Mina Mehdinia

Final Project

# A Coin Detection System using Deep Learning with a Data Centric Approach

## Introduction

This report presents the development of two coin detection systems for detecting and classifying four types of US coins: penny, nickel, dime, and quarter. First, I collected all the images using my personal phone's camera, and labeled them using an open source tool called LandingLens. First, I used the Detectron2 [1] framework which is developed by Facebook AI research on Pytorch. I used it to train an object detection model to classify and predict the bounding boxes for the coins in an image. For the second system, I used the LandingLens [2] platform. Since this platform is a data centric platform [3] where we do not have access to the model I focused on playing with the dataset and its effect on the performance of the model. I spent time training the same model but on various data with various backgrounds. I used three different backgrounds to collect my data, namely, blue, black and white. The performance of the systems is evaluated using accuracy, precision, and recall metrics.

## Dataset

I took 65 images for training, and 13 for validation sets, with an extra 3 images solely for testing purposes. In each image, there are two to eight coins. There are a total of 288 instances. Both sides of the coins are included in all the datasets.

The dataset comprises labeled images of coins with associated bounding box annotations. The annotations are attached as a CSV file containing the filenames, classes, width, height, xmin, xmax, ymin, and ymax for each coin. I used the LandingLens platform to draw the bounding boxes and had to extract the labels manually and convert  it to a csv file. The dataset is split into training, validation, and test sets for model training and evaluation. The data is captured with different backgrounds and lighting conditions to investigate the data quality and diversity of the model's performance.

Below are some images from the training dataset.

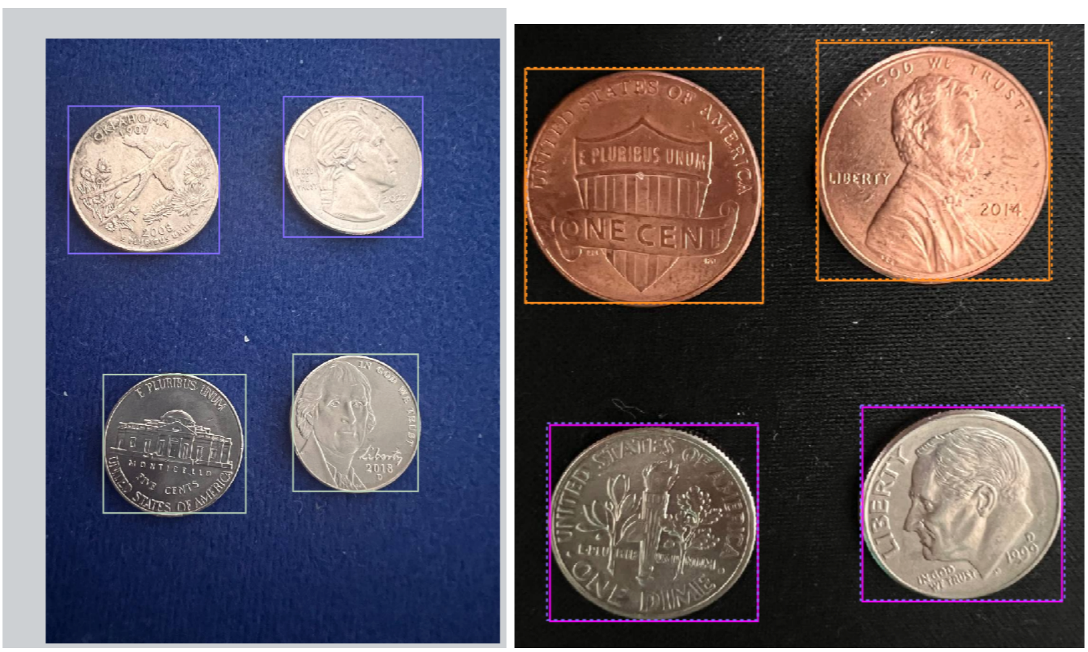


Figure1. Two sample images with different backgrounds from the training dataset.

**Methodology**

For the Detectron2 system, the dataset is registered with Detectron2, and the bounding box annotations are converted into COCO format using custom Python functions. The model is trained using the Faster R-CNN architecture with a ResNet-50 backbone. The training process includes adjusting anchor box sizes and aspect ratios, using a batch size of 4, and applying data augmentation techniques to improve model generalization. I trained the model for 5000 iterations, which is almost equal to 100 epochs according to this formula.

iterations = (number of epochs) * (number of training images) / (batch size)

For the LandingLens system, the platform is used to train the model using a dataset of coin images with their corresponding labels. The models are trained for 50 epochs. LandingLens provides a user-friendly interface for training custom models using their platform. The platform allows users to track the accuracy and performance metrics of the model over time and make necessary adjustments to improve its performance.

For both models, data augmentation is used including random brightness, random contrast, and random horizontal and vertical flips.

**Evaluation**

The model performance is evaluated using accuracy, precision, and recall metrics. For the Detectron2 system, the COCOEvaluator from Detectron2 is employed, providing Average Precision (AP) and Average Recall (AR) metrics. For the LandingLens system, four different models are trained on different datasets, and their performance is compared.

**Results**

For the first system, using Detectron2, the performance was not very desirable, the model overfitted to the training data. Figure 2 shows the training loss curve. There should have been an early stopping in place but due to shortage of GPU on google colab, I was not able to retrain the model and switched to the LandingLens platform.

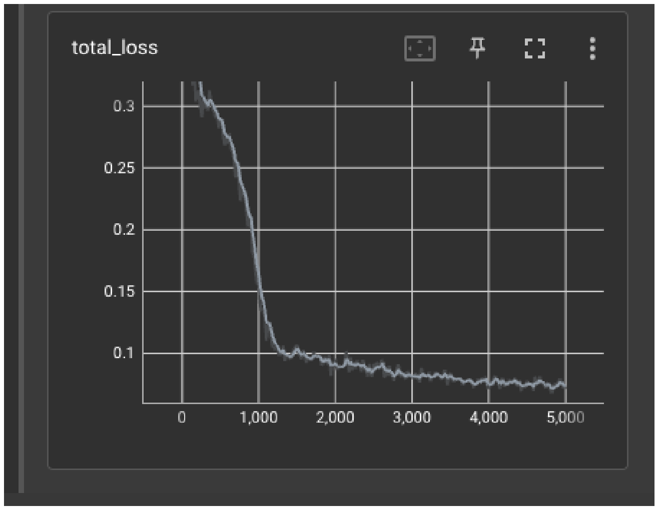


Figure 2. Training loss for the  Detectron2 model

For the LandingLens system. I trained three different models, each trained on different datasets. The idea was to keep the model the same but train on various datasets. 1) only including the blue background images 2) blue background images + black ones 3) all backgrounds including white. Each background had 21 images in the training set. The validation was constant for all the three models consisting of 4 images from each background.  Below is a summary of the performance metrics using the LandingLens platform

| Model | Model #1 | Model #2 | Model #3 |
|---|---|---|---|
| Precision | 51 % | 76 % | 93 % |
| Recall | 45 % | 69 % | 89 % |

We can see that increasing the number of data points in the training set and also using more diverse data resulted in better precision and recall in object detection. It should be noted that the confidence threshold was set to 0.5 for all the models.

In figure 3, we can see a screenshot from the platform and its performance on the validation set. You can see that most of the coins are correctly classified.

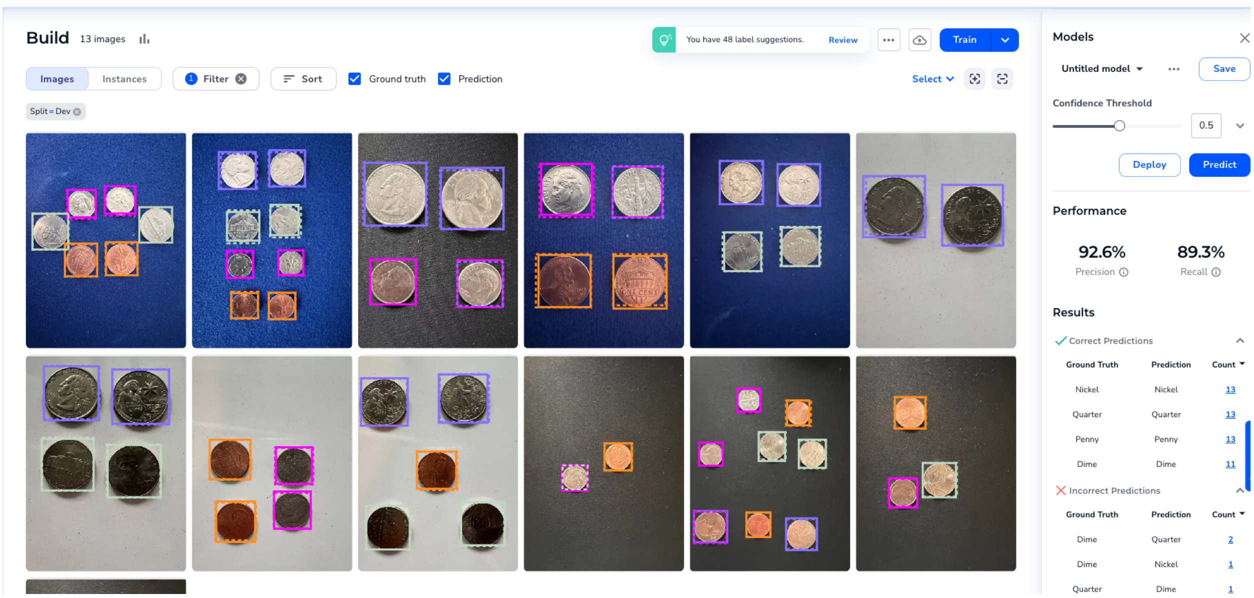


Figure 3. A screenshot of the LandingLens platform showing data and some metrics.

In figure 4, prediction for a new data point from the test set is shown. We can see that the model was able to predict each type of coin correctly (color of bounding boxes).

Figure 4. Prediction on an unseen data point from the test set

**Conclusion**

In conclusion, the end-to-end coin detection system developed using a data-centric approach and deep learning object detection models can accurately detect and classify four different types of US coins under different lighting and background conditions. The system has the potential to automate coin detection and calculation and can be used in various applications. In my future efforts, I will keep adding data to the training set and once it is better, I will deploy it and make a web app that users can upload an image and receive a prediction.

**References**


1. Facebook AI. (n.d.). Detectron2. GitHub. https://github.com/facebookresearch/detectron2
2. Landing AI. (2023). Landing AI Platform . https://app.landing.ai/app
3. Jarrahi, M. H., Memariani, A., & Guha, S. (2022). The Principles of Data-Centric AI (DCAI). *arXiv preprint arXiv:2211.14611*.