SCHOOL OF ENGINEERING

Dayananda Sagar University

Digital Image processign Project

Face Emotion Recognition

Team Members:

Spandana BV-ENG22CS0181

Shreya N-ENG22CS0165

Agenda

1. Introduction
2. Abstract
3. Methodology
4. Results
5. Conclusion

Introduction

- Face Emotion Recognition (FER) refers to the process of identifying human emotions from facial expressions using computer vision and machine learning.
- Emotions such as happiness, sadness, anger, surprise, fear, and disgust can be detected from facial cues.
- Applications in healthcare (detecting depression or anxiety), security, human-computer interaction, customer service, etc.

Problem

Why Emotion Detection?

- Enhances human-computer interaction.
- Supports mental health applications by detecting emotional states.
- Improves customer service with sentiment analysis.

Challenges:

- Variability in facial expressions.
- Real-time processing requirements.
- Dataset quality and diversity.

Abstract

- The Face Emotion Recognition System leverages computer vision and deep learning techniques to automatically classify emotions based on facial expressions in real-time.
- The system uses a dataset of labeled facial images to train a model that can predict emotions with high accuracy.
- The system uses a dataset of labeled facial images to train a model that can predict emotions with high accuracy.
- Real-time emotion prediction with low computational latency.

Methodology

Step 1:
Data Collection

- Dataset: We are using a dataset that is available on Dropbox via the link: https://www.dropbox.com/s/w3zlhing4dkgeyb/train.zip?dl=0
- This appears to be a zipped file, likely containing training images for facial emotion recognition.
- You can download it using wget, but the URL needs to be corrected to enable direct downloading.

Preprocessing:

- Resized images to 48x48 pixels.
- Normalized pixel values.
- Applied data augmentation for diversity.
    -

Step 2: Model Architecture

- CNN Model: The system uses a Convolutional Neural Network (CNN) to extract hierarchical features from facial images.
- Model Type:
- Convolutional Neural Network (CNN)
- Architecture:
- Input Layer: 48x48 grayscale image.
- Convolutional Layers: Extract features using filters.
- Pooling Layers: Downsample feature maps.
- Fully Connected Layers: Perform classification.
- Output Layer: 7 softmax neurons for emotion categories.
- Tools:
- TensorFlow/Keras for model development.
- OpenCV for image handling.

Smoothing and Sharpening for Emotion Recognition

1. Smoothing (Blurring)
   - Purpose: Reduces noise and irrelevant details.
   - Method: Apply filters to blur the image and enhance major features.
   - Example: Gaussian Blur:

     ```
     blurred_image = cv2.GaussianBlur(image, (5, 5), 0)
     ```
   - Benefit: Focuses on significant facial structures.

Smoothing and Sharpening for Emotion Recognition

2. Sharpening

- Purpose: Enhances facial details by emphasizing edges.
- Method: Apply a filter to amplify high-frequency components (edges).
- Example: Sharpening with a custom kernel:

  kernel = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])
- sharpened_image = cv2.filter2D(blurred_image, -1, kernel)
- Benefit: Highlights facial features like eyes, nose, and mouth.

Smoothing and Sharpening for Emotion Recognition

3. Integration in Emotion Recognition

- Process:
    a. Preprocess: Convert to grayscale, apply smoothing.
    b. Sharpen the image for clearer details.
    c. Detect faces and predict emotions using a trained model.
- Outcome: Improved emotion classification accuracy.

Step 3: Training

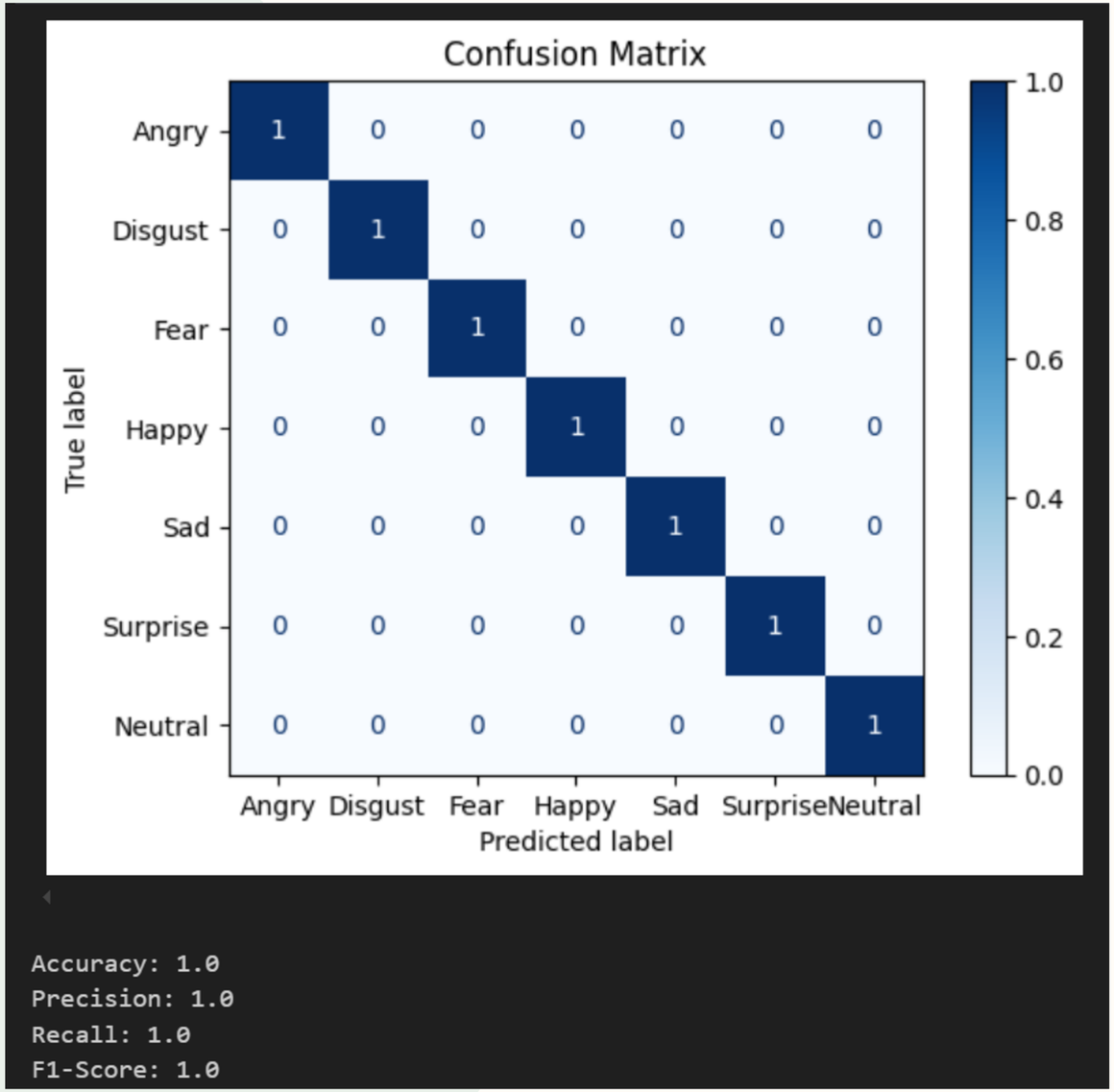
- Loss Function: Cross-entropy loss for multi-class classification.
- Optimization: Adam optimizer for training the model efficiently.
- Data Augmentation: Techniques like rotation, scaling, and flipping to improve model robustness.

Step 4: Evaluation

- Performance metrics: Accuracy, Precision, Recall, and F1-score.
- Cross-validation for model robustness.

Results

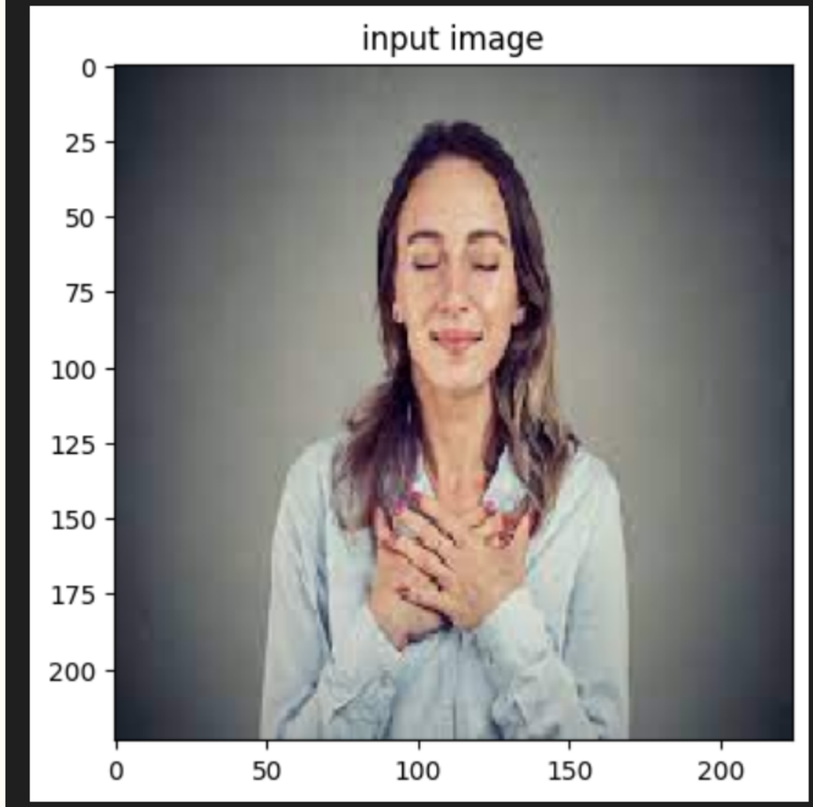
- Evaluation Metrics:



Visuals:

```
path = "/content/download.jpg"
img = load_img(path, target_size=(224,224) )

i = img_to_array(img)/255
input_arr = np.array([i])
input_arr.shape

pred = np.argmax(model.predict(input_arr))

print(f" the image is of {op[pred]}")

# to display the image
plt.imshow(input_arr[0])
plt.title("input image")
plt.show()
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 796ms/step
 the image is of Happy
```



Conclusion

Summary:

- Emotion detection systems bridge the gap between humans and machines by enabling empathetic interactions.
- This project demonstrates the feasibility of emotion recognition using CNNs.

Final Thoughts:

- The technology has vast potential in mental health, education, and entertainment.

References

- Sklearn Metrics Documentation

Provides detailed descriptions and examples of the accuracy_score, precision_score, recall_score, f1_score, and confusion_matrix functions.

- FER2013 Dataset

Learn more about the dataset commonly used for emotion detection, including class labels and dataset size.

- TensorFlow/Keras Framework

Official guides on training and evaluating machine learning models, including using confusion matrices and metrics tracking.

- Research Papers

Goodfellow et al. (2013). Challenges in Representation Learning: Facial Expression Recognition Challenge.
Describes the FER2013 dataset and its benchmarks
https://paperswithcode.com/dataset/fer2013

Thank you