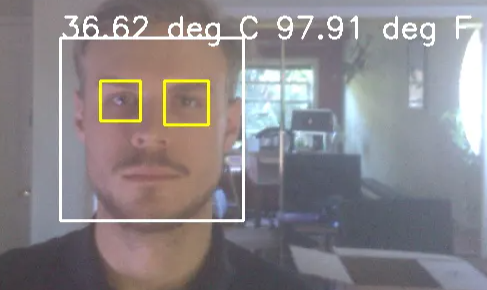
Most importantly:

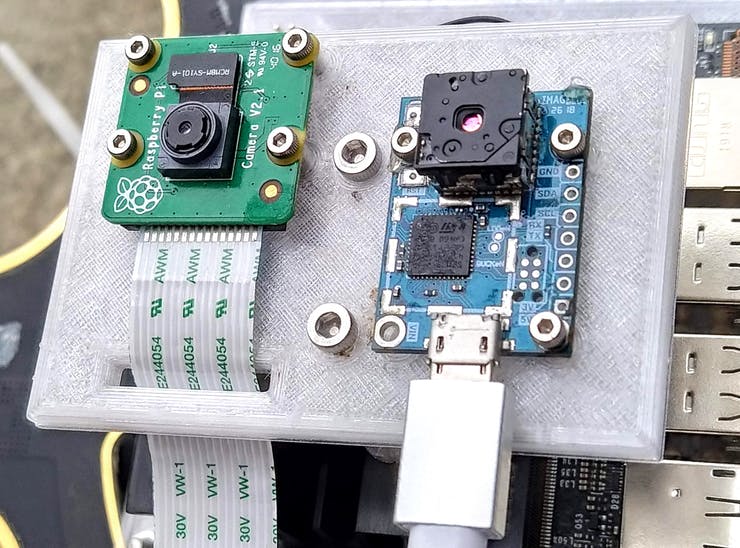
* The device automatically detects people in the scene - there's no need for a human operator!
* You can test multiple people at the time.
* You can test persons on-the-fly, without interrupting the flow.



**2. Hardware**

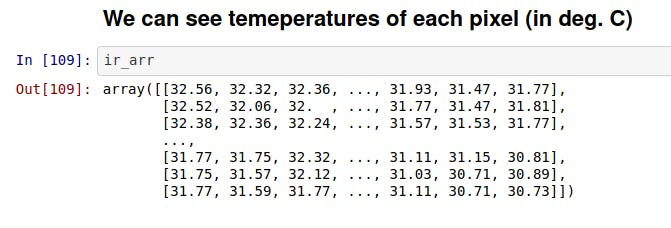
**2.1 IR Camera**

I'm using a FLIR Lepton 3.5 module. The camera offers 160×120 resolution, and it's pretty hard to beat for the price. Additionally, I use a Group Gets Pure Thermal 2 carrier board, which allows me to use a interface with the camera using USB.



Dual Camera Setup (early prototype)

The most important feature of the lepton is radiometric measurement. This means that with just a few lines of code, I can extract the temperature of every pixel in the image (not all LWIR cameras offer that).



IR image can be easily converted to an array of temperature values

With another few lines of code we can locate the hottest and coolest spot in the image.

Radiometric Measurement

**2.2 RGB (Visual Spectrum) Camera**

I decided to use Raspberry Pi Camera Module v2 for the following reasons:

* CM v2 works out-of-the-box with nvidia Jetson Nano.
* It features 4 mounting holes, so it's easy to position it next to IR camera
* It has 62.2° field of view (FOV). This is pretty close to the 57° field of FOV of the IR camera. The images still need to be aligned, but less pixels will "go to waste" than with, say a 120° FOV camera.

Below are images taken with my device:



1 / 3 • 8-bit IR spectrum image

Below are sample videos taken by my device:

* The blue channel contains the visible spectrum image (captured by Pi CM v2),
* The red channel contains the IR video thresholded at 30°C to show the person

It looks like I'm being followed by my thermal ghost!

Visible (blue channel) and IR (red channel) video overlay. Images are not aligned yet

**3. Software. Facial/Object Detection**

To figure out what the temperature of the subject, we first need to locate them in the image frame. I ended up creating two versions and testing two different approaches for the human detection task:

**3.1 General Purpose Object Detection with SSDobject detection**

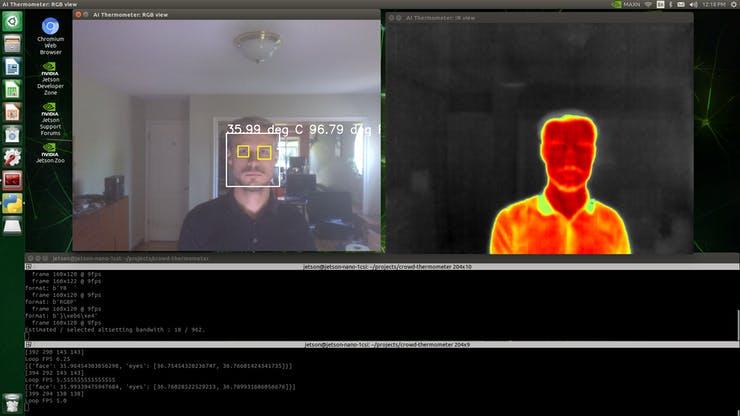
In my first approach, I used a SingleShot MultiBox Detector[[1]](https://arxiv.org/abs/1512.02325) trained on COCO dataset[[2]](https://cocodataset.org/). This lets me detect objects across 91 classes from COCO. The algorithm runs on Jetson Nano's embedded GPU at 9FPS. Below you can see an image that has a bounding box with the detected object class ("person"), the confidence score, and the temperature inferred from the IR image (in both deg. Celsius and Fahrenheit).



We are not limited to humans! With SSD object detection on-board my device can detect and measure the temperature of many common objects, like on the video below:

**3.2 Facial + Eye Detection based on Haar Cascade Classifiers.**

For the purpose of detecting fever, it is sufficient to detect faces, and not the entire body outlines of the subjects. We might even get more robust results, because the measurement will not be affected by the person's clothing. Also, in [this](http://qirt.org/archives/qirt2008/papers/03_07_17.pdf) 2008 study[4], authors identified the inner canthus (corner of the eye) as the most reliable region for contactless temperature measurement with IR cameras. That's why I also detect subject's eyes in the image.



Graphical Interface

I realized face and eye detection using a Haar Cascade Classifier. You can learn more about this algorithm from this [OpenCV tutorial.](https://docs.opencv.org/4.3.0/db/d28/tutorial_cascade_classifier.html)

Although detection algorithm is not as versatile as SSD, I found the temperature estimation to be more accurate when we estimate from the facial region (as opposed to the whole body).

**3.3Fever Detection**

Following the paper[[3]](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7168465/) by Ng, I assume that the skin emissivity equal to 0.98. The alarm threshold is set to 37.5°C.

**3.4 Interface**

One option is to simply connect the device to a computer monitor, and we can see detected persons and their temperature on the screen. I realize that this can be inconvenient for a lot of places. That's why I also included an alarm light with this 8 neopixel stick for a much more self-contained interface.

**4. Assembly. How to build your own?**

I created an enclosure for my device. The enclosure is fully 3D-printable.

**4.1 Download & print the enclosure**

You can view the 3D model here: <https://a360.co/3g8kfqV>

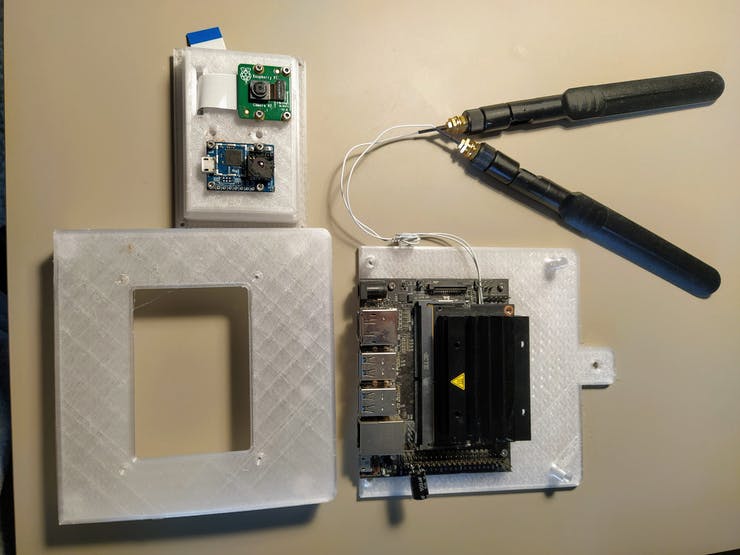
It is possible to export to your format of choice (.stl,.step, fusion 360) directly from the autodesk360 website. It's my first time using Fusion 360 export feature, so if something doesn't work, let me know -- I can also upload the files to thingiverse.

**4.2 Assemble the device**

You will need the printed parts and also:

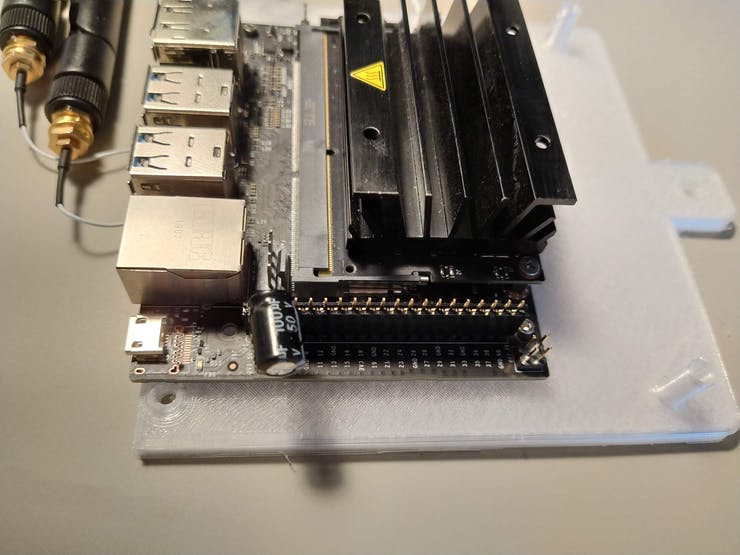
* M3x20 screws (4pcs) for the main enclosure assembly
* M2x8 screws (12pcs) for the camera assembly

4.2.1. Attach the cameras and Jetson Nano to their respective baseplates. Note that the wifi antennas for Jetson Nano are optional, but can help you integrate the AI thermometer with external service (e.g. website, mobile app notifications)

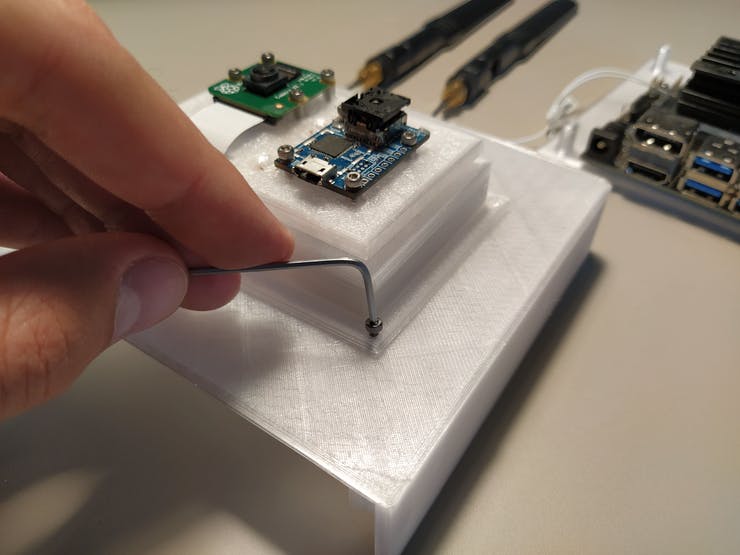


4.2.2. Add a capacitor to the 5V rail of Jetson Nano (optional).

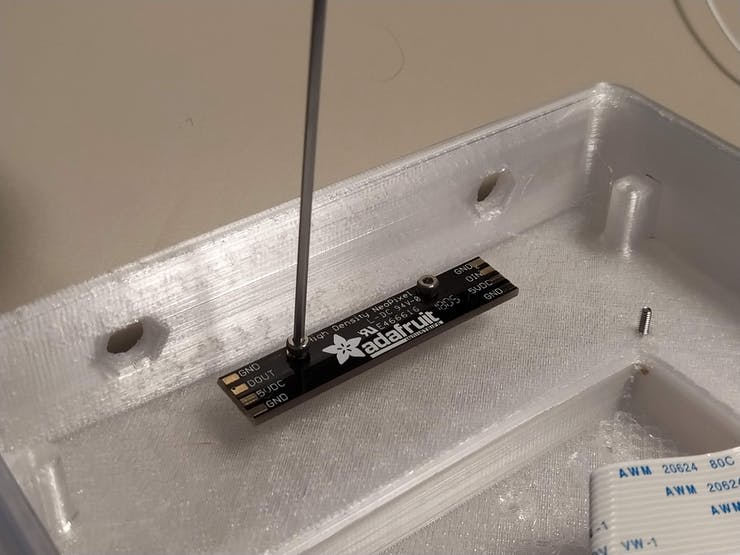
FLIR Lepton camera has high momentary current draw during a shutter event. It caused my Jetson Nano to shut off a few times and I added the capacitor to avoid this in the future. If your power supply has really good filtering, you might get away without this.



4.2.3. Put together the camera assembly with the top of the enclosure



4.2.4. Screw-in the Neopixel in its dedicated spot and add wifi antennas (if you choose to include them).



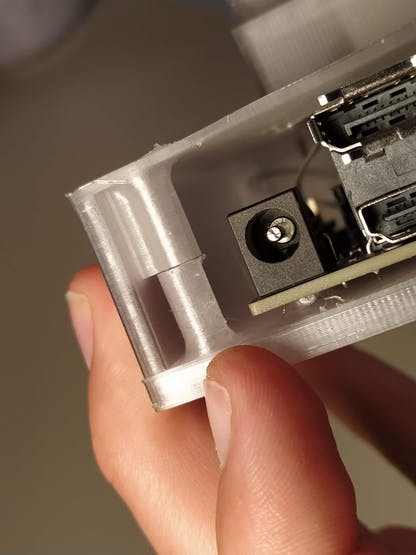
1 / 5

4.2.5. Connect the camera ribbon (this can be a little tricky, so be careful no to damage the flat-flex cable or the camera socket).



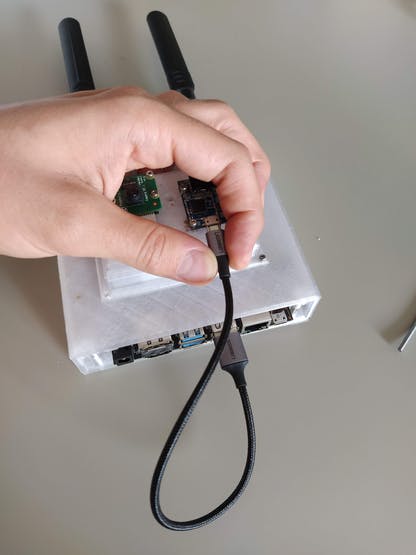
1 / 2

4.2.6. Put together the top and the bottom of the case. Use the M3 screws in this step.



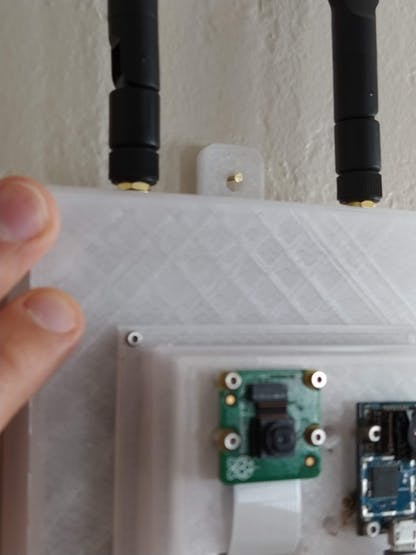
1 / 5 • Make sure the standoffs (bottom part) align with the screw sockets (top part)

4.2.7. Connect the FLIR Lepton with a short USB cable and connect the device to a 5V power source (USB or barrel jack)



1 / 2

4.2.8. Attach to the wall with a nail or double-sided tape. I found that the height ~170cm will work for capturing both short and tall people.



**4.3 Set up the software**

Go to <https://github.com/tomek-l/crowd-thermometer> and follow the readme for the most up-to-date setup. You can also "cheat" a little by downloading a pre-made SD card Image from here. If you have any issues, feel free to message me or create an issue on github.

I added [this notebook](https://github.com/tomek-l/crowd-thermometer/blob/master/Getting%20Started%20with%20Cameras.ipynb) as a good place to start working with cameras & IR image processing.

**5. Deployment**

AI Thermometer can be particularly useful in crowded places, such as:

* office entrances
* apartment building complexes
* metro/public transit stations
* store entrances



1 / 2 • Deployment near the entrance to a building

You can simply deploy it next to your doorbell.



1 / 3 • Deployment next to the door, near the doorbell

Or quite literally anywhere you can provide 5V USB power!



Deployment on a tree, with a powerbank :)

**6. Future improvements**

The improvements have more to with the accessibility of the project, rather than functionality.

* Moving to a lower cost computer. For example, moving to Raspberry Pi would cut the compute unit cost from 100USD to 35USD.
* Creating my own PCB hat with integrated LEDs a built-in socket for IR camera. With that I could further reduce the cost (by eliminating the USB carrier board for FLIR camera and the neopixel stick).

**7. Resources**

* You can see the source code for everything at my github: <https://github.com/tomek-l/crowd-thermometer>
* You can download all the 3D models from here: <https://a360.co/3g8kfqV>

**8. External References:**

[1] Liu et al. SSD: Single Shot MultiBox Detector, <https://arxiv.org/abs/1512.02325>

[2] COCO Dataset, <https://cocodataset.org/>

[3] Eddie Y.‐K. Ng Is thermal scanner losing its bite in mass screening of fever due to SARS? <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7168465/>

[4] Ring et al. Detecting Fever in Polish Children by Infrared Thermography <http://qirt.org/archives/qirt2008/papers/03_07_17.pdf>