Emotion Detection using Vision Transformers (ViT)

Overview

This project implements an **Emotion Detection system** trained on the **FER2013 dataset** using **HuggingFace's Vision Transformer (ViT)** architecture. The pipeline includes preprocessing, training with a frozen backbone, fine-tuning top encoder layers, and evaluating the model using metrics and visualizations.

Dataset

FER2013 is used, located at:

/kaggle/input/fer2013

It consists of 7 emotion classes:

- angry
- disgust
- fear
- happy
- neutral
- sad
- surprise

Split:

- Training samples: 22968
- Validation samples: 5741
- Test samples: 7178

Preprocessing

- Resized to 224x224
- Converted grayscale to 3-channel

- Normalized using HuggingFace's `AutoImageProcessor`

```
transform = transforms.Compose([
    transforms.Grayscale(num_output_channels=3),
    transforms.Resize((224, 224)),
    transforms.ToTensor()
])
```

Model Architecture

Using HuggingFace's `TFViTForImageClassification` with:

- Base model: `google/vit-base-patch16-224-in21k`
- Modified classification head: 7 output classes

```
model = TFViTForImageClassification.from_pretrained(
    "google/vit-base-patch16-224-in21k",
    num_labels=7
)
```

Training Strategy

Phase 1: Train Head Only

- Backbone (`model.vit`) frozen:

```
for var in model.vit.trainable_variables:
    var._trainable = False
```

- Only classification head is trained for 20 epochs
- Model saved to Google Drive: `/content/drive/MyDrive/models/vit-head-only`

Phase 2: Fine-Tune Top Layers

- Load previous model checkpoint
- Unfreeze top 3 encoder layers:

```
for i in [-1, -2, -3]:
    for var in model.vit.encoder.layer[i].trainable_variables:
        var._trainable = True
```

- Retrain for additional epochs with lower LR
- Model saved as: `/content/drive/MyDrive/models/vit-finetuned-3layers`

Evaluation

Metrics

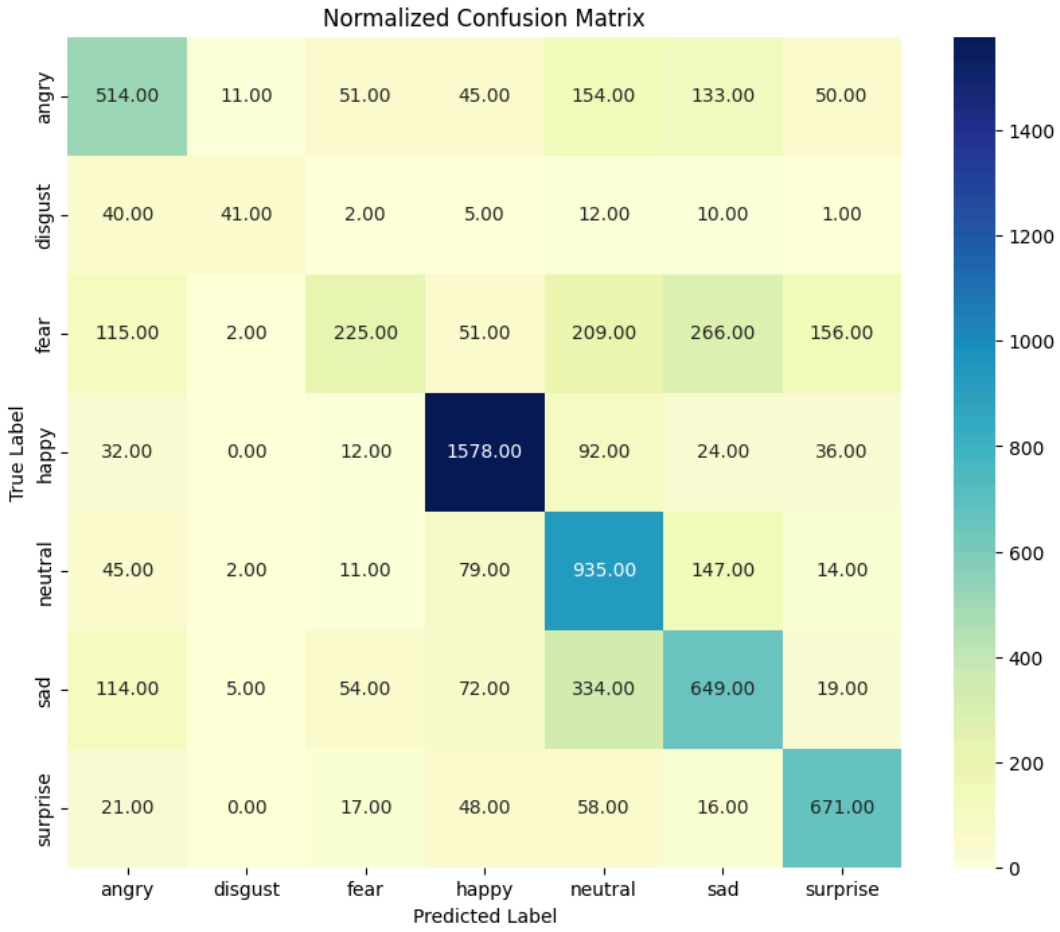
- **Accuracy:** ~64%
- **Classification Report:**

```
Classification Report:

              precision    recall  f1-score   support

       angry       0.58      0.54      0.56       958
     disgust       0.67      0.37      0.48       111
        fear       0.60      0.22      0.32      1024
       happy       0.84      0.89      0.86      1774
     neutral       0.52      0.76      0.62      1233
         sad       0.52      0.52      0.52      1247
    surprise       0.71      0.81      0.75       831

    accuracy                           0.64      7178
   macro avg       0.64      0.59      0.59      7178
weighted avg       0.64      0.64      0.63      7178
```

Confusion Matrix

- Heatmap plotted using Seaborn

sns.heatmap(cm, annot=True, cmap='Blues', xticklabels=class_names, yticklabels=class_names)



Inference

Top-3 Predictions:

probs = tf.nn.softmax(logits, axis=-1).numpy()[0]
top_indices = probs.argsort()[-3:][::-1]

Model Saving & Loading

model.save_pretrained("/path/to/save")

model = TFViTForImageClassification.from_pretrained("/path/to/save")

Results

- Best performance achieved after fine-tuning top 3 layers
- Confusion matrix shows good detection for "happy", "surprise", and "neutral"
- "Fear" and "disgust" remain hardest to predict due to fewer samples

---

Future Work

- Data augmentation (e.g., flips, crops)
- Use ViT Large or other pretrained vision architectures
- Deploy as a Flask/Gradio web app
- Integrate with OpenCV for live video prediction

Contributors

- Harsh Shivhare

Tools & Frameworks

- HuggingFace Transformers
- TensorFlow
- Matplotlib & Seaborn (for plotting)
