Perception Engine: PRD v2

Capstone Project

Team Perception
ucsbteamperception@gmail.com

Problem

Our partnered company, Aerospace Corp., seeks to analyze a large video database, in which objects should be recognizable and trackable. Currently, Aerospace does this manually by having humans watch the video and map out the object trajectories. But, now they would like to automate that process.

Project Outcome

The video data with an overlay to depict the trajectory/path of an object of interest is generated by our program automatically.

Specifications

- Input: Target Image, Video recording
- Output: Streamed (x, y) coordinates, (height, width) bounding box of the target object. At end, a processed video is outputted which has the tracking overlay.
- Action: The program finds the object in the video and outputs the current location for bounding object and it's size.

Design

- Convolutional neural network to identify desired object
- Convolutional neural network to identify object in video
- Computer vision to locate and track identified object in video
- Post processed video has overlay tracking image

Project Milestones

1: There is an intuitive GUI to interact with our program
2: Convolutional neural net(CNN) correctly identifies objects in given picture
3: CNN correctly identifies objects in a set of video frames
4: CNN correctly identifies all objects in the video
5: The desired object and its representation in the video are linked together (lables are mapped to each other).
6: Desired object coordinates in the video are correctly outputted
7: Desired object is correctly tracked in video through an overlay
8: Basic web-app frontend for client interactions

Stretch Goals

Unordered

- Run it on an embedded system
- Run object tracking on a live video stream
- Stream video to web-app as its being processed in real-time
- Basic iOS-app frontend
- NLP parsing of desired object

- Native iOS-app using CoreML and ARKit
- Use a Siamese Neural Networks instead of a CNN

Team Perception

Position	Name	Contact
Team Lead	Abhishek Bhattacharya	Abbirdboy@gmail.com
Team Scribe	Danish Vaid	danishvaid@umail.ucsb.edu
Developer	Ben Patient	bpatient24@gmail.com
Developer	Jake Can	jakecan808@gmail.com
Developer	Sai Srimat	srimat.sai@gmail.com

Backgrounds

Abhishek:

Abhishek is an undergraduate honors student at UCSB majoring in Computer Science and Biology. He is a member of the prestigious College of Creative Studies at his campus which recognizes a select group of students for their excellence in academics and research. In the last few years, he has worked with leading scientists at Stanford, UCSF and UCSB to uncover the potential of big data in computational biology. He is an avid computer programmer and software engineer, and his interests lie in deep learning precision interpretation of medical imaging for the clinics. He is involved with a number of ongoing projects in skin and breast cancer quantitative image analysis.

Danish:

A fourth year software-oriented student with a solid foundation in multi-platform object-oriented design and web development, with experience in the planning, debugging, implementation, and maintenance stages. Prior experience at NASA and AppFolio in C++ object mapping and full stack web design - skills to be utilized on our project for the web application and object tracking.

Ben:

I worked at Beachbody LLC for a year as a Video Streaming Engineer Intern where I managed the back-end workflow of content delivery. I wrote shell scripts and configured and debugged existing python scripts that encoded videos for streaming. I also interned at Belkin International as a QA Intern where I did QA on android and iOS apps and communicated with developers to resolve issues. Here I learned how to do proper tests and I plan to use this experience to help me write good tests when doing TDD. I have work and classroom experience in C++, python, Java, and bash and will utilize these skills in our capstone project.

Jake:

At SmartOhme, I worked on the messaging service between the servers and devices. This experience will help me design the routing of information in our system. As a Computer Engineer, I've decided to emphasize my academics on Computer Science, rather than ECE courses,

giving me experience in software design. I've taken a course in Artificial Intelligence which would help in developing the neural network engine for our object detection feature.

Sai:

My Computer Science background begins on my first day at UCSB and has been growing for the past 3 years. A combination of courses ranging from principle Data Structures and Algorithms all the way to Distributed Systems, AI, and Embedded Systems has budded my potential for growth in this field. My CS knowledge has been further supported by an internship at Raytheon this past summer. My experience at Raytheon is relevant insofar as the defense industry's process of product creation and management. I believe the knowledge I gained at Raytheon can be indirectly applied to the mindset necessary for Aerospace Co. and their endeavors with this project. Also, the industry tactics, such as Agile, Scrum, and TDD helped give me an idea of what to expect in terms of the development process for our Capstone Project.

Requirements

- User stories and tasks can be found on Trello at: `https://trello.com/b/4z2p6hip/capstone`
- Prototyping code and Test metrics can be found (as they come up) at our GitLab Page at: `https://gitlab.com/ucsbteamperception/perception`
- Scribe and Retro notes can be found on our Doc at: `https://docs.google.com/document/d/1SHMsGn1pRrUkvDc2wDAhb6xK2RfUuaiB6lPT6_znKJY/edit?usp=sharing`

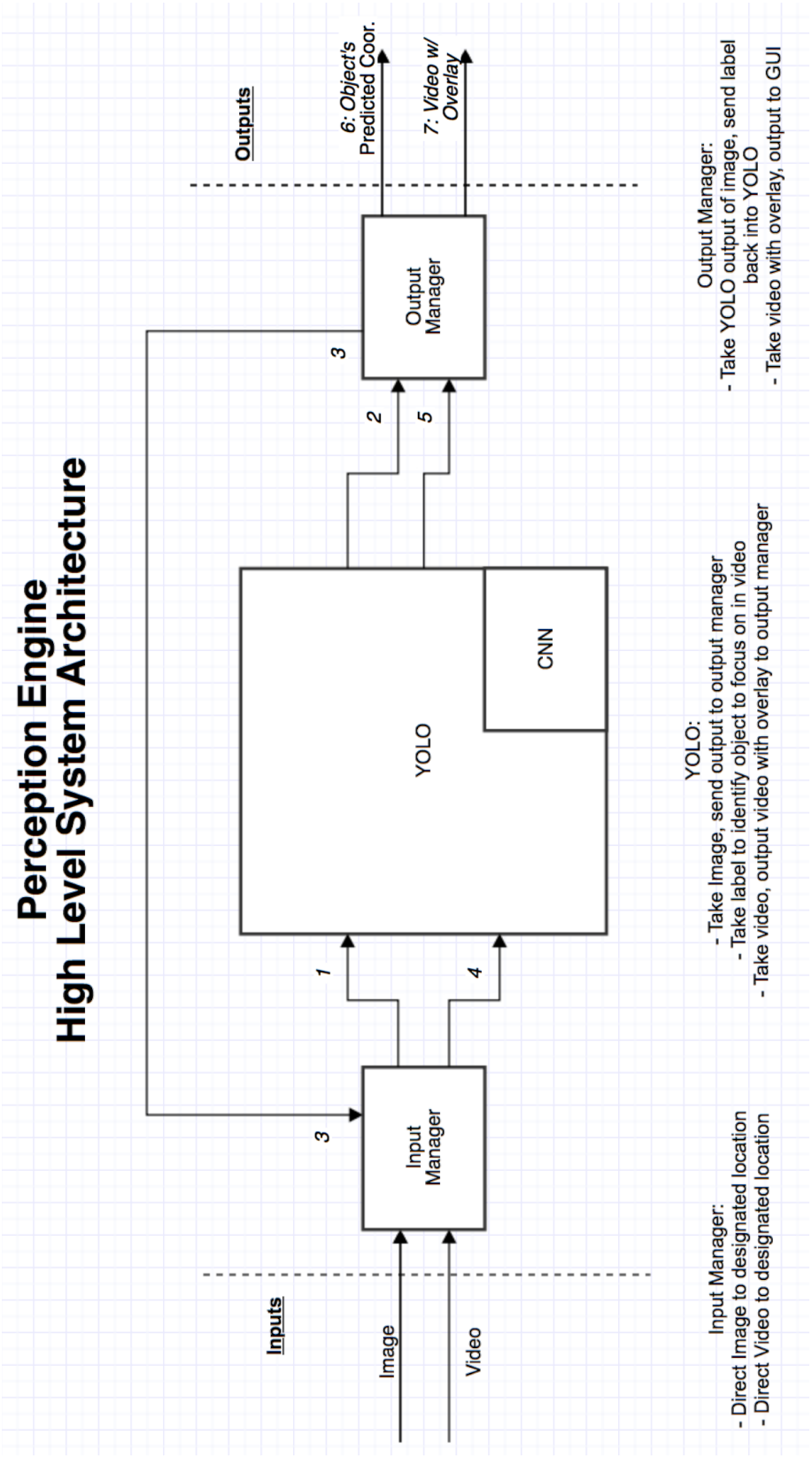
Technologies Employed

Implementation platform and technologies

1: Python base project language
2: Flask for webserver (JS, HTML, CSS Front End)
3: OpenCV Library for object tracking
4: Flask_dropzone for webapp file uploading
5: PyTorch for Siamese Neural Networks (Stretch Goal)

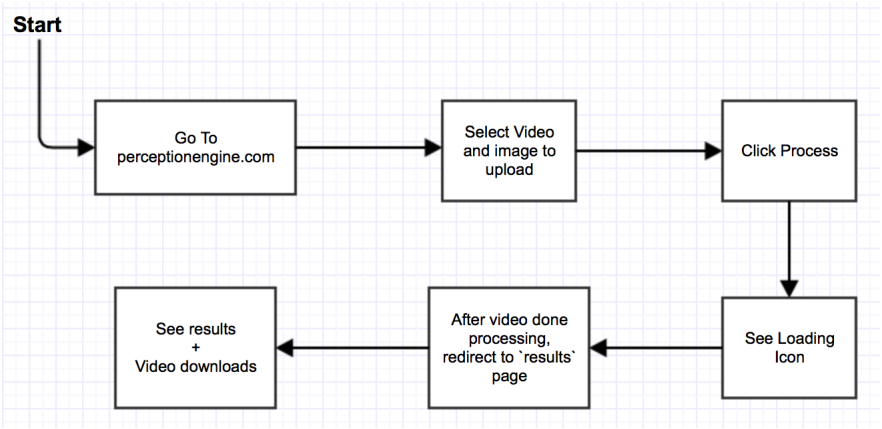
Process model to acheieve milestone

1: Anaconda/PyEnv for enviroment management
2: GitLab for project repository
3: GitLab Issues for issue tracking
4: GitLab CI/CD for continuous integration/continuous delivery
5: Trello for task management
6: Excel for burndown and retrospective data
7: Google Doc for scribe notes



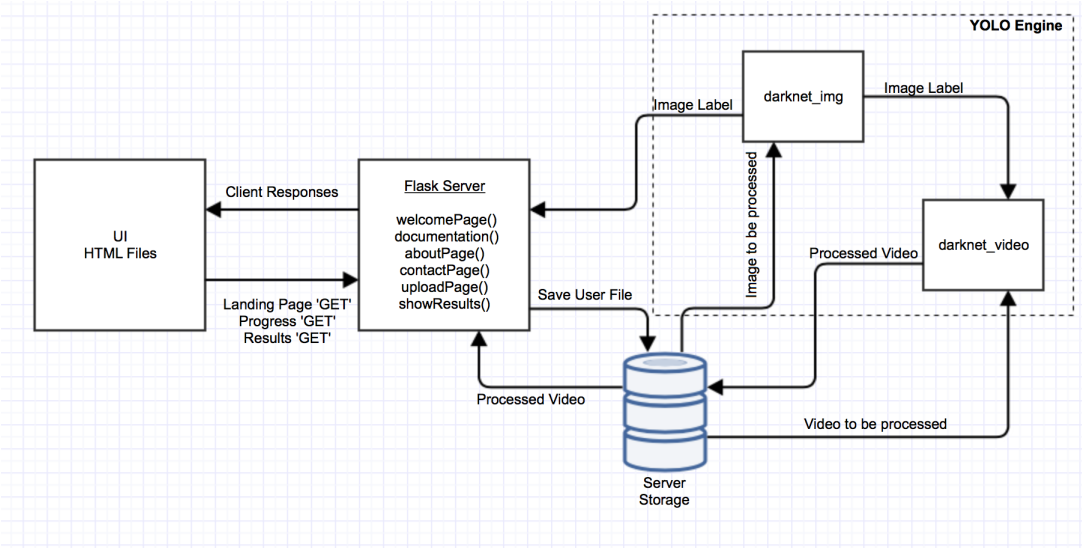
Landing Page UI Design

Landing Page UI Design



We are sticking to a minimalistic design so that our product has the main focus, without distraction. Planning on adding a bit more color and design to the current minimalist layout in the future. Looking for a good logo design for our team.

System Model



YOLO Engine Break Down

Image		Video	
file:	function	file:	function
darknet.c:	main()	darket.net:	main()
detector.c:	test_detector()	detector.c	run_detector()
		demo.c:	demo()
		demo.c:	display_in_thread()
		demo.c:	detect_in_thread()

YOLO Schematics Modifications

Summarized for our use

```
[darknet_image] command: ./darknet detect cfg/yolo.cfg ../shared/yolo.weights
    ↪ image_path
     darknet.c:
         main(tag)
             if tag == "detect"
                 test_detector() in [detector.c]
     detector.c:
         test_detector()
             - turned off opencv pop-up
             - makes image prediction
             - outputs labels, accuracies, bounding box coords


[darknet_video] command: ./darknet detector demo cfg/coco.data cfg/yolo.cfg
    ↪ ../shared/yolo.weights video_path label_name
     darknet.c:
         main(tag)
             if tag == "detector"
                 run_detector() in [detector.c]
     detector.c:
         run_detector(tag)
             if tag == "demo"
                 demo() in [demo.c]
     demo.c:
         demo()
             display_in_thread(current_frame)
                 show_image_cv(frame)
                     - compresses labeled frames into video format

             detect_in_thread(current_frame)
                 draw_detections() in [image.c]
     image.c:
         draw_detections(frame)
             - parses predictions per frame
             - uses predictions to define bounding boxes
             - generates overlays on frame for each bounding box
```