Efficient People Counting in Thermal Images

Mateusz Piechocki [mateusz.piechocki@doctorate.put.poznan.pl]

Institute of Robotics and Machine Intelligence, Poznan University of Technology, 60-965 Poznan, Poland

Introduction

Accurate occupant counting is vital aspect of an efficient Smart Building Management System. Present-day most rooms, halls, and auditoriums utilize only a binary presence detector which turns on or off all HVAC systems. This strategy is not optimal and wastes a vast amount of actuators' energy related to the control. The camera-based people counting method is a well-known area of computer vision applications. This research approach can be extended to thermal imaging. The VisDrone CC dataset [1] demonstrates the potentiality of the top view perspective in the crowd counting task.

The occupancy in a compartment can be presented as simple binary information about people's presence [2] or as a natural number (in a limited range) which is an output class of the classification model[3]. A novel approach presents [4], the UNet-based [5] neural network encodes information gathered from the thermal camera then decodes density map from the latent space that post-processed represents a number of people in the sensing area.

This poster presents a simplified UNet architecture and results of performed hardware benchmarks. The feature of the proposed model is a tiny, shallow architecture which combined with a small input size (32 × 24 pixels), enables deployment on microcontrollers. Moreover, also crucial is an assumption that continuous inference is not mandatory and several measurements per minute are enough for controlling HVAC systems. This application allows devices to be both efficient and cheap in utilization and exploitation. Using microcontrollers as inference devices ensures the possibility to place next to the sensor or integration. It reduces transmission time and maintains the compact dimensions of the hardware. Furthermore, only post-processed information about occupancy count is sent to the control unit to prevent privacy.

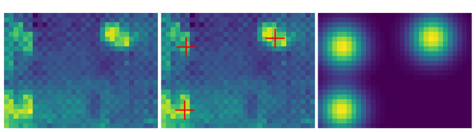
Thermo-Presence dataset

For the project purpose, the Thermo-Presence dataset[4] was utilized. It consists of 13 643 marked infrared images divided into training, validation, and test sets according to Table 1.

Table 1: Thermo Presence dataset characteristics and training, validation and test sets division

	Number of persons in a frame						
Dataset	**0**	**1**	**2**	**3**	**4**	**5**	**Total**
Training	99	105	2984	3217	1953	114	8472
Validation	0	139	631	1691	225	139	2825
Test	162	83	211	341	1235	314	2346

For every frame in the dataset provided is information about the center location of persons. The density map of occupancy was created using the convolution of the thermal image with a 2D Gaussian mask (σ=3). This procedure creates a Gaussian mixture distribution with maximums in the central locations of people and a 3×σ radius. Figure 1 depicts an example image from the dataset, corresponding localization of persons, and created density map.



(a) Input thermal image (b) Locations of people (c) Density map

Figure 1: Thermo Presence dataset example thermal image, persons location annotations and generated density map.

Methodology

The Neural Network model is based on UNet [5] architecture. It consists of encoder and decoder parts. The encoder section is shallow and consists of two convolutions at the beginning and only one nesting level with a max-pooling layer and four convolutions. The model decoder includes a concatenation layer and three convolutions. Each convolution has a 3x3 kernel size and is followed by ReLU activation. Figure 2 depicts the diagram with the model structure, N denote the number of filters that equals 16. Indicated model is smaller than the state-of-the-art solutions [3,4] and has only 46 577 parameters (respectively 8.5 and 2.8 times less).

The proposed architecture for every input thermal image model returns a density map of people localizations. The people count can be calculated by summing the output mask and dividing by the constant factor taken in the data preparation process as a value corresponding to 1 person (51.35).

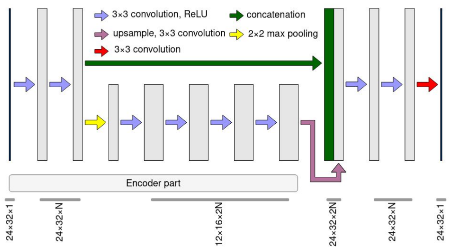


Figure 2: Proposed UNet-based neural network architecture.

Results

The standard metrics like MSE and MAE were used to evaluate the proposed model architecture. Additionally, the count-specific metrics like Counting MAE, Counting MSE, and Counting MRAE (count mean relative absolute error) that maps the MSE and MAE from density map scope to people count-level were implemented.

$$CountingMRAE = \frac{1}{n}\sum_{x=0}^{n-1}\frac{|\ count_{gt} - count_{pred}\ |}{count_{gt}} \quad (1)$$

The explored problem is defined as regression rather than classification nevertheless the output number of people in the frame is a discrete value. Furthermore, ground truth labels are available in the dataset. Therefore, outcomes include metrics like accuracy and F1 score. All results are shown in Table 2, whereas Figure 3 depicts the confusion matrix with percentage values of people number classification.

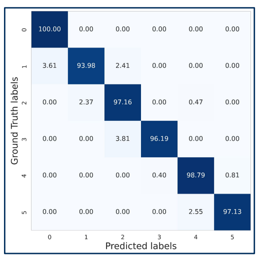


Figure 3: Confusion matrix with percentage values of people number classification.

Table 2: Model metrics

Metric Name	Result
MAE	0.1056
Counting MAE	0.0226
MSE	0.0331
Counting MSE	0.0234
Counting MRAE	0.0081
Accuracy	0.9778
F1 Score	0.9782
No. of parameters	46 577

Benchmark Hardware

To inspect the practical utilization of the proposed algorithm, both development boards with microcontrollers supporting neural networks processing and edge device - Raspberry Pi 4B with computing accelerators - were chosen (Table 3).

The deep learning model was converted and optimized from TensorFlow SavedModel to TensorFlow Lite FlatBuffer file format and then quantized to INT8 data representation. The TF Lite Micro [6] C source file was created using Linux *xxd* package that converts the TF Lite representation into a hex dump. STM32 Nucleo boards were flashed using the CubeMX tool, and the neural network architecture was optimized using the Cube AI package. Model evaluation on Intel Neural Compute Stick 2 accelerator was available after conversion from TensorFlow to ONNX and then optimization to OpenVINO Intermediate Representation with FP16 data format.

Table 3: Microcontrollers and edge devices used in benchmark.

Device	Target hardware	Evaluation framework
Arduino Nano 33 BLE Sense	nRF52840	TF Lite Micro
Arduino Portenta H7	STM32H747	TF Lite Micro
STM32 F429ZI Nucleo-144	STM32F429	STM32 Cube AI
STM32 H745ZI Nucleo-144	STM32H745	STM32 Cube AI
SiPEED MAiXDUINO	ESP32-WROOM-32	TF Lite Micro
Raspberry Pi 4B	Quad Core Cortex-A72	TensorFlow Lite
Raspberry Pi 4B + Coral TPU USB Accelerator	Google Edge TPU coprocessor	TensorFlow Lite
Raspberry Pi 4B + Intel Neural Compute Stick 2	Intel Movidius Myriad X Vision Processing Unit	OpenVINO

Hardware Evaluation

The aim of hardware evaluation was to benchmark the inference performance and current consumption. For this purpose devices characterized in Table 4 were used. The latency was measured on the whole test set. Regarding the microcontrollers' memory limitations, UART communication was utilized. Data transmission enables sending input signals and receiving an output density map. The current consumption was measured and logged using a Mooshimeter multimeter with a 9Hz log rate. The measurement time was similar for each target hardware and unrelated to the number of samples.

Table 4 contains a comparison of average inference time and max power consumption reached during measurement (at 5 V power supply). Whereas figure 4 depicts the current consumption of the Arduino Nano 33 BLE Sense board. It shows the repetitiveness of the processing period with spikes at the time of inference.

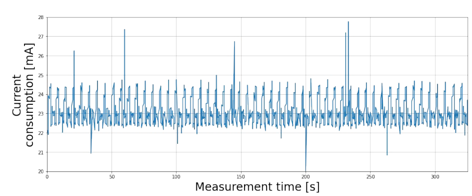


Figure 4: Measured current consumption of Arduino Nano 33 BLE Sense with nRF52840 microcontroller. Average frame process time was about 6.5 second. Measurements were recorded with 9 Hz rate using Mooshimeter multimeter.

Table 4: Hardware performance measurements.

Device	Inference time [ms]	Max power consumption [mW]
Arduino Nano 33 BLE Sense	1430.13 ±1.14	140.19
Arduino Portenta H7	137.49 ±0.05	933.08
STM32 F429ZI Nucleo-144	230.94 ±0.01	813.98
STM32 H745ZI Nucleo-144	53.26 ±0.01	1315.33
SiPEED MAiXDUINO	1259.87 ±0.02	800.93
Raspberry Pi 4B	4.19 ±0.05	4231.46
Raspberry Pi 4B + Coral TPU USB Accelerator	0.57 ±0.06	5360.91
Raspberry Pi 4B + Intel Neural Compute Stick 2	2.30 ±0.10	5410.41

Conclusion

This poster expands the research related to occupancy monitoring, focusing on hardware implementation. The performed tests ensure that the proposed method is suitable for low-cost, resource-constrained hardware like edge devices or microcontrollers as a target platform. The model consists of fewer parameters, is smaller, faster, and achieves better metrics than similar solutions.

Additionally, with low energy consumption proposed people counting algorithm can be utilized directly as a component of the HVAC perception unit. Moreover, contrary to the traditional, commonly used CO2 sensor, the proposed solution is not affected by measurements inertia. According to [4], UNet based model can ignore non-human heat sources. Furthermore, the inference latency of the algorithm allows quick response from the perception section to the HVAC system controller. It can result in better and more effective actuators adjusting to the existing conditions.

The used approach preserves privacy by sending only information about the estimated number of people. Furthermore, the algorithm uses low-resolution infrared images that prevent person recognition.

References

[1] Z. Liu et al., "VisDrone-CC2021: The Vision Meets Drone Crowd Counting Challenge Results," 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2021, pp. 2830-2838, doi: 10.1109/ICCVW54120.2021.00317.

[2] Abedi, M., & Jazizadeh, F. (2019, May). Deep-learning for occupancy detection using Doppler radar and infrared thermal array sensors. In Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC).

[3] Metwaly, A.; Queralta, J.P.; Sarker, V.K.; Gia, T.N.; Nasir, O.; Westerlund, T. Edge computing with embedded ai: Thermal image analysis for occupancy estimation in intelligent buildings. In Proceedings of the INTelligent Embedded Systems Architectures and Applications Workshop 2019, New York, NY, USA, 13–18 October 2019; pp. 1–6.

[4] Marek Kraft, Przemysław Aszkowski, Dominik Pieczyński, and Michał Fularz. Low-cost thermal camera-based counting occupancy meter facilitating energy saving in smart buildings. Energies, 14(15), 2021.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015

[6] Robert David, Jared Duke, Advait Jain, Vijay Janapa Reddi, Nat Jeffries, Jian Li, Nick Kreeger, Ian Nappier, Meghna Natraj, Tiezhen Wang, et al. Tensorflow lite micro: Embedded machine learning for tinyml systems. Proceedings of Machine Learning and Systems, 3:800–811, 2021.

POZNAN UNIVERSITY OF TECHNOLOGY

www.PosterPresentations.com