Journal of Computational Science 63 (2022) 101760

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Centroid based person detection using pixelwise prediction of the position

Petr Dolezel [a], Pavel Skrabanek [b,*], Dominik Stursa [a], Bruno Baruque Zanon [c], Hector Cogollos Adrian [c], Pavel Kryda [d]

[a] University of Pardubice, Faculty of Electrical Engineering and Informatics, Studentska 95, 532 10 Pardubice, Czech Republic
[b] Brno University of Technology, Institute of Automation and Computer Science, Technicka 2896/2, 601 90 Brno, Czech Republic
[c] University of Burgos, Edificio A, Avda. Cantabria s/n, 09006 Burgos, Spain
[d] Mikroelektronika spol. s r. o., Draby 849, 566 01 Vysoke Myto, Czech Republic

ARTICLE INFO

Keywords:
Person detection
Fully convolutional networks
Performance measure
Edge computing
Computer vision

ABSTRACT

Implementations of person detection in tracking and counting systems tend towards processing of orthogonally captured images on edge computing devices. The ellipse-like shape of heads in orthogonally captured images inspired us to predict head centroids to determine positions of persons in images. We predict the centroids using a fully convolutional network (FCN). We combine the FCN with simple image processing operations to ensure fast inference of the detector. We experiment with the size of the FCN output to further decrease the inference time. We compare the proposed centroid-based detector with bounding box-based detectors on head detection task in terms of the inference time and the detection performance. We propose a performance measure which allows quantitative comparison of the two detection approaches. For the training and evaluation of the detectors, we form original datasets of 8000 annotated images, which are characterized by high variability in terms of lighting conditions, background, image quality, and elevation profile of scenes. We propose an approach which allows simultaneous annotation of the images for both bounding box-based and centroid-based detection. The centroid-based detector shows the best detection performance while keeping edge computing standards.

1. Introduction

Automatic tracking and counting of persons using computer vision systems is an important task in surveillance of public and private places [1], and specifically in public transport [2]. Various imaging technologies including radar sensors [3], laser scanners [4], 3D laser scanners [5], infra-red sensors [6], and cameras operating in the visible spectrum of light [2] can be used for this purpose. Selection of the imaging technology is mainly driven by economic interests of manufacturers (low price of a final solution). Cameras operating in the visible spectrum of light are preferred in practice. However, their utilization often faces legal obstacles which must be reflected in final solutions. For example, orthogonal scanning of a scene (camera placed above a scene, looking directly down on the scene) is preferred at public places as it prevents unwanted identification of individuals from their faces (Fig. 1) [2].

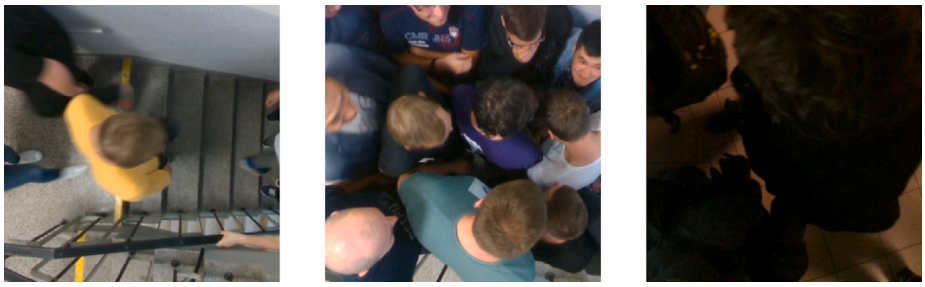
One of the key operations performed within person tracking and counting systems is detection of persons in images [2]. The person detection is a variant of a computer vision task known as object detection. It comprises localization and class recognition of objects (individuals) within the images. The localization is the task of determining the position of an object in the image. The object class recognition is a classification problem where a class is assigned to each localized object.

Object localization is usually addressed as prediction of a bounding box around a detected object [7]. However, other closed plane figures such as octagon [8] and circle [9] can be used as well. Such an approach gives us information about the approximate position and dimensions of the object. Another option is localization of objects using semantic or instance segmentation [10]. Segmentation maps give us an accurate notion of the shape and position of each recognized object. The knowledge of shapes allows us to implement various localization methods, including the bounding figure-based localization. Information about the position of one characteristic point of an object is sufficient for applications such as object tracking and object counting. Therefore, we can reduce the localization into an estimation of its centroid position [11].

State-of-the-art bounding figure-based object detection systems, as well as instance segmentation systems, rely on deep convolutional networks (deep ConvNets). Deep ConvNets typically consist of convolutional and pooling layers which convert an input image into a series of three-dimensional feature maps. The maps can be transformed into

* Corresponding author.
E-mail address: pavel.skrabanek@vut.cz (P. Skrabanek).

https://doi.org/10.1016/j.jocs.2022.101760
Received 21 September 2021; Received in revised form 10 June 2022; Accepted 28 June 2022
Available online 6 July 2022



(a) (b) (c)

Fig. 1. Examples of orthogonally acquired images for person detection. An image with a small number of persons where the persons are sharply differentiated from the background (a) promises a higher probability of correct person detection than a crowded image (b). Incomplete heads and variable distances of the heads from a camera lens (b) also complicate the head detection. In extreme cases, a head can be very close to the lens and cover a large part of the image area (c). If the colour of the head blends significantly with other objects in the scene and the lack of light makes the overall image unclear (c), the head detection becomes challenging.

a desired output by implementing a fully connected layer at the end of a network. A network which does not contain any fully connected layer is referred to as the fully convolutional network (FCN). As FCNs consist purely of convolutional and pooling layers, they are faster than ConvNets with fully connected layers. This property of FCNs is advantageous for dense pixelwise prediction tasks such as instance [10] and semantic segmentation [12], where outputs of the networks are generally three-dimensional cubes.

The bounding figure-based object detection methods can either predict bounding figures directly (one-stage methods) [7,8,13–15] or they can generate regions of interests at first stage. This is followed by sending the region proposals to the second stage for object classification and bounding figure regression (two-stage methods) [16–20]. Two-stage methods typically reach higher accuracy rates but are slower than one-stage methods. The well-established one-stage methods such as YOLO [7] or single shot multibox detector (SSD) [13] use fix-sized anchor boxes as region candidates. The main drawbacks of the anchor-based methods are the need of ad-hoc heuristics (determining the number and dimension of anchor boxes), and the large set of anchor boxes to be evaluated, which slows down the training of models [14]. These drawbacks are overcome by key-point based methods such as ExtremeNet [8], CornerNet [14], and CenterNet [15]. CornerNet predicts two key-points for each object of interest: the top-left and the bottom-right corners of a rectangular bounding box. CenterNet improves the CornerNet idea by adding a centre of gravity of the object to the prediction of the bounding box coordinates. ExtremeNet predicts coordinates of four extreme points and of one centre point for each object. These five coordinates determine an irregular octagon which determines the position of a detected object in the image.

Dense pixelwise prediction FCNs transform an input image into a map (segmentation map [10], saliency map [21], optical flow map [22], etc.). They consist of an encoder (contracting path) and a decoder module (expansive path), respectively. Convolutional and pooling layers of the encoder module gradually reduce the resolution of feature maps (pooling layers) while learning semantic information (convolutional layers). The down-sampling ensured by the pooling layers increases local receptive fields of neurons in deeper convolutional layers, thus allowing the learning of more complex features. Once dilated (atrous) convolution layers are used instead of pooling layers, the receptive fields are increased without decreasing the resolution. Note that the processing of high-resolution feature maps results in high time and space-complexities of such modified networks [23]. The decoder module, which consists of inverse layers (up-convolution and up-sampling layers), ensures recovering of output spatial dimensions. The decoder can be designed either as a mirror of the encoder [24] or it can be asymmetric to the encoder [12].

The state-of-the-art dense pixelwise prediction FCNs are directed acyclic graphs with skip connections to transfer pooling indices (SegNet [25]) or feature maps from the encoder to the decoder. The combination of feature maps from the encoder with feature maps produced by corresponding up-convolution or up-sampling layers can be ensured, for example, by concatenative skip connections (U-Net [26]), by attention gates (attention U-Net [27]), by adding an extra full-resolution stream (full-resolution residual networks (FRRNs) [28]), and by improving skip connections with higher number or complicated convolutional units (squeeze U-Net [29], gated feedback refinement network (G-FRNet) [30], global convolutional network [31] (GCN)).

The bounding figure-based object detectors, likewise dense pixelwise prediction FCNs, are trained end-to-end on a set of annotated images. Annotation of a sufficiently representative dataset is always time consuming. In the case of the bounding box-based object detection, an annotator (a domain expert) draws a rectangular boundary around each object of interest in each image in the dataset, and assigns them class labels [7]. The annotation of images for dense pixelwise prediction tasks is even more challenging. For example, in the case of the instance segmentation, the annotator creates a segmentation map for each image in the dataset. He/she must assign to each element of the map a value corresponding to the class of the object which is associated with the element.

Performance measures such as intersection over union and generalized intersection over union are commonly used for evaluation of bounding box-based object detectors [32]. These measures take into account areas defined by ground truth bounding boxes and bounding boxes predicted by an evaluated detector.

The practical implementation of tracking and counting systems tends towards data processing on edge computing devices. Their limited computing power requires the use of time efficient and yet accurate object detectors. Herein, we propose an approach ensuring fast and precise detection of persons in orthogonally captured images. We assume that the position of an object centroid is sufficient for person counting and tracking. Instead of bounding figure prediction, we predict object centroids using localization maps. We expect that generation of smaller localization maps will result in smaller time complexity of the centroid-based detector, while keeping detection performance at the level of its full resolution variant. To confirm our hypothesis, we propose two dense pixelwise prediction FCNs: the first generates localization maps of spatial resolution equal to spatial resolution of input images, and the second produces maps of quarter resolution. We compare the proposed centroid-based person detector for both map resolutions with YOLO and CenterNet detectors in terms of inference time and detection performance. We choose YOLO as it is a generally accepted baseline for real-time bounding box-based object detection, and CenterNet as it is promising keypoint-based detector with object centroid coordinates as one of its outputs. As our centroid-based object detector predicts only points (coordinates of centroids), the commonly used performance measures cannot be used for its comparison with bounding box-based person detectors. We propose a total localization error $\sum e$ which allows quantitative assessment of detection performance of both the centroid

and the bounding box-based detectors. For the training and evaluation of the detectors, we form new datasets which consist of images with orthogonally captured persons in various scenes and with variable head–lens distances. While annotation of the datasets for bounding box-based detection requires only delamination of bounding boxes, pixel-wise annotation is needed in the case of the proposed detector. To simplify the annotation, we propose an approach which allows simultaneous annotation of the images for both bounding box-based and centroid-based detection.

The key contributions of this article are as follows:

- A centroid based person detection technique in visual data is proposed.
- The technique aims at orthogonal scanning of the scene with variable head–lens distances.
- An efficient approach to transformation of the scene image into a localization map is introduced. The positions and sizes of the persons' heads are coded into gradient ellipses. These ellipses afterwards provide centroid locations of each head in the scene image. A localization map also includes heads that are only partially represented in the scene image.
- A metric suitable for evaluation of both the centroid and bounding box predictions is proposed. This metric includes inaccuracies in determining head positions, as well as false positive and false negative detections.
- New datasets are formed for the training and evaluation of the detectors. 7000 images at seven different places are acquired for the training and validation. Additionally, another 1000 images are captured at a different place from the previous locations in order to test the generalization capabilities of the proposed person detection technique.

2. Materials and methods

2.1. Bounding box-based object detection

Let an image I contain n objects of k recognized classes, where $n \in \mathbb{Z}^{0+}$, and $k \in \mathbb{Z}^{+}$. A rectangle boundary with edges parallel to the edges of the image, tightly enveloping the ith recognized object, delimits position and dimensions of the object within I. The ground truth bounding box of the ith object is a 5-tuple $b_i = (\dot{x}_i, \dot{y}_i, w_i, h_i, c_i)$, where $\dot{x}_i$ and $\dot{y}_i$ are x and y coordinates of the left-top rectangle corner respectively, w_i and h_i are width and height of the rectangle respectively, c_i is the class of the object, and $c_i \in \{1, \dots, k\}$.

A bounding box-based object detector predicts for I a set $\hat{B}$ of m bounding box predictions $\hat{b}$, where $m \in \mathbb{Z}^{0+}$. In the case of YOLO, the ith prediction is a 5-tuple

$$\hat{b}_i = (\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i, \hat{c}_i), \tag{1}$$

where $\hat{x}_i$, $\hat{y}_i$, $\hat{w}_i$, $\hat{h}_i$, and $\hat{c}_i$ are predictions of $\dot{x}_i$, $\dot{y}_i$, w_i, h_i, and c_i, respectively. In the case of CenterNet, the ith prediction is a 7-tuple

$$\hat{b}_i = (\hat{\dot{x}}_i, \hat{\dot{y}}_i, \hat{\ddot{x}}_i, \hat{\ddot{y}}_i, \hat{\bar{x}}_i, \hat{\bar{y}}_i, \hat{c}_i), \tag{2}$$

where $\hat{\dot{x}}_i$, $\hat{\dot{y}}_i$, $\hat{\ddot{x}}_i$, $\hat{\ddot{y}}_i$ are predictions of $\ddot{x}_i$, $\ddot{y}_i$, $\bar{x}_i$, $\bar{y}_i$ respectively, $\ddot{x}_i$ and $\ddot{y}_i$ are x and y coordinates of the right-bottom rectangle corner respectively, and $\bar{x}_i$ and $\bar{y}_i$ are x and y coordinates of the object centroid. It holds that $w_i = \ddot{x}_i - \dot{x}_i$ and $h_i = \ddot{y}_i - \dot{y}_i$.

2.2. Centroid-based object detection

In the case of the centroid-based object detection, the position of the ith object in I is given by x and y coordinates of its centroid $\bar{x}_i$ and $\bar{y}_i$, respectively. Let the ground truth centroid of the ith object be the 3-tuple $\gamma_i = (\bar{x}_i, \bar{y}_i, c_i)$, where $c_i \in \{1, \dots, k\}$.

A centroid-based object detector predicts for I a set $\hat{\Gamma}$ of p object centroids $\hat{\gamma}$, where $p \in \mathbb{Z}^{0+}$. The ith centroid prediction is the 3-tuple $\hat{\gamma}_i = (\hat{\bar{x}}_i, \hat{\bar{y}}_i, \hat{c}_i)$, where $\hat{\bar{x}}_i$, $\hat{\bar{y}}_i$, and $\hat{c}_i$ are predictions of $\bar{x}_i$, $\bar{y}_i$, and c_i, respectively.

2.3. Proposed centroid-based object detector

Dense pixelwise prediction FCNs are theoretically capable of transforming an image I of width w_I and height h_I into a three-dimensional centroid map Ξ of width w_Ξ, height h_Ξ and depth k (depth given by the number of recognized classes). Elements of the map are given as

$$\Xi(x, y, c) = \begin{cases} 1, & \text{if a centroid of the } c\text{-th class is at } (x, y) \text{ coordinates,} \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $x \in X_\Xi$, $y \in Y_\Xi$, $c \in C_\Xi$, $X_\Xi = \{0, \dots, w_\Xi - 1\}$, $Y_\Xi = \{0, \dots, h_\Xi - 1\}$, and $C_\Xi = \{1, \dots, k\}$.

The centroid maps Ξ are theoretically the ideal source for a centr oid-based object detector. In such a detector, a dense pixelwise prediction FCN acts as a map generator. The generator must be complemented by a localization module which searches for centroid predictions $\hat{\gamma}$ in map predictions $\hat{\Xi}$. We implement the search, as search for positions of maxima in $\hat{\gamma}$. A set of the p centroid predictions $\hat{\gamma}$ is for the image I given as

$$\hat{\Gamma} = \left(\frac{h_I}{h_\Xi}, \frac{w_I}{w_\Xi}, 1\right) \odot \underset{(x,y,c) \in S_\Xi}{\arg\max} \left\{\hat{\Xi}(x, y, c)\right\}, \tag{4}$$

where $\odot$ denotes an element-wise product, and $S_\Xi = X_\Xi \times Y_\Xi \times C_\Xi$.

Due to the limited approximation capability of real FCN models, values of $\hat{\Xi}$ elements can be assigned incorrectly which can result in false positive and false negative detections, and in incorrect localizations. We expect that the localization performance of the proposed approach can be improved once the mass of objects is considered. Rather than training a pixelwise prediction FCN to predict centroid maps Ξ, we train it to predict three-dimensional localization maps $\mathcal{L}$ of width $w_\mathcal{L}$, height $h_\mathcal{L}$ and depth k. Elements of $\mathcal{L}$ can take any real value from the interval $[0, 1]$. The values correspond to the probability of occurrence of object centroids at corresponding locations. Elements of the map are given as

$$\mathcal{L}(x, y, c) = \begin{cases} \lambda(x, y, c), & \text{for } (x, y) \text{ associated with a } c\text{-th class object,} \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where $\lambda : \mathcal{L} \to (0, 1]$, $\mathcal{L}(x, y, c) = 1$ indicates presence of the centroid of an object of the c-th class at the location (x, y), and the values decrease towards 0 with increasing distances of the elements from their centroids.

To take the advantage of localization maps $\mathcal{L}$, we must transform predictions of localization maps $\hat{\mathcal{L}}$ into predictions of centroid maps $\hat{\Xi}$. Thus, an object detector based on the localization maps must consist of a localization map generator (a pixelwise prediction FCN), a centroid counterpoint module, and the localization module (4), respectively (Fig. 2), where the centroid counterpoint module ensures the transformation of $\hat{\mathcal{L}}$ into $\hat{\Xi}$.

2.3.1. Centroid counterpoint module

The module emphasizes centroids and suppresses false detections by a series of operations. At first, we process each layer of $\hat{\mathcal{L}}$ using a maximum filter with kernel of size $h_K \times w_K$. We get a map $\hat{M}$ with elements

$$\hat{M}(x, y, c) = \max_{(s,t) \in S_{xy}} \{\hat{\mathcal{L}}(s, t, c)\}, \tag{6}$$

where S_{xy} is a set of spatial coordinates in a rectangular sub-window of size $h_K \times w_K$, centred at point (x, y).

We compare $\hat{M}$ and $\hat{\mathcal{L}}$ to highlight local maxima. The result of this operation is a binary map $\hat{M}_1$ with elements

$$\hat{M}_1(x, y, c) = \begin{cases} 1, & \text{if } \hat{M}(x, y, c) = \hat{\mathcal{L}}(x, y, c), \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

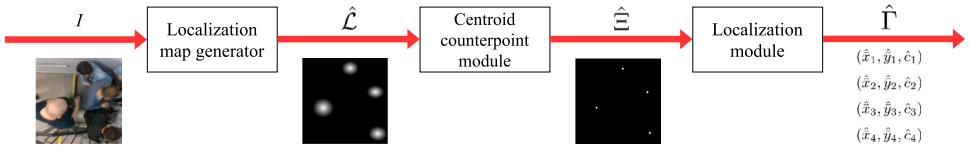


Fig. 2. A pipeline of the centroid-based object detector by head detection. A FCN in the localization map generator transforms the input image I into a localization map prediction $\hat{\mathcal{L}}$, where objects of interest (heads) are represented as gradient ellipses. The centroid counterpoint module converts $\hat{\mathcal{L}}$ into a centroid map prediction $\hat{\Xi}$, where local maxima are highlighted. The localization module identifies and localizes positions of the objects (heads) using the maxima (white spots) and returns a set of centroid predictions $\hat{\Gamma}$.

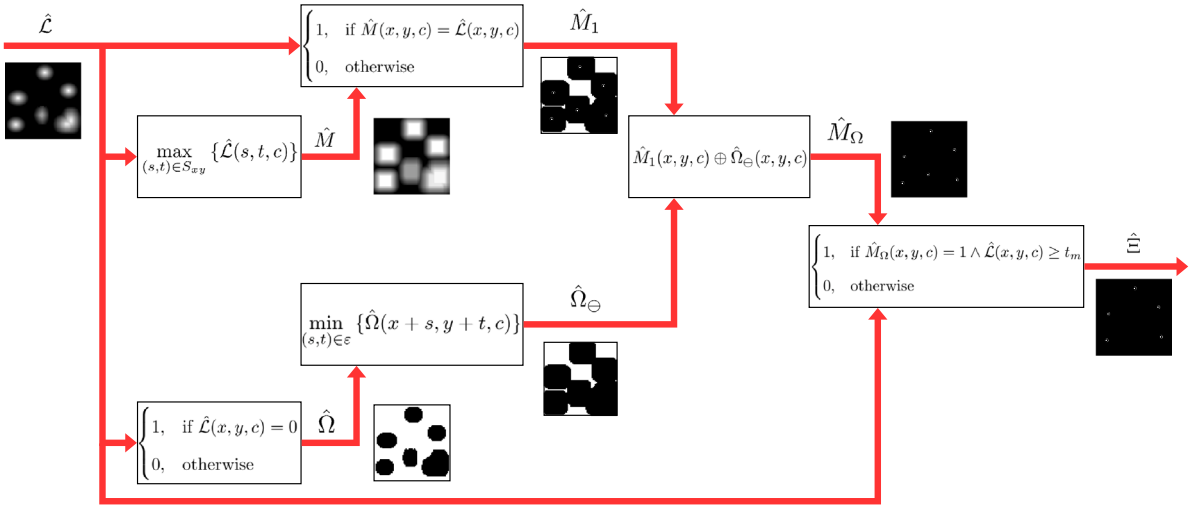


Fig. 3. Pipeline of the centroid counterpoint module by head detection. The predicted localization map $\hat{\mathcal{L}}$ for an image I of five heads contains six continuous areas with nonzero values, where the bottom-right area contains two local maxima, and the maximum of the middle-bottom area is noticeably smaller than one. The exclusive disjunction $\oplus$ of the eroded mask $\hat{\Omega}_\ominus$ and of the binary map $\hat{M}_1$ eliminates the redundance in the bottom-right area. The last operation of the pipeline removes the middle-bottom area which does not correspond to any head.

The map $\hat{M}_1$ contains the centroids as well as local maxima caused by noise of background. To suppress irrelevant regions in the map $\hat{M}_1$, we form a mask $\hat{\Omega}$ with elements

$$\hat{\Omega}(x,y,c) = \begin{cases} 1, & \text{if } \hat{\mathcal{L}}(x,y,c) = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

To suppress artefacts in the map $\hat{M}_1$ caused by the local maximum filter (6), we erode each layer of the mask $\hat{\Omega}$ using a rectangular structuring element ε of size $h_K \times w_K$ with the origin in its centre. We pad the $\hat{\Omega}$ by ones to keep the dimensions of the eroded mask $\hat{\Omega}_\ominus$ the same as dimensions of $\hat{M}_1$. Elements of $\hat{\Omega}_\ominus$ are given as

$$\hat{\Omega}_\ominus(x,y,c) = \min_{(s,t)\in\varepsilon} \{\hat{\Omega}(x+s, y+t, c)\}. \tag{9}$$

Application of the eroded mask $\hat{\Omega}_\ominus$ on $\hat{M}_1$ results in a map $\hat{M}_\Omega$ with elements

$$\hat{M}_\Omega(x,y,c) = \hat{M}_1(x,y,c) \oplus \hat{\Omega}_\ominus(x,y,c), \tag{10}$$

where $\oplus$ denotes exclusive disjunction.

We identify centroids among the maxima highlighted in the map $\hat{M}_\Omega$ by considering their values in the localization map prediction $\hat{\mathcal{L}}$. Each element of the predicted localization map $\hat{\mathcal{L}}$ associated with a centroid must be greater or equal to a threshold value t_m, where $t_m \in (0, 1]$. This operation results in the centroid map prediction $\hat{\Xi}$ with elements

$$\hat{\Xi}(x,y,c) = \begin{cases} 1, & \text{if } \hat{M}_\Omega(x,y,c) = 1 \wedge \hat{\mathcal{L}}(x,y,c) \geq t_m, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The setting of t_m is problem dependent and must reflect the quality of localization map predictions.

We summarize the pipeline of the centroid counterpoint module in Fig. 3.

2.4. Detection of persons in orthogonally captured images

Orthogonal scanning of a scene results in images where individuals are presented by their heads and shoulders. Their sizes are subject to the focal length of a camera lens and to distances of individuals from a pupil of the lens. The distances depend, for example, on heights of the individuals, elevation profile of the scene (Fig. 4), and the camera altitude.

The detection of persons in the orthogonally captured images can be treated as detection of heads and shoulders [33] or as detection of heads [2]. Detection of heads has proven to be specifically accurate on datasets with high variability in the sizes [2]. In both cases, any bounding figure-based object detector can be used for person detection.

In tracking and counting applications, individuals are usually considered to be members of the same class ($k = 1$). Great emphasis is put on correct localization of individuals in images with minimum overlooked individuals and with minimum false detections. Dimensions of persons are not essential unless they predetermine accuracy of the localization. Considering these facts, we propose using the centroid-based object detector for the detection of persons. To ensure high robustness of the person detector, we view the person detection as a problem of determining positions of heads.

2.4.1. Centroid-based person detection

Our goal is to develop a person detector allowing real time detection on devices with a single-board computer architecture. To reduce time complexity of the detector, we consider 288 × 288 pixel (px) images to be inputs of the detector. Given that heads are defined by shape and brightness gradient rather than by colour, we use greyscale images for processing. The only component of the proposed detector which allows optimization of its inference time is the localization map generator. Considering this fact, we propose two FCN topologies to

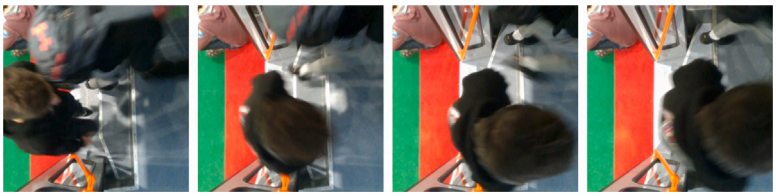


Fig. 4. Influence of scene elevation profile on variability in size, shape and general appearance of individuals in orthogonally captured images. Stairs in the scene cause the change of head size. The rotation of the head affects its shape.

be implemented in the person detector as a generator. We base them on the U-Net architecture [26]. The first topology (full resolution U-Net) generates localization maps of spatial resolution equal to spatial resolution of input images (288 × 288). The second one (reduced U-Net) aims at generation of quarter size localization maps (72 × 72).

The U-Net is a symmetric dense pixelwise prediction FCN (decoder is the mirror of encoder). U-Net modules (UMs) ensure feature extraction at four levels of the network. UM consists of five consecutive operations: convolution (Conv), rectified linear unit (ReLU), dropout (DO), Conv, and ReLU, respectively. Using a short notation, UM can be written as

$$\mathrm{Conv}(h_c \times w_c, f, s_c) \rightarrow \mathrm{ReLU} \rightarrow \mathrm{DO}(d) \rightarrow \mathrm{Conv}(h_c \times w_c, f, s_c) \rightarrow \mathrm{ReLU}, \tag{12}$$

where d is probability of dropout, s_c is stride of convolutional filters, f is the number of the filters, and h_c and w_c are their height and width, respectively. Each UM in the encoder module is followed by max-pooling (MP) with pools of height and width h_p and w_p, respectively, and stride s_p (shortly $\mathrm{MP}(h_p \times w_p, s_p)$). Feature maps produced by UMs in the encoder module are concatenated with feature maps produced in the decoder module. The transfer of the maps from the encoder to the decoder is ensured by skip connections.

We summarize topologies of the full resolution and the reduced U-Nets in Table 1 and Table 2, respectively. The columns outline operations performed within the encoder and decoder modules. The operations are arranged in rows with respect to skip connections. The data flow is symbolized using arrows, where their orientations indicate data flow directions. Simple arrows denote the main flow of data, and double ones symbolize skip connections (skip connections are numbered for clarity). We denote a concatenation of two feature maps as $[\cdot, \cdot]$. For all UMs in both topologies, we use following setting: $h_c = 3, w_c = 3, s_c = 1$ and $d = 0.2$. As the only changing parameter is the number of filters f, we use notation $\mathrm{UM}(f)$ for the description of the topologies. In the full resolution U-Net, up-sampling (US) precedes each UM in the decoder module. In the reduced U-Net, we remove the last two USs. We implement US as the Kronecker product of each input feature map with $h_u \times w_u$ matrix of all ones ($\mathrm{US}(h_u \times w_u)$), where h_u and w_u are height and width of the matrix, respectively. In the full resolution U-Net, feature maps are directly transferred between corresponding parts of the encoder and decoder. In the reduced U-Net, feature maps produced by the first and the second UMs in the encoder are reduced using max-pooling to a quarter and half of their size respectively, before their concatenations with feature maps produced within the decoder module. In both variants, the networks are closed by $\mathrm{Conv}(1 \times 1, k, 1)$ followed by a sigmoid activation function (sig), where k is the number of recognized classes. To achieve the required output dimensions (288 × 288 or 72 × 72 px), we zero-pad inputs of operations, if necessary.

2.4.2. YOLO person detection

We compare the proposed centroid-based person detector with a detector based on YOLOv2 architecture [34]. The YOLO-based person detector expects 288 × 288 px greyscale images at its input, and it returns, for each image, a set $\hat{B}$ of bounding box predictions $\hat{b}$, where the predictions are 5-tuples (1). As our aim is to compare the centroid-based person detector with a similar competitor in terms of inference time, we consider GoogLeNet [35], MobileNet-v2 [36], and SqueezeNet [37] as backbone models of the YOLO-based person detector. All the networks have proven to be successful in various time-critical computer vision applications.

Table 1
Topology of the full resolution U-Net.

Encoder	Decoder
$\downarrow\mathrm{UM}(64)\downarrow\overset{1}{\Rightarrow}$	$[\uparrow,\overset{1}{\Rightarrow}]\rightarrow\mathrm{UM}(64)\rightarrow\mathrm{Conv}(1\times1,k,1)\rightarrow\mathrm{sig}\uparrow$
$\downarrow\mathrm{MP}(2\times2,2)\rightarrow\mathrm{UM}(128)\downarrow\overset{2}{\Rightarrow}$	$[\uparrow,\overset{2}{\Rightarrow}]\rightarrow\mathrm{UM}(128)\rightarrow\mathrm{US}(2\times2)\uparrow$
$\downarrow\mathrm{MP}(2\times2,2)\rightarrow\mathrm{UM}(256)\downarrow\overset{3}{\Rightarrow}$	$[\uparrow,\overset{3}{\Rightarrow}]\rightarrow\mathrm{UM}(256)\rightarrow\mathrm{US}(2\times2)\uparrow$
$\downarrow\mathrm{MP}(2\times2,2)\rightarrow\mathrm{UM}(512)\downarrow\overset{4}{\Rightarrow}$	$[\uparrow,\overset{4}{\Rightarrow}]\rightarrow\mathrm{UM}(512)\rightarrow\mathrm{US}(2\times2)\uparrow$
$\downarrow\mathrm{MP}(2\times2,2)\rightarrow$	$\rightarrow\mathrm{US}(2\times2)\uparrow$

The table summarizes operations performed within the encoder and decoder modules. The operations are arranged in rows with respect to skip connections, where simple arrows denote the main flow of data, and double arrows symbolize skip connections (skip connections are numbered). $\mathrm{UM}(f)$ denotes the U-Net module (12) of f filters. $\mathrm{US}(h_u \times w_u)$ symbolizes up-sampling, where w_u and h_u are up-sampling factors in x and y directions, respectively. We use $\mathrm{MP}(h_p \times w_p, s_p)$ for max-pooling with pools of height and width h_p and w_p, stride s_p. $\mathrm{Conv}(h_c \times w_c, f, s_c)$ represents convolution of f filters of height h_c and width w_c, stride s_c. We use sig for the sigmoid activation function.

Table 2
Topology of the reduced U-Net.

Encoder	Decoder
$\downarrow\mathrm{UM}(64)\downarrow\rightarrow\mathrm{MP}(4\times4,4)\overset{1}{\Rightarrow}$	$[\uparrow,\overset{1}{\Rightarrow}]\rightarrow\mathrm{UM}(64)\rightarrow\mathrm{Conv}(1\times1,k,1)\rightarrow\mathrm{sig}\uparrow$
$\downarrow\mathrm{MP}(2\times2,2)\rightarrow\mathrm{UM}(128)\downarrow\rightarrow\mathrm{MP}(4\times4,2)\overset{2}{\Rightarrow}$	$[\uparrow,\overset{2}{\Rightarrow}]\rightarrow\mathrm{UM}(128)\uparrow$
$\downarrow\mathrm{MP}(2\times2,2)\rightarrow\mathrm{UM}(256)\downarrow\overset{3}{\Rightarrow}$	$[\uparrow,\overset{3}{\Rightarrow}]\rightarrow\mathrm{UM}(256)\rightarrow\uparrow$
$\downarrow\mathrm{MP}(2\times2,2)\rightarrow\mathrm{UM}(512)\downarrow\overset{4}{\Rightarrow}$	$[\uparrow,\overset{4}{\Rightarrow}]\rightarrow\mathrm{UM}(512)\rightarrow\mathrm{US}(2\times2)\uparrow$
$\downarrow\mathrm{MP}(2\times2,2)\rightarrow$	$\rightarrow\mathrm{US}(2\times2)\uparrow$

Table layout and the abbreviations are identical with Table 1.

2.4.3. CenterNet person detection

The second competitor of the proposed detector is CenterNet. The inputs of the CenterNet-based detector are 288 × 288 px greyscale images. The outputs are sets $\hat{B}$ of bounding box predictions $\hat{b}$, where the predictions are 7-tuples (2). To remain consistent with the original paper [15], we use ResNet101 as the backbone. We also consider EfficientDET D0 [38], which promises high computational efficiency.

2.5. Datasets

2.5.1. Data acquisition

Quality of datasets predetermines performance of deep ConvNet-based computer vision systems. To ensure robustness of the person detectors in the intended setting, we collect data in diverse environments which include staircases, corridors, and entries into means of transport. We capture video streams with the RealSense camera D435 orthogonally placed above walking persons at eight different locations. The walking persons are adults with and without headgear. The head–lens distance varies between 25 and 100 cm depending on the environment and situation in the scene. This setting results in a

Fig. 5. Samples of cut-outs from individual locations 1–8. Backgrounds, lighting conditions, and elevation profiles vary among the scenes. The bottom-right image shows the location which we use by forming the blind test dataset D_B. The remaining locations are included in the training and test datasets D_T and D_E.

large variance in the size of the heads and their sharpness. The lighting conditions differ among the experiments.

2.5.2. Dataset creation

We extract frames from the captured videos to create a set of 8-bit RGB images. From the frames of seven locations, we cut out 7000 square images with up to nine persons. We randomly split the set of images in the ratio 6:1 to create training and test datasets D_T and D_E, respectively. From the eighth location, we form a blind test dataset D_B of 1000 square image crops (Fig. 5). As these images are captured under different lighting conditions at a different place from the previous locations, D_B allows testing the generalization capabilities of the detectors. We resize images in all datasets to 288 × 288 px.

2.5.3. Data analysis

The images capture persons of various heights at locations of various elevation profile. Both these aspects contribute to a high variability in sizes of heads in the images. The shape of heads is elliptical (Fig. 5). The smallest width and height of the ellipses is about 20 px. The largest ellipse dimension reaches 200 px. The number of heads in a scene varies from 0 to 10. Some images contain incomplete heads (persons near edges of the images in Fig. 5). As the camera has fixed focus, some of the heads are blurred.

2.5.4. Image annotation

Training and evaluation of the proposed centroid-based person detector requires extension of the image datasets with localization maps and ground truth centroids. We must create a localization map for each image, and we must assign a real value from the interval $[0, 1]$ to each pixel of each localization map, where the non-zero values must be associated with objects of interests (heads).

To simplify the annotation process, it is reasonable to approximate the positions of objects in the maps using an appropriate geometric shape. Considering the elliptic shape of heads, we approximate the heads by gradient ellipses. We consider ellipse centroids to be identical with head centroids and ellipse circumferences to be borders between heads and background. We draw rectangles tightly enveloping complete heads within an image I. To ensure correct approximation of protruding heads, we estimate shapes of rectangles so that they include visible and invisible parts of the heads (Fig. 6). In such a way, we create for I a set $\mathfrak{R}$ of n rectangles $\mathfrak{r}$. The ith rectangle is an ordered four tuple $\mathfrak{r}_i = (\mathfrak{x}_i, \mathfrak{y}_i, \mathfrak{w}_i, \mathfrak{h}_i)$, where $\mathfrak{x}_i$, and $\mathfrak{y}_i$ are x and y coordinates of the left top rectangle corner respectively, and $\mathfrak{w}_i$, and $\mathfrak{h}_i$ are width and height of the rectangle respectively. The ellipse defining border of the ith head in the image I is given as

$$\frac{4(x-\mathfrak{x}_i)^2}{\mathfrak{w}_i^2} + \frac{4(y-\mathfrak{y}_i)^2}{\mathfrak{h}_i^2} = 1. \tag{13}$$

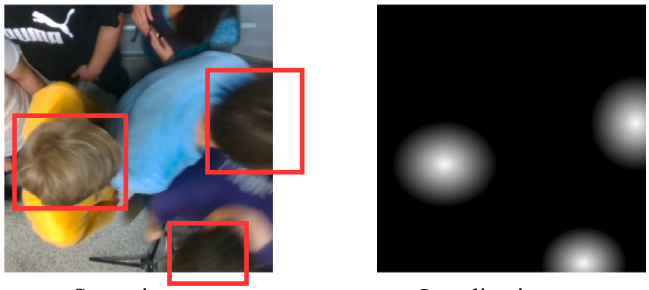


Fig. 6. Annotation of an image. For each head in the scene image I (left), an annotator draws a rectangle tightly enveloping the head (red rectangles). He/she estimates shapes of rectangles for the protruding heads. The rectangles define ground truth centroids and bounding boxes of the heads, as well as the localization map $\mathcal{L}$ for I. We approximate the heads by gradient ellipses that are delimited by the rectangles. Filling the ellipse areas with non-zero values results in $\mathcal{L}$, where head centroids are represented by values close to 1, background by 0, and values in the ellipse areas linearly decrease to 0 with increasing distances from the centroids (right).

We use the canonical ellipse equation (13) to define a piecewise linear function which assigns real values from the interval $(0, 1]$ to elements of $\mathcal{L}$ associated with areas of the ellipses. When applied to (5), we get

$$\mathcal{L}(x, y, 1) = \begin{cases} 1 - \sqrt{\frac{4(x-\mathfrak{x}_i)^2}{\mathfrak{w}_i^2} + \frac{4(y-\mathfrak{y}_i)^2}{\mathfrak{h}_i^2}}, & \text{for } (x, y) \in S_{\mathfrak{r}_i}, \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

where $S_{\mathfrak{r}_i}$ is a set of spatial coordinates in an ellipse sub-window defined by $\mathfrak{r}_i \in \mathfrak{R}$. The ground truth centroid of the ith head $\gamma_i = (\bar{x}_i, \bar{y}_i, c_i)$ is given as

$$\bar{x}_i = \mathfrak{x}_i + 0.5\mathfrak{w}_i, \quad \bar{y}_i = \mathfrak{y}_i + 0.5\mathfrak{h}_i, \quad c_i = 1. \tag{15}$$

We form a set B of ground truth bounding boxes b for the image I using the set of rectangles $\mathfrak{R}$. The ith ground truth bounding box $b_i = (x_i, y_i, w_i, h_i, c_i)$ is given as

$$x_i = \mathfrak{x}_i, \quad y_i = \mathfrak{y}_i, \quad w_i = \mathfrak{w}_i, \quad h_i = \mathfrak{h}_i, \quad c_i = 1. \tag{16}$$

We make the annotated datasets to be freely available at Kaggle [39].

2.6. Evaluation measures

2.6.1. Total localization error

Positions of objects in images are determined by coordinates of object centroids. Evaluation of the detection performance of centroid-based object detectors must consider distances of centroids predictions

to the nearest ground truth centroid. Therefore, each prediction must be associated exactly with one ground truth label, and each ground truth label must be associated exactly with one prediction. If the number of predictions does not match the number of the ground truth labels, we can add a corresponding number of virtual predictions or ground truth labels to ensure equality of their numbers. Herein, we expect the coordinates of the virtual predictions and virtual ground truth labels to be infinity. Using the above stated principle, we define the total localization error (a single class detection performance measure) as follows.

Let a dataset D consist of N annotated images I. The lth image I_l is associated with a set Γ_l of n_l ground truth centroids γ, where $n_l \in \mathbb{Z}^{0+}$. Let the detector predict for I_l a set $\hat{\Gamma}_l$ of p_l object centroids $\hat{\gamma}$, where $p_l \in \mathbb{Z}^{0+}$. The total localization error $\sum e$ of the detector on D is given as

$$\sum e = \frac{1}{N}\sum_{l=1}^{N} e_l, \tag{17}$$

where e_l is the localization error of the detector on I_l.

Let the localization error of the lth image be given as a sum of the smallest relative distances δ between ground truth coordinates and their closest predictions, where each prediction is associated exactly with one ground truth label, and simultaneously, each ground truth label is associated exactly with one prediction. The error is given as

$$e_l = \sum_{r=1}^{M} \min_{\tilde{\gamma}_j \in \tilde{\Gamma}_l^r} \left\{ \min_{\tilde{\hat{\gamma}}_j \in \tilde{\hat{\Gamma}}_l^r} \delta_l(\tilde{\gamma}_j, \tilde{\hat{\gamma}}_i) \right\}, \tag{18}$$

where $\tilde{\gamma}_j$ is the jth ground truth centroid, $\tilde{\hat{\gamma}}_i$ is the ith predicted centroid, $\delta_l(\tilde{\gamma}_j, \tilde{\hat{\gamma}}_i)$ is a relative distance between $\tilde{\gamma}_j$ and $\tilde{\hat{\gamma}}_i$, $\tilde{\gamma}_j \in \tilde{\Gamma}_l^r$, $\tilde{\hat{\gamma}}_i \in \tilde{\hat{\Gamma}}_l^r$, $\tilde{\Gamma}_l^r$ and $\tilde{\hat{\Gamma}}_l^r$ are multisets of cardinality $M - r + 1$, and $M = \max\{n_l, p_l\}$. For $r = 2, \ldots, M$,

$$\tilde{\Gamma}_l^r = \left\{ \tilde{\gamma}_i | \tilde{\gamma}_i \in \tilde{\Gamma}_l^{r-1} \wedge \tilde{\gamma}_i \neq (\bar{x}^*, \bar{y}^*, c^*)_l^{r-1} \right\}, \tag{19}$$

where

$$(\bar{x}^*, \bar{y}^*, c^*)_l^{r-1} = \arg\min_{\tilde{\gamma}_j \in \tilde{\Gamma}_l^{r-1}} \left\{ \min_{\tilde{\hat{\gamma}}_i \in \tilde{\hat{\Gamma}}_l^{r-1}} \delta_l(\tilde{\gamma}_j, \tilde{\hat{\gamma}}_i) \right\}, \tag{20}$$

Similarly,

$$\tilde{\hat{\Gamma}}_l^r = \left\{ \tilde{\hat{\gamma}}_i | \tilde{\hat{\gamma}}_i \in \tilde{\hat{\Gamma}}_l^{r-1} \wedge \tilde{\hat{\gamma}}_i \neq (\hat{\bar{x}}^*, \hat{\bar{y}}^*, \hat{c}^*)_l^{r-1} \right\}, \tag{21}$$

where

$$(\hat{\bar{x}}^*, \hat{\bar{y}}^*, \hat{c}^*)_l^{r-1} = \arg\min_{\tilde{\hat{\gamma}}_i \in \tilde{\hat{\Gamma}}_l^{r-1}} \left\{ \min_{\tilde{\gamma}_j \in \tilde{\Gamma}_l^{r-1}} \delta_l(\tilde{\gamma}_j, \tilde{\hat{\gamma}}_i) \right\}. \tag{22}$$

For $r = 1$, the multisets $\tilde{\Gamma}_l^r$ and $\tilde{\hat{\Gamma}}_l^r$ contain, with multiplicity one, all elements of the sets Γ_l and $\hat{\Gamma}_l$, respectively; and the element $(\infty, \infty, 0)$ with multiplicity $\max\{0, p_l - n_l\}$ and $\max\{0, n_l - p_l\}$, respectively.

The relative distance between the jth ground truth centroid $\tilde{\gamma}_j = (\bar{x}_j, \bar{y}_j, c_j)$ and the ith centroid prediction $\tilde{\hat{\gamma}}_i = (\hat{\bar{x}}_i, \hat{\bar{y}}_i, \hat{c}_i)$ in I_l is given as

$$\delta_l(\tilde{\gamma}_j, \tilde{\hat{\gamma}}_i) = \frac{\min\{w_{I_l}, \left|\bar{x}_j - \hat{\bar{x}}_i\right|\}}{2w_{I_l}} + \frac{\min\{h_{I_l}, \left|\bar{y}_j - \hat{\bar{y}}_i\right|\}}{2h_{I_l}}, \tag{23}$$

where w_{I_l} and h_{I_l} are width and height of the lth image I_l, respectively.

In other words, if the number of predictions p_l is equal to the number of ground truth labels n_l for the lth image (Fig. 7(a)), the error (18) can be directly calculated as the sum of the smallest relative distances of the ground truth label–prediction pairs designed according to (19)–(22). If $p_l > n_l$ (Fig. 7(b)), we add $(p_l - n_l)$ virtual ground truth labels $(\infty, \infty, 0)$ to allow calculation of the error (18). If $p_l < n_l$ (Fig. 7(c)), we add $(n_l - p_l)$ virtual predictions $(\infty, \infty, 0)$.

2.6.2. Relative inference time

Let a relative inference time of a detector be given as

$$\tau = \frac{t_D}{t_B}, \tag{24}$$

where t_D is the total inference time of the detector on a set of images, and t_B is the total inference time of a baseline detector of this set.

2.7. Experiment conditions

We implement the centroid-based detector in Python 3.6 with TensorFlow 2.0. We train the full resolution U-Net and the reduced U-Net map generators from scratch, minimizing a binary cross entropy function. We use normal distribution initialization with mean and standard deviation set to 0 and 0.05, respectively. When training the reduced U-Net, we resize the localization maps to 72×72 px (dimensions of localization maps produced by the reduced U-Net). For both variants of U-Nets, we rescale values of the images and of the localization maps to the range $[0, 1]$. In the centroid counterpoint module, we set the threshold value t_m and the size of the maximum filter $h_k \times w_k$ at 0.65 and 10×10 px, respectively. The threshold value was estimated experimentally. The filter size is set with respect to the most common size of heads in the images.

Since the Computer Vision Toolbox in Matlab offers a YOLO implementation wrapped in a very user-friendly interface including the possibility of deployment to Jetson NANO, we implement the YOLO detector (the bounding box-based detector) in MATLAB instead of the original Darknet framework. We use models pre-trained on the Imagenet for its training on the person detection task. We replace the layers after 'out_relu', 'inception_5b-output', and 'relu_conv10' (naming according MATLAB) in GoogLeNet, MobileNet-v2, and SqueezeNet, respectively, by the last YOLOv2 layers. Moreover, since these backbone architectures expect a three-channel RGB image as input, we add a convolutional layer with three trainable filters (3, 3) in front of these base models to transform a single-channel input into a three-channel input. We empirically find that the overlap threshold in non-maximum suppression at 0.75 gives good results, and 7 anchor boxes are a good balance between performance and time to process. We estimate widths and heights of the anchor boxes on the dataset D_T using a k-means clustering algorithm and the IoU distance metric [34].

We use PyTorch implementation of CenterNet from GitHub [40] with its initial setting of all parameters. As backbones, we use ResNet-101 and EfficientDET D0 models pre-trained on the COCO dataset for its training on the person detection task. We modify the backbones to process 288×288 px images analogously to the YOLO detector, and we adjust the CenterNet outputs for the one class problem.

For both the centroid-based and bounding box-based detectors, we convert the input images into greyscale. We carry out 5 training sessions. Within each session, we train the localization map generators and all variants of the bounding box-based detectors on an identical training subset. For each training session, we randomly split up the dataset D_T at the ratio 17:3 into training and validation subsets, respectively. We train the generators and the bounding box-based detectors with mini batches of 8 samples for 300 and 30 epochs, respectively. We save and validate the models on a validation subset in every epoch. We shuffle samples in every epoch.

We use the Adam optimizer for the training of the map generators and of the bounding box-based person detectors. We set up an exponential decay rate for first and second moment estimates at 0.9 and 0.999, respectively. For the generators and for the CenterNet detectors, we use an initial learning rate of 10^{-3}. In the case of CenterNet detectors, we multiply the learning rate by a factor of 0.96 every 10 epochs. For the YOLO detectors, we set up the learning rates for the last (the YOLOv2) layers at 10^{-3}, the preceding layers are not modified during learning. These settings are adapted from the sources of the individual architectures, i.e., Computer Vision Toolbox in Matlab for the YOLO detector, and GitHub source [40] for the CenterNet.

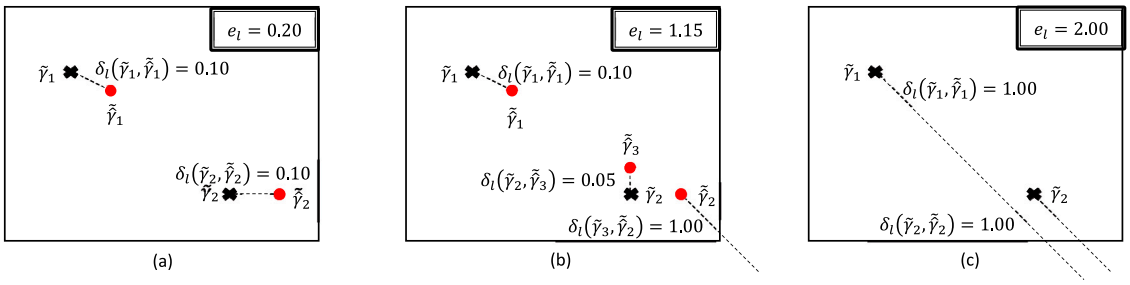


Fig. 7. Calculation of the localization error for the lth image (black solid lines) with two ground truth centroids (black crosses) in the case of (a) two, (b) three, and (c) zero centroid predictions (red circles). In the case of two predictions (a), the number of ground truth and the number of predicted coordinates are equal, and the localization error e_l is simply the sum of the smallest distances between the ground truth and the predictions (dashed black lines), i.e. the localization error of the lth image $e_l = 0.20$. In the case of three predictions (b), one of the predictions is redundant. When calculating the localization error, we first find for each ground truth centroid the nearest prediction and calculate the distance between the prediction and the ground truth. The remaining prediction is the redundant one (in this case $\tilde{\hat{y}}_2$), and we consider its ground truth $\tilde{y}_3$ to be in infinity, which corresponds to the highest possible distance in the image (i.e. $\delta_l(\tilde{y}_3, \tilde{\hat{y}}_2) = 1$). Thus, the localization error $e_l = 1.15$ in this case. In the case of zero predictions (c), two predictions are missing. We expect the predictions to be in infinity which results in $e_l = 2.00$.

We use data augmentation to avoid overfitting by the training of the map generators. Specifically, we use random rotation (range of a rotation angle: ±20 degree), random horizontal and vertical flipping with probability 0.5, random horizontal and vertical translation (up to ±20 % of image height and width, respectively), random rescaling (zoom range 0.2) and random horizontal and vertical shear (shear intensity 0.2). We perform the augmentation in filling mode set to nearest.

We evaluate performance of the person detectors on the test and blind datasets D_E and D_B, respectively. For both variants of the centroid-based and all five variants of the bounding box-based detectors, we select the best performing model (a model with the smallest value of a loss function over a validation subset obtained within the five training sessions). We calculate the total localization error (17) of the models on D_E and D_B. For each image in a dataset, we convert a set $\hat{B}$ of YOLO predictions $\hat{b}$ into a set $\hat{I}$ of centroid predictions $\hat{y}$. The coordinate of the ith centroid prediction in an image I is given as

$$\hat{\tilde{x}}_i = \hat{x}_i + 0.5\hat{w}_i, \quad \hat{\tilde{y}}_i = \hat{y}_i + 0.5\hat{h}_i. \tag{25}$$

In the case of CenterNet, we use the centroid coordinate predictions by its evaluation.

For each person detector, we detect persons in one identical image I a thousand times, while measuring total inference time of the detector t_D. We calculate for the detectors frame rates $F = t_D^{-1}$ and the relative inference time (24), where the total inference time of the centroid-based detector with the reduced U-Net map generator is the baseline t_B.

We train and evaluate detection performance of the detectors on a personal computer with Intel Core i5-8600K (3.6 GHz) CPU, internal memory 16 GB DDR4 (2666 MHz), video card NVIDIA PNY Quadro P5000 16 GB GDDR5 PCIe 3.0. For evaluation of the inference time, we use a NVIDIA Jetson NANO single-board computer with Quad-core ARM A57 1.43 GHz CPU and 4 GB RAM. To allow the unbiased comparison of the inference times, we export all detectors into the TensorRT NVIDIA CUDA parallel programming models.

3. Results

We train the centroid-based detectors as well as the competitive bounding box-based detectors with the datasets described in Section 2.5 according to the procedure addressed in Section 2.7. In order to show the performance capabilities of the detectors, we summarize the resulting values of the evaluation measures described in Section 2.6 in Table 3.

Additionally, we demonstrate absolute frequencies of differences between numbers of ground truth labels and numbers of predictions of the detectors for the test dataset (Table 4) and for the blind dataset (Table 5). A graphical representation of these differences is depicted in Fig. 8.

Table 3
Evaluation results.

Measure	$\sum e, -$		$\tau, -$	F, FPS
Dataset	D_E	D_B	$I \times 1000$	$I \times 1000$
Full resolution U-Net	0.1472	0.3712	1.4148	8.62
Reduced U-Net	**0.1352**	**0.3378**	1.0000	12.19
CenterNet-D0	1.4659	1.7200	3.9260	3.10
CenterNet-ResNet101	1.2080	1.7090	4.0215	3.03
YOLO-GoogleNet	0.6755	1.2159	1.2074	10.10
YOLO-MobileNetv2	0.3497	1.0016	1.43779	8.48
YOLO-SqueezeNet	1.9441	1.5899	**0.93549**	**13.03**

Total localization errors $\sum e$, relative inference times τ, and framerates F (first row) of the best performing models (third to ninth rows), on the test and blind datasets D_E and D_B (second row), respectively. I symbolizes an image which we used a thousand times within measurement of inference times of the models, and FPS stands for frame per second. The best result is for each measure in bold.

Table 4
Absolute frequencies of differences between numbers of ground truth labels and numbers of predictions for the test dataset D_E.

	<−2	−2	−1	0	1	2	>2
Full resolution U-Net	0	9	95	876	19	1	0
Reduced U-Net	0	4	84	**885**	24	3	0
CenterNet-D0	97	101	148	328	179	50	97
CenterNet-ResNet101	55	68	159	385	218	55	60
YOLO-GoogleNet	27	75	259	503	124	10	2
YOLO-MobileNetv2	1	22	172	700	97	8	0
YOLO-SqueezeNet	317	194	223	248	18	0	0

The test dataset consists of 1000 samples. The frequencies of multiple detections (columns for negative numbers in the first row), overlooks (columns for positive numbers in the first row), and correct detections (the column for zero in the first row) are arranged with respect to the detectors (the first column). The highest value for the correct detections indicates the best performing detector (in bold).

4. Discussion

The evaluation results presented in Table 3 speak in favour the centroid-based person detection. On both the test dataset D_E and the blind dataset D_B, the localization errors of both variants of the centroids-based detector are less than half of the localization errors of the best performing bounding box-based detector (YOLOv2 with the MobileNetv2 backbone). The centroid-based detector with the reduced U-Net map generator shows even 2.6 and 2.9-times smaller error

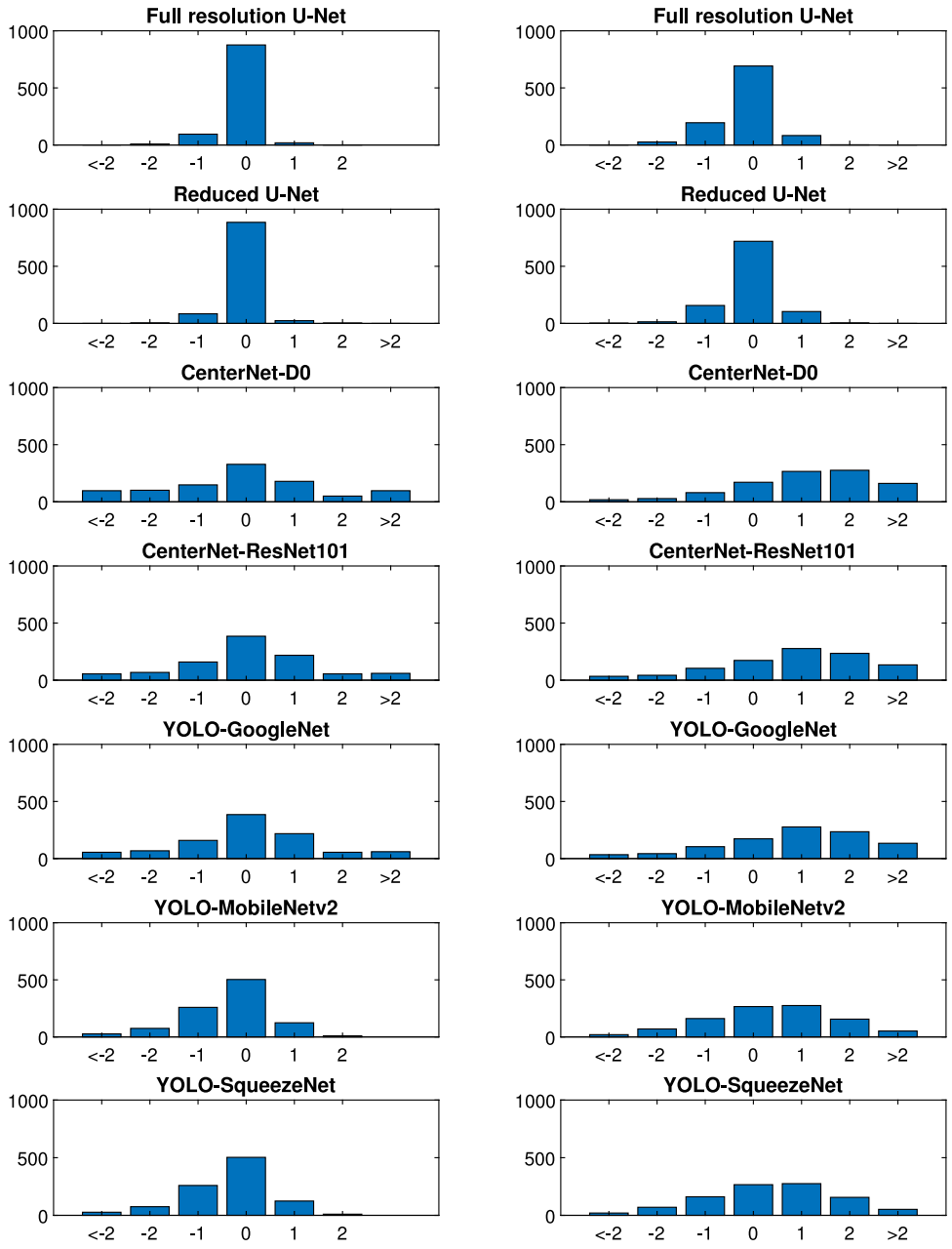


Fig. 8. Absolute frequencies of differences between numbers of ground truth labels and numbers of predictions for the test dataset D_E (left column) and for the blind dataset D_B (right column). The abbreviations are identical with Table 4.

on D_E and on D_B, respectively, compared to the YOLOv2 with the MobileNetv2 backbone.

The high localization errors of YOLO detectors (Table 3, YOLO-GoogleNet and YOLO-SqueezeNet on D_E and D_B, and YOLO-MobileNetv2 on D_B) indicate predispositions of the detectors to multiple detections or to marginalization of persons in the images. The results summarized in Tables 4–5 and in Fig. 8 confirm this suspicion. The YOLO-SqueezeNet, which has the highest errors on both datasets (Table 3), shows a clear tendency to overlook persons (Fig. 8). The YOLO-GoogleNet with the second highest errors on both datasets inclines to false detections on the blind dataset D_B. On the test dataset D_E, it leans rather to overseeing persons (Fig. 8). The high error of YOLO-MobileNetv2 on D_B and the small error on D_E indicates low generalization capability of the detector. The decline in the number of images with the correct number of detections confirms this assumption (compare the histograms for YOLO-MobileNetv2 on D_E and D_B in Fig. 8, or the results in Tables 4–5).

The tendency of multiple detections and marginalization of persons is even greater in the case of the CenterNet detectors. This is most apparent in the results in Tables 4–5; however, this property of the detectors naturally emerges in the localization errors too. The CenterNet with the EfficientDET D0 and ResNet101 backbones have the first and second highest localization errors among the tested detectors on both the test and the blind datasets.

The low values of the error for both variants of the centroid-based detector point to low numbers of false and miss-detections which also confirm the histograms shown in Fig. 8 (see Full resolution U-Net and Reduced U-Net). The values of the error on the blind dataset D_B (Table 3, Full resolution U-Net and Reduced U-Net) indicate good generalization capability of the centroid-based detector. The detector

Table 5
Absolute frequencies of differences between numbers of ground truth labels and numbers of predictions for the blind dataset D_B.

	<−2	−2	−1	0	1	2	>2
Full resolution U-Net	1	27	195	692	83	2	0
Reduced U-Net	2	14	157	**719**	104	4	0
CenterNet-D0	18	28	80	171	266	276	161
CenterNet-ResNet101	34	43	104	173	277	235	134
YOLO-GoogleNet	20	70	161	266	275	156	52
YOLO-MobileNetv2	51	96	260	343	201	40	9
YOLO-SqueezeNet	182	197	232	238	124	25	2

Table layout and the abbreviations are identical with Table 4.

with a reduced U-Net map generator has the highest number of images with the correct number of detections on both datasets (Tables 4–5). Despite the low quality of some images in the datasets, the number of detections match the number of persons for 88.5% and 71.9% of images in D_B and D_E, respectively.

The utilization of quarter size localization maps instead of the full resolution ones results in a slight improvement in detections, which is apparent from the localization errors (Table 3, Full resolution U-Net and Reduce U-Net) as well as from the numbers of 'correct detections' (Tables 4–5). Decreasing the map resolution does not change the distribution in false and miss-detection frequencies (Fig. 8, Full resolution U-Net and Reduced U-Net on D_E and D_B). The results indicate that the reduction of the map resolution contributes to a better generalization of the network within the training phase.

The simplification of the U-Net topology which we have made within the development of the reduced U-Net, allows us to reach inference time comparable to the fastest bounding box-based detector (compare τ in Table 3 for Reduced U-Net and YOLOv2-SqueezeNet). It is worth mentioning in this context that the YOLOv2 with the SqueezeNet backbone shows one of the highest localization errors among the evaluated detectors (14.4-times and 4.7-times higher errors on D_E and on D_B, compared to the centroids-based detector with the reduced U-Net map generator). When compared with the best performing bounding box-based detector, the centroid-based detector with the reduced U-Net map generator is about 40 % faster than the YOLOv2 with the MobileNetv2 backbone (Table 3). When using the full resolution U-Net, the inference times of the centroid-based detector and of the YOLOv2 with the MobileNetv2 backbone are almost identical.

The presented results confirm our expectations with respect to the advantages of the centroid-based detection by the localization of persons in the orthogonally captured images. The low localization errors of both versions of the centroid-based detector support previously published results, which point to the superiority of the centroid-based object detection over the bounding box-based object detection for counting small objects [41]. The centroid-based detector also meets the edge computing standards, especial when using the reduced U-Net as the map generator.

5. Conclusion

We proved that the determination of head centroids position, using the fully convolutional network (U-net) in combination with the presented sequence of simple image processing operations (centroid counterpoint module), is an efficient way for the fast and precise detection of persons in orthogonally captured images. The presented centroid-based person detector meets the edge computing standards, has good generalization capability, and shows small localization error even on low quality images. It efficiently operates in diverse environments including environments with high variabilities in elevation profiles. The utilization of quarter size localization maps instead of the full resolution ones allowed us to reduce inference time of the detector by 40 %. A side effect of the reduction is a slight improvement in the detection performance. Considering all these facts and the low price of visible spectrum cameras (compared to depth cameras, 3D laser scanners, etc.), we conclude that the centroid-based detector allows development of low cost and powerful commercial solutions. These solutions are particularly aimed at automatic tracking and counting of persons in public transport. The presented localization error allowed us to quantitatively compare detection performance of both centroid and bounding box-based detectors. For the annotation of datasets, we used the bounding box inspired annotation. Such an approach allowed us simultaneous annotation of the images for both bounding box-based and centroid-based detection. This considerably simplified the annotation process.

CRediT authorship contribution statement

Petr Dolezel: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing – review & editing, Project administration, Funding acquisition. **Pavel Skrabanek:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Supervision. **Dominik Stursa:** Conceptualization, Methodology, Validation, Formal analysis, Writing – review & editing. **Bruno Baruque Zanon:** Investigation, Writing – review & editing. **Hector Cogollos Adrian:** Investigation, Writing – review & editing. **Pavel Kryda:** Investigation, Resources, Data curation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Pavel Kryda is currently affiliated with a commercial company Mikroelektronika spol. s r. o. He provided expertise in dataset acquisition and supervision in this article. This does not alter the authors' adherence to Journal of Computational Science policies on sharing data and materials.

Acknowledgements

The work was supported from ERDF/ESF "Cooperation in Applied Research between the University of Pardubice and companies, in the Field of Positioning, Detection and Simulation Technology for Transport Systems (PosiTrans)" (No. CZ.02.1.01/0.0/0.0/17_049/0008394).

References

[1] D.K. Singh, S. Paroothi, M.K. Rusia, M.A. Ansari, Human crowd detection for city wide surveillance, Procedia Comput. Sci. 171 (2020) 350–359, http://dx.doi.org/10.1016/j.procs.2020.04.036.

[2] P. Skrabanek, P. Dolezel, Z. Nemec, D. Stursa, Person detection for an orthogonally placed monocular camera, J. Adv. Transp. 2020 (2020) 8843113, http://dx.doi.org/10.1155/2020/8843113.

[3] J.W. Choi, X. Quan, S.H. Cho, Bi-directional passing people counting system based on IR-UWB radar sensors, IEEE Internet Things J. 5 (2) (2018) 512–522, http://dx.doi.org/10.1109/JIOT.2017.2714181.

[4] Z. Chen, W. Yuan, M. Yang, C. Wang, B. Wang, SVM based people counting method in the corridor scene using a single-layer laser scanner, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems, ITSC, 2016, pp. 2632–2637.

[5] S. Akamatsu, N. Shimaji, T. Tomizawa, Development of a person counting system using a 3D laser scanner, in: 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014), 2014, pp. 1983–1988, http://dx.doi.org/10.1109/ROBIO.2014.7090627.

[6] A. Ahmed, N.A. Siddiqui, Design and implementation of infra-red based computer controlled monitoring system, in: 2005 Student Conference on Engineering Sciences and Technology, 2005, pp. 1–5, http://dx.doi.org/10.1109/SCONEST.2005.4382890.

[7] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 6517–6525, http://dx.doi.org/10.1109/CVPR.2017.690.

[8] X. Zhou, J. Zhuo, P. Krähenbühl, Bottom-up object detection by grouping extreme and center points, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 850–859, http://dx.doi.org/10.1109/CVPR.2019.00094.

[9] E.H. Nguyen, H. Yang, R. Deng, Y. Lu, Z. Zhu, J.T. Roland, L. Lu, B.A. Landman, A.B. Fogo, Y. Huo, Circle representation for medical object detection, IEEE Trans. Med. Imag. 41 (3) (2022) 746–754, http://dx.doi.org/10.1109/TMI.2021.3122835.

[10] A.M. Hafiz, G.M. Bhat, A survey on instance segmentation: state of the art, Int. J. Multimed. Inf. Retrieval 9 (3) (2020) 171–189, http://dx.doi.org/10.1007/s13735-020-00195-x.

[11] K. Dijkstra, L.R.B. van de Loosdrecht, M.A. Wiering, CentroidNet: A deep neural network for joint object localization and counting, in: Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, Cham, 2019, pp. 585–601.

[12] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2017) 640–651, http://dx.doi.org/10.1109/TPAMI.2016.2572683.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: Single shot MultiBox detector, in: Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 21–37.

[14] H. Law, J. Deng, CornerNet: Detecting objects as paired keypoints, Int. J. Comput. Vis. 128 (3) (2020) 642–656, http://dx.doi.org/10.1007/s11263-019-01204-1.

[15] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, CenterNet: Keypoint triplets for object detection, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV, 2019, pp. 6568–6577, http://dx.doi.org/10.1109/ICCV.2019.00667.

[16] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587, http://dx.doi.org/10.1109/CVPR.2014.81.

[17] R. Girshick, Fast R-CNN, in: 2015 IEEE International Conference on Computer Vision, ICCV, 2015, pp. 1440–1448, http://dx.doi.org/10.1109/ICCV.2015.169.

[18] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 28, Curran Associates, Inc., 2015.

[19] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916, http://dx.doi.org/10.1109/TPAMI.2015.2389824.

[20] J. Dai, Y. Li, K. He, J. Sun, R-FCN: Object detection via region-based fully convolutional networks, in: D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 29, Curran Associates, Inc., 2016.

[21] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, R. Yang, Salient object detection in the deep learning era: An in-depth survey, IEEE Trans. Pattern Anal. Mach. Intell. (2021) 1, http://dx.doi.org/10.1109/TPAMI.2021.3051099.

[22] M. Zhai, X. Xiang, N. Lv, X. Kong, Optical flow and scene flow estimation: A survey, Pattern Recognit. 114 (2021) 107861, http://dx.doi.org/10.1016/j.patcog.2021.107861.

[23] X. Cai, Y.-F. Pu, Flattenet: A simple and versatile framework for dense pixelwise prediction, IEEE Access 7 (2019) 179985–179996, http://dx.doi.org/10.1109/ACCESS.2019.2959640.

[24] Y. Yuan, M. Chao, Y. Lo, Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance, IEEE Trans. Med. Imaging 36 (9) (2017) 1876–1886, http://dx.doi.org/10.1109/TMI.2017.2695227.

[25] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495, http://dx.doi.org/10.1109/TPAMI.2016.2644615.

[26] O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional Networks for Biomedical Image Segmentation, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9351 (2015) 234–241, http://dx.doi.org/10.1007/978-3-319-24574-4_28.

[27] J. Schlemper, O. Oktay, M. Schaap, M. Heinrich, B. Kainz, B. Glocker, D. Rueckert, Attention gated networks: Learning to leverage salient regions in medical images, Med. Image Anal. 53 (2019) 197–207, http://dx.doi.org/10.1016/j.media.2019.01.012.

[28] T. Pohlen, A. Hermans, M. Mathias, B. Leibe, Full-resolution residual networks for semantic segmentation in street scenes, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 3309–3318, http://dx.doi.org/10.1109/CVPR.2017.353.

[29] N. Beheshti, L. Johnsson, Squeeze U-net: A memory and energy efficient image segmentation network, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2020, pp. 1495–1504, http://dx.doi.org/10.1109/CVPRW50498.2020.00190.

[30] M.A. Islam, M. Rochan, N.D.B. Bruce, Y. Wang, Gated feedback refinement network for dense image labeling, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 4877–4885, http://dx.doi.org/10.1109/CVPR.2017.518.

[31] C. Peng, X. Zhang, G. Yu, G. Luo, J. Sun, Large kernel matters — Improve semantic segmentation by global convolutional network, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 1743–1751, http://dx.doi.org/10.1109/CVPR.2017.189.

[32] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union, in: The IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2019.

[33] M. Belloc, S. Velastin, R. Fernandez, M. Jara, Detection of people boarding/alighting a metropolitan train using computer vision, in: IET Conference Proceedings, 2018, pp. 22–27.

[34] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, 2016, arXiv preprint arXiv:1612.08242.

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 2818–2826, http://dx.doi.org/10.1109/CVPR.2016.308.

[36] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520, http://dx.doi.org/10.1109/CVPR.2018.00474.

[37] F.N. Iandola, M.W. Moskewicz, K. Ashraf, S. Han, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size, 2016, CoRR abs/1602.07360, arXiv:1602.07360.

[38] M. Tan, R. Pang, Q.V. Le, EfficientDet: Scalable and efficient object detection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 10778–10787, http://dx.doi.org/10.1109/CVPR42600.2020.01079.

[39] P. Dolezel, D. Stursa, Person detection (orthogonally captured images), Kaggle, 2022, http://dx.doi.org/10.34740/KAGGLE/DSV/3595070, URL https://www.kaggle.com/dsv/3595070.

[40] X. Zhou, D. Wang, P. Krähenbühl, Objects as points, 2019, arXiv preprint arXiv:1904.07850.

[41] K. Dijkstra, J. van de Loosdrecht, W.A. Atsma, L.R. Schomaker, M.A. Wiering, CentroidNetV2: A hybrid deep neural network for small-object segmentation and counting, Neurocomputing 423 (2021) 490–505, http://dx.doi.org/10.1016/j.neucom.2020.10.075.

Petr Dolezel received his Ph.D. degree from the University of Pardubice, Czech Republic, in 2009. In 2017, he defended his habilitation thesis at Tomas Bata University. He works as an associate professor and vice-dean for research and development at the Faculty of Electrical Engineering and Informatics, University of Pardubice. His research interests include neural and evolutionary computation in process control and in image processing. He is the author of more than 100 scientific contributions, including 20 journal papers and lectures at CORE ranked conferences. He has been a leader or member of research teams for a dozen research and development projects.

Pavel Skrabanek received the Ph.D. degree in Technical Cybernetics in 2010 from the University of Pardubice, Czech Republic. He joined the Faculty of Electrical Engineering and Informatics, University of Pardubice, Czech Republic as an Assistant in 2007 and in 2011, he became an Assistant Professor there. Since 2018, he works as an Assistant Professor at the Institute of Automation and Computer Science, Brno University of Technology, Czech Republic. His research interests include computer vision, image processing, optimization, machine learning and fuzzy set theory. Among other things, he is the corresponding author of 5 articles published in high impact scientific journals.

Dominik Stursa is a Ph.D. student at the University of Pardubice, Czech Republic. His topic is image processing applications with the deep neural networks. Since 2019, he is a lecturer at the Process Control Department, University of Pardubice. He is the key member of a research group lead by Petr Dolezel. His research aims to robotics, signal and image processing and neural networks. He is an author of 5 journal articles and more than 10 conference papers. He has a membership with IEEE Robotics and Automation Society and IEEE Signal Processing Society.

Bruno Baruque Zanon holds an associate professor position at the University of Burgos, Spain since 2018. He obtained his Ph.D. degree in Computer Science Artificial Intelligence in 2009 at that same university. He is an active member of the GICAP research group at University Burgos and author of more than 80 research publications in indexed journals and conferences. His research interests focus on the data analysis and automated learning field, with special emphasis in artificial neural networks. He collaborates as guest editor and reviewer of several international journals and numerous international conferences related with the artificial intelligence knowledge area.

Hector Cogollos Adrian is a graduate in computer engineering (2019), and with a master's degree in computer engineering (2021). He currently works as assistant teacher at the Computer Science Department, while working on the pursuit of his Ph.D. degree at the University of Burgos. His research interest focuses on the study of artificial intelligence techniques for the analysis and improvement of mobility.

Pavel Kryda received the M.S. degree in electrical engineering from the University of Pardubice, Czech Republic, in 2012. He worked as a HW research specialist in Mikroelektronika spol. s r. o., Vysoke Myto, Czech Republic. In 2019, he was promoted to lead product manager. His previous work experiences include embedded systems programming, design of electrical devices, and sensors systems.