**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.

**Motivation**

How to extract signage location and text from an aerial imagery.

**Two-model System:**

To perform geospatial assessment to determine the consistency of business names and locations.

**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.

**Motivation**

How to extract signage location and text from an aerial imagery.

**Two-model System:**

To perform geospatial assessment to determine the consistency of business names and locations.

**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.

**Pipeline of Two-Model System**

![Pipeline of Two-Model System](image)

**Step 1: Object Detection**

- **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.

**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.

**Step 3: Evaluation Metrics**

- **Object Detection:**
  - Intersection over Union (IoU) is an evaluation metric for measuring the difference between two sequences and is a threshold.
  - IoU threshold:
    - Superior: 0.8
    - Good: 0.7
    - Threshold: 0.6
  - Epoch:
    - 110
  - Percentage Above Threshold:
    - 0.95
    - 0.78

- **Text Recognition:**
  - Levenshtein distance (Edit 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.
  - Levenshtein distance:
    - Superior: 0
    - Good: 1
    - Threshold: 2

**Results**

- **Object Detection:**
  - Approach A: CNN + BiLSTM
  - Approach B: Efficient and Accurate Scene Text Detector (EAST) + pytesseract

- **Text Recognition:**
  - Approach A: CNN + BiLSTM
  - Approach B: Efficient and Accurate Scene Text Detector (EAST) + pytesseract

**Tools**

- mxnet
- gluon
- jupyter
- 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.

**References**