ECE532 Project Proposal
Team 7
Robert Kolaja
Albert Le
Tony Wang

Project Title
Misplaced/Suspicious Package Detection and Alert System (MSPDAS)

Introduction

Safety is a major concern for public events, venues and transit. Our project goal is to develop an embedded system that will monitor public spaces in real time for suspicious packages. The embedded system will be connected to a central server via the network and issue a warning to the central server with surveillance if necessary. The server will be developed by us as well. This project is interesting as it involves developing a system that would be able to self-identify suspicious packages without human assistance. We like to undertake this project as it is technically challenging and fulfills a real-life need.

Many surveillance cameras out there today capture video but do not identify suspicious packages nor issue alerts. To identify suspicious packages, public services such as GO Transit and TTC rely on public announcements encouraging individuals to report suspicious packages directly to an employee. We believe that the current method is not ideal nor efficient. There are projects out there today that implement facial recognition and tracking in the public domain. This methodology provides complete monitoring of public spaces, however, it is also very intrusive. We aim to develop an embedded system with the sole purpose of identifying suspicious packages without incorporating the intrusive and often controversial aspects of tracking.

Project Team
This section includes short biographies on each of the team's members. Following that, a discussion on the team's strengths and weaknesses is presented.

Biography: Robert Kolaja

Robert Kolaja is a fourth-year computer engineering student focusing on computer hardware and software design. He has completed numerous academic and personal projects in these areas, all of which have presented unique learning opportunities. Most recently during his internship at Qualcomm Canada Inc., Robert developed many new scripts, designed a hardware bus traffic tracking system, implemented a memory redundancy design in SystemVerilog, and performed verification tasks for next-generation Snapdragon™ products. Some of his previous academic projects include an ultrasound-based word spelling game implemented entirely in Verilog and a remote-controlled – via the LAN – rover with camera streaming built around a Raspberry Pi. In addition to these academic projects, Robert has developed multiple websites still in use today and participated in a hackathon where his team successfully constructed an IoT personal task tracking and display device.

With respect to courses relevant to this project, Robert has studied the following subjects: Algorithms & Data Structures, Computer Hardware, Operating Systems, and Computer Networks.

Through all these experiences, Robert has gained a great deal of knowledge about the boundary between hardware and software and has become comfortable switching between thinking in terms of hardware and software design implementations.

Biography: Albert Le

Albert Le is a 4th year computer engineering student with a focus on computer hardware, signal processing and communications. Having completed an internship at Synopsys where he was a member of the DesignWare SERDES PHY Verification Team, he brings his experience in debugging and testing large, complex designs. In addition, he has experience working with SystemVerilog testbenches and creating tests from design specifications.

Academically, he has taken courses in computer networking, computer architecture, and digital signal processing. In his extracurricular endeavours, he has worked with the Raspberry Pi and Arduino and written low-level C programs to interface with digital and analog sensors. He has also written small programs with OpenCV and Python Imaging Library (PIL) to do basic image processing. All of these experiences should provide the theoretical and practical knowledge to aid the development of the system design, image processing algorithms and implementation.

Biography: Tony Wang

Tony Wang is a 4th year computer engineering student with a focus on computer hardware and software. He brings his debugging and scripting experience from working on the Radeon Technology Group Post-Silicon Verification team at AMD. In addition, he bring his knowhow in developing and executing test plans for feature verification and in tools development to aid validation efforts.

For his studies, Tony has completed courses in operating system, algorithms and computer architecture. Through his capstone project, he has experience working with numpy, PIL and scipy Python libraries to perform DCT compression and decompression of video frames. For one of his past projects, he developed a bubble shooter game entirely in hardware through Verilog. From his background in software and hardware, he should be able to aid the efforts in integrating hardware with software to develop a complete embedded system.

Discussion of Strengths & Weaknesses

In order for the project to be successful, there is a combination of theoretical knowledge and practical experience required. Theoretical knowledge includes computer networking and image processing, while useful practical experience would include digital design practices, and designing and implementing DSP algorithms in both hardware and software. Table 1 below lists the various strengths and weaknesses of each team member. At least one member has taken courses or completed personal projects involving networking and image processing, which will help inform and guide our design decisions. With regards to practical experience, all three members have completed internships where they worked on various areas of the ASIC development flow including pre-silicon design and verification, and post-silicon verification.

In terms of weaknesses, the strengths of other team members complement and compensate the weaknesses of any particular member. The team has at least one member with previous digital design, DSP and verification, and networking and video processing, which will make it easier to divide up the milestones according to our strengths.

Still, one area where we lack all experience is in the problem of object detection algorithms specifically. This will require all three members to conduct research to pick an existing algorithm and curate it to our needs. Testing image processing algorithms on hardware and in real-time is also a skill that the team will need to acquire as we work through the project, but with the number of other groups in the class doing projects with video, we are confident that this is doable within the timeframe.

Table 1: Summary of Team Member Strengths & Weaknesses

Team Member	**Strengths**	**Weaknesses**
Robert	- RTL design and implementation - Creative hardware problem solving - Developing server-side APIs and web applications - Comfortable with C/C++ programming - Version control (via Git) and project management	- Limited experience writing/working with testbenches - No experience in image/video processing
Albert	- RTL digital verification, writing testbenches and debugging	- Limited knowledge of operating systems

	- Knowledge of digital signal processing, basic image processing - Experience with using MATLAB for developing signal processing algorithms and implementing them on DSPs (Digital Signal Processors)	- C/C++ programming out of practice
Tony	- System level debugging, development in C/C++ and scripting. - Knowledge of computer networks working with client and server systems. - Experience working with video processing in Python.	- Unfamiliar with writing testbenches in SystemVerilog. - Limited knowledge of digital signal processing.

Project Description

Our project is to develop an embedded system to detect suspicious packages in public spaces in real time without human assistance. If a suspicious package is detected, an alert and supporting video data will be sent to a remote server. The server will be developed by the team as a deliverable. A suspicious package is defined as an item that remains in the same spot for a prolonged period of time; we define this time at least 5 minutes (programabe parameter, will vary with where system is deployed). Supporting video data is defined as select video frames prior to the placement of the suspicious package and after placement of the suspicious package.

Functional Requirements

- Must be able identify suspicious packages or objects in the space where the system is employed. At minimum, this space will be limited to the area visible in the camera's field of view (camera is not moving)
- Must save the first frame and time which the suspicious package first appeared in
- Must send an alert to a server when a suspicious package/misplaced item is detected, along with relevant information such as the timestamp, GPS location, and supporting video data as described in Project Description.
- Must be able to send data to and receive data from a remote server
- Be able to send video frames before and after the package was first detected to a server for review by a system operator
- The remote server is able to display/playback the received video data.

Future Enhancements

- Be able to identify potential candidates for who put the package there or how it got in frame

Acceptance Criteria

1. The embedded system is able to detect a suspicious package of dimension 56 x 36 x 23 cm that is completely within camera view and stationary for over 60s. (56 x 36 x 23 cm is roughly the maximum size of a carry on bag)
2. Upon detecting a suspicious package, the embedded system is able to send an alert message, supporting video data and location (GPS data) to the remote server.
3. The remote server is able to accept alert messages, video data and GPS data.
4. The remote server is able to display/playback the received video data.

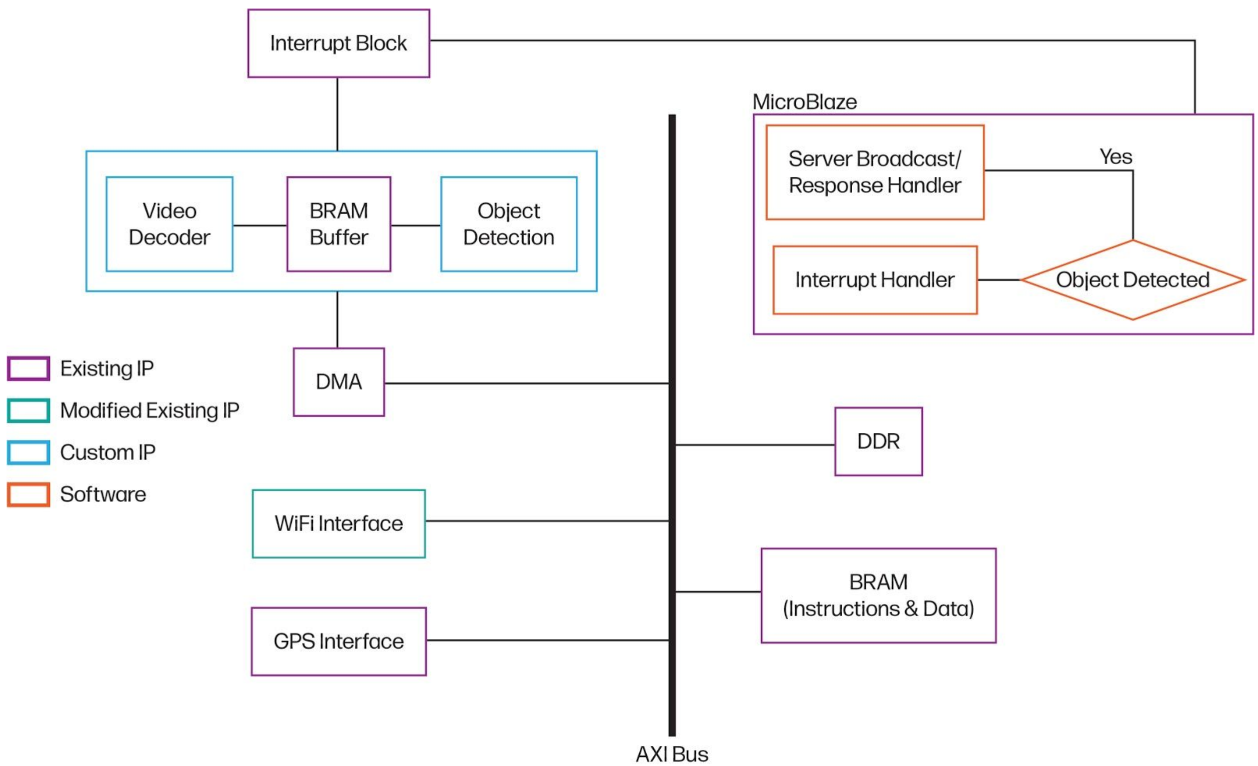

Figure 1: System Block Diagram

Table 2: Block Descriptions

Block Name	Description
Object Detection	Analyze the frames streamed in from the video decoder to determine if a package is present in the scene.
Video Decoder	Take raw image data from the camera, decode it, and buffer it in BRAM for the Object Detection block.
WiFi Interface	Interact with the WiFi PMOD
GPS Interface	Interact with the GPS PMOD
DDR	Store frames when buffering output for a long period of time (needed to meet Milestone 5)
Server Broadcast/Response Handler	This block handles interacting with the server. It sends alerts whenever our design detects a package and accepts any input sent from the server to the device..
DMA	Direct Memory Access block to handle writing camera frames without the need of the MicroBlaze. This is needed to reduce unnecessary bus traffic.
Server (Not Shown)	This will accept input from all of the instances of this device that a system administrator has deployed on their property. When an alert is sent from any one of the devices, the administrator is notified and provided with the relevant details about the event.

Testing

Milestone #1:

1. GPS
   - Be able to display GPS data in the SDK terminal.
2. WiFi
   - Be able to send messages from the FPGA to the server via WiFi.
3. Server
   - Be able to receive and print on the terminal messages from the FPGA via WiFi.

Milestone #2:

1. GPS
   - Be able to send GPS data to the server via WiFi.
2. Camera
   - Be able to write camera image data into buffer.
3. Package detection block interrupts
   - Be able to send interrupt MicroBlaze after buffer is written.

Milestone #3:

1. Edge detection algorithm
   - Be able to detect the existence of a completely visible arbitrary sized rectangular shaped box in an empty room.
   - Be able to send alert message to server after detecting box.

Milestone #4:

1. Optimize edge detection algorithm
   - Be able to detect the existence of a completely visible 56 x 36 x 23 cm duffle bag in an occupied room with people moving.

Milestone #5:

1. Buffer frames
   - Be able to selectively buffer frames before and after the package was detected.
2. Send frames
   - Be able to send buffered frames to the server.
3. Server
   - Be able to playback/display the buffer frames on the server.

Milestone #6:

1. Fine tuning and testing complete system
   - Find minimum detectable package size.
   - Increase the number of subjects in the room and evaluate package detection rate.
   - Change the intensity of the lighting in the room to evaluate package detection rate.

Project Complexity
Table 3 shows how our project's complexity score was calculated. The numbers used are from the summary chart on Piazza.

Table 3: Project Complexity Calculation

Component	**Complexity Score**
Microblaze†	1
Custom Hardware IP†	1
Network Interface†	1
Camera Kit	2
MicroBlaze Interrupts	1
GPS Receiver	2
TOTAL*	**8**

* Recommended minimum complexity score for a three-member team is 6.
† Required

Risks
This section outlines potential project risks and how they would be mitigated. The risks are in *italics*, followed by their corresponding mitigation option(s).

Unable to successfully detect package in crowded/busy environments

- Constrain environmental conditions (e.g. lighting, number of other similar-looking packages)
- Constrain the types of packages that can be detected (e.g. package colour, package shape/outline)

Package detection algorithm is too complex to implement entirely in hardware

- Select an algorithm with lower complexity
- Implement only some sections of the algorithm (i.e. individual functions) in hardware. The rest will be implemented in firmware on the Microblaze processor

Resource Requirements

Below is a list of the hardware resources required for the project:
- Nexys 4 DDR Board (or Nexys Video Board)
- PMOD Video Kit with Ribbon cable and Breakout Board
- PMOD 802.11g WiFi Interface
- PMOD GPS Receiver
- Laptop capable of running the server over WiFi

Milestones
Table 4 outlines the project's preliminary milestones.

Table 4: Preliminary Milestones

Milestone Number	**Description**
Milestone #1	Successfully interface with the PMOD modules (i.e. GPS and WiFi) to obtain some initial readings/data.
Milestone #2	Microblaze is able to obtain image data from the camera (i.e. the package detection block interrupts the MicroBlaze after some rudimentary processing task is complete) and send GPS data to the server over Wi-Fi.
Milestone #3	The beginnings of edge/object detection are working and the object detection block interrupts the MicroBlaze which then sends a simple alert to the remote server.
Milestone #4	*Mid-Project Demo* Object detection accuracy has reached acceptable levels (in accordance with the Acceptance Criteria).
Milestone #5	System successfully buffers frames before and after the object was/is detected to allow for detection of the person who left the item behind. These frames are then sent to the server for review by a system administrator.
Milestone #6	Perform additional testing to ensure the design works in different environments (e.g. different lighting conditions, people/object densities, amount of motion in the scene)
Milestone #7	*Final Demo* The project is complete, including any mitigation options that had to be exercised.