

adam_test

October 7, 2021

1 Tester

```
[ ]: import os
import cv2 as cv2
import numpy as np
import tensorflow as tf
from keras import models

DIRECTORY = '/home/hivini/learn/research/new-covid/'
DATASET_FOLDER = DIRECTORY + 'COVID-19_Radiography_Dataset'
IMG_SIZE = 150

model = models.load_model('/home/hivini/learn/research/new-covid/test/
↳adam_val_98_110.h5')

# def readImages(files, name):
#     if os.path.exists(name):
#         return np.load(name, allow_pickle=True)
#     data = []
#     for path_im in files:
#         try:
#             img_arr = cv2.imread(path_im, cv2.IMREAD_GRAYSCALE)
#             resized_arr = cv2.resize(img_arr, (IMG_SIZE, IMG_SIZE))
#             data.append(resized_arr) # Reshaping images to preferred size
#         except Exception as e:
#             print(e)
#     arr = np.array(data, dtype='object')
#     np.save(name, arr)
#     return arr

# covid_filenames = tf.io.gfile.glob(DATASET_FOLDER + '/COVID/*')
# normal_filenames = tf.io.gfile.glob(DATASET_FOLDER + '/Normal/*')
# covid_images = readImages(covid_filenames, DIRECTORY + 'cxr_covid.npy')
# normal_images = readImages(normal_filenames, DIRECTORY + 'cxr_normal.npy')
# normal_images = normal_images[:covid_images.shape[0]]
# print(covid_images.shape)
```

```
# print(normal_images.shape)
```

```
[ ]: # classes = ['covid', 'normal', 'viral_pneumonia']
# TEST_DATASET = '/home/hivini/learn/research/covid-test/chest_xray'

# def testImage(path):
#     img_arr = cv2.imread(path, cv2.IMREAD_GRAYSCALE)
#     resized_arr = cv2.resize(img_arr, (IMG_SIZE, IMG_SIZE))
#     resized_arr = resized_arr / 255.
#     img = np.reshape(resized_arr, [1, IMG_SIZE, IMG_SIZE, 1])
#     c = model.predict(img)
#     return np.argmax(c)

# normal_files = tf.io.gfile.glob(TEST_DATASET + '/NORMAL/*')
# pneumonia_files = tf.io.gfile.glob(TEST_DATASET + '/PNEUMONIA/VIRUS-*')

# correct = 0
# for f in normal_files:
#     if correct == 1000:
#         break
#     c = testImage(f)
#     # normal
#     if (c == 1):
#         correct += 1

# print('Correct: ', correct)
# print('Total: ', len(normal_files))
# print('Percent: ', (correct / len(normal_files)))

# img_arr = cv2.imread(TEST_DATASET + '/NORMAL/NORMAL-4512-0001.jpeg', cv2.
#     ↪IMREAD_GRAYSCALE)
# resized_arr = cv2.resize(img_arr, (IMG_SIZE, IMG_SIZE))
# resized_arr = resized_arr / 255.
# # print(resized_arr)
# img = np.reshape(resized_arr, [1, IMG_SIZE, IMG_SIZE, 1])
# c = model.predict(img)
# print(np.argmax(c))
```

```
[ ]: from keras.preprocessing.image import ImageDataGenerator

test_datagen = ImageDataGenerator(rescale=1./255)
train_datagen = ImageDataGenerator(rescale=1./255)
validation_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    '/home/hivini/learn/research/new-covid/small_dataset/train',
```

```

        target_size=(150, 150),
        batch_size=32,
        class_mode='categorical',
        color_mode='grayscale'
    )

    validation_generator = validation_datagen.flow_from_directory(
        '/home/hivini/learn/research/new-covid/small_dataset/validation',
        target_size=(150, 150),
        batch_size=32,
        class_mode='categorical',
        color_mode='grayscale'
    )

    test_generator = test_datagen.flow_from_directory(
        '/home/hivini/learn/research/new-covid/small_dataset/test',
        target_size=(150, 150),
        batch_size=32,
        class_mode='categorical',
        color_mode='grayscale'
    )

```

Found 10606 images belonging to 3 classes.

Found 2273 images belonging to 3 classes.

Found 2274 images belonging to 3 classes.

```

[ ]: test_loss, test_acc = model.evaluate(test_generator)
print("Loss on test set: ", test_loss)
print("Accuracy on test set: ", test_acc)

```

72/72 [=====] - 4s 49ms/step - loss: 0.0877 - accuracy: 0.9811

Loss on test set: 0.08765032887458801

Accuracy on test set: 0.9810906052589417

```

[ ]: test_loss, test_acc = model.evaluate(train_generator)
print("Loss on train set: ", test_loss)
print("Accuracy on train set: ", test_acc)

```

332/332 [=====] - 17s 50ms/step - loss: 2.2278e-05 - accuracy: 1.0000

Loss on train set: 2.2278152755461633e-05

Accuracy on train set: 1.0

```

[ ]: test_loss, test_acc = model.evaluate(validation_generator)
print("Loss on validation set: ", test_loss)
print("Accuracy on validation set: ", test_acc)

```

72/72 [=====] - 4s 49ms/step - loss: 0.0923 - accuracy:
0.9802
Loss on validation set: 0.09230506420135498
Accuracy on validation set: 0.9802023768424988