EMOtion CLASSifier

PROJECT4 GROUP1

TEAM MEMBERS:

TRUPTI RADADIYA

YAU SHU WONG

ZEJING LIANG (JEAN)

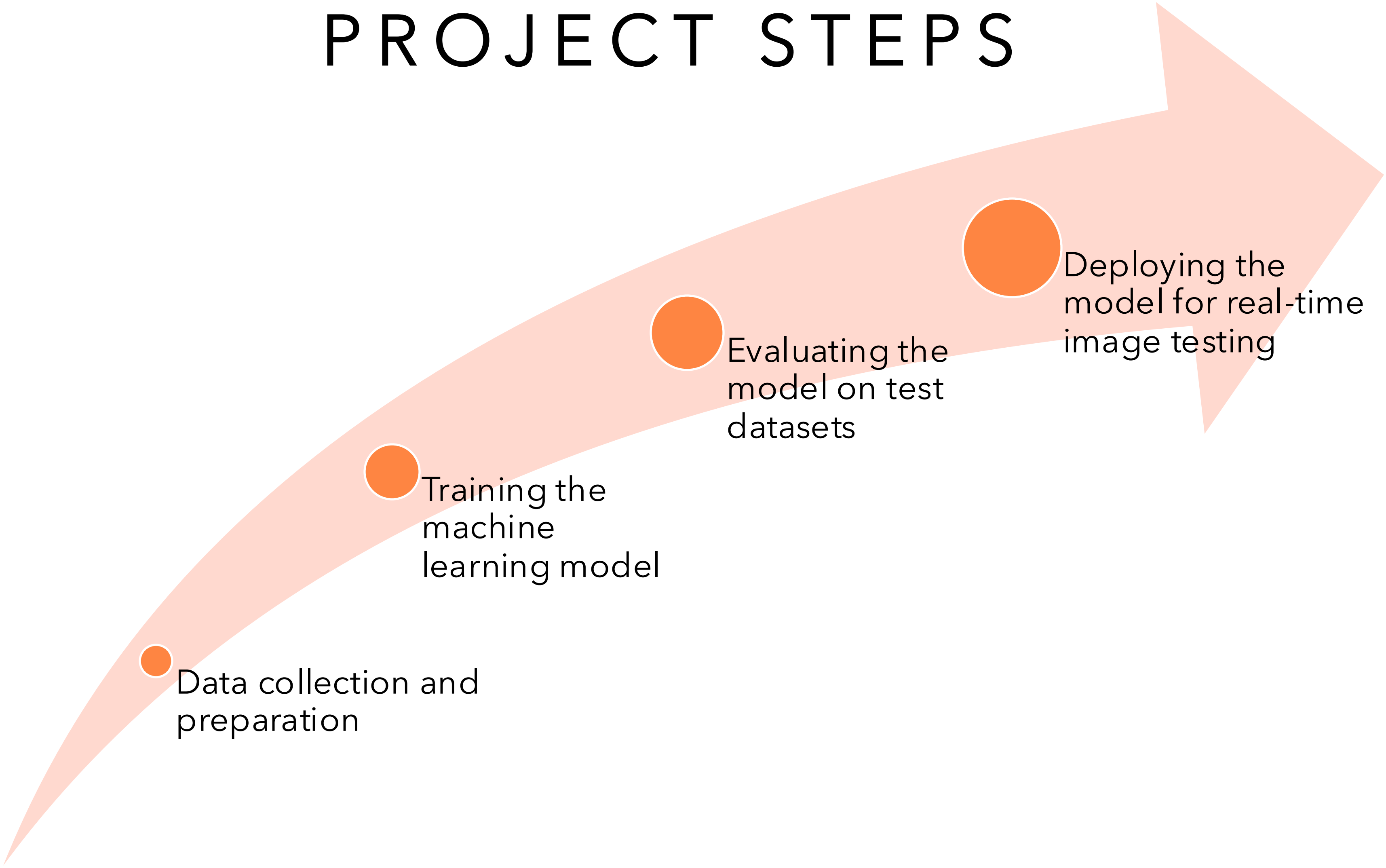
SCOPE

- **Goal**: Develop a machine learning model to detect and classify facial expressions as "happy" or "sad".
- **Dataset**: Downloaded 166 Happy photos and 100 Sad photos from various website.
- **Relevance**: The project is relevant in fields such as psychology, social sciences, and human-computer interaction.

RESEARCH QUESTIONS

1. Can the model accurately classify facial expressions as happy or sad?
2. Does the model perform equally well across different demographic groups?
3. Can the model detect mixed emotions?

PROJECT STEPS



DATA SOURCE

Dataset Source: The dataset is sourced from gettyimages & istockphoto website.

Links:

- https://www.gettyimages.com.au
- https://www.istockphoto.com

Details: The dataset contains a diverse collection of images labeled as "sad" and "happy," suitable for training and testing the emotion classification model

DATASET VALIDATION

MODEL DEVELOPMENT

Model: Neurons, Layers, and Activation Functions

Input Layer: The input layer corresponds to the flattened pixel data from the images.

Hidden Layers: The model is designed with multiple hidden layers using the ReLU activation function to capture complex patterns in the image data.

Output Layer: The output layer has one neuron with a sigmoid activation function to classify the images into sad or happy categories.

HIDDEN LAYERS

Conv2D (Convolutional Layer): The Conv2D layer is the core building block of a Convolutional Neural Network (CNN). It performs convolution operations on the input image, applying filters (or kernels) to create feature maps. Each filter detects specific features such as edges, textures, or patterns in the image.

MaxPooling2D (Pooling Layer): The MaxPooling2D layer reduces the spatial dimensions (height and width) of the feature maps, which helps in downsampling, reducing the computational load, and controlling overfitting. It selects the maximum value from a region of the feature map, retaining the most important features.

BatchNormalization: The BatchNormalization layer normalizes the output of the previous layer by adjusting and scaling the activations. This helps in speeding up training, improving model stability, and allowing higher learning rates.

STEPS TAKEN TO INCREASE MODEL PERFORMANCE

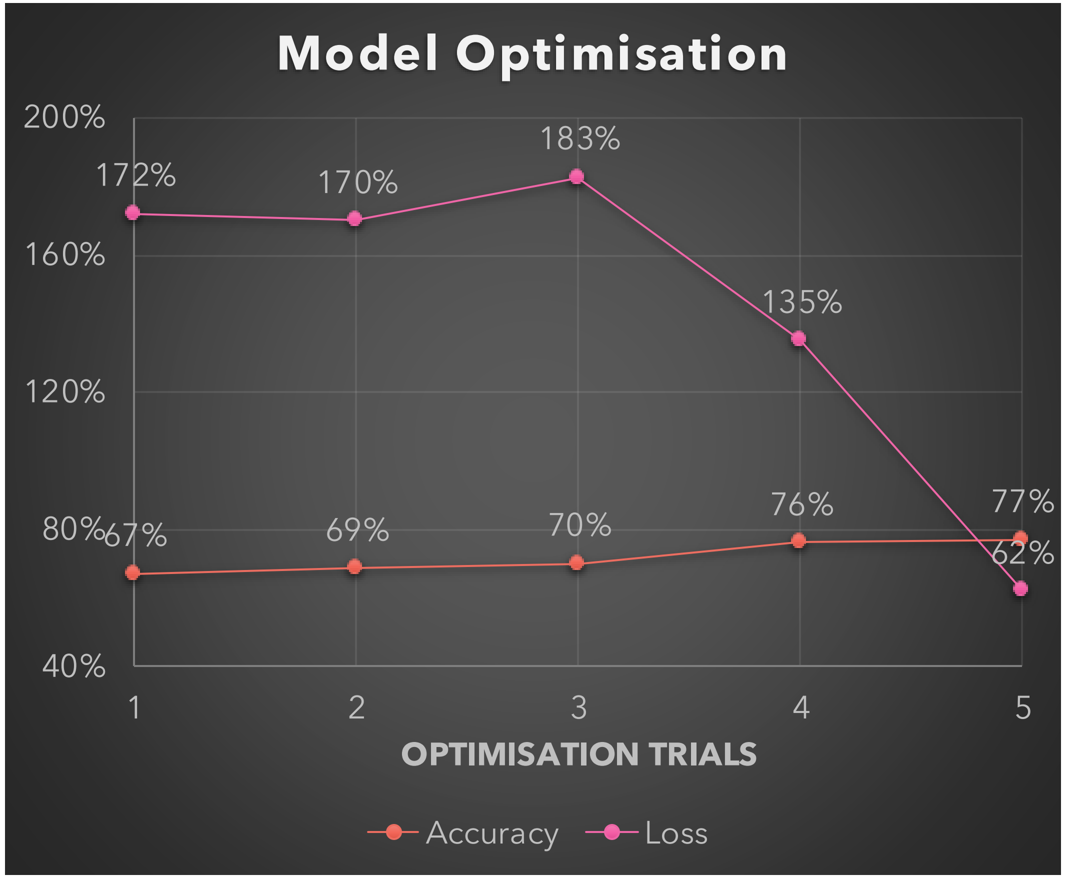
Attempt 1: In the first attempt to improve the model, additional convolutional layers with increased filters were introduced.

Attempt 2: The second attempt increased the complexity of the model by further increasing the number of filters in the convolutional layers.

Attempt 3: In the third attempt, an additional convolutional layer with 32 filters was introduced.

Attempt 4: The final attempt used hyperparameter tuning with Keras Tuner to optimize the model.

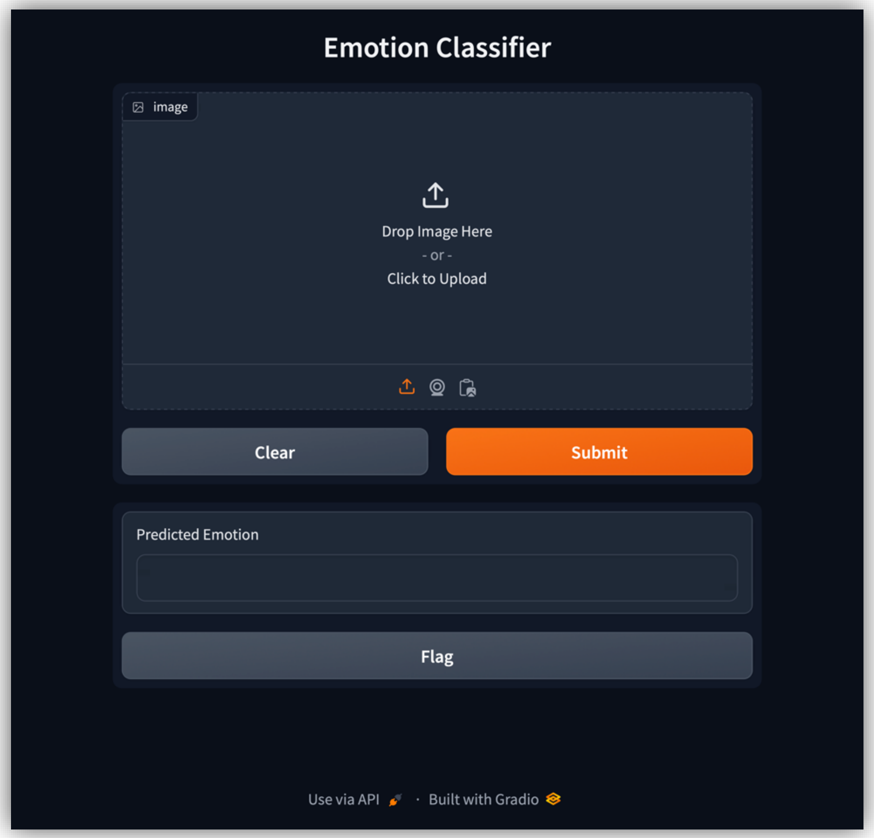
Attempt 5: In the fifth attempt, the hyperparameters identified in Attempt 4 were used, but batch normalization layers were added to the model to improve training stability and potentially increase accuracy.



DEPLOYING THE MODEL

Deploying the model for real-time image testing using Gradio.

The model is designed to be deployed as a platform where users can upload photos, and the model automatically classifies them into happy or sad categories.




```
# Load trained model
model = tf.keras.models.load_model('Models/EmotionClassifierOptimize3.h5')
```

```
(dev) yaushuwong@MacBook-Pro Emotion_Classifier % python app.py
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics`
Running on local URL:  http://127.0.0.1:7860
Running on public URL: https://9ac0ce12fef586b10a.gradio.live
```

CAN THE MODEL ACCURATELY CLASSIFY FACIAL EXPRESSIONS AS HAPPY OR SAD?

```
# Evaluate the model on the normalized test data and encoded test labels, returning the accuracy

accuracy = model.evaluate(X_test_normalized, y_test_encoded, verbose=2)
```

```
3/3 - 1s - 375ms/step - accuracy: 0.7761 - loss: 0.6248
```

DOES THE MODEL PERFORM EQUALLY WELL ACROSS DIFFERENT DEMOGRAPHIC GROUPS?

Test Scope:

- Gather 5 off pictures of each age groups:
  - Toddler | Children | Teenager | Adult | Seniors
- Gather 5 off pictures of each human diversity:
  - Hispanic | South Asians | East Asians | Caucasian | African
- Find the prediction accuracy for each category.

AGE GROUP

Age Category	Result 1	Result 2	Result 3	Result 4	Result 5	Success Rate
Children	wrong	correct	wrong	wrong	correct	40%
Toddler	correct	correct	correct	wrong	wrong	60%
Teenager	wrong	wrong	correct	correct	correct	60%
Senior	correct	wrong	correct	correct	wrong	60%
Adult	wrong	correct	correct	wrong	correct	60%

HUMAN DIVERSITY

Race	Result 1	Result 2	Result 3	Result 4	Result 5	Success Rate
South Asians	wrong	correct	wrong	wrong	correct	40%
Caucasian	wrong	wrong	correct	correct	correct	60%
Hispanic	wrong	correct	correct	wrong	correct	60%
East Asian	correct	correct	correct	correct	wrong	80%
African	wrong	wrong	correct	correct	correct	80%

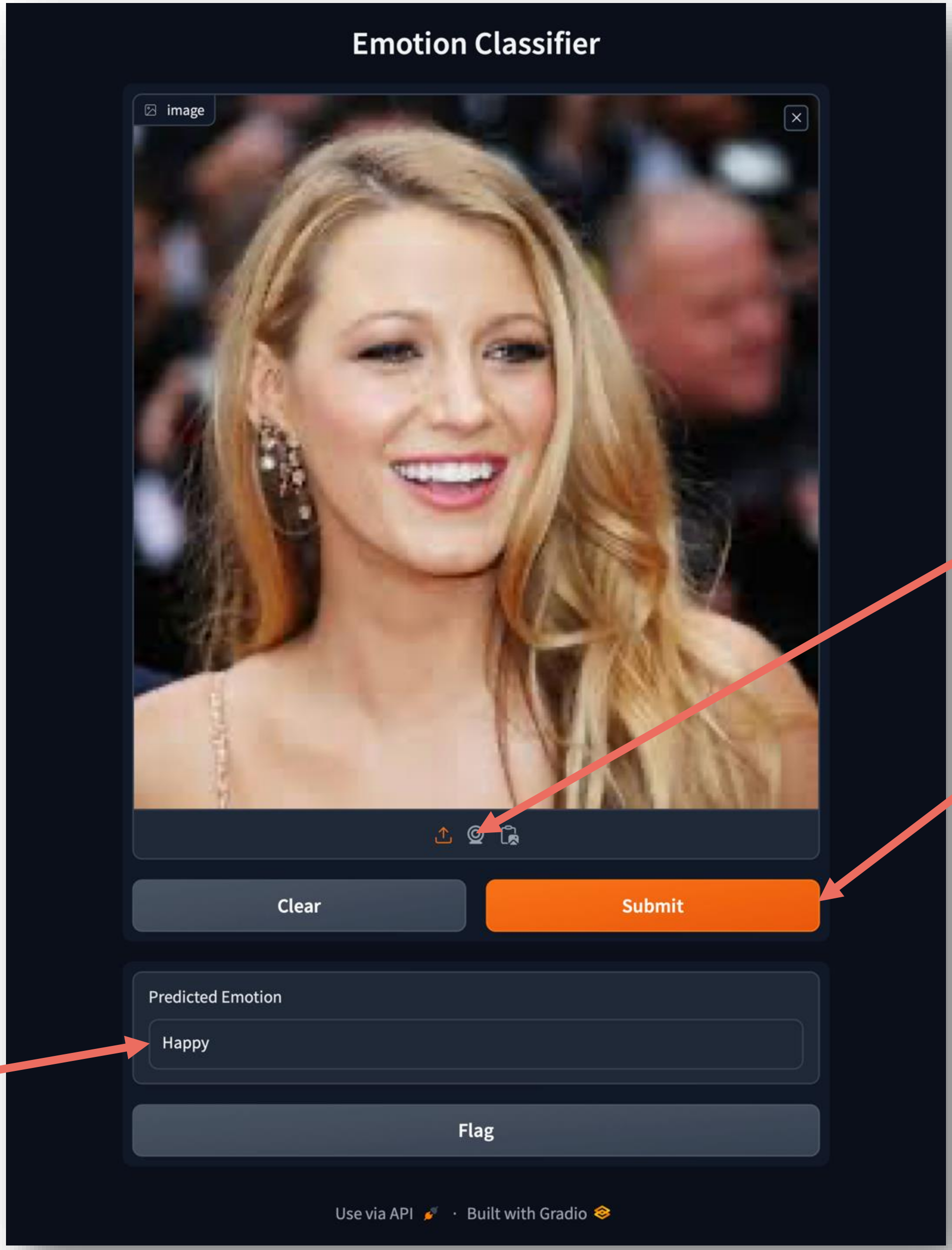
CAN THE MODEL DETECT MIXED EMOTIONS?

Detection of Mixed Emotions:

The model, in its current form, is a binary classifier that distinguishes between happy and sad expressions. It does not detect mixed emotions or provide nuanced classifications like "slightly happy" or "neutral."

DEMO

Try it out!



1 Access webcam

2 Submit

3 Prediction

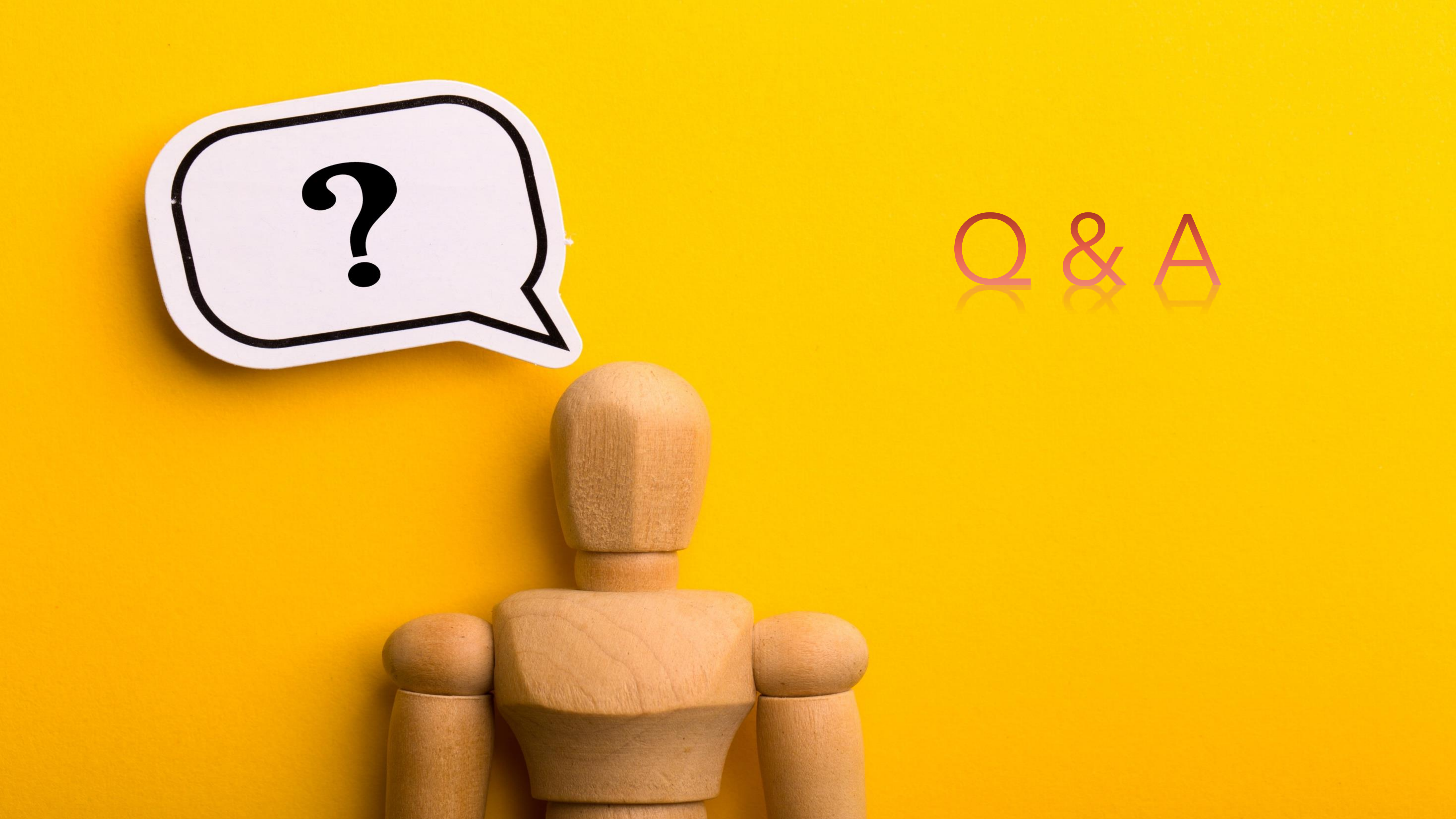
CONCLUSION

- Recap: The emotion detection model developed provides a foundational approach to identifying emotional states from facial images. While it achieved moderate success, there is room for further improvement.
- Significance: This project highlights the potential of machine learning in emotion detection, with future enhancements possibly leading to more accurate and refined models, benefiting fields like psychology and human-computer interaction.

RECOMMENDATIONS

- **Ensemble Methods:** Consider exploring ensemble methods such as Random Forest or Gradient Boosting for potentially better performance.
- **Feature Engineering:** Invest in feature engineering techniques to extract more informative features from the image data.
- **Hyperparameter Tuning:** Continue experimenting with different hyperparameters and model architectures to enhance performance.
- **Datasets:** Find better quality portraits with more variety and categories. Increase the number of datasets from 200+ to 2,000+ or more.


THANK YOU
Special thanks to Camilo & Jeff for carrying us through!
Congrats to everyone for finishing the course!


Q & A