

|  |  |
| --- | --- |
| Simon Chaisouang  shchaiso@uark.edu  010878218  Late Days: 0 | Venkata Kumari  vg021@uark.edu 011088021  Late Days: 0 |

CSCE 50103: Full Stack Deep Learning, Spring 2025

Homework #2

**Introduction:**  
  
**Objective:** The purpose of this project is to utilize and configure a deep learning network to detect characters in license plates or other text regions.

**Model used:** YOLOX

**Problem statement:** Highlighting the challenges of character detection in real-world scenarios.

**Dataset:** For this project, a custom dataset was used, focusing on detecting and recognizing alphanumeric characters (A-Z, 0-9). The dataset was designed to simulate real-world scenarios involving characters in various forms, such as printed, handwritten, or displayed on objects like license plates, signs, or documents.

**Key Features**

1. **Classes**:
   * The dataset includes 36 classes representing all uppercase letters (A-Z) and digits (0-9).
2. **Annotation**:
   * Each image is annotated with bounding boxes corresponding to the location of each character. These annotations were created using [mention tool, e.g., Labeling, VoTT, or custom scripts].
3. **Dataset Size**:
   * Total images: [provide the total number].
   * Training set: [number of images, e.g., 70% of the total dataset].
   * Validation set: [number of images, e.g., 20% of the total dataset].
   * Test set: [number of images, e.g., 10% of the total dataset].
4. **Variations in Data**:
   * Images include variations in font style, size, orientation, and lighting conditions.
   * Backgrounds are diverse, including plain, textured, and cluttered settings.
5. **Input Size**:
   * Images were resized to a standard resolution of 640x640 pixels for model training and evaluation.

A number on a white surface

Description automatically generated

**Fig:1: Shows the Default image**

**Experiment details:**

**Model 1:**

The model was trained with the configuration below:

* Batch Size:16
* Number of epochs:30
* Depth:0.33
* Width:0.50

With these configurations the below figure was trained:

**A number with numbers and numbers

Description automatically generated with medium confidence**

**Fig 2: Results of Model 1 on Test Image 1**

Based on the number of epochs the model was training. With the above configurations the alphabetical values are not training to detect the character.

**Model 2:**

The model was trained with the configuration below:

* Batch Size:16
* Number of epochs:50
* Depth:0.33
* Width:0.50

**A number with numbers and numbers

Description automatically generated with medium confidence**

**Fig 3.1: Results of Model 2 on Test Image 1**

In this case the model was trained perfectly for numerical values, and it does not detect the alphabet. It is based on character detection.

For the same model we utilized a different test image. From the results shown below, the model is able to detect smaller characters that are not so evident compared to the big five.

A digital screen with numbers

Description automatically generated

**Fig 3.1: Results of Model 2 on Test Image 2**

**Model 3:**

**Augmentation and Model Input:**

self.input\_size = (640, 640)

self.test\_size = (640, 640)

self.mosaic\_prob = 0.5

self.mixup\_prob = 0.2

self.hsv\_prob = 1.0

self.flip\_prob = 0.5

self.max\_epoch = 50

self.warmup\_epochs = 5

self.eval\_interval = 1

self.data\_num\_workers = 4

A screenshot of a video game

Description automatically generated

**Fig 3: Results of Model 3 on Test Image 1**

**Model 4:**

We use the following configurations listed below to define model 4:

* Batch Size: 8
* Number of epochs: 30
* Depth: 0.7
* Width: 1.0

For this model, we investigate expanding the depth and width of our network in contrast to our previous models. The depth controls the number of layers in the network while width controls the number of channels in each convolutional layer. The goal for this model is to see how we can achieve a more accurate model at the cost of more training time and risks of overfitting.

The model was able to achieve a average precision of 75.54 which is sufficient for a character detection model. We refer to the images below from the results of testing on the test data.

A number on a screen

AI-generated content may be incorrect.

**Figure 4.1: Results of Model 4 on Test Image 1**

A digital screen shot of a number

AI-generated content may be incorrect.

**Figure 4.2: Results of Model 4 on Test Image 2**

From the results, we can see the model performs comparably, but slightly worse than our previous models. We can infer from here that, number of epochs play a significant role in how well a model may perform. The number of epochs controls the number of times the model is exposed to the entire training set during training. Providing our model with a greater number of epochs could influence its ability to learn underlying patterns within the dataset. Due to limitation of resources, fine tuning greater hyperparameters could not be experimented but it is alluded that it may achieve more efficient and robust models in real-world applications.