Data

The given data for staff attire detection is a video that used for performing the detection task of name tag on staff attire. The video is about 53 seconds and is used as the training data. The detail properties is as figure below:

A screenshot of a computer

Description automatically generated

*(Figure of detail properties of “sample.mp4”)*

Model

There are a variety of models that can be used for object detection task and the model being selected is YOLOv8 which is the new model built by Ultralytics. It incorporates a new transformer-based architecture, offering enhanced accuracy and performance compared to the previous version.

Tools

The tools that used for dataset generation including data preprocessing, data augmentation and splitting is Roboflow.

Data Preprocessing

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 training model is YOLOv8 model and is being trained on Kaggle notebook with GPU P100.

There are two separate training runs with different numbers of training epochs:

1. 100 Epochs: The first training is set to 100 epochs and evaluate the model performance with test dataset.

1. 50 Epochs: The second training is set to 50 epochs to make the comparison with the model running with 100 epochs. This shorter training period is designed to see if a reduced number of epochs still yields satisfactory results. It's an exploratory step to check if model performance can be maintained or even improved with fewer epochs.

The decision to perform two separate training runs with different epoch settings is to compare the model's performance between the two runs and assess whether overfitting occurs with the more extended training duration. A shorter training period is used in the second run to gauge whether model performance remains consistent or if further training provides diminishing returns.

Evaluation

Roboflow has provide some output of evaluation metrics and graphs after performing model training to ease on the process of model evaluation.

1. Metric
2. Box\_loss, Cls\_loss, dfl\_loss

The loss metric used during training and validating. "Box\_loss" measures the error in predicting object bounding boxes, "Cls\_loss" evaluates the accuracy of object classification, and "dfl\_loss" assesses the loss from dynamic feature learning.

1. Precision, Recall

Precision is a metric that quantifies the accuracy of positive predictions, while recall measures the ability of a model to identify all relevant instances. They are commonly used in classification tasks to evaluate a model's performance in terms of false positives and false negatives.

1. mAP50 (Mean Average Precision at IoU 0.5)

mAP50 is a metric used to assess the accuracy of object detection models. It calculates the average precision of object detection across different classes and is particularly useful for evaluating how well a model detects objects at a specific Intersection over Union (IoU) threshold of 0.5.

1. mAP50-95 (Mean Average Precision from IoU 0.5 to 0.95)

mAP50-95 is similar to mAP50, but it considers a broader range of IoU thresholds from 0.5 to 0.95, providing a more comprehensive evaluation of a model's detection performance.

1. Graphs
2. Confusion Matrix

A confusion matrix is a graphical representation of the performance of a classification model. It displays the true positive, true negative, false positive, and false negative predictions, offering insights into a model's accuracy and errors.

1. F1 Confidence Curve

The F1 Confidence Curve is a graph that shows how the F1 score of a model changes with different confidence thresholds. It helps determine the optimal threshold for making predictions, balancing precision and recall.

1. Precision-Recall Curve

This curve illustrates the trade-off between precision and recall at various thresholds. It is useful for assessing the model's ability to balance accurate positive predictions (precision) with the ability to capture all positive instances (recall).

1. Precision-Confidence Curve

The Precision-Confidence Curve visualizes how the precision of a model varies with different confidence thresholds. It helps in understanding how confident the model's positive predictions are.

1. Recall-Confidence Curve

Similar to the Precision-Confidence Curve, the Recall-Confidence Curve depicts how the recall of a model changes with different confidence thresholds. It provides insights into the model's ability to capture positive instances across different confidence levels.

Detection Outcome and Results

1. Video output – detection output video
2. Image output – detection output image
3. Dataframe output – detection output that store the labels xywh values in csv file

Further Enhancement

1. Bounding Box Augmentation:

Bounding Box Augmentation takes traditional image augmentation a step further by modifying the content within specific bounding boxes. This technique offers greater control over training data, making it more suitable for diverse problem conditions. It allows developers to alter the brightness, darkness, or blur within bounding boxes relative to their backgrounds. Bounding Box Level Augmentation leverages the idea of creating specialized augmentation, which is particularly beneficial for tasks capturing rapidly moving objects or where content within bounding boxes needs specific adjustments.

Use Cases:

* Modifying colors of objects in OCR images
* Introducing blur for rapidly moving objects like pets or cars
* Rotating objects such as board game pieces
* Flipping the orientation of objects to simulate mirrored effects

1. Random Rotate Augmentation:

Random Rotate Augmentation involves rotating source images clockwise or counterclockwise by a random number of degrees, thus changing the object's position in the frame. Importantly, the bounding boxes must be adjusted to encompass the rotated objects. This technique is particularly valuable in scenarios where objects can appear at various angles during production, even if initially captured from only one perspective. It helps prevent overfitting and increases variation in the dataset, enhancing the model's ability to generalize.

When to Use Random Rotate Augmentation:

* In cases where the camera position is not fixed relative to the subjects (e.g., mobile apps)
* When capturing objects from various angles is impractical
* To prevent overfitting and promote variation in the dataset
* When to Not Use Random Rotate Augmentation:

When valuable content exists in the original corners of the images

* In cases where objects in the images do not naturally rotate (e.g., street signs)
* When the bounding boxes are mostly skinny rectangles, as rotation can cause bounding boxes to expand, potentially affecting model predictions