FINAL PROJECT

NAAN MUDHALVAN

VELMURUGAN S
2021503317

TITLE: FACIAL EMOTION RECOGNITION USING CNN

PROBLEM STATEMENT:

Facial emotion recognition using CNN aims to develop a system that can accurately identify emotions such as happiness, sadness, anger, and others from images of human faces. This involves training a convolutional neural network (CNN) to learn patterns and features in facial expressions, enabling it to classify emotions correctly. The goal is to create a robust and reliable model that can interpret facial cues and provide accurate emotion labels in real-time applications like virtual assistants, customer service bots, and psychological research.

NOVELTY:

Hybrid Data Augmentation : Combine traditional data augmentation techniques with advanced ones like Generative Adversarial Networks (GANs) to create synthetic data, increasing the robustness and diversity of training data.

Multi-Task Learning: Design a model that simultaneously learns emotion recognition and other related tasks

Custom CNN Architecture: Create a unique CNN architecture that outperforms traditional models in terms of accuracy or efficiency.

Emotion Recognition in Complex Environments: Develop a system that can accurately detect emotions in real-world scenarios, like crowded streets or dynamic lighting conditions, adding robustness to the model.

MODELLING:

Step 1: Data Collection and Preprocessing:

Dataset Selection: Use a dataset designed for facial emotion recognition, such as FER-2013, AffectNet, or CK+. Ensure the dataset contains a variety of emotions and is balanced across classes.

Data Augmentation: Implement data augmentation techniques to increase the diversity of training samples and reduce overfitting. Common techniques include rotation, flipping, scaling, and color jitter.

Normalization and Scaling: Normalize pixel values (e.g., between 0 and 1) and ensure consistent scaling and resizing of images. This helps with training stability and consistency.

Step 2: Model Architecture:

Convolutional Layers: Start with a few convolutional layers to extract features from the images. Consider using varying kernel sizes to capture different levels of detail.

Dropout and Regularization: Add dropout layers to prevent overfitting and ensure robustness. You can also use L2 regularization for additional regularization.

Dense Layers: After feature extraction, add dense (fully connected) layers to aggregate the extracted features and prepare for classification.

Output Layer: The output layer should correspond to the number of emotion classes, typically with a softmax activation function for multi-class classification.

Step 3: Model Training

Train-Test Split: Split your dataset into training and validation sets. A common split is 80% for training and 20% for validation.

Batch Size and Epochs: Choose an appropriate batch size and number of epochs. A typical batch size is between 32 and 128, and the number of epochs depends on convergence.

Learning Rate and Optimizer: Use a learning rate scheduler or early stopping to adjust learning as training progresses. The Adam optimizer is a common choice for CNNs.

Model Fit: Train the model with the specified parameters, using the training data

Step 4: Model Evaluation

Validation Set Performance: Use the validation set to evaluate the model's performance and ensure there's no overfitting.

Accuracy and Confusion Matrix: Assess the model's accuracy and create a confusion matrix to understand how well the model distinguishes between different emotions.

Generalization and Testing: Test the model on an unseen test set to ensure generalization. Use metrics like precision, recall, and F1-score to evaluate overall performance.

PROPOSED WORK:

- Use a publicly available dataset like FER-2013, AffectNet, or CK+, each containing labeled facial emotion images.
- The dataset should encompass various emotions, age groups, and ethnicities.
- Normalize pixel values (e.g., to [0, 1] or [-1, 1]) and ensure consistent image sizes for all input images.

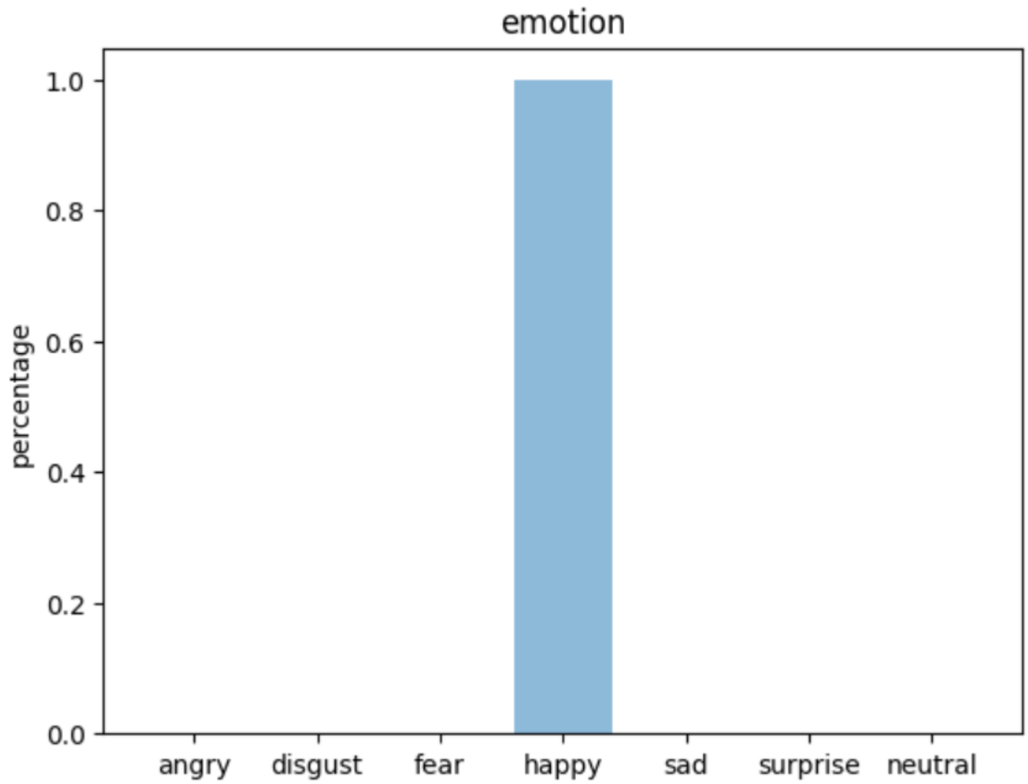
- Build a sequence of convolutional layers with varying kernel sizes and activation functions (e.g., ReLU) to extract key features from facial images.
- Split the data into training, validation, and test sets (commonly 80% training, 10% validation, and 10% test).
- Define suitable batch sizes (commonly 32, 64, or 128) and train for a specified number of epochs.
- Evaluate the model's performance using metrics like accuracy, precision, recall, and F1-score.
- Ensure that data used for training is collected ethically and that personal information is kept confidential.
- Evaluate the model for potential bias across different demographic groups and ensure fairness in emotion classification.

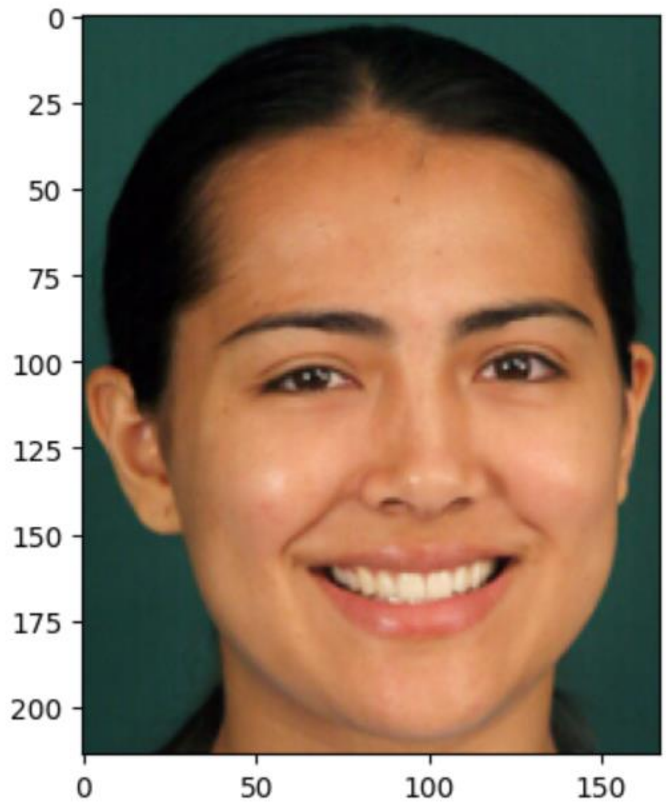
DATASET:

RESULT:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 46, 46, 32)        320

 conv2d_1 (Conv2D)           (None, 44, 44, 64)        18496

 max_pooling2d (MaxPooling2  (None, 22, 22, 64)        0
 D)

 dropout (Dropout)           (None, 22, 22, 64)        0

 conv2d_2 (Conv2D)           (None, 20, 20, 128)       73856

 max_pooling2d_1 (MaxPoolin  (None, 10, 10, 128)       0
 g2D)

 conv2d_3 (Conv2D)           (None, 8, 8, 128)         147584

 max_pooling2d_2 (MaxPoolin  (None, 4, 4, 128)         0
 g2D)


 dropout_1 (Dropout)         (None, 4, 4, 128)         0

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 1024)              2098176

 dropout_2 (Dropout)         (None, 1024)              0

 dense_1 (Dense)             (None, 7)                 7175

=================================================================
Total params: 2345607 (8.95 MB)
Trainable params: 2345607 (8.95 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
1/1 [==============================] - 0s 244ms/step
```





Expression Prediction: Happy