

ProjectPlan

ENEE408N Capstone Project Plan

Project Title

Aiding Blind Individuals with Camera-Based Object Detection and Depth Sensing

Project Team

- **Naitik** – Project Planner, API Implementation Research, Emotion and person Detection
- **Micah** – Object Detection Research & Implementation
- **Eliav** – Depth Sensing Research & Implementation

Project Overview

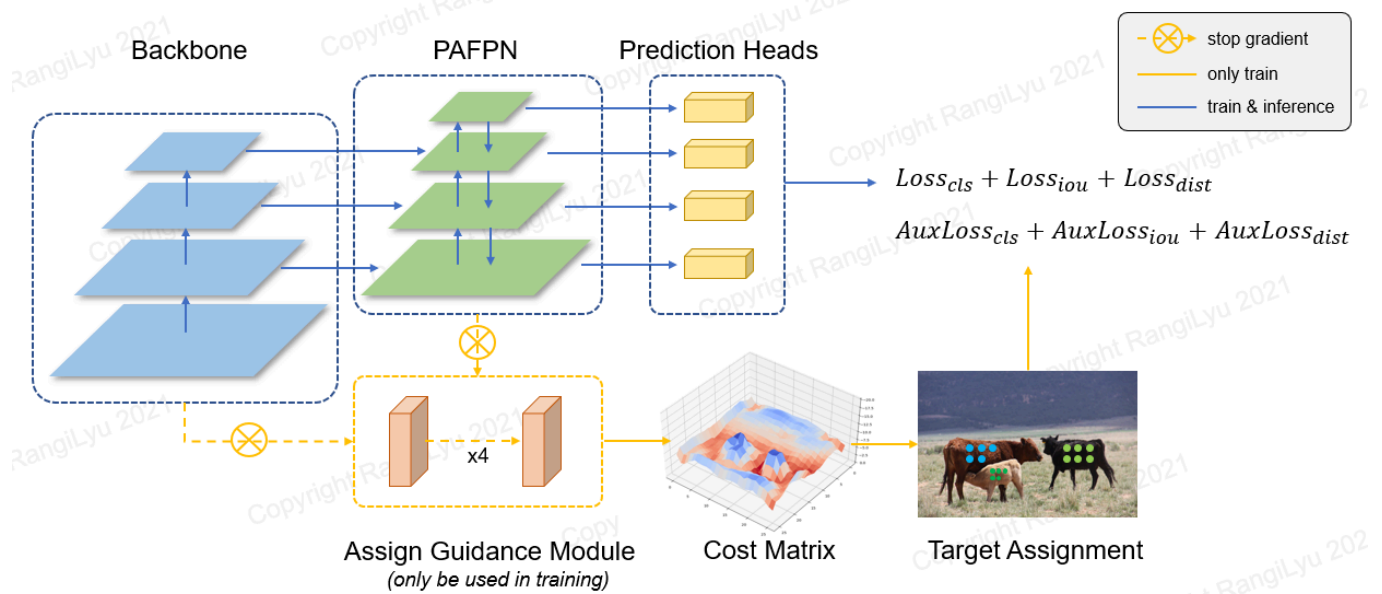
NaviGatr seeks to enrich the environmental awareness of the visually impaired by utilizing machine learning models.

NaviGatr will run three models: an object detection model, a depth estimation model, and an emotion classification model.

By capturing an image and synthesizing the outputs of the models, we can inform users via an auditory prompt about what objects are nearby, where they are, and the emotional states of nearby people.

NanoDet Project Model

This project leverages the NanoDet pretrained model. Compared to similar models, the NanoDet is extremely lightweight at only 2.3[MB] for the NanoDet-Plus-m version. During comparative testing against other YOLO models, we found only the YOLOFastestV2 to be of better performance in speed. The NanoDet had superior accuracy of 30.4[mAP] compared to 24.1[mAP] for the YOLOFastestV2. Additionally, the NanoDet is an anchor-free model which is why its memory profile is very low.



Accuracy [mAP]	Speed [ms/img]	Size [MB]	Resolution [px-by-px]
27.0	5.25	2.3	320x320

Source: <https://github.com/Rangilyu/nanodet>

Because of NanoDet's speed, low memory size, and efficient power consumption; we chose this model as our foundational model for the project scope.

This model is planned to be implemented on a RaspberryPi4 or similar.

<https://github.com/Qengineering/YoloX-ncnn-Jetson-Nano?tab=readme-ov-file> This repository is a benchmark assessment for object detection on many YOLO models. Since all models were used on the same test set and under similar conditions, it gives a fair comparison between models and edge device implementation.

<https://github.com/Rangilyu/nanodet> Official NanoDet-Plus repository. The ideal benchmark metrics and architecture layout is described throughout the README of the repository.

Project Workflow & Timeline

Week 3/2: Research & Setup (Current Week)

- **Naitik:** Creating the project plan & researching camera API integration.
- **Micah:** Researching object detection techniques and existing models.
- **Eliav:** Researching depth sensing methodologies.
- **Team Task:** Set up a **Git repository** and establish an **Agile workflow** for managing tasks efficiently.

Week 3/9: Initial Development (Pre-Spring Break)

- **Micah:** Begin implementing object detection.
- **Eliav:** Begin working on depth sensing.
- **Naitik:** Begin working on Emotion Detection and Person Detection
- **Plan:** Object detection will be prioritized first, followed by depth sensing integration followed by emotion detection

Week 3/16:(Spring Break)

- Try to work on things but mostly break

Week 3/23: Finalizing Models

- **Micah:** Finalize the object detection model.
- **Eliav:** Finalize the depth sensing model.
- **Naitik:** Finalize the emotion detection model.

Week 3/30: Integration & App Development (Milestone review)

- **Project Tasks**
 - **Team Task:** Integrate object detection, depth sensing, and emotion detection into the camera application. Implement full camera API integration.
 - **Testing:** Validate system functionality and performance.
- **Milestone Tasks(Due Friday)**
 - **Main Template:**
 1. Main Motivation
 2. Objectives
 3. Approaches
 4. Key Results/ Takeaways
 5. Key Issues Encountered
 6. Pivot Or Adjust.
 7. Plan For the Remaining Weeks
 - Slides
 - **First Slide:** Introduction To the Project
 - 1 , 2
 - **Second Slide:** Approach to the Project
 - 3
 - **Third and Fourth:** Object Detection

- 4 Through 7
- Talking points described below.
- **Fifth and Sixth::** Depth Sensing
- **Seventh and Eighth::** Emotion Detection
- **Naitik:** Research and understand current model. Justification for the same(To include comparing alternate models, EfficientNet and FER Dataset). Current Progress[2 Slides]
- **Eliav:**
- **Micah:** Research and understand current model. Justification for the same(To include comparing alternate models). Current Progress[2 Slides]

Week 4/06: Hardware Integration

- **Deadlines**
 - April 7th: Milestone Review Due
 - April 8th: Presentation due
- **Team Task:** Porting from computers to Raspberry Pi
- **Questions:** Who is paying for the hardware, is there a capstone fund
- **Micah:** Extensions, models, Transformers, modules specific to raspberry pi from regular computers.(Involves making a branch)
- **Eliav:** Webcam Integration with the project and hardware.
- **Naitik:** Setting up ArchLinux, Cora TPU, pytorch, git repository.

Week 04/13: Advertisement and presentation

- **Team Task:** Working on banners, Presentations, demonstration.
- **Questions:** Who is paying for the banners, is there a standardized setup
- **Micah:** Key points to talk about, present, demonstration plans.
- **Eliav:** Banners, posters, Trifold
- **Naitik:** Presentation for slideshow on one of our computers.

Week 04/20: Buffer Week1

- **Team Task:** Integrate object detection, depth sensing, and emotion detection into the camera application.
- **Naitik:** Implement full camera API integration.
- **Testing:** Validate system functionality and performance.

Week 04/27: Finalize all aspects and recap

- **Team Task:** fully integrate project, test raspberry pi, proof reading all our stuff, running demonstrations for our projects
- **Testing:** Validate system functionality and performance.

Week 05/04: Rehearsals

- **Team Task:** Integrate object detection, depth sensing, and emotion detection into the camera application.
- **Naitik:** Implement full camera API integration.
- **Testing:** Validate system functionality and performance.

Expo Date : May 7th 2025 Wednesday

Development Plan

1. **Object Detection Phase** (Lead: Micah)
 - Identify suitable object detection models (YOLO, SSD, etc.).
 - Train/test the model with real-world objects.
 - Optimize performance for real-time processing.
2. **Depth Sensing Phase** (Lead: Eliav)
 - Implement depth estimation using stereo cameras or LiDAR.
 - Integrate depth information with detected objects.
 - Fine-tune depth accuracy.
3. **Emotion Detection and Person Detection Phase** (Lead: Naitik)
 - Try out currently existing emotion detection models alongwith person detection models
 - Train/Test models with our specific team members(improve detection for specifically us)
 - Develop a database with people for our specific class. (Optional, add integration to add a specific person to the database using the model for future meets)
 - Run model and test accuracy with person and emotion detection
4. **Integration & Testing**
 - Merge object detection, depth sensing, and emotion detection.
 - Validate system performance.
 - Optimize latency and accuracy.

Agile Workflow

- Weekly sprint planning & stand-ups.
- Tasks managed via Git issues/boards.

- Continuous integration and testing.

Next Steps

- Complete research phase.
- Define object detection model for implementation.
- Prototype initial object detection functionality.
- Plan depth sensing integration strategy.

This document will be updated as we progress through the project.

This project was developed with the aid of ChatGPT, specifically the structure of the document.

Last updated: 3/7/2025