Thermal sensor based human detection model

Sanan Suleymanov
Department of Computer Systems
Tallinn University of Technology
Tallinn, Estonia

I. Introduction

The detection of human using infrared cameras is going to be used widely because of some advantages as detection in different environmental conditions which normal cameras are not able to give appropriate results. In this project, for detection of human from infrared camera images YOLOv5n model is used. For training of model two classes are applied: human and non-human. Non-human object has high temperature but it is differentiated from human with its physical characteristics and the detection result is quite satisfactory.

II. Project Overview

This project defines one of the ongoing research field which solves important problem. This problem is the detection of the human in different environmental conditions. The RGB cameras can't perform well at night and as well as in different weather conditions. At the result, it makes infrared cameras the crucial part of the surveillance system [1]. The challenging point in this project is the detection of human and differentiating this human from non-human object which has also roughly equal temperature. It solved in the project using the a number of series captured images with different number of human together non-human object in the training of model. The project has two main goals: creating object detection model which can detect human and non-human objects from infrared images, and has appropriate memory size for microcontrollers.

III. YOLOv5 MODEL

YOLO is the abbreviation of "you only look once" which divides image to the grids and each cell is in charge of detecting objects within itself. Model is applied to the image at different scales and locations. The region which is scored high is considered as detection.

Another feature of YOLO model is its using of single network evaluation for prediction not like other models as R-CNN which for single image uses thousands. It makes YOLO 1000 times faster than R-CNN and 100 times faster than Faster R-CNN. YOLOv5 combines architectures and models for detection of objects that are trained using COCO dataset and Pytorch framework[2]. The main reasons of the using of YOLOv5 are its speed and accuracy. The comparison of YOLOv5 models speed with EfficientDet is shown in Fig.1.

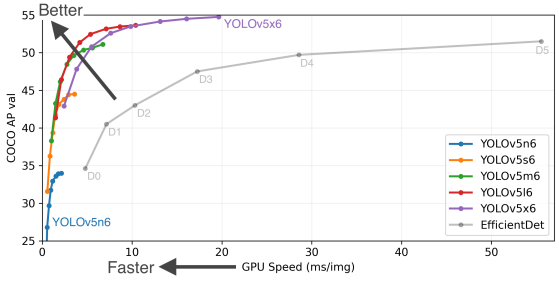

Fig. 1. YOLOv5 models speed and average precision comparison [2]

YOLOv5 family has models which are differ from each other because of their precision and size. For this project two of them are experimented: YOLOv5n and YOLOv5s, while the result of YOLOv5n is chosen as suitable. These two models are preferable for this project because of the size of model and appropriate high precision value. This can be seen by the Fig.2.

Model	size (pixels)	mAP^{val} 0.5:0.95	mAP^{val} 0.5	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.4	46.0	**45**	**6.3**	**0.6**	**1.9**	**4.5**
YOLOv5s	640	37.2	56.0	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.2	63.9	224	8.2	1.7	21.2	49.0
YOLOv5l	640	48.8	67.2	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7
YOLOv5n6	1280	34.0	50.7	153	8.1	2.1	3.2	4.6
YOLOv5s6	1280	44.5	63.0	385	8.2	3.6	12.6	16.8
YOLOv5m6	1280	51.0	69.0	887	11.1	6.8	35.7	50.0
YOLOv5l6	1280	53.6	71.6	1784	15.8	10.5	76.7	111.4
YOLOv5x6	1280	54.7	**72.4**	3136	26.2	19.4	140.7	209.8
+ TTA	1536	**55.4**	72.3	-	-	-	-	-

Fig. 2. YOLOv5 models [2]

YOLOv5n model architecture consists on backbone and head and they are described below:

backbone:
[[-1, 1, Conv, [64, 6, 2, 2]],
[-1, 1, Conv, [128, 3, 2]],
[-1, 3, C3, [128]],
[-1, 1, Conv, [256, 3, 2]],
[-1, 6, C3, [256]],
[-1, 1, Conv, [512, 3, 2]],

[-1, 9, C3, [512]],
[-1, 1, Conv, [1024, 3, 2]],
[-1, 3, C3, [1024]],
[-1, 1, SPPF, [1024, 5]],
]

head:
[[-1, 1, Conv, [512, 1, 1]],
[-1, 1, nn.Upsample, [None, 2, 'nearest']],
[[-1, 6], 1, Concat, [1]],
[-1, 3, C3, [512, False]],

[-1, 1, Conv, [256, 1, 1]],
[-1, 1, nn.Upsample, [None, 2, 'nearest']],
[[-1, 4], 1, Concat, [1]],
[-1, 3, C3, [256, False]],

[-1, 1, Conv, [256, 3, 2]],
[[-1, 14], 1, Concat, [1]],
[-1, 3, C3, [512, False]],

[-1, 1, Conv, [512, 3, 2]],
[[-1, 10], 1, Concat, [1]],
[-1, 3, C3, [1024, False]],

[[17, 20, 23], 1, Detect, [nc, anchors]],
Detect(P3, P4, P5)]

IV. Dataset for Project

The task for this project is creating of model which can handle detection of human and non-human objects. For the project, given dataset is divided to train, test and validation. The 168 images for train, 21 images for test and 17 images for validation are used. These images are labeled using LabelImg software with bounding box method and the labeling information is saved at .txt format.

For training the model, two classes are chosen: human and non-human and the images labeled according to these classes. The description of bounding boxes in .txt file is shown below where 0 is human and 1 is non-human:

0 0.227429 0.643293 0.196571 0.496951

0 0.421714 0.509909 0.102857 0.266768

1 0.590286 0.775915 0.058286 0.091463

The problems in the labeling are finding human, differentiation from non-human object which also has high temperature and drawing exact bounding boxes for them. It is solved by looking at the series of the number of images and at the result it is seen that human can be differentiated from other static object because of its behaviour and also physical characteristics.

V. YOLOv5s Model Result

In the first phase of the project, YOLOv5s model was tried and high precision model developed. For the training, epoch 100 and batch size 16 were chosen. The validation result of model after training is described below:

```
Model Summary: 213 layers, 7015519 parameters, 0 gradients
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.37it/s]
                 all         17         63      0.955      0.978      0.971      0.708
               Human         17         46      0.917      0.956      0.947      0.686
           Non-human         17         17      0.994          1      0.995       0.73
```

The test result of model in one of the test images is shown in Fig. 3.

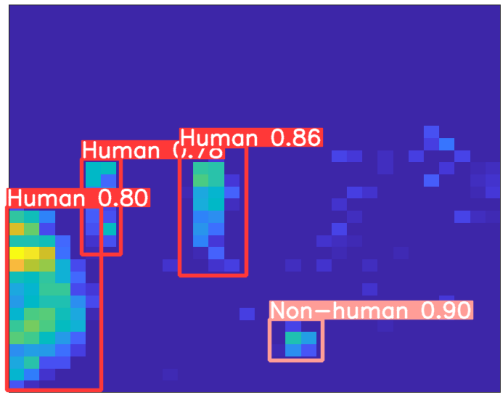

Fig. 3. Detection of human and non-human objects using YOLOv5s model

The problem in the YOLOv5s model is its size. Without optimization and conversion to .tflite, model size is 14.4 MB and after optimization and conversion, it is 7.4 MB which is higher for MCU.

VI. YOLOv5n Model Training

For the training purpose, we use pretrained model which is YOLOv5n, and batch size and the number of epochs are 16 and 100 respectively. For the determining of the precision of the detection minimum average precision metric (mAp) is used.

The training results are quite satisfactory and the speed of training is higher. The 100 epochs was completed in 0.178 hours and at the end of the training value of mAp 0.5 is equal to 0.966 and mAp 0.5:0.95 is equal to 0.68.

One of the advantage of using YOLO, is its availability of determining best performed model during the training process and the results of best model:mAp 0.5 is equal to 0.965 (for human 0.935 and for non-human 0.995) and mAp 0.5:0.95 is equal to 0.73 (for human o.65 and for non-human 0.757).

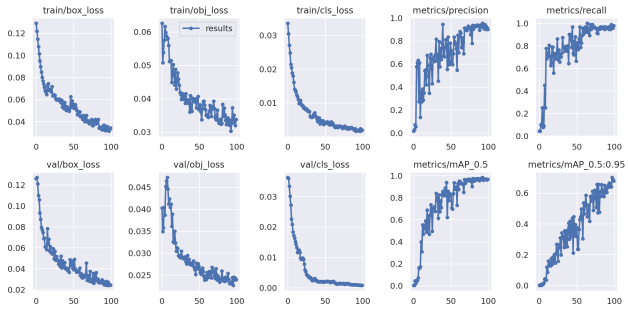

Fig. 4. YOLOv5n model training results

VII. YOLOv5n Model Testing

After training process, the model is testes using test dataset for best performed model as well as the last model. The detection of human and non-human objects in one of the test image is shown in the Fig.5 and Fig.6 for best and last model respectively.

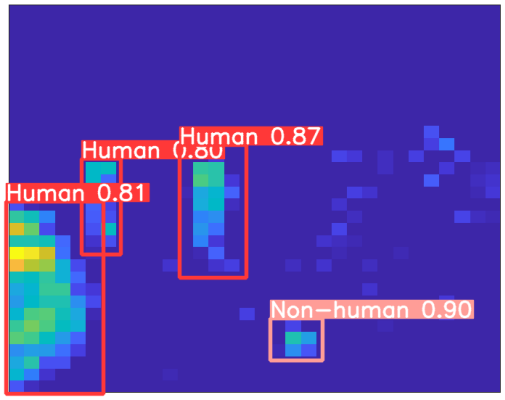

Fig. 5. Detection of human and non-human objects using YOLOv5n best model

For analyzing of the accuracy of detection, one method is confusion matrix.

The YOLOv5n model size is equal to 3.9 MB before convertion to the .tflite format.In YOLOv5, there is export.py file which allows to export model in different formats and one of them is Tensorflow Lite. After quantization and exporting model, the model size is equal to the 2.0 MB. This memory size is appropriate for MCUs and there is possibility to reduce it more by doing regulations such as decreasing number of parameters in the architecture layers.

VIII. Problems

The model size and accuracy of detection covers the requirements and expectations for accomplishment of project. However, one problem appeared during uploading .tflite model to the STM32CubeMX that gives error

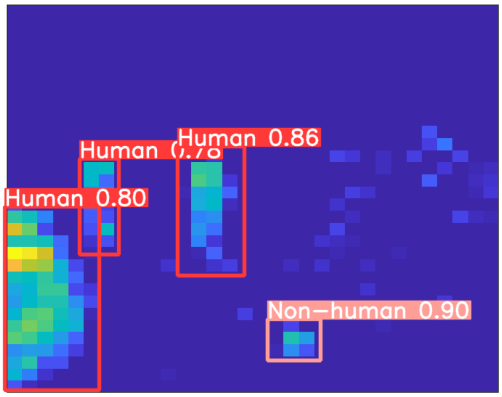

Fig. 6. Detection of human and non-human objects using YOLOv5n last model

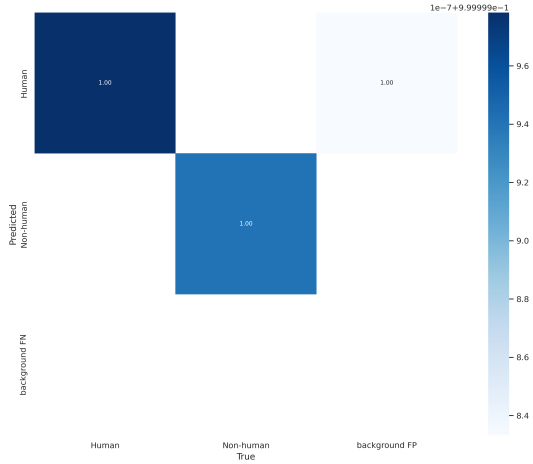

Fig. 7. Confusion matrix of detection of YOLOv5 best model

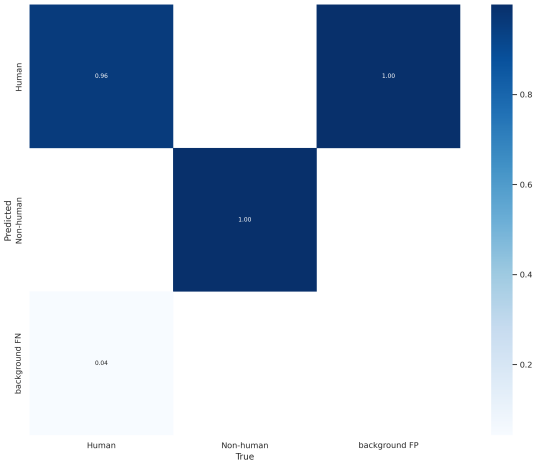

Fig. 8. Confusion matrix of detection of YOLOv5 last model

without description. There are some possibilities for the reason of error: some layers of model are not supported by STM32CubeMX, model architecture is complex or model size is not small enough. If the current error is related to memory size, it must have its own error message and that's why first two options are more actual. The error message when analyzing the model is shown in Fig.9.

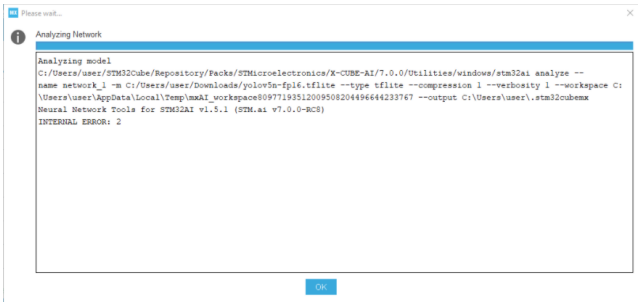

Fig. 9. Error message of analyzing of Tensorflow Lite model

IX. Conclusions

Detection of the human from the infrared camera has advantage in different application areas. The project has the goal to develop a detection model which has low memory size and able to detect human and non-human objects from the infrared images. In this project, YOLOv5n and YOLOv5s model are experimented for detection. Both of the models have close precision for getting high quality detection. However, we need a model which has appropriate size for microcontroller and that is why YOLOv5s model is suitable for the project. The YOLOv5n model has both lower size and high precision. However, for the uploading model to the STM32 based microcontroller, it was analysed using STM32CubeMX and during analysing error message was given. It could be because of the complexity of the architecture of model or some layers can't be supported by software. Generally, the project fulfill main goals and can be improved by the solving issue of uploading to the MCU or another type of Software can be tried in the next phase of project.

References

[1] M. Kristo, M. Ivasic-Kos, and M. Pobar, "Thermal object detection in difficult weather conditions using yolo," *IEEE Access*, vol. 8, pp. 125 459–125 476, 2020.

[2] "Redmon, j., 2022. yolo: Real-time object detection. [online] pjreddie.com," https://pjreddie.com/darknet/yolo/, accessed 7 January 2022.