

Report

A: System Design

From a high-level perspective, the design of the system includes a photocell sensor, and temperature sensor that collect data, that data is then transmitted via a Bluetooth controller module to the CoolTerm application. The data is then used to calculate how to build a machine learning model to accurately predict the on-off screen times of the device. Some of the challenges I faced while setting up this system were configuring the ports and ADC's to accept analog inputs and process everything with interrupts. The initial sampling frequency used to capture data was set at 125k for the ADC's, and the interrupt rate was set to 62.5 ns assuming a 16 MHz clock for the period? I was able to successfully reprogram the individual components to report during each experiment which also took some re-configuration time.

B: Experiment

In order for me to conduct the proper experiments for this project I chose three temperature sensitive test environments (bedroom, living room, outside(nighttime), two different sources of ambient light in each room(On, Off), and two different devices to test with (computer screen, cell phone screen on and off in each case), to conduct the screen/non-screen ground truth data collection. I brought my cell phone and computer with me onto the outside deck, during the nighttime, and let the computer and microcontroller rest outside for a couple minutes to acclimate the sensors and hardware. Next, I brought the micro-controller with the light sensor up to the lit-up screen computer and recorded the data. For every environment I changed to I repeated the same data collection process for getting values with screen on and off with and without ambient light. I then did this same experiment to collect data when the micro-controller was pulled away from the screen. After the ambient light test in this environment, I began taking temperature readings for the 1-minute time frame. Once I collected all this data, I repeated the same process in the living room and then again in the bedroom. I then labeled all the data files correctly and stored them in the CSV file format.

C: Results

For the experiments I used two classifiers, temperature, and light sensor values to determine if the screen was on or off. In regard to performance measures, the accuracy of the model generated when using the data from just the light value created a 95% accuracy.

=== Confusion Matrix For Tempval and Lightval Model === 99% accuracy

a b <-- classified as

41808 74 | a = off

282 40911 | b = on

Sensitivity – $a/a+b = 41808/41808+282 \Rightarrow 99\%$

=== Confusion Matrix Light value Model === 95% accuracy

a b <-- classified as

40482 1400 | a = off

2281 38912 | b = on

Sensitivity – $a/a+b = 40482/40482+2281 \Rightarrow 94.66\%$

The rest of the results are documented in the appendix PDF files including recall, and precision from the confusion matrix. In order to experimentally minimize the sampling frequency in the experiment we can slow down the interrupt handler to whatever interval needed. I believe the photocell sensor contributes the most heavily to the resulting event classification. The algorithm is just an if statement block so in terms of performance it will perform according to the results of the confusion matrix above. During the experiment I was able to extract features such as on off screen, light sensor sensitivity, room temperatures. I only decided to use the on off sensor feature to use in the final classification model. In order to validate the training classification I used the 10-fold validation method discussed in class. Weka allowed me to select my validation method. The results of the classifications can be found in the appendix PDF's.

D: Discussions

The process of feature extraction and selection had involved much arduous data collection and formatting to be feedable to the weka interface. I was able to select classifiers by setting the 'ground truth' data and feeding it into the algorithm and validation methods provided in Weka. This was very useful in creating the actual model because Weka provides an easy to use tree view method that helped me implement the model. I was able to extract the on/off screen feature, the temperature in each room, and if ambient light was on or off from the data. I ended up only using the Light value sensor readings to determine screen on or off. From the results of the real-time screen monitoring it is easy to see that the system is working but not perfect. If I were to expand this system I would have to build a larger model that can classify data based on more interesting features and go through and label more data. In the real time monitoring system, I just carried my computer around the room and put the sensor near or far from the screen and did the same with a cell phone screen. I watched as the results unfolded properly. My system does perform better when I supply more vibrant values to the light sensor as when I bring the sensor closer and more distinctly in view of the screens. I believe that this design is well thought out and could be optimized with more features, data, and testing. I would try to rebuild this system using much more accurate sensors and build much larger models with more accurate and precise results. I could attempt to implement the models of different algorithms and see if I get different results as a good test practice. The more accurate data I am able to provide, the better I can classify and build the models.