# Revisiting Effective State Detection: A Comparative Analysis of Facial Expression Recognition Models in Human-Computer Interaction

**Ali Ullah** ALI.ULLAH@FAU.DE
*Master of Science - Artificial Intelligence, Technische Fakultät*
*Friedrich-Alexander-Universität Erlangen-Nürnberg*
*Matriculation Number: 23269239*

**Fida Hussain** FIDA.HUSSAIN@FAU.DE
*Master of Science - Artificial Intelligence, Technische Fakultät*
*Friedrich-Alexander-Universität Erlangen-Nürnberg*
*Matriculation Number: 23209327*

**Hamza Naeem** HAMZA.NAEEM@FAU.DE
*Master of Science - Artificial Intelligence, Technische Fakultät*
*Friedrich-Alexander-Universität Erlangen-Nürnberg*
*Matriculation Number: 23173252*

**Ghulam Mustafa** GHULAM.MUSTAFA@FAU.DE
*Master of Science - Artificial Intelligence, Technische Fakultät*
*Friedrich-Alexander-Universität Erlangen-Nürnberg*
*Matriculation Number: 23186746*

**Submitted to:** Prof. Dr.-Ing. Tobias Günther
Department of Computer Science
Friedrich-Alexander-Universität Erlangen
tobias.guenther@fau.de

## Abstract

In this project, we have developed an interactive visualization Web Application for real-time facial emotion detection. By using Flask, a Python framework, and fundamental HTML and CSS, the website offers users the capability to detect and analyze facial emotions in live video streams. The core functionality involves the implementation of three distinct models: a Convolutional Neural Network (CNN), the Chehra model, and a Deep Neural Network (DNN). These models have been trained to recognize various emotional states accurately.

Our project stands out for its emphasis on interactive visualization, showcasing detected emotion values dynamically on the web page. Furthermore, to facilitate comprehensive analysis, the Web app renders these emotion values through a bar chart graph, providing users with a visual representation of detected emotions over time.

Throughout the development process, our focus has been on achieving robust and accurate emotion detection. We have evaluated the performance of each model, considering factors such as accuracy, speed, and resource efficiency. By employing a diverse set of models, we aim to provide users with a comprehensive tool for real-time emotion analysis.

This project serves as a practical demonstration of the potential applications of machine learning in real-time emotion detection and interactive visualization. The integration of Flask and basic web technologies enables seamless development and accessibility, making our solution suitable for a wide range of applications, from educational tools to interactive experiences.

## 1. Introduction

In a world increasingly driven by technology, understanding human emotions in real-time holds significant value, whether applied in mental health assessments, user experience enhancement, or educational contexts, the system aims to contribute meaningfully to various domains. Introducing the Facial Emotion Recognition System, a vital project aimed at leveraging cutting-edge technologies to enhance user experiences. Through our system, users can seamlessly engage with their web camera, allowing us to analyze and categorize emotions using specialized models. This not only provides users with valuable insights into emotional dynamics but also enables them to make informed decisions about the specific emotion category they wish to explore. The real-time labeling and visualization of emotions on the front end, accompanied by detailed statistics, offer a holistic understanding of the emotional landscape. The primary objective of the Facial Emotion Recognition System is to provide a sophisticated and user-friendly platform for real-time analysis and categorization of human emotions through facial expressions. In essence, our project addresses the need for intuitive emotional analysis, offering a practical and user-centric solution that can find applications in diverse fields, from mental health to user experience design. We envision a future for the Facial Emotion Recognition System that involves anticipating its integration into various domains, influencing diverse facets of society and technology. As advancements in artificial intelligence and human-computer interaction continue, the system is poised to become a staple in applications ranging from mental health and well-being assessments to immersive user experiences.

## 2. Literature Review

### 2.1 Chehra Descriptor for Facial Emotion Detection

Facial expression recognition is an important topic in affective computing, with far-reaching implications in fields such as human-computer interaction and healthcare. The precise representation of facial features is crucial to this subject, and approaches such as the Chehra descriptor are used to accomplish this objective. This review investigates the Chehra descriptor's efficacy in facial emotion identification, with a specific emphasis on its use in the proposed methodology.

The Chehra descriptor, developed by Asthana et al. (2014), uses a geometric technique to identify 49 face landmarks, including the brows, eyes, nose, lips, and mouth. This approach detects face landmarks in real-time via a cascade of linear regressions, even in uncontrolled environments. By expressing these landmarks as Cartesian coordinates, a 98-dimensional feature vector is created, serving as the foundation for subsequent research.

However, intrinsic obstacles exist because of variances in facial forms, which may compromise recognition accuracy (Salah et al., 2010). To address this, Martinez (2011) presented a new feature representation technique based on intra-facial component distances. This approach generates an 1176-dimensional feature vector by computing the Euclidean distances between all facial landmark points, allowing it to capture complicated facial configurations and improve descriptor discrimination.

In tandem, recent advances in machine learning have had a substantial impact on face emotion recognition. Studies by Liew and Yairi (2015) and Lopes et al. (2017) demonstrated the effectiveness of Support Vector Machine (SVM) and Convolutional Neural Networks (CNN) in attaining high classification accuracies on benchmark datasets. These algorithms successfully extract discriminative patterns from facial data by utilizing techniques such as Histogram of Oriented Gradients (HOG) and deep learning architectures.

In conclusion, the Chehra descriptor offers a promising approach to face emotion recognition when combined with cutting-edge feature representation methods and complex classification frameworks. Through the combination of geometric methods and cutting-edge machine learning algorithms, scientists can advance emotion detection systems and make it easier for them to be used in practical settings.

### 2.2 Convolutional Neural Networks (CNN)

A key component of human-computer interaction is facial emotion recognition (FER), which has uses in everything from virtual reality to mental health diagnosis. In this field, convolutional neural networks, or CNNs, have become extremely effective instruments by providing notable improvements in efficiency and accuracy.

Researchers are always coming up with new ways to improve the accuracy and resilience of CNN architectures, which are essential to FER systems. FER tasks have led to adjustments and improvements made to early CNN models like as AlexNet, VGG, and ResNet. To improve feature extraction in FER, Gao et al. (2020) suggested modifying the ResNet model and adding attention methods to it. Similarly, a lightweight CNN architecture was presented by Liu et al. (2019) and tailored for real-time FER applications on devices with limited resources.

Several strategies are used when training CNNs for FER to improve generalization and model performance. The use of transfer learning, which involves fine-tuning pre-trained CNN models on FER datasets, has grown in popularity since it makes use of information from extensive image datasets. Zhang et al. (2021), for example, trained a CNN for FER using transfer learning from ImageNet, resulting in state-of-the-art performance on benchmark datasets. Rotation, scaling, and flipping are examples of data augmentation approaches that have been used to enhance training data and increase model robustness (Chang et al., 2019).

CNN-based FER systems are evaluated using measures including F1-score, accuracy, precision, and recall. For benchmarking, datasets such as CK+, MMI, and FER2013 are frequently utilized. According to recent research, CNN models routinely outperform conventional machine learning techniques, yielding astonishing outcomes. For instance, Li et al. (2022) used a deep CNN ensemble model to attain an accuracy of over 90% on the CK+

dataset.

To sum up, CNNs have transformed the field of facial emotion recognition by providing cutting-edge results on a wide range of datasets and applications. With cutting-edge training methods and ongoing CNN architecture development, even further gains in FER efficiency and accuracy are anticipated. Research is still being done on issues including managing occlusions, a variety of facial expressions, and practical implementation.

## 3. Comprehensive Software Development Overview

### 3.1 DataSet

For our project, we utilized the "FER-CK-KDEF" dataset, which combines the Facial Expression Recognition (FER), Cohn-Kanade (CK), and Karolinska Directed Emotional Faces (KDEF) datasets. This dataset is publicly available on Kaggle at the following URL: https://www.kaggle.com/datasets/sudarshanvaidya/corrective-reannotation-of-fer-ck-kdef.

The "FER-CK-KDEF" dataset consists of a vast collection of 32,900+ grayscale images, categorized into eight unique emotion classes: anger, contempt, disgust, fear, happiness, neutrality, sadness, and surprise. Each image contains a grayscale human face or sketch, ensuring consistency in the dataset's content. The images are standardized to 224 x 224 pixels and are stored in PNG format.

During the preprocessing stage, we resized all images to a uniform size of 48 x 48 pixels to facilitate model training and consistency. This resizing process ensures that all images have the same dimensions, thereby simplifying the training process and enhancing computational efficiency.

The richness of the "FER-CK-KDEF" dataset, combined with its large number of images and variety of emotion categories, provides a robust foundation for training and validating our emotion detection models. By using this dataset, we aimed to develop models that could accurately recognize and classify various facial expressions in real time, thereby enhancing the effectiveness and usability of our interactive visualization platform.

Integrating such a comprehensive dataset into our software development process enabled us to train models capable of accurately detecting a wide range of emotions, ensuring the reliability and performance of our final solution. Moreover, the availability of this dataset on Kaggle promotes transparency and reproducibility, allowing other researchers and developers to validate and build upon our work in the field of facial emotion recognition.

### 3.2 System Workflow

**Explanation of Process:**

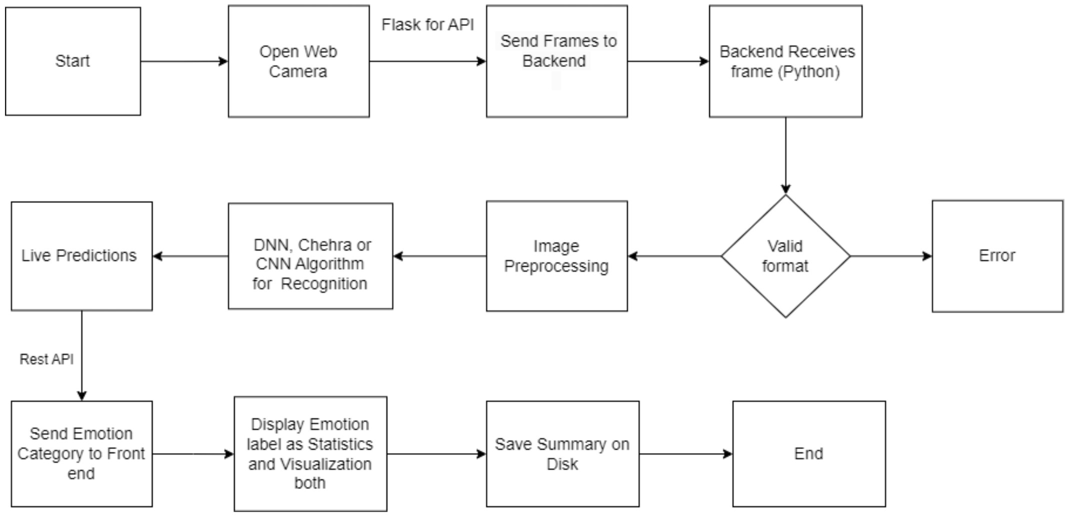



Figure 1: Facial Emotion Recognition System Workflow

The flowchart outlines the process of a facial emotion recognition system that operates in real-time, using a web camera and a backend server with machine learning capabilities.

**Start:** The process begins with the initialization of the system.

**Open Web Camera:** The system activates the web camera to capture live video frames.

**Flask for API:** The captured frames are sent to the backend server via an API developed using Flask, which is a micro web framework written in Python.

**Backend Receives frame (Python):** The backend server, running a Python script, receives the video frames for processing.

**Valid Format Check:** Upon receipt, the backend performs a check to ensure the frames are in a valid format. If the format is not valid, an error is raised and the process stops.

**Image Preprocessing:** Valid frames undergo preprocessing, which may include resizing, normalization, and other image processing techniques to prepare the data for emotion recognition.

**DNN, Chehra, or CNN Algorithm for Recognition:** The preprocessed images are then fed into one of the selected algorithms for facial emotion recognition: Deep Neural Network (DNN), Chehra Model, or Convolutional Neural Network (CNN).

**Live Predictions:** The algorithm processes the frames and provides live predictions of the emotions detected from the facial expressions.

**Display Emotion labels as Statistics and Visualization both:** The recognized emotion labels, along with relevant statistics and visualizations, are displayed on the front end for the user to see.

**Save Summary on Disk:** A summary of the emotion recognition results, along with any statistics and visualizations, is saved to the disk for future reference or analysis.

**Send Emotion Category to Front End (via Rest API):** Alongside real-time display, the emotion category detected is also sent back to the front end through a REST API, which allows for asynchronous data transfer between the server and the front end.

**End:** After saving the summary, the system process concludes.

Throughout the entire flow, the system relies on real-time data capture, processing, and machine learning algorithms to accurately identify and display facial emotions, providing insights into the user's emotional state. This system could be used in various applications, such as user experience research, psychological analysis, or interactive applications.

### 3.3 Unit Testing Table

The Unit Testing Table provides a detailed overview of the functional testing performed on various components of the facial emotion recognition system. It contrasts the expected outcomes with the actual results, highlighting the system's reliability and the areas requiring further optimization.

Table 1: Unit Testing Table

| **Testing Table** | **Component** | **Description** | **Expected Outcome** | **Actual Outcome** |
|---|---|---|---|---|
| UT01 | Live Streaming | Test input frames | Successful processed | Successful processed |
| UT02 | Model Detection | Test metrics of all algorithms | Good performance | Good accuracy |
| UT03 | Algorithms Testing | Integration with front end | Smooth operation | Minor lag observed between communication. |
| UT04 | Live Emotion Information | Test the graphs on front end | Fluctuate based on algorithm prediction | Successfully triggered |
| UT05 | Summary of Emotion Results | Test the emotion result folders | Saves into local disk with parameters | Perfectly saved |

### 3.4 Models Implementation

#### 3.4.1 Chehra Descriptor Approach

The Chehra descriptor strategy for facial emotion detection is implemented via a series of critical processes, all of which are intended to identify and utilize facial landmarks for precise emotion classification.

1. Facial Landmark Detection: Facial landmark detection is the first step in the procedure when 49 important areas on the human face are identified using the Chehra tool. This program uses a cascade of linear regressions based on discriminative facial deformable models to find face points automatically in real time, even in uncontrolled natural environments.

2. Feature Representation: Following the detection of the facial landmarks, a 98-dimensional feature vector is produced by representing the landmarks as Cartesian coordinates. However, another feature representation strategy is investigated because of the shortcomings of Cartesian coordinates in capturing various facial emotions.

3. Configural Feature Extraction: Configurable characteristics indicating intra-facial component distances are extracted to address the variety of facial shapes and enhance resilience. In doing so, the Euclidean distances between each facial landmark point are computed, yielding an 1176-dimensional feature vector that is more detailed.

4. Classification Stage: A densely connected neural network is used in the classification step once the retrieved feature vectors have been input into it. The mapping between facial features and associated emotional states is taught to this classifier by training on annotated datasets.

5. Evaluation and Validation: Finally, benchmark datasets are used to assess and validate the implemented model's performance. To evaluate how well the model recognizes facial expressions in a range of emotional states, metrics like recall, accuracy, precision, and F1-score are calculated.

#### 3.4.2 Convolutional Neural Network (CNN) Approach

On the other hand, a different methodology is used in the CNN approach to facial emotion recognition, which makes use of deep learning architectures to automatically learn discriminative features from unprocessed image data.

1. Data Preprocessing: To ensure consistency and make model training easier, the method starts with preparing the input image data. This includes scaling images to a common size (such as 224x224 pixels) and normalizing pixel values.

2. Architecture Design: Two distinct CNN architectures with various convolutional, pooling, and fully linked layer layers are created. With each layer picking up more abstract and sophisticated information, these layers are stacked to produce a hierarchical representation of facial traits.

3. Training Phase: Gradient descent optimization and backpropagation are used to train the created CNN model. By modifying its weights and biases in response to the discrepancy between the expected and ground truth labels, the model learns to minimize a predetermined loss function during training.

4. Fine-Tuning and Hyperparameter Tuning: To further maximize its performance, the trained CNN model may be subjected to hyperparameter and fine-tuning adjustments. To get the optimal outcomes, this entails modifying variables like learning rate, batch size, and network architecture.

5. Evaluation and Testing: Finally, a different test dataset is used to assess how well the trained CNN model performs. The model's accuracy in classifying facial expressions is evaluated using metrics including accuracy, precision, recall, and confusion matrices.

In summary, the CNN approach uses deep learning architectures to automatically learn features directly from raw image data, offering distinct advantages in facial emotion recognition tasks, whereas the Chehra descriptor approach depends on handcrafted feature extraction and conventional machine learning techniques.

## 3.5 Back-End and Front-End Development and Integration

### 3.5.1 Back-End Development

The back end of our facial emotion recognition system serves as the computational core, handling the heavy lifting of data processing and model inference. It was developed using Flask, a lightweight and flexible Python web framework that enables rapid development and straightforward integration with machine learning libraries. The back end is responsible for the following key functionalities:

- **Video Frame Processing:** Continuous capture and processing of live video frames from the web camera feed.

- **Emotion Recognition:** Utilization of the trained CNN, Chehra, and DNN models to perform real-time emotion detection on the preprocessed frames.

- **API Endpoints:** Creation of RESTful API endpoints to receive frame data from the front end and send back emotion detection results.

- **Data Storage:** Implementation of a system to save session summaries, including emotion statistics and user interactions, for further analysis.

Flask's ability to work well with other Python libraries allowed us to integrate TensorFlow and OpenCV seamlessly, facilitating model operations and image manipulations. The back-end was also optimized for performance, ensuring that the real-time processing demands of the system could be met efficiently.

### 3.5.2 Front-End Development

The front end of our application was crafted with the user experience in mind, offering a clean and intuitive interface for interaction with the emotion recognition system. We used a combination of HTML, CSS, and JavaScript to build a responsive design that adapts to various devices and screen sizes. The front end includes:

- **Live Video Stream:** An embedded video player that displays the live feed from the user's web camera.
- **Emotion Display:** Dynamic visualization of detected emotions, updating in real-time as the user interacts with the camera.
- **Graphical Feedback:** A bar chart graph that renders the intensity of detected emotions over time, offering users an analytical view of their emotional trends.
- **User Controls:** Interactive elements that allow users to start or stop the emotion detection process and view their session summaries.

The front end communicates with the back end via AJAX calls, ensuring a seamless and asynchronous data exchange. This decoupled architecture allows for independent scaling and maintenance of each part of the system.

### 3.5.3 Integration

The integration between the back-end and front-end was achieved through carefully designed API endpoints and data contracts. JSON was used as the data interchange format, providing a lightweight and language-independent method for data transfer. The integration process involved:

- **Data Flow Design:** Establishing a clear and efficient data flow between the front-end and back-end, ensuring that video frames and emotional data are transmitted accurately and promptly.
- **Synchronous Operations:** Synchronizing the frame capture and emotion detection processes to provide real-time feedback without noticeable lag.
- **Error Handling:** Implementing robust error handling to manage communication failures or processing errors, thereby enhancing system reliability.
- **Security Considerations:** Ensuring the privacy and security of the user's data through secure API design and adherence to best practices in web application security.

The successful integration of the back-end and front-end components resulted in a coherent system that allows users to experience the power of real-time facial emotion recognition in a user-friendly web application.

# 4. Results

## 4.1 Performance Metrics

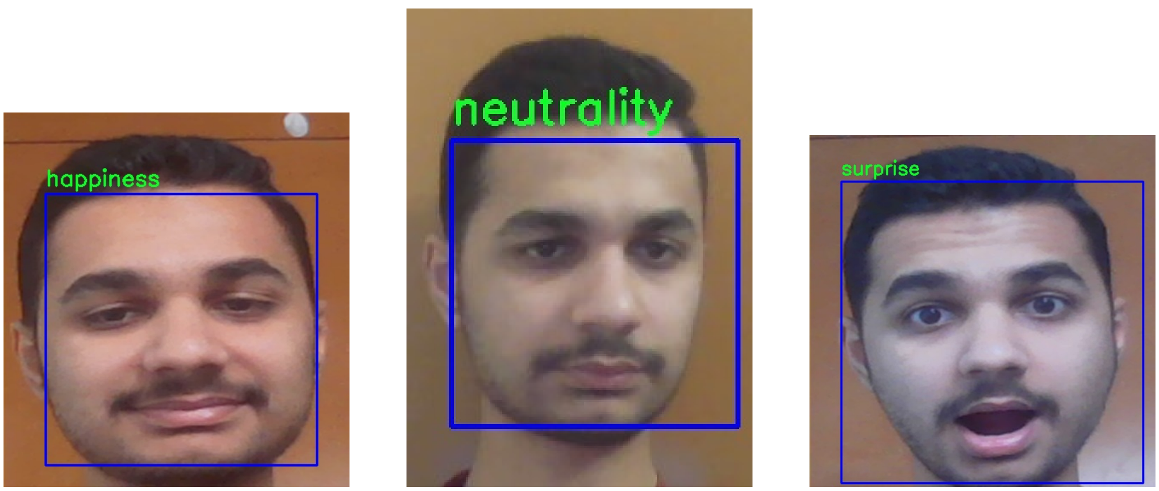
The table below summarizes the performance metrics of three distinct facial emotion recognition models. It highlights the accuracy, training duration (epochs), complexity (layers), and learning rates, offering insights into the effectiveness and efficiency of each model.

Table 2: Results and Performance Analysis

| **Model** | **Accuracy** | **Epochs** | **Layers** | **Learning Rate** |
|---|---|---|---|---|
| Convolutional Neural Network (CNN) | 81% | 100 | 4 | 0.001 |
| Chehra Model | 61.25% | 100 | 4 | 0.001 |
| Deep Neural Network (DNN) | 65% | 50 | 4 | 0.001 |

## 4.2 Outcomes and Observations

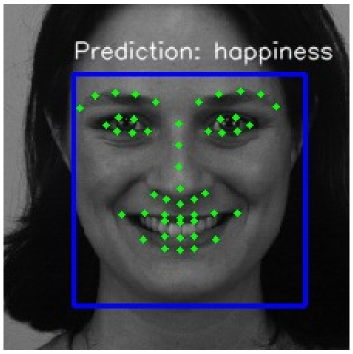
- The **Convolutional Neural Network (CNN)** exhibited the highest accuracy among the three models, achieving an accuracy of 81% after 100 epochs of training.



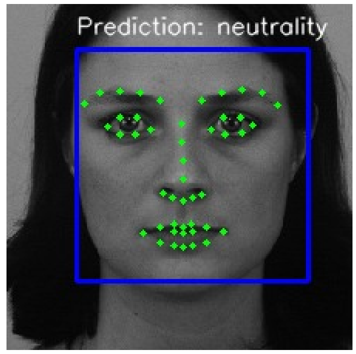

(a) Happy - CNN (b) Neutral - CNN (c) Surprised - CNN

Figure 2: Emotion predictions by the CNN model.

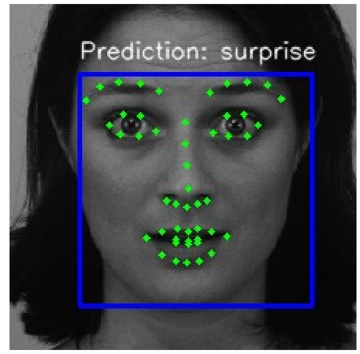
- The **Chehra Model**, despite utilizing a simpler architecture, yielded a lower accuracy of 61.25%, suggesting potential limitations in its ability to effectively capture the complexities of facial expressions.




(a) Happy - Chehra




(b) Neutral - Chehra

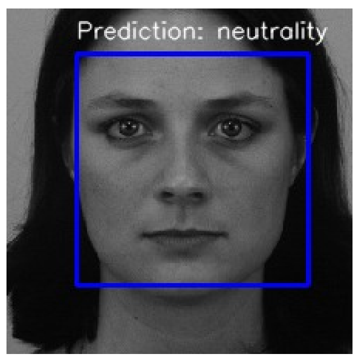



(c) Surprised - Chehra

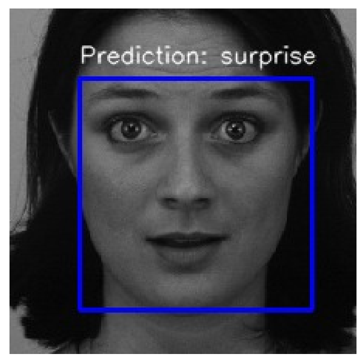
Figure 3: Emotion predictions by the Chehra model.

- The **Deep Neural Network (DNN)** performed moderately, with an accuracy of 65%, demonstrating competitive performance compared to the Chehra Model but falling short of the CNN's accuracy.


(a) Happy - DNN




(b) Neutral - DNN




(c) Surprised - DNN

Figure 4: Emotion predictions by the DNN model.

- All models were trained using a learning rate of 0.001 to facilitate convergence during training.
- The CNN's architecture included multiple convolutional and pooling layers, contributing to its ability to extract intricate features from facial images, leading to higher accuracy.
- In contrast, the Chehra Model and Deep Neural Network demonstrated simpler architectures, possibly limiting their capacity to discern subtle nuances in facial expressions, resulting in comparatively lower accuracies.
- Overall, the CNN model emerged as the most effective in capturing and classifying facial emotions accurately, underscoring the significance of employing deep learning techniques and intricate architectures in facial emotion recognition tasks.

## 4.3 Bugs and Resolution Table

Table 3: Bugs and Resolution Table

| **Bug ID** | **Description** | **Resolution** |
|---|---|---|
| B01 | Model Conversion, Accuracy Issue | Initially used TFJS for conversion, but performance dropped due to dependency issues. Moved to Flask service. |
| B02 | Inaccurate predictions for certain images | Adjusted model parameters and retrained |
| B03 | Overfitting challenges for models | Uses Dropout, Early stopping techniques etc |
| B04 | Storage issues of images | Uses Low dimensional image for training to cater Ram |
| B05 | Different variation images issues in testing | Implemented data augmentation techniques |

## 4.4 Visualization

The Visualization component is a crucial aspect of the Interactive Visualization Project, which provides a user-friendly and engaging representation of the emotion recognition results. This section of our web application displays the processed data through both numerical and graphical means, allowing users to gain quick and clear insights into the emotional breakdown of the facial expressions captured by the system.

### 4.4.1 Front-End View

The front-end interface, depicted in Figures 6 and 5, is designed to be intuitive and straightforward, allowing users of all technical backgrounds to interact with the system effectively. Key features of the front-end include:

- Real-time video feed with emotion detection overlay.
- Selection dropdown for choosing the desired emotion recognition model.
- Live updates of emotion recognition results.
- A "Generate Report" button for users to obtain a session summary.

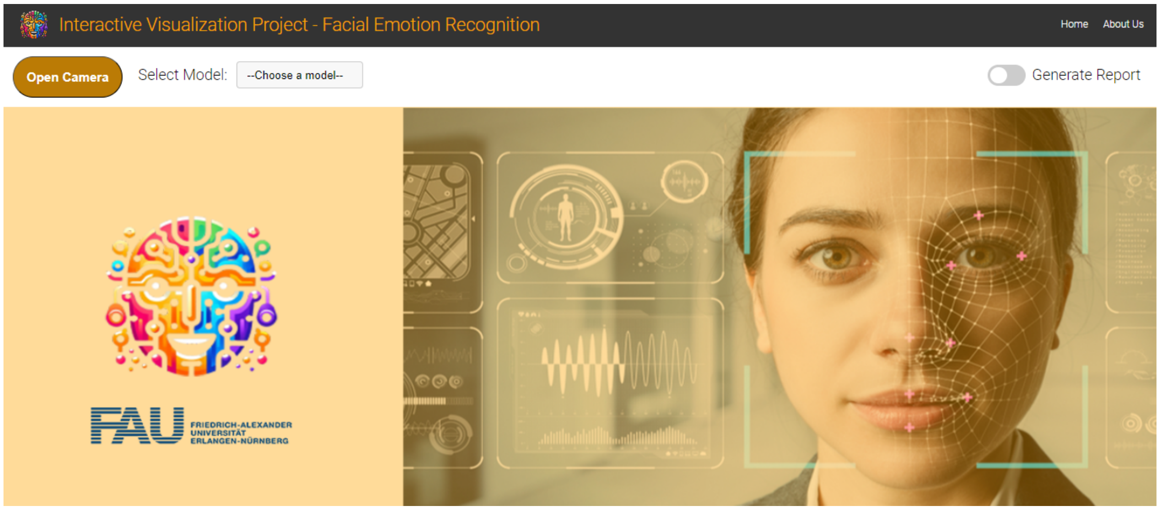



Figure 5: Interactive Front-End View: Emotion Recognition Model Selection

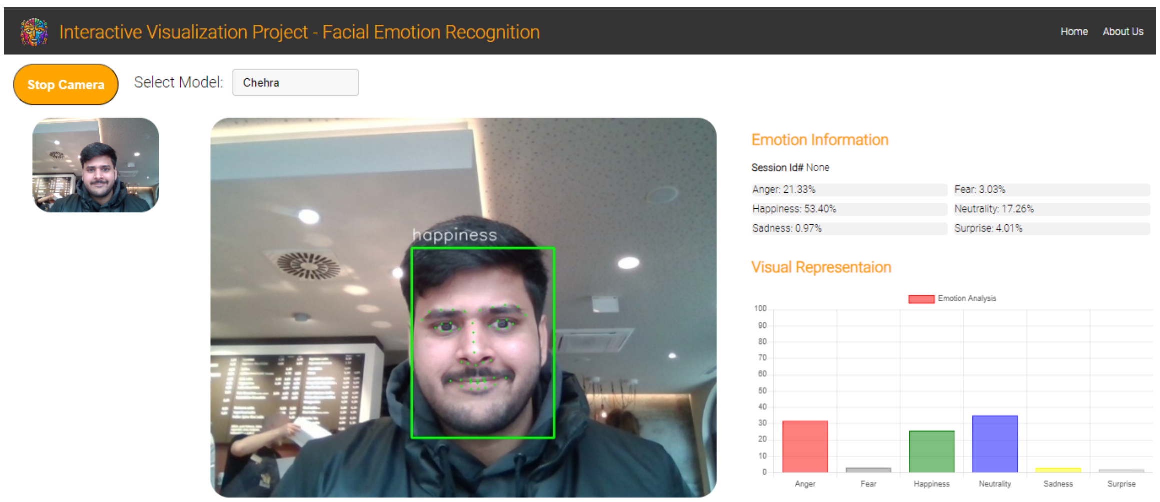


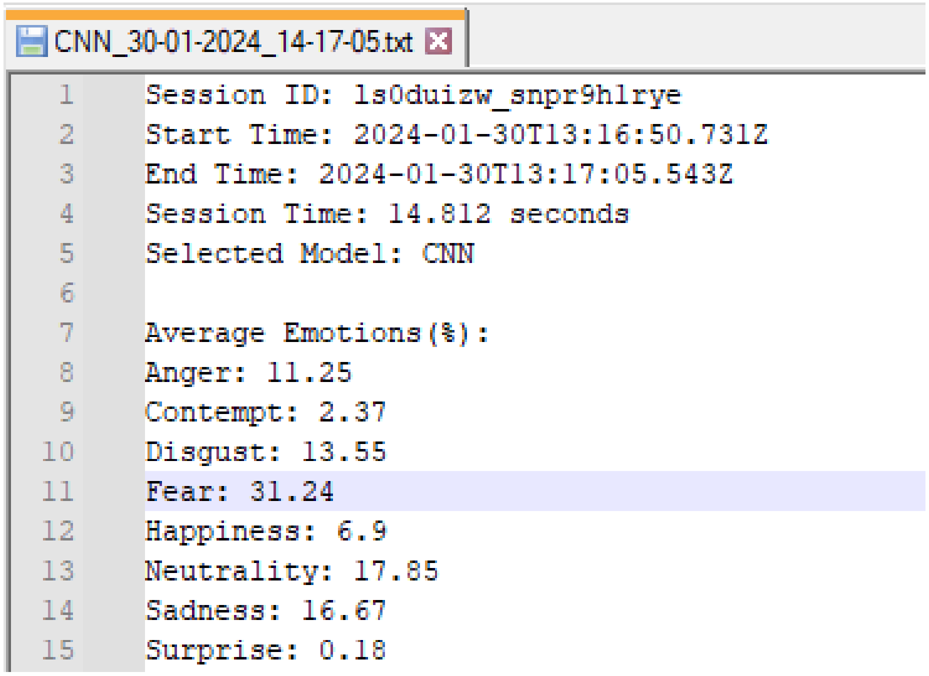

Figure 6: Interactive Front-End View: Real-time Emotion Detection

#### 4.4.2 Emotion Summary

After each session, the system generates a comprehensive summary of the emotions recognized during the session. This summary includes the following key details:

An example of such a summary is shown in Figure 7.

- **Session ID:** A unique identifier for the session, allowing users to reference or retrieve it at a later time.
- **Emotion Percentages:** A breakdown of the detected emotions expressed as percentages, giving a quick overview of the dominant emotions recognized in the session.



CNN_30-01-2024_14-17-05.txt

```
Session ID: ls0duizw_snpr9hlrye
Start Time: 2024-01-30T13:16:50.731Z
End Time: 2024-01-30T13:17:05.543Z
Session Time: 14.812 seconds
Selected Model: CNN

Average Emotions(%):
Anger: 11.25
Contempt: 2.37
Disgust: 13.55
Fear: 31.24
Happiness: 6.9
Neutrality: 17.85
Sadness: 16.67
Surprise: 0.18
```

Figure 7: Session Summary with Emotion Percentages

### 4.4.3 Graphical Representation

The emotional data are not only presented in a textual format but also visualized through an interactive bar graph. This graph displays the intensity of each emotion, offering a visual comparison among the different emotions detected. The graphical interface enhances user engagement and provides a more digestible format for understanding complex data patterns. An illustration of the graph can be seen in Figure 8 and a visual representation in Figure 9.

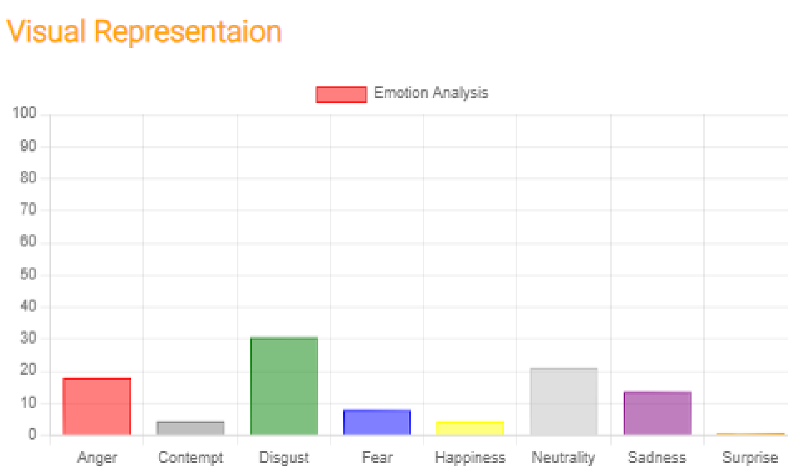



Figure 8: Graphical Representation of Emotion Analysis


Figure 9: Visual Representation of Emotion Analysis

### 4.5 Future Work

The current iteration of our facial emotion recognition system has laid a solid foundation for real-time emotion analysis. However, there remains a vast potential for further enhancements and developments. Future work will focus on several key areas to refine the system's performance and extend its capabilities:

- **Model Enhancement:** We aim to explore the integration of more advanced neural network architectures and learning algorithms to improve the accuracy and robustness of emotion recognition. This includes experimenting with larger datasets and implementing state-of-the-art techniques in deep learning.
- **User Experience Optimization:** Efforts will be made to refine the user interface and interaction design to provide a more intuitive and seamless experience. This includes personalizing the emotion feedback and enhancing the visualization dashboard.
- **Performance Tuning:** We plan to optimize the system for better performance, especially in terms of speed and resource utilization. This will ensure that the application can run smoothly on a wider range of devices, including those with limited processing power.
- **Real-World Testing:** To ensure the system's efficacy in practical scenarios, extensive field testing will be conducted. Feedback from these tests will inform further refinements to the system.
- **Privacy and Security:** As the system involves processing sensitive personal data, we will prioritize the development of robust security measures to protect user privacy. This includes implementing data encryption and secure data storage solutions.
- **Cross-Platform Compatibility:** Future developments will aim to make the system compatible across different platforms and devices, expanding its accessibility and ease of use.
- **Diverse Application Use Cases:** We intend to explore and establish partnerships for deploying our system across various domains such as healthcare, online education, and customer service to leverage the benefits of emotion recognition in diverse settings.

In addition to these specific goals, ongoing research will be devoted to understanding the ethical implications of emotion recognition technology and ensuring that its application respects user consent and societal norms. Our commitment is to advance the field of affective computing while upholding the highest standards of ethical responsibility.

## 5. Conclusion

In conclusion, both the Chehra descriptor approach and Convolutional Neural Network (CNN) approach present compelling methodologies for facial emotion recognition. Using geometric-based approaches, the Chehra descriptor method extracts features from facial landmarks in an organized manner. It offers a strong framework for categorization by using configurable distances to represent face expressions. CNNs, on the other hand, benefit from deep learning's capacity to automatically identify discriminative features from unprocessed picture data. Their hierarchical feature extraction method achieves state-of-the-art performance in emotion detection tests by enabling subtle pattern recognition. CNNs perform better with large-scale picture datasets, whereas the Chehra descriptor excels in interpretability and feature engineering. The decision between these two approaches depends on factors like processing resources and dataset characteristics.

Forward-looking, combining the advantages of CNN and the Chehra descriptor techniques shows the potential to improve facial emotion recognition. Through the integration of deep learning capabilities with geometric-based feature extraction, researchers may effectively address the limitations of each approach and improve overall accuracy. These developments are critical to applications in emotional computing, mental health monitoring, and human-computer interaction. These approaches will probably become more and more important as technology develops in interpreting human emotions and improving human-machine interactions, leading to a better comprehension of human behavior and emotions.

## References


[1] Corrective Re-annotation of FER-CK-KDEF. *Sudarshan Vaidya (2020).* The dataset is available at: `https://www.kaggle.com/datasets/sudarshanvaidya/corrective-reannotation-of-fer-ck-kdef`

[2] "Affective State Detection via Facial Expression Analysis within a Human–Computer Interaction Context," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 2175–2184, 2019, doi: 10.1007/s12652-017-0636-8.

[3] Corneanu C.A., Simon M.O., Cohn J.F., Guerrero S.E. (2016) "Survey on RGB, 3D, Thermal, and Multimodal Approaches for Facial Expression Recognition: History, Trends, and Affect-Related Applications," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 8, pp 1548-1568.

[4] Lopes A.T., de Aguiar E., De Souza A.F., Oliveira-Santos T. (2017) "Facial Expression Recognition with Convolutional Neural Networks: Coping with Few Data and the Training Sample Order," Pattern Recognition, vol. 61, pp 610-628.

[5] Koelstra S., Muhl C., Soleymani M., Lee J-S., Yazdani A., Ebrahimi T., Pun T., Nijholt A., Patras I. (2012) "DEAP: A Database for Emotion Analysis; Using Physiological Signals," IEEE Transactions on Affective Computing, vol. 3, no. 1, pp 18-31.

[6] FER Dataset, *Challenges in Representation Learning: Facial Expression Recognition Challenge*, Available at: `https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge`

[7] CK+ Dataset, *Facial Expression Recognition on CK+ dataset*, Hosted by WuJie1010 on GitHub, Available at: `https://github.com/WuJie1010/Facial-Expression-Recognition.Pytorch/tree/master/CK%2B48`

[8] KDEF Dataset, *Karolinska Directed Emotional Faces*, Available at: `https://www.kdef.se/download-2/register.html`

[9] "Python Documentation." Python Software Foundation. Available at: `https://docs.python.org/3/`

[10] "OpenCV Documentation." OpenCV. Available at: `https://docs.opencv.org/master/`

[11] "DLIB Library." Davis E. King. Available at: `http://dlib.net/`

[12] "Itertools — Functions creating iterators for efficient looping." Python Software Foundation. Available at: `https://docs.python.org/3/library/itertools.html`

[13] McKinney, W. (2010). "Data Structures for Statistical Computing in Python." In *Proceedings of the 9th Python in Science Conference*, pp. 51-56. Available at: `https://pandas.pydata.org/`

[14] Pedregosa et al. (2011). "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, 12, pp. 2825-2830. Available at: `https://scikit-learn.org/`

[15] Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). "Array programming with NumPy." *Nature*, 585(7825), 357-362. Available at: `https://numpy.org/`

[16] Chollet, F., et al. (2015). "Keras." Available at: `https://keras.io/`

[17] Cortes, C., & Vapnik, V. (1995). "Support-Vector Networks." *Machine Learning*, 20(3), 273-297.

[18] Breiman, L. (1996). "Bagging Predictors." *Machine Learning*, 24(2), 123-140.

[19] Jolliffe, I.T. (2002). "Principal Component Analysis and Factor Analysis." In *Principal Component Analysis*, Springer Series in Statistics, 115-128.