**<FACE EMOTION DETECTION>**

**Submitted for**

**Statistical Machine Learning CSET211**

Submitted by:

**E23CSEU1648 DIPANSHU SINGH**

**E23CSEU1623 ANIRUDDH MISHRA**

Submitted to

**DR. SHAKSHI SHARMA**

**July-Dec 2024**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

A close-up of a logo

Description automatically generated

ABSTRACT

This project focuses on face emotion detection by developing and comparing two models: a Convolutional Neural Network (CNN) and a Support Vector Machine (SVM). The objective was to classify facial expressions into seven categories: 'angry', 'disgust', 'fear', 'happy', 'neutral', 'sad', and 'surprise'. The CNN was constructed using TensorFlow and Keras, with Pandas, NumPy, and OpenCV aiding in data handling and preprocessing. Jupyter Notebook provided an interactive environment, and tqdm was used for progress tracking.

Challenges included ensuring sufficient data preprocessing and managing the computational intensity of training the CNN. Problems such as overfitting and balancing dataset variability were mitigated through image augmentation and dropout techniques. The SVM model, while simpler and faster to train, faced difficulties in handling complex spatial features compared to the CNN.

Statistical comparisons revealed that the CNN outperformed the SVM, achieving higher accuracy and better precision, recall, and F1-score. Specifically, the CNN achieved an accuracy of approximately X%, whereas the SVM achieved Y%, showcasing the deeper network’s capability to learn intricate image features more effectively.

Overall, this project highlights the importance of model architecture choice and preprocessing steps in emotion detection tasks.

INTRODUCTION

This project focuses on face emotion detection by developing and comparing two models: a Convolutional Neural Network (CNN) and a Support Vector Machine (SVM). The objective was to classify facial expressions into seven categories: 'angry', 'disgust', 'fear', 'happy', 'neutral', 'sad', and 'surprise'. The CNN was constructed using TensorFlow and Keras, with Pandas, NumPy, and OpenCV aiding in data handling and preprocessing. Jupyter Notebook provided an interactive environment, and tqdm was used for progress tracking.

Challenges included ensuring sufficient data preprocessing and managing the computational intensity of training the CNN. Problems such as overfitting and balancing dataset variability were mitigated through image augmentation and dropout techniques. The SVM model, while simpler and faster to train, faced difficulties in handling complex spatial features compared to the CNN.

Statistical comparisons revealed that the CNN outperformed the SVM, achieving higher accuracy and better precision, recall, and F1-score. Specifically, the CNN achieved an accuracy of approximately X%, whereas the SVM achieved Y%, showcasing the deeper network’s capability to learn intricate image features more effectively.

Overall, this project highlights the importance of model architecture choice and preprocessing steps in emotion detection tasks.

**Introduction**

**Problem Statement** The primary challenge in emotion recognition from facial expressions lies in accurately identifying subtle differences across various emotions in images. Detecting emotions such as 'disgust' or 'fear' requires sophisticated feature extraction and robust model training due to their nuanced nature and possible overlaps in expression features.

**About the Dataset** This project utilized the Kaggle dataset "Face Expression Recognition Dataset," available at [Kaggle](https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset). The dataset includes thousands of labeled images representing the seven targeted emotions, captured under different lighting, angles, and facial orientations to enhance variability and model training efficacy.

**Summary of the Dataset** The dataset consists of grayscale images with a resolution of 48x48 pixels, structured into training, validation, and testing subsets. Each image is annotated with a label corresponding to one of the seven emotion categories. This diverse collection supports thorough training and helps prevent overfitting by presenting the model with varied examples for each emotion.

METHODOLOGY

The methodology for building a face emotion detection model using Convolutional Neural Networks (CNN) involves several key steps, which include data collection, preprocessing, model development, training, and evaluation. The following is the detailed approach:

1. **Data Collection**:  
   The dataset used for training the model is the FER-2013 dataset, which contains labeled images of human faces expressing different emotions. The dataset includes seven emotions: 'angry', 'disgust', 'fear', 'happy', 'neutral', 'sad', and 'surprise'.
2. **Data Preprocessing**:
   * **Image Resizing**: Images are resized to a uniform dimension (e.g., 48x48 pixels) to ensure consistency across the dataset.
   * **Normalization**: Pixel values are normalized to the range [0, 1] by dividing by 255. This helps in faster and more efficient training.
   * **Data Augmentation**: Techniques such as rotation, flipping, and zooming are used to artificially increase the size of the training dataset and improve the model's generalization capability.
3. **Model Architecture**:  
   The core of the emotion detection model is based on a **Convolutional Neural Network (CNN)**. The architecture includes several layers:
   * **Convolutional Layers**: To extract features such as edges, textures, and shapes from the face images.
   * **Max-Pooling Layers**: To downsample the feature maps and retain the most important information.
   * **Fully Connected Layers**: To make the final classification based on the features extracted by the CNN.
   * **Dropout Layers**: To prevent overfitting by randomly setting a fraction of the input units to zero during training.
4. **Model Training**:  
   The model is trained using a **cross-entropy loss function** for multi-class classification. The optimizer used is **Adam**, which adapts the learning rate during training to speed up convergence. Training takes place on a training set, and the validation set is used to tune hyperparameters and check the model's performance on unseen data.
5. **Model Evaluation**:  
   After training, the model is evaluated using the test set. Common evaluation metrics include:
   * **Accuracy**: The percentage of correct predictions.
   * **Confusion Matrix**: To visualize how well the model performs for each emotion class.
   * **F1-Score**: To assess the balance between precision and recall.
6. **Deployment**:  
   Once the model is trained and evaluated, it is deployed using a Python-based application that utilizes **OpenCV** to capture real-time video from a webcam. The model predicts the emotion of a person based on the facial expression in the frame.

RELATED WORK

**AjayK47/Face-Emotion-Detection\_CNN:**

* **Summary: This project involves training a Convolutional Neural Network (CNN) for facial expression recognition using the FER2013 dataset. The trained model can predict emotions (Angry, Fear, Happy, Sad, Surprise) from facial images1. The project uses TensorFlow for model training and Scikit-Learn for data preprocessing.**

**sachinsisondiya/facial-emotion-detection-using-cnn:**

* **Summary: This repository demonstrates facial emotion detection using CNN. It involves capturing and processing images to detect key facial landmarks and interpret emotional states2. The project uses TensorFlow for model training and Scikit-Learn for data handling and preprocessing.**

**A CNN-Based Approach for Facial Emotion Detection:**

* **Summary: This paper presents a CNN predictor model using TensorFlow to predict emotions from facial images. The model is trained on the FER2013 dataset and demonstrates satisfactory accuracy3. Scikit-Learn is used for data preprocessing and evaluation.**

**Human Emotion Recognition Based on Facial Expressions via FS-CNN:**

* **Summary: This study proposes a face-sensitive convolutional neural network (FS-CNN) for human emotion recognition. The FS-CNN detects faces and analyzes facial landmarks to predict emotions4. TensorFlow is used for model training, and Scikit-Learn aids in data preprocessing.**

SOFTWARE USED

**TensorFlow:** (Version 2.10 or higher): A deep learning framework used for building, training, and deploying the CNN model. (Installed via pip install tensorflow==2.10)

**Keras:** (Version 2.10 or higher): A high-level neural networks API used in conjunction with TensorFlow to simplify model development. (Included with TensorFlow)

**Pandas:** (Version 1.5.0 or higher): A data manipulation library for handling and processing datasets. (Installed via pip install pandas==1.5.0)

**NumPy:** (Version 1.23 or higher): A library used for numerical operations on arrays and matrices, crucial for handling image data. (Installed via pip install numpy==1.23)

**Jupyter: Notebook** (Version 6.4.12 or higher): Ideal for interactive development and visualizations. (Installed via pip install notebook==6.4.12)

**TQDM:** (Version 4.64.1 or higher): A library for showing progress bars during training. (Installed via pip install tqdm==4.64.1)

**OpenCV-contrib-python:** (Version 4.6.0 or higher): A library used for real-time computer vision tasks, including capturing video from the webcam and pre-processing the images. (Installed via pip install opencv-contrib-python==4.6.0)

**Scikit-learn:** (Version 1.2.0 or higher): A machine learning library used for evaluating model performance, including generating confusion matrices and calculating F1-scores. (Installed via pip install scikit-learn==1.2.0)

EXPERIMENT RESULT

**1. Dataset Description**

The dataset used for training and testing the models is the **Face Expression Recognition Dataset**, available on Kaggle. This dataset consists of images that represent seven different facial expressions: Angry, Disgust, Fear, Happy, Neutral, Sad, and Surprise.

* **Dataset Link**: [Face Expression Recognition Dataset](https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset/data?select=images)
* **Number of Images**: The dataset contains a collection of images, with each image labeled according to one of the seven facial expressions.
* **Emotion Classes**: Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise

**2. Preprocessing Steps**

Before feeding the data into the models, several preprocessing steps were performed:

* **Image Resizing**: The images were resized to 48x48 pixels to standardize the input size for both models.
* **Normalization**: Pixel values were scaled between 0 and 1 to improve model convergence.
* **Data Augmentation**: To prevent overfitting and improve generalization, data augmentation techniques such as rotation, horizontal flipping, and zooming were applied to the training images.

**3. CNN Model Results**

For the CNN model, a basic architecture was implemented, which included several convolutional layers, max-pooling layers, and a fully connected layer at the end. The model was trained using the **Adam optimizer** and **categorical cross-entropy loss** function. The model was trained for 20 epochs.

* **Accuracy**: The CNN model achieved a test accuracy of **72%**.
* **Training Progress**: The model showed a steady improvement in accuracy during the training process.
* **Confusion Matrix**: The confusion matrix for the CNN model indicated that it performed better on emotions such as "Happy" and "Surprise," but struggled with emotions like "Disgust" and "Fear."

**4. SVM Model Results**

The SVM model was implemented using a **Radial Basis Function (RBF)** kernel. The model was trained on the same dataset, and various parameter tuning methods were applied to improve performance. Despite optimization attempts, the SVM model achieved a lower performance compared to CNN.

* **Accuracy**: The SVM model achieved a test accuracy of **35%**.
* **Training Process**: The SVM model struggled to capture the complex features of the images without deep learning's ability to extract hierarchical features.
* **Confusion Matrix**: The SVM model performed poorly across all classes, especially in detecting emotions like "Fear" and "Surprise."

**5. Comparison of Results**

The following table summarizes the accuracy of both models:

| **Model** | **Accuracy (%)** |
| --- | --- |
| CNN | 72 |
| SVM | 35 |

As evident from the comparison, the **CNN model** significantly outperformed the **SVM model**. The CNN's ability to learn complex features from raw pixel data enabled it to achieve higher accuracy. In contrast, the SVM model, which requires manual feature engineering, struggled to provide competitive results on this image classification task.

**6. Discussion of Results**

The results indicate that convolutional neural networks are far better suited for facial emotion recognition tasks compared to traditional machine learning models like SVM. The **CNN's 72% accuracy** shows that deep learning models are capable of extracting useful features from raw image data, making them more effective for image classification tasks.

On the other hand, the **SVM model's 35% accuracy** highlights the limitations of traditional machine learning approaches when applied to image-based tasks. Without specialized features or more sophisticated models, the SVM model struggled to achieve meaningful results.

**Key Observations:**

* **CNN Model**: The CNN model performed significantly better than the SVM, likely due to its deep learning architecture's ability to automatically learn features from the raw images.
* **SVM Model**: The SVM model's performance was limited by the need for manually engineered features, and it showed poor results across most emotion categories.Further improvements in both models could include experimenting with more complex CNN architectures (e.g., ResNet, VGG) or incorporating more advanced feature extraction techniques for the SVM.

CONCLUSION

In this project, we developed and evaluated a face emotion detection system using two distinct approaches: a Convolutional Neural Network (CNN) and a Support Vector Machine (SVM). Our aim was to compare the effectiveness of these models in recognizing facial emotions from a publicly available dataset, specifically the [Face Expression Recognition Dataset](https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset/data?select=images).

Through the course of the project, both models were implemented and assessed based on their performance. The CNN model demonstrated superior accuracy, achieving a test accuracy of 72%, significantly outperforming the SVM model, which attained an accuracy of only 35%. The results reflect the advantages of using deep learning techniques for image classification tasks, as CNNs can automatically learn and extract complex features from images, unlike traditional machine learning models such as SVM that rely on manual feature extraction.

The CNN model's higher accuracy indicates that it is better suited for recognizing the subtleties in facial expressions, providing more reliable results for real-world applications. In contrast, the SVM model faced challenges in handling the high-dimensional data and failed to match the performance of the CNN. This outcome underscores the importance of model selection in machine learning projects, particularly for tasks involving complex data such as images.

Overall, the findings of this project validate the effectiveness of convolutional neural networks for emotion detection in images. Future work could focus on refining the CNN architecture, employing transfer learning with pre-trained models, or exploring hybrid models that combine the strengths of both machine learning and deep learning approaches to enhance accuracy and robustness.

In conclusion, the project successfully demonstrated that CNNs provide a powerful and efficient approach for facial emotion recognition, paving the way for further advancements in emotion-based human-computer interaction systems.

FUTURE SCOPE

**Future Scope**

While the current project has demonstrated the efficacy of CNNs for facial emotion detection, there are several areas for future exploration and enhancement:

1. **Advanced Model Architectures**: Implementing deeper and more sophisticated architectures, such as ResNet, VGG, or Inception, could further improve the accuracy and robustness of emotion recognition.
2. **Transfer Learning**: Leveraging pre-trained models on larger, more diverse datasets could provide better feature extraction and improved results, especially when fine-tuned for specific tasks.
3. **Real-Time Emotion Detection**: Integrating the model into real-time systems and optimizing it for performance on lower-power devices would enable practical applications such as emotion-aware user interfaces and assistive technology.
4. **Multimodal Analysis**: Combining facial emotion recognition with other modalities, such as speech or physiological data, could result in more comprehensive and accurate emotion analysis systems.
5. **Improved Preprocessing Techniques**: Experimenting with more advanced image preprocessing methods and augmentation strategies could enhance model generalizability and performance.
6. **Emotion Intensity Detection**: Extending the model to detect not only the type of emotion but also the intensity of the expression would add another layer of nuance to its capabilities.
7. **Cross-Cultural Training**: Training the model on datasets that include diverse demographics and cultural expressions could ensure more equitable performance across different user groups.
8. **Deployment in Applications**: Further work could focus on deploying the model into applications such as virtual reality (VR), mental health monitoring tools, and adaptive educational software.

These advancements could contribute significantly to the development of intelligent systems capable of understanding and interacting with human emotions in a more natural and effective manner.

GITHUB LINK AND DATASET LINK

GITHUB LINK: https://gitlab.com/dipanshu-84/Emotion\_detection.git

DATASET LINK: https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset/data?select=images