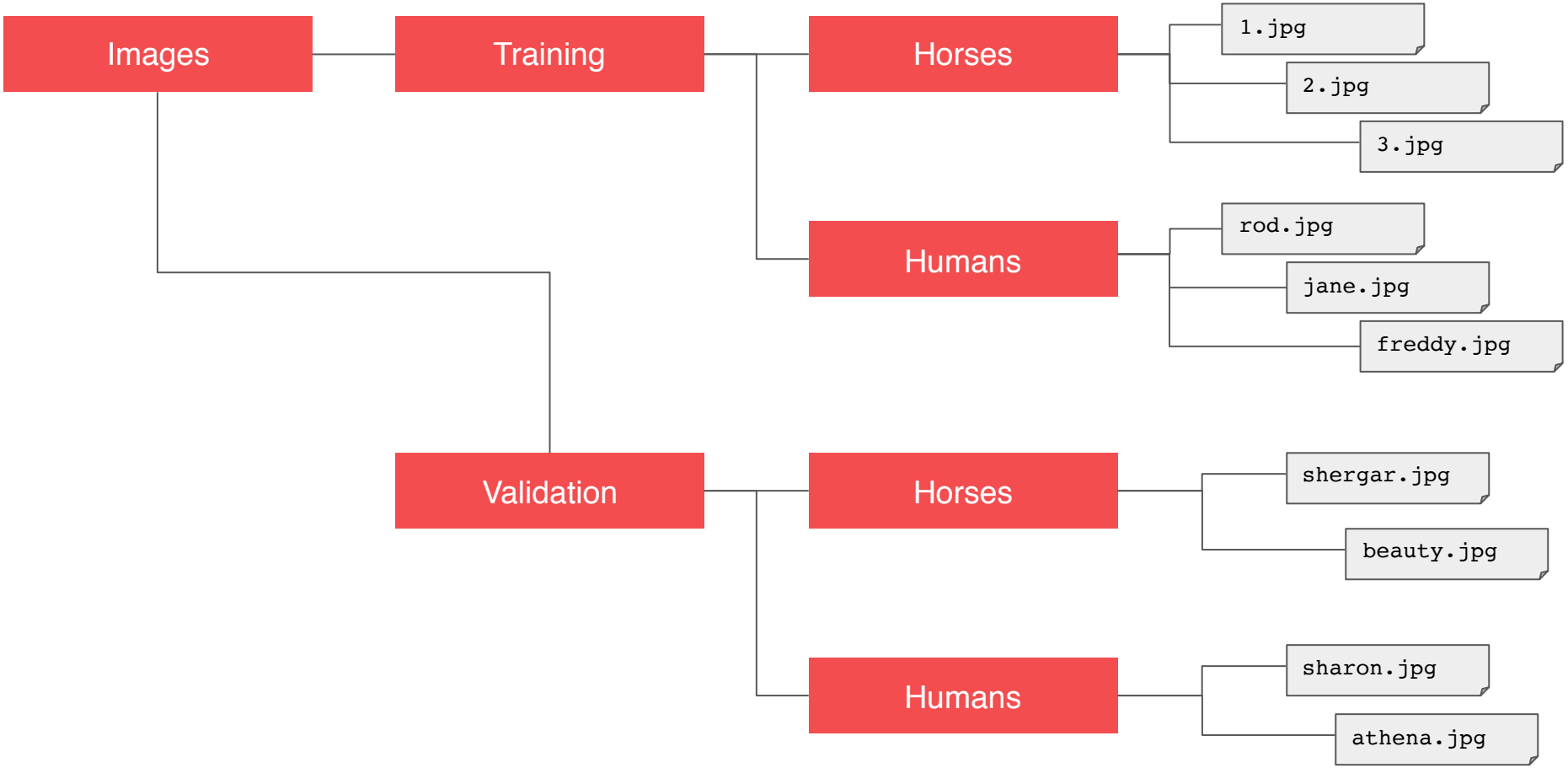
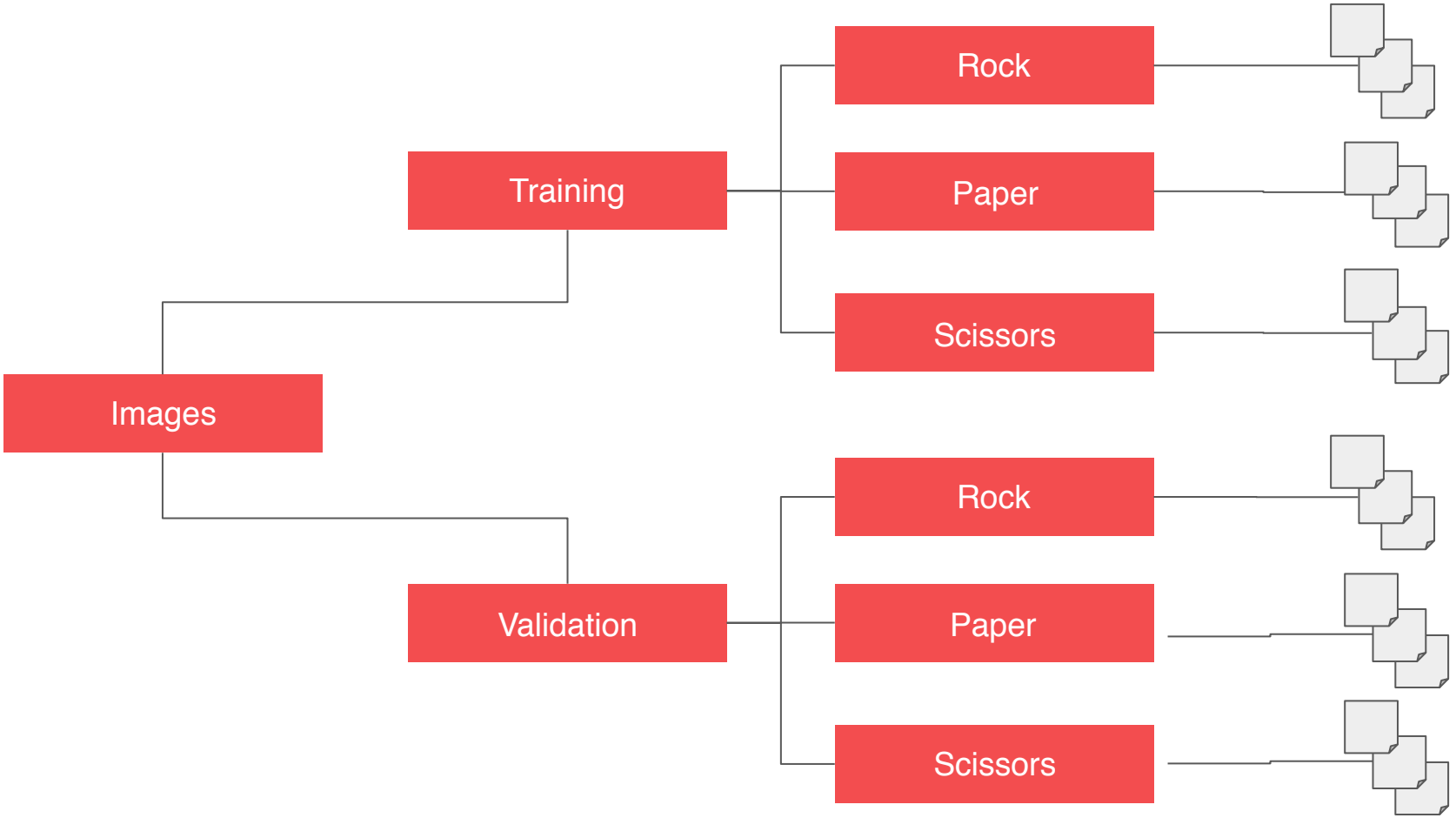
Copyright Notice


Images
Training
Horses
1.jpg
2.jpg
3.jpg
Humans
rod.jpg
jane.jpg
freddy.jpg
Validation
Horses
shergar.jpg
beauty.jpg
Humans
sharon.jpg
athena.jpg


Images
Training
Validation
Rock
Paper
Scissors
Rock
Paper
Scissors

http://www.laurencemoroney.com/rock-paper-scissors-dataset/

Rock Paper Scissors Dataset

Introducing Rock Paper Scissors – A multi class learning dataset

Abstract

Rock Paper Scissors is a dataset containing 2,892 images of diverse hands in Rock/Paper/Scissors poses. It is licensed CC By 2.0 and available for all purposes, but it's intent is primarily for learning and research.

Overview

Rock Paper Scissors contains images from a variety of different hands, from different races, ages and genders, posed into Rock / Paper or Scissors and labelled as such. You can download the training set here, and the test set here. These images have all been generated using CGI techniques as an experiment in determining if a CGI-based dataset can be

FOLLOW ME ON TWITTER

Laurence Moroney Retweeted

TensorFlow
@TensorFlow

In the third episode of the Intro to Google Colaboratory series, @lmoroney covers how to quickly build a neural network for basic Breast Cancer classification.

Watch this #CodingTensorFlow → bit.ly/2SPD4XC

P.S. Don't forget to add your homework below!

```
train_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
        train_dir,
        target_size=(300, 300),
        batch_size=128,
        class_mode='binary')
```

```
train_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
        train_dir,
        target_size=(300, 300),
        batch_size=128,
        class_mode='categorical')
```

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3,3), activation='relu',
                           input_shape=(300, 300, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3,3), activation='relu',
                           input_shape=(300, 300, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(3, activation='softmax')
])
```

Rock: 0.001

Paper: 0.647

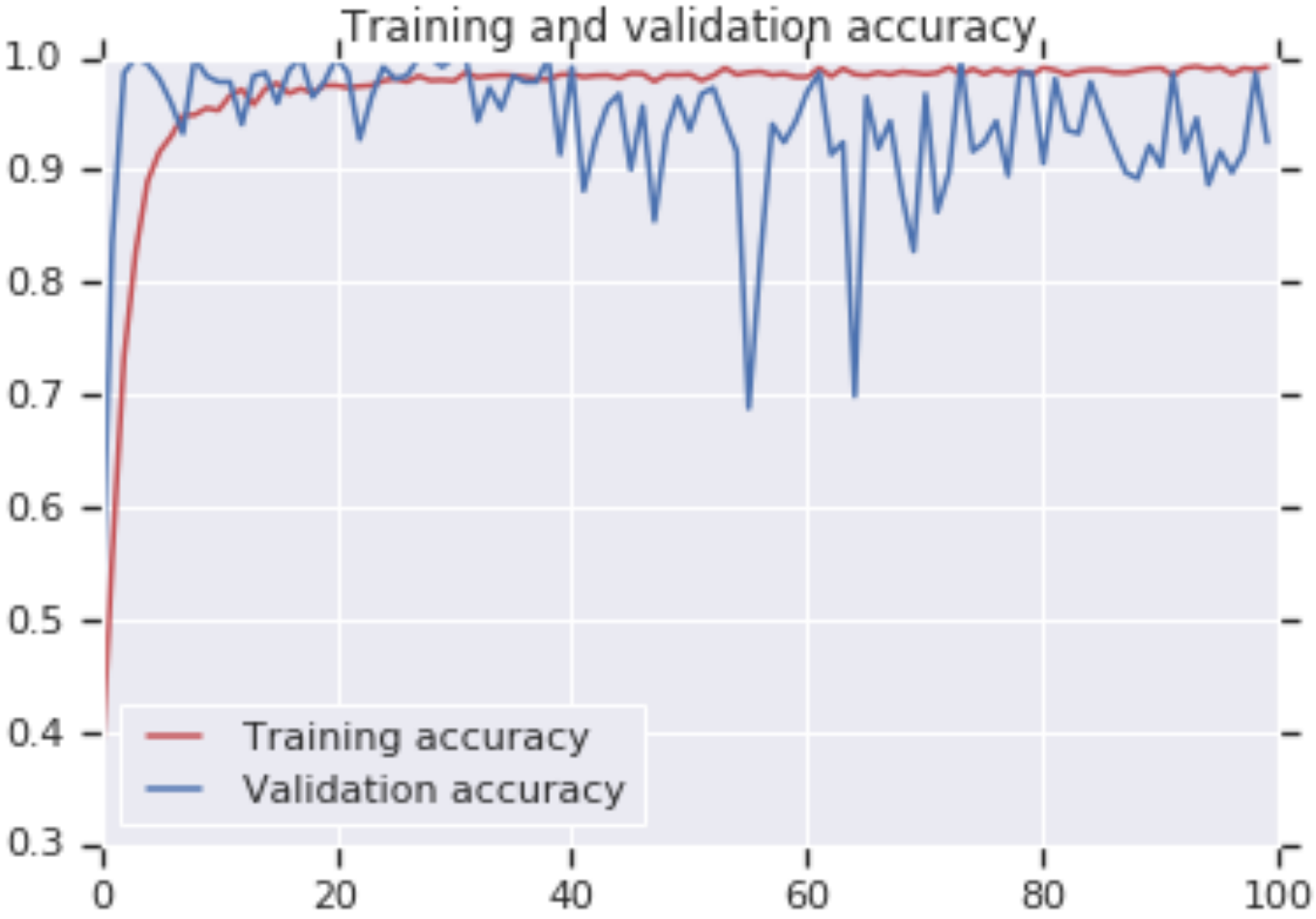
Scissors: 0.352

```
from tensorflow.keras.optimizers import RMSprop


model.compile(loss='binary_crossentropy',
              optimizer=RMSprop(lr=0.001),
              metrics=['acc'])
```

```
from tensorflow.keras.optimizers import RMSprop


model.compile(loss='categorical_crossentropy',
              optimizer=RMSprop(lr=0.001),
              metrics=['acc'])
```


Training and validation accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
20
40
60
80
100
Training accuracy
Validation accuracy