DeepVision

Fall Quarter Design Review

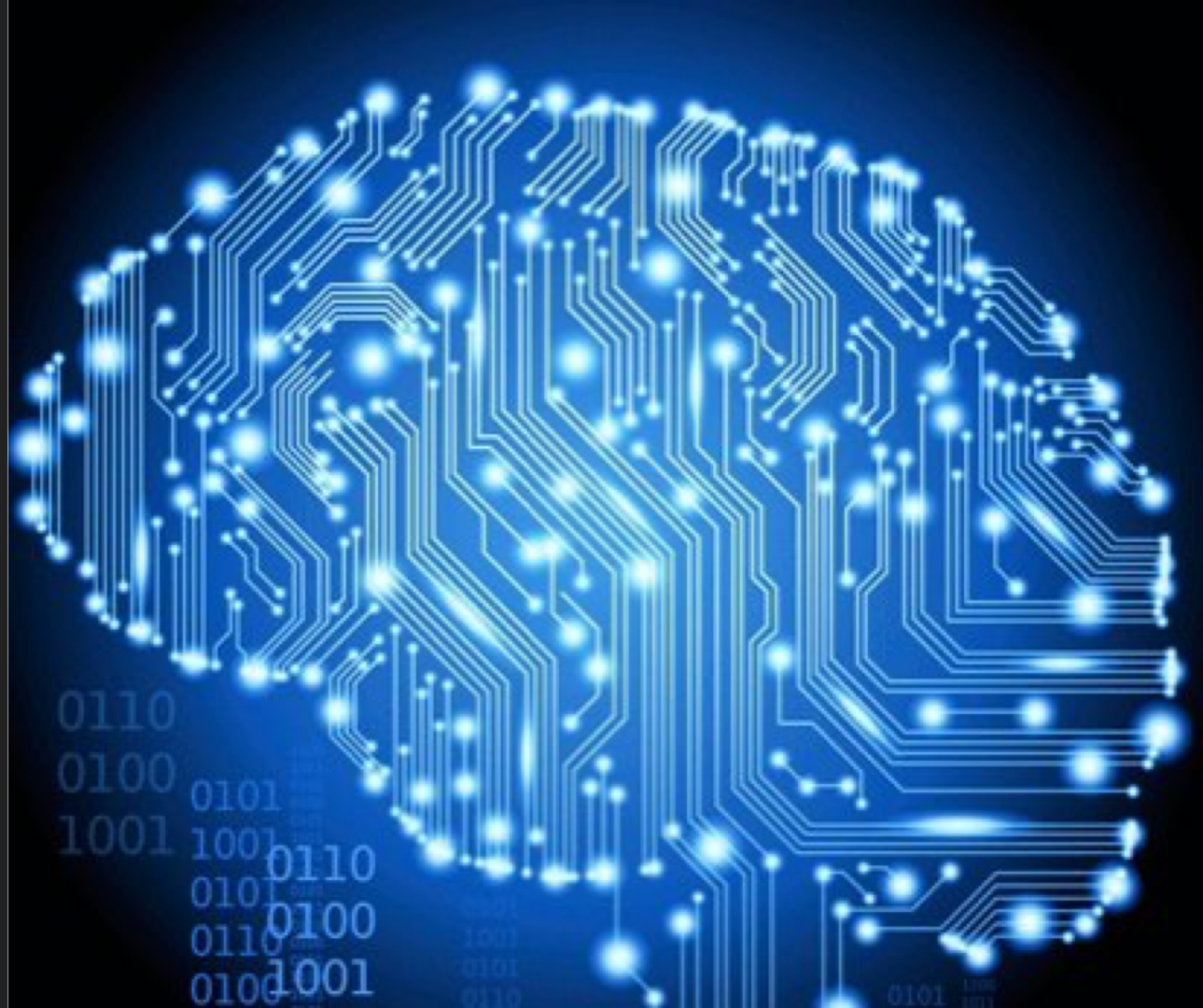
Purpose

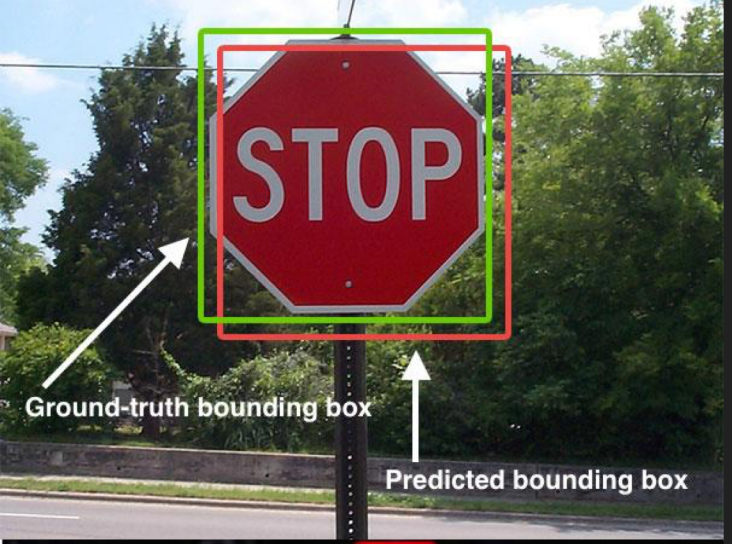
We aim to build a **Deep Neural Network Image Recognition and Detection algorithm** on a embedded GPU device

Attending **2018 DAC Contest**

Sponsored and collaborated with **Prof. Xie** and **SEAL LAB**

Finally implemented on drones as well





Software - Algorithm

- ❏ Detect and tracks people and objects in video captured by drones.
- ❏ Problem with conventional tracking algorithm without deep learning...
- ❏ Deep learning - use training dataset to train computer to recognize people and be able to track them, even in difficult scenarios.
- ❏ We are implementing our design based on the state-of-the-art YOLO algorithm
- ❏ Energy Consumption vs. Throughput

2018 DAC Contest

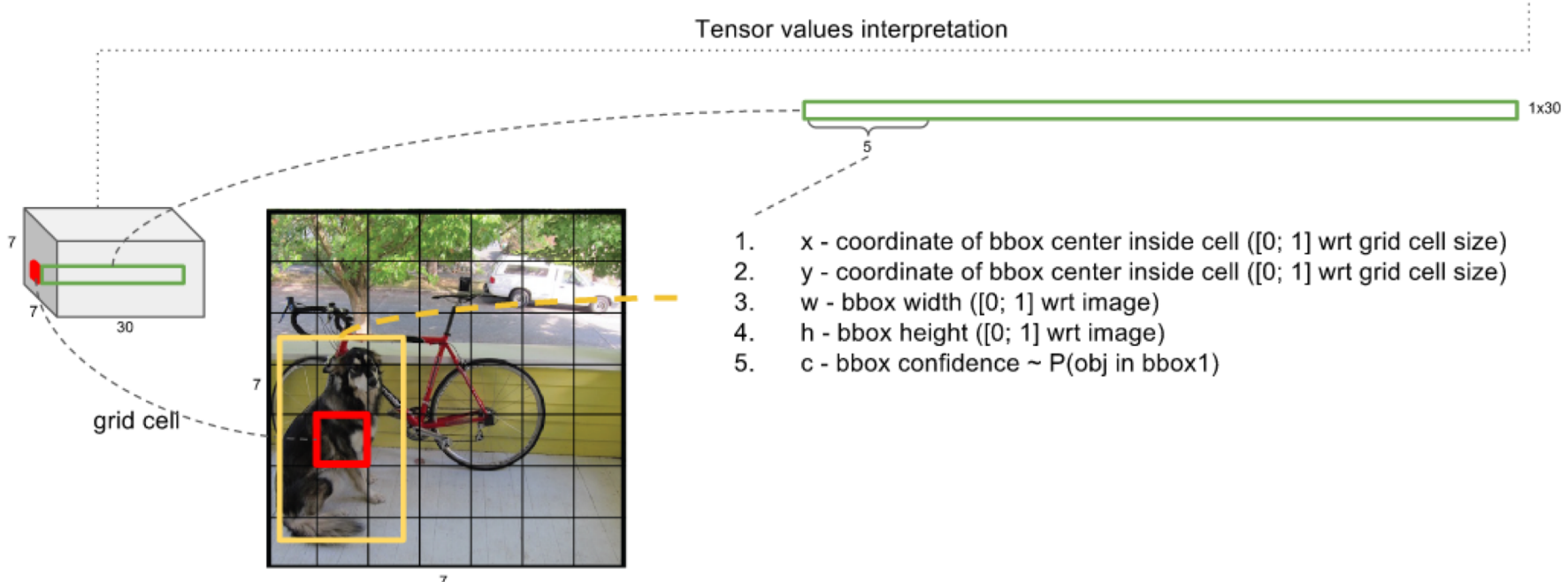
- Features embedded system implementation of neural network based object detection for drones
- Two categories by platform: FPGA and GPU. we registered GPU category, hardware provided by the contest
- Evaluations

Update On Contest

- Received Hardware Jetson TX2
- IO Specification set
- Detection and tracking
- Speculation on focusing on optimization
- First implementation due in January for evaluation

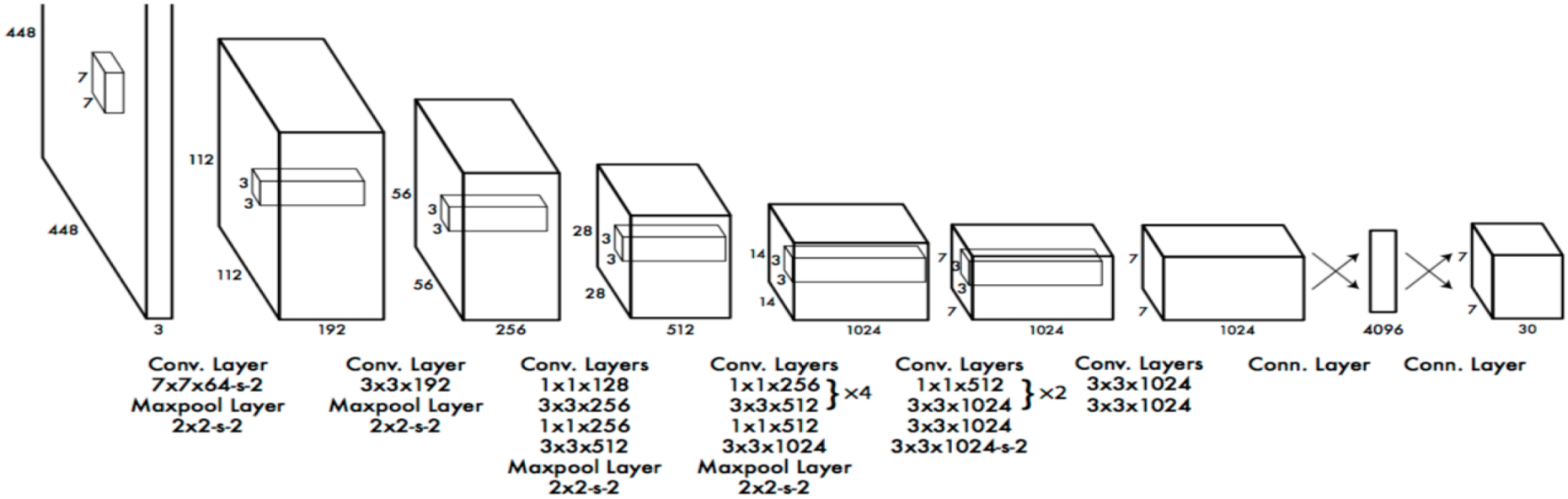
YOLO

- Unified Detection
  - x, y, w, h, c



YOLO

- The Darknet framework
  - Inspired by GoogLeNet

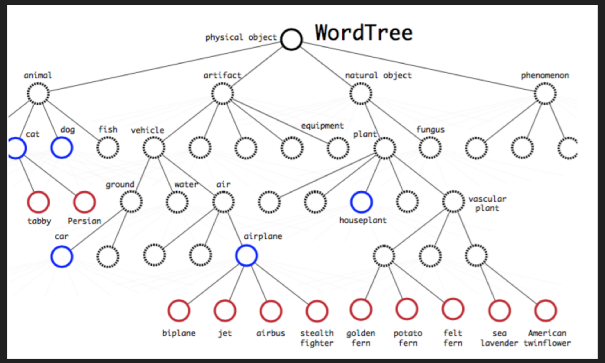


YOLO

- Training Algorithm
  - More layers
  - Higher resolution
  - Optimized loss function
  - Adjusted learning rate
  - Dropout and extensive data augmentation

YOLO to YOLOv2

- Better
  - Direct location prediction
- Faster
  - Darknet-19
- Stronger
  - Hierarchical classification



Improvement

- Implement YOLO on PyTorch instead of in C and Tensorflow.
  - More easy-to-use library package
  - More simple and concise compared to Tensorflow
- Create the model in a more explicit way
  - The open-source code is too complex
- Focus on reducing energy
  - Network compression
- Switch back to C and CUDA at the end
  - Remove the library wrapper
  - For better performance

Hardware - Nvidia Jetson TX2

JETSON TX2 MODULE

- NVIDIA Pascal™ Architecture GPU
- 2 Denver 64-bit CPUs + Quad-Core A57 Complex
- 8 GB L128 bit DDR4 Memory
- 32 GB eMMC 5.1 Flash Storage
- Connectivity to 802.11ac Wi-Fi and Bluetooth-Enabled Devices
- 10/100/1000BASE-T Ethernet

JETSON CAMERA MODULE

- 5 MP Fixed Focus MIPI CSI Camera

BUTTONS

- Power On/Off
- Reset
- Force Recovery
- User-Defined

I/O

- USB 3.0 Type A
- USB 2.0 Micro AB (supports recovery and host mode)
- HDMI
- M.2 Key E
- PCI-E x4
- Gigabit Ethernet
- Full-Size SD
- SATA Data and Power
- GPIOs, I2C, I2S, SPI, CAN*
- TTL UART with flow control
- Display Expansion Header*
- Camera Expansion Header*
  - *I/O expansion headers: refer to product documentation for header specification.

POWER OPTIONS

- External 19V AC Adapter

KIT CONTENTS

- NVIDIA Jetson TX2 Developer Board
- AC Adaptor
- Power Cord
- USB Micro-B to USB A Cable
- USB Micro-B to Female USB A Cable
- Rubber Feet (4)
- Quick Start Guide
- Safety Booklet
- Antennas to Connect to Wi-Fi-Enabled Devices (2)

- Ubuntu 16.04 LTS
- Jetpack 3.0 SDK
  - Deep Learning: TensorRT, cuDNN, NVIDIA DIGITS™ Workflow
  - Computer Vision: NVIDIA VisionWorks, OpenCV
  - GPU Compute: NVIDIA CUDA, CUDA Libraries
  - Multimedia: ISP Support, Camera imaging, Video CODEC

Drone sends back video to be processed on GPU.

Main Difficulty: Interface between the board and drones.

PARROT BEBOP 2

Camera: 14 mega-pixels with fish-eye lens

Video resolution: 1920 x 1080p (30 fps)

Battery life: 25 minutes flying time (with 2700 mAh battery)

GPS: Yes

Processor: Dual core processor with quad-core GPU

Storage: 8 GB flash storage system

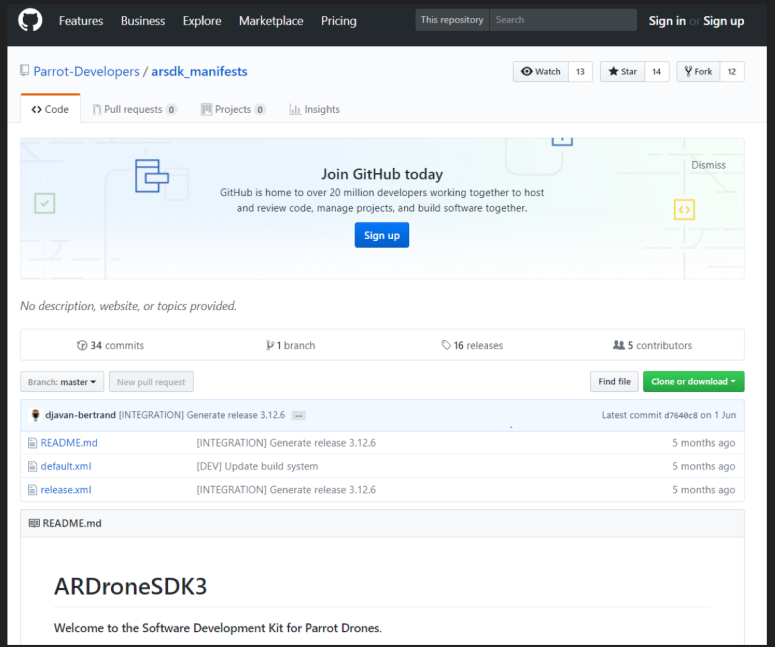
Connectivity: Wi-Fi 802.11a/b/g/n/ac

Signal range: 300 m

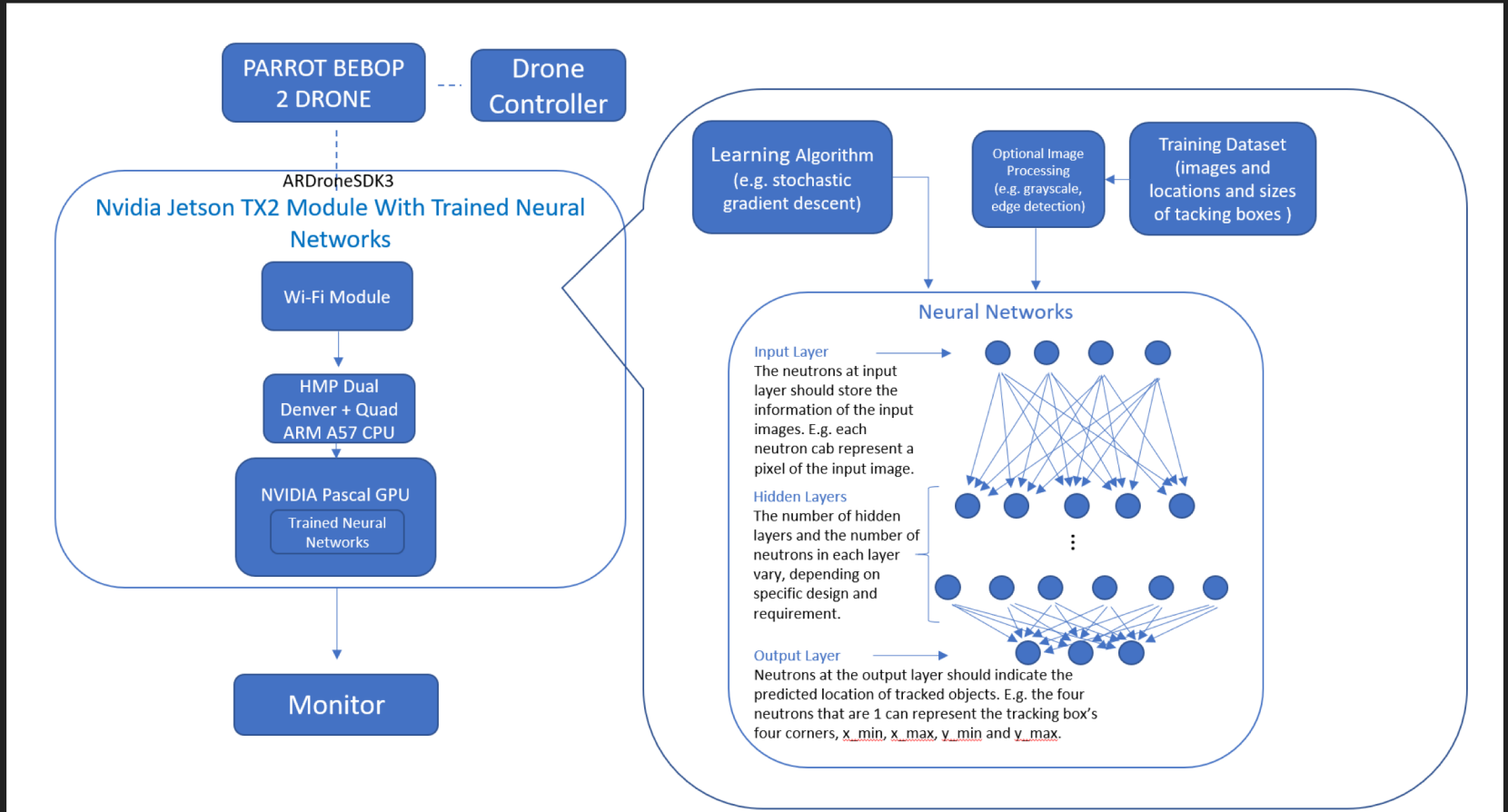
ARDroneSDK3

Platform: IOS, Android, Linux

- Discover the drones on the network
- Connect the drones
- Send piloting and camera commands
- Configure the drones
- Get informations (depends on drones capabilities)
- Get H264 video stream on bebop
- Get MJpeg video stream on Jumping Sumo
- Transfer photos / videos
- Update the drones
- Handle Drone Academy / Mavlink files



Block Diagram



Demo

Using YOLO on pytorch to detect and track objects

Video

Team Responsibility

Charlie Xu

Group leader, Algorithm Design

Terry Xie

Software Framework, Hardware Interface Design

Jenny Zeng

Deep Learning Algorithm Design

Chenghao Jiang

Hardware and Software Interface Design

Project Report

We have a working algorithm running on TX2, implementation of YOLO on Pytorch, able to train and get the right result.

14fps and 76 mAP

Future Plans

Optimize YOLO - Improve fps and accuracy, limit power consumption

Update each month

See how we do on the contest

Final Goal: After we have our version of the algorithm, implement it on drones.

Questions?

Collaborators / Mentors / Sponsor

Prof. Yuan Xie (UCSB SEAL LAB)

Xing Hu (UCSB SEAL LAB)

Liu Liu (UCSB SEAL LAB)

Prof. Yogananda Isukapalli (ECE 189)

NVIDIA (SPONSOR)

DJI (SPONSOR)