# **INTRODUCTION** This project demonstrates the use of machine learning techniques including both supervised and unsupervised learning methods. The aim is to identify human emotions from facial expression using Convolutional neural networks (CNN) for classification and K means clustering with Principal component analysis (PCA) for emotion grouping. With this advancement of artificial intelligence, automatic recognition has improved multiple sectors such as healthcare, surveillance and education fields building a system capable of analysing comparing, detecting and classifying different Facial emotions Combining deep learning knowledge for accurate recognition and clustering for better structure of data. The core goal of developing a unified application capable of detecting and grouping emotion expression using two such as TensorFlow, keras, open CV, and Sckit- learn to implement and evaluate the models. Supervised learning models on this project has been trained on FER2013 and CK48+ For this project performing a validation accuracy of 56% over 10 epochs. The set of images structured into different emotions label in to train and test directories to allow easier manipulation with open CV an efficiency in training using PIXELS grayscale images 48X48 Classify into happy, sad, angry, discussed, here, surprised to permit real time detection with webcam based on emotion. **Dataset Selection and Preprocessing**

In this project, open-source Facial expression data set from Kaggle (FER2013 & CK48+) containing 48x48 pixels images categorized in seven emotions: Angry, discussed, Fear, sad, surprised, and neutral. This image is for the being converted into image folder format Was used to simplify preprocessing. Including main directories: train/ and test/ each containing subfolders for each emotion, to easily support both training CNN model I'm clustering the images. Preprocessing steps included:  
  
**Greyscale normalisation** changing the picture value scale to [0, 1] range.  
**Image resizing** to 48x48 for consistency   
**Data augmentation** in CNN training to increase viability  
**Flattening images and PCA** for image to be reduced to 2D using PC a unsupervised clustering to ensure a uniform input for both supervised and unsupervised pipelines.  
  
**Machine Learning Methods implementation**  
  
 Supervised machine learning implementation  
 This model CNN implementation in tensorflow and keras including convolutional layers, Max pooling dropout for regularisation and a dense layer connected to be trained to classify images into one of this seven emotion categories. As key aspect of this supervise model, input size of 48x48 grayscale images was used with the activation of functions ReLU for hidden layers and softMax for output with loss function categorical cross entropy which led to a promising accuracy in real time detection when integrated with open CV and haar cascade for face detection and trained model.  
  
**Key configurations:**

**Loss function:** Categorical Crossentropy

**Optimizer:** Adam

**Activation:** ReLU (hidden), Softmax (output)

**Output classes:** 7 emotions  
  
 Unsupervised machine learning implementation  
   
To the contrast of the supervised model, this implemented strategy using k-means clustering to flatten and normalise the images using principal component analysis (PCA) to reduce dimensionality to two components of visualisation. Following some key steps such as image loading and flattening from data set, clustering into 7 clusters coma scatter plot visualisation showing group emotion based on pick said intensity patterns. Although unsupervised learning cannot cluster directly, visualisation revealed this grouping that roughly correspond to the different emotions categories.  
  
 All these represented with a great and suitable graphical user interface design and function ability using Tkinter to provide a user friendly application to be able to demonstrate its capability which help launch supervised that's real camera webcam emotion detection also launched on supervised display PCA clustered Facial image  
 comparing models side by side to provide output of comparison from supervised and unsupervised approach.  
  
Key configurations:

**Dimensionality reduction:** PCA (n=2)

**Clustering:** KMeans (k=7)

**Evaluation:** Cluster visualization, manual emotion matching  
  
  
**ML System Design and Development**  
  
 Machine learning system design and development used for this project that support every time emotion recognition using the camera an unsupervised emotion clustering includes:  
  
 **Architectural overview**  
  
Input Layer: Webcam (real-time) or dataset images

Preprocessing: Grayscale conversion, resizing, normalization

**Supervised Pathway:**

Face detection using Haar Cascades (OpenCV)

Emotion classification via trained CNN model (emotion\_recognition\_model.h5)

Display emotion predictions in real-time via OpenCV window

**Unsupervised Pathway:**

Load dataset from folders

Convert images to 48x48 grayscale and flatten

Use PCA to reduce dimensionality

Apply KMeans clustering to organize images into 7 clusters

Plot the PCA-reduced data points with colors representing different emotions

**Graphical User Interface (GUI):**

A Tkinter-based interface allows the user to:

Launch Supervised Mode (real-time webcam classification)

Launch Unsupervised Mode (clustering visualization)

**System Tools Used:**

TensorFlow/Keras for CNN architecture

Scikit-learn for PCA and KMeans

OpenCV for camera access and face detection

Matplotlib for cluster visualization

Tkinter for GUI controls

This design ensures scalability, testability, and visual interpretability while maintaining user-friendly usability for non-technical users, Launch Comparison View (evaluate results from both pipelines)  
A diagram of a pipeline

AI-generated content may be incorrect.  
  
**Result and Discussion**

The results from both supervised and unsupervised provided valuable insights for each approach. The CNN model reaches an accuracy of approximately 83% on both FER2013 & CK48+ which demonstrate the model high precision and recall in recognising emotion such as happy, and surprise due to the distinct facial patterns. Emotions like disgust and fear mostly turned out to be misclassified reflecting the inherent similarity between the expressions. Applying the clustering K means + PCA, result into scattered plot displayed visually separable clusters. (k=7) where each cluster was manually associated to an emotion based on the observation of representative image within that cluster. This jewel model setup made the project more complete and visually interpretable providing a much more deeper understanding and more interactive user experience.

**A graph of a graph with blue and orange lines

AI-generated content may be incorrect.**

This graph of CNN model accuracy surges a learning relevant feature without overfitting. From the graph above, the epochs shows that the training set is having a large amount of intra class variation making it harder to fit. Relatively small number of an epoch so the model hasn't overfitted yet. More conventional layers could be tested do too and upward consistent trajectory of both curves indicating healthy convergence however to improve the result, hyperparameters tuning could be a solution such as changing learning rate to yield better generalisation, using a large all pre augmented data set could boost the performance especially for classes like disgust. The CNN model was trained over 10 epochs using grayscale official emotion images of size 48X48 from data set to classify Fisher expression into seven emotion that is happy, disgust, angry, fear, sad and neutral, surprise. This performance during the training track by accuracy metrics on both training and validation data set.

F1-score: varied by emotion (best: happy, weakest: disgust)

**Validation included:**

Manual webcam testing

Cross-checking with known emotion samples

Inspecting cluster purity (KMeans)

**Conclusion and Future Work**  
  
 Several challenges were encountered during the development of this project such as  
 database format which was in CSV format end required to be converted into image version so that it can easily be manipulated also, model overfitting was faced with regularisation techniques like dropout used to prevent CNN over fitting. Webcam bees realtor recognition required permission for it to be able to execute normally and obey the ethics of AI, image errors which fail to load, dimension issues importation and integration of the data set into the right path document to be able to easily read by the computer, solved by converting images from CSV to JPEG and PNG using AI, moving image folder into the right workspace folder, resizing the images to the right format to ease readability by the system and ensure proper execution.  
  
Improvements and Future work  
  
Important improvement on the long run use to improve current project capacities and extend its capabilities will include factors such as   
  
**Emotion History Tracking :** Which will provide the users with recent emotions over time to observe their emotion trends   
Comparison dashboard : Which will display predictions from both models side-by-side with evaluation metrics at real-time.  
  
**Model tuning :**To enhance the accuracy exploring advanced architectures like ResNet   
  
**Larger dataset :** Providing more diverse datasets to trail the model for a better improved generalization.  
 **Emotional intensity :**Predicting not just the type but the emotion intensity (e.g. Slightly angry vs very angry )

# **References**

1. Mollahosseini, A., Hasani, B. and Mahoor, M.H., 2016. AffectNet: A database for facial expression, valence, and arousal computing in the wild. IEEE Transactions on Affective Computing.
2. Kaggle. FER2013 Dataset. [https://www.kaggle.com/datasets/msambare/fer2013]
3. OpenCV. Haarcascade Documentation. [https://docs.opencv.org/3.4/db/d28/tutorial\_cascade\_classifier.html]

# **Appendices**

* Sample code snippets (see GitHub)
* Cluster plot visualization screenshots
* Link to GitHub project: [https://github.com/ByEmG/F.E-version-2-]
* Link to video demo: [To be added before submission]