Applied Intelligence (2022) 52:13824–13838
https://doi.org/10.1007/s10489-022-03172-5

Monitoring social-distance in wide areas during pandemics: a density map and segmentation approach

Javier Antonio Gonzalez-Trejo[1] · Diego A. Mercado-Ravell[2] · Uziel Jaramillo-Avila[1]

Accepted: 30 December 2021 / Published online: 5 April 2022

Abstract

With the relaxation of the containment measurements around the globe, monitoring the social distancing in crowded public spaces is of great importance to prevent a new massive wave of COVID-19 infections. Recent works in that matter have limited themselves by assessing social distancing in corridors up to small crowds by detecting each person individually, considering the full body in the image. In this work, we propose a new framework for monitoring the social-distance using end-to-end Deep Learning, to detect crowds violating social-distancing in wide areas, where important occlusions may be present. Our framework consists in the creation of new ground truth social distance labels, based on the ground truth density maps, and the proposal of two different solutions, a density-map-based and a segmentation-based, to detect crowds violating social-distancing constraints. We assess the results of both approaches by using the generated ground truth from the PET2009 and CityStreet datasets. We show that our framework performs well at providing the zones where people are not following the social-distance, even when heavily occluded or far away from the camera, compared to current detection and tracking approaches.

Keywords Visual social distancing · COVID-19 · Crowds monitoring · Density maps · Segmentation · Deep learning

1 Introduction

After the outbreak of the COVID-19 pandemic, the whole world witnessed how the health system was threatened to the edge of collapse. Furthermore, due to the previous lack of a vaccine or even a proper treatment for this new virus, social-distancing became the only viable strategy to contain the massive wave of contagions. Nevertheless, this also came with the price of bringing the economic activities almost to a complete stop, hence putting the social and economical stability to a sever risk. Even up to date, regardless of the successful development of vaccines against the virus, the global demand is too high, and the logistics too complicated that we would need to wait some time to see the world to come back to its normality, not to forget the always present risk of a virus mutation resistant to the available vaccines.

In this context, at the unavoidable need to reactivate the economy and prevent the collapse of society; people, companies and governments have been forced to relax the strict isolation measurements, in spite of the latent risk of a new wave of contagions. Accordingly, automatic social-distance monitoring, also referred as Visual Social Distancing (VSD) has emerged as an interesting research topic that will assist authorities to prevent massive contagions while people slowly recover their normal lifestyle.

This article belongs to the Topical Collection: *Artificial Intelligence Applications for COVID-19, Detection, Control, Prediction, and Diagnosis*

✉ Diego A. Mercado-Ravell
diego.mercado@cimat.mx

Javier Antonio Gonzalez-Trejo
javier.gonzalez@cimat.mx

Uziel Jaramillo-Avila
uziel.jaramillo@cimat.mx

[1] Center for Research in Mathematics CIMAT AC, campus Zacatecas, Avenida Lasec, Andador Galileo Galilei, Manzana 3 Lote 7, Parque Quantum, Zacatecas 98160, Mexico

[2] Investigador CONACyT at Center for Research in Mathematics CIMAT AC, campus Zacatecas, Zacatecas 98160, Mexico

Due to its actual great relevance, a few works have recently been proposed in order to tackle this problem using computer vision [1–7]. However, all the precedent solutions encountered in the literature rely on the same principle idea of using a state-of-the-art detector to locate each person individually and calculate their inter-personal distance. Some of the classical computer vision problems involved in this kind of solution are object detection, multi-object tracking, pose estimation, homography transformation, metric scale and depth estimation, multi-view fusion, etc.

Inspired by the recent success of density maps in the crowd detection and counting tasks, and in contrast to the commonly used detect and track approach, we propose to tackle the VSD problem as a segmentation problem, and train Deep Neural Networks (DNN) to directly detect those groups of people not in compliance with social-distance restrictions, based only on the people's heads. Also, we propose an alternative solution using density maps to detect crowds not in compliance with social-distancing, using the people's density information. We believe that these are unexplored and interesting alternative solutions, which may offer better performance in wide scenarios with larger crowds and significant occlusions, which are common in real urban spaces. To do so, our contributions are summarized as follows:

- We propose a framework to train DNN to solve the VSD problem based in either density maps and segmentation approaches.
- Using the head's annotations in available public crowds datasets, and the homography from the camera, we create the VSD ground truth by removing the social-distance conforming people.
- Based on the VSD ground truth, we propose a metric to evaluate the density map and segmentation approaches for detecting non social-distance conforming crowds.
- To our knowledge, this is the first solution to the VSD problem by using density maps and segmentation, which appear as interesting alternatives for wider scenarios, where larger crowds subject to important occlusions may be present.

The article is organized as follows: Section 2 presents a literature review of related works, while in Section 3 we discuss the VSD problem and provide a formal definition of social-distance, in Section 4 we explore the framework to generate the ground truth annotations and to train the solutions. Then, in Section 5 we detail the training stage, whereas in Section 6 we present our results and compare our best solution against two open available solutions in the state-of-the-art. Finally, in Section 7 we provide our final conclusions and future work.

2 Related work

Due to the great relevance to help to prevent massive contagions and recover the most important social and economical activities without jeopardizing public health, several VSD solutions are quickly steaming in the literature. Up to date, all the reported works rely on the same intuitive principle idea: use a state-of-the-art object detector and find some sort of inter-personal distance between each individual instance. The most common detectors for this task are YOLO-based (You Only Look Once) [1, 2, 4, 7, 8], but SSD (Single Shot Detector) [5], Mask R-CNN and Faster R-CNN (Region-based Convolutional Neural Network) have also been proposed, [2, 6] respectively. Some of these works [1, 4, 6] further combine the detector with a tracking algorithm such as DeepSORT [9], in order to improve time consistency along video streams, further enhancing the system precision.

In [10] Cristani et al. introduced the VSD problem, as the automatic estimation of the inter-personal distance from an image, and the characterization of related people aggregations. The authors discuss the problem not only as a geometric one, but also considering the social implications, and even ethic aspects. Moreover, they identify the most common strategy for this problem, which consists in detecting each person individually and track them along a video stream, while calculating the inter-personal distance either in image space or in ground space.

Following this principle idea, one of the most interesting works is the one proposed by Rezaei and Azarmi [1], where a new DNN based on YOLOv4, called DeepSOCIAL, is presented for this particular task. There, the same detection and tracking framework using YOLO-based detectors and DeepSORT trackers is adopted, but the authors further assess online infection risk by statistical analyzing the spatio-temporal data from people's moving trajectories and the rate of social distancing violations.

Another interesting work in the same lane is the one by Yang et al. [2], where a vision-based social distancing system is studied using either YOLOv4 or Faster R-CNN detectors in the image plane, then the detections are projected to the head's plane, where the inter-personal distance is retrieved, and non-intrusive audio-visual cues are send to the crowd in case of social distancing violations. Furthermore, the authors define the critical social density for a region of interest, it is, the critical number of people within an area below which the probability of contagions can be held close to zero. Finally, the authors released their implementation as open-source software.

Besides using the same person-by-person detection strategy, in [8] the authors propose a novel method to estimate the inter-personal distance without computing the homography transformation, which can be useful when there is not available information about the camera pose. In this work, the spatial patterns of social distances are also analyzed along time, by means of heat maps of social distancing violations.

In parallel with social distancing monitoring, some works such as [11–13] aim to further assist in the pandemics control by also detecting whether people are wearing masks or not. This is an interesting complementary approach that may be used along with VSD to assess the risk of infections in certain region, given that people wearing masks are less prone to contagions.

Although person-by-person detection and tracking has proven to be a valid solution to the VSD problem, becoming the most popular, not to say the only, available kind of solution, it still presents some drawbacks inherent to the detection itself, particularly in more challenging scenarios where wider areas and larger crowds are covered, specially when severe occlusions are present, as is common in real urban scenarios. In order to attenuate this issue, but following the same person-to-person detection strategy, Shao et. al. proposed to use a head detector with PeeleNet [3], instead of aiming to detect the full body of the person, resulting in an improvement in cases where important occlusions are considered. Additionally, the authors also consider the use of an aerial drone to monitor the social distancing.

Nevertheless, other modern deep learning techniques have proven to be more effective in such scenarios, as is the case of density maps. Density map generators are better suited for crowd counting and crowd location since they are trained to localize human head features, which are the most visible parts of a person from upper views, for instance from security cameras or drones, specially when there are severe occlusions in dense crowds or other type of visual obstacles [14]. Recently, density maps generators using Deep Learning have achieved excellent results in the detection and counting tasks for dense crowds, using modern techniques such as MCCN (Multi Column Neural Network) [14]. Current research on density maps not only includes the design of new architectures [15, 16], but also the proposal of new loss functions specific to the task [17, 18], counting from images taken from drones far above the crowd [19], proposing new frameworks where the data and the neural networks are processed before and after the training [20], and combining images taken from different types of cameras [21].

Following these cues, we propose a new framework, and two different alternative solutions using the same ideas from density maps (see Fig. 1), where we do not intent to detect each person individually, but rather directly infer through a DNN those groups of people that are not in compliance with the social distance. The first solution consists on a density map detector, where the people's density is then used to evaluate the level of risk of contagions. The second solution is an end-to-end segmentation algorithm, using either FCN_7 (Fully Convolutional Network) [22] or U-Net [23] as a backbone, and particularly tailored to detect those groups of people that are non conforming with the social distance restrictions. To do this, we present a method to generate new annotations on public available crowd datasets, with labels for people violating the social distance constrain. A comparison study suggests that our proposed approach is superior with respect to detect and track available solutions in scenarios where severe occlusions occur, and the people is observed from a far away perspective.

Finally, VSD is not the only solution to the social distancing control problem, other interesting ideas have been recently explored. For instance, in [24], the authors propose to use Internet of Things (IoT) technologies, to send GPS (Global Positioning System) coordinates from personal mobile phones to detect social distancing violations, and send warning messages to users violating this restriction. Alternatively, in [25] the authors present an interesting work using a legged robot equipped with multiple cameras and a 3D range laser sensor, to estimate the inter-personal distance of people around it, and interact with them by sending human-friendly messages to persuade them to keep an appropriate social distance.

3 Problem statement

In this paper, the objective is the automatic detection of groups of people non conforming with the social-distance, as seen in Fig. 2. For this matter, we consider a set of fixed cameras $C = \{c_1, c_2..., c_n\}$ each having a body reference frame $\mathbf{F}_{c_i}$ where $i \in |C|$. The cameras are pointed to the same scenario with a global reference frame $\mathbf{F}_w$, from different perspectives. The crowd appears located in $\mathbf{F}_w$, but we are only interested in the heads' location, since the head is the most visible part of the body given a highly occluded scenario [14]. In that regard, we make the predictions in the head's plane P located in the frame of reference $\mathbf{F}_w$ with the center at coordinates $P_0 = (0, 0, h_h)$, where h_h is the average height of a person. Since the images produced by the cameras operate in the image plane I_i, we need to transform the images to the global reference frame in order to know the distance between each person. With this goal, we define two transformations; $\mathbf{T}_w^{c_i}$ the transformation from the camera frame $\mathbf{F}_{c_i}$ to the global frame $\mathbf{F}_w$, better known as the extrinsic camera parameters, and the transformation from the image plane I_i to the camera frame of reference $\mathbf{F}_{c_i}$, also referred as the intrinsic camera parameters $\mathbf{K}$.

Definition 1 (**Social-Distance Compliance (SDC)**) Let us define $H = \{\mathbf{h}_0, ..., \mathbf{h}_n\}$ as the set of the n people present in the scene, where $\mathbf{h}_i = [x_i, y_i, z_i]$ represents the i-th person's location in the global frame $\mathbf{F}_w$. Then we can establish the social-distance d_i for a person i as the minimum inter-personal Euclidean distance $\|\cdot\|$ with respect to any other person in the scene, it is $d_i = \min_{j \neq i}(\|\mathbf{h}_i - \mathbf{h}_j\|), \forall j \in [1, ..., n]$. A person is considered to be in compliance with social-distancing (SDC) if and only if its social-distance is bigger than a security threshold d_t (normally around 2 meters), that is if $d_i > d_t$, and is considered not in compliance (NSDC) otherwise.

Then, the main goal is to develop computer vision algorithms using deep learning, in order to detect those groups of people from video streams which are not in compliance with the social-distance constrain (NSDC) (see Fig. 2). To do so, in the following section we describe a novel approach based on DNN segmentation.

4 Proposed approach

By applying the coordinate transformations, we can project the ground truth head annotations from public available crowd datasets to the head's plane P and remove the head labels that are in accordance with the social-distance constrain. With these new ground truth annotations, we can generate both density maps and segmentation models to train a DNN in the head's plane P or directly in the image plane I.

In the following subsections, we will describe in detail the steps to generate the ground truth from the crowd counting databases and the training procedures for the density map generator and the segmentation algorithm, for NSDC density maps.

4.1 Ground truth annotations

In crowd counting, the most common annotations are the coordinates at the center of the visible part of the head in an image, since given an extremely dense crowd, it is the most visible part of the people [14]. By itself, this kind of annotation is not useful for detecting NSDC crowds since they do not provide the position of a person with respect to each other. Knowing this, we project the annotations to the head's plane P. Thus, having the transformations $\mathbf{T}_w^{c_i}$ and the intrinsic camera parameters $\mathbf{K}$, the projection of the head annotation in the image plane $\mathbf{a}_I = (x_I, y_I, 1)$ onto the annotation in the head's plane $\mathbf{a}_P = (x_P, y_P, h_h, 1)$ is given by:

$$\begin{bmatrix} x_P \\ y_P \\ h_h \\ 1 \end{bmatrix} = \lambda(\mathbf{K}\mathbf{T}_w^{c_i})^{-1} \begin{bmatrix} x_I \\ y_I \\ 1 \end{bmatrix}, \qquad (1)$$

where λ is a scale factor. The cases where the expression $\mathbf{K}\mathbf{T}_w^{c_i}$ is invertible are discussed in [26]. Since each camera in C has a different point of view on the same scene, the redundant annotations coming from the multiple views of the same scene can be used to adjust the annotations' position, and add new annotations that are not visible by the other cameras, similar to the process described in [22]. Once we have all the annotations in the head's plane $\mathbf{a}_P$, we manually remove all the people that are correctly following the social-distancing. In other words, we remain only with the annotations $\mathbf{a}_P^* \subset \mathbf{a}_P$ which are NSDC, as stated in Definition 1.

Now, we describe how the annotations are used to generate the ground truth density maps and ground truth segmentation for training.

4.2 Density map generator

Commonly, the DNN would not be able to learn directly from the head annotations without a pre-processing stage [14]. Hence, in this work we use a Gaussian kernel to blur the head annotations in order to cover features from all the head, in either the image plane I or the head's plane P. The result is known as a density map D_n, which contains the location and the number of people in an image. The density maps cover more features of the people's heads, making it a more suitable learning objective compared with the single point annotations. To learn how to generate these density maps D_n, we use the Late Fusion algorithm from [22]. It is composed by two DNN, and the sampler module from the Spatial Transformers Network [27]. The first DNN is a 7 layers Fully Convolutional Network (FCN_7), which is used to generate density maps in the image plane. For each camera in C, a FCN_7 is trained. Once the density maps are generated for all the cameras, the density maps are projected to the ground plane using (1), and the sampler module from [27]. The projected density maps are normalized and concatenated in a single tensor to be fed into the Fusion DNN. The module learns to fuse the projected density maps and remove the deformation caused by the projection [22]. Once the full DNN is trained, a bird-eye-view density map of the scene can be generated as the projected density map in the head's plane P. From the projected density map we obtain a mask as a visual indicator in where the NSDC crowds are located. Then, since we also have the number

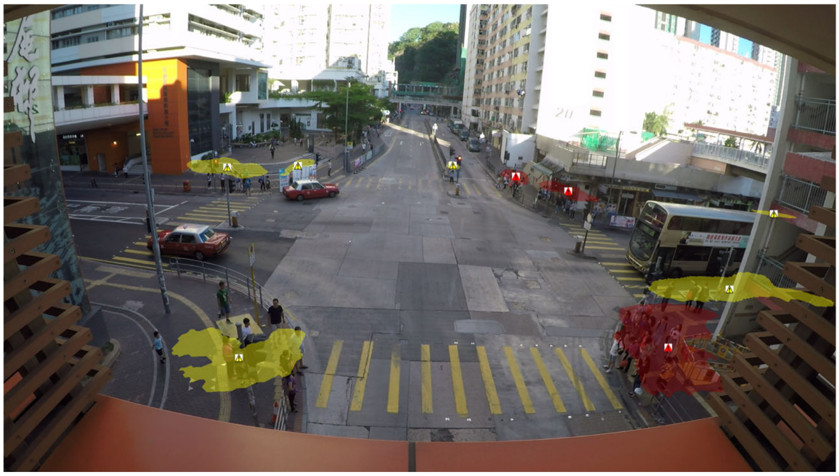


Fig. 1 Automatic monitoring social-distance in wide public areas using density maps

of people in a region of the image provided by the density map, hereafter referred as the people count, we classify the crowds with a risk level and assign a "Danger" or "Warning" label, like in Fig. 1. Finally, we further purge the detection by removing the masks that have less than a threshold number of detected people per area.

4.3 Crowd segmentation

Generating density maps involves two tasks in one, that is, while a DNN is training, it is learning how to count people and where the crowd is located in the image plane I or the head's plane P. An alternative approach to increase the detection accuracy of any DNN is to only train them to localize the crowd. In that regard, we also propose the use of image segmentation to localize the crowds non conforming with the social-distance in the image plane I. We employ the ground truth density maps in the head's plane P obtained in the previous stage, in order to generate the ground truth segmentation. As a first step, we normalize the density map values and remove all the values of the density map below a threshold t_s, then we use the closing morphological transformation in order to create a single segmentation with no gaps between NSDC crowds subgroups. Finally, we project the segmentation in the head's plane back to one of the cameras in the set C using the inverse of (1).

The architectures considered to learn the NSDC crowds are FCN_7 [22] and U-Net [23]. Despite FCN_7 not being designed for segmentation, but to produce density maps, it is still suitable for the segmentation task. FCN_7 produces its output with only high level features, hence reducing the overall visual quality of the segmentation. Therefore, the ground truth segmentation has to be at the same resolution as the output of the DNN. We use FCN_7 as is given by Zhang and Chan [22].

Alternatively, we also tested the U-Net architecture in the crowd segmentation task. This choice comes from its decoder-encoder architecture, that allows to use the ground truth segmentation at the same resolution as the input in the training stage. This produces a better defined segmentation, while, in theory, improving the precision. The trade-off, compared with FCN_7, is an increase in inference and training times.

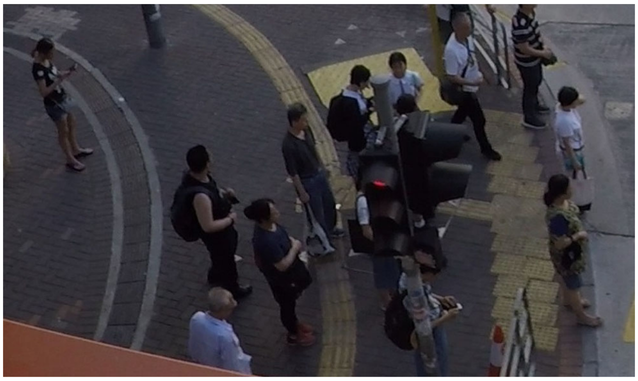


Fig. 2 Example of an urban scene where a crowd is not in compliance with social-distancing (NSDC). Only the person farthest to the left is respecting social-distancing

5 Training stage

In this section, we provide the technical details used for training the DNN for the task of detecting NSDC crowds and the metrics to compare the overall performance.

5.1 Metrics

First, we will discuss the metrics used to evaluate the methods in their respective tasks, crowd counting and segmentation. Afterwards, we will discuss the metrics for evaluating the performance of the task of interest, detecting NSDC crowds. These metrics shall allow us to compare our different approaches, such as density maps and segmentation, to solve the VSD problem.

For the task of crowd counting, Mean Average Error (MAE) and Mean Square Error (MSE) are the most commonly used evaluation metrics [28]. MAE and MSE are defined as follows:

$$MAE = \frac{1}{Q}\sum_{q=1}^{Q}|N_q - \hat{N}_q| \tag{2}$$

$$MSE = \sqrt{\frac{1}{Q}\sum_{q=1}^{Q}|N_q - \hat{N}_q|^2} \tag{3}$$

where Q is the total number of images in the set, N_q is the ground truth people count, $q \in [1, ..., Q]$, and $\hat{N}_q$ is the predicted total number of people for the image q. MAE is used to evaluate the total people count in the image while MSE highlights big errors, thus MSE is usually bigger than MAE.

For the segmentation task, the *Dice* score is often used to evaluate the trained models. It evaluates the similarity of the predicted segmentation and the ground truth segmentation, by calculating the ratio of the size of the overlap between the predicted segmentation and the ground truth segmentation, divided by the total area of both segmented regions. More formally, the *Dice* score is defined as:

$$Dice = \frac{2 * TP}{2 * TP + FP + FN} \tag{4}$$

where TP are the true positives, FP are the false positives and FN are the false negatives, all of them measured pixel-wise in the segmentation problem.

All of this metrics are sufficient to be used to evaluate their respective task, but by themselves they do not answer the question on how good are these methods at detecting the NSDC crowds while not detecting the SDC people. In this regard, we use the ground truth density maps of conforming D_c and non conforming D_n crowds to get how many people were correctly classified. More formally, we compute the pixel-wise TP, FP, True Negative (TN) and FN as follows:

$$TP = \hat{M} \cdot D_n \tag{5}$$

$$FP = \hat{M} \cdot D_c \tag{6}$$

$$TN = \hat{M}^{-1} \cdot D_c \tag{7}$$

$$FN = \hat{M}^{-1} \cdot D_n \tag{8}$$

where $\hat{M}$ is the predicted segmentation region such that $\hat{M}_{i,j} \in \{0, 1\}$, and $\hat{M}^{-1}$ is the function returning the pixels not predicted as NSDC. Having defined our TP, FP, TN and FN, we can use the traditional definitions of precision, recall, sensitivity and F1 score. Precision, recall and F1 are used to compare the methods based on how well they captured the NSDC crowds in the scene, and sensitivity for how well they do not wrongly classified the SDC crowds as NSDC.

5.2 Datasets

The datasets used for this paper are CityStreet [22] and PETS2009 [29]. **PETS2009** is a multi-view dataset designed for multiple tasks including crowd counting. In this dataset, people were told how to move and position themselves in order to challenge the solutions for the different tasks for which the dataset was conceived. In average, each frame contains 20 people per frame. The dataset is composed of a total of 8 different views, but only three are considered for the present work. 794 images with a resolution of 576 × 768 pixels, are used for the purposes of this paper [29].

On the other hand, **CityStreet** is a multi-view crowd counting dataset from which 385 annotated images are used to train the solutions here proposed. The dataset is taken from an uncontrolled urban environment where the crowd moves at will, with a total people count between 50 to 100 people per frame. The images have a resolution of 1520 × 2704 pixels, which we down sample to 480 × 848 for our experimentation. Both datasets provide information about the camera pose.

5.3 Density maps generators

In order to train the density map generator for our first solution, we need density maps D_n of NSDC crowds in both, the head's plane P and the image plane I, for each camera in C. For that matter, we set the average head's position to $h_h = 1.75m$ [22]. Next, to separate the SDC head annotations from the NSDC, we used a social-distance threshold $d_t = 2m$ for the two crowd datasets, the CityStreet [22] and the PETS2009 [29], both offering challenging scenarios in open urban areas, allowing us to test against different levels of occlusion. Furthermore, both datasets include information about the camera pose, facilitating the recovery of the scale λ, and the homography transformation between planes as stated in (1). Once the safety threshold d_t is defined in meters, we project the head annotations to the head's plane P using (1), and calculate the inter-personal distances between each head annotation, disregarding those that are SDC, and keeping only NSDC annotations. After we have separated the head annotations, we produce the density maps D_n^P in the head's plane P by applying a Gaussian kernel on the head annotations, in order to increase the number of head's features for the network to learn. Note that it is desirable for the kernel size to be roughly the size of the head, in order to better characterize it. The size of the kernel is 5 with a variance $\sigma = 15$ for the CityStreet dataset, and a Gaussian kernel of size 4 with a variance $\sigma = 15$ is used for the PETS2009 dataset, assuming that the head size, or kernel size, is on average the same across all the scenes in real world coordinates. Adaptive kernels can be used instead in order to relax this assumption [14]. For our segmentation approach, we generate the NSDC density maps in the image plane D_n^I using a Gaussian kernel of size 10 with a variance $\sigma = 30$ for the CityStreet dataset, and a Gaussian kernel of size 4 with a variance $\sigma = 15$ for the PETS2009 dataset, selected to generate filled blobs for each NSDC group of people, reducing the gaps between each group.

In the first stage, a FCN_7 is trained for each camera in C, where the cardinality is set to $|C| = 3$, for both datasets. After a careful tuning, we set the learning rate $lr = 0.001$ using the Adam optimizer during 150 epochs. Next, we freeze all the FCN_7 DNN and train only the Fusion DNN with a learning rate $lr = 1e^{-4}$, using the Adam optimizer during 150 epochs, reducing the learning rate in case of plateau in the performance validation each 10 epochs, with a patience of 1 and a minimum learning rate $min(lr) = 5e^{-5}$. Finally, we perform fine-tuning in the Late Fusion DNN by unfreezing the FCN_7 models with a learning rate $lr = 5e^{-5}$, using the Adam optimizer during 150 epochs, reducing the learning rate in case of plateau in the performance validation each 10 epochs with a patience of 0, and a minimum learning rate $min(lr) = 5e^{-6}$. All of this hyper parameters were selected empirically, and are the same for both datasets. Refer to Table 1 for a summary of the hyper parameters used to train the density map approach.

At inference time, the predicted density map $\hat{D}$ is normalized. Then, to generate the predicted segmentation $\hat{M}$, we saturate to 1 all the pixels values that are above a threshold equal to $\frac{20}{255}$, and set them to 0 otherwise. Thereafter, we select the masks that contain a people count estimate bigger than 0.5 and 2 people for the Citystreet and PETS2009 dataset respectively.

5.4 Segmentation

For the segmentation task, we use the NSDC density map in the head's plane D_n, to create our segmentation in the image plane I. First, we normalize the density maps and set all the non-zero pixel values to 1, to obtain a binary mask, since for segmentation we do not require the people count. Then, we apply a morphological dilation transformation with a 7×7 ones matrix kernel $\mathbf{1}^{7\times 7}$, and pass it trough the density map 2 times. Next, we use the morphological erosion transformation with a kernel equal to a ones matrix $\mathbf{1}^{4\times 4}$ for the CityStreet dataset, and $\mathbf{1}^{5\times 5}$ for the PETS20009 dataset, also applying it trough the density map 2 times. This pre-processing transformations were found empirically and allow us to obtain uniform blobs, removing small gaps between them. Finally, we project this segmentation mask back to the image plane I.

Table 2 Comparison results between the different proposed methods in the CityStreet dataset

Method	Precision	Recall	Specificity	F1
Density map	0.889	0.690	0.743	0.777
FCN_7	0.882	0.730	0.728	0.799
U-Net	0.888	0.748	0.728	0.812

We train the FCN_7 and U-Net models using the Adam optimizer during 150 epochs, reducing the learning rate in case of plateau in the performance validation, each single epoch with a patience of 3 and a minimum learning rate $min(lr) = 1e^{-8}$ for both datasets. As for the learning rate, we set it to $lr = 5e^{-4}$ and $lr = 0.001$ for the FCN_7 and U-Net models respectively.

At inference time, since the segmentation per pixel is given as a value between 0 and 1, we saturate all the values above a threshold equal to 0.3 for the CityStreet dataset, while for the PETS2009 dataset the best results were obtained by thresholds of 0.6 and 0.9, for the FCN_7 and U-Net respectively. Refer to Table 1 for a summary of the hyper parameters used to train both segmentation approaches.

6 Results and discussion

In Tables 2 and 3, we present the quantitative results for all the approaches proposed in this article, on both datasets CityStreet and PETS2009. All the algorithms were trained and evaluated using a computer equipped with a processor Intel Core i7 9750H paired with a Nvidia RTX 2070 Mobile GPU.

For the CityStreet dataset, we can appreciate in Table 2 that both FCN_7 and U-Net trained for NSDC crowd segmentation have similar performance, having the U-Net

Table 1 Hyper parameter selection

Method	Learning rate	Epochs	Fine tuning	Threshold
Density map	0.001, Fine tuning: $1e^{-4}$	300	Yes	$\frac{20}{255}$
FCN_7	$5e^{-4}$	150	No	Citystreet: 0.3, PETS2009: 0.6
U-Net	0.001	150	No	Citystreet: 0.3, PETS2009: 0.9

Table 3 Comparison results between the different proposed methods in the PETS2009 dataset

Method	Precision	Recall	Specificity	F1
Density map	0.947	0.575	0.614	0.716
FCN_7	0.910	0.677	0.4780	0.776
U-Net	0.961	0.761	0.618	0.849

as the best overall. We are looking for the method that has the highest F1 score without leaving the Specificity behind. Although the density map approach does not stay behind the U-Net and FCN_7 in both Precision, Specificity and F1, it is the worst at recalling all the NSDC people inside a scenario, which can be exemplified in Fig. 3f where it does not detect the NSDC crowd at the center of the image.

Indeed in Fig. 3 we can observe the results of three different scenarios from the CityStreet dataset, using the 3 presented methods. The density map colors with yellow and red the regions where crowds violating the social-distance constrain are found, according to the risk of contagion given by the people's concentration per pixel, yellow for "*warning*" and red for "*danger*". We can observe that FCN_7 and U-Net performed almost equally as depicted on Table 2, each having better performance in different situations. For example in Fig. 3c we see that the two people at the lower right were labeled as NSDC, while in Fig. 3d only one person is partially detected. On the other hand, in Fig. 3k the FCN_7 model mistakes part of the ground at the left as a NSDC crowd, while in Fig. 3l this effect is mitigated.

More in detail, in Fig. 4 we can observe a zoomed image of the same scenario, from where it is clearer how the U-Net performs better at detecting the three people at the center of the image as SDC, while making the same FP mistakes as the FCN_7 model with the isolated people at the bottom.

For the PETS2009 dataset, we observe from the results on Table 3 that U-Net showed the best performance overall, while the density map approach yielded a good result in Specificity. This could be due to the number of examples of SDC people being considerable lower with respect to the NSDC people in the PETS2009 dataset, as seen in Fig. 5, making the task more challenging. For example, in Fig. 5c, g and k, the FCN_7 segmentation wrongly detects at least one conforming person as non conforming, while almost all the NSDC crowds are correctly segmented as non conforming. In our density map approach, we can see that it performs better at not classifying conforming people, although, as seen in Fig. 5j it has some problems at classifying all the NSDC people. As for the U-Net model trained for segmentation, we can encounter the best balance between correctly classifying NSDC crowds, having some minor errors around the SDC people, as seen in Fig. 5h, l, mainly due to being segmented as NSDC with low probability but removed by the threshold, leaving only the ones with higher probability.

Also, in Fig. 6 we observe more in detail a zoomed frame from the PETS2009 dataset. From there, it can be

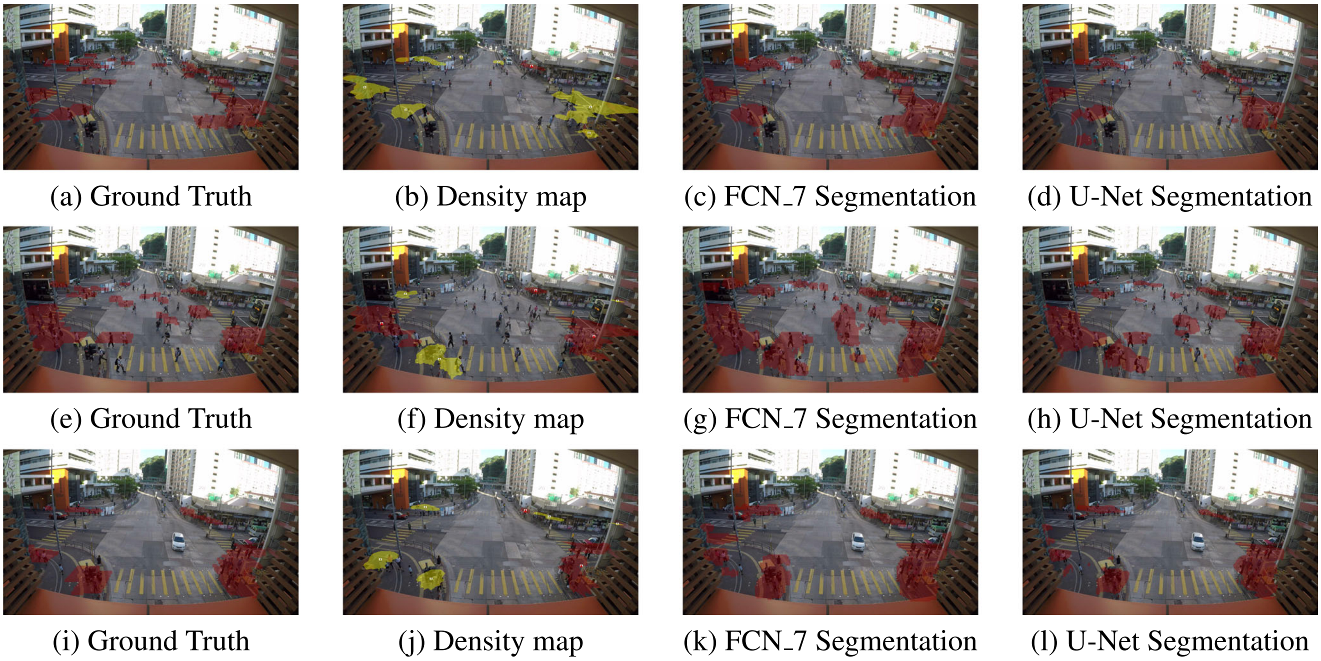
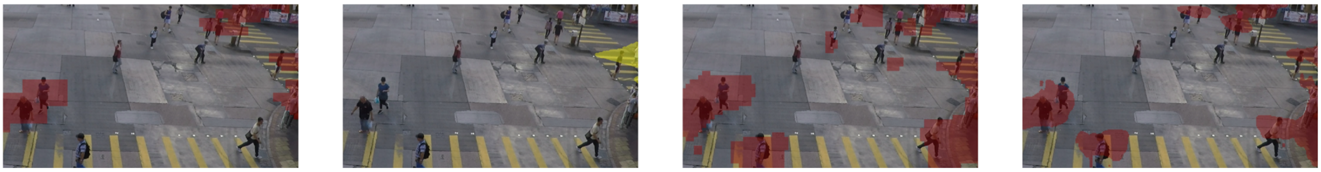
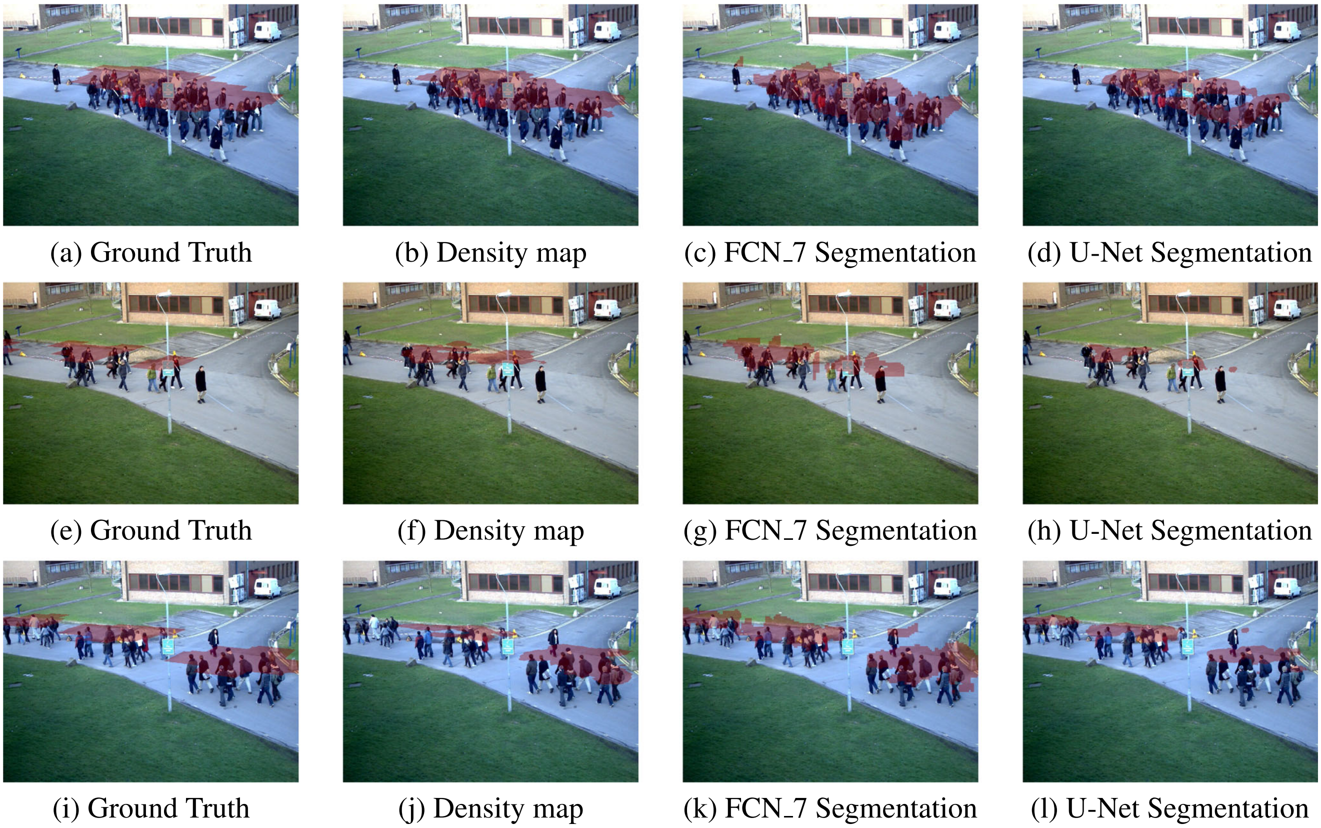

(a) Ground Truth (b) Density map (c) FCN_7 Segmentation (d) U-Net Segmentation
(e) Ground Truth (f) Density map (g) FCN_7 Segmentation (h) U-Net Segmentation
(i) Ground Truth (j) Density map (k) FCN_7 Segmentation (l) U-Net Segmentation

Fig. 3 Results of the detection of Non Social-Distance Conforming crowds (NSDC) in the CityStreet dataset. We can see that the Density map based approach tends to under estimate the non conforming crowds mostly at the center of the scene. Both FCN_7 and U-Net perform similarly, with U-Net having the edge



(a) Ground Truth (b) Density map (c) FCN_7 Segmentation (d) U-Net Segmentation

Fig. 4 Zoomed images from the CityStreet dataset, here we can see that U-Net performed the best out of the three approaches in this scenario, despite of some False Positive regions

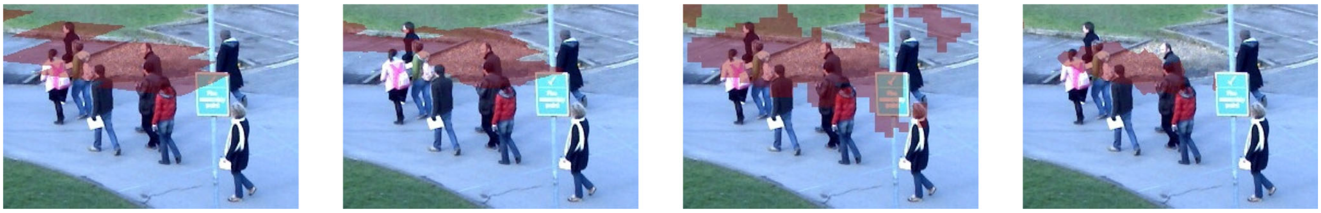


(a) Ground Truth (b) Density map (c) FCN_7 Segmentation (d) U-Net Segmentation

(e) Ground Truth (f) Density map (g) FCN_7 Segmentation (h) U-Net Segmentation

(i) Ground Truth (j) Density map (k) FCN_7 Segmentation (l) U-Net Segmentation

Fig. 5 Results of the detection of Non Social-Distance Conforming crowds (NSDC) in the PETS2009 dataset. U-Net achieves the best visual results followed by the density map and FCN_7 segmentation



(a) Ground Truth (b) Density map (c) FCN_7 Segmentation (d) U-Net Segmentation

Fig. 6 Zoomed images in the dataset PETS2009. The density map and U-Net segmentation are able to detect fairly well the crowds violating the social-distance, while FCN_7 trained for segmentation tends to over estimate the location of the crowd

seen that the density map and U-Net approaches correctly classified in CityStreet all the NSDC people, failing only with two people, while the FCN_7 model over estimates the segmentation and leaves artifacts around the conforming people.

Finally, we show a video using the U-Net model trained for segmentation over various video sequences from the PETS2009 dataset, excluding the ones used for training. The video can be found at: https://youtu.be/TwzBMKg7h_U.

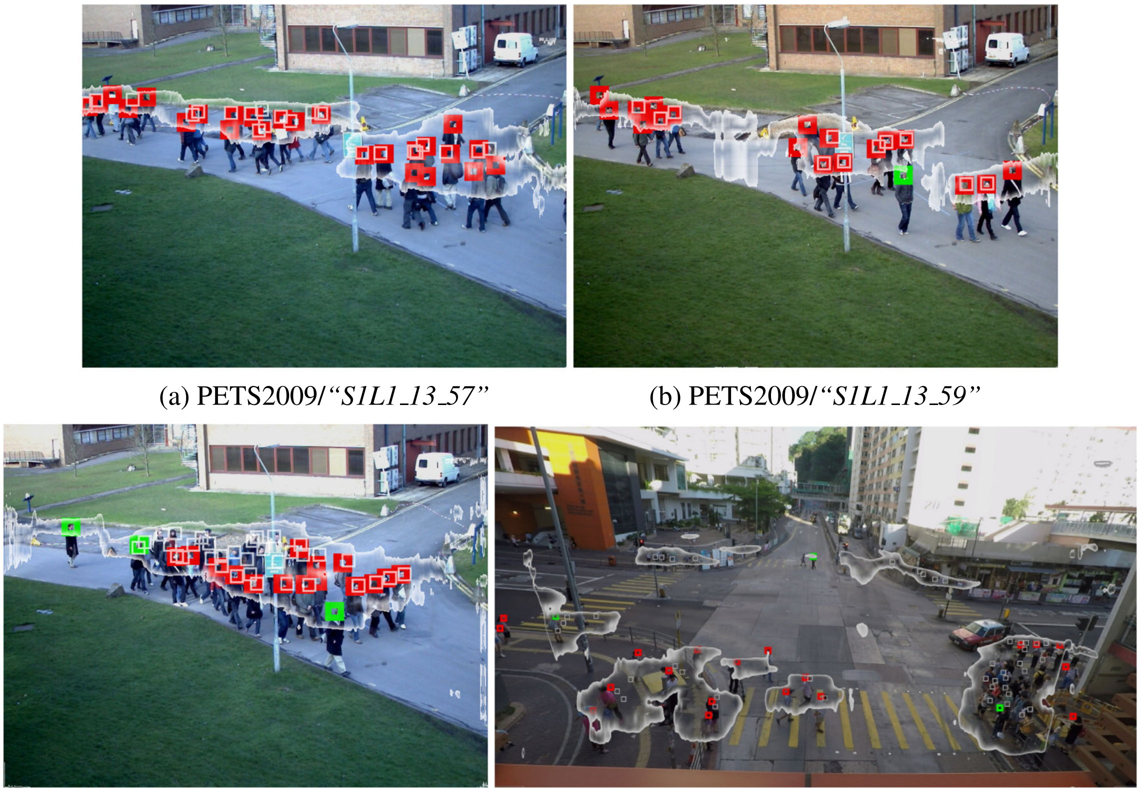
6.1 Comparison to object detection techniques

As previously described, most of the currently available approaches are based on the idea of first performing object detection and then measuring the social distance, making difficult an apples to apples comparison to the work here proposed. However, by making some concessions, we aim to make a fair comparison by putting our model in terms of object detection, and evaluating how well different approaches are able to find NSDC people in terms of Average Precision (AP) [30].

Accordingly, we tested our best solution, the U-Net segmentation, against DeepSOCIAL [1] and Yang et al. [2], two prominent models in the literature for VSD, and to our knowledge the only ones that provide openly available solutions. The evaluation was done in both, the *CityStreet* dataset, as well as in three scenarios from the PETS2009 test set, namely *"S1L1_13_57"*; *"S1L1_13_59"*; *"S1L2_14_06"*, where different kinds of people distribution and different levels of occlusion are observed, as depicted in Figs. 7 and 8.

Our main interest is not on the exact location of people breaking social distancing rules, but in general areas where this happens. This has the advantage of mitigating privacy concerns, while finding problematic areas (such as crosswalk bottlenecks), where procedures can be taken to alleviate pedestrian congestion. Being so, we take leniently the location of the head bounding boxes for both approaches as follows:

In the cases of [1] and [2], we consider head bounding boxes of 20 × 20 pixels, same as we do for the ground truth data, at the center-top of the bounding boxes of the whole person's body, which the algorithm has found to be



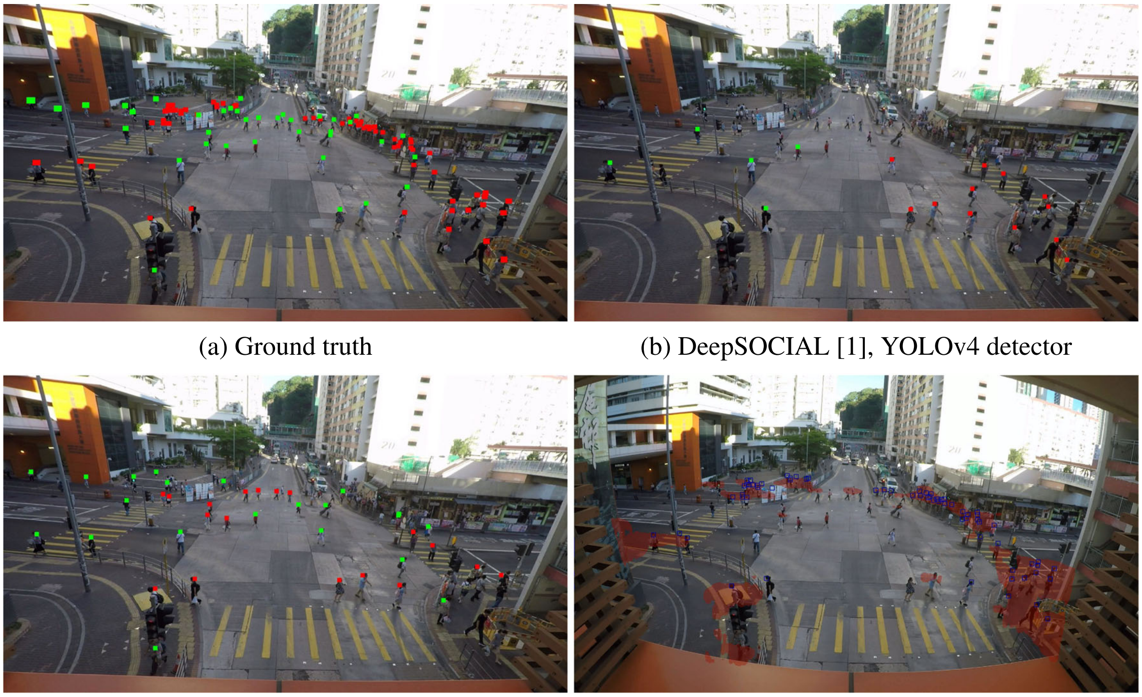
(a) PETS2009/*"S1L1_13_57"*

(b) PETS2009/*"S1L1_13_59"*

(c) PETS2009/"*S1L2_14_06*"

(d) "*CityStreet*"

Fig. 7 Comparison analysis between a detector-based solution (DeepSocial) against our segmentation-based one, for VSD under different scenarios. For DeepSOCIAL, the people detected that do not comply with social distancing are shown in red bounding boxes, while those that do comply are presented in green bounding boxes. U-Net segmentation is overlapped in gray blobs. **b** shows relatively sparse pedestrians, where DeepSOCIAL performs well, but there are considerable more occlusions in **c** affecting the detector performance, illustrating a key benefit of the U-Net segmentation. **d** shows a challenging urban scenario with smaller people's instances, subject to sever occlusions, resulting in a poor performance with detector-based approaches


(a) Ground truth

(b) DeepSOCIAL [1], YOLOv4 detector

(c) Yang et al. [2], Faster R-CNN detector

(d) U-Net segmentation

Fig. 8 Comparison study between different solutions for the NSDC crowds detection in a challenging urban scenario from the *CityStreet* dataset. **a** shows the labeled ground truth, where the green bounding boxes show people respecting the social distancing (SDC) and the red bounding boxes represent people violating the social distance constrain (NSDC). **b** depicts the results obtained by DeepSOCIAL [1], whereas (**c**) presents the results from Yang et. al. [2]. U-Net Segmentation is presented in (**d**) as a red cloud

breaking social-distancing. This is done in the image plane I. Then, we project the ground truth annotations to the image plane I, as obtained in Section 4, and generate the ground truth NSDC head bounding boxes. Then we measure AP, using a very low Intersection over Union threshold ($\mathrm{IoU}(A, B) = \frac{A \cap B}{A \cup B}$). The AP is plotted in Fig. 9 for four test videos using an IoU $= 0.1$. Blue bars represent the results obtained by DeepSOCIAL, while the performance of Yang et. al. is shown with orange bars. The upper error margin shows the same metric using an IoU $= 0.01$, which makes a slight improvement. The lower error margin is the AP with a IoU $= 0.3$, empirically showing that even though the bounding boxes are of an arbitrary size, their estimation is in the correct region.

Provided that the *Citystreet* dataset is subject to important distortions induced by the fish-eye lenses, an appropriate behavior for the approaches using a detection stage ([1, 2]) requires to make a lens distortion correction pre-processing. Otherwise, the interpersonal distance computation gets considerably affected, specially towards the corners of the frame. Figure 8a–c are examples of the undistorted frames, whereas Fig. 8d shows a frame as obtained from *Citystreet*, with fish-eye lens distortion.

In order to get a fair AP evaluation for the DeepSOCIAL approach [1], we bypassed its tracking algorithm in the *Citystreet* dataset, because in this dataset the frames are randomly sampled from a video stream, without a fixed frame rate, handicapping the advantages of using tracking techniques and affecting the final solution. Hence the Simple Online and Real-time (SORT) [31] tracking algorithm is bypassed by giving it very lenient parameters (given that the full source code for DeepSOCIAL is not accessible); the minimum number of associated detections before tracking is initialized equal to zero, with an IoU threshold equal to 0.01, disregarding the tracking and remaining only with the detection.

Meanwhile, for our U-Net segmentation approach, a segmentation map is obtained in the image plane I, and normalized between [0, 1]. Using the ground truth bounding boxes location, also in the image plane I, we only consider the detection as a true positive if the mean value of all the segmented pixels inside a ground truth bounding box is above a given threshold. In Fig. 9 (yellow bars), and its accompanying Table 4, this threshold is set to 0.85. The lower and upper error margins represent the same metric, with a threshold of 0.9 and 0.8, respectively.

The quantitative results depicted in Fig. 9 along with Table 4, showcase the superior performance of our segmentation-based solution for these scenarios in terms of AP, for the NSDC detection problem. Please note that

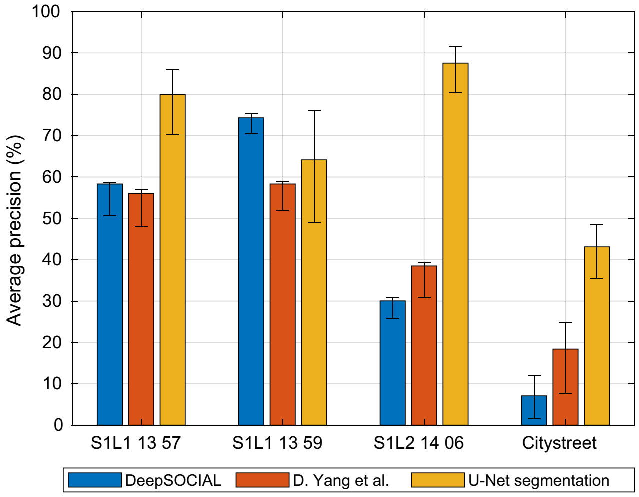


Fig. 9 Average precision (AP) performance in the NSDC detection problem, for DeepSOCIAL [1], using the YOLOv4 object detector (in blue), Yang et al. [2], using Faster R-CNN (in orange), and our U-Net Segmentation solution (in yellow), using the adaptations described in Section 6.1. For the test videos *PETS2009 View 001*; *"S1L1_13_57"*, *"S1L1_13_59"*, *"S1L2_14_06"* and *"CityStreet"*, respectively. The present study suggests that our proposed Segmentation approach outperforms instance-wise detection strategies on both datasets, particularly in wider scenarios as the ones presented in *CityStreet*, or when larger occlusions are present in PETS2009

we are not measuring the AP performance of the object detector, but how well the solution finds people violating the social distance constrain, hence, a few people wrongly detected may result in important errors for the NSDC AP. Also, in Fig. 7 we present an example of the performance of our segmentation-based solution against DeepSOCIAL for each case study scenario, each showing different people sparsity and level of occlusion. People detected as NSDC by DeelpSOCIAL are presented with red bounding boxes, while SDC people are shown with green boxes (although we are only evaluating those who are not in compliance). U-Net segmentation is presented as gray blobs. As seen in Fig. 7b, in the video sequence *"S1L1_13_59"* pedestrians are relatively sparse, consequently, instance-wise detection approaches, such as [1] and [2], detect most of them properly. But in the sequences *"S1L1_13_57"* (Fig. 7a) and *"S1L2_14_06"* (Fig. 7c) with considerable more occlusions, the detectors tend to fail substantially more. Furthermore, Fig. 8 shows a comparison between the three solutions over a challenging urban scenario from *CityStreet*, with a lot of people moving at will, and subject to important occlusions. Although the object detectors perform fairly well in the *CityStreet* dataset for the people detection task (see video linked bellow), DeepSOCIAL under-performs in the NSDC detection task mostly in this dataset, partially due to its adherence to the tracking algorithm used, but mainly due to missed detections caused by occlusions and far away instances. Moreover, both DeepSOCIAL and Yang et al. use complete body detectors, whilst the social distance should be measured from head to head, resulting in inaccuracies on the interpersonal distance computation, which may become important depending on the people's pose, hence leading to errors in the NSDC classification. Nevertheless, both DeepSOCIAL and Yang et al. have the advantage of providing concrete bounding boxes of the people, as well as tracking identifications of the people breaking social distancing rules, which may come handy for some applications.

Overall, our segmentation solution appears to perform better against difficulties such as higher occlusions and smaller instances, even performing better where there are a lot of people than in cases where there are only a few. In conclusion, this study suggests that our proposed solutions are better suited for this kind of scenarios, where detectors tend to fail due to occlusions, perspective variations and size of the "person" instances relative to the image. A video showcasing the performance of the three

Table 4 Quantitative comparison results in terms of Average Precision (AP) (also shown in Fig. 9), for 3 scenarios of the PETS2009 dataset and one of the *CityStreet dataset*. Bold characters highlight the best result for each dataset

Video input	Average precision, AP		
	DeepSOCIAL [%]	Yang et al. [%]	U-Net [%]
"S1L1_13_57"	58.31	56.01	**79.92**
"S1L1_13_59"	**74.31**	58.31	64.15
"S1L2_14_06"	30.04	38.49	**87.54**
PETS2009 overall	55.56	51.53	**76.50**
Citystreet	7.08	18.39	**43.11**

AP is obtained with an IoU threshold of 0.1, DeepSOCIAL is used with YOLOv4 at a resolution of 608 × 608 while Yang et al. uses the Pytorch default implementation of Faster R-CNN with ResNet-50 (Residual Network). The *"S1L2_14_06"* segment of the PETS2009 dataset presents the most crowded case, where object detection implementations struggle the most. While in the *"S1L1_13_59"* people are more dispersed and can be found well through these methods. Overall, the U-Net segmentation performs better than those approaches based on object detectors approaches in these scenarios is provided at https://youtu.be/XuQU-zaHMXE

6.2 Discussion

Although the proposed strategies showed promising results, they are still subject to some limitations, suggesting that the best solution would depend on the particular scenario, where low levels of occlusion with larger object instances in the image would benefit object detectors, while larger scenarios with higher levels of occlusion and smaller people instances would be better suited for end-to-end solutions like the ones proposed in this paper.

On the other hand, provided that our proposed solutions rely only on head's features, an upper view is required to work better, while traditional detectors should work also with frontal views, although both strategies would struggle with occlusions. Nonetheless, this should not be an issue given that aerial drones and most security cameras offer views from above the head. In particular, our approach requires a tilt angles between 20° and 90°, measured from the horizontal plane. We have observed good performance working around a tilt angle of 45°, at least at $5m$ away from the crowd in order to cover a larger area. Different particular scenarios, or different camera setups, may require further tuning to improve the performance, for example using transfer learning.

Another limitation, on the majority of the available solutions, is the requirement of knowledge on the camera pose relative to the ground. This is not a problem most of the times, since cameras are normally fixed at a-priori known locations, or its pose can be recovered from proprioceptive sensors, such as in drones or even mobile phones. In case the pose of the camera is unknown, extra computation would be required to estimate the homography transformation between planes, for example as in [32].

Also, most of the available solutions rely on the assumption that people on the scene are about the same average height, and their heads lie around a common plane where the inter-personal distance can be computed. This may be a strong constrain in some scenarios, where the ground is not flat or it is irregular; when some people are sitting while others are standing; or when children are mixed with adults. Nonetheless, we believe that the proposed solutions will still provide acceptable results for monitoring applications, provided that we have trained our algorithms using multi-view fusion which helps to capture depth information between the annotations, 3d solutions would probably be better suited for this scenarios, where there is not a good approximated common plane for the heads. However, 3d solutions will require special depth sensors such as stereo cameras or laser scanners, along with 3d annotations, further increasing the complexity of the solution and the effort required for its deployment.

An important aspect of our proposed framework is its compatibility with commonly available hardware in public spaces, where monitoring the social distancing may be of interest, given that they may be already equipped with security cameras looking the crowd from above, otherwise portable cameras mounted on drones can be used instead. Also, our proposed framework can be easily trained with any other of the multiple available datasets with people annotations, as long as the homography transformation is available or can be computed. Furthermore, our framework to create density and segmentation maps do not depend on the neural network algorithm, such that, in future works, we can propose alternative solutions to tackle the aforementioned limitations in our current approach, or update the DNN for segmentation according to the future state-of-the-art. Indeed, based on the results showed in this paper, the framework here proposed acts a solid foundation to the proposal of new algorithms based on density and segmentation maps.

In summary, the comparison analysis suggests that the proposed framework offers an interesting alternative for VSD monitoring, especially when larger occlusions and smaller people instances are present, where available detector-based solutions tend to fail. However, it would seem that the best overall solution would depend on the particular scenario, or a smart fusion of both approaches. Furthermore, coupling the VSD problem with mask usage and gaze detectors would help to better assess the risk of infection.

7 Conclusions and future work

In this work, we present a new framework to deal with the visual social distancing problem (VSD). Our framework proved to be useful at training Deep Neural Networks in the task of detecting non social-distance conforming crowds (NSDC), providing promising alternatives to the popular detect and track approach, specially in wider scenarios with more people, subject to important occlusions.

Using the proposed framework, we presented two different solutions to the visual social distancing problem in wide scenarios, a density-map-based, and a segmentation-based approach. Furthermore, we evaluated the performance of these approaches for three different networks, a FCN_7 density map generator, a FCN_7 segmentation and a U-Net segmentation, proving that solutions based on density maps or segmentation are capable of learning the notion of social-distance by providing the ground truth annotation of only the non-conforming crowds. Moreover, we found that the

U-Net segmentation showed the best performance out of the three strategies for both datasets, PETS2009 and CityStreet, achieving above 0.8 in the F1 score and above 0.6 in the Specificity score for both datasets. This is probably because it is a model better suited for the segmentation task. Meanwhile, the FCN_7 model trained to detect the NSDC crowds using density maps performed better than FCN_7 trained for segmentation in the PETS2009 dataset, possibly due to the lack of enough examples of SDC people.

Additionally, a comparison study was carried out between two state-of-the-art detection-based approaches and our U-Net segmentation solution, demonstrating better performance from our U-Net segmentation strategy in both studied datasets, specially in wide scenarios with high level of occlusion.

In future works, we aim at improving the results of our algorithms further evaluating other models. Also, we would like to provide more information about the distance in NSDC crowds in the loss function or directly in the model, and assign a level of risk accordingly. Another interesting axis of research would be to further detect the people's gaze and mask usage to better assess the risk of infection. Finally, it would be interesting to monitor these crowds using mobile cameras.

Funding This work was supported by the Mexican National Council of Science and Technology CONACYT, and the FORDECyT project 296737 "Consorcio en Inteligencia Artificial".

References

1. Rezaei M, Azarmi M Deepsocial: social distancing monitoring and infection risk assessment in covid-19 pandemic. Appl Sci, 10 (21). https://www.mdpi.com/2076-3417/10/21/7514
2. Yang D, Yurtsever E, Renganathan V, Redmill KA, Özgüner Ü A vision-based social distancing and critical density detection system for covid-19. Sensors, 21 (13). https://www.mdpi.com/1424-8220/21/13/4608
3. Shao Z, Cheng G, Ma J, Wang Z, Wang J, Li D (2021) Real-time and accurate uav pedestrian detection for social distancing monitoring in covid-19 pandemic. IEEE Transactions on Multimedia, 1–1. https://doi.org/10.1109/tmm.2021.3075566
4. Ahmed I, Ahmad M, Rodrigues JJ, Jeon G, Din S (2021) A deep learning-based social distance monitoring framework for covid-19. Sustain Cities Soc 65:102571. https://www.sciencedirect.com/science/article/pii/S2210670720307897
5. Ahamad AH, Zaini N, Latip MFA (2020) Person detection for social distancing and safety violation alert based on segmented roi. In: 2020 10th IEEE International conference on control system, computing and engineering (ICCSCE), pp 113–118
6. Gupta S, Kapil R, Kanahasabai G, Joshi SS, Joshi AS (2020) Sd-measure: a social distancing detector. In: 2020 12th International conference on computational intelligence and communication networks (CICN), pp 306–311
7. Hou YC, Baharuddin MZ, Yussof S, Dzulkifly S (2020) Social distancing detection with deep learning model. In: 2020 8th International conference on information technology and multimedia (ICIMU), pp 334–338
8. Zuo F, Gao J, Kurkcu A, Yang H, Ozbay K, Ma Q (2021) Reference-free video-to-real distance approximation-based urban social distancing analytics amid covid-19 pandemic. J Transp Health 21:101032. https://doi.org/10.1016/j.jth.2021.101032
9. Wojke N, Bewley A, Paulus D (2017) Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International conference on image processing (ICIP). IEEE, pp 3645–3649
10. Cristani M, Bue AD, Murino V, Setti F, Vinciarelli A (2020) The visual social distancing problem. IEEE Access 8:126876–126886
11. Srinivasan S, Singh RR, Biradar RR (2021) Covid-19 monitoring system using social distancing and face mask detection on surveillance video datasets. In: 2021 International conference on emerging smart computing and informatics (ESCI). https://doi.org/10.1109/esci50559.2021.9396783
12. Bhambani K, Jain T, Sultanpure KA (2020) Real-time face mask and social distancing violation detection system using yolo. In: 2020 IEEE Bangalore humanitarian technology conference (B-HTC), p. nil. https://doi.org/10.1109/b-htc50970.2020.9297902
13. Rakhsith L, Karthik B, D AN, V KK, Anusha K (2021) Face mask and social distancing detection for surveillance systems. In: 2021 5th International conference on trends in electronics and informatics (ICOEI). https://doi.org/10.1109/icoei51242.2021.9452973
14. Zhang Y, Zhou D, Chen S, Gao S, Ma Y (2016) Single-image crowd counting via multi-column convolutional neural network. In: 2016 IEEE Conference on computer vision and pattern recognition (CVPR). https://doi.org/10.1109/cvpr.2016.70
15. Ranjan V, Shah M, Nguyen MH Crowd transformer network, arXiv:1904.02774v1
16. Huynh V, Tran V, Huang C (2019) Danet: depth-aware network for crowd counting. In: 2019 IEEE International conference on image processing (ICIP), pp 3001–3005
17. Ma Z, Wei X, Hong X, Gong Y (2019) Bayesian loss for crowd count estimation with point supervision. In: Proceedings of the IEEE international conference on computer vision, pp 6142–6151
18. Wang B, Liu H, Samaras D, Hoai M (2020) Distribution matching for crowd counting. In: Advances in neural information processing systems
19. Wang Q, Gao J, Lin W, Li X (2020) Nwpu-crowd: a large-scale benchmark for crowd counting and localization. IEEE Trans Pattern Anal Mach Intell, 1–1
20. Bai S, He Z, Qiao Y, Hu H, Wu W, Yan J (2020) Adaptive dilated network with self-correction supervision for counting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4594–4603
21. Liu L, Chen J, Wu H, Li G, Li C, Lin L (2021) Cross-modal collaborative representation learning and a large-scale rgbt benchmark for crowd counting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 4823–4833
22. Zhang Q, Chan AB (2019) Wide-area crowd counting via ground-plane density maps and multi-view fusion cnns. In: 2019 IEEE/CVF Conference on computer vision and pattern recognition (CVPR), p. nil. https://doi.org/10.1109/cvpr.2019.00849
23. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 234–241
24. Ksentini A, Brik B (2020) An edge-based social distancing detection service to mitigate covid-19 propagation. IEEE Internet Things Mag 3(3):35–39. https://doi.org/10.1109/iotm.0001.2000138

25. Chen Z, Fan T, Zhao X, Liang J, Shen C, Chen H, Manocha D, Pan J, Zhang W (2021) Autonomous social distancing in urban environments using a quadruped robot. IEEE Access 9:8392–8403. https://doi.org/10.1109/access.2021.3049426
26. Liu W, Lis K, Salzmann M, Fua P (2019) Geometric and physical constraints for drone-based head plane crowd density estimation. In: 2019 IEEE/RSJ International conference on intelligent robots and systems (IROS). https://doi.org/10.1109/iros40897.2019.896785
27. Jaderberg M, Simonyan K, Zisserman A, Kavukcuoglu K (2015) Spatial transformer networks. In: Proceedings of the 28th international conference on neural information processing systems - volume 2, NIPS'15. MIT Press, Cambridge, pp 2017–2025
28. Kang D, Ma Z, Chan AB (2018) Beyond counting: comparisons of density maps for crowd analysis tasks—counting, detection, and tracking. IEEE Trans Circuits Syst Video Technol 29(5):1408–1422
29. Ferryman J, Shahrokni A (2009) Pets2009: dataset and challenge. In: 2009 Twelfth IEEE International workshop on performance evaluation of tracking and surveillance, pp 1–6
30. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338
31. Bewley A, Ge Z, Ott L, Ramos F, Upcroft B (2016) Simple online and realtime tracking. In: 2016 IEEE international conference on image processing (ICIP). IEEE, pp 3464–3468
32. Liebowitz D, Zisserman A (1998) Metric rectification for perspective images of planes. In: Proceedings IEEE computer society conference on computer vision and pattern recognition (Cat No.98CB36231), pp 482–488

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Javier Antonio Gonzalez-Trejo has a B.S. degree in Computer Science from the Benemerita Universidad Autonoma de Zacatecas and a M.Sc. in Software Engineering from the Center for Research in Mathematics (CIMAT), in Mexico. He is currently working as a Senior Software Engineer. His main areas of interest are the research of crowd detection and counting, lightweight CNN architectures for embedded applications and Deep Learning in general.

Diego A. Mercado-Ravell was born in Mexico City. He received his B.S. degree in Mechatronics Engineering from the Universidad Panamericana in Aguascalientes, Mexico, the M.Sc. degree in Electrical Engineering option Mechatronics from CINVESTAV-IPN, Mexico City, and the Ph.D. in Automation, Embedded Systems and Robotics from the University of Technology of Compiègne, France. Professor Mercado has held post-doctoral positions at the Mechanical and Aerospace Department at Rutgers, the State University of New Jersey, USA, and at the French-Mexican Laboratory on Computer Science and Control UMI-LAFMIA 3175 at CINVESTAV Mexico. He is currently full-time research professor at CIMAT-Zacatecas, Mexico, and member of the national research system (SNI), level I since 2018. His research topics include robotics, modeling and control, unmanned aerial/underwater vehicles, autonomous navigation, real-time embedded applications, state estimation, data fusion, computer vision and deep learning applications.

Uziel Jaramillo-Avila has a B.S. degree in Mechatronics and a M.Sc. in Electrical Engineering from the Universidad de Guanajuato. He earned a Ph.D. in Automatic Control and Systems Engineering from the University of Sheffield in 2021. The same year, he joined the Center for Research in Mathematics (CIMAT) in Zacatecas, Mexico. His main areas of research are computer vision in embedded systems, bioinspired robotics, and human-robot interaction.