**Facial Emotion Detection using Deep Learning**

- Vinay Sivvala

# Abstract

Facial expression detection plays a key role in computer vision, as it enables the recognition of human emotions by analysing facial characteristics. Such emotion identification is particularly valuable for enhancing human-computer interaction, sentiment analysis, and affective computing. In this project, a modified ResNet-50 model is leveraged for classifying images into six different facial expressions - anger, disgust, fear, happiness, pain, and sadness. Using pre-trained ImageNet weights, the network is fine-tuned for emotion recognition, efficiently extracting facial features from images and assigning them to one of the six categories with high accuracy. This approach is tailored for real-time emotion detection applications, including healthcare, surveillance, and interactive AI systems.

# Introduction

Facial expressions are a primary channel through which humans convey emotions. Recognizing and interpreting these expressions is vital for enhancing user experiences across numerous AI applications, such as virtual assistants, healthcare diagnostics, and human-computer interaction. While traditional emotion recognition systems often rely on complex, rule-based methods, deep learning approaches particularly convolutional neural networks have demonstrated considerable effectiveness in this domain.

# Model Architecture

The chosen model for facial expression detection builds on ResNet-50, a deep convolutional neural network featuring residual connections. By incorporating skip connections between layers, residual networks (ResNets) effectively alleviate the vanishing gradient issue. Their depth enables them to learn intricate feature representations, making them particularly adept at image classification.

## Base Model: ResNet-50

* **ResNet-50**, a 50-layer version of ResNet, is widely recognized for its strong capacity to extract deep-layer features from images.
* **Pretrained Weights:** The network was initialized with weights derived from **ImageNet**—a large-scale dataset containing more than 14 million labeled images spread over 1000 categories. This approach facilitates transfer learning, enabling the model to leverage general features before fine-tuning them specifically for facial expression recognition.

## Customization for Emotion Classification

## The final fully connected layer of ResNet-50 was reconfigured to output the six emotion categories i.e., anger, disgust, fear, happiness, pain, and sadness.

* **Dropout Layer:** A dropout layer (with a 50% rate) was added to the fully connected layer to reduce overfitting and improve generalization.
* **Softmax Activation:** A softmax function was applied after the dense layer to produce class probabilities, ensuring that the emotion label with the highest probability is selected as the final prediction.

## Architecture Overview

* **Input:** An RGB image sized at 224×224.
* **Feature Extraction Layers:** A series of convolutional layers that progressively learn hierarchical representations of the input.
* **Fully Connected Layers:** A specialized dense layer, incorporating dropout, followed by a softmax activation to classify the image into one of six emotion categories.

## Training Process

* **Dataset**: A specialized dataset was used, comprising facial expression images classified into six categories: anger, disgust, fear, happy, pain, and sad.
* **Data Augmentation**: Techniques such as random cropping, rotation, and color jittering were applied to the training images, thereby enriching data variety and mitigating overfitting.
* **Loss Function**: Cross-Entropy Loss, a standard choice for multi-class classification, was employed.
* **Optimizer**: The Adam Optimizer was selected for its adaptive learning rate capabilities, facilitating improved convergence.
* **Learning Rate Scheduler**: **ReduceLROnPlateau** was utilized to automatically decrease the learning rate when the validation loss reached a plateau, further refining the model’s performance.

# Application Flow

Effective data preprocessing plays a vital role in successfully training deep learning models. The input data underwent the following procedures:

* **Resizing**: Each image was scaled to 224×224 pixels, aligning with ResNet-50’s input requirements.
* **Normalization**: Pixel values were standardized using the mean and standard deviation from the ImageNet dataset, facilitating more efficient learning.
* **Augmentation**: Random manipulations such as cropping, rotation, and jittering were applied to enhance the model’s adaptability and robustness against unseen samples.

## Model Inference

After the model has been trained, its inference phase is initiated whenever a user uploads an image via the interface.

* **Input**: A facial image is uploaded by the user and then subjected to resizing and normalization.
* **Processing**: The preprocessed image is fed into the fine-tuned ResNet-50 model.
* **Output**: The model classifies the image into one of the six emotion classes, selecting the category with the highest predicted probability.

The model’s output index is matched to its corresponding emotion label by referencing a dictionary that maps each numerical prediction to one of the six expressions.

**expression\_mapping = {**

**0: "anger",**

**1: "disgust",**

**2: "fear",**

**3: "happy",**

**4: "pain",**

**5: "sad"**

**}**

## Backend (Flask)

A Flask backend hosts an API endpoint, **/predict**, which handles image uploads through an HTTP POST request.

1. **Image Upload**: The user provides an image to the system.
2. **Preprocessing**: The image is resized and normalized so it matches the model’s expected input format.
3. **Prediction**: The processed image is fed into the ResNet-50 model, and the predicted emotion is sent back as a JSON response.

## Frontend (HTML/JavaScript)

The user interface includes a straightforward web page where users can upload an image and retrieve the model’s prediction. Specifically:

1. **File Input**: A form field is provided for uploading an image.
2. **Prediction Button**: After selecting an image, the user clicks “Predict” to initiate the model’s inference.
3. **Prediction Display**: Once the backend returns its response, the predicted facial expression is shown on the page.

## System Diagram

A diagram of a software process

AI-generated content may be incorrect.

# Results

## Accuracy and Performance

A separate test dataset was used to evaluate the model’s performance, focusing on these primary metrics:

* **Validation Accuracy**: Achieved 72% on the validation set.
* **Test Accuracy**: Attained 74% on the held-out test set.
* **Consistency**: Demonstrated reliable performance in classifying images into the six targeted emotion categories.

UI(Output) : Below are some of the outputs I obtained:

A screenshot of a phone

AI-generated content may be incorrect. A screenshot of a person's face

AI-generated content may be incorrect.

A screenshot of a person's face

AI-generated content may be incorrect. A screenshot of a person's face

AI-generated content may be incorrect.

A screenshot of a person's face

AI-generated content may be incorrect.

# Key Applications

1. **Healthcare**: Real-time emotion monitoring can be applied to assess patients’ emotional well-being and mental health conditions.
2. **Customer Interaction**: Integrating emotion recognition into AI-driven services (e.g., chatbots, virtual assistants) can enrich user experiences and improve responsiveness.
3. **Surveillance**: Observing individuals’ emotional states in public spaces can enhance safety measures and aid behavioral analysis.
4. **Education**: Tracking students’ engagement and emotional responses can provide insights for improving online teaching methods and examinations.
5. **Gaming**: Incorporating emotion detection in games allows for adaptive gameplay tailored to the player’s emotional state.

# Challenges

1. **Data Quality**: Model performance is heavily influenced by both the diversity and overall caliber of the dataset.
2. **Class Imbalance**: Unequal representation of certain emotions might skew predictions toward more frequently represented classes.
3. **Generalization**: The model could encounter difficulties recognizing novel facial expressions or handling variations in lighting, thereby limiting its adaptability to unfamiliar conditions.

# Future Scope

1. **Expand Dataset**: Incorporate a broader range of facial expressions and emotions to enhance the model’s ability to generalize.
2. **Multimodal Emotion Recognition**: Integrate data from various sources, such as facial expressions, voice, and text, to build a more comprehensive emotion recognition system.
3. **Edge Deployment**: Tailor the model for resource-constrained settings (e.g., mobile or edge devices), allowing for real-time predictions in environments with limited computing capacity.

# Conclusion

This work illustrates the effectiveness of deep learning for emotion recognition through a ResNet-50 model. By harnessing transfer learning, applying data augmentation, and utilizing advanced training methods, the system achieves reliable facial expression classification. The user-friendly web interface, combined with a Flask backend, makes it well-suited for real-world uses such as healthcare, education, and customer service.

**References:**

**ResNet:**

* <https://www.productteacher.com/quick-product-tips/resnet18-and-resnet50>
* [https://wandb.ai/mostafaibrahim17/ml-articles/reports/The-Basics-of-ResNet50---Vmlldzo2N](https://wandb.ai/mostafaibrahim17/ml-articles/reports/The-Basics-of-ResNet50---Vmlldzo2NDkwNDE2)
* [DkwNDE2](https://wandb.ai/mostafaibrahim17/ml-articles/reports/The-Basics-of-ResNet50---Vmlldzo2NDkwNDE2)

**DataSet:**

<https://www.kaggle.com/datasets/yousefmohamed20/sentiment-images-classifier>