You have a new puppy, and you want to be able to see how their doing being in a kennel while you're at work. You hear about these smart web cams like the nest cam. But it costs more than $200 and you can't justify that cost for a puppy camera. Or you hear about break ins in your neighborhood and you want a smart camera that can not only show you what you what the camera is seeing, but also be notified when it detects motion. Again, security cameras are expensive, more than $200. Ladies and gentlemen, allow me to introduce to you the IoTCam. This camera can do all that I just mentioned for under $100. That’s right less than half the price of our competitors. And it can do it all, except for washing your laundry or dishes, that might be in a future version.

Why?

I designed this camera because my wife and I have recently gotten a new puppy and we wanted to be able to check on her to see how she is doing. And as the reasons before the current options on the market are either too expensive or have a subscription cost. As a result, this cheaper camera that I can choose the endpoint it communicates with is a perfect option.

What?

This was built using a RaspberryPi 2 with an old Xbox 360 Vision camera I had lying around. Making the build for this the cost of the pi, which I already had. The libraries I'm using allow me to be able to use any camera I can connect to the raspberry pi, USB or a PiCam. Pretty much all the magic happens on the raspberry pi. It takes the pictures and upload them to the server. Then it takes the latest image and compares it to the previous one and determines a percent difference. Though due to limitations on the algorithms for computing the differences between images I had to use 2 different algorithms so that slight changes in light level don't trigger it, but also it can be triggered by slow moving objects. If it detects a decent difference between the two images it will call and endpoint on the server that will handle sending out notifications. Though in theory I could have had the pi do that since I'm just using gmail. But I do have plans using the DreamSpark software to setup my own smtp server that would be able to send emails without needing google. Which would be a benefit having on the server, plus being able to utilize a database for sending multiple emails as well as attachments. But I'm getting ahead of myself, that was out of scope for this initial phase for the MVP (Minimum Viable Product).

Scale/Reliable?

Since so much of the work is happening on the device not the server, that makes it easy for this system to scale as the server power needed per device is tiny compared to the power needed on the device. However, this won't handle intermittent internet connectivity since it does communicate with a server. But I feel this is acceptable since if you lose internet how do you expect to receive notifications from you IoTCam? And if you say, "Why don't you use Bluetooth?" I say "Why don't you get up off your butt and check out that noise? Exercise can be good for you!".

UX?

The user interface is fairly easy once it is connected to the in home WiFi, and provided power, you can just forget about it. Let it run forever no need to interact with it, heck you could even forget where exactly it is and it will keep going about it's business without a care in the world.

The Future?

I would like to be able to adjust settings on the server that the pi then will receive to be able to customize the sensitivity of the image comparison, and even try to utilize the MQTT communication for being able to notify users if their IoTCam ungracefully disconnects. I'd also like to get the program that I wrote to start running when the pi boots so that I don't have to manually log in and start running it.

Different?

If I had more time, or had known about it sooner, I would have bought a Raspberry Pi Zero W for this. While I do have a Raspberry Pi Zero I couldn't utilize it for this project as I needed 1 USB port for power, 1 for the wifi dongle, and 1 for the camera. My current raspberry pi zero as only 2 usb ports so that would leave me short one. However, there isn't any reason why a raspberry pi zero w cannot run this program. So, if you grab my code from github, put in on a pi zero and on a server (making the needed changes to the url in the pi code) you should be able to build this for about $15 since I have seen $5 usb cameras and the pi zero is $10.

Demo