

OPEN-SOURCE SOFTWARE FOR THERMOGRAPHIC ANALYSIS OF THE HUMAN HAND USING YOLO INSTANCE SEGMENTATION

Marcin PIOTROWSKI

Department of Simulator Studies and Aeromedical Training, Military Institute of Aviation Medicine, Warsaw, Poland

Source of support: This work was supported by the Ministry of Science and Higher Education (Poland) under Grant [SPUB/SN/628251/2025].

Author's address: M. Piotrowski, Military Institute of Aviation Medicine, 54/56 Krasynskiego Street, 01-755 Warsaw, Poland, e-mail: mpiotrowski@wiml.waw.pl

Introduction: This note presents the concept and implementation of dedicated software for the automated analysis of thermograms of the skin surface of the human hand. Standard tools supplied by thermal camera manufacturers often restrict access to raw radiometric data, hamper batch data processing, and force reliance on closed ecosystems.

Methods: In response to these limitations, a custom data processing chain was developed. The process comprises the extraction of raw radiometric matrices and calibration parameters from the EXIF metadata of JPEG files, followed by the conversion of the radiometric signal into temperatures using a formula that constitutes an algebraic inversion of the simplified Planck model. The key component of the system is a YOLOv8n-seg instance segmentation model, initialized with weights pretrained on the COCO dataset and fine-tuned on a set of 52 precisely labeled thermal images (prepared in Label Studio) covering hands in three thermal states: after warming, at room temperature, and after cooling. The use of a model based on deep convolutional neural networks (CNNs) made it possible to isolate the hand region from the background automatically and to extract precise temperature statistics exclusively from pixels within the segmentation mask.

Results: The trained model achieved very high segmentation precision ($mAP50 = 0.995$ and $mAP50-95 = 0.944$). A demonstration comparison of two measurement sessions of the same person revealed a difference in mean hand surface temperature of -7.4°C between the hand at rest and the cooled hand, while the model maintained high detection confidence (0.974 and 0.964) in both cases.

Conclusions: The demonstration results indicate the effectiveness of the method in detecting skin-temperature differences between measurement sessions, providing a basis for further validation of the software in physiological studies.

Keywords: thermography; hand temperature analysis; You Only Look Once (YOLO) model; deep convolutional neural networks (CNNs)

Cite this article: Piotrowski M. Open-Source Software for Thermographic Analysis of the Human Hand Using Yolo Instance Segmentation. Pol J Aviat Med Bioeng Psychol 2025; 31(4): 36-44. DOI: 10.13174/pjambp.28.10.2026.04

Publisher: Military Institute of Aviation Medicine, 54/56 Krasynskiego St., 01-755 Warsaw, Poland • **License:** CC BY-NC 4.0 • **Indexation:** Ministry of Science and Higher Education (Poland) • **Full-text PDF:** <http://www.pjambp.com>

INTRODUCTION

Thermal imaging cameras are widely available research tools used in both technical inspection and the biomedical sciences [11]. Analysis of the temperature distribution across the human body surface can provide information about, for example, the condition of the circulatory system or metabolic processes. The reliability of such studies, however, depends on the ability to measure temperature precisely, reproducibly, and automatically within a strictly defined anatomical region — without operator intervention for each individual image.

The FLIR C8 thermographic camera [4] used in this work saves images in JPEG format, in which the raw radiometric data are hidden in the EXIF section of the file. These data, once extracted, allow the full temperature matrix to be reconstructed at the native resolution of the IR detector (320×240 px) — without the loss of information that results from compressing the visual image. This file architecture provides the starting point for building a custom data processing chain independent of the manufacturer's software.

Limitations of standard manufacturer software

The standard computer software supplied by thermal camera manufacturers is optimized for rapid technical inspection, which often proves insufficient in advanced research applications. The main barriers limiting effective work with thermograms include the following:

1. Difficult extraction of raw data. Manufacturer software often hides the radiometric data behind proprietary file formats, limiting the researcher's access to the original radiometric matrix in a format that would permit independent statistical analysis.
2. Exporting data to external analytical tools is usually possible only in a processed form, which precludes full control over the calibration of the radiometric signal and its conversion into temperature.
3. Time-consuming data processing. Most commercial packages require per-image processing, that is, processing each image individually during a separate operator session. This approach makes the automated mass analysis of large datasets impossible and thus rules out applications that require processing hundreds or thousands of thermograms in a reproducible manner and without human supervision.
4. Economic model and software life cycle. The high licensing costs, the ability to install the

software only on one specific operating system, and the uncertain duration of technical support — so-called software end-of-life — associated with proprietary, closed-source commercial software pose a real risk to long-term research projects. A change in software version or the discontinuation of manufacturer support may make it impossible to reproduce earlier results, which is particularly important from the standpoint of the reproducibility of scientific research.

METHODS

Proposed solution — a custom analytical system

Building a custom, dedicated software solution allows full control over the research process and eliminates all of the limitations listed above. The proposed radiometric data processing chain, shown schematically in Fig. 1, comprises four main stages performed in the Python programming environment using open-source libraries.

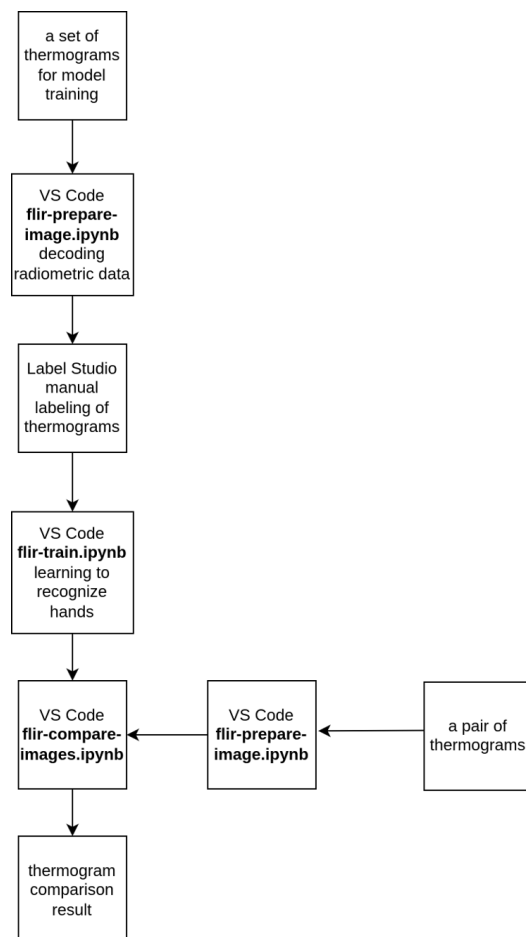


Fig. 1. Radiometric data processing chain diagram (original design).

The data processing chain comprises four sequential stages. In the first stage, the `flir-prepare-image.ipynb` script decodes the radiometric data from the thermal camera's JPEG files and generates clean thermal-visualization images in PNG format. In the second stage, the generated images are labeled manually in the Label Studio tool [9], where a polygonal mask outlining the contour of the hand is created for each thermogram. In the third stage, the `flir-train.ipynb` script trains the YOLOv8n-seg instance segmentation model on the prepared set of labeled images. The fourth stage constitutes the actual use of the system: for each new pair of thermograms, the `flir-compare-images.ipynb` script decodes the radiometric data, runs the trained model to obtain the hand mask, and computes temperature statistics from the region defined by the mask.

The source code of the custom scripts `flir-prepare-image.ipynb`, `flir-train.ipynb`, and `flir-compare-images.ipynb` is available in a public repository at <https://github.com/brzytwa/flir> [12].

Decoding radiometric data from a JPEG file

The FLIR C8 thermal camera used for the tests saves, in the EXIF section of every JPEG file, a raw matrix of 16-bit detector signal values together with a set of calibration constants describing the detector's characteristics.

Data extraction proceeds as follows: the raw matrix of 16-bit detector signal values embedded in the EXIF section is extracted using the ExifTool software [8]. To allow the matrix to be read by the matplotlib library [2], the data are temporarily saved as a TIFF file, which is deleted once it has been read. In parallel, the complete set of Planck calibration constants is read: R_1 , R_2 , B , F , O , along with the environmental parameters: surface emissivity (default $\varepsilon = 0.95$), reflected apparent temperature, and atmospheric temperature.

The conversion of the raw signal into temperature in degrees Celsius is performed using Equation (1), which is an algebraic inversion of the simplified Planck model (describing the radiant intensity of a blackbody as a function of temperature and wavelength) used by the camera manufacturer [5]:

$$T_{\text{obj}} = \frac{B}{\ln\left(\frac{R_1}{R_2(S_{\text{obj}} + O)} + F\right)} - 273,15 \quad (1)$$

where:

- T_{obj} – calculated object surface temperature [$^{\circ}\text{C}$],
- S_{obj} – raw radiometric signal extracted from the image matrix,

- R_1, R_2, B, F, O – calibration constants extracted from the device-specific EXIF metadata.

The notation of the calibration parameters used in Equation (1) (R_1, R_2, B, F, O) follows the metadata-extraction standard of the ExifTool utility and corresponds to the Planck calibration parameters used by the camera manufacturer (denoted in the manufacturer's documentation sometimes as R, J_1, B, F, J) [5].

Because the camera detector records not only the radiation emitted by the measured object but also the radiation reflected from its surface that originates from the surroundings, the signal must be corrected. The signal isolated for the measured object, S_{obj} , is calculated from Equation (2):

$$S_{\text{obj}} = \frac{S_{\text{raw}} - (1 - \varepsilon) S_{\text{refl}}}{\varepsilon} \quad (2)$$

where:

- S_{obj} – radiation signal isolated for the measured object, corrected for the contribution of reflected ambient radiation and surface emissivity.
- S_{raw} – raw detector signal value read directly from the numerical matrix in the range of 0–65535 (2^{16} values) embedded in the EXIF section of the JPEG file.
- ε – emissivity of the measured object's surface (a value of 0.95 was assumed for human skin).
- S_{refl} – signal corresponding to the reflected apparent temperature, calculated using the same Planck equation for a given reference temperature.

Preparation of the training dataset

The machine-learning process consists of two main stages: training, during which the model learns to recognize patterns in the prepared data, and deployment. This second stage, in which the trained model processes new, previously unseen images, is termed inference, and its operation under real-world conditions is referred to as production use.

For automatic segmentation of the hand region in the thermograms, the YOLOv8n-seg model [13] was selected — a variant of the YOLO (You Only Look Once) architecture optimized for instance segmentation, characterized by high processing speed while maintaining good accuracy. Preparing a dedicated training dataset was an essential prerequisite for training the model on this specific type of thermal image.

A key design decision was to label the hand not on the original JPEG images from the camera — which contain graphics overlaid by the camera software (scale bar, manufacturer logo, crosshair, temperature

values) — but on clean thermal-visualization images generated from the temperature matrix using the matplotlib inferno colormap, a color palette that renders low temperatures as dark colors (violet, black) and high temperatures as bright colors (yellow, white), passing through shades of red and orange.

This approach ensures consistency between the training data and the input data during inference: in production use, the YOLOv8n-seg model always receives as input an image generated by the same procedure, which eliminates the risk of misdetection caused by the graphics overlaid by the camera software that are present in the original file.

The dataset comprised a total of 52 thermal images, deliberately covering three characteristic thermal states of the hand:

- warmed hand — a high, uniform surface temperature, with yellow-orange colors dominating in the visualization based on the inferno colormap,
- hand at room temperature — a moderate temperature gradient, typical of resting conditions,
- cooled hand — a distinct thermal gradient between the warm metacarpus and the cold fingers, with violet colors dominating in the fingertip region in the visualization based on the inferno colormap.

To obtain varied thermographic images of the hand surface, the hand was immersed in a vessel of water at a temperature of +5°C (cooling) or +40°C (warming). All 52 training images were obtained from a single participant, which was deemed sufficient at the present stage of validating the measurement method. The immersion time was deliberately varied, which made it possible to obtain different degrees of cooling or warming of the hand and thereby to broaden the range of thermal gradients represented in the training set. After the hand was removed from the vessel, it was placed on a flat surface and photographed with the thermal camera. The photographs deliberately covered various positions of the fingers relative to one another — from fingers held together to widely spread apart — as well as different capture distances and angles, in order to increase the geometric diversity of the training set and improve the model's ability to segment the hand correctly under conditions representative of real-world use.

The thermal diversity of the dataset was essential, because the YOLOv8n-seg model learns to recognize the hand based on the color pattern produced by the inferno colormap — not on the basis of pure shape geometry. Without a representation of cold hands in

the training data, the model was unable to segment the fingertips correctly, because their color blends visually into the background.

Labeling was performed manually using Label Studio [9] and polygonal masks. The polygonal mask consisted of a closed polygon formed by manually adding successive vertices along the contour of the hand. The dataset was split at an 80/20 ratio into training (41 images) and validation (11 images) subsets. Fig. 2 shows an example view of a hand with a manually created mask in the Label Studio tool.

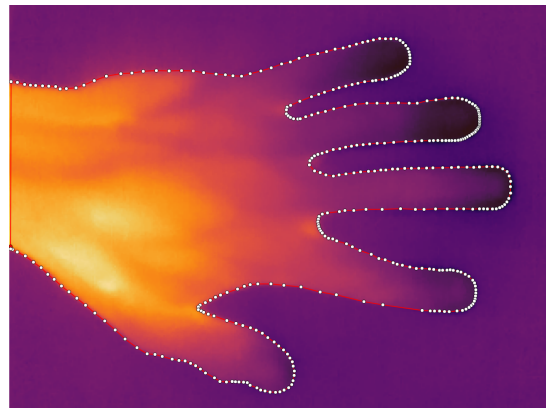


Fig. 2. Polygon mask of the hand contour created using the Label Studio tool (original design).

Label Studio is an open-source data-labeling tool designed for preparing training sets for machine-learning models. It enables manual and semi-automatic annotation of data of various types — images, text, audio, and video — offering a broad selection of labeling tools, including rectangles, polygons, semantic masks, and key points.

Training the YOLOv8n-seg segmentation model

The machine-learning process consists of two main stages: training, during which the model learns to recognize patterns in the prepared data, and inference, i.e., the processing of new, previously unseen images.

For instance segmentation, the YOLOv8n-seg model was initialized with weights pretrained on the COCO dataset and then fine-tuned on a set of 52 hand thermograms, with the validation subset used to monitor the training process. Training parameters:

- training subset: 41 images,
- validation subset: 11 images,
- number of epochs (passes through the entire dataset): 100,
- input image size: 640 × 640 px,
- batch size: 4,
- data augmentation: enabled (rotations of ±15°,

- mirror reflections, changes in saturation and brightness),
- computing device: NVIDIA GeForce GTX 1660 Ti (6 GB VRAM),
- environment: Visual Studio Code, Python 3.14, PyTorch 2.10, CUDA, Linux 6.19.10-200.fc43.x86_64.

Owing to the small size of the available dataset (52 images), a two-way split into training and validation subsets (80%/20%) was applied, forgoing the separation of a third, test subset. The separation of a dedicated test set — a formal requirement of rigorous evaluation — will be introduced in subsequent iterations, together with the expansion of the dataset.

The YOLOv8n-seg model performs two tasks simultaneously: object detection, which involves determining the bounding box around the hand, and instance segmentation, that is, the precise determination of the contour and area of an individual object — in this case the hand — at the pixel level. Both tasks are evaluated separately within the Ultralytics evaluation pipeline [13] using the mAP50 and mAP50-95 (Mean Average Precision) metrics, computed in accordance with the MS COCO standard evaluation protocol [10]. Both metrics are

based on the IoU (Intersection over Union) index, the conceptual basis of which is discussed in [7]. It measures the degree of overlap between the mask predicted by the model and the reference mask: a value of 1.0 denotes a perfect match and 0.0 indicates no overlap. mAP50 is calculated at an IoU threshold of 0.50 — the model considers a detection correct if its mask overlaps the reference by at least 50%. mAP50-95 is a more rigorous measure, averaged over IoU thresholds from 0.50 to 0.95 in steps of 0.05. In the application described in this work, only the quality of the segmentation mask matters, because it is this mask that defines the region from which the temperature statistics are extracted. It is worth noting that mAP50-95 for segmentation (0.944) is higher than for detection (0.812); this results from the characteristic shape of the hand: the bounding box necessarily encompasses the spaces between the fingers, lowering the geometric precision of detection, whereas the segmentation mask accurately reproduces the object's contour.

The characteristics of the dataset (size, division into subsets) and the performance metrics of the trained YOLOv8n-seg model — the mAP50 and mAP50-95 metrics separately for detection and segmentation, as well as the training time — are presented in Table 1.

Tab. 1. Characteristics of the dataset and quality parameters of the YOLOv8n-seg model.

Category	Parameter	Value
Dataset	Number of image pairs (thermal + mask)	52
	Train / Validation split	41/11 (80%/20%)
	Input resolution	640 × 640 px
Detection (Box)	mAP50	0.995
	mAP50-95	0.812
Segmentation (Mask)	mAP50	0.995
	mAP50-95	0.944
Efficiency	Training time (100 epochs)	< 3 min

The thermal-visualization images were generated at a resolution of 320 × 240 px, matching that of the IR detector of the FLIR C8 camera, which ensures full spatial correspondence between the image and the temperature matrix — each image pixel corresponds to exactly one temperature measurement. The YOLOv8n-seg architecture requires a square input image of 640 × 640 px, so before the image is input to the network, automatic resizing with padding (letterboxing) is applied: the image is proportionally enlarged to 640 × 480 px, and the missing areas are filled with neutral bars. This operation introduces no geometric distortion and is performed identically during training and inference, which ensures the consistency of the two stages.

Extraction of temperature statistics from the mask region

During inference, the following steps are carried out for each analyzed image:

1. Extraction of the temperature matrix from the JPEG file.
2. Rendering the matrix as a PNG image using the inferno colormap, identically to the training-data preparation process.
3. Running the YOLOv8n-seg model on the generated PNG file — obtaining a binary mask of the hand region.
4. Scaling the mask to the required size of the temperature matrix (if the resolutions differ).

5. Cleaning the mask: filling closed internal holes using the `binary_fill_holes` function (scipy.ndimage library) and retaining only the largest connected region (elimination of detached artifacts).
6. Extraction of temperature pixels solely from the mask region and calculation of statistics: $T_{\min}$, T_{P5} , $T_{\max}$, T_{P95} , T_{mean} , T_{median} , T_{std} .

Capabilities of the dedicated software approach

Custom analytical tools, in contrast to manufacturers' closed software, make it possible to do the following:

1. Automate the comparison of thermograms. The ability to implement algorithms that detect subtle thermal changes between measurement sessions of the same person. The key comparative metrics are the differences between sessions in the temperature percentiles (T_{P5} , T_{P95}), mean, median, and standard deviation of the hand region.
2. Integration with machine-learning models. The architecture of the data processing chain allows the segmentation module to be swapped freely — instead of YOLOv8n-seg, other architectures can be used, e.g., SAM (Segment Anything Model) or models designed for medical thermography. The Label Studio tool used for labeling also supports semi-automatic, model-assisted labeling, which speeds up the expansion

of the dataset.

3. Full reproducibility of results. All calibration parameters (Planck constants, emissivity, ambient temperature) are read from the EXIF metadata of each image and saved explicitly together with the results. This makes it possible to fully reconstruct the calculations at any time.
4. Technological independence. The ability to combine data from cameras of different manufacturers within a single, coherent analytical ecosystem. The entire technology stack is based on open-source programs and libraries (ExifTool, matplotlib, numpy, scipy, Python, ultralytics), which are not subject to vendor-imposed end-of-life constraints or changes in licensing policy.
5. Batch processing. Processing of entire folders of images in batch mode with automatic generation of reports in CSV format, without operator intervention for each individual image.

RESULTS

Demonstration results

To verify the operation of the data processing chain, a demonstration comparison was conducted between two measurement sessions of the same person, representing markedly different thermal states of the hand: the hand at rest (image A) and the cooled hand (image B). The results are presented in Table 2, and the thermal visualization together with the generated segmentation masks in Fig. 3.

Tab. 2. Hand surface temperature comparison between Image A and Image B.

Parameter	Image A	Image B	$\Delta T (B - A)$
$T_{\min}$	22.6 °C	17.4 °C	-5.2 °C
T_{P5}	25.4 °C	19.1 °C	-6.3 °C
$T_{\max}$	31.6 °C	30.2 °C	-1.5 °C
T_{P95}	30.8 °C	25.0 °C	-5.9 °C
T_{mean}	28.8 °C	21.4 °C	-7.4 °C
T_{median}	29.2 °C	21.2 °C	-8.0 °C
T_{std}	± 1.6 °C	± 1.9 °C	—
YOLO Confidence	0.974	0.964	—
Pixels	32632	31666	—

In addition to the extreme values, the set of reported parameters included ($T_{\min}$, $T_{\max}$), with percentiles (T_{P5} , T_{P95}), as well as measures of central tendency (T_{mean} , T_{median}) and variability (T_{std}). The choice of the T_{P5} and T_{P95} percentiles in addition to the extreme values $T_{\min}$ and $T_{\max}$ is justified by their robustness to artifacts: a single pixel at the edge of the mask or in a region of glare can artificially inflate $T_{\max}$ or deflate $T_{\min}$, failing to reflect the actual temperature distribution of the hand. The

T_{P5} percentile describes the temperature above which only 5% of the pixels lie — thus eliminating extreme values while preserving information about the upper thermal range of the hand. Analogously, T_{P95} characterizes the lower thermal range. The advantage of this approach is clearly visible in the data: $T_{\max}$ differs between the sessions by only 1.5°C, whereas T_{P95} shows a difference of 5.9°C, which far better reflects the actual cooling of the hand surface.

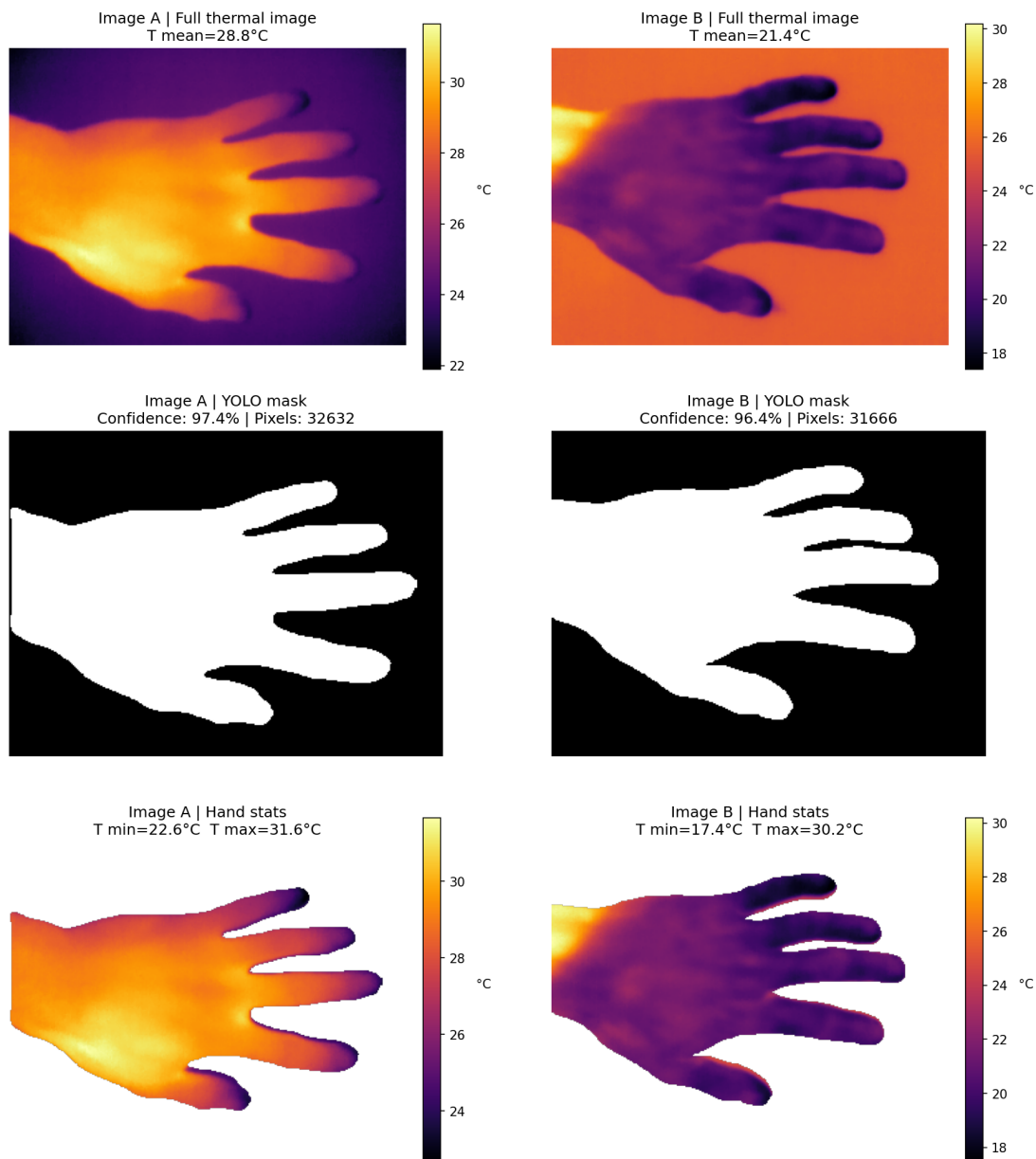


Fig. 3. Thermal image processing stages: from input thermogram (top), through automated hand segmentation (middle), to temperature statistics extraction (bottom) (original design).

T_{mean} and T_{median} yield similar values in both sessions (28.8°C vs. 29.2°C for image A and 21.4°C vs. 21.2°C for image B, respectively), which indicates an approximately symmetric temperature distribution and the absence of strong artifacts. The median, however, serves as an additional safeguard — as a positional statistic, it is robust to the mask-edge artifacts that may slightly bias the mean.

T_{std} increased from 1.6°C to 1.9°C in session B. This change is quantitatively small but directionally consistent with physiological expectations: a cooled hand should exhibit a larger temperature gradient between the warm metacarpus and the cold

fingertips. More informative are the differences in T_{mean} (-7.4°C) and TP5 (-6.3°C), which attest to a significant reduction in the temperature of the entire hand surface between sessions. The detection confidence of YOLOv8n-seg remained high in both cases (0.974 and 0.964), and the number of mask pixels was similar, which confirms the reproducibility of the segmentation regardless of the thermal state of the hand.

DISCUSSION

Limitations and directions for further work

The current implementation constitutes a proof of concept and has a number of limitations that need to be addressed in subsequent iterations:

1. Small training dataset. A set of 52 images is sufficient for demonstrating the concept but insufficient for clinical deployment. Expansion of the set should include different individuals, different camera-to-hand distances, and images of both the back and the palm of the hand.
2. Absence of a separate test set. In the current implementation, the dataset was split only into training and validation subsets (80/20). A split into three subsets — training, validation, and test — is a formal requirement of rigorous model evaluation: the validation set is used to monitor the training process and indirectly influences the choice of hyperparameters, which means that the reported metrics may slightly overestimate the model's actual ability to generalize to new data. This limitation stems from the small size of the available dataset — with 52 images, separating a third, test subset would substantially reduce the amount of training data. At the present stage, which constitutes a preliminary, proof-of-concept study, the adopted split was deemed sufficient. Expanding the dataset in subsequent iterations will make it possible to introduce a full three-way split.
3. Dependence on the colormap. The model learns a visual pattern specific to the inferno colormap. Changing the colormap in the rendering parameters would require retraining the model or expanding the dataset with images that use different colormaps. In subsequent iterations, it is worth considering training on normalized single-channel images. In contrast to a standard RGB color image — in which each pixel is described by three numbers denoting the intensity of the red, green, and blue components — a single-channel image stores only one value for each pixel. In the case of thermal data, this value would correspond directly to the measured temperature: a dark pixel would denote a low temperature and a bright one a high temperature, without the mediation of any color palette. Instead of rendering the temperature matrix through matplotlib into an RGB image, where the pixel color depends on the inferno palette, the model would receive raw thermal information at its input. Such an approach would make the model independent of the matplotlib library and the inferno palette and could, at the same time, improve its ability to generalize to data from other manufacturers' cameras. It does, however, require training the model from scratch with a suitably modified network-input configuration — YOLOv8n-seg by default expects three-channel (RGB) images, so it would be necessary to change the number of input channels from three to one.
4. Absence of a standardized measurement protocol. The reliability of comparisons between sessions depends on the reproducibility of the measurement conditions: camera distance, imaging angle, ambient temperature, and skin emissivity [3]. Developing a formal measurement protocol is an essential step before applying the method in scientific research.
5. Absence of clinical validation. The thermometric results were not compared with reference methods of temperature measurement. Such validation is necessary before the method is applied in any diagnostic context.

CONCLUSIONS

The data processing chain presented here demonstrates that it is possible to build a functional, open system with a fully documented and reproducible computational process, intended for the thermographic analysis of the human hand using a thermal camera and open-source tools [6]. The approach based on segmentation with the YOLOv8n-seg model allows the hand region to be determined automatically, without manual operator intervention [1], which enables high-throughput analysis of large sets of thermograms. The trained model achieved high segmentation accuracy ($mAP_{50-95} = 0.944$) on a dataset covering three thermal states of the hand, which confirms the effectiveness of the adopted approach to preparing the training data. The demonstration results indicate the effectiveness of the method in detecting temperature differences on the order of several degrees Celsius between measurement sessions, which constitutes a promising point of departure for further applied research.

ACKNOWLEDGMENTS

The author thanks Mariusz Krej of the Military Institute of Aviation Medicine for his patient and practical introduction to the field of machine learning, which contributed substantially to the completion of this work.

AUTHORS' DECLARATION

Concept and study design: Marcin Piotrowski. **Data Collection:** Marcin Piotrowski. **Software source code:** Marcin Piotrowski. **Graphical presentation of results:** Marcin Piotrowski. **Manuscript preparation:** Marcin Piotrowski. **Revision of the manuscript following review:** Marcin Piotrowski. **Approval of the final version:** Marcin Piotrowski. The author declares that there is no conflict of interest.

REFERENCES

1. Belard A, Buchman T, Forsberg J, Potter BK, Dente CJ, Kirk A, et al. A Systematic Review of YOLO-Based Object Detection in Medical Imaging: Advances, Challenges, and Future Directions. *Comput Mater Contin.* 2025; 85(2):225–303.
2. Droettboom M. Matplotlib [Internet]. [cited 10 April 2026]. Available from: <https://matplotlib.org/>
3. Fernández-Cuevas I, Bouzas Marins JC, Arnáiz Lastras J, Gómez Carmona PM, Piñonosa Cano S, García-Concepción MÁ, et al. Influence of infrared camera resolution and operator reproducibility in the assessment of skin temperature of the hands. *Case Stud Therm Eng.* 2025; 74:106956.
4. FLIR. Flir C8 Compact thermal imaging camera [Internet]. FLIR; [cited 8 April 2026]. Available from: <https://www.flir.com/products/c8/>
5. FLIR. FLIR Cameras - Temperature Measurement Formula [Internet]. 2025. Available from: https://flir.custhelp.com/app/answers/detail/a_id/3321/_flir-cameras—temperature-measurement-formula
6. Garduno-Ramon CE, Cruz-Albarran IA, Garduño-Ramón MA, Morales-Hernandez LA. Automatic segmentation of regions of interest in thermal images in the facial and hand area. In: 2025 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA). 2025. p. 1–6.
7. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 3rd Edition Intelligent Systems (3rd Edition). 2022.
8. Harvey P. ExifTool [Internet]. [cited 8 April 2026]. Available from: <https://exiftool.org/>
9. Human Signal. Label Studio [Internet]. [cited 8 April 2026]. Available from: <https://labelstud.io>
10. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: Common Objects in Context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T, editors. *Computer Vision – ECCV 2014*. Cham: Springer International Publishing; 2014. p. 740–55.
11. Nowakowski AZ, Kaczmarek M. Artificial Intelligence in IR Thermal Imaging and Sensing for Medical Applications. *Sensors.* 2025; 25(3):891. <https://doi.org/10.3390/s2503089>
12. Piotrowski M. flir-prepare-image.ipynb, flir-train.ipynb, flir-compare-images.ipynb [Internet]. 2026 [cited 8 April 2026]. Available from: <https://github.com/brzytwa/flir>
13. Ultralytics. YOLOv8n-seg [Internet]. [cited 8 April 2026]. Available from: <https://www.ultralytics.com>