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 Product

To address the current bottleneck, we propose an intelligent, iterative data pipeline that integrates automation, human feedback, and model optimization for continuous improvement.

1. **Unlabelled Image Ingestion**
   The pipeline begins by collecting raw, unlabelled images from the client's data sources.
2. **Automated Pre-Labelling**
   AI models generate initial labels for key objects (e.g `Fire`, `Smoke`, etc), reducing the need for manual annotation.

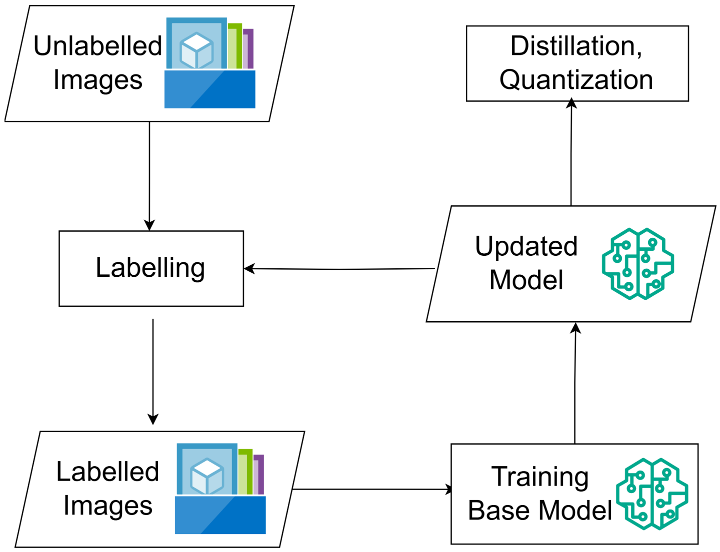

Figure 1: High-level overview of the data pipeline

3. **Human-in-the-Loop Review**
   Experts review and correct the pre-labels to ensure high-quality annotations with minimal manual effort.

4. **Model Retraining with Verified Labels**
   Verified labelled images are used to retrain the wildfire detection model, improving its accuracy and adaptability over time.

5. **Model Optimization for Deployment**
   Techniques like **distillation** and **quantization** compress the model, making it suitable for deployment on edge devices.

The pipeline is implemented through six modular scripts: `fetch_data.py`, `labelling.py`, `augmentation.py`, `train.py`, `distill_quantize.py`, and `save_model.py`. These modules may evolve during development.

3.2 Fetch the Data

3.2.1 Data description

The pipeline processes image data with bounding box annotations. For prototyping, we assume local storage, though the full dataset resides on GCS. It includes over 2 million unlabelled images, with 500 added monthly. The images contain five classes: `Fire`, `Smoke`, `Lightning`, `Vehicle`, and `Person`.

3.2.2 Process

`fetch_data.py` loads data locally (for dev/test purposes) but can later fetch from cloud storages.

3.2.3 Output

The following image directories should be produced:

- `raw/images` – used as input for the pre-labelling stage
- `raw/distilled_images` – used during the model distillation step

These folders are assumed to be locally accessible for the prototype.

3.3 Labeling Pipeline

The proposed semi-automated image labeling pipeline aims to efficiently annotate a large volume of unlabeled wildfire imagery. It 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 pipeline consists of three main stages implemented in three corresponding Python scripts.

1. Object Detection and Segmentation (`labeling.py`)

A function in this script first applies **You Only Look Once (YOLO)**, a real-time object detection model, to generate initial bounding boxes and class labels (e.g., fire, smoke, vehicle) for each image (Bochkovskiy, Wang, and Liao 2020). These predictions, including the class labels, are then passed to a second function that uses the **Segment Anything Model (SAM)**. SAM generates pixel-level segmentation masks for each labeled object using the corresponding bounding boxes as prompts (Kirillov et al. 2023).

2. Matching and Filtering (`matching.py`)

A key challenge is reconciling the outputs from YOLO (bounding boxes) and SAM (segmentation masks). A matching function in the script evaluates each pair using either **Intersection**

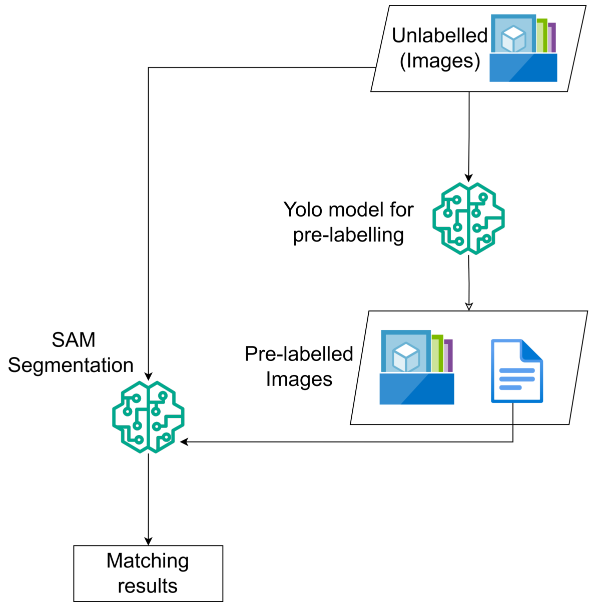

Figure 2: Overview of the proposed labeling pipeline combining YOLO for object detection, SAM for segmentation, and a matching process.

over Union (IoU), which measures the overlap between bounding box and segmentation mask, or **mask containment**, which assesses how much of the segmentation area lies within the predicted bounding box. If the match score falls below a defined threshold, the case is flagged for manual review.

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

3. Human-in-the-Loop Review (`human_intervention.py`)

Flagged cases will be passed to **Label Studio**, an open-source annotation tool, for human verification (Heartex 2021). Reviewers will inspect and correct mismatches or low-confidence predictions, ensuring high labeling quality. This hybrid approach balances automation with manual oversight.

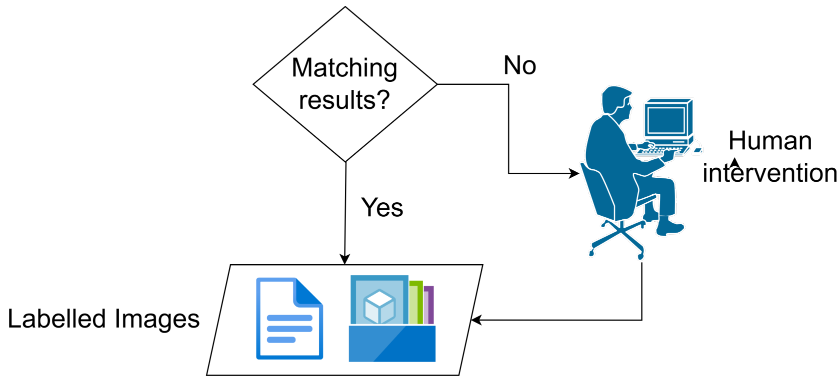

Figure 4: Human-in-the-loop flow using Label Studio to validate flagged predictions.

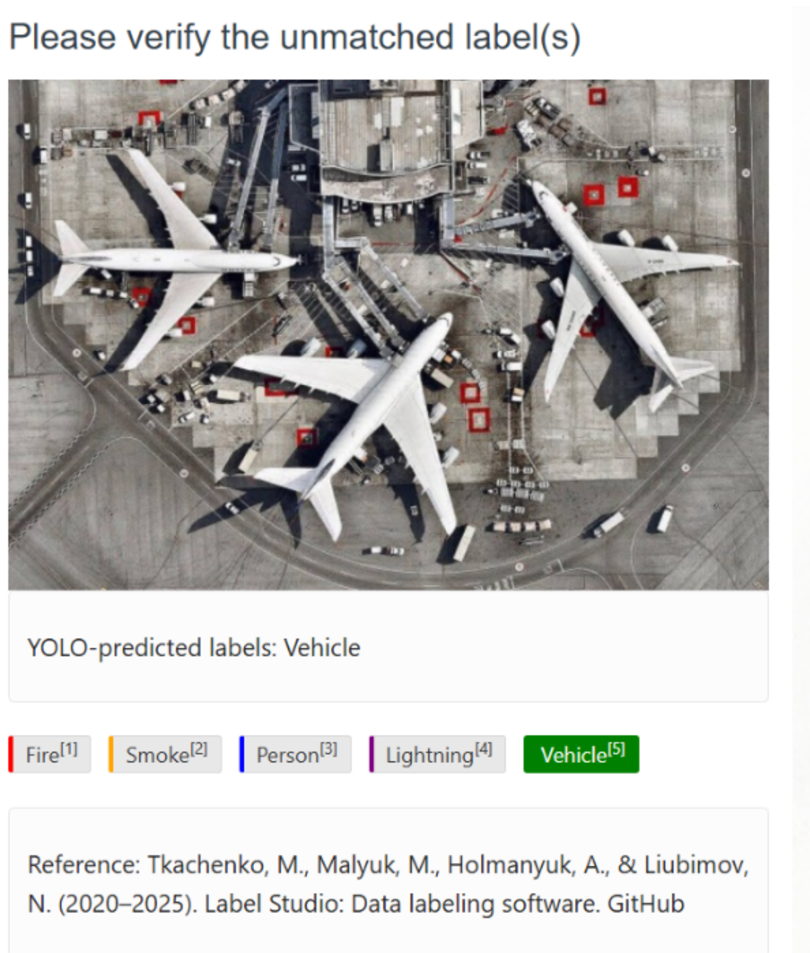

Figure 5: Label Studio interface displaying pre-labeled objects for reviewer validation.

3.3.3 Output

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

3.4 Model Optimization Pipeline

The model optimization pipeline consists of 4 core stages that follow the labeling phase: data augmentation, model training, distillation and quantization, and final model saving. These stages are implemented through a series of Python scripts that prepare the model for production use.

3.4.1 Input

The input to this pipeline will be the labeled images from the labeling pipeline, distillation images, and configuration files for augmentation, training, and distillation.

3.4.2 Process

1. Data Augmentation (`augmentation.py`)

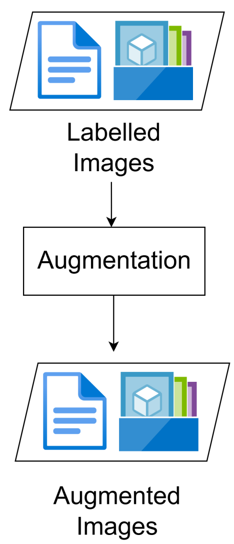

Figure 6: Overview of the augmentation pipeline

This script increases the size and diversity of our dataset through common augmentation techniques (e.g., flipping, brightness/contrast adjustment, noise injection), as specified in the configuration file. These augmentations help reduce overfitting and improve generalization during model training.

2. Model Training and Retraining (`train.py`)

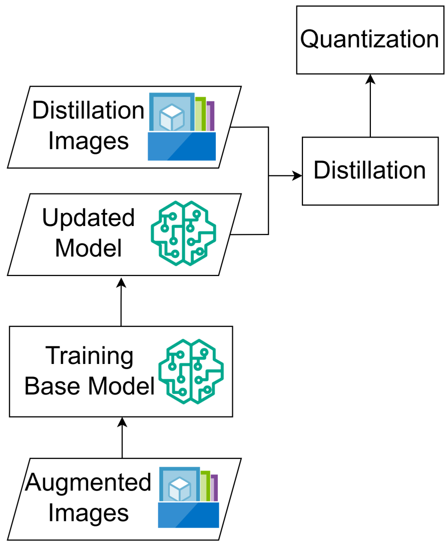

Figure 7: Overview of the model training, distillation, and quantization pipeline

This script fine-tunes the YOLOv8 base model using the augmented dataset and a configuration file, and produce a full trained model file (`.pt`). The full trained model will be registered for use in future labeling iterations and passed on to the distillation stage for optimization.

This training process will be repeated as new labeled data becomes available, enabling continuous model improvement over time.

3. Model Distillation and Quantization (`distill_quantize.py`)

This script optimizes the trained model in two stages. First, it performs **distillation** using the distillation images to produce a distilled model that is smaller and faster, while preserving accuracy. Then it applies **quantization** to further reduce model size and enhance deployment efficiency, enabling deployment on lightweight platforms.

4. Deployment (`save_model.py`)

This script finalizes the pipeline by registering all three model versions in the model registry. The quantized model will be marked as the current production version for deployment.

3.4.3 Output

The optimization pipeline will produce the following outputs:

- Full trained model (`full_trained_model.pt`)
- Distilled model (`distilled_model.pt`)
- Quantized model (`quantized_model.pt`)
- Updated configuration files for training and distillation

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

Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. "YOLOv4: Optimal Speed and Accuracy of Object Detection." https://arxiv.org/abs/2004.10934.

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.

Heartex. 2021. "Label Studio: Data Labeling Platform." https://labelstud.io/.

Kirillov, Alexander, Eric Mintun, Nikhila Ravi, et al. 2023. "Segment Anything." https://segment-anything.com/.