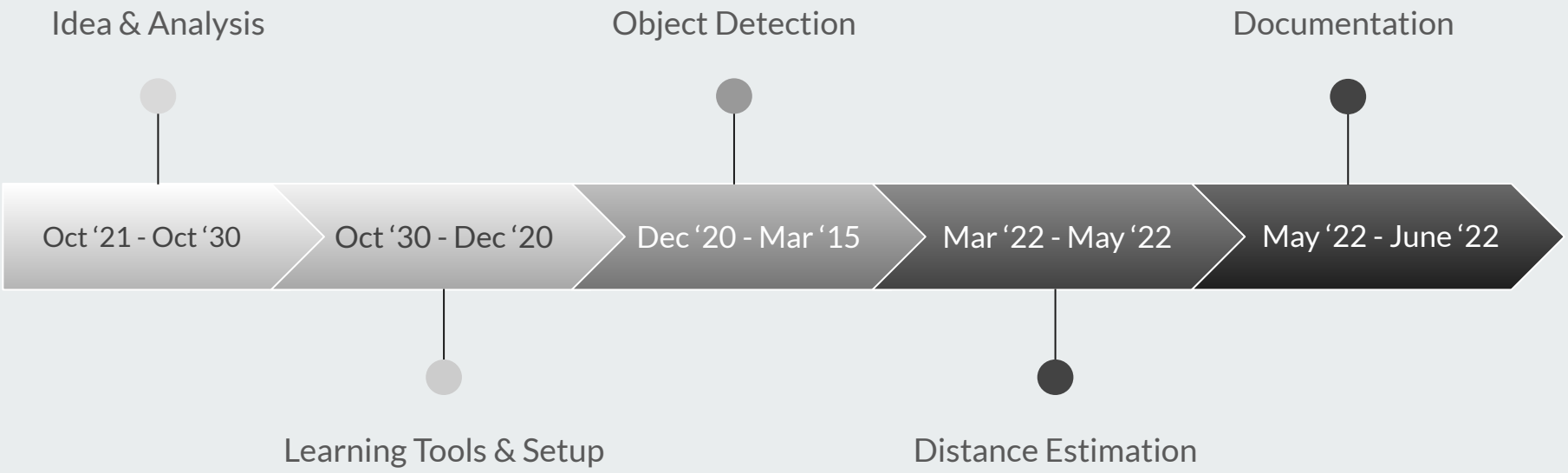
Distance Estimation using Thermal Imaging - Driving Assistance

Agenda

What we have done!

- Project Timeline
- Phase A: Specifications & Setup
- Phase B: Development
  - Core A: Object Detection
  - Core B: Distance Estimation
- Phase C: Integration
- Phase D: Documentation

Project Timeline



Phase A: Specifications & Setup

In this phase, we worked on the following main aspects:

Selecting Idea & Competitive Analysis - What

- The most important criteria we used in evaluating ideas to choose the appropriate idea is uniqueness, and it solves a real problem. Therefore, we changed our minds many times.

Phase A: Specifications & Setup

Selecting Idea & Competitive Analysis - What

- After selecting the idea, we proceeded to search for more information about the problem to learn more about all aspects of the problem, its available solutions, and the advantages & disadvantages of its available solutions, if they exist.
- The information we got through our search helped us to build an intuition about how can we solve this problem & how to work on currently available solutions shortcomings.

Phase A: Specifications & Setup

Selecting Idea & Competitive Analysis - How

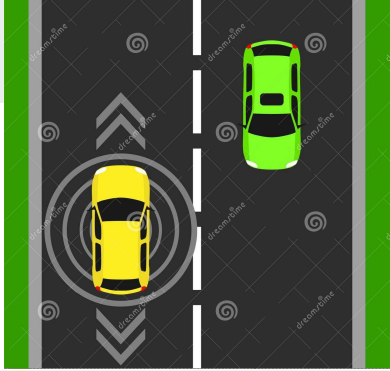
After exploring **what** we have done about selecting idea & analysis, lets see **how**.

- Problem & Challenge - Summary
    - Problem: drivers face vision problems while driving in unclear vision (e.g. foggy weather and darkness), which leads to many accidents.
    - Challenge: How to reduce car accidents due to the unclear vision during driving?

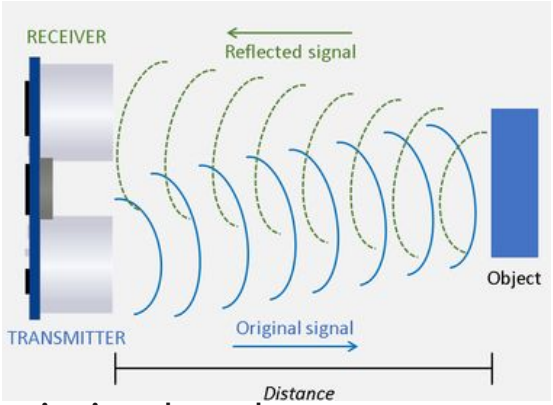
Phase A: Specifications & Setup

Selecting Idea & Competitive Analysis - How

- Available Solutions - Summary
    - Detection, Tracking and Estimating Objects' Distance, Direction & Speed using Normal Imaging
        - Uses a DL model trained on normal images to detect objects and estimate their distance and speed.
        - This solution works well in clear vision. But in the presence of clouds, fogs, and darkness, we have a problem.



Phase A: Specifications & Setup



Selecting Idea & Competitive Analysis - How

- Available Solutions - Summary
    - Arduino Ultrasonic Sensors
        - Estimates the distance only.
        - It has two main components: transmitter which transmit signals and receiver which receive reflected signals.
        - But can’t detect objects whose distance more than 20m. And can’t detect objects which move with high speed.

Phase A: Specifications & Setup

Selecting Idea & Competitive Analysis - How

- Available Solutions Pros and Cons- Summary

Solution	Pros	Cons
Detection and Estimation using Normal Imaging.	- estimate distance.	- Doesn't work in unclear vision.
Arduino Ultrasonic Sensor	- Works well regardless of vision. - Measures distance.	- Can't detect objects which distance farther than 70 feet and move at high speed.

Phase A: Specifications & Setup

Selecting Idea & Competitive Analysis - How

Big Question: How our solution will overcome available solutions shortcomings?

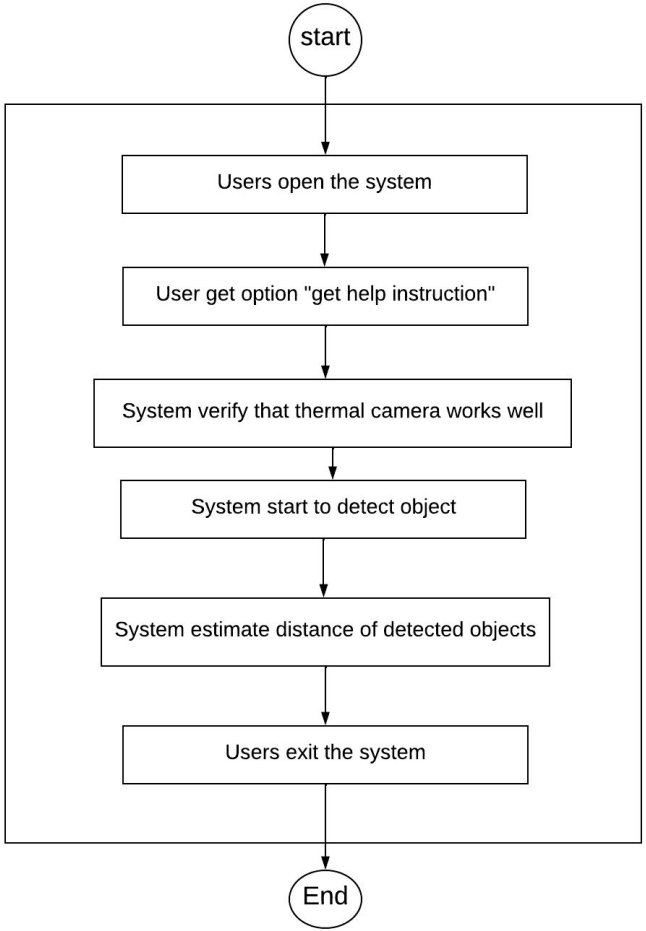
- Our Proposed Solution - Summary
    - We solve this problem by estimating **distance** for detected objects in the roads using thermal imaging.
    - We will use distance the detected objects to provide some information to the driver which in turn will reduce car accidents.

Phase A: Specifications & Setup

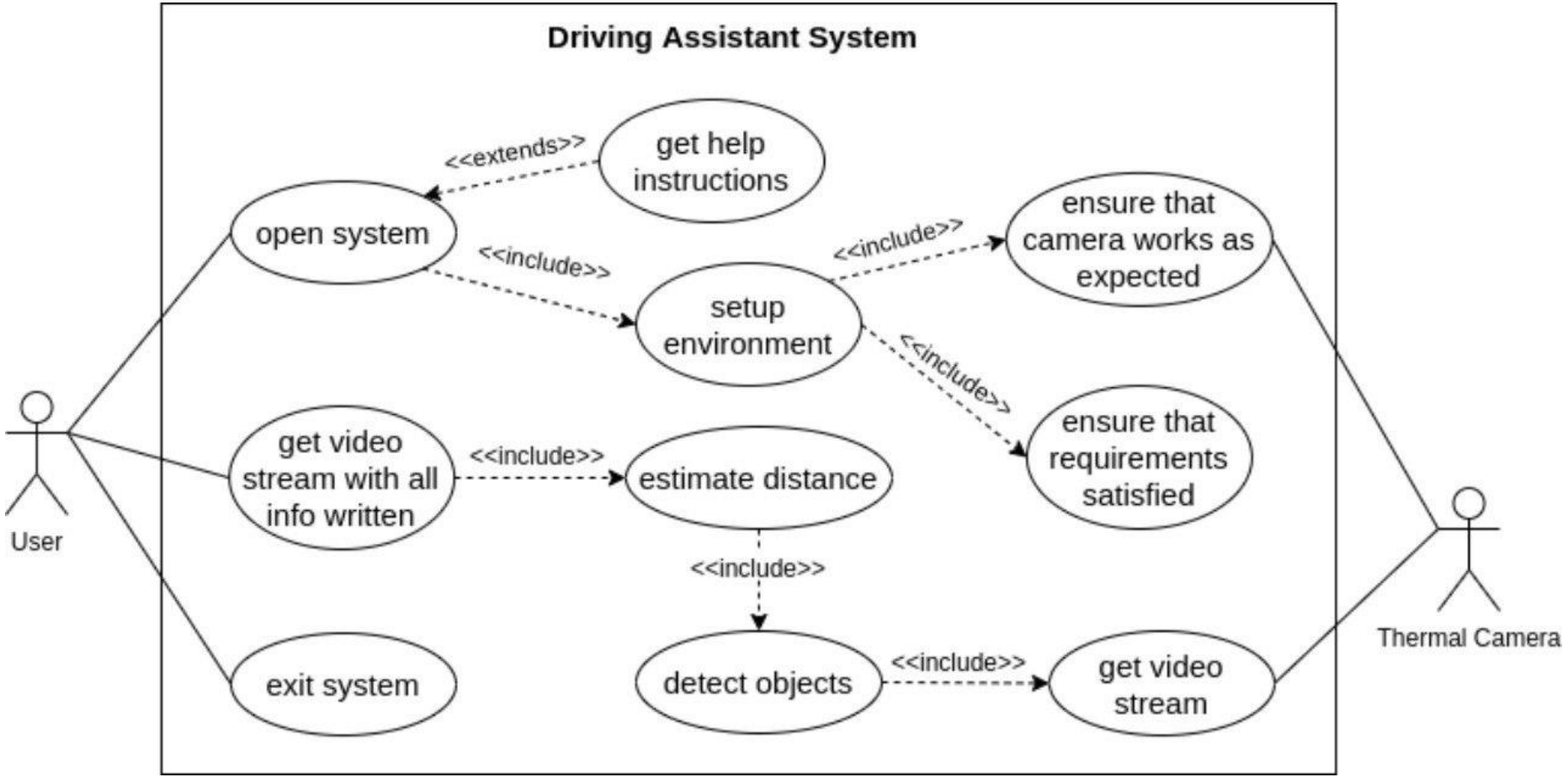
Exploring Scenarios & Build Initial Architecture

Exploring user scenarios and how the user would interact with the system enabled us to specify requirements used to build an initial architecture that satisfies these requirements.

Scenario Example



Use Case Example



Phase A: Specifications & Setup

Learning Tools & Technologies

I and my team didn't have any prior knowledge about ML/DL. Maybe minimal knowledge about AI classical algorithms discussed in AI introductory courses.

Therefore, we took an amount of time to learn the basics of these topics and to familiarize ourselves with technologies & tools we will use.

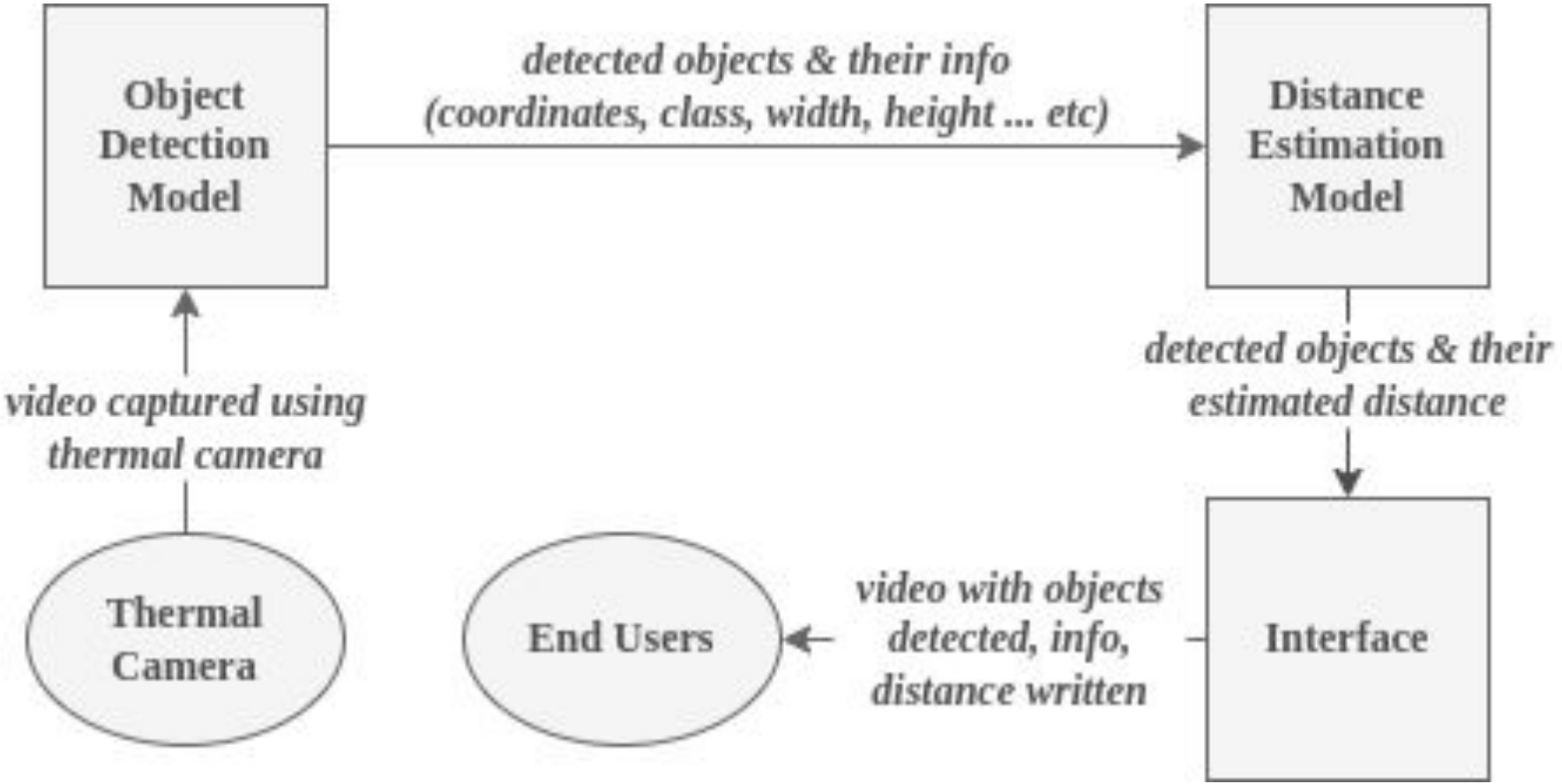
Phase B: Development

In this phase, we are working on the following two parts:

- Object Detection
- Distance Estimation

Before explaining every part in detail & what we have done. Let's take a look at the high-level conceptual architecture of our system.

Architecture



Phase B - Core A: Development - Object Detection

In this part, it is supposed to develop a DL model to help us detect objects in thermal images/videos, identify their types/classes (e.g. car, person), and determine the coordinates of their locations.

We will use this information afterward to estimate the distance between the objects and source camera.

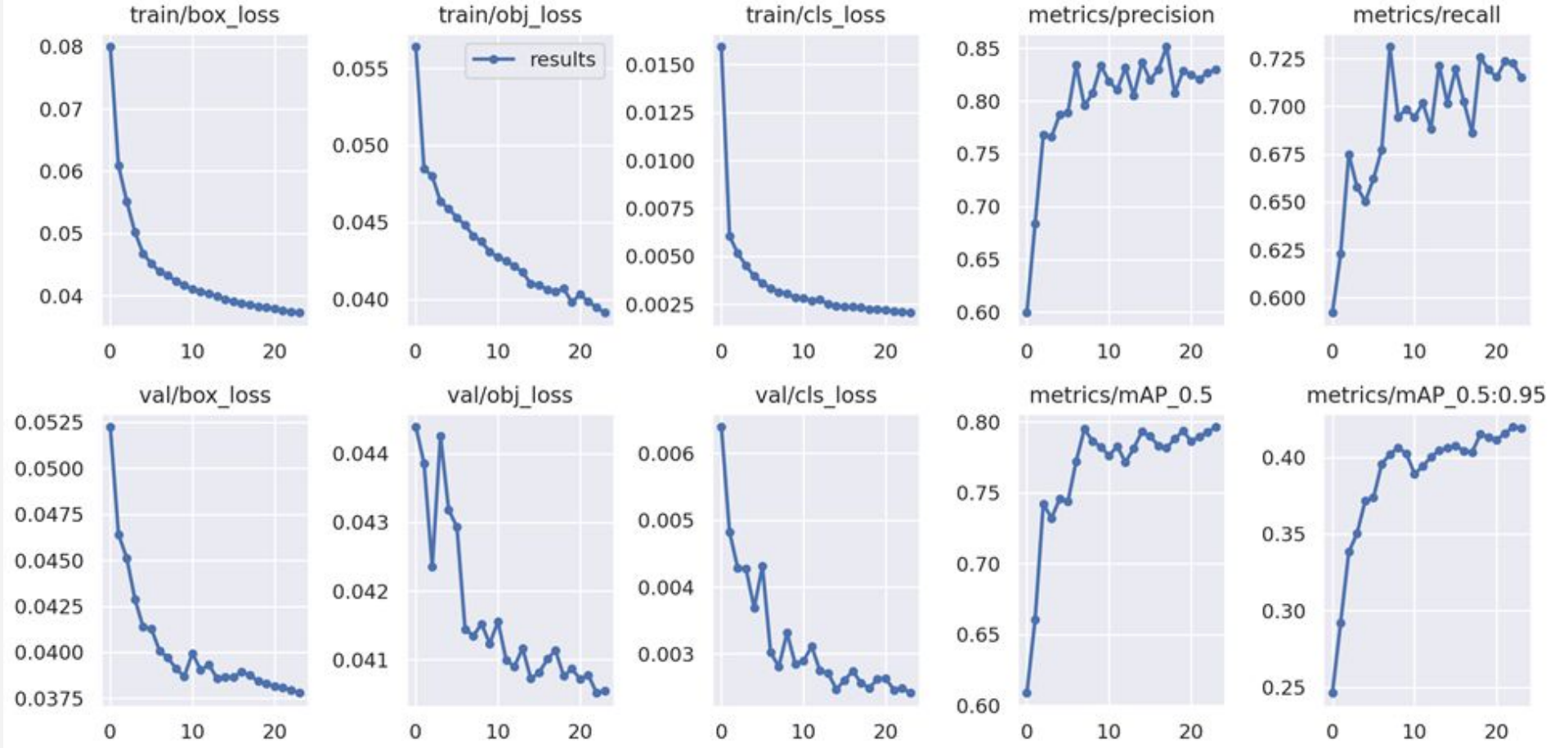
Phase B - Core A: Development - Object Detection

- We built a custom thermal images dataset for YOLO v5 (darknet compatible).
- We used custom dataset to fine-tune a pre-trained YOLO v5 model for detecting and classifying objects (e.g. vehicle, person) from thermal images/videos.
- Consider the input of the object detection model is a video. Then the output is a video with detected and classified objects bordered with a box around them.

Phase B - Core A: Development - Object Detection

- We modified the model output format to output a generated sheet containing detected objects coordinates.
- The sheet has a row for each frame and a column with coordinate (xmin, ymin, xmax, ymax).
- We will use information included in the sheet in the next step (distance estimation).

Object Detection Model Results



Object Detection Output - Examples

Object Detection Output - Examples

*unannotated image - validation set

*its annotated version - validation set

*annotated version - model's output

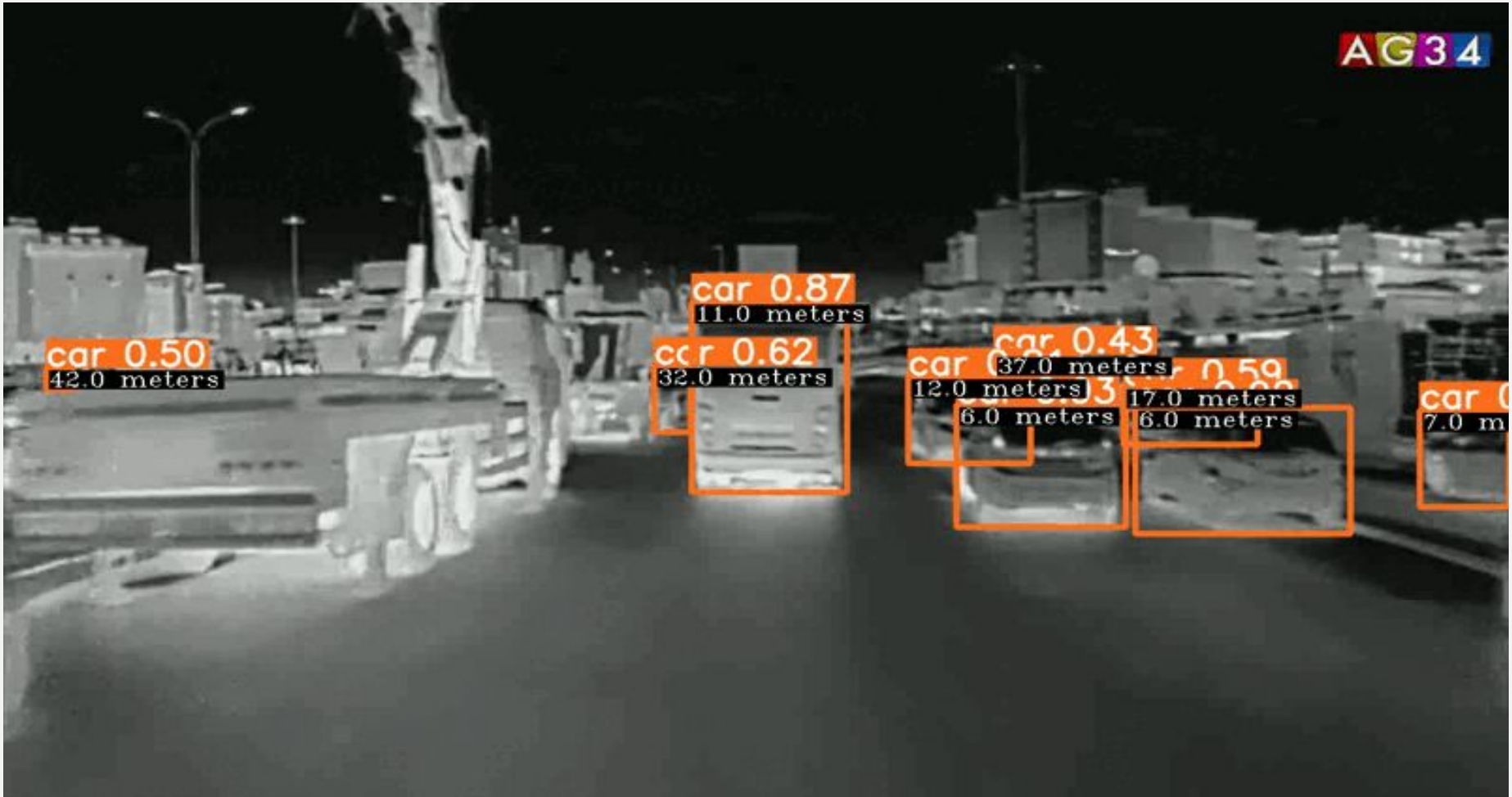
Phase B - Core A: Development - Distance Estimation

- Our goal in the part to inform the driver of the distance between him and the surrounding objects to avoid accidents.
- In this part, we worked to build a model that estimates the distance using object detection results.
- Model uses the detected objects coordinates to determine the distance between the source camera and the objects.

Phase B - Core A: Development - Distance Estimation

- We used x-axis and y-axis and estimated z-axis positions to estimate the distance.
- Then we use this predictions to create a nice visualization to the user - write this info back to the different video frames and generate the original input video with all info written back to it.

Distance estimation Output - Examples



Phase C: Integration

We have integrated two models. The first model expected to take a video captured by a thermal camera and outputs video frames annotated and annotation sheet with the coordinates of detected objects. The second model takes the first model output as an input and then estimate the distance and visualize the results (e.g. write the info back to the video / stream).

So, this means that the distance estimation model is stacked after object detection model.

Phase D: Documentation

We have worked on documentation for the project, tutorial notebooks explaining how to run, train, and use models to detect or estimate some results.