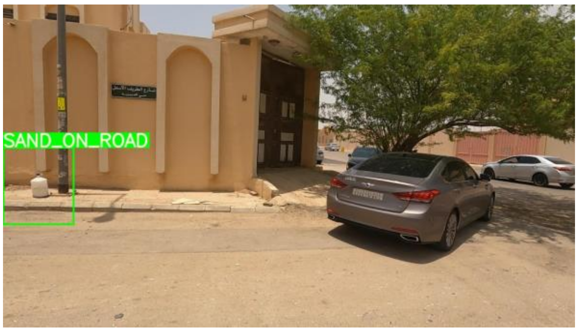
Project Overview

- We have used YOLO V5 pretrained model trained on COCO dataset as starting point. We have retrained model on our data set to reduce training time & more accurate model
- We have tried data augmentation to reduce class imbalance
- Used customized stratified train test split to avoid data bias
    - Split 80-20% of class with minimum number of samples , if number of samples is 1 than move it into training
    - For next class identify how many classes is already available in training (since its multilabel) & how many ideally should be based on 80-20 split
    - We randomly sample those many numbers of rows from the training if number of classes less than expected else don't sample anything move everything to test set
- Experiment tracking using weights and biases
- Different technique tried to improve data Quality.

Challenges faced in data preparation

1. **Wrongly labelled data** - we have **manually removed wrongly annotated class data**.



2. **Multiple labels** – To further clean the data and **avoid model confusion** to predict multiple bounding boxes, we removed multiple bounding boxes data in training set as show in slide

3. **Class imbalance:** We did **custom stratified split for training and validation** to make sure we have proper distribution while training. We did try to apply class weights but there was no significant gain.

4. **Bounding Box improvement:** We have created a custom bounding box functions to solve bounding box going outside the image.

5. **Final image size adjustment:** on larger image, if we multiply bounding box by 2 then we are increasing bounding box size by **factor of 2** which may create **bad training data**. To solve this challenge, we reduced the size of the **image by 50%**

How to make it more scalable

Data annotation scalability

Active learning-based data selection

Active learning is a machine learning strategy in which a model can interactively query the user (or some other information source) to obtain the most informative data for improving its performance. It is particularly useful in cases where labelled data is scarce or expensive to obtain.

Active learning techniques:

- Uncertainty sampling: This technique selects images that the model is least confident about, to improve its performance on those images. There are several ways to measure uncertainty, such as entropy, variance, or the least confidence method.
- Query-by-committee (QBC): This technique involves training multiple models, and then selecting the images that are most disagreed upon by the models. This can help to identify images that are difficult to classify and would benefit from additional annotation.
- Expected error reduction (EER): This technique selects images that are likely to have the greatest impact on reducing overall classification error. This can be done by using techniques such as active learning by learning, which uses the model's current performance to estimate the expected error reduction of different images.
- Active Learning by Simulation: This technique uses a simulation model to generate images that are similar to the images in the real dataset. These simulated images can be used to train a model without the need for manual annotation.
- Active Learning by Human Feedback: This technique allows human annotators to provide feedback on the model's predictions. The model can then use this feedback to select the images that would be most beneficial to annotate.

Reducing inter annotation variance

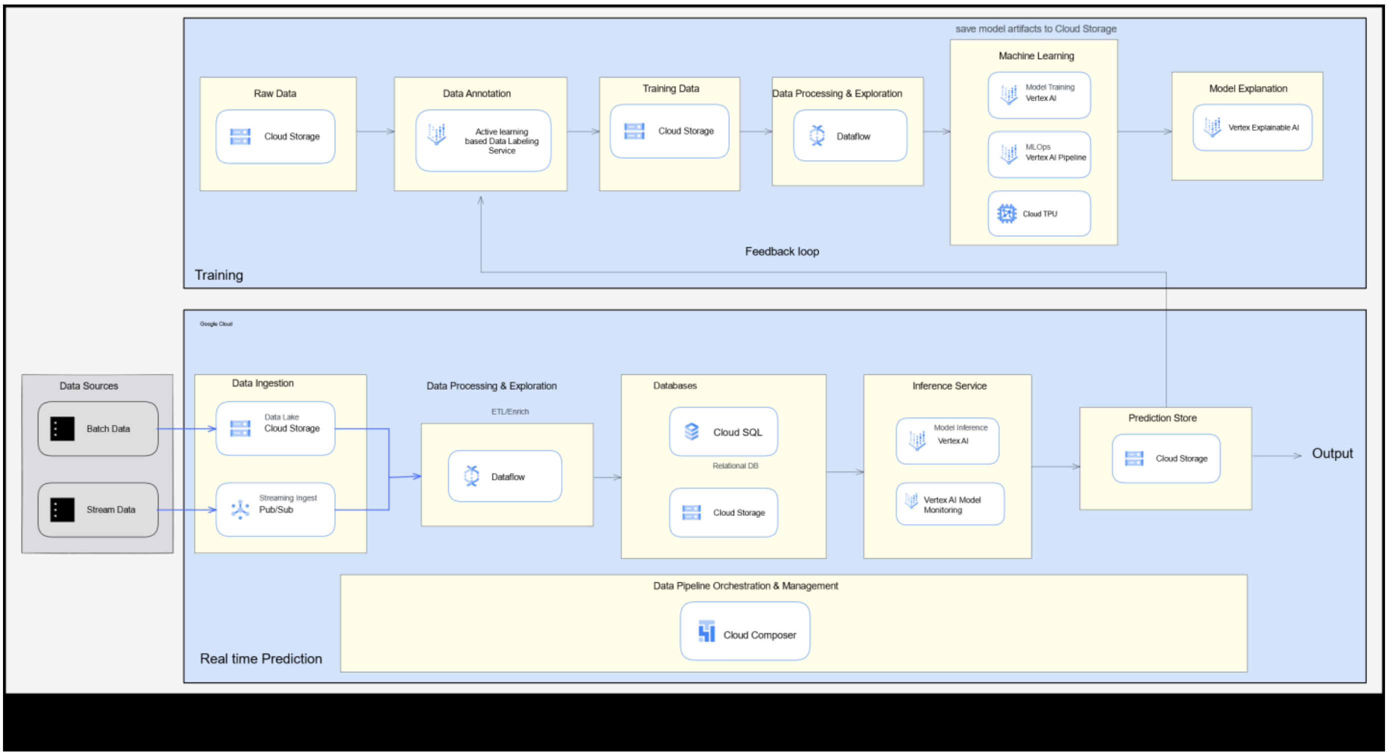
Inter-annotator variance refers to the inconsistency in the labels assigned to the same image by different annotators. It is a common issue in image labelling and can negatively impact the performance of a machine learning model. Here are a few ways to reduce inter-annotator variance in image labelling:

- Clearly defined guidelines: Providing clear and detailed guidelines for annotators can help to reduce inter-annotator variance. The guidelines should include definitions of the labels, examples of the types of images that should be labelled with each label, and instructions on how to handle ambiguous cases.
- Training: Providing training to annotators on the guidelines, labelling conventions and examples can improve the consistency of annotations.
- Quality control: Regularly checking the annotations for quality can help to identify and correct errors and inconsistencies. This can be done by having a set of images labelled by multiple annotators and comparing their labels.
- Active Learning: Using Active Learning techniques can also help to reduce inter-annotator variance by allowing the model to select the images that are most beneficial to annotate, instead of relying on annotators to label all images indiscriminately.

- Use of pre-trained models: Using pre-trained models can help to reduce inter-annotator variance by providing a baseline for the annotations, which can help to ensure that the labels are consistent across different annotators.
- Use of Crowdsourcing: Crowdsourcing image labelling can also help to reduce inter-annotator variance by obtaining labels from multiple annotators for each image, and then taking a consensus of these labels

Scalable Architecture for training & inference

a1.svg



Open-Source Software used

- Pytorch
- Pandas
- Numpy
- Opencv

Things I have tried

1. **Prediction/Error Analysis:** To improve precision, the model can be producing **wrong classification, localization or both**. We wanted to create custom script to understand issues with prediction to improve precision value (Similar to below analysis)
2. **Augmentation**: More augmentation to reduce class imbalance.
3. **Better model**: would love to try different modelling techniques.
4. **Data Cleaning**: Clean training data by reducing confusing labels & improved bounding boxes.
5. **AutoML**: Google has very strong AUTO ML Service would love to try and benchmark our solution with them.

Pseudo Label-Guided Sampling - we train a network twice and use the pseudo labels generated from the first model to guide the training of the second model. We filter prediction results of a trained model to generate pseudo labels to complement sparse annotation for unannotated regions