FACIAL EMOTION DETECTION

CAPSTONE PRESENTATION

DR. KAREN FIORAVANTI
12/21/2024

AGENDA

PROBLEM DEFINITION

PROBLEM TO DETERMINE

PROPOSED MODEL SOLUTION

RESULTS

FINAL MODEL SOLUTION

PROPOSED BUSINESS SOLUTIONs

RECOMMENDATIONS

EXECUTIVE SUMMARY

PROBLEM DEFINITION

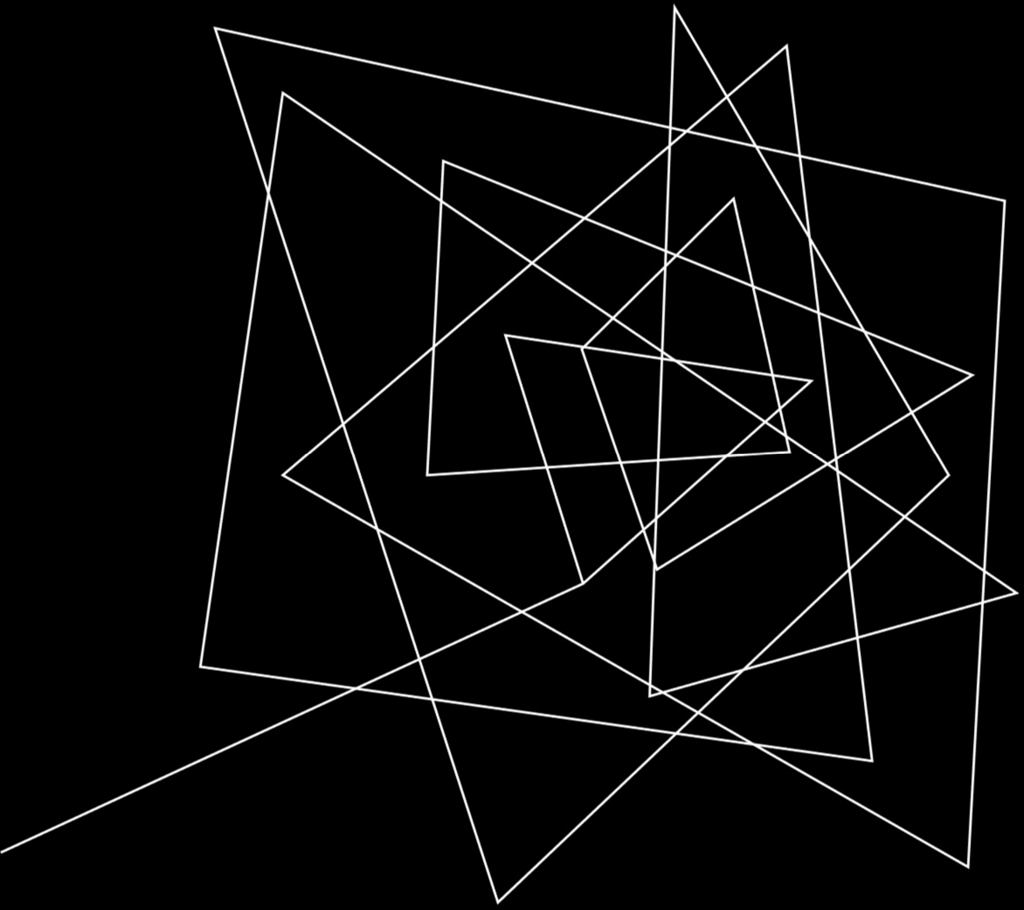
Problem Definition

The objective of this project is to find the most accurate model in order to detect 1 of 4 emotions given digital images of people.

PROBLEM TO DETERMINE

PROBLEM TO DETERMINE

To determine which machine learning (ML) model is the most accurate in processing emotional images and if this concept is feasible for a larger implementation in either the business, medical field or beyond.



PROPOSED MODEL

PROPOSED MODEL

Convolutional Neural Network (CNN)

- Low pre-processing of data
  - Ideal for images
- Translation Invariant
  - Good for identifying facial features
- Scalable

RESULTS

RESULTS

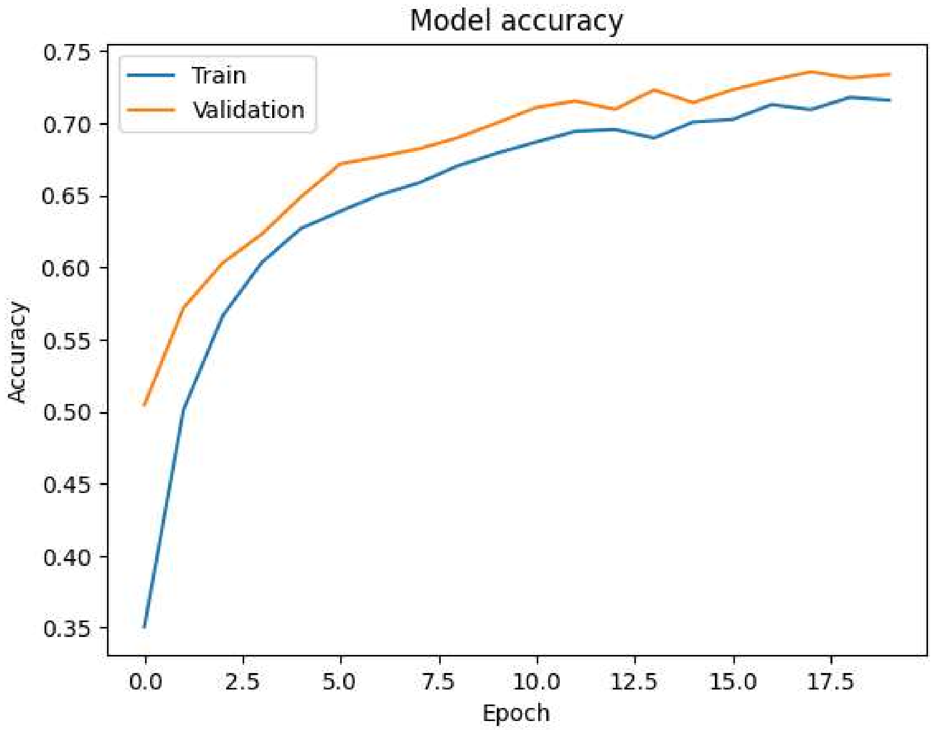
Model	Training Accuracy
CNN – model 1	0.7266
CNN – model 2	0.7656
VGG16	0.5312
ResNet V2	0.5546875
EfficientNet	0.5312
5 Convolutional Blocks	0.7188

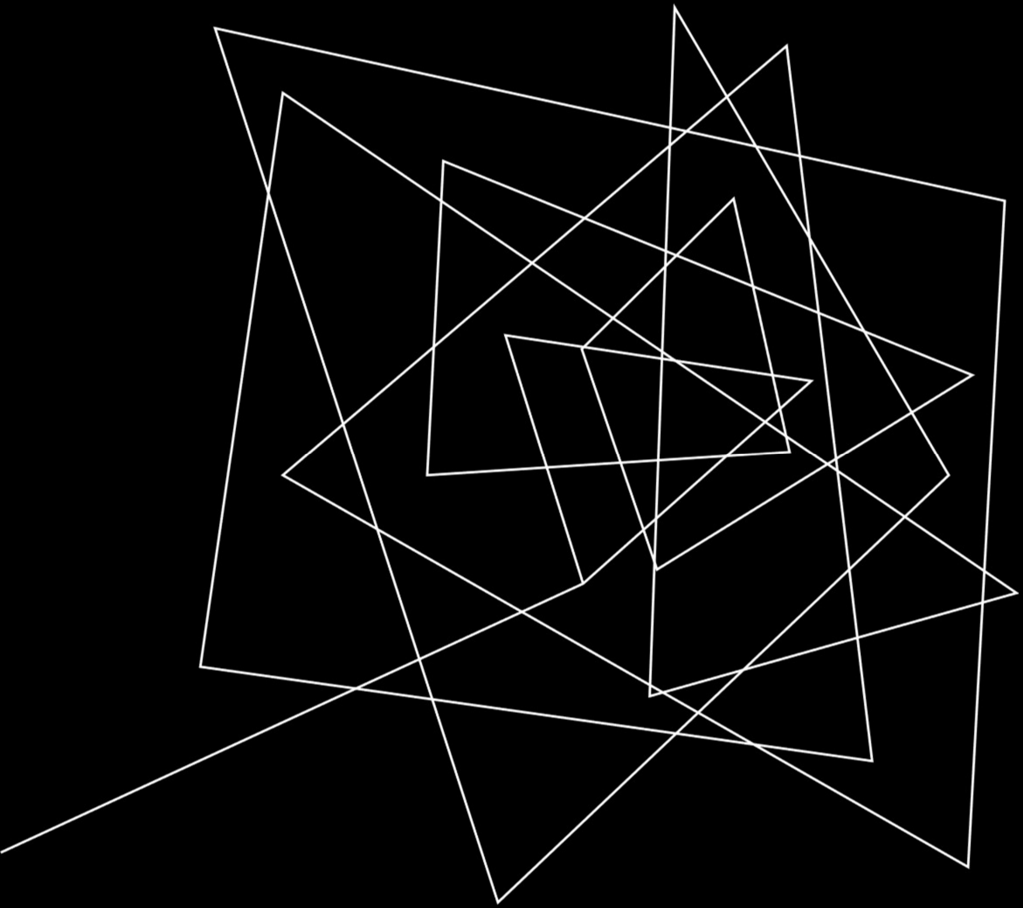
FINAL MODEL

FINAL MODEL

CNN

1. RGB Color mode
2. Relu for activation
3. Pooling (2,2)
4. 0.5 Dropout
5. Softmax activation
6. 20 epochs





PROPOSED BUSINESS SOLUTIONS

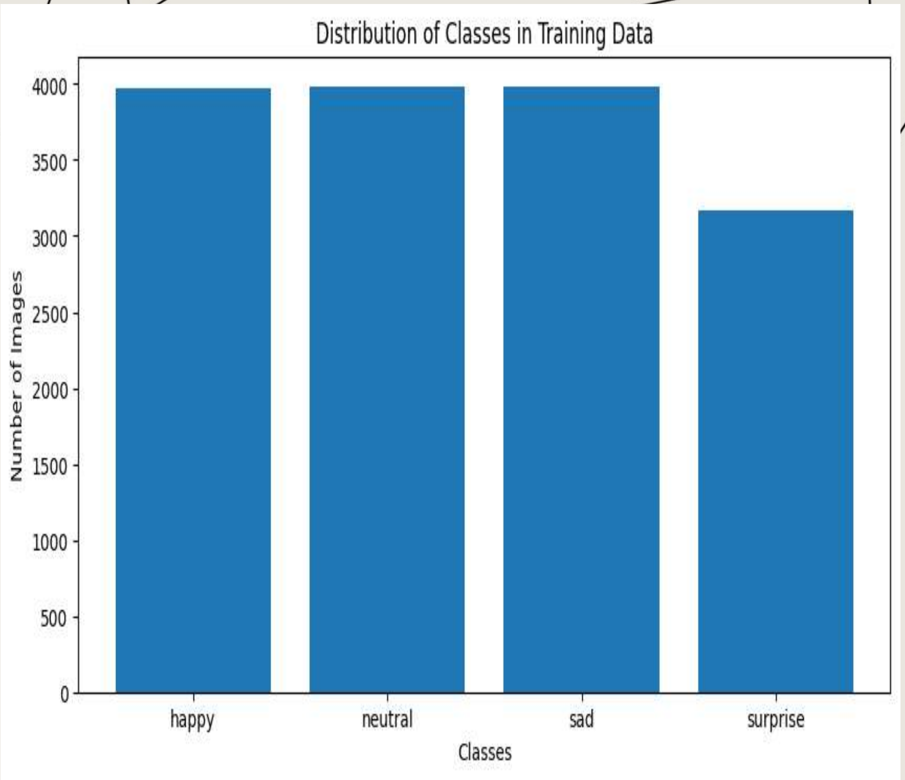
PROPOSED BUSINESS SOLUTIONS

- Retail
  - Account Access
  - AI Assistant evaluation
- Marketing
  - Reaction to advertising/products
- Medical
  - Mental Health diagnosis/treatment
- Information Security
  - Biometrics

RECOMMENDATIONS

RECOMMENDATIONS

- Larger data set
- Balanced data set
- Less emphasis on exceptions, more emphasis on typical representations



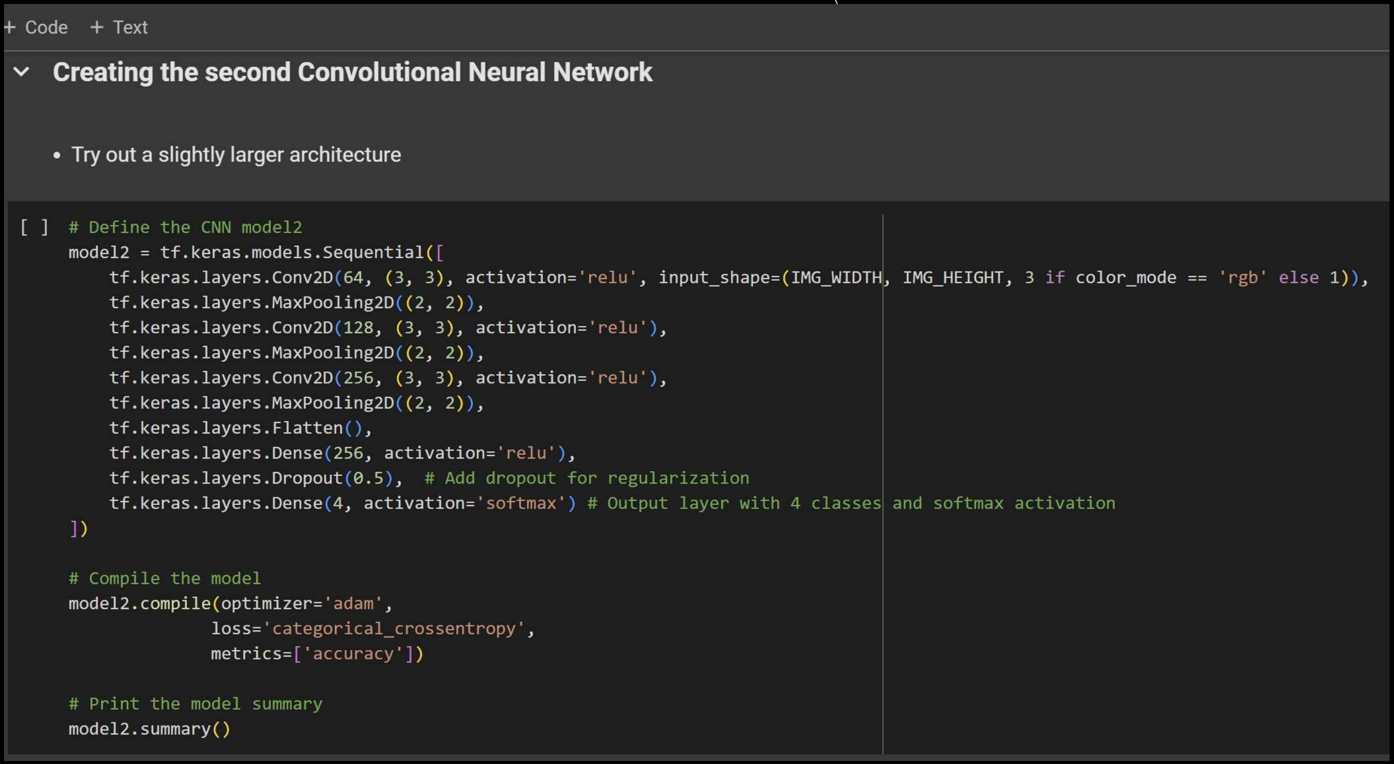
Imbalance existing in data set

EXECUTIVE SUMMARY

- CNN is the preferred model to use
- Work on balancing out the dataset
- Increase examples from which to train
- Uses in the retail, marketing, medical, and information security industries

APPENDIX

CODE SAMPLE



Creating the second Convolutional Neural Network

- Try out a slightly larger architecture

```
# Define the CNN model2
model2 = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu', input_shape=(IMG_WIDTH, IMG_HEIGHT, 3 if color_mode == 'rgb' else 1)),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Conv2D(128, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Conv2D(256, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dropout(0.5),  # Add dropout for regularization
    tf.keras.layers.Dense(4, activation='softmax') # Output layer with 4 classes and softmax activation
])

# Compile the model
model2.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Print the model summary
model2.summary()
```

CNN MODEL 2