AI powered solution for Alexithymia

CS-251 course project

Muhammad Ahmed
Hasham Nadeem

Introduction

This project aims to leverage advances in neural networks and computer vision to develop a robust system for accurately extracting emotions from live faces. Emotion recognition plays a vital role in interpersonal communication and psychological well-being. However, for individuals with alexithymia, a condition characterized by difficulty in recognizing and expressing emotions, this ability is often impaired. Manual analysis of emotions is not only subjective but also time-consuming. Hence, the need for an automated system that can precisely recognize emotions from facial expressions is paramount. By harnessing computer vision techniques and neural networks, this project endeavors to address this need and provide valuable insights to individuals with alexithymia.

Motivation

The motivation behind this project stems from the challenges faced by individuals with alexithymia in understanding and expressing emotions. Traditional methods of emotion recognition rely heavily on manual analysis, which can be prone to biases and inconsistencies. Moreover, the sheer volume of information in today's digital age exacerbates the difficulty in accurately discerning emotions. By developing an intelligent system capable of automating emotion recognition from facial expressions, we aim to alleviate these challenges and empower individuals with alexithymia to better understand their emotional states and those of others.

Methodology

1. Data Collection: Collecting a diverse dataset of facial images encompassing a wide range of emotional expressions, including happiness, sadness, surprise, anger, contempt, disgust, and fear.
2. Data Preprocessing: Preprocessing the facial images to enhance quality, including grayscale conversion, resizing, and augmentation techniques such as random horizontal flipping and affine transformations.

3. Model Architecture Design: Designing a convolutional neural network (CNN) architecture tailored for emotion recognition tasks, incorporating multiple layers of convolutions, activations, and pooling operations.
4. Model Training: Training the CNN model on the preprocessed dataset using appropriate optimization algorithms such as Adam, and monitoring performance metrics on validation data to prevent overfitting.
5. Model Evaluation: Evaluating the trained model on a separate test dataset to assess its performance in accurately recognizing emotions from facial expressions.
6. Application Development: Developing an application interface that integrates the trained model, allowing users to capture live video feeds and receive real-time feedback on emotional states.

Dataset

In 2000, the Cohn-Kanade (CK) database was released for the purpose of promoting research into automatically detecting individual facial expressions. Since then, the CK database has become one of the most widely used test-beds for algorithm development and evaluation. During this period, three limitations have become apparent: 1) While AU codes are well validated, emotion labels are not, as they refer to what was requested rather than what was actually performed, 2) The lack of a common performance metric against which to evaluate new algorithms, and 3) Standard protocols for common databases have not emerged. As a consequence, the CK database has been used for both AU and emotion detection (even though labels for the latter have not been validated), comparison with benchmark algorithms is missing, and use of random subsets of the original database makes meta-analyses difficult. To address these and other concerns, we present the Extended Cohn-Kanade (CK+) database. The number of sequences is increased by 22% and the number of subjects by 27%. The target expression for each sequence is fully FACS coded and emotion labels have been revised and validated. In addition to this, non-posed sequences for several types of smiles and their associated metadata have been

added. We present baseline results using Active Appearance Models (AAMs) and a linear support vector machine (SVM) classifier using a leave-one-out subject cross-validation for both AU and emotion detection for the posed data. The emotion and AU labels, along with the extended image data and tracked landmarks will be made available July 2010.

Data preprocessing

The data preprocessing steps outlined here are crucial for preparing facial images for subsequent analysis, particularly for emotion recognition tasks.

Firstly, the **Grayscale()** transformation converts the input images into grayscale format, reducing the dimensionality of the data while retaining essential features related to facial expressions. Grayscale images simplify subsequent processing steps by eliminating color information, making the model more robust to variations in lighting conditions and skin tones.

Next, the **Resize((img_size, img_size))** transformation standardizes the size of the images to a fixed resolution of 48x48 pixels. This ensures uniformity across the dataset and facilitates efficient processing by the neural network model. Standardizing the image size also helps mitigate potential issues related to variations in facial sizes and aspect ratios.

The **RandomHorizontalFlip()** transformation introduces variability into the dataset by randomly flipping images horizontally with a probability of 0.5. This augmentation technique helps enhance the model's ability to generalize by simulating different viewing angles and orientations of facial expressions. It also helps prevent overfitting by exposing the model to a diverse range of facial orientations during training.

The **RandomAffine(0, translate=(0.1, 0.1))** transformation further augments the dataset by randomly applying affine transformations, such as translation, rotation, and shearing, to the images. In this case, only translation is utilized, with a maximum displacement of 0.1 times the image size in both the horizontal and vertical directions. These transformations introduce variations in facial position and alignment, improving the model's robustness to spatial distortions and enhancing its generalization capabilities.

Subsequently, the **ToTensor()** transformation converts the processed images into PyTorch tensor format, which is the required input format for neural network models. This

transformation also normalizes the pixel values of the images to the range [0, 1], facilitating convergence during model training and improving gradient stability.

Finally, the **Normalize((0.5,), (0.5,))** transformation further standardizes the pixel values of the images by subtracting the mean (0.5) and dividing by the standard deviation (0.5) along each channel. This normalization step helps center the pixel values around zero and scales them to a range suitable for the activation functions used in the neural network model, promoting more stable and efficient learning.

Convolutional Neural Network (CNN) Architecture:

The convolutional neural network (CNN) architecture devised for emotion recognition from facial images comprises several key components meticulously crafted to extract and discern salient features. It begins with a sequence of convolutional layers, starting with the first layer termed 'layer1'. This initial layer applies 32 distinct filters to the grayscale input images, each filter scanning through a 3x3 window. The resultant feature maps are then subjected to rectified linear unit (ReLU) activation, introducing non-linearity to the network. Following activation, max-pooling is employed to downsample the feature maps by a factor of 2, enhancing computational efficiency while preserving crucial information. This process is repeated in the subsequent layers, namely 'layer2' and 'layer3', each building upon the extracted features of the preceding layer. 'Layer2' elevates the complexity by increasing the number of filters to 64, further refining the feature representation. Similarly, 'layer3' intensifies the feature extraction process, employing 128 filters to capture intricate patterns indicative of various emotions. After convolution and pooling, the feature maps are flattened into a 1D vector, facilitating seamless integration with the fully connected layers. Here, the flattened features traverse through 'fc1', a densely connected layer comprising 128 neurons, where another ReLU activation fosters non-linear transformations. Subsequently, 'fc2', the final fully connected layer, transforms the 128-dimensional feature vector into predictions across the seven

emotion classes. As this layer serves as the output layer, no activation function is applied, enabling raw predictions to be produced. This meticulously designed CNN architecture harmoniously blends convolutional and fully connected layers, leveraging their collective prowess to distill nuanced facial features and discern emotions with precision

Training procedure

The training procedure for the emotion recognition CNN model is meticulously structured to optimize classification accuracy. It begins with the initialization of a loss function and optimizer, setting the stage for subsequent model optimization. The choice of Cross-Entropy Loss, a standard for multi-class classification tasks, and the Adam optimizer, known for its effectiveness in optimizing neural networks, provides a solid foundation. The learning rate, typically set to a value such as 0.001, governs the step size during optimization, influencing the convergence of the model. Each training cycle consists of multiple epochs, representing complete passes through the dataset. Within each epoch, the model operates in a training mode, where it learns from the data to adjust its parameters.

During the epoch-wise training loop, the model is systematically exposed to mini-batches of data, enabling efficient processing and gradient updates. These mini-batches not only facilitate faster computations but also help in avoiding memory constraints, especially when dealing with large datasets. Within each mini-batch iteration, the images and their corresponding labels are transferred to the appropriate hardware device, such as a GPU, to leverage hardware acceleration if available. The model then computes predicted class probabilities for the input images and compares them with the ground truth labels to compute the loss using the defined criterion. This loss serves as a measure of dissimilarity between the predicted and actual labels and guides the optimization process.

The optimization step involves backpropagation, where the gradients of the loss with respect to the model parameters are computed. These gradients indicate the direction and magnitude of parameter updates required to minimize the loss. The optimizer utilizes these gradients to adjust the model parameters iteratively, effectively reducing the loss and improving the model's performance. The running loss, accumulated over the mini-batches within an epoch, provides insights into the model's training progress and convergence

behavior. At the end of each epoch, the average loss is computed and reported, offering a quantitative measure of the model's performance on the training data.

Following the completion of each epoch, the model's performance is evaluated on a separate validation dataset. This validation step is crucial for assessing the model's generalization ability and preventing overfitting. By evaluating the model on unseen data, one can gauge its performance in real-world scenarios. The validation accuracy, computed as the percentage of correctly classified samples out of the total validation dataset, serves as a key metric for model evaluation. This iterative training-validation cycle continues until the model achieves satisfactory performance or convergence. Through this systematic training process, the emotion recognition CNN model gradually learns to accurately classify facial expressions, honing its ability to discern subtle emotional cues in images.

Evaluation metrics

The evaluation metrics computed using the sklearn library provide comprehensive insights into the performance of the emotion recognition model on the test dataset. Here's a detailed summary of each metric:

Test Accuracy (0.95):

Test accuracy represents the proportion of correctly classified samples out of the total number of samples in the test dataset.

In this context, the model achieved an accuracy of 95%, indicating that it correctly predicted the emotions in 95% of the test images.

Precision (0.95):

Precision quantifies the model's ability to correctly classify positive samples (correctly predicted emotions) out of all samples predicted as positive.

A precision score of 0.95 indicates that 95% of the predicted positive classifications (emotions) were indeed correct.

Recall (0.95):

Recall, also known as sensitivity, measures the model's ability to identify all positive samples (correctly predicted emotions) out of all actual positive samples.

With a recall score of 0.95, the model successfully captured 95% of all actual positive emotions present in the test dataset.

F1 Score (0.95):

The F1 score is the harmonic mean of precision and recall, providing a balanced measure that considers both false positives and false negatives.

An F1 score of 0.95 signifies that the model achieved a balanced trade-off between precision and recall, with high precision and recall values.

Conclusion and future vision

In conclusion, the developed emotion recognition model demonstrates promising performance in accurately classifying facial expressions, with high test accuracy, precision, recall, and F1 score, all averaging around 95%. This indicates the model's robustness and effectiveness in discerning emotions from live faces, offering potential assistance to individuals with alexithymia by providing valuable insights into emotional states.

Looking ahead, there are several avenues for future enhancements and expansions. Firstly, the model could benefit from additional fine-tuning and optimization to further improve its accuracy and generalization capabilities, especially in diverse real-world scenarios. Additionally, the dataset used for training could be augmented with a broader range of facial expressions, including nuanced emotions, to enhance the model's understanding and recognition abilities.

Moreover, exploring more sophisticated architectures or incorporating advanced techniques such as attention mechanisms or ensemble learning could potentially elevate the model's performance to even greater heights. Furthermore, the deployment of the model into real-world applications, such as integrating it into assistive technologies or wearable devices, holds promise for providing timely support to individuals with alexithymia in various contexts. Overall, the future vision encompasses continuous refinement and innovation to develop more robust, reliable, and accessible solutions for emotion recognition and support.