AHMAD ATTORIQ

MACHINE LEARNING PEMULA

DICODING ACADEMY

```
import tensorflow as tf
# download dataset
!wget --no-check-certificate \
        https://github.com/dicodingacademy/assets/releases/download/release/rockpaperscissors.zip \
        -O /tmp/rockpaperscissors.zip

#ekstrak dataset
import zipfile,os,shutil
local_zip = '/tmp/rockpaperscissors.zip'
zip_ref = zipfile.ZipFile(local_zip, 'r')
zip_ref.extractall('/tmp')
zip_ref.close()

# Penggunaan Callback mencegah overfitting dan menghentikan training setelah akurasi terpenuhi
class myCallback(tf.keras.callbacks.Callback):
  def on_epoch_end(self, epoch, logs={}):
    if(logs.get('accuracy') > 0.97):
      print("\nAkurasi melewati 97%, hentikan proses training!")
      self.model.stop_training = True

callbacks = myCallback()
```

```
--2023-11-23 11:55:47--  https://github.com/dicodingacademy/assets/releases/download/release/rockpaperscissors.zip
Resolving github.com (github.com)... 140.82.113.3
Connecting to github.com (github.com)|140.82.113.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/391417272/7eb836f2-695b-4a46-9c78-b65867166957?X-
--2023-11-23 11:55:48--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/391417272/7eb836f2-695b-4a46-9c78-
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.111.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 322873683 (308M) [application/octet-stream]
Saving to: ‘/tmp/rockpaperscissors.zip’

/tmp/rockpapersciss 100%[===================>] 307.92M   270MB/s    in 1.1s

2023-11-23 11:55:49 (270 MB/s) - ‘/tmp/rockpaperscissors.zip’ saved [322873683/322873683]
```

```
# split dataset kemudian membuat direktori
base_dir = '/tmp/rockpaperscissors'
train_dir = os.path.join(base_dir,'train')
validation_dir = os.path.join(base_dir, 'val')
roc_dir = os.path.join(base_dir,'rock')
pap_dir = os.path.join(base_dir, 'paper')
sci_dir = os.path.join(base_dir, 'scissors')


os.mkdir(train_dir)
os.mkdir(validation_dir)


train_roc = os.path.join(train_dir, 'rock')
train_pap = os.path.join(train_dir, 'paper')
train_sci = os.path.join(train_dir, 'scissors')
val_roc = os.path.join(validation_dir, 'rock')
val_pap = os.path.join(validation_dir, 'paper')
val_sci = os.path.join(validation_dir, 'scissors')


os.mkdir(train_roc)
os.mkdir(train_pap)
os.mkdir(train_sci)
os.mkdir(val_roc)
os.mkdir(val_pap)
os.mkdir(val_sci)
```

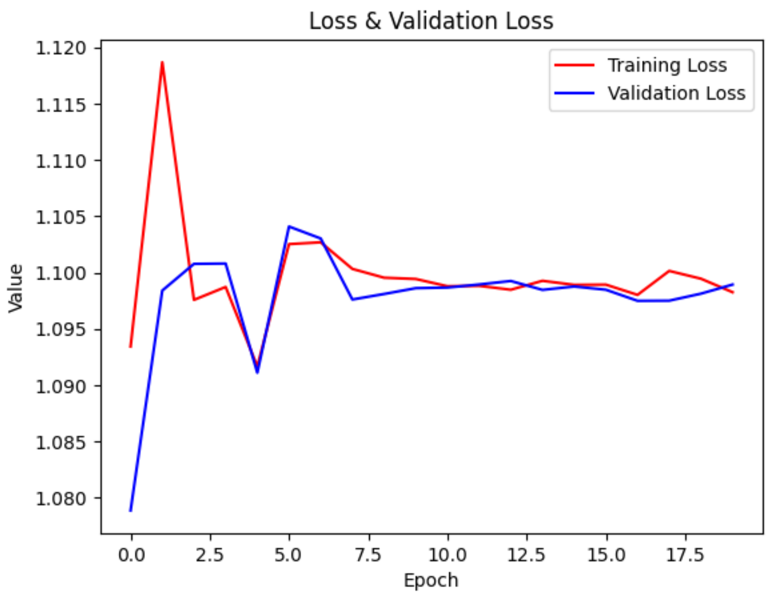
```
from sklearn.model_selection import train_test_split

train_roc_dir, val_roc_dir = train_test_split(os.listdir(roc_dir), test_size = 0.40)
train_pap_dir, val_pap_dir = train_test_split(os.listdir(pap_dir), test_size = 0.40)
train_sci_dir, val_sci_dir = train_test_split(os.listdir(sci_dir), test_size = 0.40)


for file in train_roc_dir:
  shutil.copy(os.path.join(roc_dir, file), os.path.join(train_roc, file))
for file in train_pap_dir:
  shutil.copy(os.path.join(pap_dir,file), os.path.join(train_pap,file))
for file in train_sci_dir:
  shutil.copy(os.path.join(sci_dir,file), os.path.join(train_sci,file))
for file in val_roc_dir:
  shutil.copy(os.path.join(roc_dir, file), os.path.join(val_roc,file))
for file in val_pap_dir:
  shutil.copy(os.path.join(pap_dir,file), os.path.join(val_pap,file))
for file in val_sci_dir:
  shutil.copy(os.path.join(sci_dir,file), os.path.join(val_sci,file))


from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(
                    rescale=1./255,
                    rotation_range=20,
                    horizontal_flip=True,
                    shear_range = 0.2,
                    fill_mode = 'nearest')

test_datagen = ImageDataGenerator(
                    rescale=1./255,
                    rotation_range=20,
                    horizontal_flip=True,
                    shear_range = 0.2,
                    fill_mode = 'nearest')

train_generator = train_datagen.flow_from_directory(
        train_dir,  # direktori data latih
        target_size=(150, 150),  # mengubah resolusi seluruh gambar menjadi 150x150 piksel
        batch_size=4,
        # karena ini merupakan masalah klasifikasi 3 kelas, gunakan class_mode = 'categorical'
        class_mode='categorical')

validation_generator = test_datagen.flow_from_directory(
        validation_dir, # direktori data validasi
        target_size=(150, 150), # mengubah resolusi seluruh gambar menjadi 150x150 piksel
        batch_size=4, # karena ini merupakan masalah klasifikasi 3 kelas gunakan class_mode = 'categorical'
        class_mode='categorical')

model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(32, (3,3), activation = 'relu', input_shape= (150,150,3)),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Conv2D(64,(3,3), activation= 'relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Conv2D(128,(3,3), activation= 'relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dropout(0.5),
  tf.keras.layers.Dense(512, activation= 'relu'),
  tf.keras.layers.Dense(3, activation= 'softmax')
])

model.summary()
model.compile(loss='categorical_crossentropy',
              optimizer=tf.optimizers.Adam(),
              metrics=['accuracy'])
```

```
Found 1312 images belonging to 3 classes.
Found 876 images belonging to 3 classes.
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 148, 148, 32)      896

 max_pooling2d (MaxPooling2  (None, 74, 74, 32)        0
 D)
 conv2d_1 (Conv2D)           (None, 72, 72, 64)        18496

 max_pooling2d_1 (MaxPoolin  (None, 36, 36, 64)        0
 g2D)

 conv2d_2 (Conv2D)           (None, 34, 34, 128)       73856

 max_pooling2d_2 (MaxPoolin  (None, 17, 17, 128)       0
 g2D)

 flatten (Flatten)           (None, 36992)             0

 dropout (Dropout)           (None, 36992)             0

 dense (Dense)               (None, 512)               18940416

 dense_1 (Dense)             (None, 3)                 1539

=================================================================
Total params: 19035203 (72.61 MB)
Trainable params: 19035203 (72.61 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
history = model.fit(
    train_generator,
    steps_per_epoch = 41, # 1312 images = batch_size * steps
    epochs = 20,
    validation_data = validation_generator,
    validation_steps = 27, # 876 images = batch_size * steps
    verbose =2,
      callbacks=[callbacks]
)
```

```
Epoch 1/20
41/41 - 29s - loss: 1.0934 - accuracy: 0.3659 - val_loss: 1.0789 - val_accuracy: 0.3519 - 29s/epoch - 704ms/step
Epoch 2/20
41/41 - 28s - loss: 1.1187 - accuracy: 0.2927 - val_loss: 1.0984 - val_accuracy: 0.3426 - 28s/epoch - 692ms/step
Epoch 3/20
41/41 - 28s - loss: 1.0976 - accuracy: 0.3841 - val_loss: 1.1008 - val_accuracy: 0.2500 - 28s/epoch - 686ms/step
Epoch 4/20
41/41 - 28s - loss: 1.0987 - accuracy: 0.3537 - val_loss: 1.1008 - val_accuracy: 0.3056 - 28s/epoch - 686ms/step
Epoch 5/20
41/41 - 30s - loss: 1.0917 - accuracy: 0.4207 - val_loss: 1.0911 - val_accuracy: 0.4352 - 30s/epoch - 723ms/step
Epoch 6/20
41/41 - 30s - loss: 1.1025 - accuracy: 0.3049 - val_loss: 1.1041 - val_accuracy: 0.3148 - 30s/epoch - 728ms/step
Epoch 7/20
41/41 - 30s - loss: 1.1027 - accuracy: 0.2988 - val_loss: 1.1030 - val_accuracy: 0.3056 - 30s/epoch - 726ms/step
Epoch 8/20
41/41 - 36s - loss: 1.1003 - accuracy: 0.2988 - val_loss: 1.0976 - val_accuracy: 0.3519 - 36s/epoch - 874ms/step
Epoch 9/20
41/41 - 28s - loss: 1.0995 - accuracy: 0.2927 - val_loss: 1.0981 - val_accuracy: 0.3426 - 28s/epoch - 691ms/step
Epoch 10/20
41/41 - 29s - loss: 1.0994 - accuracy: 0.2927 - val_loss: 1.0986 - val_accuracy: 0.3426 - 29s/epoch - 715ms/step
Epoch 11/20
41/41 - 29s - loss: 1.0988 - accuracy: 0.2805 - val_loss: 1.0987 - val_accuracy: 0.3056 - 29s/epoch - 713ms/step
Epoch 12/20
41/41 - 29s - loss: 1.0988 - accuracy: 0.3171 - val_loss: 1.0989 - val_accuracy: 0.2778 - 29s/epoch - 717ms/step
Epoch 13/20
41/41 - 29s - loss: 1.0985 - accuracy: 0.3720 - val_loss: 1.0993 - val_accuracy: 0.2685 - 29s/epoch - 715ms/step
Epoch 14/20
41/41 - 29s - loss: 1.0993 - accuracy: 0.2927 - val_loss: 1.0985 - val_accuracy: 0.3333 - 29s/epoch - 714ms/step
Epoch 15/20
41/41 - 28s - loss: 1.0989 - accuracy: 0.3049 - val_loss: 1.0988 - val_accuracy: 0.3796 - 28s/epoch - 679ms/step
Epoch 16/20
41/41 - 29s - loss: 1.0989 - accuracy: 0.2683 - val_loss: 1.0985 - val_accuracy: 0.3519 - 29s/epoch - 714ms/step
Epoch 17/20
41/41 - 29s - loss: 1.0980 - accuracy: 0.3780 - val_loss: 1.0975 - val_accuracy: 0.3796 - 29s/epoch - 716ms/step
Epoch 18/20
41/41 - 30s - loss: 1.1001 - accuracy: 0.2439 - val_loss: 1.0975 - val_accuracy: 0.3333 - 30s/epoch - 722ms/step
Epoch 19/20
41/41 - 29s - loss: 1.0995 - accuracy: 0.3293 - val_loss: 1.0981 - val_accuracy: 0.3796 - 29s/epoch - 715ms/step
Epoch 20/20
41/41 - 30s - loss: 1.0983 - accuracy: 0.3780 - val_loss: 1.0989 - val_accuracy: 0.3148 - 30s/epoch - 725ms/step
```

```
import matplotlib.pyplot as plt
plt.plot(history.history['loss'], 'r', label='Training Loss')
plt.plot(history.history['val_loss'], 'b', label='Validation Loss')
plt.title('Loss & Validation Loss')
plt.ylabel('Value')
plt.xlabel('Epoch')
plt.legend(loc="upper right")
plt.show()
```



```
import numpy as np
from google.colab import files
from keras.preprocessing import image
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
%matplotlib inline

uploaded = files.upload()

for fn in uploaded.keys():

  path = fn
  img = image.load_img(path, target_size =(150,150))
  imgplot = plt.imshow(img)
  x = image.img_to_array(img)
  x = np.expand_dims(x, axis=0)

  images = np.vstack([x])
  classes = model.predict(images, batch_size=10)

  print(fn)
  if classes[0,0]!=0:
    print('Gunting')
  elif classes[0,1]!=0:
    print('Batu')
  else:
    print('Kertas')
```

Pilih File WhatsApp Im…t 21.52.20.jpeg

- **WhatsApp Image 2023-11-20 at 21.52.20.jpeg**(image/jpeg) - 89789 bytes, last modified: 23/11/2023 - 100% done

```
Saving WhatsApp Image 2023-11-20 at 21.52.20.jpeg to WhatsApp Image 2023-11-20 at 21.52.20.jpeg
1/1 [==============================] - 0s 221ms/step
WhatsApp Image 2023-11-20 at 21.52.20.jpeg
Gunting
```