ELECTRICAL & COMPUTER ENGINEERING

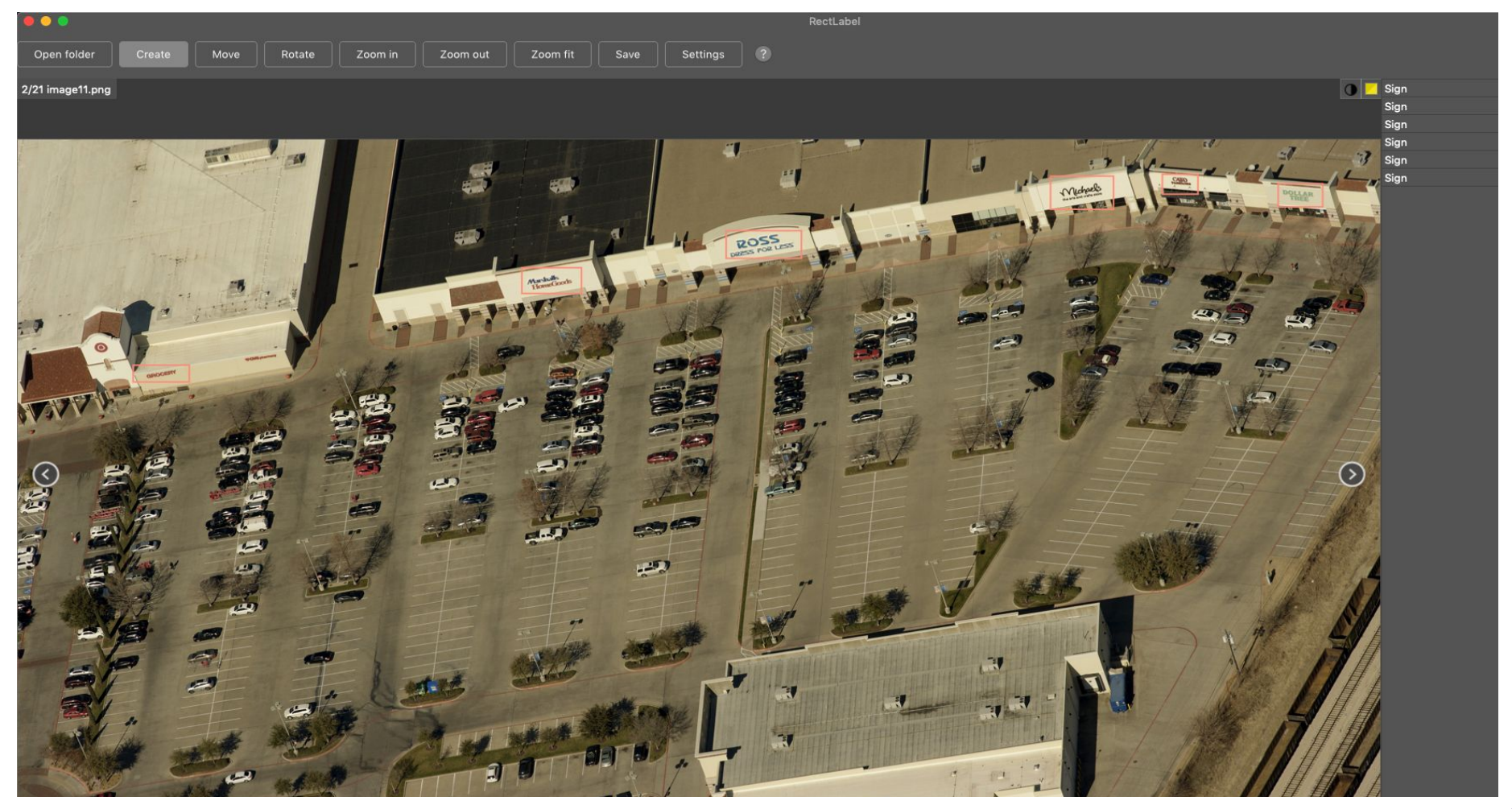
Sign Reading in Oblique Aerial Imagery

Student: Xinbei Gong, Truong Nguyen, Mengqi Chen
Faculty Adviser: Prof. Robert Bruce Darling
Industry Mentors: Shay Strong, Lilly Thomas

eagleview™

Problem Statement

- **Motivation**:
  - To perform geospatial assessment to determine the consistency of business names and locations.
- **Problem**:
  - How to extract signage location and text from an aerial imagery.
- **Approach:**
  - Two-model System:
    - An object detection model to detect signs from oblique aerial imagery.
    - A sign reading model to extract text from the detected signs.



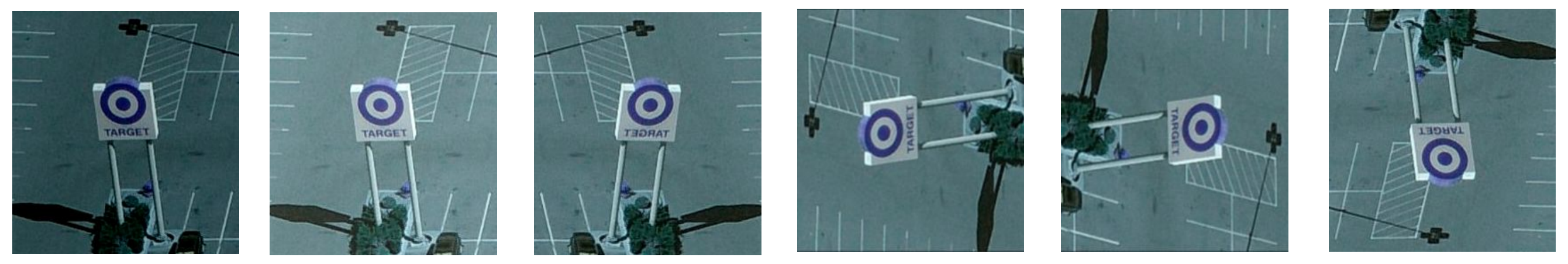
Tools

mxnet GLUON jupyter jupyterhub aws

Dataset

Sign Detection:

We started with approximately 110 cropped unique signs. Through augmentation (flipping, rotation, and brightness) we obtained a training set of 6,500 images and a test dataset of 800 images.

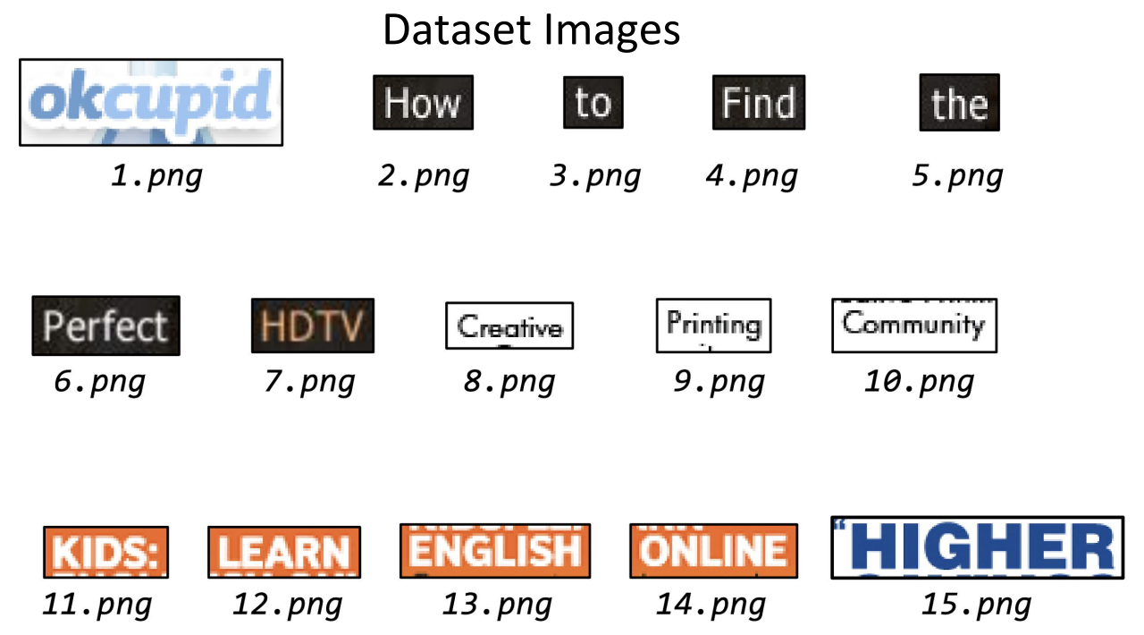


Text Recognition:

We used the ICDAR 2013 datasets to retrain a text recognition model.

ICDAR 2013 Robust Reading Competition

Dataset Images

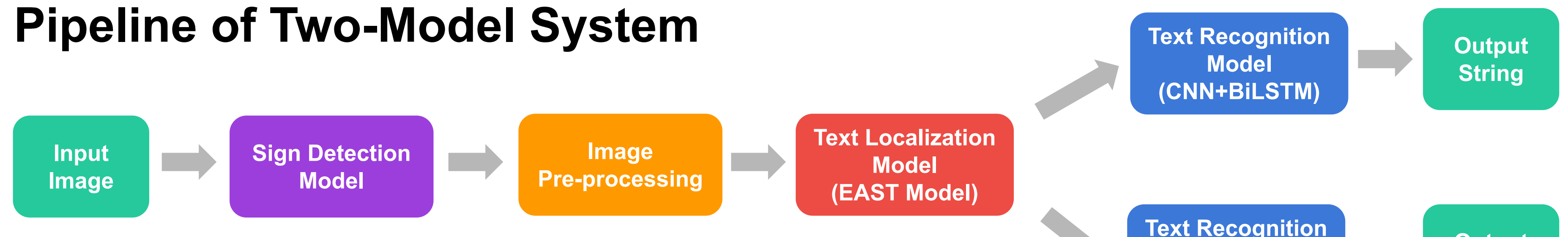


Ground Truth (single text file for all the words)

```
1.png, "okcupid"
2.png, "How"
3.png, "to"
4.png, "Find"
5.png, "the"
6.png, "Perfect"
7.png, "HDTV"
8.png, "Creative"
9.png, "Printing"
10.png, "Community"
11.png, "KIDS:"
12.png, "LEARN"
13.png, "ENGLISH"
14.png, "ONLINE"
15.png, "\"HIGHER "
```

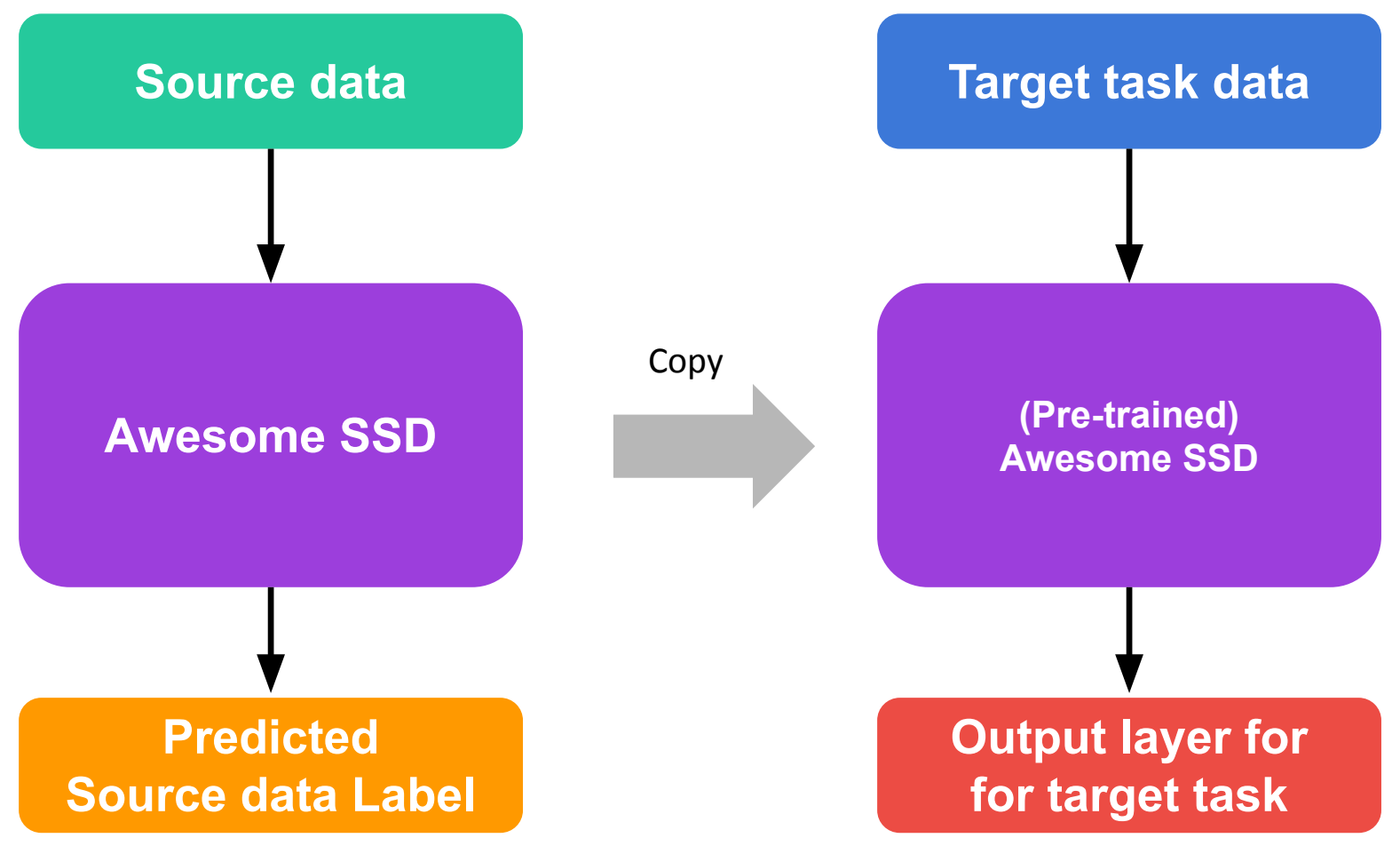
Method

Pipeline of Two-Model System


Input Image
Sign Detection Model
Image Pre-processing
Text Localization Model (EAST Model)
Text Recognition Model (CNN+BiLSTM)
Output String
Text Recognition Model (Pytesseract)
Output String

Step 1: Object Detection

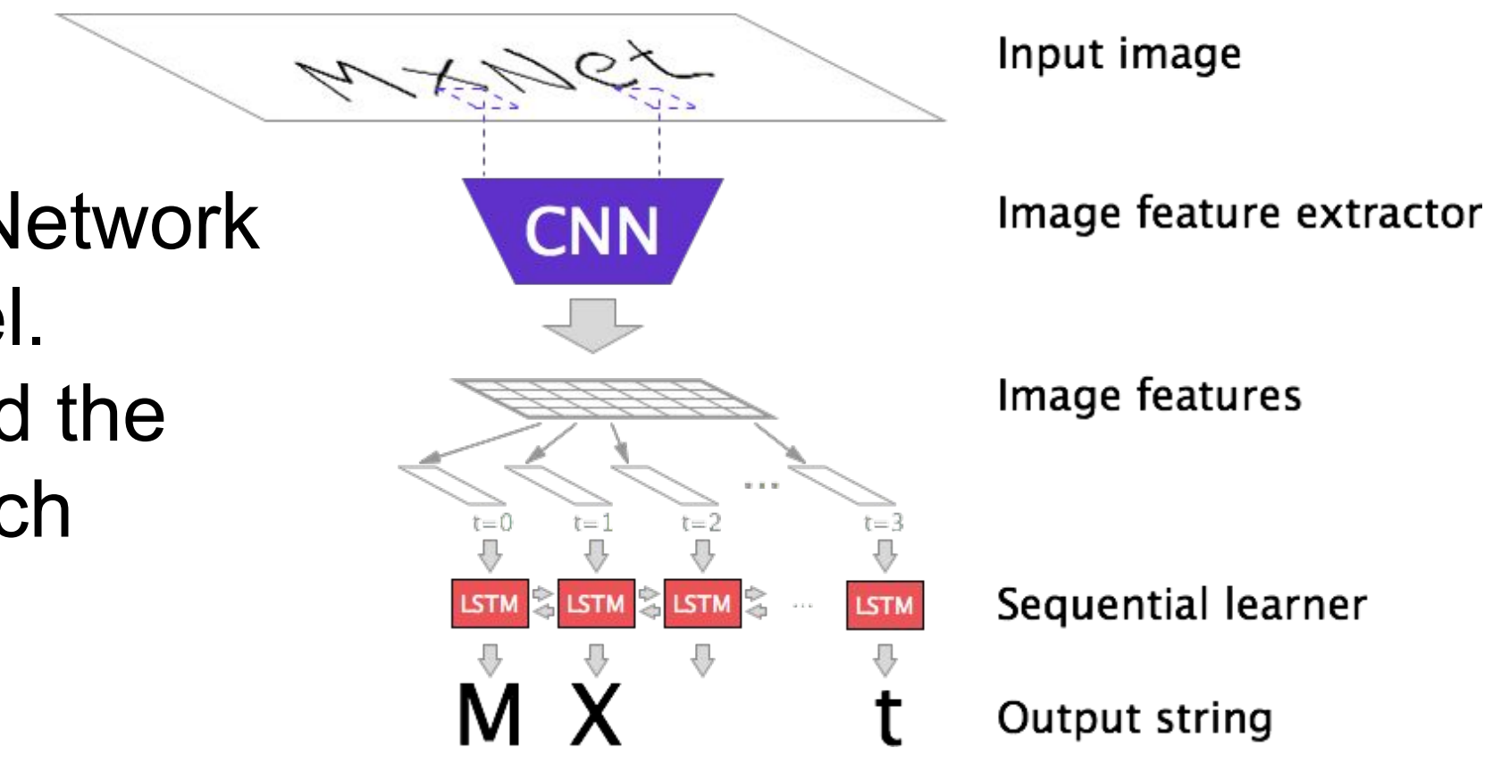
- Finetune a pre-trained Single Shot MultiBox Detector (SSD) to train a new object detection model.
- Fine-tuning pre-trained model can reduce the risk of overfitting.
- SSD has a good speed as YOLO and a good accuracy as Fast R-CNN.


Source data
Target task data
Awesome SSD
Copy
(Pre-trained) Awesome SSD
Predicted Source data Label
Output layer for for target task

Step 2: Text Recognition Model

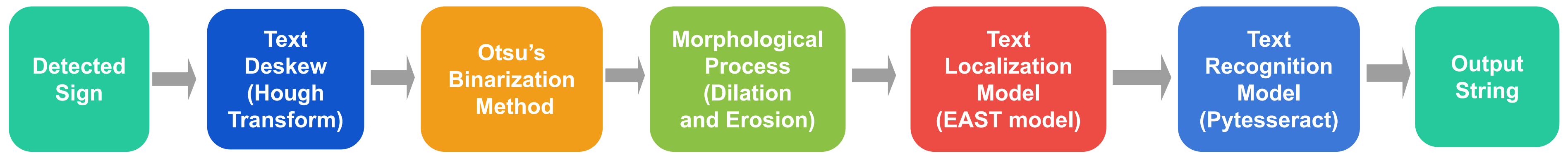
❑ Approach A: CNN + BiLSTM

- The network consists of a Convolutional Neural Network and bidirectional Long Short-Term Memory model.
- The input is an image containing a line of text and the output is a matrix containing the probability of each appearing character.



❑ Approach B: Efficient and Accurate Scene Text Detector (EAST) + pytesseract

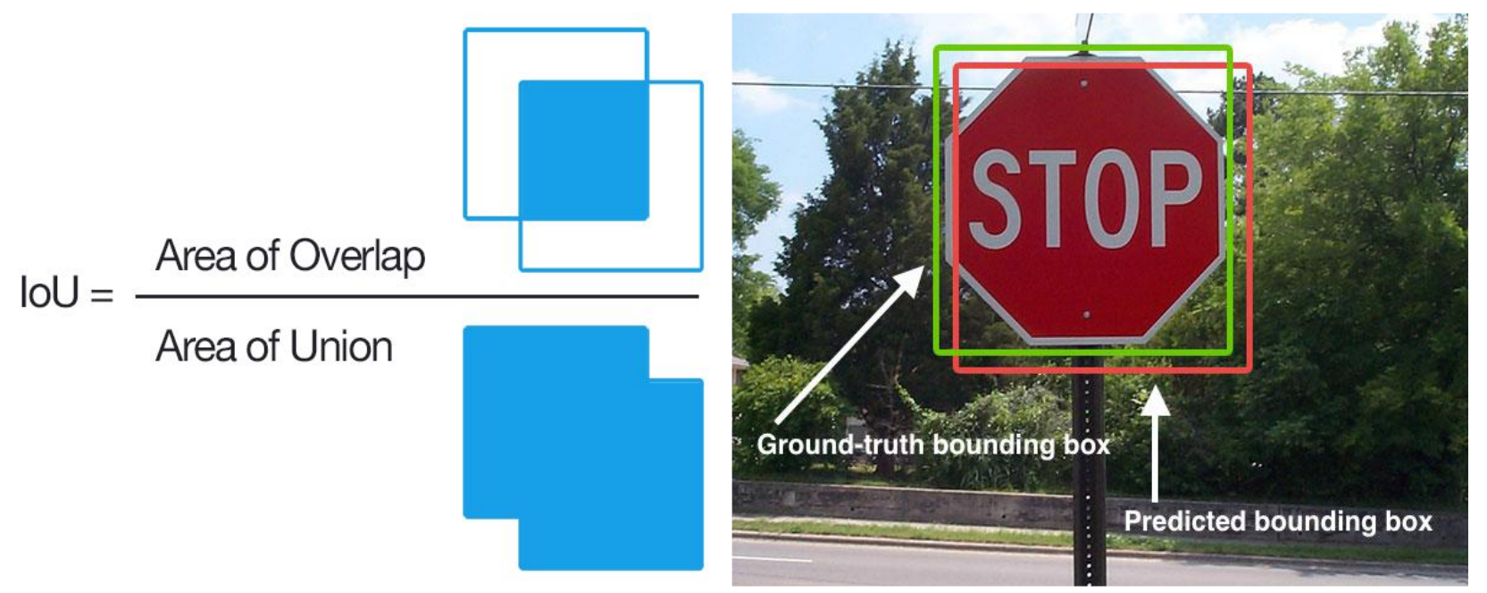
- Pre-process the text image to meet the Pytesseract standard requirements.
  - Text-deskewing, binarization, erosion, and dilation.
- Detect the text area within the sign with the EAST model.
- Input the text region into Pytesseract for text recognition.


Detected Sign
Text Deskew (Hough Transform)
Otsu's Binarization Method
Morphological Process (Dilation and Erosion)
Text Localization Model (EAST model)
Text Recognition Model (Pytesseract)
Output String

Step 3: Evaluation Metrics

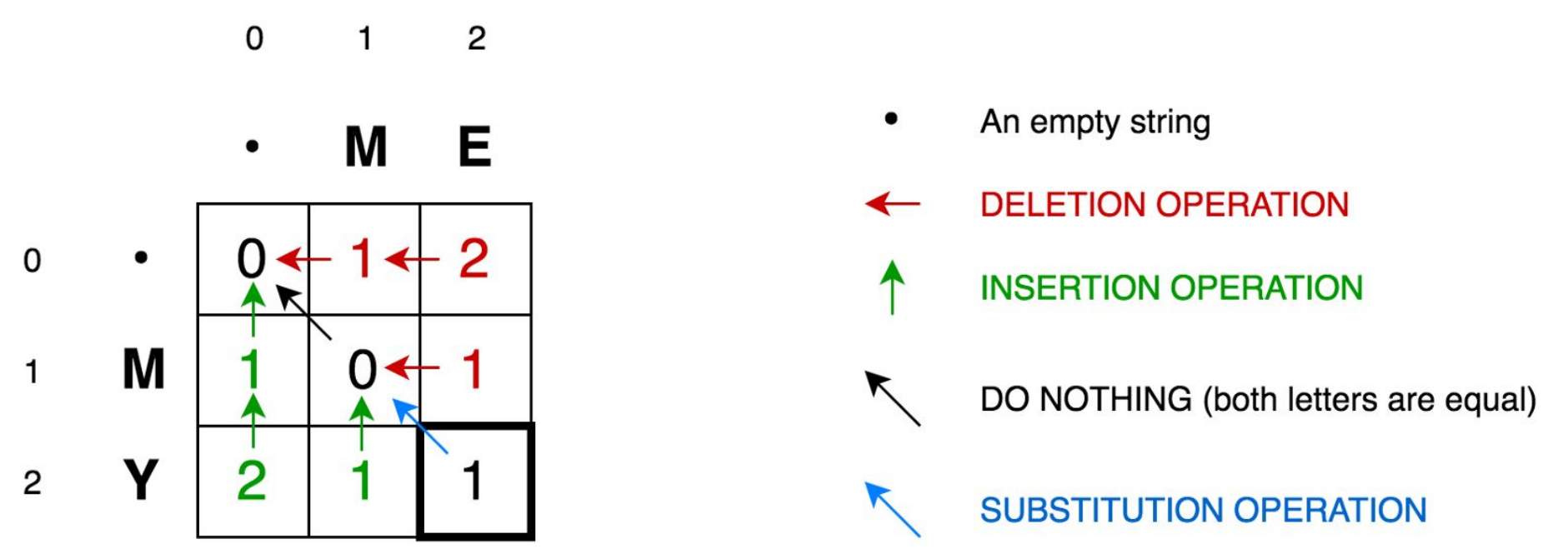
❑ Object detection—Intersection over Union

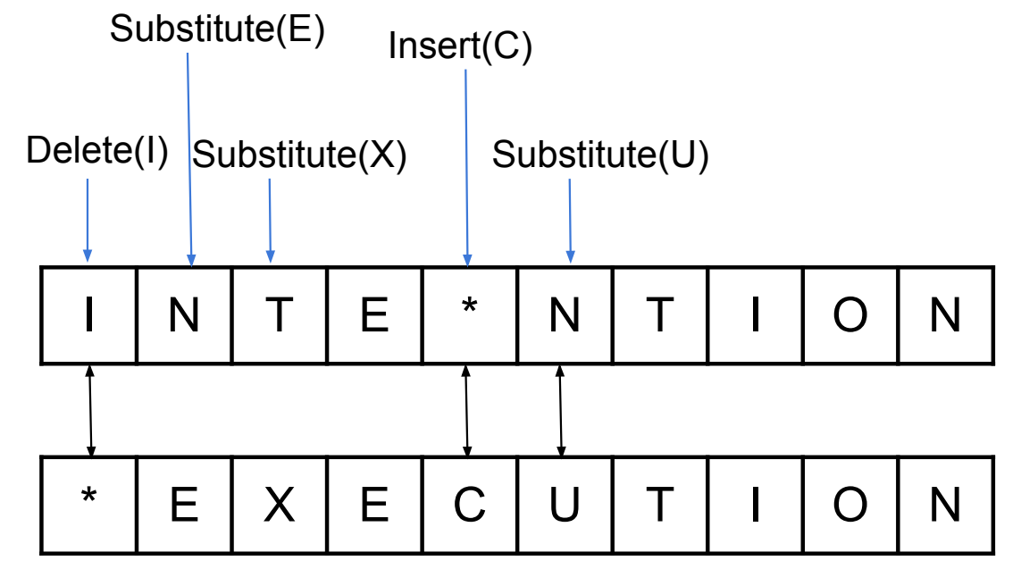
Intersection over Union (IoU) is an evaluation metric used to measure the accuracy of an object detector on a particular dataset.



❑ Text recognition—Levenshtein distance (Edit distance)

Levenshtein distance is a string metric for measuring the difference between two sequences, it computes the minimum number of single-character edits (i.e. insertions, deletions, or substitutions) required to change one word into the other.

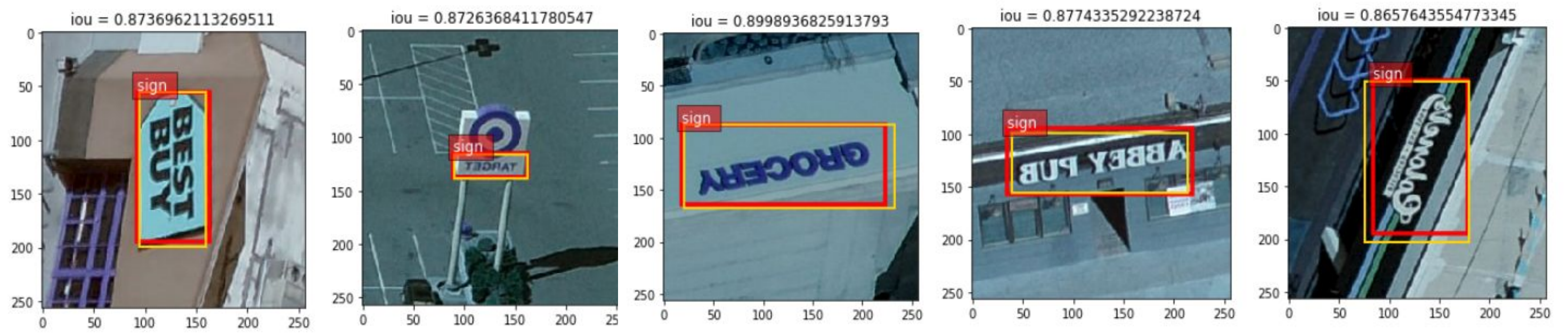




Results

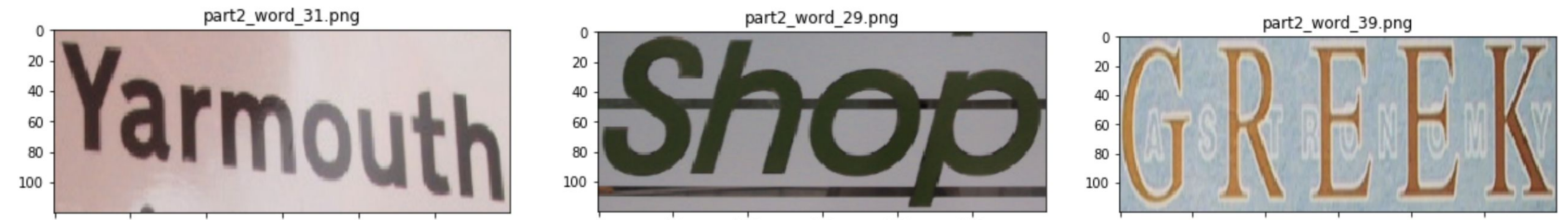
Object Detection:

IoU threshold	Epoch	Percentage Above Threshold
0.7	110	0.95
0.8	110	0.78



Text Recognition:

❑ Approach A: CNN + BiLSTM

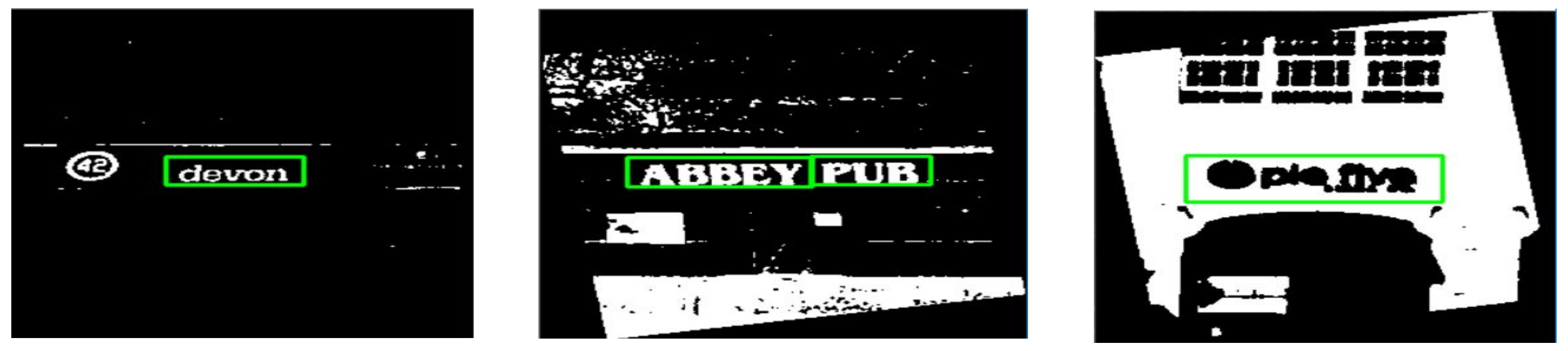


```
Predict text: YarnouS
True label: Yarmouth
Edit distance: 3 , length of word: 8
Normalized edit distance: 0.375
```

```
Predict text: a
True label: Shop
Edit distance: 4 , length of word: 4
Normalized edit distance: 1.0
```

```
Predict text: GREEA
True label: GREEK
Edit distance: 1 , length of word: 5
Normalized edit distance: 0.2
```

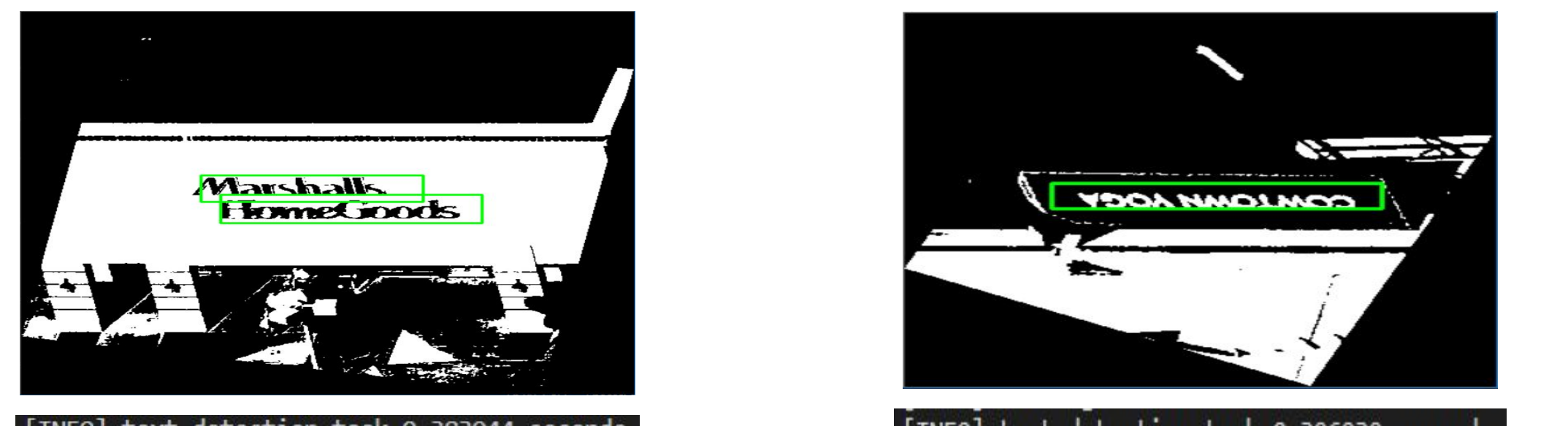
❑ Approach B: Pytesseract



```
[INFO] text detection took 0.396933 seconds
Detected text: devon
```

```
[INFO] text detection took 0.381975 seconds
Detected text: ABBEY
Detected text: PUB
```

```
[INFO] text detection took 0.392946 seconds
Detected text: @ple five
```



```
[INFO] text detection took 0.383944 seconds
Detected text: Warshalts,
Detected text: Hometsoods
```

```
[INFO] text detection took 0.396939 seconds
Detected text: Aa at ST ae el
```

References

1. "Tutorials¶," Tutorials - gluoncv 0.4.0 documentation. [Online]. Available: https://gluon-cv.mxnet.io/tutorials/index.html. [Accessed: 15-Mar-2019].
2. "Intersection over Union (IoU) for object detection," PyImageSearch, 24-Jun-2018. [Online]. Available:https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/. [Accessed: 15-Mar-2019].
3. Aleju, "aleju/imgaug," GitHub, 13-Mar-2019. [Online]. Available: https://github.com/aleju/imgaug. [Accessed: 15-Mar-2019].
4. lanbing510, "lanbing510/ImageLabel," GitHub, 07-Sep-2015. [Online]. Available: https://github.com/lanbing510/ImageLabel. [Accessed: 15-Mar-2019].
5. "Intersection over Union (IoU) for object detection," PyImageSearch, 24-Jun-2018. [Online]. Available:https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/. [Accessed: 15-Mar-2019].
6. Chung, J., & Chung, J. (2018, September 04). Handwriting OCR: Handwriting recognition and language modeling with MXNet Gluon. Retrieved from ttps://medium.com/apache-mxnet/handwriting-ocr-handwriting-recognition-and-language-modeling-with-mxnet-gluon-4c7165788c67
7. Tesseract-Ocr. (2019, May 16). Tesseract-ocr/tesseract. Retrieved from https://github.com/tesseract-ocr/tesseract
8. Scikit-Image. (2019, May 14). Scikit-image/scikit-image. Retrieved from https://github.com/scikit-image/scikit-image
9. Zhou, Xinyu, Yao, Wen, Wang, Yuzhi, . . . Jiajun. (2017, July 10). EAST: An Efficient and Accurate Scene Text Detector. Retrieved from https://arxiv.org/abs/1704.03155
10. Kakul. (2017, February 28). Kakul/Alyn. Retrieved from https://github.com/kakul/Alyn/tree/master/alyn
11. Python Advanced Course Topics. (n.d.). Retrieved from https://www.python-course.eu/levenshtein_distance.php