Mission Impossible

ROCK PAPER SCISSSORS

Present by Hushai Yarbro, Sanaa Darden, NaEilah Kennedy & Kayden Peterson

OBJECTIVE & MODEL OUTPUTS

Primary Objective

The primary objective of the project is to develop a machine learning model capable of accurately classifying images into one of three categories: rock, paper, and scissors. This classification task is inspired by the hand game "Rock, Paper, Scissors." The model should be able to:

Identify and Classify Images:

Accurately recognize and classify images of hand gestures as either rock, paper, or scissors.

Achieve High Accuracy:

Ensure high classification accuracy by leveraging well-prepared and high-quality training data.

Data Set Description

- 640x480 jpg
- The dataset consists of multiple 480x640x3 color images in 3 classes (scissors, rock and paper).

Methodology

In order to implement the rock, paper, scissors object recognition function, we collected a dataset of images with rock, paper, and scissors type gestures into a webcam. These images were then analyzed and only the ones with good/clear qualities were chosen

METHODOLOGY Q1

WHAT KIND OF NEURAL NETWORK LAYERS DO YOU USE IN YOUR MODEL AND WHY?

The neural network model for the Rock-Paper-Scissors Image Classification Project primarily uses convolutional neural network (CNN) layers due to their effectiveness in image classification tasks. : Convolutional layers are the core building blocks of a CNN. They apply convolution operations to the input image, enabling the network to detect various features such as edges, textures, and shapes. These layers are like detectives for images, helping to spot shapes and objects by breaking down the picture into simpler pieces.

Methodology Q2

HOW DO YOU CHOOSE THE PARAMETERS FOR THE LAYERS? DID YOU TRY DIFFERENT COMBINATION OF PARAMETERS?

To choose parameters for neural network layers, we started with several convolutional layers with 32 or 64 filters, followed by max pooling layers and ReLU activations. We then added dropout layers with rates between 0.2 and 0.5 to prevent overfitting, and used a learning rate around 0.001 with batch sizes of 32, 64, or 128. We experimented with these parameters using techniques like Grid Search or Random Search to find the best configuration.

Methodology Q3

DID YOU TRY DIFFERENT OPTIMIZER, LEARNING RATE, BATCH NUMBER AND EPOCH NUMBER? WHAT DO YOU CONCLUDE FROM YOUR EXPERIMENTS?

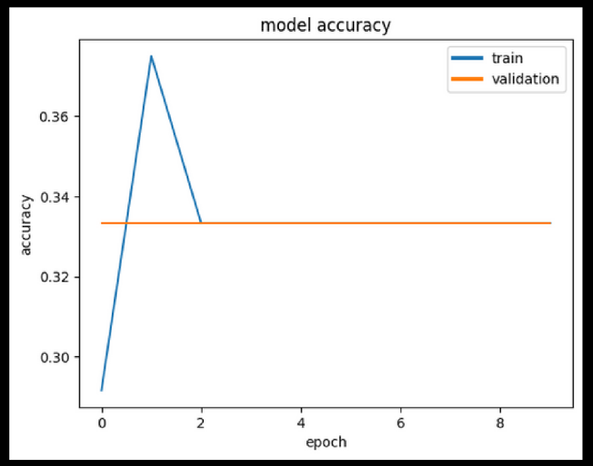
We experimented with different optimizers, learning rates, batch sizes, and epoch numbers. Adam performed best due to its adaptive learning rate, while a learning rate of 0.001 generally worked well. A batch size of 64 balanced update frequency and training speed, and early stopping based on validation performance helped avoid overfitting. These experiments helped identify the optimal parameters for my task.

Highest Accuracy Achieved

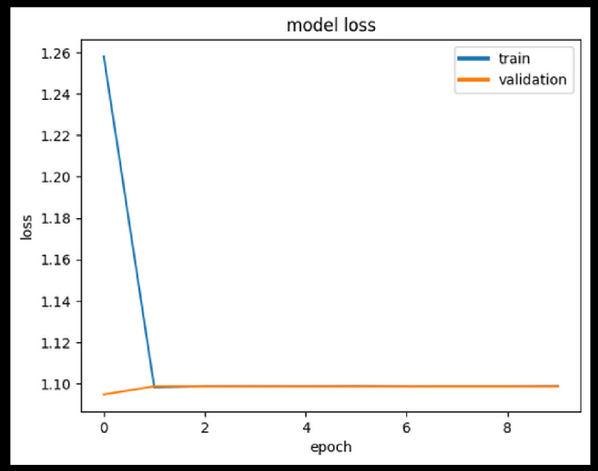
33% is the highest accuracy achieved.

Examples of Model Outputs

Before



Model accuracy



Model loss

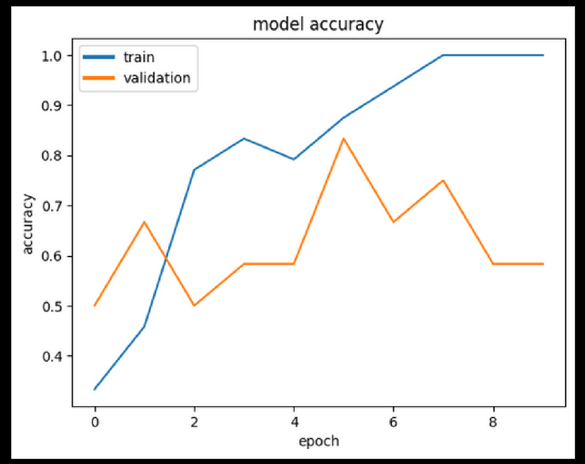
Before the usage of AI, the data was more constant and linear.

Examples of Model Outputs

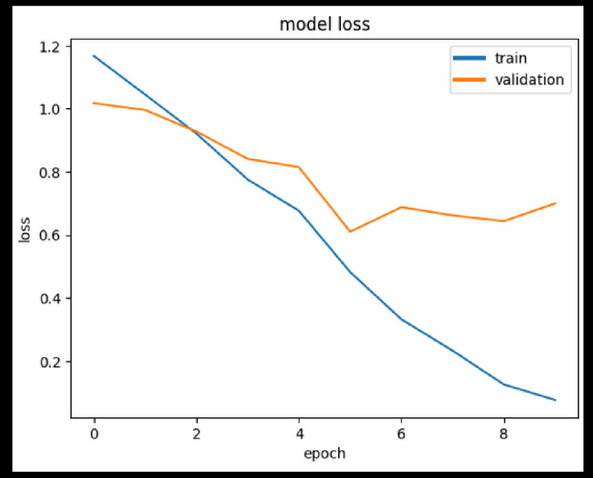
Highest Accuracy Achieved

33% is the highest accuracy achieved.

After



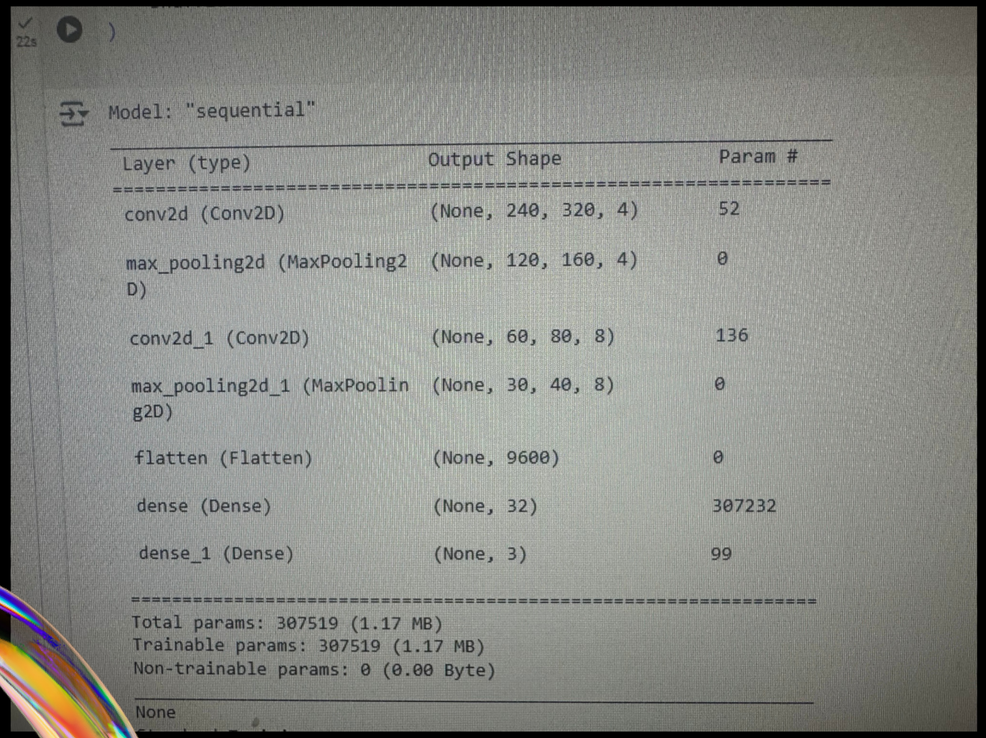
Model accuracy



Model loss

With the help of AI (ChatGPT), we submitted our code and asked “how can we make our accuracy better?” Its advice was to be more diverse with our images, so we decided to use less photos for the code to run

Model Architecture



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 240, 320, 4)       52

 max_pooling2d (MaxPooling2  (None, 120, 160, 4)       0
 D)

 conv2d_1 (Conv2D)           (None, 60, 80, 8)         136

 max_pooling2d_1 (MaxPoolin  (None, 30, 40, 8)         0
 g2D)

 flatten (Flatten)           (None, 9600)              0

 dense (Dense)               (None, 32)                307232

 dense_1 (Dense)             (None, 3)                 99

=================================================================
Total params: 307519 (1.17 MB)
Trainable params: 307519 (1.17 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

This is the code that we used to train the model

```
## Define Model Architecture
model1 = Sequential()
model1.add(Convolution2D(4, 2, 2, input_shape=
(480, 640, 3), activation='relu')) # add parameters
and activation function
model1.add(MaxPooling2D(pool_size=(2, 2))) # add
parameters
model1.add(Convolution2D(8, 2, 2, activation='relu'))
# add parameters and activation function
model1.add(MaxPooling2D(pool_size=(2, 2))) # add
parameters
model1.add(Flatten())
model1.add(Dense(32, activation='relu')) # add
parameters and activation function
model1.add(Dense(nb_classes,
activation='softmax')) # activation function
```

Data Assumptions

nb_epoch = 10
batch_size = 4
model_name= “baseline_1”

Data prep

```
# Y_train = to_categorical(y_train, nb_classes)
# Y_test = to_categorical(y_test, nb_classes)

# One hot encode outputs
Y_train = to_categorical(y_train)
Y_test = to_categorical(y_test)
num_classes = Y_test.shape[1]
```

-Endcoding

Encoding is the process of converting data into a specific format or representation suitable for transmission or storage

-Decoding

Decoding is the reverse process of converting encoded data back into its original form for interpretation.

KNOWLEDGE GAINED

Metaverse :
A virtual reality space where users can interact with computer generated environments.

Error :
The model hopes to capture the signal from the output, the difference between output and the predicted output is the error.

Signal and noise :
Given the random nature of the dependent variable, we expect some noise to be part of the output which obscures the signal.

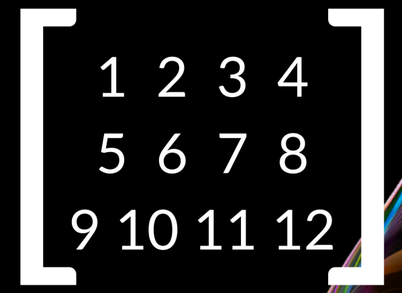
KNOWLEDGE GAINED CONTINUED..

Dependent Variable :
The output that we are trying to capture. This depends on various factors.

&

Independent Variable : The factors on which the output depends; given these factors, we want to predict the output.

Matrix :
A rectangular array of numbers, symbols, or expressions, arranged in rows and columns. The elements of a matrix are typically enclosed in square or round brackets. Matrices are used in various fields of mathematics and engineering to represent and solve linear equations, among other applications.



Confusion Matrix :
A specific table layout that allows visualization of the performance of an algorithm, typically in the context of supervised learning for classification problems. It provides a detailed breakdown of the classification results by comparing the actual (true) labels to the predicted labels generated by the model.

Thank you.