About Me

Name: Sri Siddarth Chakaravarthy P
LinkedIn: https://www.linkedin.com/in/sri-siddarth-chakaravarthy-p-145675192/
University(Education) :

- Vellore Institute of Technology(VIT), Vellore (July 2018- Present)
- Nanyang Technological University (NTU), Singapore (Jan 2022- Present)

E-mail : srisiddarth16541@gmail.com
Time Zone : India (Chennai)(GMT +5:30)
GitHub : https://github.com/Sidd1609
Resume: https://drive.google.com/file/d/1Xe5haCSDcMl4Hvw1NY6zAMLcCihpDN6l/view?usp=sharing

Hi, I am an final year undergrad pursuing a bachelor's degree in Computer Science steered in the direction of machine learning and data science as areas of interest. I am looking to pursue a master's degree in Computer Science in the near future specializing in the area of machine learning. I work on things that interest me and am enthusiastic to learn new things every day. I hope to work on autonomous vehicles sector building ADAS for autonomous vehicles.

Related Experience:

Vidrona-LTD, United Kingdom R&D Project Intern (AI domain) **Mar-Nov 2021**
Worked on the V-sense tool, developed object detection models for detecting Insulators, Transmission Towers, Armor grips, and SAG detection in Transmission Towers. Worked on Object Detection Models- Yolo, SSD, Faster-RCNN. Annotation tools - LabelMe, Instance Segmentation, Anomaly Detection, Pre-processing techniques- Histogram normalization, Thermal image processing, etc.

Samsung-R&D Institute, Bangalore R&D Project Intern **Dec-Present 2021**
Developing a machine learning model to optimize transmit power for 5G and 6G systems using beam data, path loss coefficient and interference management in base stations.

Automotive Research Centre(ARC)-VIT, Vellore Research Assistant **Oct-Present 2021**
Working on sensor fusion algorithms using Intel realsense Camera module, LiDAR sensors, along with path tracking for EGO vehicles in scenarios of high beam light interference from vehicles in opposite directions in the Automotive Research Centre(ARC) in VIT, Vellore.

Skills:

Programming – C/C++, Python
Python – Numpy, Pandas, Tensorflow, OpenCV, Scikit-Learn, SciPy, Keras, Seaborn
Embedded Systems – Arduino, Nvidia Jetson Nano, Raspberry Pi, Turtlebot Waffle pi3
Tools – Docker, Git, LabelMe, MATLAB-Simulink, Postman, Netflix Conductor

//Publications of thesis and internship (Vidrona) under review

Projects (Autonomous Vehicle using Udacity simulator)

- Lane detection (by using projection matrix and distortion coefficients extension of canny edge detector and Hough transform for better results)
- Path planning (using sensor fusion data and sparse matrix of wave points)
- Vehicle Detection (compared Object detection models for better real-time detection results of vehicles)
- Sensor fusion for Light Interference modelling in autonomous vehicles (Kalman Filter)
- Object Detection Models COCO for Transmission Towers (Yolo, Darknet, Faster-RCNN, Detectron)
- RCE Neural Network for Geometric Shape detection

Project of Interest (From Project List): Lightweight object detection models, Realtime object tracking models

Title: Lightweight Object detection and tracking for resource restricted usage

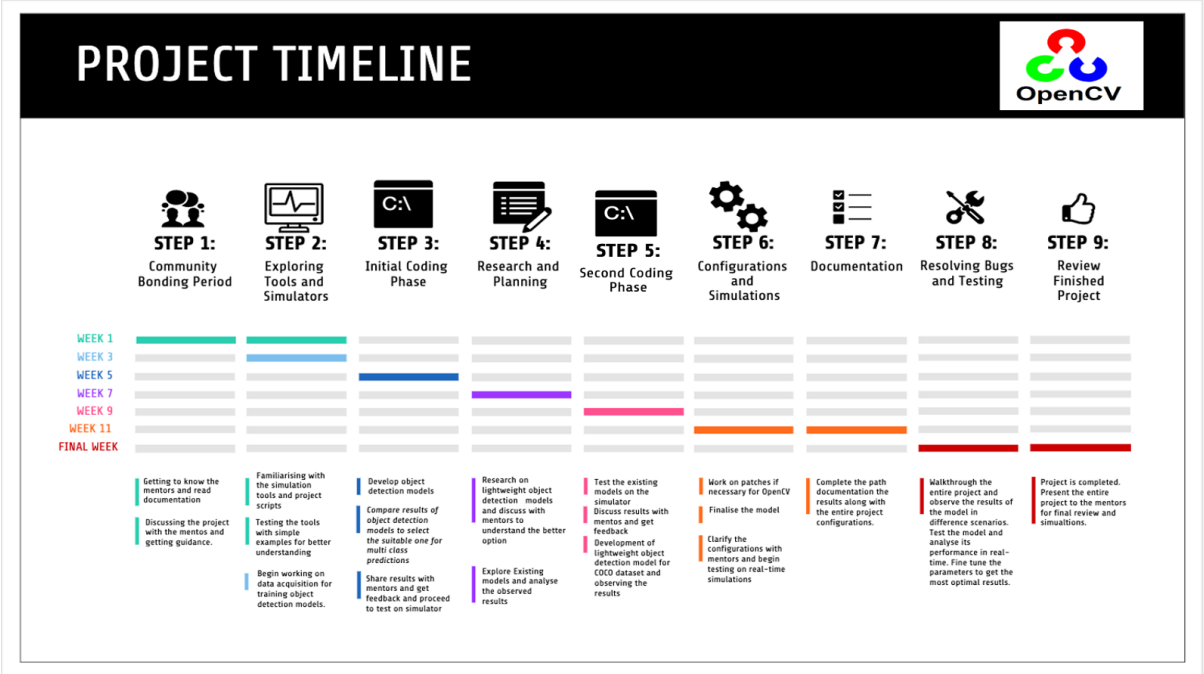
Synopsis (Summary)

Object detection has made great progress in the past few years along with the development of deep learning. However, most current object detection methods are resource hungry, which hinders their wide deployment to many resource restricted usages such as usages on always-on devices, battery-powered low-end devices, etc. This project would aim to propose the use of tiny-DSOD for resource restricted usage. This considers the resource and accuracy trade-off for resource-restricted usages during designing the whole object detection framework. Based on the deeply supervised object detection (DSOD) framework. Tiny-DSOD introduces two innovative and ultra-efficient architecture blocks: depthwise dense block (DDB) based backbone and depthwise feature-pyramid-network (D-FPN) based front-end. We conduct extensive experiments on the COCO benchmark and compare Tiny-DSOD to the state-of-the-art ultra-efficient object detection solutions such as Tiny-YOLO, MobileNet-SSD (v1 & v2), SqueezeDet, Pelee, and Faster-RCNN.

Benefits to Community

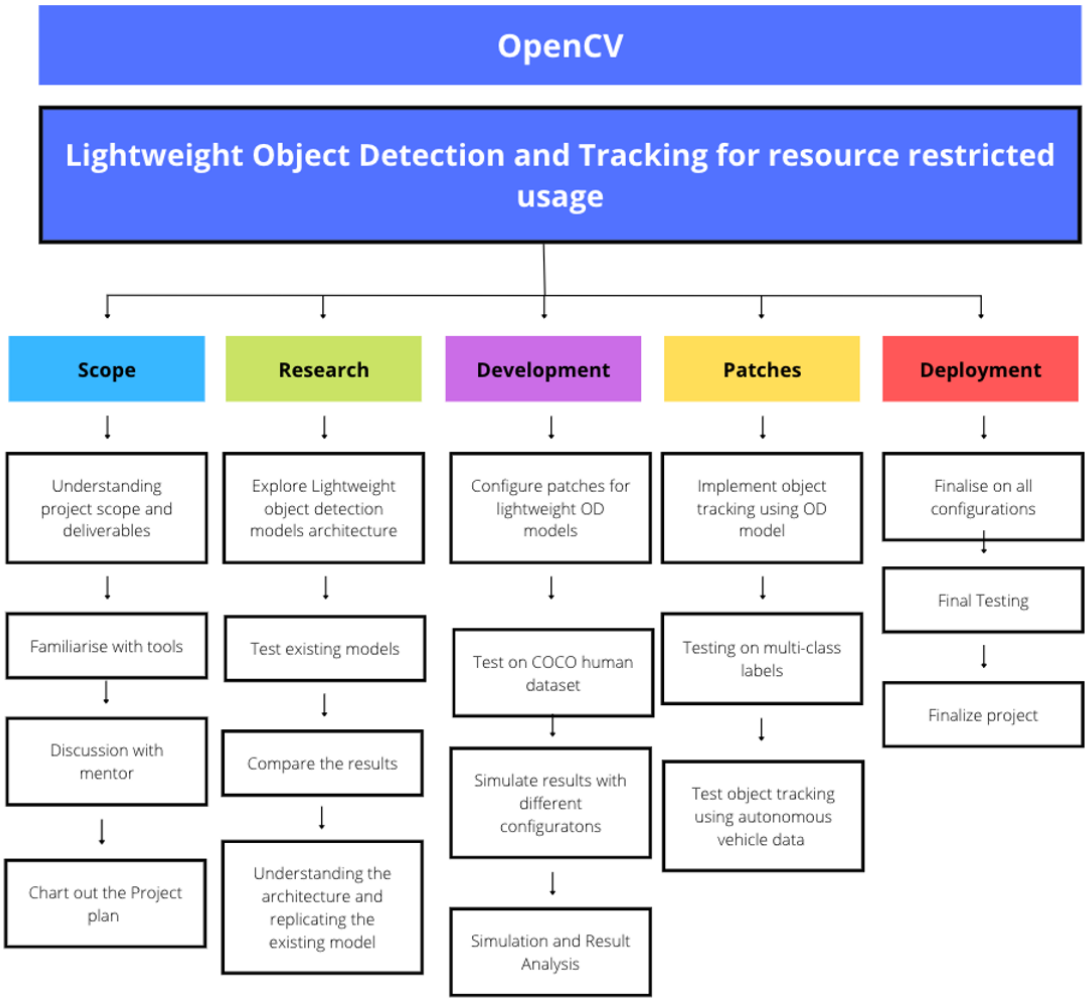
The intrinsic ability of humans to rapidly detect, differentiate and classify objects allows us to make quick decisions in regards to what we see. Several appliances can make use of a fast and lightweight automated object detection for images or videos. Throughout the last 5 years, the technology industry has constantly introduced computational and hardware solutions, such as devices with impressive processing and storage capabilities. However, object detection methods usually require either high processing power or large storage availability, making it hard for resource constrained devices to perform the detection in real-time without a connection to a powerful server. I believe that the projects from your idea lists target the most crucial areas of object detection in real-time considering the usage of these models has been increasing due to the shift to autonomous systems, it is certain that the work carried out by your organization would have a significant impact on the society. Implementation of a lightweight object detection and tracking would enhance ADAS systems and would have a huge impact in the autonomous vehicles sector.

Project Timeline



Deliverables

I plan on focusing my deliverables in three-parts Investigation, Coding and Documentation. I believe that the work we carry out today will be the guiding force to another research, this is how we learn collectively, helping each other. Most of the research do not have a detailed report of the project, hence my fundamental goal is to document all the project details in a simple yet comprehensive report so that it would help others who are interested in extending the work. Codes and testing results would also be uploaded on GitHub from time-to-time with versions to keep track of the changes. A detailed WBS is given below which states how I would tackle this problem.



Project Outcome:

- One or more lightweight object detection models trained (or borrowed if the license is appropriate), quantized if accuracy does not drop too much, and submitted to OpenCV Model Zoo.
- Necessary patches, if any, for OpenCV DNN to support the provided lightweight object detection models.
- Examples in C++ and Python that demonstrate the use of provided models.

-X-X-