Thermal sensor based human detection model

Kirill Tihhonov
Department of Computer Systems
Tallinn University of Technology
Tallinn, Estonia

Jelizaveta Kärner
Department of Computer Systems
Tallinn University of Technology
Tallinn, Estonia

I. Introduction

In this paper we will describe thermal sensor based human detection model project, which purpose is to develop a model that detects whether there is a real human in front of the sensor. Thermal sensor used in experiment outputs 32x32 array that measures temperature up to 6 meters from the sensor. Thermal sensor is attached to the ceiling of the room and measures any warm object temperature that emits heat.

There was done research before creating own solutions. However, in was decided to proceed with own approach.

II. State of the art

We have read 2 articles solving similar issues. First one is called "Indoor Human Detection Based on Thermal Array Sensor Data and Adaptive Background Estimation" [1], where authors present an innovative technique to detect people indoors using low-resolution thermal array sensor. Authors used a Kalman filter for noise filtering and background estimation technique to separate humans from background. Final results in real environment showed human detection accuracy up to 97%. Second one is called "Human Detection in Thermal Imaging Using YOLO" [3]. Authors check standard deep-learning methods applicable for object detection and recognition in RGB images, on thermal sensor images. Result of 90% dependent on scenario range was achieved.

We decided to pay special attention to the first article and the idea of separating a person from the surrounding background. However, if the source uses this methodology to train a neural network, then we decided to use more simplified methods of separating human temperature objects from the background exclusively at the data annotation stage. The second source uses thermal sensors with a higher resolution, which is not entirely relevant for this work.

III. Annotation

Dataset is prerecorded with 10 frames per second in JSON format and saved into .csv extension file. Each frame is stored in one row and consists of 32x32 readings together with timestamp, sensor ID, sensor size, room temperature and RSSI values.

It was chosen to detect the presence of human in sensor view by classification of 4 classes. Classes were chosen to be: None, 1 human, 2 humans, 3 humans. 3 humans were chosen as a limit since there was not present more than 3 human at the same time in given dataset. Classification of other objects especially warm non-human objects was considered to leave out of the scope of this project as there was not specified in the dataset presence of such objects as well as quantitative presence of those was unknown. The last is required for equal class distribution in training dataset.

Since dataset annotation is the most important part in machine learning related task, it was decided to give the most effort here. Studied research did not contain any answers how the data was annotated, so we came up with our approach. We wanted annotation to be done automatically to save the time required to label 10000 pictures from given dataset.

Unfortunately, in given dataset it was not specified what is present in picture. At least knowing how many humans would be helpful at this stage. So, we had to apply various filters and colormaps which helped us to highlight special image features. In the Figure 1 one example of created annotation program output is seen.

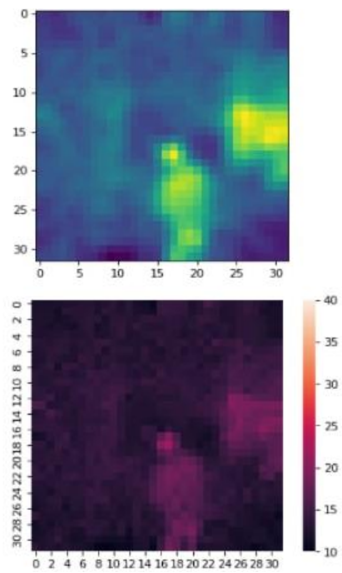

Figure 1 Annotation program output

As soon as we could determine how many humans are there in the picture, we created an edge detecting program which helped us to automate annotation. Some pictures like Figure 1 were hard to tell for sure how many humans are present there. For Figure 1 it was agreed that 3 humans are present in the picture. Also, since 2 supposed humans in the right side of the Figure 1 represent 1 contour it was impossible for an edge filter to output anything other than 1 in similar cases. Such labels had to be corrected manually. Nevertheless, in most cases edge detector worked perfect like is in seen in Figure 2.

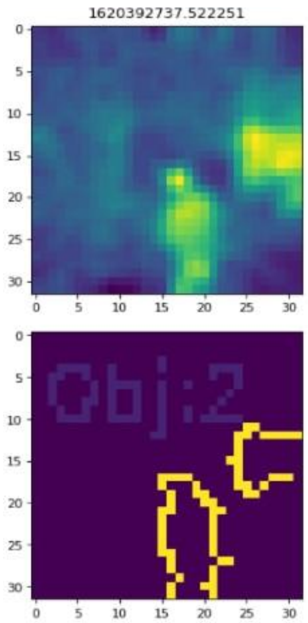

Figure 2 Edge Detector Output

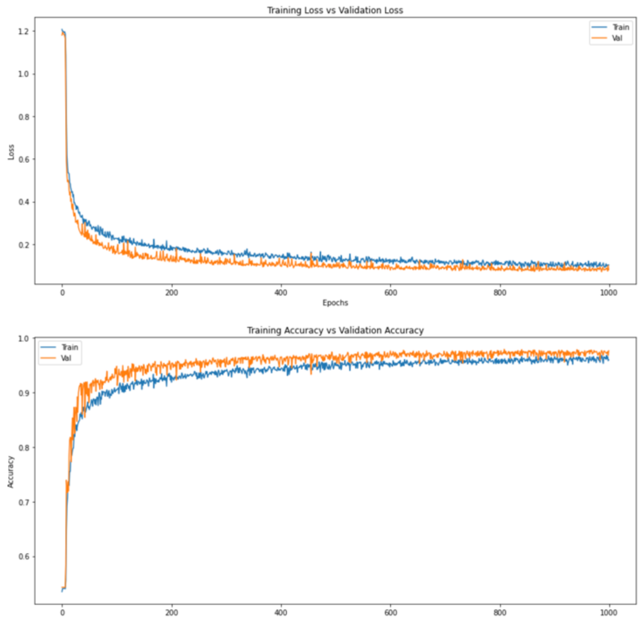

Figure 3 Optimized LeNet 5 accuracy

There were also faced issues with edge detector when sudden cold areas were detected as contours. This issue was solved filtering out anything colder than 15 degrees Centigrade. This number was chosen since it is significantly colder than median temperature for images having a human on those.

Thus, in total, a little over 4 thousand samples were annotated. Since there were practically no moments in the proposed datasets without any objects of human temperature, we also created about 5 thousand samples with random values in the range from 12 to 17 degrees.

IV. Model Creation

There were tried various architectures like MobileNet V2 and ResNet-152 for model creation. Accuracy achieved with MobileNet V2 was about 98%, however it turned out that model with MobileNet V2 architecture weighted 8.7MB, which is a lot more that STM32F429I-DISC1 controller could fit. As arguments inlude_top was chosen False since it was decided not to to include the fully connected layer at the top of the network; weights to equal to imagenet, so the pre-traing could be done on ImageNet, input_tensor equal to custom input layer (32x32x3), for pooling maximal and average were tried.

So, it decided to turn to other simpler architectures. One among those was LeNet 5 optimized architecture. It showed the same accuracy as MobileNet V2 and weighted significantly lesser – about 0.2MB. Thus, it was decided to continue with optimized LeNet 5.

During evaluation not only accuracy was examined. Having randomly distributed test dataset, distribution is shown in Figure 4, confusion matrix, seen in Figure 5, was created to give more information regarding prediction results. Then images and predictions were manually studied. Authors were satisfied with results.

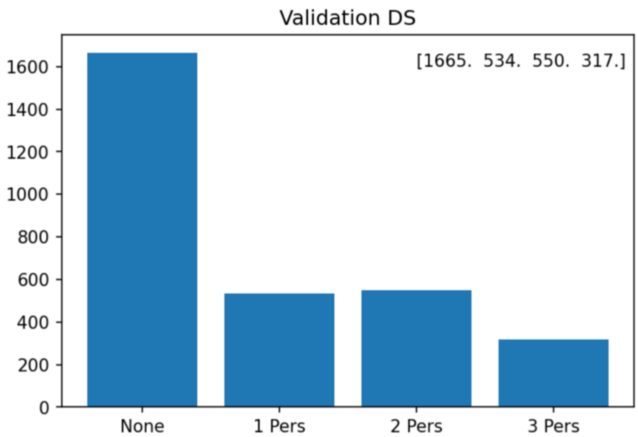

Figure 4 Test dataset distribution

Figure 5 Test dataset prediction confusion matrix

V. Deployment on Controller

TFLite model created in previous stage was imported into CubeMX. In CubeMX project was created from scratch. Timers were set the same as in the projects used in IAS0360 Machine Learning for Embedded Systems course lab works. The parts of the code related to accessing the model were partially taken from the course materials.

The program was written to check the accuracy of the model, so the buttons and screen were not included in the process. As an input, the program accepts an array from the dataset given along with the task. As an output, the program gives 2 predicts along with the probability as shown in Figure 6. Since the screen is not involved in the code, the program output is is carried out through the UART.

```
NN First Guess: 2  0.696080
NN Second Guess: 3  0.287377
```

Figure 6 Program output

VI. Conclusion

During this work, various filters were used, which made it possible to almost completely automate the data annotation process. Thus, more than 4 thousand samples were annotated and a date for a class without people was created in the amount of 5 thousand samples. Several neural network architectures were considered, of which LeNet 5 turned out to be the most suitable. This model, in turn, was also optimized to 7396 parameters. The TFlight version is launched on the MCU STM32F429 Discovery, which fully meets the requirements of the task of this project.

Authors are very satisfied with project. The launch of the model on the board proved its ability to identify 2 people standing behind each other. At the same time, the model makes the assumption that there may be 3 people in the image. In the case when the image contains neither people nor objects close in temperature to a human, the model determines the absence of a person with a probability of 100%. Theoretical accuracy of about 98% was achieved. However, it is important to clarify that this project is not able to distinguish a person from a warm object of the same temperature. Thus, the results can be improved in further work with a more complete dataset, which also takes into account the distance from the sensor and the size of objects.

VII. References

[1] Anna A. Trofimova, A. M. (2017). Indoor Human Detection Based on Thermal Array Sensor Data and Adaptive Background Estimation. *Journal of Computer and Communications*, 5, 16-28.

[2] K, D. (2018). *LeNet-5 CNN with TensorFlow - 98.5%*. Retrieved 2021, from https://www.kaggle.com/curiousprogrammer/lenet-5-cnn-with-tensorflow-98-5

[3] Marina Ivašić-Kos, M. K. (2019). Human detection in thermal imaging using YOLO. *the 2019 5th International Conference*.