1. Configuration

1.1 Define some constants

```
GCS_PATH = 'gs://tf-everywhere-col'
DATASET_NAME = 'rock_paper_scissors'
BATCH_SIZE = 128
NUM_CLASSES = 3
DENSE_UNITS = 128
DROPOUT = 0.4
```

1.2 Validate GPU connection

```
!nvidia-smi
```

```
Fri Mar 19 00:32:45 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.56       Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   54C    P8    10W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

1.3 Authenticate

```
from google.colab import auth

auth.authenticate_user()
```

1.4 Setup dependencies

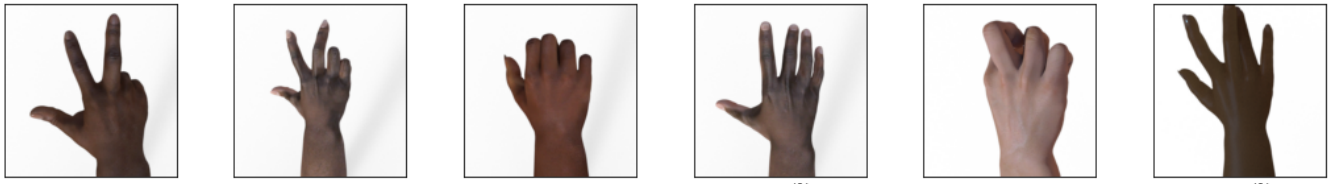
```
!pip install tensorflow_addons
import tensorflow as tf
import tensorflow_addons as tfa
import numpy as np
import matplotlib.pyplot as plt
import tensorflow_datasets as tfds
from tensorflow.keras import Model, layers, callbacks
from datetime import datetime
```

```
Collecting tensorflow_addons
  Downloading https://files.pythonhosted.org/packages/74/e3/56d2fe76f0bb7c88ed9b2a6a557e
     |████████████████████████████████| 706kB 7.6MB/s
Requirement already satisfied: typeguard>=2.7 in /usr/local/lib/python3.7/dist-packages
Installing collected packages: tensorflow-addons
Successfully installed tensorflow-addons-0.12.1
```

2. Dataset definition

```
train_ds, train_ds_info = tfds.load(DATASET_NAME,
                                    split="train",
                                    with_info=True,
                                    shuffle_files=True,
                                    data_dir=f'{GCS_PATH}/{DATASET_NAME}/train')
val_ds, val_ds_info = tfds.load(DATASET_NAME,
                                split="test",
                                with_info=True,
                                shuffle_files=False,
                                data_dir=f'{GCS_PATH}/{DATASET_NAME}/val')

tfds.visualization.show_examples(train_ds, train_ds_info, rows=2, cols=6);
```



2.1 Data Preprocessing

```
from tensorflow.data.experimental import AUTOTUNE

@tf.function
def preprocess(example, height=224, width=224):
    image = example['image']
    label = example['label']
    image = tf.image.resize_with_crop_or_pad(image, height, width)

    image = tf.keras.applications.resnet.preprocess_input(image)
    label = tf.one_hot(label, depth=NUM_CLASSES)
    return image,label

def get_datasets(train_ds, val_ds, batch_size):
    norm_train_ds = train_ds.map(preprocess, num_parallel_calls=AUTOTUNE)\
            .shuffle(500).batch(batch_size).prefetch(AUTOTUNE)
    norm_val_ds = val_ds.map(preprocess, num_parallel_calls=AUTOTUNE)\
            .batch(batch_size).cache()
    return norm_train_ds, norm_val_ds
```

2.3 Create train/val datasets

```
norm_train_ds, norm_val_ds = get_datasets(train_ds, val_ds,
                batch_size=BATCH_SIZE)
```

3. Model definition

```
def create_model(num_classes, units, dropout):

  input_layer = layers.Input(shape=(224,224,3))

  feature_extractor = tf.keras.applications.ResNet50(include_top=False, weights='imagenet', i
  feature_extractor.trainable = False

  features = layers.GlobalAveragePooling2D()(feature_extractor.output)
  x = layers.Dense(units, 'relu')(features)
  x = layers.Dropout(dropout)(x)
```

```
    x = layers.Dense(num_classes, 'softmax', dtype='float32')(x)

    model = Model(inputs=[input_layer], outputs=[x], name='feature_extractor')
    model.summary()
    return model
```

3.1 Create & Compile

Create and compile as usual

```
model = create_model(NUM_CLASSES, DENSE_UNITS, DROPOUT)
model.compile(optimizer='adam',
              loss=tf.keras.losses.CategoricalCrossentropy(label_smoothing=0.2),
              metrics=[tf.keras.metrics.CategoricalAccuracy(),
                       tfa.metrics.F1Score(num_classes=NUM_CLASSES, average='macro')])
```

```
                                                                 _      _ _  -
__________________________________________________________________________________________________
conv5_block1_out (Activation)   (None, 7, 7, 2048)   0           conv5_block1_add[0]
__________________________________________________________________________________________________
conv5_block2_1_conv (Conv2D)    (None, 7, 7, 512)    1049088     conv5_block1_out[0]
__________________________________________________________________________________________________
conv5_block2_1_bn (BatchNormali (None, 7, 7, 512)    2048        conv5_block2_1_conv
__________________________________________________________________________________________________
conv5_block2_1_relu (Activation (None, 7, 7, 512)    0           conv5_block2_1_bn[0
__________________________________________________________________________________________________
conv5_block2_2_conv (Conv2D)    (None, 7, 7, 512)    2359808     conv5_block2_1_relu
__________________________________________________________________________________________________
conv5_block2_2_bn (BatchNormali (None, 7, 7, 512)    2048        conv5_block2_2_conv
__________________________________________________________________________________________________
conv5_block2_2_relu (Activation (None, 7, 7, 512)    0           conv5_block2_2_bn[0
__________________________________________________________________________________________________
conv5_block2_3_conv (Conv2D)    (None, 7, 7, 2048)   1050624     conv5_block2_2_relu
__________________________________________________________________________________________________
conv5_block2_3_bn (BatchNormali (None, 7, 7, 2048)   8192        conv5_block2_3_conv
__________________________________________________________________________________________________
conv5_block2_add (Add)          (None, 7, 7, 2048)   0           conv5_block1_out[0]
                                                                 conv5_block2_3_bn[0
__________________________________________________________________________________________________
conv5_block2_out (Activation)   (None, 7, 7, 2048)   0           conv5_block2_add[0]
__________________________________________________________________________________________________
conv5_block3_1_conv (Conv2D)    (None, 7, 7, 512)    1049088     conv5_block2_out[0]
__________________________________________________________________________________________________
conv5_block3_1_bn (BatchNormali (None, 7, 7, 512)    2048        conv5_block3_1_conv
__________________________________________________________________________________________________
conv5_block3_1_relu (Activation (None, 7, 7, 512)    0           conv5_block3_1_bn[0
__________________________________________________________________________________________________
conv5_block3_2_conv (Conv2D)    (None, 7, 7, 512)    2359808     conv5_block3_1_relu
__________________________________________________________________________________________________
conv5_block3_2_bn (BatchNormali (None, 7, 7, 512)    2048        conv5_block3_2_conv
__________________________________________________________________________________________________
conv5_block3_2_relu (Activation (None, 7, 7, 512)    0           conv5_block3_2_bn[0
__________________________________________________________________________________________________
conv5_block3_3_conv (Conv2D)    (None, 7, 7, 2048)   1050624     conv5_block3_2_relu
```

```
conv5_block3_3_bn (BatchNormali (None, 7, 7, 2048)   8192        conv5_block3_3_conv
__________________________________________________________________________________
conv5_block3_add (Add)          (None, 7, 7, 2048)   0           conv5_block2_out[0]
                                                                 conv5_block3_3_bn[0
__________________________________________________________________________________
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0           conv5_block3_add[0]
__________________________________________________________________________________
global_average_pooling2d (Globa (None, 2048)         0           conv5_block3_out[0]
__________________________________________________________________________________
dense (Dense)                   (None, 128)          262272      global_average_pool
__________________________________________________________________________________
dropout (Dropout)               (None, 128)          0           dense[0][0]
__________________________________________________________________________________
dense_1 (Dense)                 (None, 3)            387         dropout[0][0]

==================================================================================
Total params: 23,850,371
Trainable params: 262,659
Non-trainable params: 23,587,712
__________________________________________________________________________________
```

3.2 Train & Validate model

```
%%time
history = model.fit(norm_train_ds, validation_data=norm_val_ds, epochs=20)
```

```
Epoch 1/20
20/20 [==============================] - 52s 786ms/step - loss: 1.5593 - categorical_acc
Epoch 2/20
20/20 [==============================] - 12s 504ms/step - loss: 0.6191 - categorical_acc
Epoch 3/20
20/20 [==============================] - 12s 510ms/step - loss: 0.5671 - categorical_acc
Epoch 4/20
20/20 [==============================] - 12s 508ms/step - loss: 0.5534 - categorical_acc
Epoch 5/20
20/20 [==============================] - 12s 515ms/step - loss: 0.5446 - categorical_acc
Epoch 6/20
20/20 [==============================] - 13s 531ms/step - loss: 0.5380 - categorical_acc
Epoch 7/20
20/20 [==============================] - 12s 513ms/step - loss: 0.5301 - categorical_acc
Epoch 8/20
20/20 [==============================] - 12s 512ms/step - loss: 0.5284 - categorical_acc
Epoch 9/20
20/20 [==============================] - 12s 511ms/step - loss: 0.5288 - categorical_acc
Epoch 10/20
20/20 [==============================] - 12s 513ms/step - loss: 0.5240 - categorical_acc
Epoch 11/20
20/20 [==============================] - 12s 513ms/step - loss: 0.5274 - categorical_acc
Epoch 12/20
20/20 [==============================] - 13s 518ms/step - loss: 0.5230 - categorical_acc
Epoch 13/20
20/20 [==============================] - 13s 521ms/step - loss: 0.5213 - categorical_acc
```

```
Epoch 14/20
20/20 [==============================] - 13s 525ms/step - loss: 0.5220 - categorical_acc
Epoch 15/20
20/20 [==============================] - 13s 521ms/step - loss: 0.5193 - categorical_acc
Epoch 16/20
20/20 [==============================] - 13s 519ms/step - loss: 0.5197 - categorical_acc
Epoch 17/20
20/20 [==============================] - 13s 523ms/step - loss: 0.5171 - categorical_acc
Epoch 18/20
20/20 [==============================] - 13s 527ms/step - loss: 0.5182 - categorical_acc
Epoch 19/20
20/20 [==============================] - 13s 523ms/step - loss: 0.5160 - categorical_acc
Epoch 20/20
20/20 [==============================] - 13s 524ms/step - loss: 0.5149 - categorical_acc
CPU times: user 6min 2s, sys: 21.8 s, total: 6min 24s
Wall time: 4min 49s
```