



kaggle.com/datasets/nikit



Search



NIK · UPDATED 3 YEARS AGO



67

New Notebook

Download

planets_dataset

Planet: Understanding the Amazon from Space

Data Code (48) Discussion (2) Metadata

Dataset Notebooks



Search notebooks

All Your Work Shared With You Bookmarks

**2c2120043StageD Hamoye**Updated 34m ago
0 comments · planets_dataset**Hamoye_Internship_STAGE_D_Project**Notebook copied with edits from Micheal Edekin · Updated 2d ago
0 comments · planets_dataset**planet_amazon_draft**Updated 3mo ago
0 comments · planets_dataset**Planet-Amazon**Updated 8mo ago
0 comments · planets_dataset**PyTorch Resnet18 [0.93]**Updated 8mo ago
Score: 0.9302 · 0 comments · Planet: Understanding the Amazon from Space +1**amazon analysis**Updated 1y ago
0 comments · planets_dataset**Amazon form Space 🌴🌿 (Classification)**Notebook copied with edits from Rough Rider 404 · Updated 1y ago
0 comments · Planet: Understanding the Amazon from Space +1**Understanding Amazon from Space**Updated 1y ago
0 comments · planets_dataset**Challenge: India_VGG16**Updated 1y ago
0 comments · planets_dataset +1



2c2120043StageD Hamoye

Notebook Data Logs Comments (0) Settings

1

Edit



In [21]:

```
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()
# fit model
history = model.fit(train_it, steps_per_epoch=len(train_it),
                    validation_data=test_it, validation_steps=len(test_it), epochs=5, verbose=1)
```

Epoch 1/5

2022-09-20 16:11:20.383612: I tensorflow/stream_executor/cuda/cuda_dnn.cc:369] Loaded cuDNN version 8005

443/443 [=====] - ETA: 0s - batch: 221.0000 - size: 63.9616 - loss: 0.0024 - fbeta: 0.9990

/opt/conda/lib/python3.7/site-packages/keras/engine/training.py:2470: UserWarning: 'Model.state_updates' will be removed in a future version. This property should not be used in TensorFlow 2.0, as 'updates' are applied automatically.

warnings.warn('Model.state_updates' will be removed in a future version.

443/443 [=====] - 14s 15ms/step - batch: 221.0000 - size: 63.9616 - loss: 0.0024 - fbeta: 0.9990 - val_loss: 9.7276e-40 - val_fb eta: 1.0000

Epoch 2/5

443/443 [=====] - 7s 15ms/step - batch: 221.0000 - size: 63.9616 - loss: 9.9885e-37 - fbeta: 1.0000 - val_loss: 9.7276e-40 - va l_fbeta: 1.0000

Epoch 3/5

443/443 [=====] - 7s 16ms/step - batch: 221.0000 - size: 63.9616 - loss: 9.9883e-37 - fbeta: 1.0000 - val_loss: 9.7276e-40 - va l_fbeta: 1.0000

Epoch 4/5

443/443 [=====] - 7s 15ms/step - batch: 221.0000 - size: 63.9616 - loss: 9.9883e-37 - fbeta: 1.0000 - val_loss: 9.7276e-40 - va l_fbeta: 1.0000

Epoch 5/5

443/443 [=====] - 7s 15ms/step - batch: 221.0000 - size: 63.9616 - loss: 9.9883e-37 - fbeta: 1.0000 - val_loss: 9.7276e-40 - va l_fbeta: 1.0000

In [22]:

```
# evaluate model
loss, fbeta = model.evaluate(test_it, steps=len(test_it), verbose=0)
print('> loss=%.3f, fbeta=%.3f' % (loss, fbeta))
```

> loss=0.000, fbeta=1.000

In [23]:

```
#save the model
model.save('/kaggle/working/model_2.h5')
```

In [24]:

```
#dividing my test_labels into two part for test-jpg and test-jpg-additional
test = sam_subs[0 : 40669]
files = sam_subs[40669 : ]
```

In [25]:

```
test_img = []

for image_name, tags in tqdm(test.values, miniters=1000):
    arr = cv2.imread('../input/planets-dataset/planet/planet/test-jpg/{}.jpg'.format(image_name))
    test_img.append(cv2.resize(arr, (64, 64)))

for image_name, tags in tqdm(files.values, miniters=1000):
    arr = cv2.imread('../input/planets-dataset/test-jpg-additional/test-jpg-additional/{}.jpg'.format(image_name))
    test_img.append(cv2.resize(arr, (64, 64)))

test_img = np.array(test_img, np.float32)/255.0
```

100%|██████████| 40669/40669 [04:05<00:00, 165.67it/s]

100%|██████████| 20522/20522 [02:04<00:00, 165.13it/s]



2c2120043StageD Hamoye

Notebook Data Logs Comments (0) Settings

1

Edit



2022-09-20 16:11:15.776337: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero

2022-09-20 16:11:15.775941: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 AVX512F FMA

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

2022-09-20 16:11:15.776374: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero

2022-09-20 16:11:15.777482: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero

2022-09-20 16:11:15.778482: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero

2022-09-20 16:11:17.719297: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero

2022-09-20 16:11:17.720850: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero

2022-09-20 16:11:17.722063: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero

2022-09-20 16:11:17.722967: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 15401 MB memory: --> device: 0, name: Tesla P100-PCIE-16GB, pci bus id: 0000:00:04.0, compute capability: 6.0

Scores are 0.574 (sklearn) and 0.574 (keras)

Evaluating a baseline model**

```
In [16]:
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(64, 64, 3)))
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(MaxPooling2D((2, 2)))
```

```
In [17]:
model.add(Flatten())
model.add(Dense(64, activation='relu', kernel_initializer='he_uniform'))
model.add(Dense(17, activation='sigmoid'))
```

```
In [18]:
from tensorflow.keras.optimizers import Adam
# compile model
opt = Adam()
model.compile(optimizer=opt, loss='binary_crossentropy', metrics=[fbeta])
```

```
In [19]:
# create data generator
datagen = ImageDataGenerator(rescale=1.0/255.0)
```

```
In [20]:
# prepare iterators
train_it = datagen.flow(X_train, Y_train, batch_size=64)
test_it = datagen.flow(X_test, Y_test, batch_size=64)
```

```
In [21]:
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()
```