face-mask-detection

January 31, 2024

0.1 Install Modules

[ ]:
```
# !pip3 install torch==1.9.0+cu102 torchvision==0.10.0+cu102 torchaudio===0.9.0
 ↪-f https://download.pytorch.org/whl/torch_stable.html
# !pip3 install detecto
# !pip3 install labelImg
```

0.2 Import Modules

[1]:
```
from PIL import Image
import os
import matplotlib.pyplot as plt
from detecto import core, utils, visualize
```

0.3 Format the Images

[2]:
```
def format_images(directory, size):
    for i, img in enumerate(os.listdir(directory)):
        im = Image.open(directory+img)
        im_resize = im.resize(size, Image.ANTIALIAS)
        im_resize.save(directory+str(i)+'.jpg')
        os.remove(directory+img)
```

[3]:
```
format_images('images/', (800, 600))
```

0.4 Label the Images

[4]:
```
# annotate the images
!labelImg
```

```
Image:C:\Users\Aswin\notebooks\Data Science Projects\Deep Learning\Face Mask
Detection - Object Detection\images\12.jpg ->
Annotation:C:/Users/Aswin/notebooks/detecto/images\12.xml
Image:C:\Users\Aswin\notebooks\Data Science Projects\Deep Learning\Face Mask
Detection - Object Detection\images\0.jpg ->
Annotation:C:/Users/Aswin/notebooks/Data Science Projects/Deep Learning/Face
Mask Detection - Object Detection/images\0.xml
```

```
Image:C:\Users\Aswin\notebooks\Data Science Projects\Deep Learning\Face Mask
Detection - Object Detection\images\1.jpg ->
Annotation:C:/Users/Aswin/notebooks/Data Science Projects/Deep Learning/Face
Mask Detection - Object Detection/images\1.xml
```

0.5 Train the Model

[5]:
```
dataset = core.Dataset('images/')
model = core.Model(['mask'])
model.fit(dataset)
```

```
  0%|
| 0/16 [00:00<?, ?it/s]

Epoch 1 of 10
Begin iterating over training dataset

100%|
    | 16/16 [00:16<00:00,  1.01s/it]
  0%|
| 0/16 [00:00<?, ?it/s]

Epoch 2 of 10
Begin iterating over training dataset

100%|
    | 16/16 [00:14<00:00,  1.07it/s]
  0%|
| 0/16 [00:00<?, ?it/s]

Epoch 3 of 10
Begin iterating over training dataset

100%|
    | 16/16 [00:14<00:00,  1.07it/s]
  0%|
| 0/16 [00:00<?, ?it/s]

Epoch 4 of 10
Begin iterating over training dataset

100%|
    | 16/16 [00:14<00:00,  1.07it/s]
  0%|
| 0/16 [00:00<?, ?it/s]

Epoch 5 of 10
Begin iterating over training dataset

100%|
    | 16/16 [00:14<00:00,  1.07it/s]
  0%|
| 0/16 [00:00<?, ?it/s]
```

```
Epoch 6 of 10
Begin iterating over training dataset

100%|
    | 16/16 [00:14<00:00,  1.07it/s]
  0%|
| 0/16 [00:00<?, ?it/s]

Epoch 7 of 10
Begin iterating over training dataset

100%|
    | 16/16 [00:14<00:00,  1.07it/s]
  0%|
| 0/16 [00:00<?, ?it/s]

Epoch 8 of 10
Begin iterating over training dataset

100%|
    | 16/16 [00:14<00:00,  1.07it/s]
  0%|
| 0/16 [00:00<?, ?it/s]

Epoch 9 of 10
Begin iterating over training dataset

100%|
    | 16/16 [00:14<00:00,  1.07it/s]
  0%|
| 0/16 [00:00<?, ?it/s]

Epoch 10 of 10
Begin iterating over training dataset

100%|
    | 16/16 [00:14<00:00,  1.07it/s]
```

0.6 Test the Model

[6]:
```
image = utils.read_image('test/test.jpg')
labels, boxes, scores = model.predict_top(image)
print(labels)
print(scores)
visualize.show_labeled_image(image, boxes, labels)
```

```
['mask']
tensor([0.9837])
```

0.7 Adding Augmentations

[8]:
```
from torchvision import transforms
augmentations = transforms.Compose([
    transforms.ToPILImage(),
    transforms.RandomHorizontalFlip(0.5),
    transforms.ToTensor(),
    utils.normalize_transform()
])

dataset = core.Dataset('images/', transform=augmentations)
loader = core.DataLoader(dataset, batch_size=2, shuffle=True)

model = core.Model(['mask'])
```

[9]:
```
losses = model.fit(loader, epochs=10, learning_rate=0.001, lr_step_size=5,␣
↪verbose=True)
```

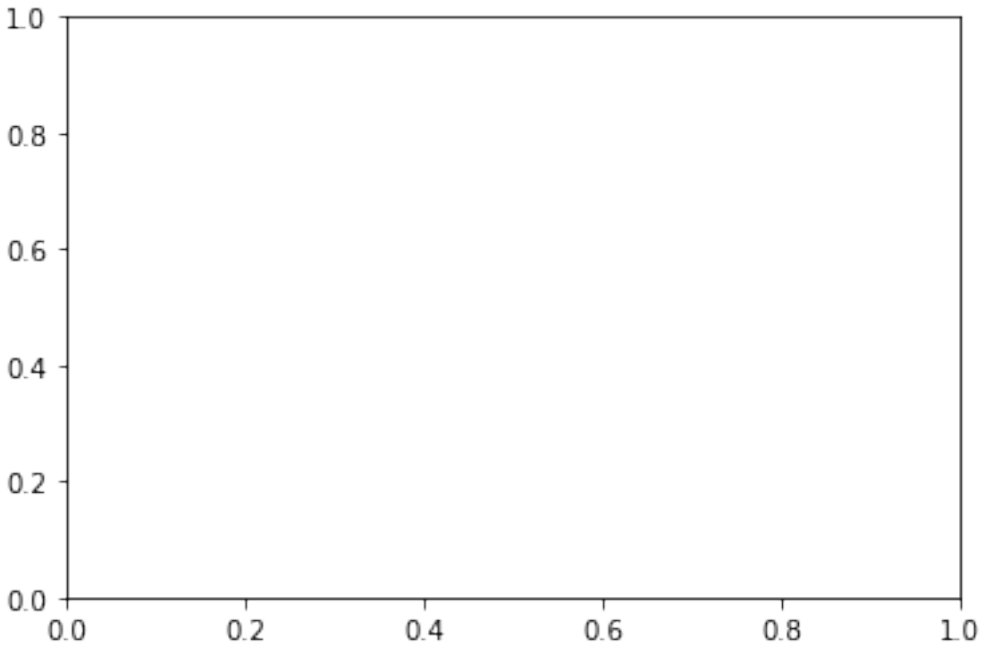
```
  0%|
| 0/8 [00:00<?, ?it/s]

Epoch 1 of 10
Begin iterating over training dataset

100%|
     | 8/8 [00:14<00:00,  1.82s/it]
```

```
  0%|
| 0/8 [00:00<?, ?it/s]

Epoch 2 of 10
Begin iterating over training dataset

100%|
     | 8/8 [00:14<00:00,  1.86s/it]
  0%|
| 0/8 [00:00<?, ?it/s]

Epoch 3 of 10
Begin iterating over training dataset

100%|
     | 8/8 [00:13<00:00,  1.66s/it]
  0%|
| 0/8 [00:00<?, ?it/s]

Epoch 4 of 10
Begin iterating over training dataset

100%|
     | 8/8 [00:13<00:00,  1.63s/it]
  0%|
| 0/8 [00:00<?, ?it/s]

Epoch 5 of 10
Begin iterating over training dataset

100%|
     | 8/8 [00:14<00:00,  1.77s/it]
  0%|
| 0/8 [00:00<?, ?it/s]

Epoch 6 of 10
Begin iterating over training dataset

100%|
     | 8/8 [00:14<00:00,  1.86s/it]
  0%|
| 0/8 [00:00<?, ?it/s]

Epoch 7 of 10
Begin iterating over training dataset

100%|
     | 8/8 [00:14<00:00,  1.86s/it]
  0%|
| 0/8 [00:00<?, ?it/s]

Epoch 8 of 10
Begin iterating over training dataset
```

```
100%|
     | 8/8 [00:14<00:00,  1.86s/it]
  0%|
| 0/8 [00:00<?, ?it/s]

Epoch 9 of 10
Begin iterating over training dataset

100%|
     | 8/8 [00:14<00:00,  1.86s/it]
  0%|
| 0/8 [00:00<?, ?it/s]

Epoch 10 of 10
Begin iterating over training dataset

100%|
     | 8/8 [00:14<00:00,  1.86s/it]
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-9-d81de906e448> in <module>
      1 losses = model.fit(loader, epochs=10, learning_rate=0.001,␣
 ↪lr_step_size=5, verbose=True)
----> 2 plt.plot(losses)
      3 plt.show()

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\pyplot.py in plot(scalex,
 ↪scaley, data, *args, **kwargs)
   2986 @_copy_docstring_and_deprecators(Axes.plot)
   2987 def plot(*args, scalex=True, scaley=True, data=None, **kwargs):
-> 2988     return gca().plot(
   2989         *args, scalex=scalex, scaley=scaley,
   2990         **({"data": data} if data is not None else {}), **kwargs)

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py in␣
 ↪plot(self, scalex, scaley, data, *args, **kwargs)
   1603         """
   1604         kwargs = cbook.normalize_kwargs(kwargs, mlines.Line2D)
-> 1605         lines = [*self._get_lines(*args, data=data, **kwargs)]
   1606         for line in lines:
   1607             self.add_line(line)

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\axes\_base.py in␣
 ↪__call__(self, data, *args, **kwargs)
    313                 this += args[0],
    314                 args = args[1:]
--> 315             yield from self._plot_args(this, kwargs)
    316
```

```
    317     def get_next_color(self):

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\axes\_base.py in␣
 ↪_plot_args(self, tup, kwargs, return_kwargs)
    460         # element array of None which causes problems downstream.
    461         if any(v is None for v in tup):
--> 462             raise ValueError("x, y, and format string must not be None"
    463
    464         kw = {}

ValueError: x, y, and format string must not be None
```



[12]:
```
image = utils.read_image('test/test.jpg')
labels, boxes, scores = model.predict_top(image)
print(labels)
print(scores)
visualize.show_labeled_image(image, boxes, labels)
```

```
['mask']
tensor([0.8742])
```

[15]:
```
image = utils.read_image('test/test2.jpg')
labels, boxes, scores = model.predict_top(image)
print(labels)
print(scores)
visualize.show_labeled_image(image, boxes, labels)
```

```
['mask']
tensor([0.9276])
```


mask
0
100
200
300
400
500
0
100
200
300
400
500
600
700

[ ]: