# **Emotion Recognition Using Vision Transformers (ViT)**

Prepared by: Alperen Yıldırım

VESTEL Electronic Industry and Trade Inc.

02.08.2024

## **Introduction**

This report outlines the development and evaluation of an emotion recognition system utilizing Vision Transformers (ViT). The system was trained and tested on combined datasets, standardized for consistency, and fine-tuned for optimal performance.

## **1. Model Loading**

* A pre-trained Vision Transformer model, specialized for facial expression recognition, was loaded using Hugging Face.

## **2. Label Modification and Remapping**

* Given the variation in label definitions across three integrated datasets, we standardized labels to a unified set. This ensured that the model's output was consistent and correctly aligned with the training data.

## **3. Dataset Manipulation**

* The datasets library was employed for efficient data handling. This library facilitated dataset loading, shuffling, and splitting, crucial for robust training and evaluation.

## **4. Image Preprocessing**

* **AutoImageProcessor Initialization:** An AutoImageProcessor was initialized to configure and standardize image inputs according to the model's requirements.

## 

## **5. Data Transformations**

* **Normalization:**
  + Normalized image pixel values to align with the model's expected input range.
* **Data Augmentation Techniques for Training Data:**
  + **Resize:** Ensured all images, both in training and validation sets, were resized to the model's input size.
  + **RandomRotation:** Applied to introduce variability in training images by rotating them randomly.
  + **RandomAdjustSharpness:** Enhanced image sharpness to diversify the training data.
  + **RandomHorizontalFlip:** Provided additional variability by flipping images horizontally.
  + **ToTensor:** Converted images to PyTorch tensors, essential for model processing.
  + **Normalize:** Standardized image values across the dataset.

## **6. Batch Processing Function**

* A collate\_fn function was implemented to process batches of image examples. It stacks image tensors and collects corresponding labels into tensors, preparing data for model ingestion.

## **7. Dataset Splitting**

* The dataset was divided into 80% for training and 20% for testing, using a stratified split to ensure representative distribution of emotion labels.

## **8. Model Initialization**

* The model was initialized with pre-trained weights, and the label mappings were adjusted to align with the unified label set. This setup enabled the model to correctly interpret and classify emotions.

## **9. Training Configuration**

* **Trainer Setup:** The Hugging Face Trainer class was used to manage the training process, with configurations set for learning rate, batch size, number of epochs, and evaluation steps.

## 

## 

## **10. Accuracy and Metric Calculation**

* **compute\_metrics Function:** The function calculated metrics such as accuracy by comparing predicted labels (predictions) with true labels (label\_ids) from the evaluation set.

## **11. Data Collation for Training/Evaluation**

* The collate\_fn function was critical in batching image data and labels, converting them into tensor format for seamless processing by the ViT model.

## 

## **12. Hugging Face Authentication**

* Authentication was handled through notebook\_login() to access Hugging Face's model and dataset resources, facilitating model fine-tuning and evaluation.

## **13. Training and Evaluation**

* The model was trained for a specified number of epochs, with intermediate evaluation on the validation set. Metrics such as loss and accuracy were monitored to track model performance.

## **14. Metric Logging and Saving**

* **Metrics Logging:** Post-training, evaluation metrics were logged and saved, providing a detailed overview of model performance over time.

## **15. Prediction Generation**

* The model's performance on the test set was assessed by generating predictions and comparing them against ground truth labels. Metrics including accuracy and F1 score were calculated.

## **16. Pipeline Initialization**

* A pipeline object was initialized for the trained model, enabling streamlined prediction and inference.

## 

## **17. Results Visualization**

* **Confusion Matrix and Classification Report:** A confusion matrix was plotted to visualize the model's performance across emotion categories. The classification report provided detailed metrics, including precision, recall, and F1 scores for each emotion class.

## **Final Performance Metrics**

* **Accuracy:** The model achieved an accuracy of 91.77%, demonstrating its capability to accurately classify emotions across the dataset.
* **F1 Score:** The macro average F1 score was 91.45%, indicating balanced performance across different emotion classes.

## 

## **18. Real-Time Emotion Recognition**

* **OpenCV for Real-Time Prediction:** The system was integrated with OpenCV for real-time emotion recognition, allowing the model to process live video feed and detect emotions on the fly.
* **Enhanced Accuracy with FaceNet's Face Detection:** To improve the accuracy of predictions, FaceNet's face detection was used to accurately locate and crop faces in the video stream. This preprocessing step ensured that only relevant facial regions were analyzed, reducing noise and enhancing the reliability of the emotion recognition.