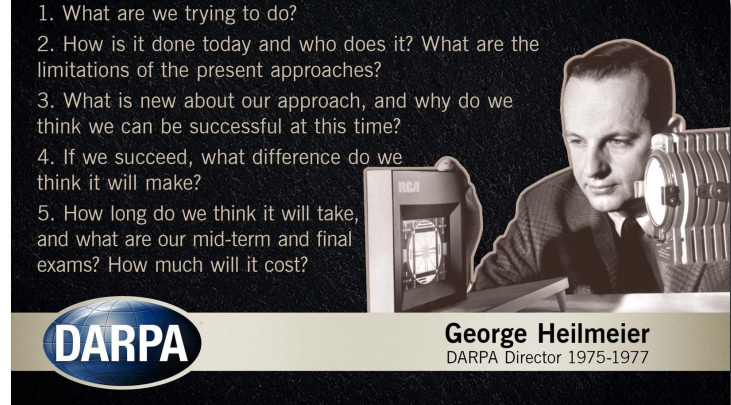


Team C2 – Sign Sight 260

Justin Bell, Sean Johnson, Paul Wright



Add your 12 slides after this slide... [remember, 12 min talk + 3 min Q/A]

For more information about formatting or importing slides see:

<https://gsuite.google.com/learning-center/products/slides/get-started/>

Make sure to cover

(refer to the Proposal Presentation Guidance):

- Use Case
 - Who is your customer, what product do they want to buy?
- Use-Case Requirements
 - What do they expect your product to do? [Answer this Quantitatively]
- Technical Challenges
 - What do you need to technically achieve so your product can meet these requirements?
- Solution Approach
 - Show us step by step what your proposed solution path(s) looks like – incorporate flow diagrams
- Testing, Verification and Metrics
 - Detail your testing & verification plan (and how you will measure & verify the results) of your product
- Tasks and Division of Labor
 - Technical Task List to build your proposed system, WHO is going to do WHAT in your team?
- Schedule
 - Gantt Chart (task timeline, demo system/components, MVP, stretch goal, include slack time)

Consider that this slide already works as a introduction slide so use your first slide wisely

Remember – Relevant figures (and tables) can be worth
“a thousand words”!

Use Case - Why Sign Sight 260?

- Autonomous sidewalk delivery vehicles must safely navigate shared pedestrian space.
- Current low cost platforms (Raspberry Pi, etc.) rely on slow and power hungry vision pipelines.
- Delayed perception leads to unsafe stopping, missed movements, and collisions.
- Our focus on perception latency over perfect classification accuracy.
- FPGAs are less power intensive than GPUs and more powerful than microcontrollers



Use Case (Customer and Product) - Last Mile Delivery

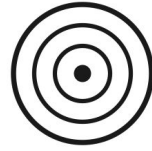
Customer: Companies developing small autonomous sidewalk delivery robots. Delivering small packages to a person where it would not be practical or safe to use a person or large vehicle.

We are trying to improve upon existing solutions through better navigation around populated areas and being less of a nuisance. Vehicle follows predefined sidewalk route while encountering and reacting in real time to obstacles, pedestrians, signs, and crosswalks.

ECE Areas of Study Covered

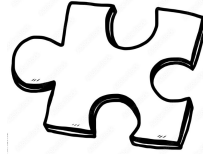
- Hardware systems: Logic design of computer vision
- Software systems: Pre/post processing of frames, decision-making. Robotics and communication between different parts.

Use Case - Requirements



- Classify objects with 75 percent accuracy in proper lighting within 2 meters.
- End to end latency of image capture to decision making under 50 ms.
- Safely stop within .5 meters of detecting obstacle within 2 meters
- Power consumption of FPGA 25 W
- Frames processed per second at least 24
- Perform 360 degree turns within a 915mm x 915mm area
- Move along straight paths within a 610mm horizontal space for at least 5 meters
- Operate continuously for > 30 minutes on battery power

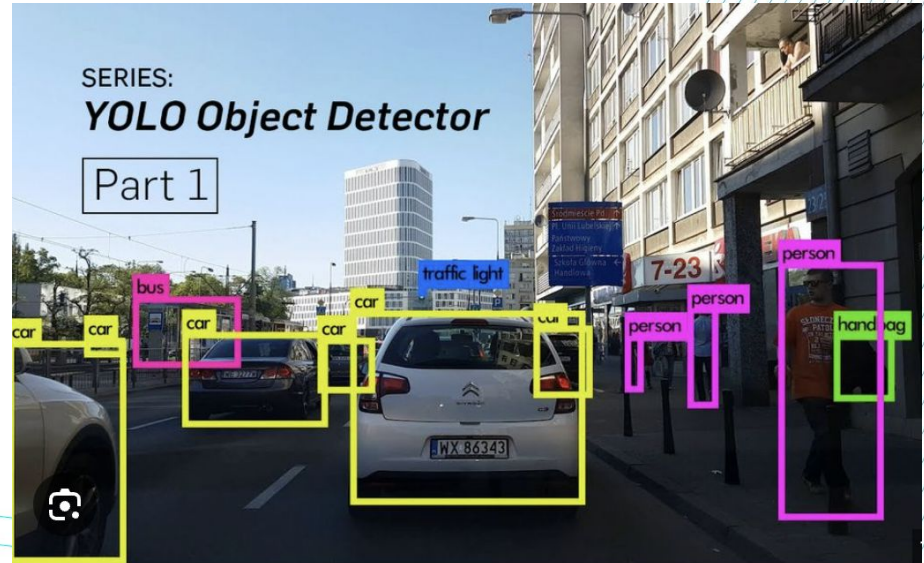
Key Technical challenges



- Low latency image processing on limited power
- Accurate perception under changing light conditions
- Integration between perception and control
- Speed of arm SOC operation for robotics controls and preprocessing
- Consistency of bot movement
- Communication of measurements (distance, position) between different components and SOC.
- Training for a model specifically for a FPGA
- Robotics with the Kria Board

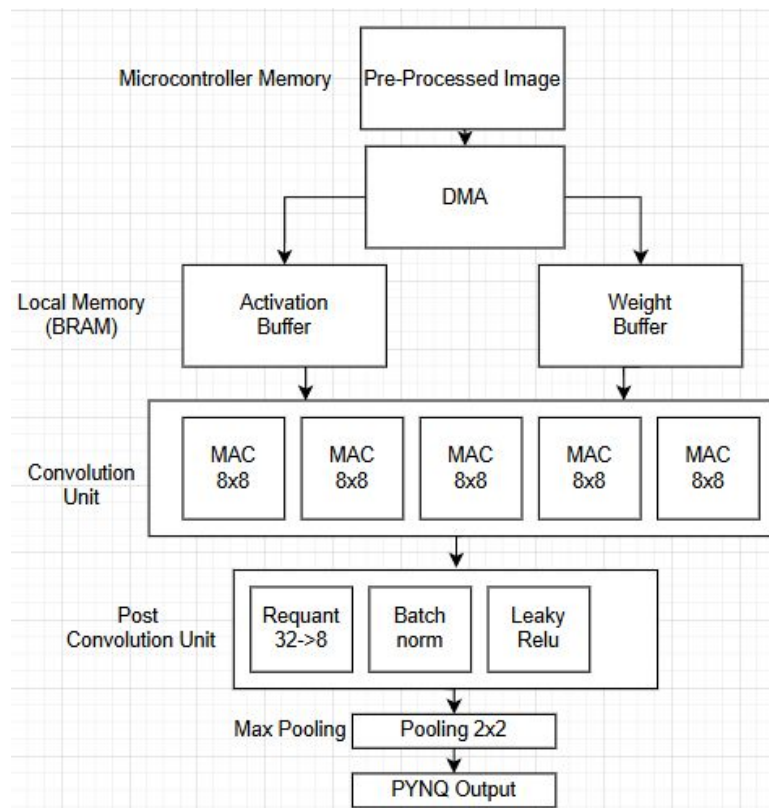
Solution Approach: Computer vision

- Train YoloV3-Tiny on an INT8 data set containing people, signs, and crosswalks, and obstacles with pytorch
- Quantize to INT8
- Extract Weights for RTL
- Deploy on FPGA
- Receive output
- Then Camera uses bounding box to approximate distance



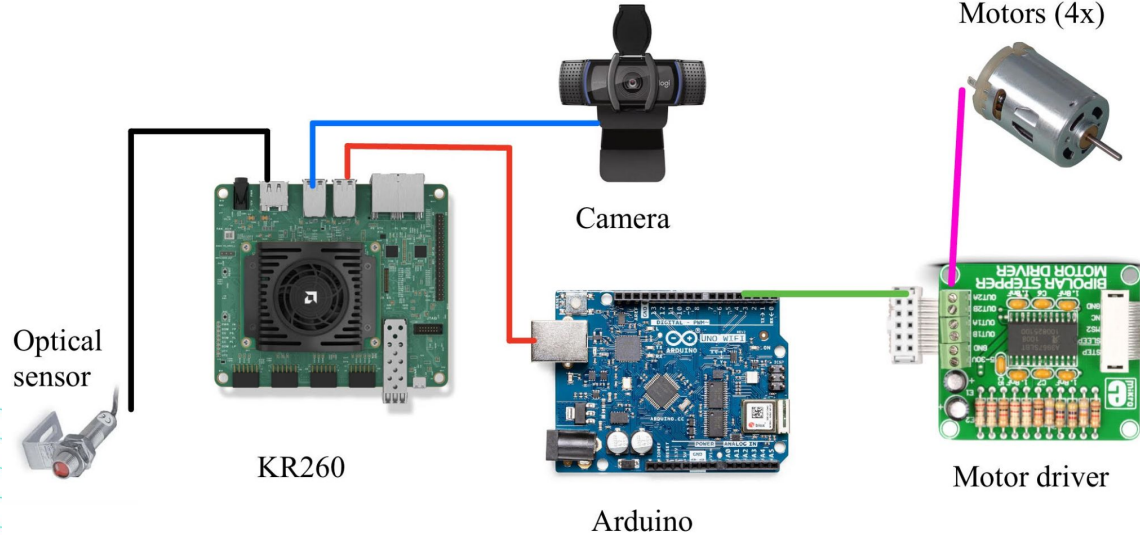
Solution approach: RTL

- Implement YoloV3-Tiny
 - Define layer architecture (conv, pool, activation)
 - Design streaming datapath with AXI4-Stream
 - Implement BRAM-based weight/activation storage
 - Build control FSM for layer sequencing
 - Integrate modules into top-level accelerator
 - Simulate & verify in Vivado
 - Synthesize & deploy on Kria KR26



Solution approach: Robotics and Hardware

- Frame processing and decision-making performed by FPGA and ARM SOC on KR260
- Bounding box of image used to approximate distance
- Arduino for motor controls
- Optical sensor for position



Testing, Verification, and Metrics

Reaction Latency Measurement

- Timestamp: An object enters the cameras view and a hardware signal marks the exact moment it is captured.
- Process and Measure: The system processes the input normally and sends a motor command, at which point a second hardware signal will assert. Logic analyser measures time between signals.

Perception Performance

- Test using a controlled course with known visual cues
- Measure correct detection rate and false detection rates.

Testing, Verification, and Metrics

Safety validation

- Measure stopping distance following obstacle or crosswalk detection
- Verify stopping behavior meets use case safety constraints of .5 meters when object is within 2 meters

Power Evaluation:

- Measure s  runtime under continuous operation

Verification Strategy

- All tests dire  map to quantitative use case requirements

Tasks and Division of Labor

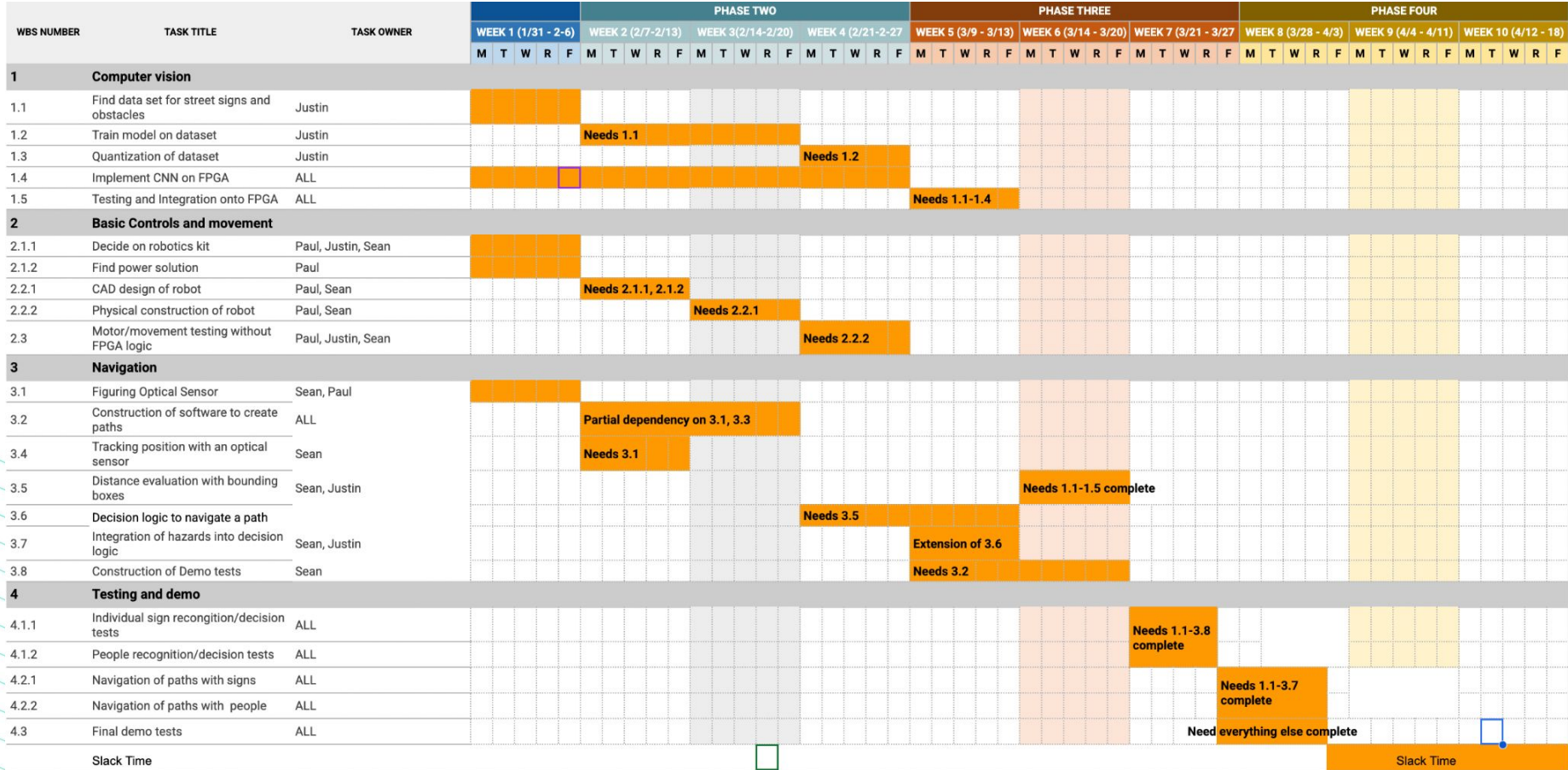
Paul: Contributions to CNN, computer vision testing, physical implementation and logic-free movement.

Justin: ALL of computer vision, majority of CNN, parts decisions, distance evaluation,

Sean: Contributions to CNN, computer vision testing, movement logic and infrastructure for creating paths

ALL: Evaluation and testing, final demo

Schedule Gantt Chart



Conclusion

The Challenge: Sidewalk delivery robots need fast, power efficient perception to safely navigate shared pedestrian spaces.

Our Solution: Sign Sight 260 uses FPGA hardware acceleration to achieve:

- Sub 50ms latency from perception to action
- 25W power budget
- Real time object detection at 20+ FPS with YOLOv3-Tiny on the Kria KR260

Why it Matters: We will deliver on both speed and power allowing for safer more responsive autonomous delivery in complex urban environments.

Next Steps: RTL implementation, model training, robotic controls