Methodology

Role of Student vs. Mentor

The creation and annotation of the syringe dataset was done by the student. The training, testing, and analysis of the model was also done by the student. The mentor provided helpful tips for dataset creation, and also helped guide the student along the STEM engineering process throughout the project. The development of the project occurred from October 2024 to February 2025.

Equipment & Materials

LabelImg was used to annotate all captured syringe images. The COCO dataset was used to pre-train a YOLO11 model (Lin et al., 2015). For training and testing the neural network, a Tesla T4 GPU was used over the cloud through Google Colab.

When training the neural network, it was essential to capture enough images of damaged and undamaged syringes so that the neural network can accurately find defects in the syringes presented to it. With too few photos to train off, the model would not be able to accurately make predictions of syringe defects. Additionally, the model could overfit and be unable to make accurate predictions for images of syringes that slightly vary from training data. Since the YOLO11 model is pre-trained on the COCO dataset, which contains around 200,000 images of common objects such as humans, bikes, and helmets (Lin et al., 2015), training dataset sizes do not need to be as large as a model training from scratch. Therefore, 278 images were taken for training, and 72 images were taken for validation. Four classes were specified: “good needle,” “good plunger,” “bent needle,” and “torn plunger” and each image in the

dataset was annotated using bounding boxes to indicate if any of those four classes were present. The class distribution used for training is shown in Figure 1.

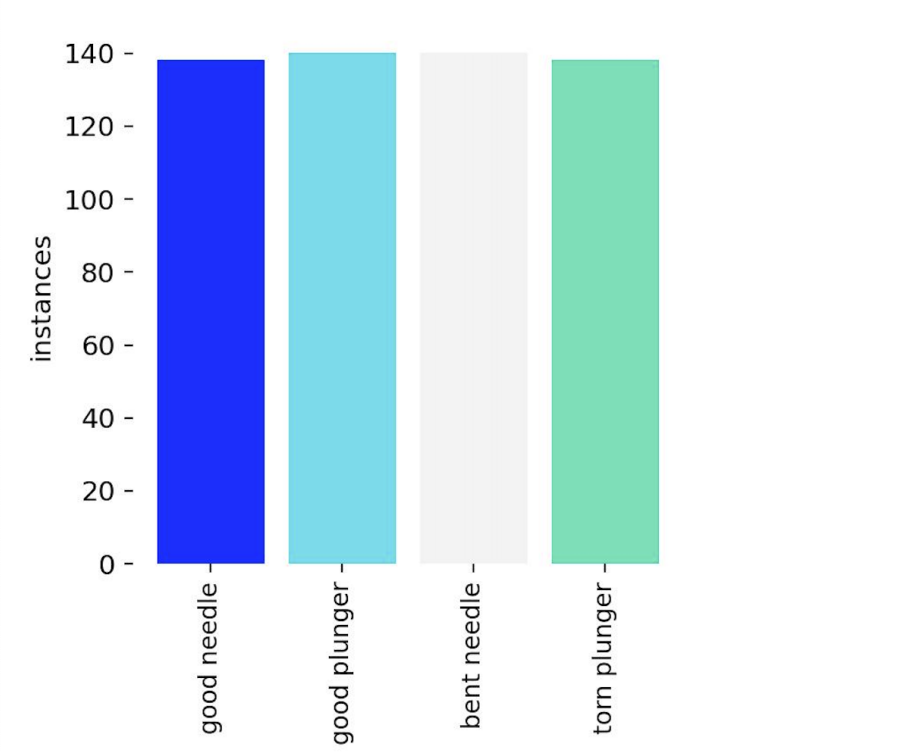

Figure 1: Instance spread of defined classes in training dataset

Statistical Test: To determine if a consistent distribution of all classes was used, a Chi-Square Goodness of Fit Test was performed. Considering that there were 278 training images with 2 detections in each image, the expected value of each class is 139. However, 138 good needles, 140 good plungers, 140 bent needles, and 138 torn plungers were observed. The test was conducted with three degrees of freedom. A p-value of 0.999 was obtained, meaning there was no significant skew in the dataset's distribution (***p<0.5**). Therefore, the dataset was unbiased and would be proper to proceed with.

Preparation of Environment: Afterwards, some prerequisites for training were set up in Colab. Data augmentation was enabled for training, where the model would artificially create

training images by combining and modifying existing images. Validation was also enabled at the end of each epoch so that the performance of the model could be measured as the model's learning progressed. However, validation would need to be modified because only two classes out of the four total would be outputted for the final syringe defect detection: one out of the two potential detections for the needle, and one out of the two potential detections for the plunger. Two different approaches were used to solve this problem, with one of them generating better results than the other.

Technique 1

In this approach, a function was created inside of the validation processing file, named "filter_by_class". Without the function, multiple predictions can exist of the same object. For example, the model can identify the needle of a syringe as both a "good needle" and a "bad needle". Therefore, the function takes the raw model detections and filters them to take the most confident detection for each separate component (needle and plunger). Therefore, there will only be two detections present on a syringe: one for the needle, and one for the plunger.

However, during testing, the function seemed to eliminate possible correct detections as well, which drastically reduced model accuracy. Therefore, technique 1 was found to be problematic, and a different approach was then tried.

Technique 2

In this approach, the maximum number of detections was specified in the code of the validation process. Instead of using a function that would eliminate detections, a hard cap was placed on

the number of detections, which would avoid the potential risk of removing good detections. This method drastically improved accuracy results during testing compared to Technique 1.