extract-text-from-image

March 30, 2024

[1]:
```
import pandas as pd
import numpy as np

from glob import glob
from tqdm.notebook import tqdm

import matplotlib.pyplot as plt
from PIL import Image

plt.style.use('ggplot')
```

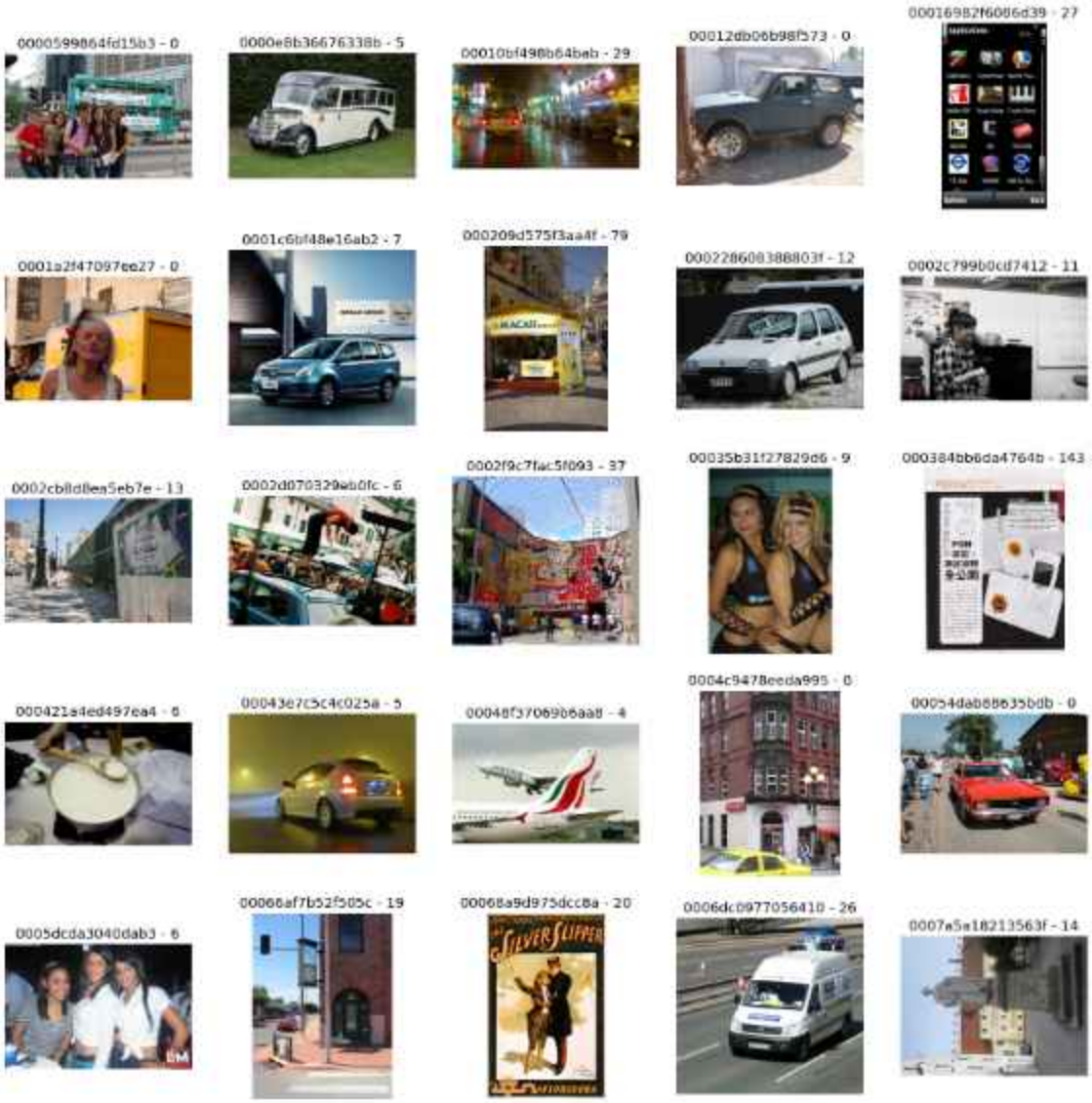
[2]:
```
annot = pd.read_parquet(r"D:\Notebook\1705\Dataset\annot.parquet")
imgs = pd.read_parquet(r"D:\Notebook\1705\Dataset\img.parquet")
img_fns = glob(r"D:\Notebook\1705\Dataset\train_val_images\train_images\*")
```

[3]:
```
fig, ax = plt.subplots(figsize=(10, 10))
ax.imshow(plt.imread(img_fns[0]))
ax.axis('off')
plt.show()
```

[4]:
```
image_id = img_fns[0].split("\\")[-1].split('.')[0]
annot.query('image_id == @image_id')
```

[4]:
```
Empty DataFrame
Columns: [id, image_id, bbox, utf8_string, points, area]
Index: []
```

[5]:
```
fig, axs = plt.subplots(5, 5, figsize=(20, 20))
axs = axs.flatten()
for i in range(25):
    axs[i].imshow(plt.imread(img_fns[i]))
    axs[i].axis('off')
    image_id = img_fns[i].split('\\')[-1].rstrip('.jpg')
    n_annot = len(annot.query('image_id == @image_id'))
    axs[i].set_title(f'{image_id} - {n_annot}')
plt.show()
```



[ ]:
```
pip install pytesseract
```

[6]:
```
import pytesseract

# Set Tesseract executable path
pytesseract.pytesseract.tesseract_cmd = r"C:\Program␣
 ↪Files\Tesseract-OCR\tesseract.exe"

# Example call
print(pytesseract.image_to_string(img_fns[11], lang='eng'))
```

[7]:
```
fig, ax = plt.subplots(figsize=(10,10))
ax.imshow(plt.imread(img_fns[11]))
ax.axis('off')
plt.show()
```

[ ]:
```
pip install easyocr
```

[8]:
```
import easyocr

reader = easyocr.Reader(['en'], gpu = True)
```

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

[9]:
```
results = reader.readtext(img_fns[11])
```

[10]:
```
pd.DataFrame(results, columns=['bbox','text','conf'])
```

[10]:

	bbox	text	conf
0	[[163, 211], [211, 211], [211, 249], [163, 249]]	en	0.787766
1	[[43.217272924571674, 165.05127422674474], [17…	Heinel	0.549905
2	[[270.32140589354253, 216.0462630228603], [413…	Helneken |	0.404501

```
3  [[484.052175279236, 276.11391864486745], [556…      Meken   0.164014
4  [[588.1675694221406, 297.0875762840607], [682…    Feheken   0.035622
```

[13]:
```
pip install keras-ocr==0.9.3
```

```
Collecting keras-ocr==0.9.3Note: you may need to restart the kernel to use
updated packages.

  Using cached keras_ocr-0.9.3-py3-none-any.whl.metadata (8.6 kB)
Requirement already satisfied: editdistance in
c:\users\megha\anaconda3\lib\site-packages (from keras-ocr==0.9.3) (0.8.1)
Requirement already satisfied: efficientnet==1.0.0 in
c:\users\megha\anaconda3\lib\site-packages (from keras-ocr==0.9.3) (1.0.0)
Requirement already satisfied: essential_generators in
c:\users\megha\anaconda3\lib\site-packages (from keras-ocr==0.9.3) (1.0)
Requirement already satisfied: fonttools in c:\users\megha\anaconda3\lib\site-
packages (from keras-ocr==0.9.3) (4.25.0)
Requirement already satisfied: imgaug in c:\users\megha\anaconda3\lib\site-
packages (from keras-ocr==0.9.3) (0.4.0)
Requirement already satisfied: pyclipper in c:\users\megha\anaconda3\lib\site-
packages (from keras-ocr==0.9.3) (1.3.0.post5)
Requirement already satisfied: shapely in c:\users\megha\anaconda3\lib\site-
packages (from keras-ocr==0.9.3) (2.0.3)
Requirement already satisfied: tqdm in c:\users\megha\anaconda3\lib\site-
packages (from keras-ocr==0.9.3) (4.65.0)
Requirement already satisfied: validators in c:\users\megha\anaconda3\lib\site-
packages (from keras-ocr==0.9.3) (0.18.2)
Requirement already satisfied: keras-applications<=1.0.8,>=1.0.7 in
c:\users\megha\anaconda3\lib\site-packages (from efficientnet==1.0.0->keras-
ocr==0.9.3) (1.0.8)
Requirement already satisfied: scikit-image in
c:\users\megha\anaconda3\lib\site-packages (from efficientnet==1.0.0->keras-
ocr==0.9.3) (0.22.0)
Requirement already satisfied: six in c:\users\megha\anaconda3\lib\site-packages
(from imgaug->keras-ocr==0.9.3) (1.16.0)
Requirement already satisfied: numpy>=1.15 in c:\users\megha\anaconda3\lib\site-
packages (from imgaug->keras-ocr==0.9.3) (1.26.4)
Requirement already satisfied: scipy in c:\users\megha\anaconda3\lib\site-
packages (from imgaug->keras-ocr==0.9.3) (1.11.4)
Requirement already satisfied: Pillow in c:\users\megha\anaconda3\lib\site-
packages (from imgaug->keras-ocr==0.9.3) (10.2.0)
Requirement already satisfied: matplotlib in c:\users\megha\anaconda3\lib\site-
packages (from imgaug->keras-ocr==0.9.3) (3.8.0)
Requirement already satisfied: opencv-python in
c:\users\megha\anaconda3\lib\site-packages (from imgaug->keras-ocr==0.9.3)
(4.9.0.80)
Requirement already satisfied: imageio in c:\users\megha\anaconda3\lib\site-
```

```
packages (from imgaug->keras-ocr==0.9.3) (2.33.1)
Requirement already satisfied: colorama in c:\users\megha\anaconda3\lib\site-
packages (from tqdm->keras-ocr==0.9.3) (0.4.6)
Requirement already satisfied: decorator>=3.4.0 in
c:\users\megha\anaconda3\lib\site-packages (from validators->keras-ocr==0.9.3)
(5.1.1)
Requirement already satisfied: h5py in c:\users\megha\anaconda3\lib\site-
packages (from keras-applications<=1.0.8,>=1.0.7->efficientnet==1.0.0->keras-
ocr==0.9.3) (3.10.0)
Requirement already satisfied: networkx>=2.8 in
c:\users\megha\anaconda3\lib\site-packages (from scikit-
image->efficientnet==1.0.0->keras-ocr==0.9.3) (3.1)
Requirement already satisfied: tifffile>=2022.8.12 in
c:\users\megha\anaconda3\lib\site-packages (from scikit-
image->efficientnet==1.0.0->keras-ocr==0.9.3) (2023.4.12)
Requirement already satisfied: packaging>=21 in
c:\users\megha\anaconda3\lib\site-packages (from scikit-
image->efficientnet==1.0.0->keras-ocr==0.9.3) (23.1)
Requirement already satisfied: lazy_loader>=0.3 in
c:\users\megha\anaconda3\lib\site-packages (from scikit-
image->efficientnet==1.0.0->keras-ocr==0.9.3) (0.3)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\megha\anaconda3\lib\site-packages (from matplotlib->imgaug->keras-
ocr==0.9.3) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
c:\users\megha\anaconda3\lib\site-packages (from matplotlib->imgaug->keras-
ocr==0.9.3) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\megha\anaconda3\lib\site-packages (from matplotlib->imgaug->keras-
ocr==0.9.3) (1.4.4)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\megha\anaconda3\lib\site-packages (from matplotlib->imgaug->keras-
ocr==0.9.3) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\megha\anaconda3\lib\site-packages (from matplotlib->imgaug->keras-
ocr==0.9.3) (2.8.2)
Using cached keras_ocr-0.9.3-py3-none-any.whl (42 kB)
Installing collected packages: keras-ocr
  Attempting uninstall: keras-ocr
    Found existing installation: keras-ocr 0.8.5
    Uninstalling keras-ocr-0.8.5:
      Successfully uninstalled keras-ocr-0.8.5
Successfully installed keras-ocr-0.9.3
```

[11]:
```
import easyocr
import pandas as pd
import numpy as np
```

```
import matplotlib.pyplot as plt
from tqdm import tqdm

# Initialize EasyOCR reader
reader = easyocr.Reader(['en'], gpu=True)

# Function to plot and compare EasyOCR results
def plot_compare(img_fn, easyocr_df):
    img_id = img_fn.split('/')[-1].split('.')[0]
    fig, axs = plt.subplots(1, 1, figsize=(15, 10))

    easy_results = easyocr_df.query('img_id == @img_id')[['text', 'bbox']].
 ↪values.tolist()
    easy_results = [(x[0], np.array(x[1])) for x in easy_results]

    image = plt.imread(img_fn)
    for text, bbox in easy_results:
        bbox = bbox.reshape(-1, 2)
        # Draw bounding box
        axs.plot(np.r_[bbox[:, 0], bbox[0, 0]], np.r_[bbox[:, 1], bbox[0, 1]],␣
 ↪'g-', linewidth=2)
        # Display text
        axs.text(bbox[0, 0], bbox[0, 1], text, va='top', fontsize=12, color='k')

    axs.imshow(image)
    axs.set_title('EasyOCR results', fontsize=24)
    plt.show()

# List to store EasyOCR results for all images
dfs = []

# Loop over images to extract text using EasyOCR
for img_fn in tqdm(img_fns[:25]):
    result = reader.readtext(img_fn)
    img_id = img_fn.split('\\')[-1].split('.')[0]
    img_df = pd.DataFrame(result, columns=['bbox', 'text', 'conf'])
    img_df['img_id'] = img_id
    dfs.append(img_df)

# Concatenate results into a single DataFrame
easyocr_df = pd.concat(dfs)

# Loop over images and plot EasyOCR results
for img_fn in img_fns[:25]:
    plot_compare(img_fn, easyocr_df)
```

```
Neither CUDA nor MPS are available - defaulting to CPU. Note: This module is
```

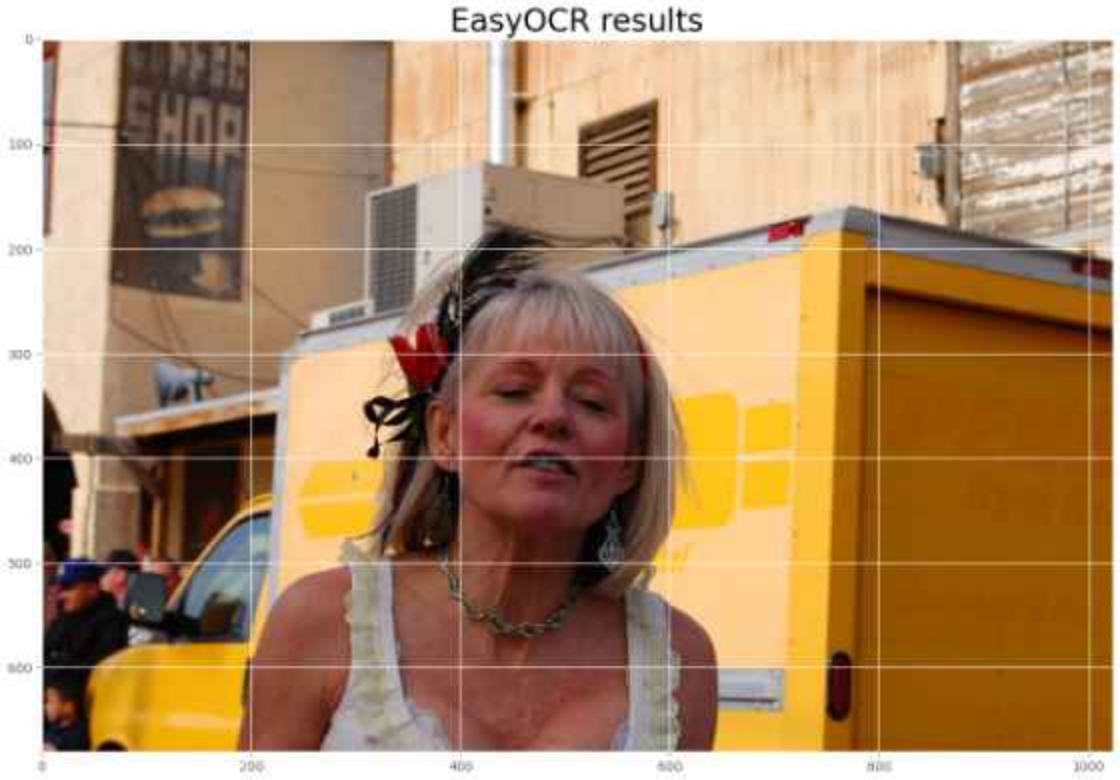
```
much faster with a GPU.
100%|
    | 25/25 [01:28<00:00,  3.54s/it]
```


EasyOCR results


EasyOCR results


EasyOCR results
708
0
100
200
300
400
500
600
700
0
200
400
600
800
1000


EasyOCR results
Applications
12:15
LightSabre
ScreenSnap
Sports Trac...
Adobe PDF
Touch Guitar
Touch Piano
UpCode
Zip
YouTube
TfL Map
WinRaR
Mail for Exc...
Options
Back
0
200
400
600
800
1000
0
100
200
300
400
500


EasyOCR results
0
100
200
300
400
500
600
0
200
400
600
800
1000


EasyOCR results
NISSAN GENISS
NISSAN
GENISS
0
200
400
600
800
1000


EasyOCR results
0
200
400
600
800
1000
0
100
200
300
400
500
600
MACAU BEER
MACAU
澳門啤酒


EasyOCR results
FOR SALE
9274 390 059
SF7533
0
100
200
300
400
500
600
700
0
200
400
600
800
1000


EasyOCR results
Percussion
0
100
200
300
400
500
600
0
200
400
600
800
1000


EasyOCR results
ERIC NETCOM
ALLO
021-35-35-32
0
100
200
300
400
500
600
0
200
400
600
800
1000

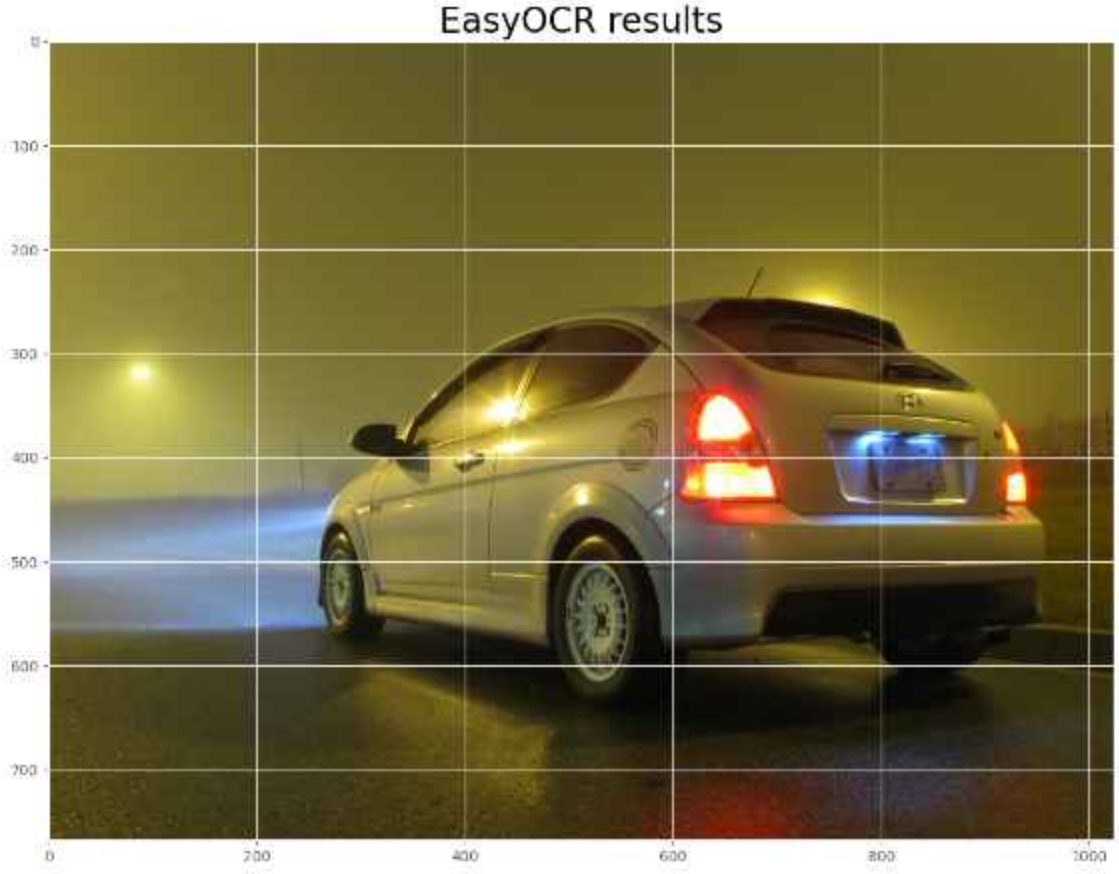

EasyOCR results
0
100
200
300
400
500
600
700
0
200
400
600
800
1000
Heineken
Heineken


EasyOCR results
0
200
400
600
800
0
200
400
600
800
1000
HELLO
HOLA

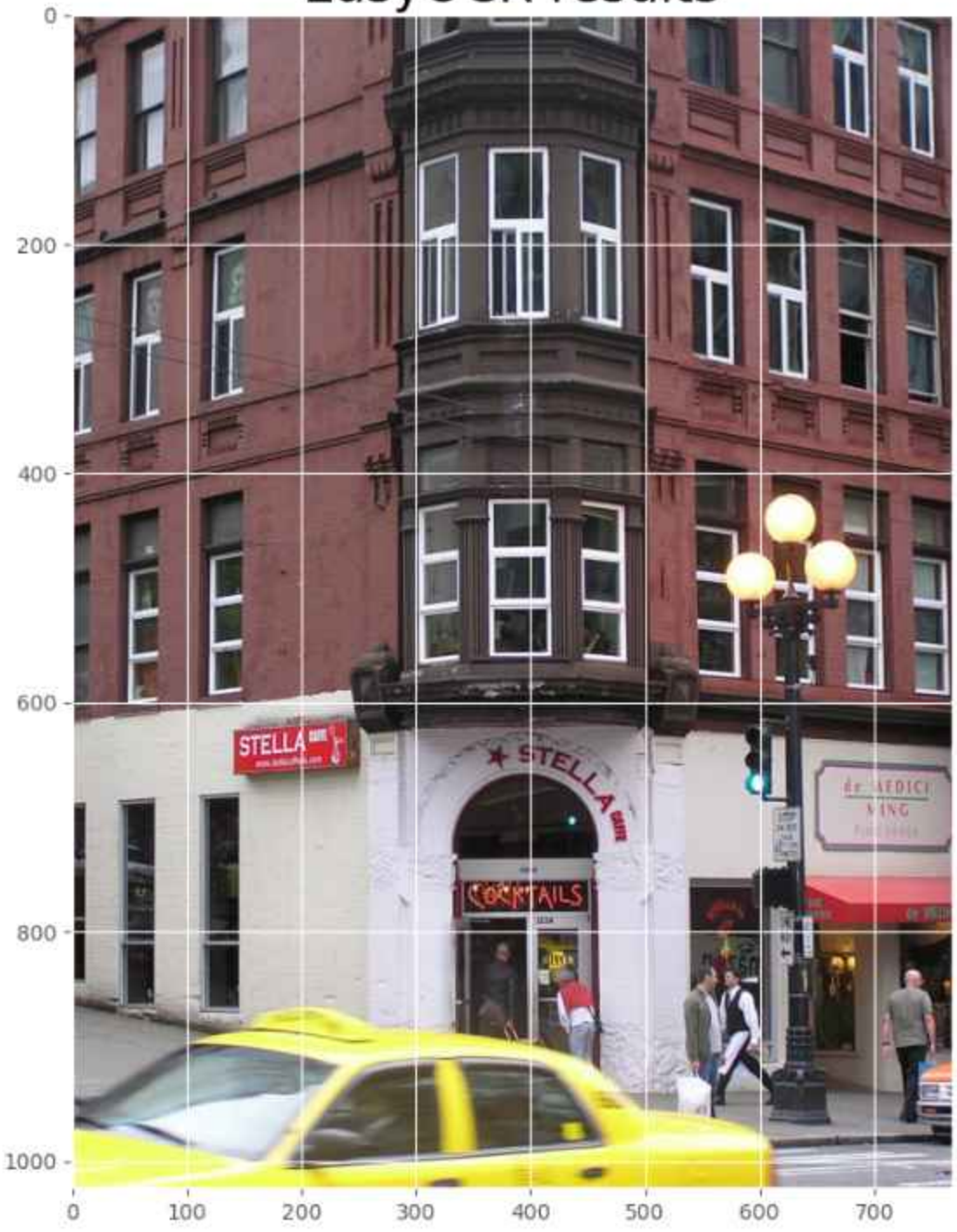

EasyOCR results
0
200
400
600
800
1000
0
100
200
300
400
500
600

EasyOCR results


EasyOCR results
0
100
200
300
400
500
600
0
200
400
600
800
1000

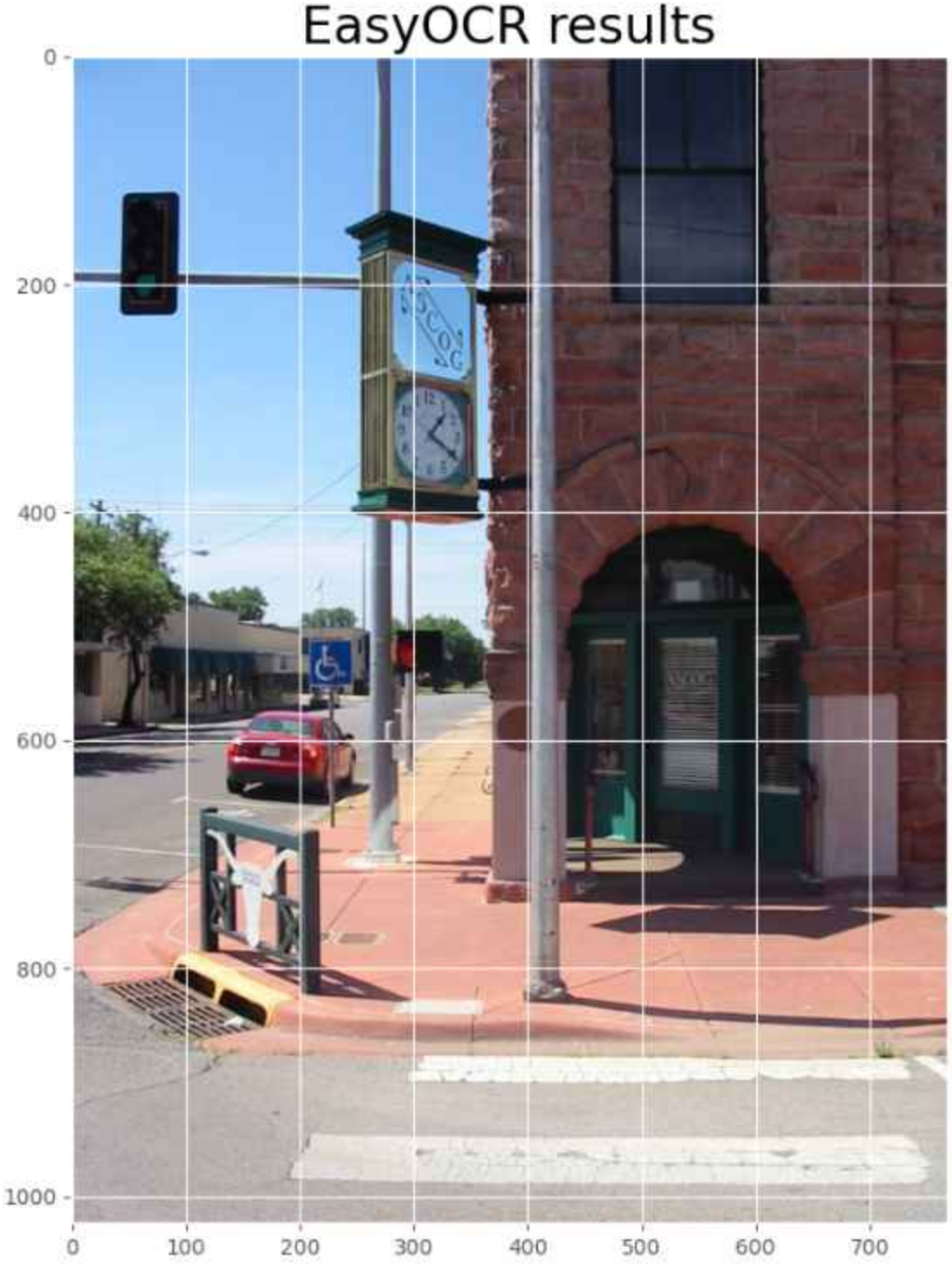

EasyOCR results
0
100
200
300
400
500
600
700
0
200
400
600
800
1000


EasyOCR results
SriLankan
www.srilankan.aero
4R-ABB
0
100
200
300
400
500
600
0
200
400
600
800
1000

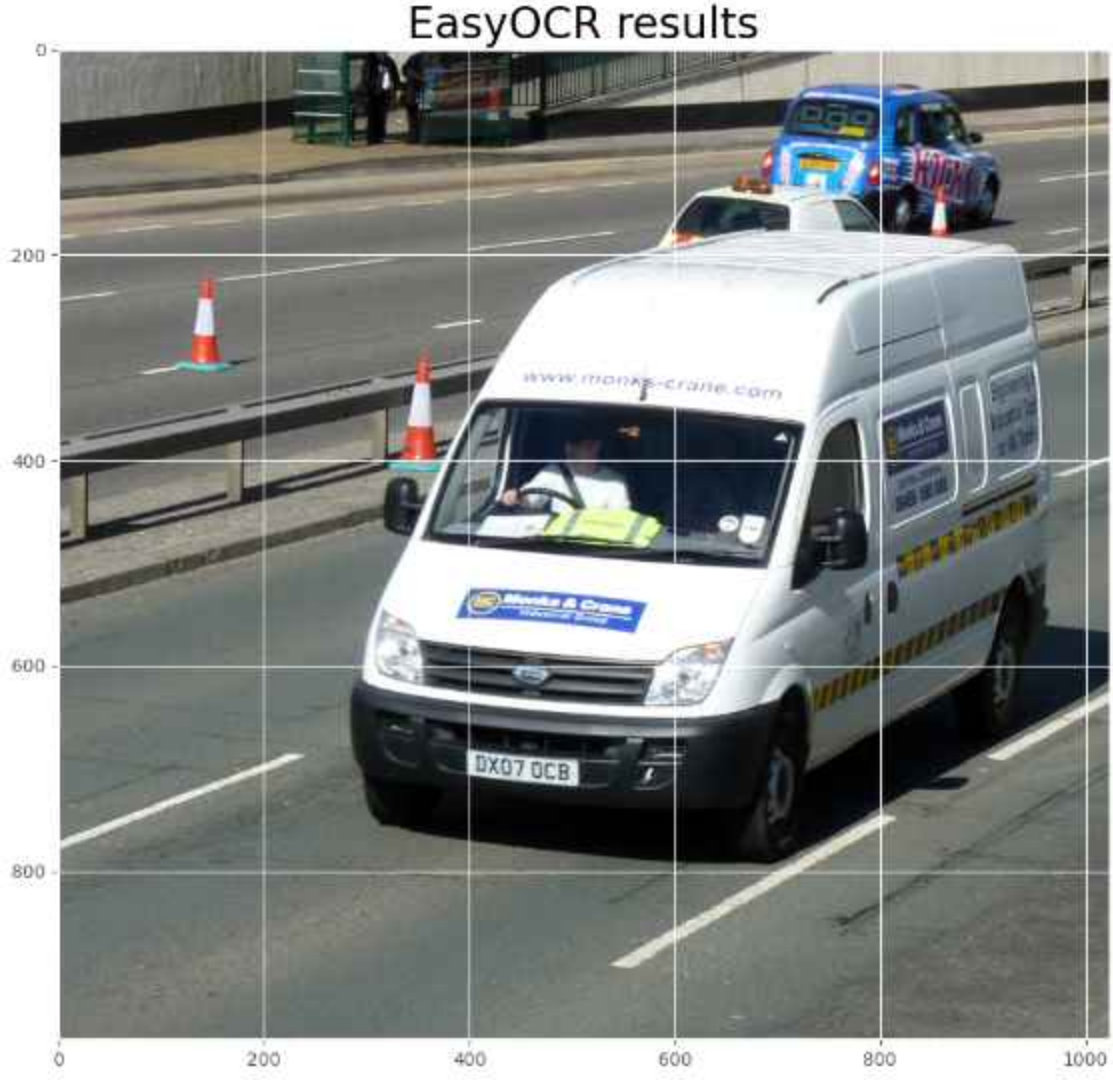

EasyOCR results
0
200
400
600
800
1000
0
100
200
300
400
500
600
700
STELLA
STELLA
COCKTAILS

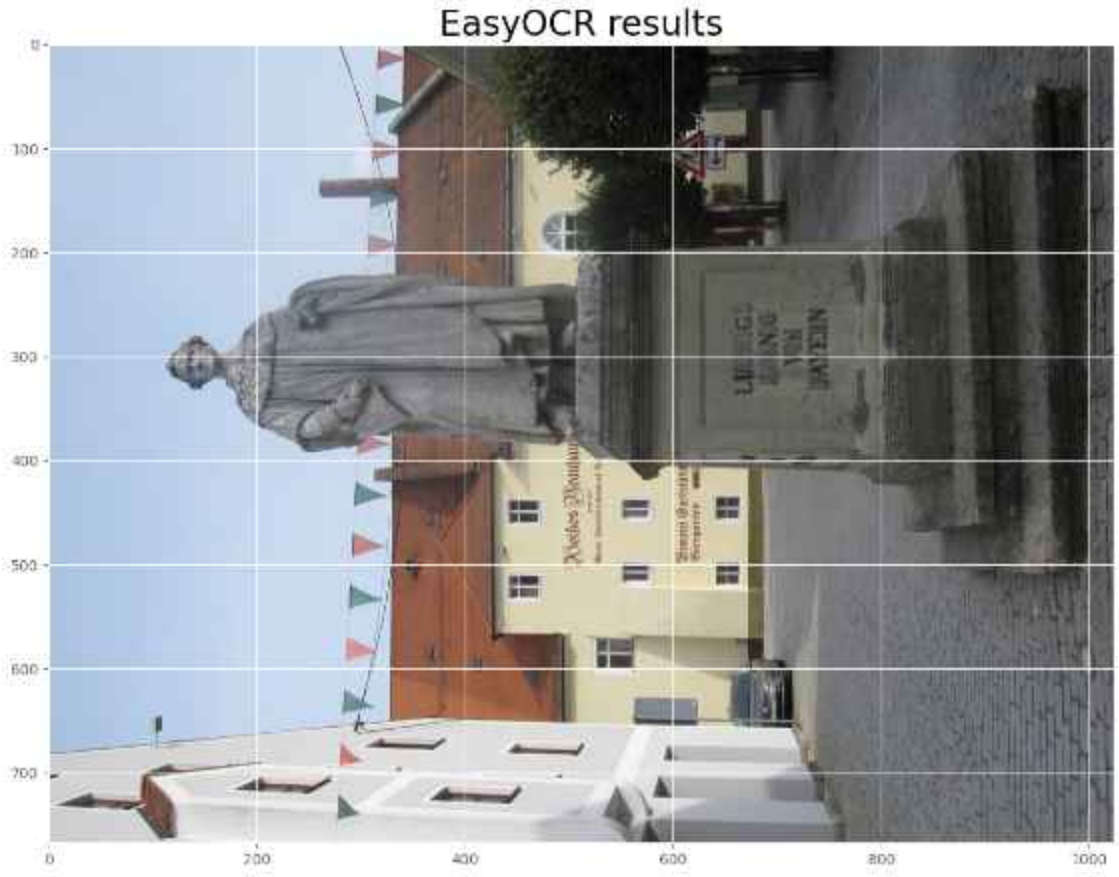

EasyOCR results
BECKER AUTO BODY
0
100
200
300
400
500
600
700
0
200
400
600
800
1000


EasyOCR results
www.LosMecanos.com
0
100
200
300
400
500
600
0
200
400
600
800
1000


EasyOCR results
0
200
400
600
800
1000
0
100
200
300
400
500
600
700

EasyOCR results


EasyOCR results
0
200
400
600
800
0
200
400
600
800
1000
DX07 OCB



[24]:
```
import easyocr
import pandas as pd
from tqdm import tqdm

# Initialize EasyOCR reader
reader = easyocr.Reader(['en'], gpu=True)

# List to store extracted text from images
extracted_text = []

# Loop over images to extract text using EasyOCR
for img_fn in tqdm(img_fns[:25]):
    result = reader.readtext(img_fn)
    img_id = img_fn.split('\\')[-1].split('.')[0]
    for bbox, text, conf in result:
        extracted_text.append({'img_id': img_id, 'text': text, 'confidence':␣
 ↪conf})

# Convert extracted text into a DataFrame
```

```
extracted_text_df = pd.DataFrame(extracted_text)

# Display extracted text
print(extracted_text_df)
```

```
Neither CUDA nor MPS are available - defaulting to CPU. Note: This module is
much faster with a GPU.
100%|
    | 25/25 [01:02<00:00,  2.52s/it]
```

	img_id	text	confidence
0	0000599864fd15b3	#	0.204666
1	0000599864fd15b3	B4*	0.144779
2	0000599864fd15b3	om.hk	0.985261
3	0000599864fd15b3	aebekae : 2926 7222 =	0.058661
4	0000e8b36676338b	IdBu eeg1	0.134325
..	…	…	…
200	0006dc0977056410	@ne	0.149766
201	0006dc0977056410	Ioooonouotocd	0.001343
202	0007a5a18213563f	3	0.475753
203	0007a5a18213563f	1	0.344312
204	0007a5a18213563f	K	0.233526

```
[205 rows x 3 columns]
```

[29]:
```
import easyocr
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from tqdm import tqdm

# Initialize EasyOCR reader
reader = easyocr.Reader(['en'], gpu=True)

def plot_easyocr_results(img_fn, easyocr_df):
    img_id = img_fn.split('\\')[-1].split('.')[0]
    fig, ax = plt.subplots(figsize=(15, 10))

    image = plt.imread(img_fn)
    ax.imshow(image)

    # Filter EasyOCR results for the specific image
    easy_results = easyocr_df[easyocr_df['img_id'] == img_id]

    # Plot bounding boxes and text
    for index, row in easy_results.iterrows():
```

```
        bbox = row['bbox']
        text = row['text']
        confidence = row['conf']

        # Check if bbox is a list or numpy array
        if isinstance(bbox, np.ndarray):
            # Convert numpy array to list
            bbox = bbox.tolist()

        # Plot bounding box
        xmin, ymin = bbox[0][0], bbox[0][1]
        width, height = bbox[2][0] - bbox[0][0], bbox[2][1] - bbox[0][1]
        rect = plt.Rectangle((xmin, ymin), width, height,
                             fill=False, edgecolor='g', linewidth=2)
        ax.add_patch(rect)

        # Plot text and confidence score
        ax.text(xmin, ymin, f'{text} (Confidence: {confidence:.2f})',
                fontsize=12, bbox=dict(facecolor='g', alpha=0.5))

    ax.set_title('EasyOCR Results', fontsize=24)
    plt.show()

# List to store EasyOCR results for all images
easyocr_results = []

# Loop over images to extract text using EasyOCR
for img_fn in tqdm(img_fns[:25]):
    img_id = img_fn.split('\\')[-1].split('.')[0]
    result = reader.readtext(img_fn)

    # Append extracted text along with image ID to the results list
    for bbox, text, conf in result:
        easyocr_results.append({'img_id': img_id, 'bbox': bbox, 'text': text,␣
 ↪'conf': conf})

# Create a DataFrame from the EasyOCR results
easyocr_df = pd.DataFrame(easyocr_results)

# Loop over images and plot EasyOCR results
for img_fn in img_fns[:25]:
    plot_easyocr_results(img_fn, easyocr_df)
```

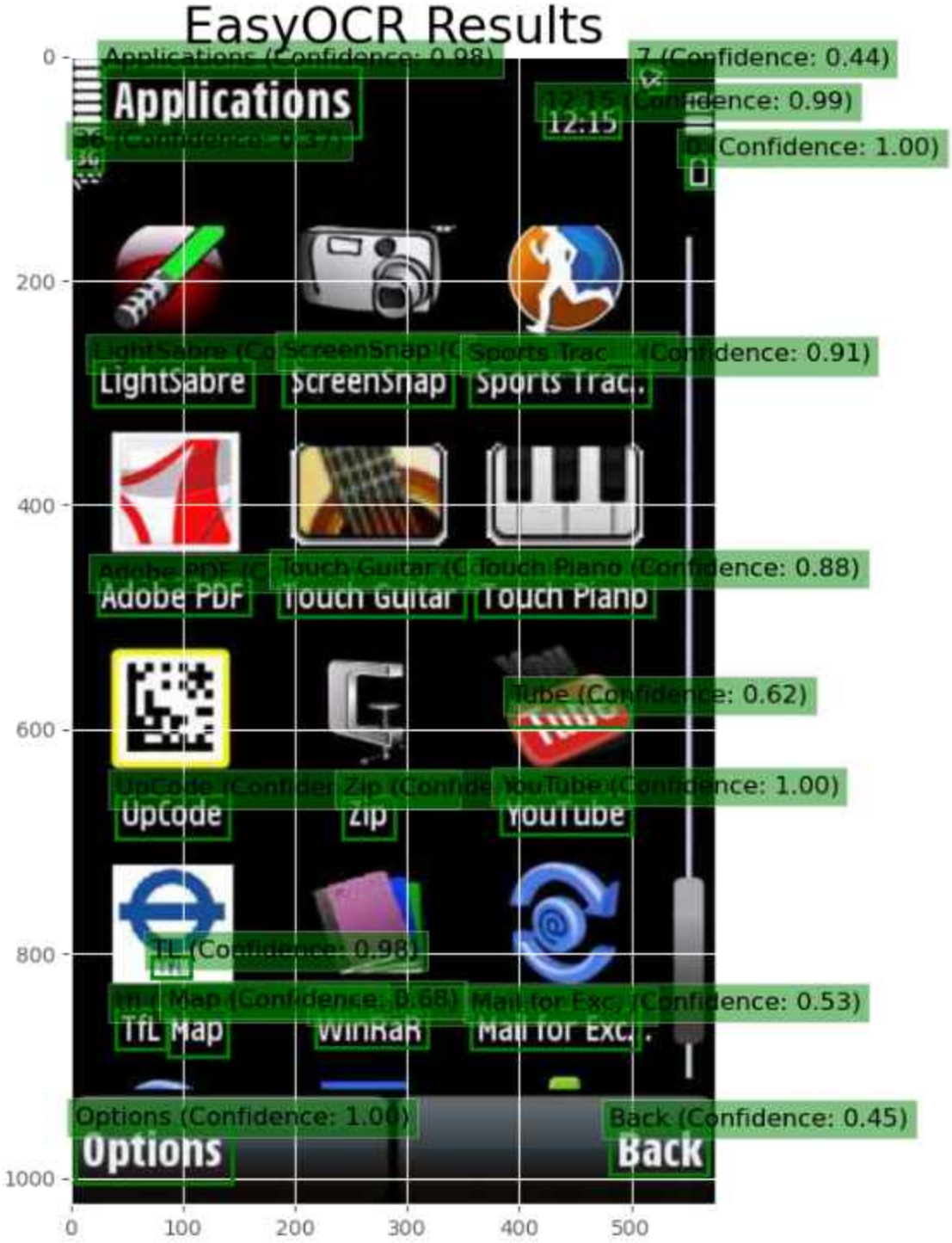
```
Neither CUDA nor MPS are available - defaulting to CPU. Note: This module is
much faster with a GPU.
100%|
    | 25/25 [01:03<00:00,  2.54s/it]
```


EasyOCR Results


EasyOCR Results
PRIVATE (Confidence: 0.58)


EasyOCR Results
0
100
200
300
400
500
0
200
400
600
800
1000

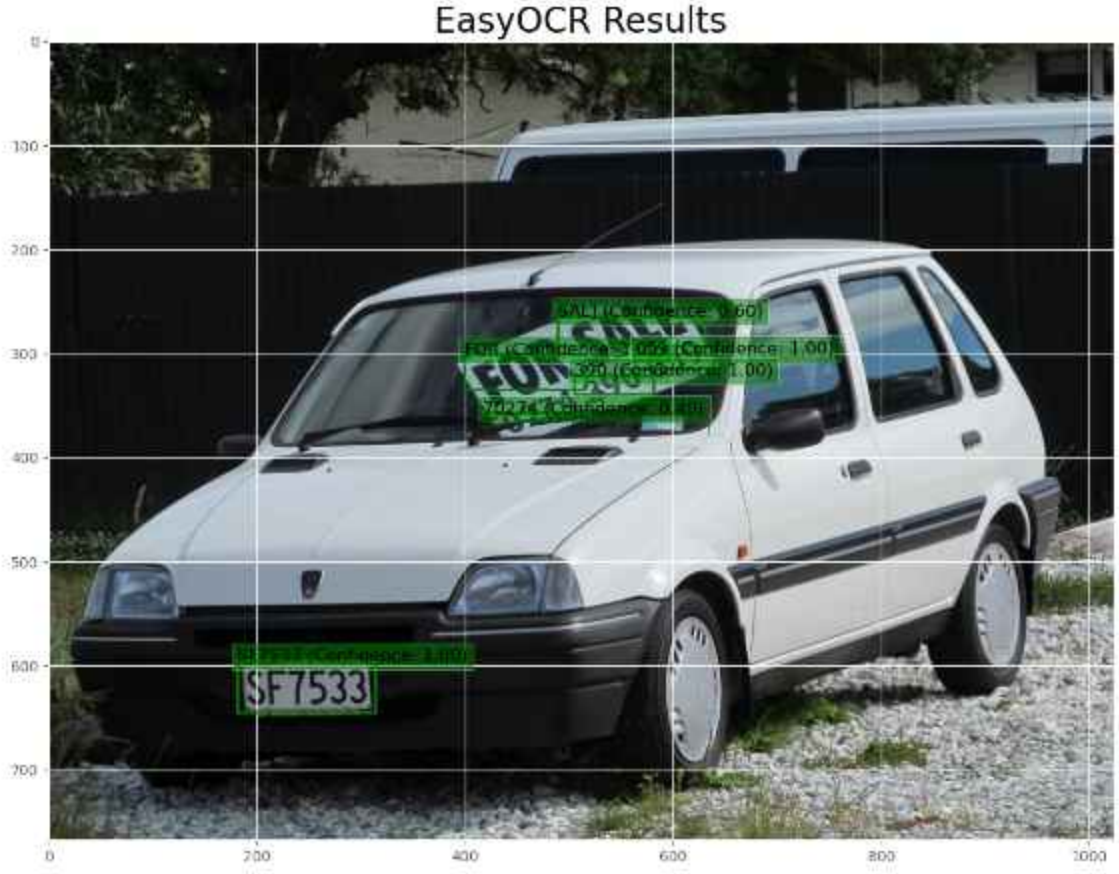

EasyOCR Results
708 (Confidence: 1.00)
708
0
100
200
300
400
500
600
700
0
200
400
600
800
1000


EasyOCR Results
Applications (Confidence: 0.98)
7 (Confidence: 0.44)
Applications
12:15
12:15 (Confidence: 0.99)
(Confidence: 1.00)
LightSabre (Co
ScreenSnap (C
Sports Trac
(Confidence: 0.91)
LightSabre
ScreenSnap
Sports Trac.
Adobe PDF (C
Touch Guitar (C
Touch Piano (Confidence: 0.88)
Adobe PDF
Touch Guitar
Touch Piano
Tube (Confidence: 0.62)
UpCode (Confider
Zip (Confide
YouTube (Confidence: 1.00)
UpCode
Zip
YouTube
TfL (Confidence: 0.98)
TfL Map (Confidence: 0.68)
Mail for Exc (Confidence: 0.53)
TfL Map
WinRAR
Mail for Exc.
Options (Confidence: 1.00)
Back (Confidence: 0.45)
Options
Back
0
200
400
600
800
1000
0
100
200
300
400
500


EasyOCR Results
0
100
200
300
400
500
600
0
200
400
600
800
1000

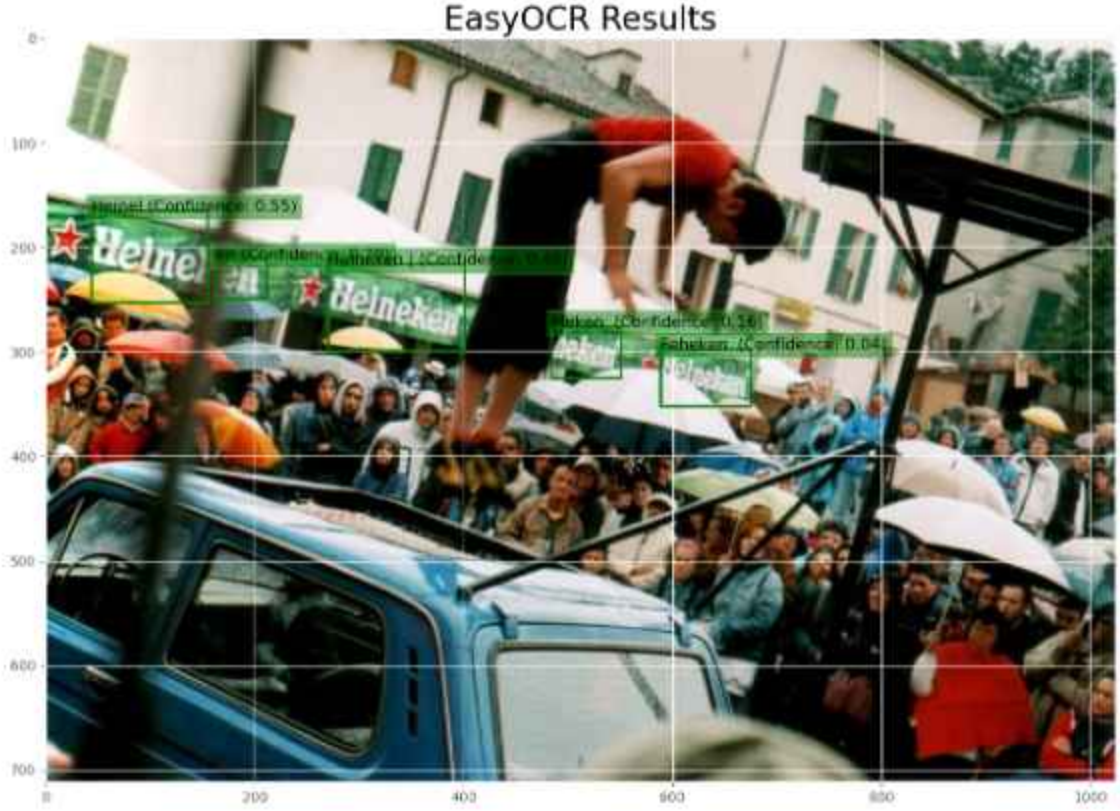

EasyOCR Results
NISSAN GENISS (Confidence:
NISSAN (Confidence: 0.98)
NISSAN GENISS
#1 (Confidence: 0.07)
GENISS (Confidence: 0.89)
GENISS
0
200
400
600
800
0
200
400
600
800
1000


EasyOCR Results
wjsodenutt (Confidence: 0.00)
MACAU (Confidence: 1.00)
MACAU (Confidence: 1.00)
wpi mex (Confidence: 0.01)
#aahzil (Confidence: 0.13)
MACAU (Confidence: 1.00)
MACN (Confidence: 0.45)
0
200
400
600
800
1000
0
100
200
300
400
500
600


EasyOCR Results
SF7533
0
100
200
300
400
500
600
700
0
200
400
600
800
1000

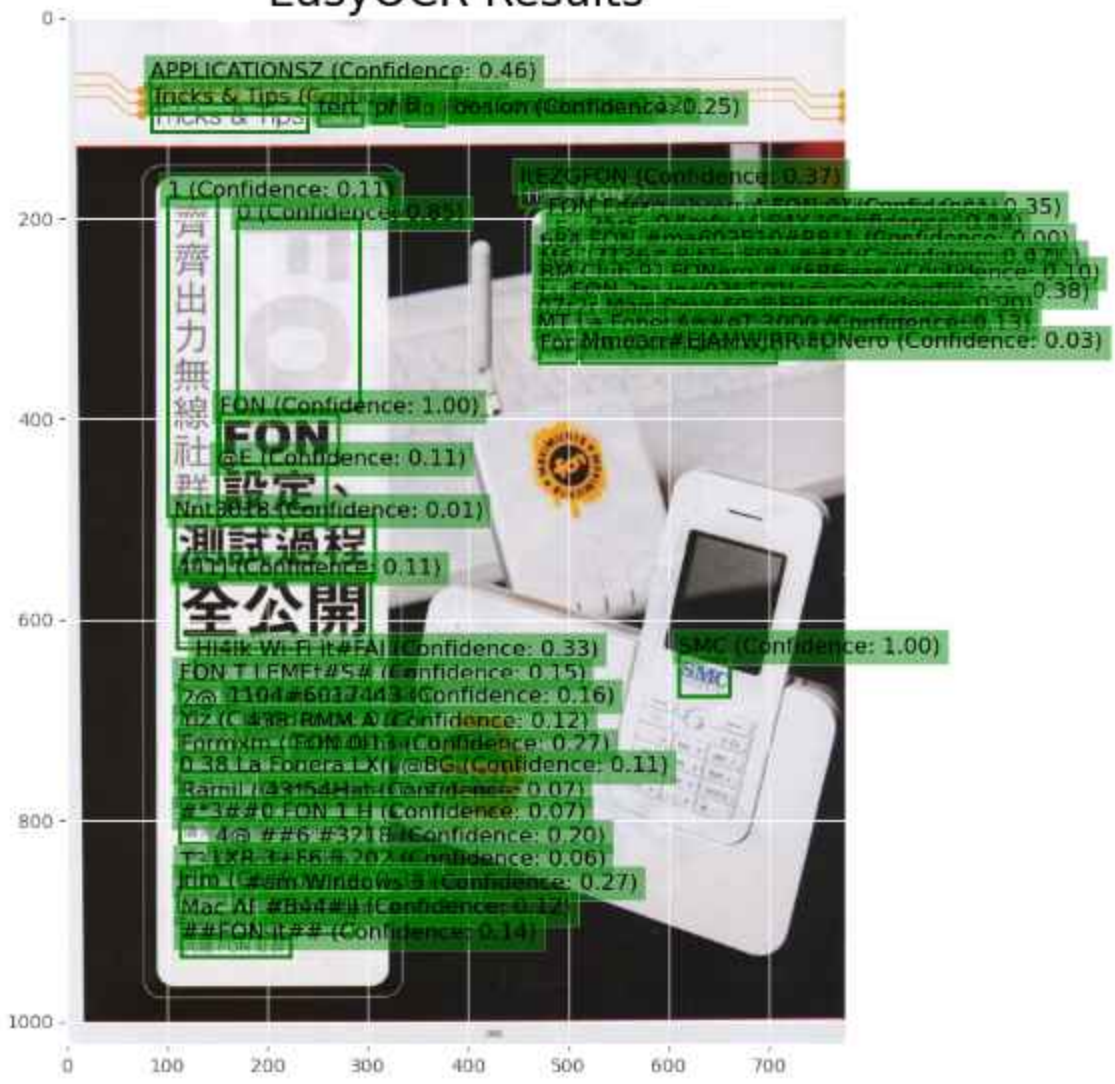

EasyOCR Results


EasyOCR Results

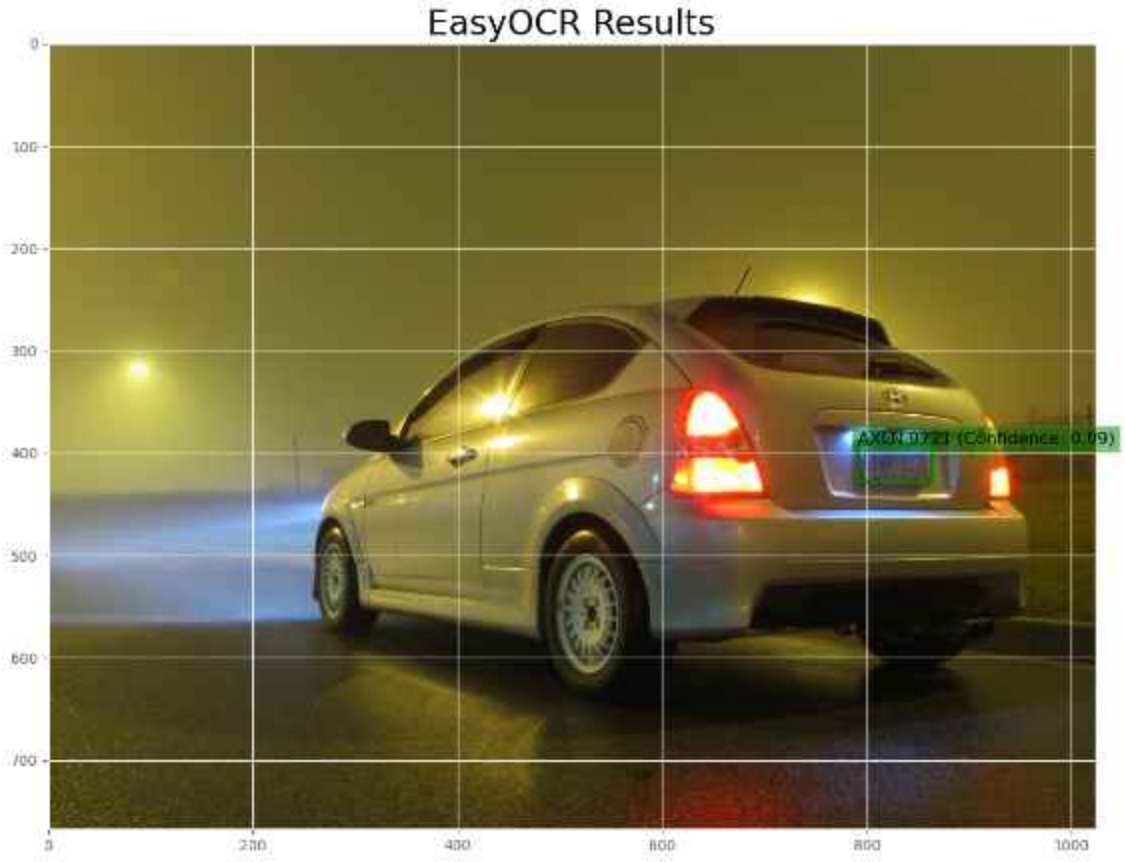

EasyOCR Results
0
100
200
300
400
500
600
700
0
200
400
600
800
1000


EasyOCR Results
Geque (Confidence: 0.11)
HELLO (Confidence: 0.65)
utena (Confidence: 0.05)
MnovbreES (Confidence: 0.18)
0
200
400
600
800
1000


EasyOCR Results
C (Jorge Mejia p (Confidence: 0.69)
VIDSON: (Confidence: 0.51)
Y-DAVIDSO (Confidence: 0.40)
Agri-Co (Confidence: 0.98)
0
200
400
600
800
1000
0
100
200
300
400
500
600

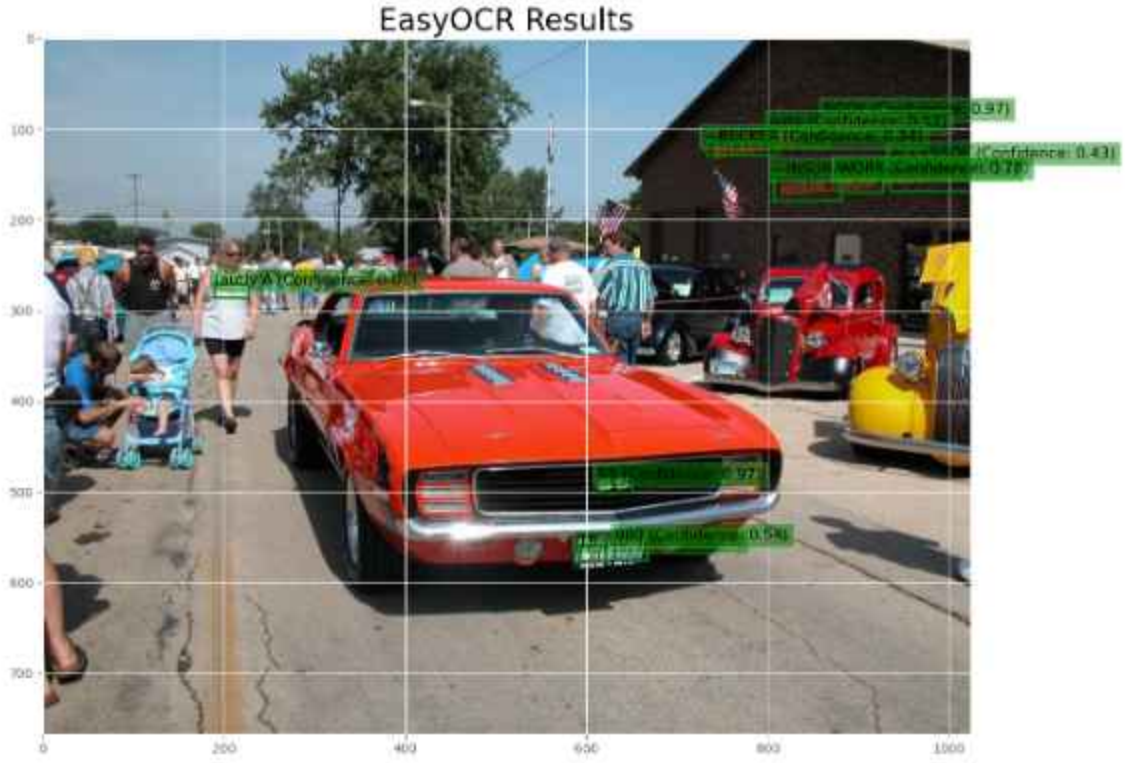

EasyOCR Results
APPLICATIONSZ (Confidence: 0.46)
1 (Confidence: 0.11)
0 (Confidence: 0.85)
FON (Confidence: 1.00)
FON
全公開
SMC (Confidence: 1.00)
0
100
200
300
400
500
600
700
200
400
600
800
1000


EasyOCR Results
0
100
200
300
400
500
600
0
200
400
600
800
1000

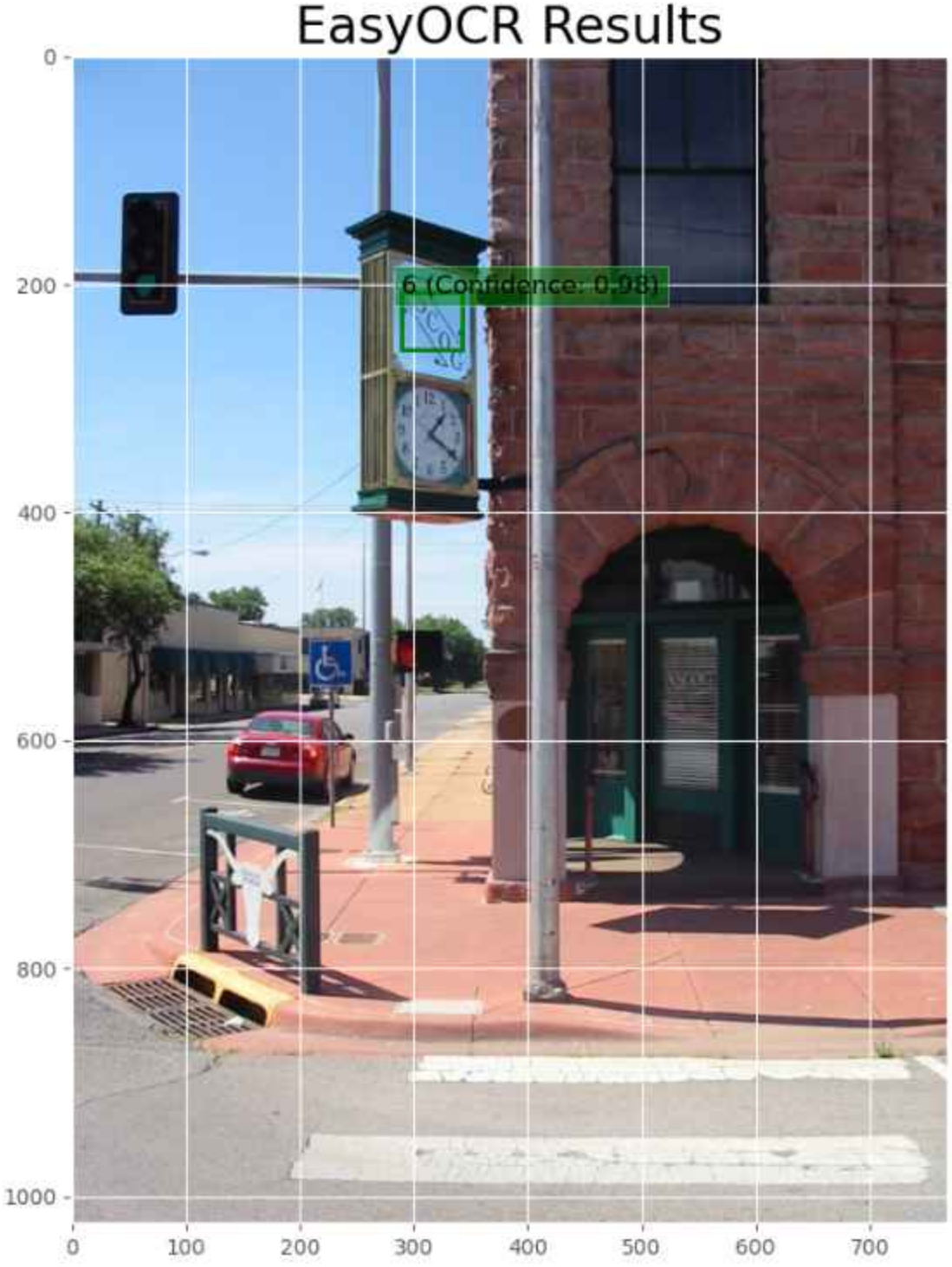

EasyOCR Results
0
100
200
300
400
500
600
700
0
200
400
600
800
1000
AX00 0721 (Confidence: 0.09)


EasyOCR Results
Srilankan (Confidence: 0.44)
www.srilankan.aero
0
100
200
300
400
500
600
0
200
400
600
800
1000

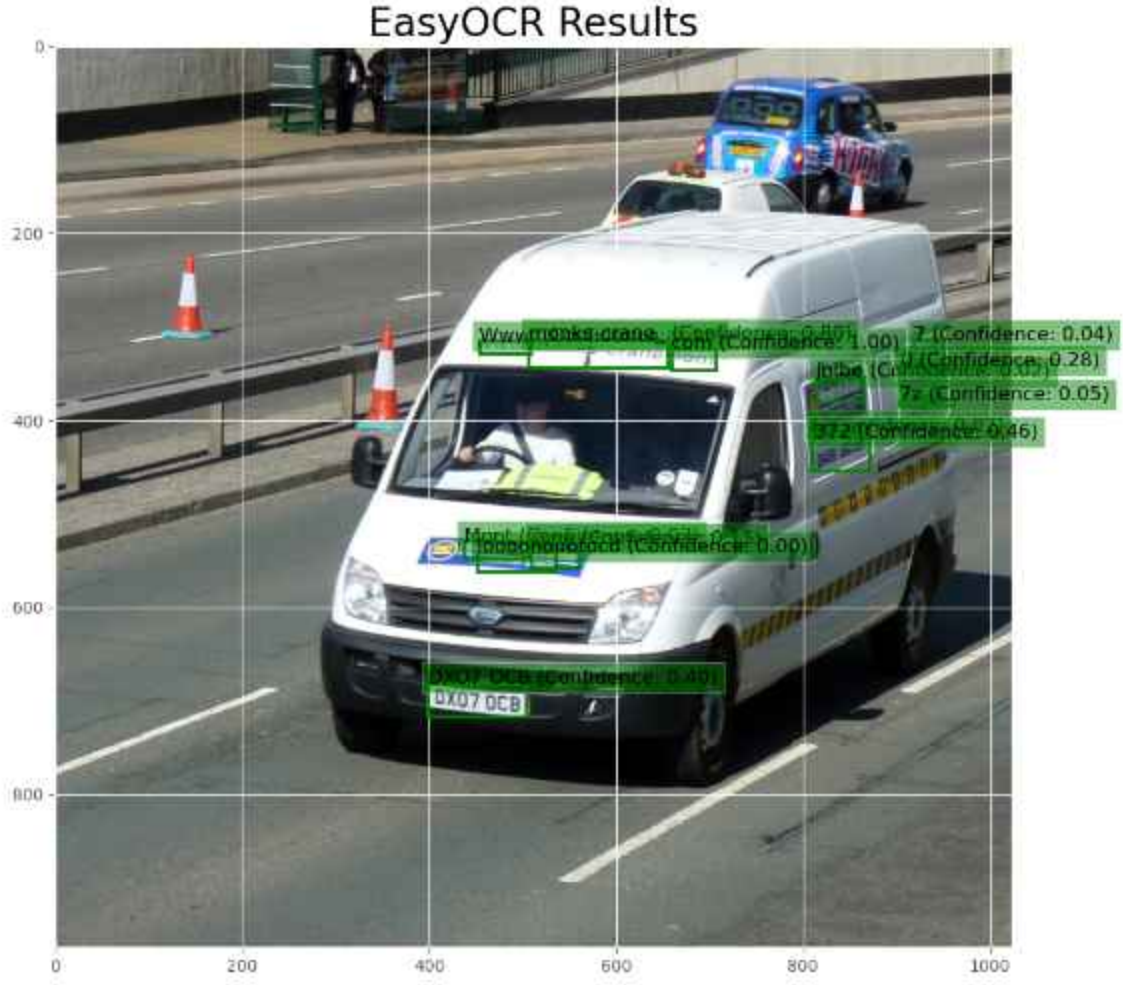

EasyOCR Results
0
200
400
600
800
1000
0
100
200
300
400
500
600
700
de MEDICI (Confidence: 0.56)
Fine Raper (Confidence: 0.01)
COCKTAILS (Confidence: 0.39)
de MEDI (Confidence: 0.69)

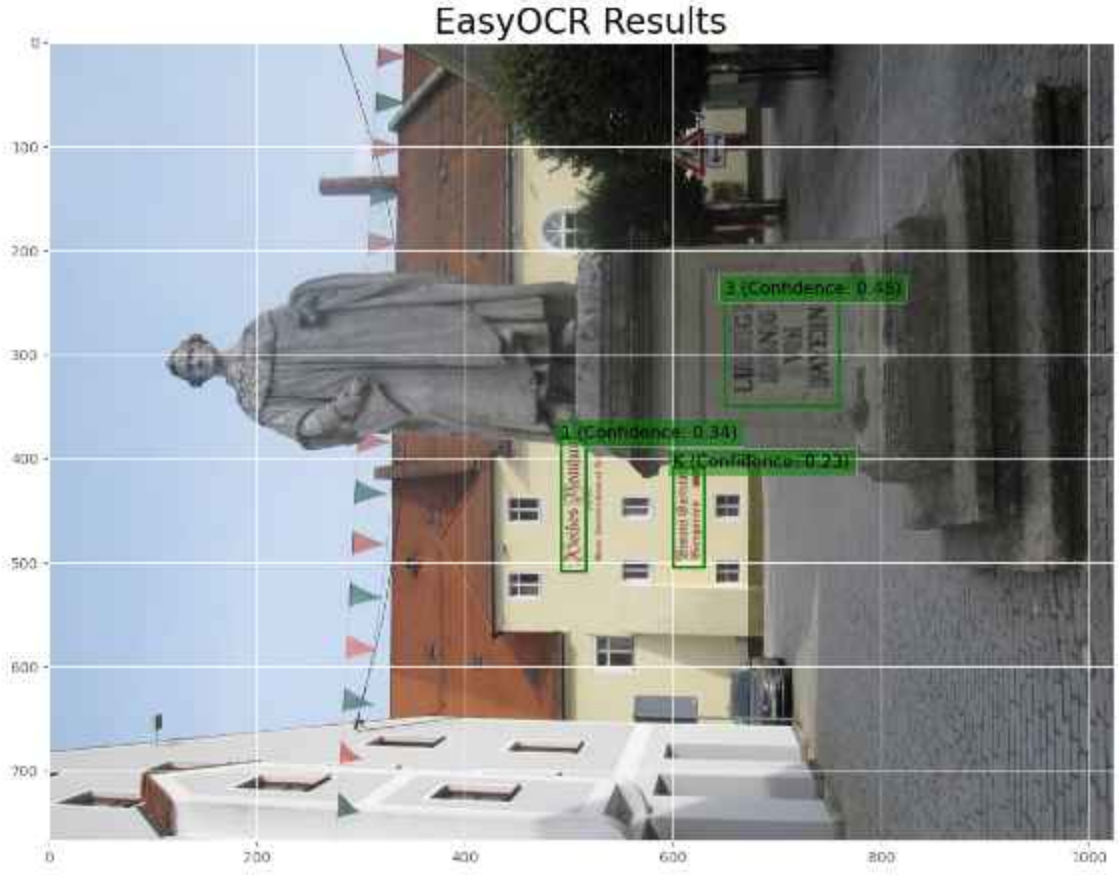

EasyOCR Results


EasyOCR Results


EasyOCR Results
6 (Confidence: 0.98)
0
200
400
600
800
1000
0
100
200
300
400
500
600
700


EasyOCR Results
JoHN (FISHERS STUPENDoUS MUSicaL PRoduction (Confidence: 0.26)
LIPPER (Confidence: 0.48)
ILVER (Confidence: 0.49)
The (Confidence: 0.37)
SLIPPER (Confidence: 0.62)
FOUND (Confidence: 0.98)
OWENHALL (Confidence: 0.68)
AuTH(SFLORODORA (Confidence: 0.45)
LESLIE STUART (Confidence: 0.97)
0
100
200
300
400
500
600
0
200
400
600
800
1000


EasyOCR Results
7 (Confidence: 0.04)
7z (Confidence: 0.05)
DX07 OCB
0
200
400
600
800
1000


EasyOCR Results
3 (Confidence: 0.55)
1 (Confidence: 0.34)
2 (Confidence: 0.23)
0
100
200
300
400
500
600
700
0
200
400
600
800
1000

[ ]: