Proposal Report

Sepehr Heydarian, Archer Liu, Elshaday Yoseph, Tien Nguyen

Table of contents

1 Abstract

Hello

2 Introduction

Wildfires are tragic environmental disasters that pose a threat to ecosystems and communities. Our Capstone partner, Bayes Studio, a Vancouver-based startup utilize advanced AI tools for environmental monitoring and early detection of wildfires using computer vision. The problem our partner is faced with is keeping detection models up to date as new data becomes available. The current workflow involves manual labeling of incoming images using existing `YOLO` object labeling model, followed by retraining and redeployment. As new data becomes available frequently, this manual and repetitive approach results in delays, bottlenecks in model improvement, and ultimately slower response times in wildfire detection. This has real-world consequences as early detection of fire ignitions is crucial in mitigation efforts to limit damages of wildfires to nearby communities (Defence Research and Development Canada 2022).

Our objective is to automate and streamline this update process through a reproducible data science pipeline. When new images arrive, the pipeline will initiate automated pre-labelling using the partner's `YOLO` model. To ensure reliability, a secondary model - Meta's Segment Anything Model (`SAM`) - will be utilized to verify YOLO's predictions. If the labels from both models agree, the image is accepted. Otherwise, it is routed to a human-in-the-loop labelling interface built with open-source tools such as `Label Studio`. We will also ensure that the model is edge-deployable by introducing distillation and quantization steps when model has undergone retraining.

The final product delivered to our partner will be an end-to-end automated pipeline, deployed via Github Actions, which takes new images and outputs and updated YOLO model file (`.pt`). This approach reduces human efforts, shortens turnaround time for model updates, and increases labelling accuracy.

3 Proposed Pipeline

3.1 High-Level Overview of the Proposed Data Pipeline

With a large amount of anticipated unlabelled imags coming in, a major challenge is how to keep the model current and effective as new data arrives. Manual labeling and retraining will soon no longer be sustainable, especially under the time pressures involved in wildfire detection.

To solve this, we propose an intelligent and iterative data pipeline. This solution combines automation, human oversight, and model optimization to support continuous improvements. Below is a high-level summary of the process:

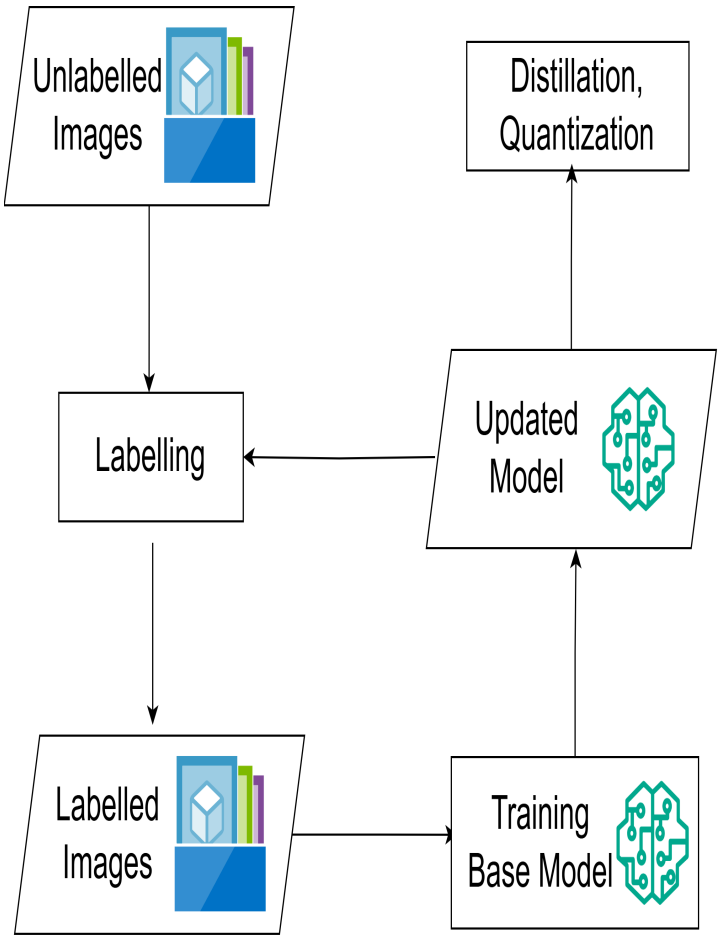

Figure 1: High-level overview of the data pipeline

1. **Unlabelled Image Ingestion**
   The pipeline begins by collecting raw, unlabelled images from the client's data sources.

2. **Automated Pre-Labelling**
   One or more AI models are used to perform initial labeling. They identifies objects such as fire, smoke, or vehicles. This step helps reduce the time and effort required for manual annotation.

3. **Human-in-the-Loop Review**
   Human experts then review and correct the AI-generated labels. This step ensures high accuracy while avoiding the burden of fully manual labeling.

4. **Model Retraining with Verified Labels**
   The corrected labels are used to retrain or fine-tune the main wildfire detection model. This helps the model learn from new data and maintain strong performance over time.

5. **Model Optimization for Deployment**
   Optimization techniques such as **distillation** and **quantization** are applied. These help reduce the size and complexity of the model, making it more efficient for deployment on edge devices.

Keypoint: This pipeline is designed for continuous learning. Each iteration allows the system to improve, leading to better detection accuracy and faster response in real-world situations.

Further details on the technical components and implementation will be provided in the following sections.

3.2 The Data

The proposed pipeline is designed to process image data annotated with bounding boxes in text format. The client currently maintains a repository of over 2 million unlabelled images, with approximately 500 new unlabelled images expected to be added each month from various sources. While the full dataset is stored on Google Cloud Storage (GCS), for the purpose of the prototype, we assume the data is stored and accessed locally to simplify development and testing.

To support early experimentation, our team has been provided with a mixed dataset consisting of both labelled and unlabelled images. The object detection model will focus on five key classes: Fire, Smoke, Lightning, Vehicle, and Person.

3.3 Labeling Pipeline

We propose to develop a semi-automated image labeling pipeline to efficiently annotate a large volume of unlabeled wildfire imagery. This pipeline integrates object detection, image segmentation, and human-in-the-loop validation to improve annotation accuracy while reducing manual effort.

3.3.1 Input

The input to the pipeline will be a collection of unlabeled images obtained from the project's dataset.

3.3.2 Process

The labeling pipeline will proceed through the following steps:

3.3.3 Object Detection using You Only Look Once (YOLO)

We will first use **You Only Look Once (YOLO)**, a real-time object detection model, to generate initial bounding boxes and class labels for objects of interest such as smoke, fire, and vehicles.

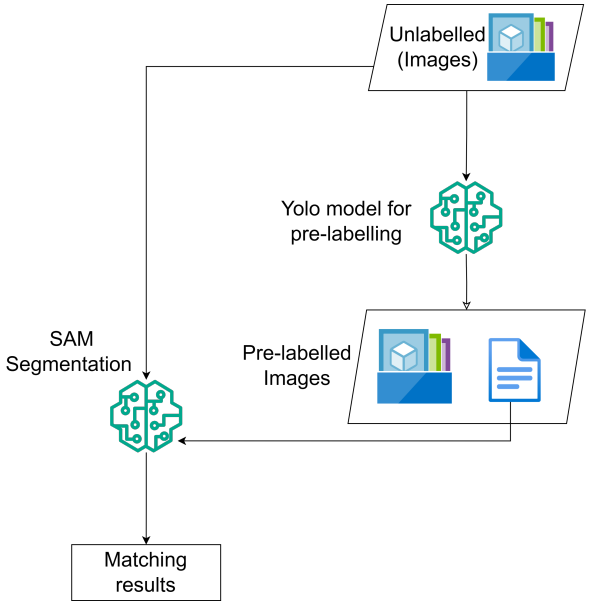

Figure 2: **Figure 3.1**: Overview of the proposed labeling pipeline combining YOLO for object detection, SAM for segmentation, and a matching process.

3.3.4 Image Segmentation using Segment Anything Model (SAM)

These YOLO-generated bounding boxes will serve as input prompts for the **Segment Anything Model (SAM)**, which produces precise, pixel-level segmentation masks of the detected objects. This allows for more accurate spatial annotation compared to bounding boxes alone.

3.3.5 Matching and Filtering

A key challenge is reconciling the outputs from YOLO (bounding boxes) and SAM (segmentation masks). We will define a matching criterion that uses either **Intersection over Union (IoU)**, which measures the overlap between the bounding box and segmentation mask, or **mask containment**, where we assess how much of the segmented area lies within the bounding box. If the overlap or containment falls below a predefined threshold, the annotation will be flagged for manual review.

Figure 3: **Figure 3.2**: Visual comparison of YOLO's bounding box and SAM's segmentation mask, highlighting the need for a matching criterion.

3.3.6 Human-in-the-Loop Review

Flagged annotations will be reviewed using **Label Studio**, an open-source annotation tool. Human reviewers will inspect and correct mismatched or low-confidence predictions to ensure the final dataset maintains high labeling quality. This step balances automation with manual oversight to maximize reliability.

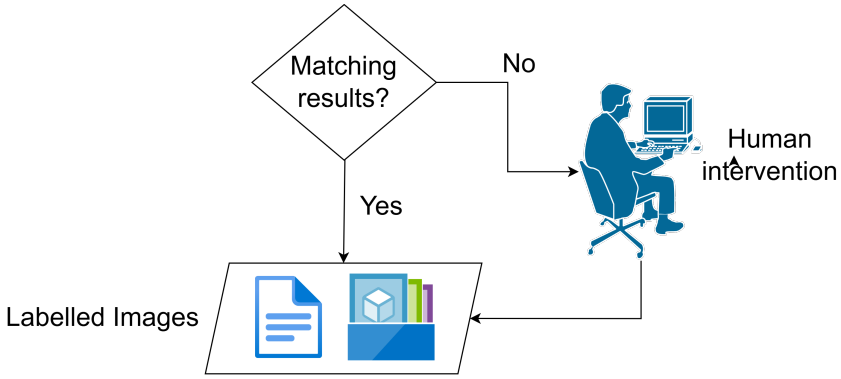

Figure 4: **Figure 3.3**: Human-in-the-loop flow using Label Studio to validate flagged predictions.

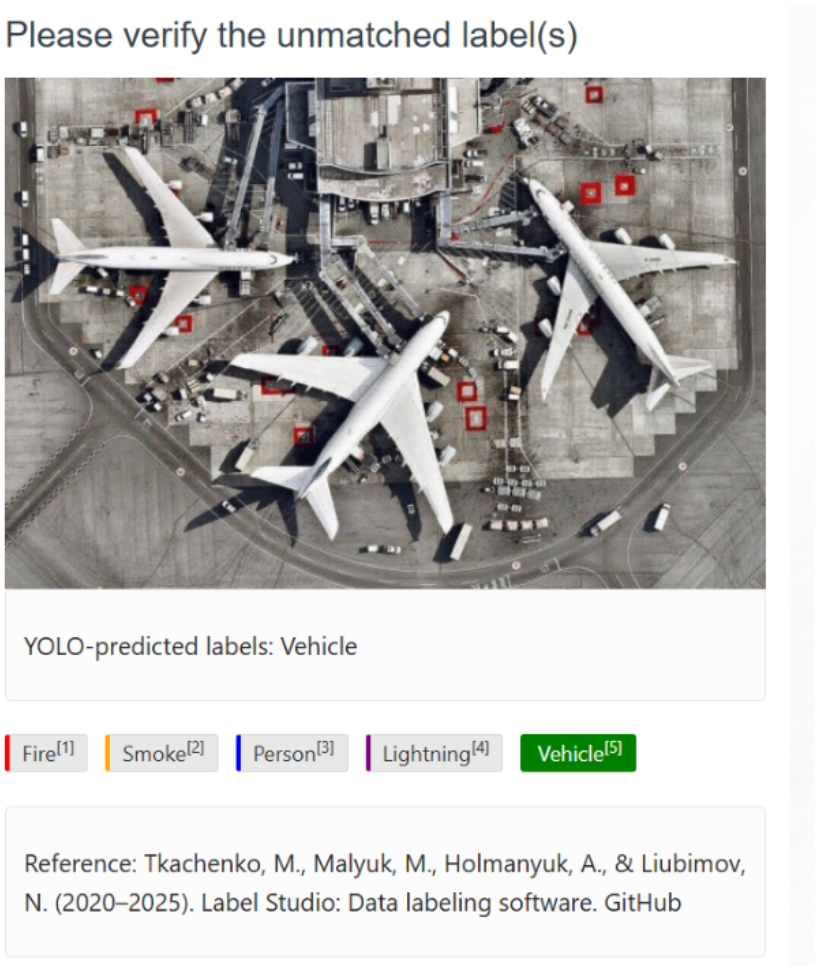

Figure 5: **Figure 3.4**: Label Studio interface displaying pre-labeled objects for reviewer validation.

3.3.7 Output

The final output of the labeling pipeline will be a set of high-quality annotated images. These images will either be auto-labeled with high confidence or validated through human review and will be used to train downstream models in later stages of the project.

3.4 Data Augmentation

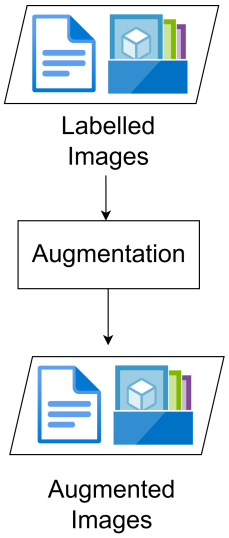

Figure 6: Overview of the augmentation pipeline

To improve generalization and robustness, we will apply data augmentation techniques to the labeled images. This will increase the size and diversity of our dataset, reducing model overfitting. Planned augmentations may include:

- Horizontal flipping
- Brightness and contrast adjustment
- Random cropping and noise injection

3.5 Model Training and Retraining

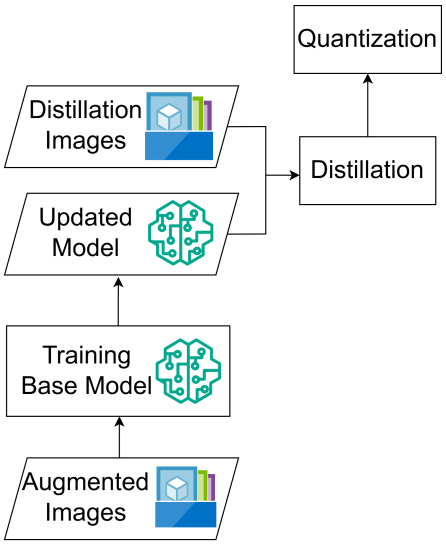

Figure 7: Overview of the model training, distillation, and quantization pipeline

Once we generate augmented images from the labeled dataset, we will use them to train a full base model. The training process will take as input:

Input:

- Augmented images
- A training configuration file specifying hyperparameters, model architecture, etc.

Output:

- A fully trained model file (`.pt`)

The fully trained model will be registered for future use: updating the pre-labeling model in the next iteration and serving as input to the distillation stage. We will repeat this training loop regularly as new labeled data becomes available, enabling continuous training across iterations.

3.6 Model Distillation and Quantization

After model training, we will apply distillation using a subset of the original dataset. The distillation module will take the fully trained model and the configured distillation dataset to produce a distilled model that is smaller and faster, while preserving accuracy.

Following distillation, we will apply quantization to further reduce model size and enhance deployment efficiency, enabling deployment on lightweight platforms.

Input:

- Full trained model (`.pt`)
- Distillation dataset
- A distillation configuration file specifying hyperparameters, model architecture, etc.

Output:

- Distilled model (`.pt`)
- Quantized model (`.pt`)

Both the distilled and quantized models will be stored in the model registry, with the latest quantized model pushed to Vertex AI as the deployable version.

4 Timeline

4.1 Timeline

With our proposed data pipeline, we've laid out our timeline across 7 tasks:

Task	Description	Date
1	Project setup and creation of the overall pipeline.	May 5 - May 9
2	Add pre-labeling + SAM check; implement human review interface.	May 12 - May 15
3	Apply data augmentation; integrate model training into the pipeline.	May 19 - May 23
4	Integrate distillation and quantization for deployment.	May 26 - May 30
5	Run full pipeline test to ensure end-to-end functionality.	June 2 - June 6
6	Submit runnable data product.	June 9 - June 11
7	Finalize data product and written report based on partner feedback.	June 14 - June 25

References

Defence Research and Development Canada. 2022. "DRDC Tests Early Detection Wildfire Sensors." https://science.gc.ca/site/science/en/blogs/defence-and-security-science/drdc-tests-early-detection-wildfire-sensors.