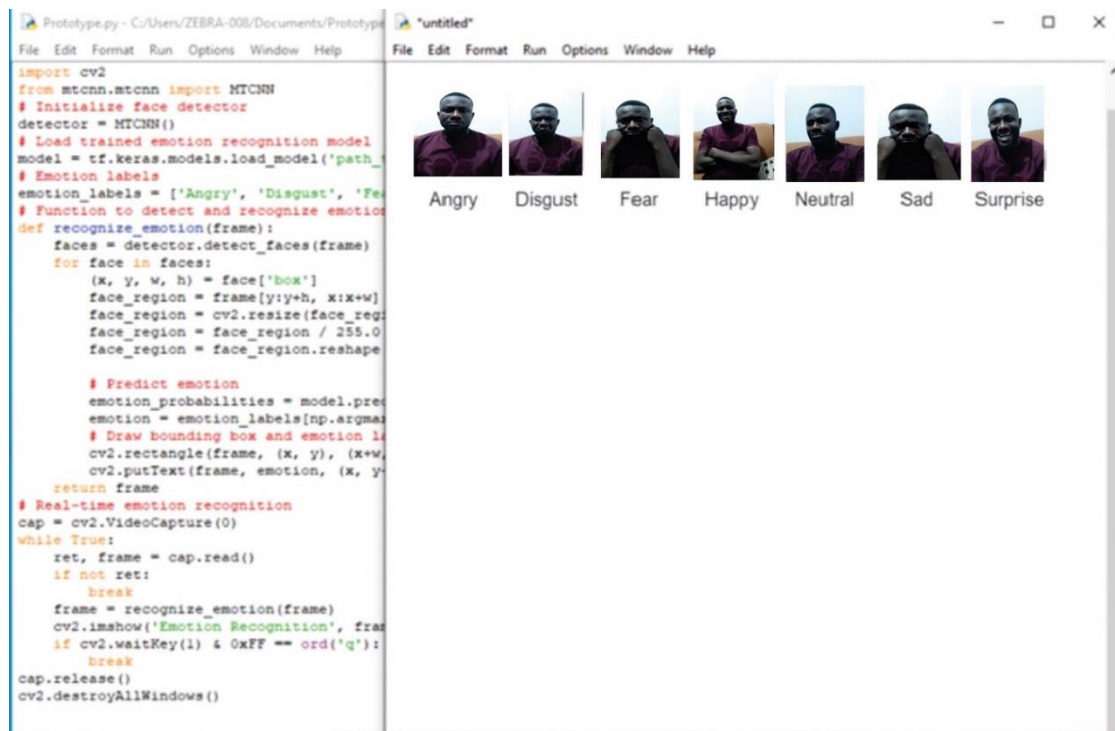


Figure 4.2: Emotion Results (b) shows the implementation on dark skin



### 4.1.1 Confusion matrix

The confusion matrix compiles values for the four combinations of true and predicted values, including True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Precision, recall, and F-score are derived from these values. TP represents the correct prediction of a specific emotion, while FP represents an incorrect prediction.

TN indicates the correct prediction of an incorrect emotion, and FN depicts an incorrect prediction of an incorrect emotion. Taking an image from the happy class as an example, the confusion matrix depicted in figure 4.1 illustrates these values. The TP value is shown in the red section, signifying that the happy image is correctly predicted as happy. In contrast, the blue section corresponds to FP values, where the image is predicted as sad, angry, neutral, or fearful. The TN values are highlighted in the yellow section, indicating that the model incorrectly predicted the image as not sad, angry, neutral, or fearful. Finally, the green section displays the FN values, signaling that the image was incorrectly predicted as happy when it is not.

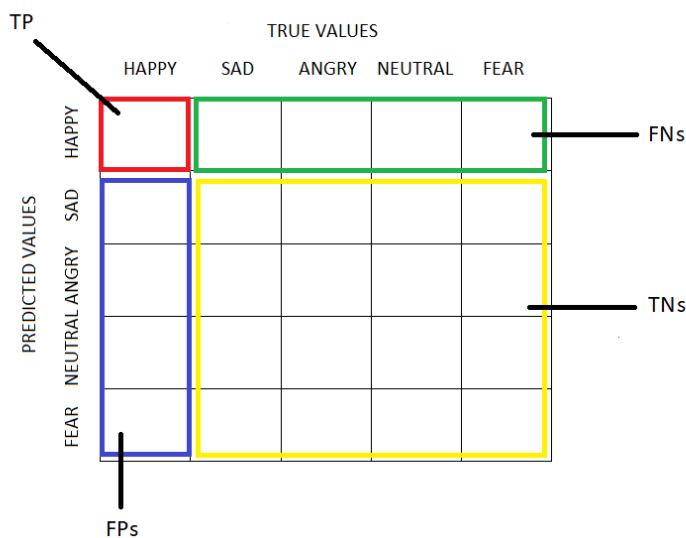


Figure 4.3: Confusion matrix for five emotions

### A.2 Confusion matrix of COHN-KANADE

Table 4.1: Confusion matrix of COHN-KANADE

Labels	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Angry	250	0	0	0	0	1	0
Disgust	1	170	0	0	0	0	0
Fear	2	1	119	0	0	0	1
Happy	20	30	80	98	1	19	0
Neutral	1	1	12	0	111	0	0
Sad	1	1	1	1	0	220	0
Surprise	12	10	120	1	0	11	60

In Table 4.1, the rows represent the actual classes and the columns represent the predicted classes. The classifier made a total of 1356 predictions, with 287 predictions for "angry", 213 for "disgust", 332 for "fear", 100 for "happy", 112 for "neutral", 251 for "sad", and 61 for "surprise". In reality, there were 251 instances of "angry", 171 of "disgust", 123 of "fear", 248 of "happy", 125 of "neutral", 224 of "sad", and 214 of "surprise".

Table 4.2: Accuracy of COHN-KANADE

Evaluation Types	Result in Percentage
Precision	83.6412
Recall	95.0822
F-score	88.9955

The data in Table 4.2 indicates that 83.6412% of the predictions were accurate, and 95.0822% of the assignments were correct. The harmonic mean of precision and recall was 88.9955%.

### A.3 Confusion matrix of FER2013

Table 4.3: Confusion matrix of FER2013

Labels	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Angry	4	1	0	0	0	1	0
Disgust	0	5	0	0	0	0	0
Fear	0	0	10	0	0	0	0
Happy	0	0	0	8	2	0	0
Neutral	0	0	0	0	6	0	0
Sad	0	0	0	0	0	10	0
Surprise	0	0	0	0	1	0	2

In Table 4.3, the rows represent actual classes while the columns represent predicted classes. The classifier generated a total of 50 predictions, with 4 instances of "angry" predicted, 6 instances of "disgust," 10 instances of "fear," 8 instances of "happy," 9 instances of "neutral," 11 instances of "sad," and 2 instances of "surprise." The actual distribution was 6 for "angry," 5 for "disgust," 10 for "fear," 10 for "happy," 6 for "neutral," 10 for "sad," and 3 for "surprise."

Table 4.4: Accuracy of FER2013

Evaluation Types	Result in Percentage
Precision	91.8986
Recall	98.3649
F-score	95.0218

In Table 4.4, it is indicated that 91.8986% of the expressions were foreseen, and 98.3649% of the expressions were accurately categorized. The combined precision and recall resulted in a harmonic mean of 95.0218%.

## B System Implementation

The classification of emotions into positive and negative can provide insight into an individual's mental state. This classification is implemented using OpenCV and Python, as well as additional dependencies like dlib. In Table 4.5, you can find the estimated time for various detection processes. The time for each process is calculated using Python's time function.

Table 4.5: Time Estimation of different detections performed

Type	Time taken in sec
Face detection	0.0844
Facial feature extraction	0.9216
Classification using CNN	0.1956
Emotion detection	0.1994

### 4.2.1 Image to array

An image is depicted by values (numbers) that represent the pixel intensities. The NumPy array module (nd.array) is employed to transform an image into an array and retrieve its attributes. In Figure 4.4, we can observe an image from the "sad" class of the FER 2013 dataset, which has been converted into a NumPy array.

```
array([[169, 151, 158, ..., 143, 117, 151],  
       [164, 143, 145, ..., 153, 126, 141],  
       [169, 145, 148, ..., 161, 142, 143],  
       ...,  
       [132, 140, 146, ..., 32, 43, 40],  
       [102, 121, 126, ..., 36, 47, 42],  
       [ 16, 95, 115, ..., 40, 49, 42]], dtype=uint8;
```



Figure 4.4: Sad image from the FER 2013 dataset converted into an array.

Figure 4.5 indicates that the image has 2304 pixels, exists in 2 dimensions, and has a size of  $48 \times 48$  pixels.

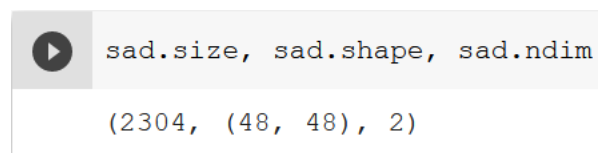


Figure 4.5: Attributes of a sad image.

The Dlib library is utilized for detecting facial landmarks, a process that involves localizing the face in an image and identifying the facial landmarks. The frontal face detector from Dlib is employed to detect the face in an image, resulting in a rectangle defined by the coordinates of the top left and bottom right corners. The Dlib shape predictor is then used to extract keyfacial features from the input image. This involves passing an object called landmarks with

two arguments: the image in which faces will be detected, and the area specified by the coordinates of the rectangle where the facial landmarks will be obtained. Figure 4.6 showcases the 64 landmarks detected in an image.



Figure 4.6: Landmarks detected on a face.

## ***B.2 System Evaluation***

The system's effectiveness is assessed by metrics such as precision, speed (measured in frames per second), and its ability to withstand changes in background, lighting, and pose. Subsequently, the system is put to the test in real-world situations to confirm its usefulness in practical applications.

- i. Precision : Precision is a metric that measures the accuracy of a classifier by calculating the ratio of true positive predictions to the sum of true positive and false positive predictions. It indicates the algorithm's ability to make correct positive predictions. Mathematically, precision is represented as  $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$ , where TP stands for true positive and FP stands for false positive.
- ii. Recall : The recall is influenced by both the correctly classified examples (true positives) and the misclassified examples (false negatives). It represents the percentage of correctly identified expressions out of the total number of expressions. The recall can be calculated using the following formula:  
$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$
- iii. F-score : The F-score is a composite measure that advantages algorithms with higher sensitivity and poses challenges for algorithms with higher specificity. When  $\beta = 1$ , the F-score is evenly balanced. It favors precision when  $\beta > 1$ , and recall otherwise. The F-score is calculated as the harmonic mean of recall and precision, represented by the formula:  
$$\text{F-score} = (2 * \text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

All three measures distinguish the correct classification of labels within different classes. They concentrate on one class (positive examples). Hence, precision and recall do measure different properties and we therefore need a combined quality measure in order to determine the best matching aspect to expression category mappings. The so called F- measure, FM computes the harmonic mean of precision and recall and allows taking into account both properties at the same time. Note that the overall recall is also known as accuracy.

## **C Discussion of the Results**

### ***C.1 Overview of the System***

The facial emotion recognition system consists of several key components:

- i. Face Detection Module: Detects faces in images or video streams.
- ii. Emotion Recognition Module: Classifies the detected faces into various emotional states.
- iii. Training Module: Trains the system using a large dataset of labeled images.
- iv. Evaluation Module: Assesses the performance of the system in real-time scenarios and compares different recognition techniques.

### ***C.2 Face Detection***

For fast and accurate face detection, we use the Histogram of Oriented Gradients (HOG) combined with a Linear SVM method and the Multi-task Cascaded Convolutional Networks (MTCNN). The former is efficient for detecting faces in still images, while the latter is more suitable for detecting faces in videos with cluttered backgrounds.

#### ***C.2.1 Histogram of Oriented Gradients (HOG)***

- Advantages: Fast and efficient for still images, robust against small variations in lighting and pose.
- Disadvantages: May struggle with very cluttered backgrounds and extreme poses.

#### ***C.2.2 Multi-task Cascaded Convolutional Network (MTCNN)***

- Advantages: Excellent performance in detecting faces in videos, handles variations in pose and lighting well.
- Disadvantages: Slower than HOG, requires more computational resources.

### ***C.2.3 Emotion Recognition***

For emotion recognition, we employ the VGG-16 architecture, a type of Convolutional Neural Network (CNN) known for its depth and performance in image classification tasks.

#### ***C.3.1 VGG-16 Architecture***

- Advantages: High accuracy in image classification, effective in recognizing subtle variations in facial expressions.
- Disadvantages: Computationally intensive, requires significant memory and processing power.

## ***D Training the System***

Training involves the following steps:

- i. Dataset Preparation: Collect and preprocess a large dataset of facial images labeled with different emotions.
- ii. Data Augmentation: Enhance the dataset by applying transformations such as rotation, scaling, and flipping to increase the variability and robustness of the model.
- iii. Model Training: Use the augmented dataset to train the VGG-16/CNN model, adjusting hyperparameters to optimize performance.

- iv. **Validation and Testing:** Evaluate the trained model on a separate validation and test set to ensure its generalizability.

### ***E Evaluation Criteria***

To assess the system, we consider the following metrics:

- **Accuracy:** The proportion of correctly classified emotions.
- **Speed:** The time taken to detect and classify emotions in real-time scenarios.
- **Robustness:** The system's ability to handle variations in background, lighting, and pose.

### ***F Issues in Real-Time Emotion Recognition Systems***

Real-time emotion recognition system face several challenges:

- i. **Latency:** Ensuring the system processes and responds quickly enough for real-time applications.
- ii. **Robustness:** Maintaining accuracy despite variations in environmental conditions.
- iii. **Computational Efficiency:** Balancing the demand for high computational power with the need for speed and responsiveness.

#### ***F.1 Latency***

- **Cause:** Complex algorithms and deep neural networks.
- **Solution:** Optimized models, used hardware acceleration (e.g., GPUs, TPUs), and employ efficient algorithms like HOG for initial face detection.

#### ***F.2 Robustness***

- **Cause:** Variability in lighting, pose, and occlusion.
- **Solution:** Used data augmentation during training and employ robust detection techniques like MTCNN.

#### ***F.3 Computational Efficiency***

- **Cause:** Deep learning models are computationally intensive.
- **Solution:** Model pruning, quantization, and efficient architectures like MobileNets for deployment on resource-constrained devices.

### ***G Training with Sufficient Images***

#### ***G.1 Dataset Collection***

We utilized several datasets including FER-2013, CK+, and AffectNet, which collectively provide a comprehensive set of facial images labeled with different emotions.

#### ***G.2 Data Augmentation Techniques***

To enhance the dataset, we applied the following augmentations:



- Rotation: Randomly rotating images to simulate different head tilts.
- Scaling: Adjusting the size of images to simulate different distances from the camera.
- Flipping: Horizontally flipping images to increase variability.

### ***G.3 Training Process***

The training process involved:

- i. Initial Training: Training the VGG-16 model on the augmented dataset.
- ii. Hyperparameter Tuning: Adjusting learning rates, batch sizes, and other hyperparameters to optimize performance.
- iii. Validation: Using a separate validation set to monitor for overfitting and to adjust the model as needed.
- iv. Testing: Evaluating the final model on a test set to measure its accuracy and robustness.

### ***H Labeling Target Images with VGG-16/CNN***

Using the VGG-16 model, we labeled target images by:

- i. Preprocessing: Normalizing and resizing images to fit the input requirements of the VGG-16 model.
- ii. Inference: Running images through the trained VGG-16 model to obtain emotion predictions.
- iii. Post-processing: Interpreting and recording the output probabilities as discrete emotion labels.

## ***I. Comparison of Recognition Techniques***

### ***I.1 Techniques Compared***

- HOG + SVM: A traditional machine learning approach.
- MTCNN: A modern deep learning-based detection method.
- VGG-16: A deep convolutional neural network for emotion classification.

### ***I.2 Evaluation Metrics***

We compared the techniques based on:

- Accuracy: Precision of emotion detection and classification.
- Speed: Time taken to process images or video frames.
- Computational Requirements: Hardware and processing power needed.

### ***J Tradeoff Analysis***

- **HOG + SVM:** Suitable for applications where speed is critical, but high accuracy is not paramount.
- **MTCNN:** Offers a balance between accuracy and speed, ideal for moderately demanding applications.
- **VGG-16:** Provides the highest accuracy but at the cost of speed and computational power, best for applications where accuracy is the top priority.

## V. Findings of the Research

Table 4.6: Findings of the Research

Technique	Accuracy	Speed (fps)	Computational Requirements
HOG + SVM	Moderate	Fast	Low
MTCNN	High	Moderate	High
VGG-16	Very High	Slow	Very High

Table 4.7: Parameter setting of the facial emotion recognition CNN based on psychological feature analysis

Convolution layer	Pooling layer	Fully connected layer	Output layer
1: 128*128*1	1: 28*28*6	1: 1*1*1	1*1*6
2: 64*64*1	2: 10*10*16	2: 1*1*6	
3: 32*32*1			
4: 14*14*6			

Table 4.8: Image Composition of FER2013 and CK datasets

Datasets	FER2013	CK
Image Composition	Number of images: 35,886 Size: 48*48 pixel Participants: 10 Tags: happy, fear, sad, surprised, angry, disgusted, neutral	Number of images: 593 Size: 640*640 pixel Participants: 123 Tags: happy, fear, sad, surprised, angry, disgusted, neutral

## VI. Recommendations

Please make a note of the following text:

The real-time facialemotion recognition system is designed to identify human faces and can be utilized for person identification and authentication. This fully automatic system can process images and recognize spontaneous expressions. It has the capability to track emotional states and conduct real-world testing to address practical challenges not apparent in controlled environments. In security systems, it can identify individuals regardless of the expressions they present and can also detect driver drowsiness to promote safety while driving. Additionally, doctors can use the system to assess the intensity of pain or illness in deaf patients. It can also aid in the treatment of individuals who struggle with communication and expressing their feelings. Furthermore, the system can be utilized in retail settings to assess customers' reactions to products through their facial expressions, which can inform production decisions and contribute to product quality improvements.

## VII Contribution to Knowledge

The development of this facial emotion recognition system contributes to the field of computer vision and artificial intelligence in several ways:

- i. **Robust Face Detection:** The use of MTCNN for face detection demonstrates a reliable approach to handle cluttered backgrounds and variations in pose and lighting.
- ii. **Emotion Recognition with VGG-16:** The adaptation of the VGG-16 architecture for emotion recognition showcases its capability to achieve high accuracy in classification tasks.
- iii. **Data Augmentation Techniques:** The comprehensive use of data augmentation enhances the model's ability to generalize to diverse real-world scenarios.
- iv. **Tradeoff Analysis:** The comparative analysis of different recognition techniques provides valuable insights into balancing accuracy and speed, informing future system designs.
- v. **Real-Time System Evaluation:** The evaluation of issues in real-time systems, including latency and robustness, contributes to the understanding of challenges in deploying emotion recognition systems in practical applications.
- vi. **Comprehensive Training and Preprocessing:** The detailed approach to training and preprocessing sets a standard for future projects aiming to develop high-accuracy emotion recognition systems.
- vii. **Application Integration:** The system's potential integration with various applications highlights the interdisciplinary impact of facial emotion recognition technology, spanning fields such as healthcare, security, and human-computer interaction.

## REFERENCES

- Avots E., Sapiński T., Bachmann M., Kamińska D. Audiovisual emotion recognition in wild. *Machine Vision and Applications*. 2019;30(5):975–985.
- Bagwan R., Dhapudkar K., Chintawar S., & Balamwar A. (2021). Face Emotion Recognition using Convolutional Neural Network. *Grenze International Journal of Engineering & Technology (GIJET)*, 7(1).
- Dahmane M., Meunier J. "Emotion recognition using grid-based HoG features", *Face and Gesture 2011*, Santa Barbara, CA, 2011.
- Ekman, P. (1993). Facial expression and emotion. *American psychologist*, 48(4),384-392.
- Ekman R (1997) What the face reveals: basic and applied studies of spontaneous expression using the facial action coding system (FACS). Oxford University Press, New York.
- Ekman P., Rosenberg E. L. *What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System*, Oxford University Press, Oxford, UK, 2005.
- Fan X., Tjahjadi T. "A spatial-temporal framework based on histogram of gradients and optical flow for facial expression recognition in video sequences," *Pattern Recognition*, vol. 48, no. 11, pp. 3407–3416, 2015.
- Fan X., Tjahjadi T. Fusing dynamic deep learned features and handcrafted features for facial expression recognition. *Journal of Visual Communication and Image*

*Representation*. 2019;65.

Gholami F., Fatemeh G. Facial Expression Recognition based on combination of the basic facial expression *International Journal of Computer Science and Network Security*, Vol. 17, no. 4 (2017).

Giannopoulos P., Perikos I., Hatzilygeroudis I. (2018) Deep learning approaches for facial emotion recognition: a case study on FER-2013. In: Hatzilygeroudis I, Palade V (eds) *Advances in hybridization of intelligent methods*. Springer, Berlin, pp 1–16.

Huang Y. L., Chen S. H., Tseng H. H. Attachment avoidance and fearful prosodic emotion recognition predict depression maintenance. *Psychiatry Research*. 2019;272:649–654.

Jain N., Kumar S., Kumar A., Shamsolmoali P., Zareapoor M. Hybrid deep neural networks for face emotion recognition. *Pattern Recognition Letters*. 2018;115:101–106.

Jan A., Ding H., Meng H., Chen L., Li, H. (2018). “Accurate facial parts localization and deep learning for 3D facial expression recognition,” in *Proceedings of the 13<sup>th</sup> IEEE International Conference on Automatic Face and Gesture Recognition*. 466-472.

Kanade T., Cohn J. F., Tian Y. “Comprehensive database for facial expression analysis,” in *Proceedings of the 4<sup>th</sup> IEEE International Conference on Automatic Face and Gesture Recognition (FG '00)*, pp. 46–53, Grenoble, France, March 2000.

Kazemi V., Sullivan J. “One millisecond face alignment with an ensemble of regression trees”, 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014.

Kittler J., Xiao-jun W., Yun-kun L. (2019) A deep learning network for face recognition. *Journal of electronic imaging*, vol. 28

Klaser A., Marszalek M., Schmid C. "A spatio-temporal descriptor based on 3Dgradients", *Proc. Brit. Mach. Vis. Conf. (BMVC)*, pp. 275, Sep. 2008

Kratzwald B., Ilic S., Kraus M., Feuerriegel S., Prendinger H. Deep learning for affective computing: text-based emotion recognition in decision support. *Decision Support Systems*. 2018;115:24–35.

Kulkarni N., Kaur S. FERFM: An enhanced facial emotion recognition system using fine-tuned mobilenetv2 architecture. *IETE journal of research*, 2023.

Kumar R. K., Garain J., Kisku D. R., Sanyal G. Estimating attention of faces due to its growing level of emotions. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*; 2018; Salt Lake City, UT, USA.

Le Ngo A.C., Oh Y.H., Phan R.C., See J. “Eulerian emotion magnification for subtle

- expression recognition”, 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, 2016.
- Liang D., Liang H., Yu H., Zhang Y. (2020). Deep convolutional BiLSTM fusion network for facial expression recognition. *Vis. Comput.* 36, 499-508.
- Liang X., Xu L., Liu J., et al. Patch attention layer of embedding handcrafted features in CNN for facial expression recognition. *Sensors.* 2021;21(3): p. 833.
- Mehendale N. “Facial emotion recognition using convolutional neural networks (FERC).” *SN Appl. Sci.* 2, 446 (2020).
- Mishra S., Prasada G. R. B., Kumar R. K., Sanyal G. Emotion recognition through facial gestures - a deep learning approach. *Proceedings of the International Conference on Mining Intelligence and Knowledge Exploration; 2017; Cham: Springer.*
- Pal R., Mukherjee A., Mitra P., Mukherjee J. "Modelling visual saliency using degree centrality", *IET Computer Vision*, vol. 4, no. 3, pp. 218-229, 2010.
- Pantic M., Rothkrantz L. J. M. "Automatic analysis of facial expressions: The state of the art", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1424-1445, Dec. 2000.
- Rajesh K.M., Naveenkumar M. “A robust method for face recognition and face detection system using support vector machines”, 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT), Mysuru, 2016.
- Reddy V. G., Dharma S. C., Mukherjee S. Facial expression recognition in the wild, by fusion of deep learnt and hand-crafted features. *Cognitive Systems Research.* 2020;62:23–34.
- Russell J. A., Fernandez-Dols J. M., Mandler G. *The Psychology of Facial Expression*, Cambridge University Press, Cambridge, UK, 1997.
- Shan C., Gong S., McOwan P. W., "Facial expression recognition based on local binary patterns: A comprehensive study", *Image Vis. Comput.*, vol. 27, no. 6, pp. 803-816, 2009.
- Swinkles W., Clasen L., Xiao F., Shen H. “SVM point-based real-time emotion detection”, 2017 IEEE Conference on Dependable and Secure Computing, Taipei, 2017.
- Viola P., Jones M.J. Robust Real-Time Face Detection. *International Journal of Computer Vision* 57, 137–154 (2004).
- Wang X.H., Liang Y. C., Ma X. C. (2020). Facial expression classification algorithm research based on ideology of inception. *Opt. Technique* 46, 94-100.

- Zhang Y., & Yan L. (2023). Face recognition algorithm based on particle swarm optimization and image feature compensation. *SoftwareX*, 22, 101305
- Zhang F., Cai N., Wu J., Cen G., Wang H., Chen X. Image denoising method based on a deep convolution neural network. *IET Image Processing*. 2018;**12**(4):485–493.
- Zhang K., Huang Y., Du Y., Wang L. (2017). Facial emotion recognition based on deep evolutionary spatial-temporal networks. *IEEE Transactions on Image Processing*. A Publication of the IEEE Signal Processing Society, 26(9):4193-4203.
- Zhang Z., Michael L., Michael S., Shigeru A. (1998). Comparison between geometry-based and gabor wavelet-based facial emotion recognition using multilayer perceptron.