Computer Vision Testing

The CV was not tested extensively before this iteration ended. The Pi Cam was able to detect pedestrians in a lab setting, but that's it. For the next iteration, the tests described below will be performed.

To test the software algorithm that determines what walks by a node, the nodes can be set up in a controlled environment, such as inside of a lab. A person can then walk by with or without a bike, and the node should be able to identify if the person is walking or riding a bike. The subject should walk past the node at various speeds, and the node should still be able to make a capture. The node should be able to pass captured information to the rest of the nodes, and to the gateway so it can be recorded by the database.

Various lighting scenarios should be tested as well. In a lab environment, the lighting should be adjusted so it gets dimmer until the node is unable to make accurate readings. This will be the minimum light needed for operation, and the node might not require frames to be captured at times like this in the field. During the lab test, a light should be positioned at various angles, such as behind the node shining on the target, behind the target shining on the node, overhead, and to the left and right of the node. Regardless of where the light is positioned, the node should still operate accurately.

When the node is built, and the software written, a stress test that can be performed is to deploy the nodes on the RIT quarter mile, which gets a lot of foot traffic. This test will also show just how accurate and fast a node is. If the node just cannot keep up with the traffic, some redesigning might need to occur.

Table 7 - Computer Vision Test Descriptors

Test ID	Phase	Description	Pass Condition
V.1	Early - Image Sensor	Verify that the image sensor is correctly capturing images	Images in front of the lens are rendered correctly in

	Component	by connecting the Pixy Cam via USB, loading the GUI and observing.	the Pixy Cam GUI window. It can be assumed that the M0 is functioning correctly.
V.2	Early - Image Sensor Component	Verify that the image sensor is correctly identifying colors by running the included color identification algorithms with the board connected via USB	The chosen color is amplified in processed images, as shown in the Pixy Cam GUI window. It can be assumed that the M0 is functioning correctly.
V.3	Intermediate - Infrared Component	Connect the board via USB and connect an Infrared sensor to the camera via GPIO or UART. Dim the room of all light sources. Verify that the sensor data are correctly displayed in the Pixy Cam GUI window by walking in front of the camera and observing..	The image should be correctly represented with more orange to red colors showing higher temperatures. The background should contain blue and purple colors for lower temperatures.
V.4	Intermediate - CV Algorithm Component	Compile and load the CV software onto the Pixy Cam. Walk by the Pixy Cam to verify whether the algorithm is functioning properly. *(To be tested with and without Infrared sensor connected.)*	The image should be identified, and the correct CV size capture, speed and direction determinations are made.
V.5	Intermediate - CV Algorithm Component	Vary speeds at which the subject walks by the Pixy Cam to verify that the algorithm still functions properly. *(To be tested with and without Infrared sensor connected.)*	The image should be identified and the correct CV size capture, speed and direction determinations are made.
V.6	Advanced - CV Algorithm Component	Walk by the Pixy Cam with a bike or other large object to verify that the algorithm still	The CV algorithm is able to correctly discern between a walking and

		functions properly. *(To be tested with and without Infrared sensor connected.)*	biking commuter based on speed of the individual, and the correct output displays in the GUI.
V.7	Advanced - CV Algorithm and Infrared Component Integration	Compile and load the CV software onto the Pixy Cam. Dim the lights in the room. Walk by the Pixy Cam to verify whether the CV algorithm is functioning properly. *(To be tested after CV and IR component tests pass.)*	The algorithm should be able to use the Infrared data with the lights dimmed to accurately detect size, direction and speed of an individual.
V.8	Advanced - Computer Vision and Networking System Level Integration	Compile and load the CV software onto the Pixy Cam. Connect the cam via USB. Connect the CMUCam to the XBee as well via GPIO or UART header and provide power to it via pre-charged Li-Ion battery connected to custom voltage regulation unit. Dim the lights in the room. Walk by the Pixy Cam to verify whether the CV algorithm is functioning properly. Turn on the lights and perform the same tests. *(To be tested after CV, IR, voltage regulation and GPIO component tests pass. The website GUI should also be operational, but data can alternatively be viewed through local GUI.)*	The algorithm should be able to use the Infrared data with the lights dimmed to accurately detect size, direction and speed of an individual. The algorithm should also work with lights on. These data should be broadcasted to a central node, server, and pushed to a website to view for confirmation.
V.9	Advanced - Computer Vision and Power Management System Level Integration	Compile and load the CV software onto the Pixy Cam. Connect the CMUCam to the secondary boost converter. *(Warning: Do not perform this step until component level testing of the*	The algorithm should be able to use the Infrared data with the lights dimmed to accurately detect size, direction and speed of an individual. The algorithm should also

		secondary boost converter is complete.) Connect the CMUCam to the XBee as well via GPIO or UART header and provide power to it via pre-charged Li-Ion battery connected to custom voltage regulation unit. Leave out everything prior to the battery in the Signal Conditioning circuit. Walk by the Pixy Cam to verify whether the CV algorithm is functioning properly. Turn on the lights and perform the same tests.	work with lights on. This data should be broadcasted to a central node, server, and pushed to a website to view for confirmation.
V.10	Advanced - Computer Vision Acceptance Testing	Compile and load the CV software onto the Pixy Cam. Connect the CMUCam to the secondary boost converter. *(Warning: Do not perform this step until component level testing of the secondary boost converter is complete.)* Connect the CMUCam to the XBee as well via GPIO or UART header and provide power to it via pre-charged Li-Ion battery connected to custom voltage regulation unit and add signal acquisition and conditioning circuit including primary boost converter, rectifier, and windbelt. Walk by the Pixy Cam to verify whether the CV algorithm is functioning properly. Turn on the lights	The algorithm should be able to use the Infrared data with the lights dimmed to accurately detect size, direction and speed of an individual. The algorithm should also work with lights on. This data should be broadcasted to a central node, server, and pushed to a website to view for confirmation. Extended testing should be performed at this level once all system level and integration is complete. The node should be able to process CV algorithms for an extended period with sustainable energy available.

		and perform the same tests. Vary size of CV subjects. *(Perform this test after System level integration is complete for each system involved.)*	