ex7-text-recognition

August 5, 2024

[1]:
```
import cv2
from matplotlib import pyplot as plt
import easyocr
```

[2]:
```
reader = easyocr.Reader(
    ["en"]
)  # this needs to run only once to load the model into memory
```

```
Neither CUDA nor MPS are available - defaulting to CPU. Note: This module is
much faster with a GPU.
```

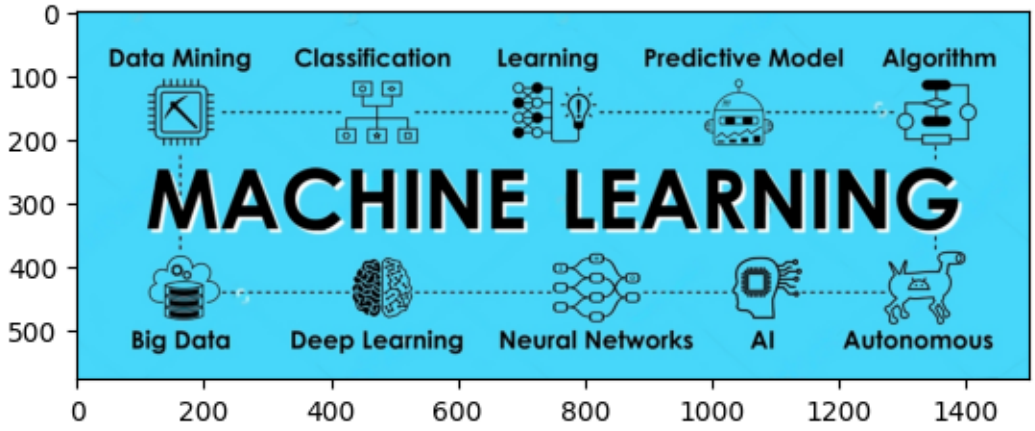
```
/opt/hostedtoolcache/Python/3.11.9/x64/lib/python3.11/site-
packages/easyocr/detection.py:78: FutureWarning: You are using `torch.load` with
`weights_only=False` (the current default value), which uses the default pickle
module implicitly. It is possible to construct malicious pickle data which will
execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for
more details). In a future release, the default value for `weights_only` will be
flipped to `True`. This limits the functions that could be executed during
unpickling. Arbitrary objects will no longer be allowed to be loaded via this
mode unless they are explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control of the
loaded file. Please open an issue on GitHub for any issues related to this
experimental feature.
  net.load_state_dict(copyStateDict(torch.load(trained_model,
map_location=device)))
/opt/hostedtoolcache/Python/3.11.9/x64/lib/python3.11/site-
packages/easyocr/recognition.py:169: FutureWarning: You are using `torch.load`
with `weights_only=False` (the current default value), which uses the default
pickle module implicitly. It is possible to construct malicious pickle data
which will execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for
more details). In a future release, the default value for `weights_only` will be
flipped to `True`. This limits the functions that could be executed during
unpickling. Arbitrary objects will no longer be allowed to be loaded via this
mode unless they are explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
```

```
`weights_only=True` for any use case where you don't have full control of the
loaded file. Please open an issue on GitHub for any issues related to this
experimental feature.
  state_dict = torch.load(model_path, map_location=device)
```

[3]:
```
IMAGE_PATH = "datasets/ocr.jpeg"
```

[4]:
```
img = cv2.imread(IMAGE_PATH)
plt.imshow(img)
```

[4]:
```
<matplotlib.image.AxesImage at 0x7fb3b0f75e50>
```

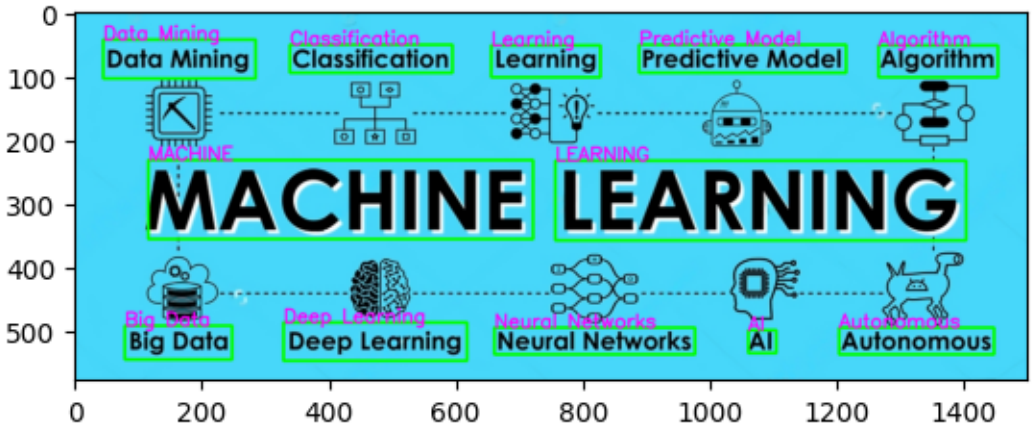


[5]:
```
result = reader.readtext(IMAGE_PATH)
img = cv2.imread(IMAGE_PATH)
for finding in result:
    top_left = tuple(finding[0][0])
    bottom_right = tuple(finding[0][2])
    text = finding[1]
    font = cv2.FONT_HERSHEY_SIMPLEX
    img = cv2.rectangle(img, top_left, bottom_right, (0, 255, 0), 3)
    img = cv2.putText(img, text, top_left, font, 1, (255, 0, 255), 2, cv2.
 ↪LINE_AA)

print([r[1] for r in result])

plt.imshow(img)
plt.show()
```

```
['Data Mining', 'Classification', 'Learning', 'Predictive Model', 'Algorithm',
'MACHINE', 'LEARNING', 'Big Data', 'Deep Learning', 'Neural Networks', 'Al',
'Autonomous']
```


Data Mining
Classification
Learning
Predictive Model
Algorithm
MACHINE
LEARNING
Big Data
Deep Learning
Neural Networks
AI
Autonomous