**Introduction**

This document provides a concise overview of the development process for an object detection model aimed at identifying staff name tags. The primary components discussed include data preparation, augmentation, model training, evaluation, and detection results. The model being used for the object detection task is YOLOv8 and the tools for data preparation is Roboflow.

**Data Preprocessing**

A 53-second video, 'sample.mp4,' served as the training data. The preprocessing steps are:

1. **Split the frame image from video**

The training images need to be extracted from the sample video. The frame rate of the video is 25 frames/second, so the estimated frame images will be 25\*53 = 1325 frames.

1. **Perform data annotation / Draw boxes of name tag**

The extracted frame images is then being annotated with the bounding box that detecting the name tag area. The drawn box is recorded in a separate text file which includes the information about the class number and xywh values (xy coordinates and width height value).

1. **Null data annotations**

For those frame images that do not consist of name tag will be labelled as null data, which is used for ensuring that the dataset maintains a balance between images with annotations and images without annotations. This annotation enables the model to differentiate between images with and without name tag accurately. The requirement is set to have at least 80% of the images contain annotations. This means that most of the images in the dataset must have staff members with name tags, and only a smaller proportion can be null data.

1. **Split data**

The data is then being split to train (724), valid (68) and test (36) set, in total of 828 images as the 80% of the dataset is data with annotations and 20% is data without annotations or data labelled as null. Therefore, it is lesser than the total frame images being extracted and not all of the null data images will be added.

A close up of a card

Description automatically generated

1. **Auto-Oriented and Resize**

Auto-Oriented is being performed to ensure that the images are correctly displayed, respecting their EXIF orientation. This prevents issues caused by displaying images without considering their orientation, which can be a common bug in computer vision projects. The images is also being resized to 960x720 to ensure the consistency of input size same with the video width and height.

A white background with black text

Description automatically generated

**Data Augmentation**

There are 2 types of augmentation being performed to the dataset which is image flipping and rotation.

1. **Flipping** (Horizontal and Vertical)

As part of the data augmentation process, one key step is the "Mirror Effect." This technique involves flipping an image both horizontally and vertically, creating mirrored versions of the original image. By doing so, it addresses several important aspects of improving machine learning models. It can aid in the recognition of mirrored or rotated objects, which can be challenging for models and helps the model become invariant to such transformations and also enhancing its robustness.

1. **90° Rotation** (Clockwise and Counter-Clockwise)

This rotation brings about orientation variation by changing the orientation of objects within the image. This helps training models to recognize objects in different positions, adding robustness to their capabilities and handling objects presented in various orientations effectively. In summary, it enhances a model's capability to handle objects from different perspectives, ultimately contributing to improved performance and accuracy.

**Model Training**

The YOLOv8 model was trained on a Kaggle notebook with a GPU P100. Two training runs were performed: the first with 100 epochs, and the second with 50 epochs to compare performance and assess overfitting. This approach aimed to maintain or improve model performance with fewer training epochs.

**Evaluation**

Roboflow provided evaluation metrics, including box\_loss, cls\_loss, dfl\_loss, precision, recall, mAP50, and mAP50-95. Graphs, such as the confusion matrix, F1 confidence curve, precision-recall curve, precision-confidence curve, and recall-confidence curve, were used to visualize model performance.

**Detection Outcome and Results**

The model's output included a detection video, detection images, and a DataFrame storing labels (xywh values) in a CSV file.