DPPL Hallway Tracker: Hospital Contact Tracing During the COVID-19 Pandemic

Christian Marinoni[1], Valerio Ponzi[2,3] and Danilo Comminiello[1]

[1]Dpt. of Information Engineering, Electronics and Telecommunications, Sapienza University of Rome, Via Eudossiana 18, Roma, 00184, Italy
[2]Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto 25, Roma, 00185, Italy
[3]Institute for Systems Analysis and Computer Science, Italian National Research Council, Via dei Taurini 19, Roma, 00185, Italy

Abstract

During the COVID-19 pandemic, the use of a people tracking system could have been crucial, particularly in sensitive environments, such as hospitals. DPPL Hallway Tracker is a framework that uses security camera footage to determine which rooms in a corridor a person has entered. It generates a database containing all the people identified and allows quick identification of potential cases of infection based on the time spent in a room and its maximum capacity. DPPL Hallway Tracker is structured in two phases: detection and re-identification. In the first phase, it exploits Mask RCNN to identify people and room doors. In the second one, it uses the deep association metric model from DeepSORT to re-identify a person as he leaves a room.

Keywords

People Tracking, COVID-19 tracking systems

1. Introduction

Managing a pandemic has proved to be a difficult challenge despite the technological developments of the past decades. Containment measures based on restrictions on personal mobility (such as lockdowns) have proved to be very effective for infection containment [1, 2, 3]. However, these turn out to be short-term solutions that are not extendable throughout the whole virus's life cycle.

As with Covid-19, the presence of a potentially infected individual in a closed environment is a central problem and the risk of contagion increases with exposure time. Face masks, in combination with good room ventilation, help to reduce the risk of transmission. However, it is not sufficient to eliminate all the risks. Tracking operations are required to ensure the identification of the chain of contacts and the estimation of the relative risk of contagion. Tracking turns out to be even more essential in public settings, such as public offices and hospitals [4, 5].

Some countries, such as Italy and Germany, used specific tracking apps (respectively, Immuni and Corona-Warn-App) for a Bluetooth-based contact estimation [6]. These solutions, although potentially effective, have shown evident limitations, such as low diffusion in the population, constraints on the version of the smartphone OS, poor estimation of distances and related false positives. While they may be effective in the short term since they are employable on a big scale, other solutions prove to be more effective in the long run. Among these, the security cameras already installed in many public-private contexts can represent an excellent solution in terms of scalability and minimum requirements for the citizen. Indeed, they allow for the estimation of people's distances as well as the detection of room entrances and exits.

This project aims to create an offline framework for tracing the entrances and exits of people in one or multiple rooms facing a hallway. In this way, it is possible to extract some valuable information for estimating the risk of infection, such as the duration of the stay and the level of saturation of the room given its maximum capacity. The methodology described relies solely on Deep Learning solutions, and it employs two networks to detect doors and people and assign them appearance descriptors. A specific algorithm is in charge of tracking people's movements, exploiting the characterization of the hallway environment and the descriptors generated.

In particular, unlike other solutions that exploit motion features to determine a distribution of the positions where a subject can stay in the next frame [7, 8, 9], this project - named DPPL Hallway Tracker - uses only appearance features. A person is first identified in the scene and segmented using Mask R-CNN; then, their mask is passed to a Re-ID network to obtain an identifier (an array) that "describes" the way they appear in the scene. The descriptors are finally compared with those of the people already known to verify the person's identity. Another contribution, in addition to the general approach adopted, is the use of three new datasets to fine-tune the networks, built from scratch or starting from existing ones.

DPPL Hallway Tracker appears to be very effective

SYSYEM 2023: 9th Scholar's Yearly Symposium of Technology, Engineering and Mathematics, Rome, December 3-6, 2023
christian.marinoni@uniroma1.it (C. Marinoni);
ponzi@diag.uniroma1.it (V. Ponzi);
danilo.comminiello@uniroma1.it (D. Comminiello)

in tracking people entering and leaving rooms facing a corridor. The use of appearance features turns out to be sufficiently robust to allow correct identification, even if it is less effective in recognizing people who reappear in the corridor without leaving a room.

This report describes the project's workflow, from the description of the datasets to the results' analysis.

2. Related works

The object tracking problem is one of the classic problems in Computer Vision. Being able to determine the position of an object, even in the presence of partial or total occlusions, can be beneficial in many contexts, such as automated surveillance, video indexing, human-computer interaction, traffic monitoring, vehicle navigation and many others. A solution to the object tracking problem should manage multiple complexities: the loss of information caused by the projection of the 3D world on a 2D image, the complexity of the movement of objects, the presence of occlusions and changes in the scene illumination can make this task highly challenging.

The approaches can be divided into several categories based on their implementation and conceptual characteristics. In this Section, some solutions based on the "tracking-by-detection" strategy are mentioned. This strategy consists in doing a type-specific object detection or motion detection and then conducting (sequential or batch) tracking to link detection hypotheses into actual trajectories.

An example of an application is the one proposed by Bewley et al. [10], known as SORT (Simple Online and Realtime Tracking). It uses CNN-based detection - more specifically, Faster R-CNN [11]- to identify people in the scene. At that point, SORT associates a state $x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T$ with each target, where u and v represent the horizontal and vertical pixel location of the centre of the target, s and r are the scale (area) and the constant aspect ratio of the target's bounding box and, finally, $\dot{u}$, $\dot{v}$, $\dot{s}$ are the corresponding first derivatives (velocities) of u,v and s. The state gets updated at every new frame based on the related new detection and a Kalman Filter framework [12].

A related work is DeepSORT [7]. It expands the SORT framework by providing a re-identification network that takes as input the portion of the image showing the person and returns an appearance descriptor (a vector of size 128). This vector makes it easier to correctly assign identities to people by reducing the number of inter-frame ID switches.

SORT and DeepSORT, as well as other methods that use motion features, are effective tools for people tracking; however, they are not the best option in case of people entering and leaving rooms. Indeed, the states of multiple people entering the same room collapse at the same value, thus providing no valuable information for the ID attribution when a person leaves the room. On the contrary, the use of a re-identification network based on appearance features in DeepSORT is functional for the current application and is therefore also implemented in this project.

In today's literature, at best of our knowledge, there are no studies aimed at analyzing the specific context of tracking and re-identifying people who enter and leave rooms. Pedestrians on streets or people moving around indoors are usually the focus of most approaches. Other works specialize in counting people in some particular environments. For example, Rabaud and Belongie [13] investigate the possibility of counting people passing through crowded environments; [14], [15], [16] focus on counting passengers getting in/out of a bus and [17] of a metropolitan train; [18] counts people walking through a corridor or a door, without keeping into account their identities.

The absence of a similar application makes the comparison between the implementation proposed in this project with a baseline more complex. Therefore, in the following Sections, the individual modules that constitute it are compared with corresponding existing solutions, in the attempt to offer an objective yardstick on the choices made.

3. People and Door detection

The fundamental principle behind this project is the search for practical but effective solutions for tracking people entering and leaving rooms. As said in Section 2, in the "tracking-by-detection" strategy the first main challenge is object detection, i.e., producing a bounding box (and, eventually, a mask) for both people and doors in the image. The framework can thereby determine the position of a person at each frame and their relative distance from the doors detected in the scene. This Section describes the datasets used, as well as the implementation choices and the results obtained.

3.1. Object semantic segmentation

In order to obtain people tracking, it is crucial to identify the position of people and doors to understand which room they enter and leave. There are generally two ways to accomplish this task: object detection and image segmentation. Object detection focuses on defining the position of objects in an image, whereas image segmentation locates an object and defines a mask of pixels that represent it. This project exploits the second one - and, more in particular, its subclass known as *instance segmentation* - because of the benefits it provides in the re-identification

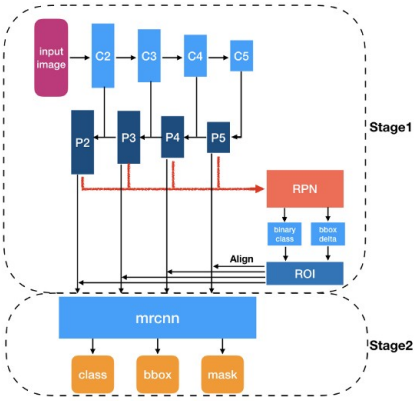

Figure 1: General scheme of the R-CNN Mask framework. The layers indicated with the letters C and P are convolutional layers that represent the backbone network. The classic pyramid architecture improves the detection of objects of various sizes.

task. More specifically, it employs the Mask RNN framework [19, 20], which derives from Faster RNN [11, 21] (in turn, one of the evolutions of the original R-CNN [22]) but adds a third parallel head used to generate the masks. It also introduces further improvements, like the support to pixel-to-pixel alignment between network inputs and outputs (ROI-Align). Figure 1 shows the different stages that characterize the network.

Initially, the image is passed as input to a convolution-based Feature Pyramid Network [23], which has the task of extracting meaningful information from differently-sized feature maps. An object can appear in the foreground (and therefore very large in the image) or further away from the camera; hence, this pyramidal structure facilitates its detection. The features thus extracted are passed to the Region Proposal Network (RPN), which produces several Regions Of Interest (ROI), each with its bounding box. At this point, the first-mentioned ROI-Align is applied and its result is passed to the second stage of the network, from which a series of fully connected layers allow to refine the position of the bounding box, the class of the object it contains and its mask.

Moreover, assuming the camera to be static and, therefore, the position of the doors to be fixed over time, this project exploits two distinct models: one for the door detection only and the other for people detection. Door detection is applied just in the starting phase of the framework while, from then on, people detection is performed. The process of generating the two models and the related results are analyzed below.

3.1.1. Door detection

To provide door detection, Mask R-CNN[19] was fine-tuned with a dedicated dataset, assembled for the purpose. It includes a selection of 2773 out of 3000 RGB images of the DeepDoors2 dataset [24], which is freely available online. These images represent one or multiple doors in different outdoors and indoors scenarios, which do not necessarily correspond to a corridor: in fact, the large majority of them represent doors from the front. They also include obstacles that partially occlude part of the doors. The annotations in the DeepDoors2 data set are provided as additional images where each one has a black background and different coloured masks for the doors. Being interested in this project more in the portion of space occupied by the door than in the profile of the door itself, all the images are re-masked to segment exclusively the door casing. Hence, almost all images have quadrilateral-shaped masks (thus with four vertices only). Moreover, the generated annotation files are no more encoded as images like in the original DeepDoors2 dataset, but they are fully compatible with the COCO dataset specifications [25]. In fact, the annotation files are JSON files containing: (1) references to all images, each having a unique ID, as shown in the first row of Table 1; (2) a mask and bounding box (bbox) associated to each image (second row of Table 1).

```
{"images": [
    {"id": 514, "width": 1080,
    "height":1920, "file_name":"frame.jpg"},
    ...
    ]
}
```

```
{"annotations": [
    {"id": 519, "iscrowd": 0,
    "image_id": 514, "category_id": 1,
    "segmentation": [[587.52,...,1097.77]],
    "bbox": [467.20,581.407,295.90,809.02],
    "area": 121068.87},
    ...
    ]
}
```

Table 1
An example of the formatting of JSON files containing image annotations according to COCO specifications is represented in this table. The first row shows the data structure used to list all the images in the dataset, the second row instead shows the one used to specify the annotations associated with each image, thus including the mask ("segmentation") and the bounding box ("bbox"). The "category_id" field is always set to 1, as there is only one category (door or person, depending on the dataset).

The dataset is split into training, validation and test

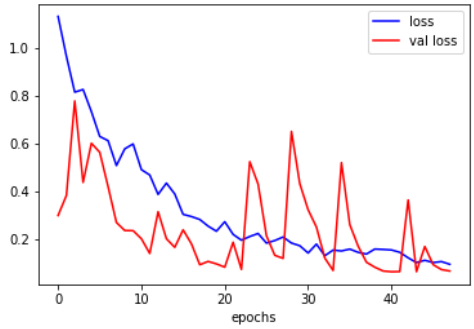

Figure 2: Training and validation losses during training with the Dppl dataset.

Figure 3: In this example, door detection is performed correctly with two of the three instances. Masks are shown in light red, while the center of the door is shown as a red dot.

sets. These subsets are disjoint; the training set contains 70% (1941) of the images, while the remaining 30% is equally divided between the validation and test sets (416 each).

With the new dataset available, called Dppl, we fine-tuned the model pre-trained with the COCO dataset, which is available on the framework's GitHub repository. Consequently, ResNet101 was used as the backbone, and training was done in the same manner as the framework's authors. In particular, we trained the head only for the first ten epochs; for the following thirty epochs, we fine-tuned stages four and above of the backbone too; finally, in the last ten epochs, we extended the training to the entire network. Unlike [19], the learning rate is initially set to 0.001 (rather than 0.02) to keep the weights from exploding; moreover, it is divided by a factor of 10 during phases two and three of the training. The other parameters are left unchanged, such as the weight decay of 0.0001 and momentum of 0.9. Finally, mini-masks were used (i.e. the masks were resized to the size of 56x56 px) to lessen the risk of memory problems. Data augmentation (horizontal flipping) was also applied. Figure 2 shows the train and validation losses got during training.

On the test set, the AP metric was used to assess the quality of the results produced by the training. AP, the acronym for Average Precision, computes the average precision value for recall values over 0 to 1. In practice, AP is computed as the mean of precision values at a set of R equally spaced recall levels, as defined by the following formula

$$AP = \frac{1}{R} \sum_{r \in \{0,\dots,1\}} p_{interp}(r)$$

where, given $p(\cdot)$ the precision, $p_{interp}(r) = \max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r})$ and $R = 101$ in COCO. AP@k stands for the average precision for IoU (Intersection over Union, i.e. how much the predicted mask overlaps with the ground truth) of k. More specifically, in the computation of AP@k, an estimated mask is considered to be True if its IoU is greater or equal than k, false otherwise.

The primary challenge metric for the COCO dataset is AP@[.50:.05:.95] (usually referred to simply as AP), which is the average AP for IoU (Intersection over Union) from 0.5 to 0.95 with a step size of 0.05. This metric is also used to evaluate the results of our test set. In particular, with the Dppl dataset and the training procedure described above, we got an AP of 85.7 and AP@.75 of 95.8. We also report the Average Accuracy, which is calculated by counting how many pixels out of those belonging to a specific area are correctly classified. In this case, rather than the whole image, the considered area is the smallest rectangular portion of the image that contains both the ground-truth mask and the one produced by the model. In numerical terms, we obtained an Average Accuracy of 95.34% in the case of Door Detection.

Figure 3 displays the situation in a corridor not included in the dataset: the door on the right that is particularly "thinned" from the perspective is indeed not detected. Precisely for this reason, the framework provides a specific graphical interface that allows adding new door positions, as shown in Section 4.3.

3.1.2. People detection

Similarly to what was done with the doors, a model for people detection is also generated. Mask R-CNN with the weights of COCO is already alone able to detect and segment people with acceptable accuracy. However, fine-tuning was done using a dedicated dataset built specifically for the occasion from videos captured along a hallway. More in detail, the dataset contains 793 frames

captured in a corridor by a 1080x1920 px resolution camera that was positioned a few centimeters from the ceiling (approximately 2.9 meters from the floor) with a vertical image layout. In the scene, six people appear walking down the hallway and entering/exiting the adjoining rooms. They wear various types of clothing (including a white coat to simulate the presence of a doctor); they are of different ages and all wear face masks. One of the people has a foot cast and crutches. All frames are hand-annotated to generate high-quality masks, accurately respecting the person's shape. The related annotation files follow the COCO specifications, as described before. The split of files between training (555 images), validation (119) and test (119) sets follows the same proportion as the Dppl dataset.

With this second dataset available, called dPPL, we once again fine-tuned the model pre-trained with the COCO dataset. All the Mask R-CNN's parameters are kept the same, but Gamma Contrast is used as a data augmentation technique in conjunction with horizontal flipping in this case.

Figure 4 shows the graph of the training and validation losses. As for the performance on the test set, Table 2 shows the comparative Average Precision values between the use of a model trained only with COCO and that obtained by doing fine-tuning with the dPPL dataset. This second option provides better results for both AP and AP@.75. The same applies for the Average Accuracy. These good results should be evaluated considering

Method	AP	AP@.75	Acc.
COCO only	70.5	92.9	99.08%
COCO+fine-tuning on dPPL	**76.3**	**95.5**	**99.74%**

Table 2
Comparison between the use of Mask R-CNN trained on COCO only and the same network trained with COCO and fine-tuned with dPPL dataset. *AP* stands for Average Precision; *Acc.* stands for Average Accuracy (calculated by counting how many pixels out of those belonging to smallest rectangular portion of the image that contains both the ground-truth mask and the one produced by the model are correctly classified).

the not very high number of images that compose the dataset. Indeed, environments with completely different illumination and compositions will certainly attenuate the good performances provided by this model.

3.2. People Re-identification

The detection of doors and people in the scene does not suffice to ensure accurate tracking. As mentioned above, one can use additional information extracted from the images within more or less complex systems, which may exploit appearance, movement and shape features. An

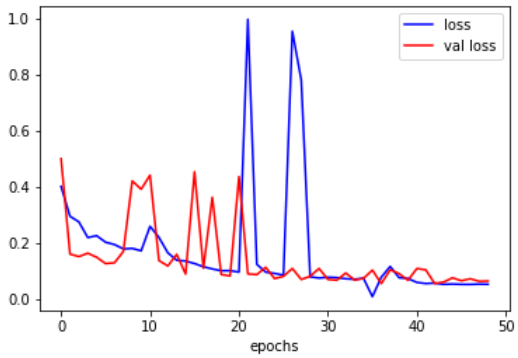

Figure 4: Training and validation losses during training with the dPPL dataset.

example is DeepSORT [7], which uses the Kalman filter to predict the position of a person in the next frame and integrates appearance information based on a deep appearance descriptor. Despite DeepSORT being a powerful tool, the use of the Kalman Filter turns out to be less effective when the subject disappears for long periods from the camera view. Indeed, the Kalman Filter modulates the state estimate of the system (in this case, the position in the frame of a subject) as a Gaussian distribution whose variance strictly depends on the observations over time. When a person disappears from the scene, the degree of uncertainty increases and the same happens to the distribution variance. Furthermore, the Kalman Filter would be practically useless if several people enter the same room: the states of those subjects would collapse into the same value, making this information useless to distinguish a person from the others when they leave the room. Nevertheless, the solution undertaken in DeepSORT on the use of appearance features turns out to be quite effective whenever the Kalman Filter is not since it relies on visual cues. For this reason, DPPL Tracker is primarily based on appearance features, though it also takes advantage of some assumptions related to the work environment (a corridor).

In this project, Deep Cosine Metric Learning [26], the same used in DeepSORT for appearance re-identification, is used. It applies a variation of Softmax classifier called Cosine Softmax Classifier, which allows obtaining a different representation space in which compact clusters are formed based on the appearance features. This is achieved by first applying the $l2$ normalization, which uses the $l2$-norm to normalize the input values so that, if squared and summed, they would result in the value 1, and, secondly, by normalizing the weights. Finally, the cosine softmax classifier is applied, which is defined as

follows:

$$p(y_i = k|r_i) = \frac{\exp(\kappa \cdot \tilde{w}_k^T r_i)}{\sum_{n=1}^{C} \exp(\kappa \cdot \tilde{w}_n^T)}$$

where κ is a free scaling parameter.

Table 3 summarizes the entire network, which is made up of convolutional and residual layers. Dropout of 0.4 is used within the Residual layers.

Layer	Patch Size/Stride	Output
Conv1	3 ×3/1	32 × 128 × 64
Conv2	3 ×3/1	32 × 128 × 64
Maxpool	3 ×3/2	32 × 64 × 32
Residual 4	3 ×3/1	32 × 64 × 32
Residual 5	3 ×3/1	32 × 64 × 32
Residual 6	3 ×3/2	64 × 32 × 16
Residual 7	3 ×3/1	64 × 32 × 16
Residual 8	3 ×3/2	128 × 16 × 8
Residual 9	3 ×3/1	128 × 16 × 8
Dense 10	-	128
l_2 normalization	-	128

Table 3
Overview of the CNN architecture of the Re-ID network

The dataset used for training the re-ID network is MARS [27], a large scale video-based person re-identification dataset that extends the Market-1501 dataset [28]. It consists of 1261 different pedestrians, who are captured by at least two of the six near-synchronized cameras placed on the Tsinghua University campus. It also includes over 1 million bounding boxes and 3248 distractors to make it more realistic. The goal of the Re-Identification network is to provide useful information on the person's identity starting from how they appear in the image. In the case of MARS, it will have to try to learn this information from images that also include backgrounds of different colours and patterns. To concentrate solely on the subject, we preprocessed the MARS dataset by using the Mask R-CNN network to detect people. Therefore, the result is a new dataset where each image of size 256x128 px represents a segmented person and a black background (as shown in Figure 5).

The network has been trained for 100.000 steps, with a constant learning rate of 0.001 and weight decay of 1×10^{-8}; moreover, the input images are scaled to 128x64 px.

The use of the masked MARS dataset proves to be beneficial for the network training since it provides improved results according to the CMC Rank@K and mAP metrics[1], as shown in Table 4. The table also shows the results of two state-of-the-art solutions on the original MARS dataset. Both largely outperform the solution proposed in this project, however, they also use much more sophisticated methods or networks with many more parameters.

[1]Computed through the MARS evaluation tool, available at https://github.com/liangzheng06/MARS-evaluation

(a) (b)

Figure 5: Examples of the resulting images in the MARS dataset after applying object instance segmentation.

Method	Rank1	Rank5	mAP
DCML on MARS[a]	72.93	86.46	56.88
DCML on masked MARS[b]	75.73	90.08	60.72
B-BOT + Attention & CL loss[c]	88.6	96.2	82.9
MGH[d]	90.0	96.7	85.8

Table 4
Comparison between the Deep Cosine Metric Learning (abbreviated to DCML) on the original MARS dataset and the masked version and some state-of-the-art solutions. [a]Results from [26] - [b]Proposed in this project - [c]Results from [29] - [d]Results from [30]. *mAP* stands for *mean Average Precision*

4. DPPL Tracker framework

People tracking is offered through a specific framework that employs Mask R-CNN and the above-mentioned re-identification network. It also provides additional features to improve the user experience and optimize the search for people. More precisely, the workflow is the

following: the first frame is first passed as input to Mask R-CNN for doors detection. Once doors are located, that frame and the following ones are passed to the same network (with different weights) for people detection. The portion of the image containing each person is then multiplied by the corresponding mask (to have a black background) and, after being resized to 128 x 64 px, is passed to the re-identification network. The latter has its head cut off so that it outputs an array of size 128 (generated by the last Dense layer). This array is a descriptor of the person's appearance and is used by the framework's main algorithm to associate a unique identity ID with each person.

4.1. Main algorithm

After selecting the video, the first frame is analyzed through mask-RCNN to locate the doors in the scene. If one or more doors are not detected, the user can manually add additional ones, as shown in Section 4.3. Only at that point, the analysis of the following frames begins. Pseudocode 1 shows the main steps. As previously described, Mask R-CNN is again used to identify people, while the re-ID network provides the people appearance descriptors. At that point, for each person, the *find_nearest* function allows identifying the already-known closest identifier to the detected descriptor, if any. In this way, it is possible to determine whether that person already appeared in the past and, depending on their position and on the knowledge derived from past frames, a log is added to the database if they are leaving a room. If there is no similar person, the algorithm adds a new one to the scene. The final *for* loop finds all people who were in the environment up to the previous frame but are now missing. In this case, there are two alternatives: the person may either have entered a room (if in the preceding frame they were sufficiently close to the relative door) or may have disappeared, for example, because they left the hallway or are temporarily occluded. To improve the efficacy of the algorithm, the framework starts tracking a person when he appears entirely in the scene and his bounding box is at a minimum distance from the image edges. Furthermore, it uses the area of the bbox to interrupt (temporarily or not) the tracking when an object/person occludes the subject or when the tracked person has nearly entirely entered a room.

A fundamental step is the one implemented by the *find_nearest* function, shown in Pseudocode 2. It uses differentiated searches to find the already-known person with the most similar identity to the one passed as input. First, it searches among the people visible in the scene in the previous frame. In case of failure, if the detection is close enough to a door - according to a given threshold - it searches among the people who are known to be in that room. As a last chance, it starts searching among the people who last left the corridor, then moving on to all the known people. The similarity between two identifiers IDa and IDb is computed with the cosine similarity, as follows

$$\textit{cosine similarity} = \frac{\mathrm{ID}a \cdot \mathrm{ID}b}{\|\mathrm{ID}a\|\|\mathrm{ID}b\|}$$

Two identifiers are more similar as the cosine similarity goes to one. Hence the need to define, for each of the listed searches, a threshold that defines when two descriptors must be considered sufficiently similar (and therefore belonging to the same person) or not. The choice of the threshold heavily influences the tracking effectiveness. In the various phases a different threshold is used, more specifically: (1) if a person is walking along the corridor without other people in the close vicinity and, if compared to the previous frame, that person has not moved too far from their previous position in the scene, then a greater dissimilarity between the descriptors is tolerated; (2) in other cases, the threshold is set to a value between 0.85 and 0.9. Section 5 discusses some critical issues regarding the choice of the threshold.

Algorithm 1: Main algorithm

```
Data: maskRCNN_result, frame
Result: People identified
currently_detected ← [];
for person in maskRCNN_result do
    mask, bbox ← person;
    imgportion ← frame[bbox[0] : bbox[2], bbox[1] : bbox[3]];
    imgportion_masked ← imgportion * mask;
    identifier ← get_person_identifier(imgportion_masked);
    personID, roomID ← find_nearest(person, identifier);
    if pID == -1 then
        // New person appeared
    else
        // Person in the corridor or exited from a room
    end
    currently_detected ← person
end
for person in get_people_in_scene() do
    if person not in currently_detected then
        if person close to a room then
            // Person entered in a room
        else
            // Person disappeared from the scene (may due to an occlusion)
        end
    end
end
```

Algorithm 2: Find nearest identity

```
Data: identifier
Result: Person id
currently_detected ← [];
if identifier in the scene then
    // Person in the scene, return the ID
end
for door in room do
    if person close to door then
        // Look among people inside that room
    end
end
// Look among last detected people;
// Look among all people;
```

4.2. Database

Whenever a person enters and leaves a room, a corresponding log is added to the database. Each log has the following structure:

```
frameID personID roomID "in/out/new"
```

where `frameID` is an incremental value representing the currently processed frame, `personID` is a unique integer associated to a person (different from the identifier representing the way that person looks in the scene), `roomID` is the ID of the room the person is entering/leaving - if any - and it is equal to -1 otherwise. The last label has the value "`in`" or "`out`" when "`roomID`" is different from -1, while it assumes the value "`new`" when a new person appears in the scene.

For simplicity, the database is implemented via a simple CSV file containing all the logs, but more complex and scalable solutions (such as NoSQL) are also possible. Knowing the video framerate, the framework derives an estimate of the time spent in the room, to highlight possible dangerous situations. The same is done by counting the number of people in the same room and alerting when the maximum capacity is exceeded.

4.3. GUI

A simple user interface, implemented with the PySimpleGUI library, is also available to provide the user with more flexible interaction with the framework. The user can select a file or directory containing the needed frame images, as well as add new doors that Mask R-CNN did not detect. In this second case (shown in Figure 6), by using a simple library such as Matplotlib, it is possible to offer a response in real-time on the location of the new doors and their heights (used by the algorithm). Finally, at the end of the processing of all frames, the user can search all the times a particular ID has entered and left a room (Figure 7). In the latter case, the interface highlights the riskiest situations (for example, if the room capacity has been exceeded) in addition to providing all records linked to the entered ID.

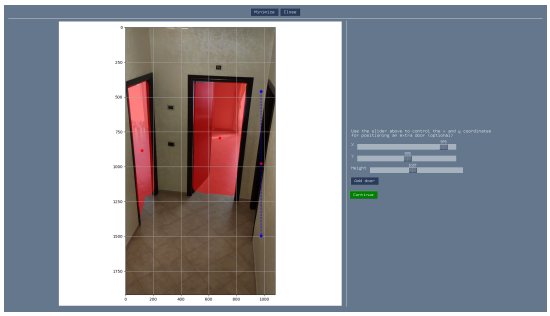

Figure 6: The user can add multiple additional doors through the user interface. The position of the center of the new door is shown by a red dot, while its height by a dashed line with two blue dots at the ends.

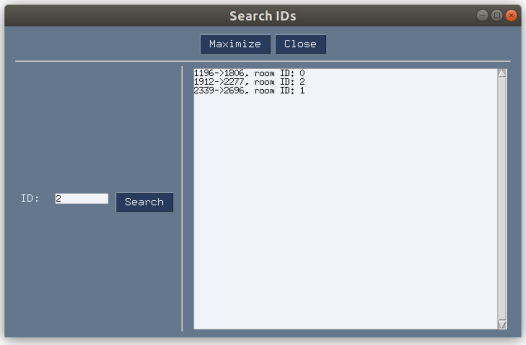

Figure 7: The user can visually see a list of rooms a particular ID has entered through the user interface.

5. Analysis and results

The behaviour of the framework is evaluated in two different setups of incremental difficulty. In the first setup, people walk down a corridor one after the other, in a perfect flow that limits the occasions when two or more people are simultaneously in the same room. This modality allows focusing mainly on an inter-frame re-identification and on the correct detection of people entering and leaving the rooms. In the second setup, multiple people can enter the same room. The challenge, in this case, is to be able to identify the identity of a person when he leaves the room. The results show that the algorithm can handle a wide range of situations with ease, producing results that are similar - if not identical - to the ground truth.

First of all, it is beneficial to analyze how accurately the framework can detect the presence of one or more people in the scene. To calculate the overall accuracy of the detections we used two methods. The first consists of considering only those frames in which a person is shown entirely (i.e., he is not hidden - even partially - by objects or other people). The second way is to consider all frames, including all borderline cases in which only a portion of a person's arm or leg appears in the frame. Figure 8 shows an example of the frames considered with both methods. The results - obviously better in numerical terms in the first case - are shown in Table 5.

	Overall (Detection) Accuracy
Method 1	100%
Method 2	91.76%

Table 5
The accuracy of people detection computed with two methods is shown here. With the first one, we only considered those frames in which people bodies are shown wholly in the image. The second method also includes those frames in which a person is only partially visible.

Figure 8: The frame on the left is an example of those considered with Method 1 for calculating the Overall Detection Accuracy. The person's body is entirely included in the scene. The frame on the right is instead an example of those considered with Method 2, that takes also into account all borderline cases in which only a portion of a person's arm or leg appears in the frame. In this case, the two people in the scene are only partially visible and the arm of the uppermost person is not detected by the model.

Having ascertained that the framework can detect the presence of people with good reliability, we then move on to analyze the accuracy of people tracking. In particular, the inter-frame re-identification of a person in the scene scores 100% accuracy, even in the case of several people in the corridor; the same happens when the person leaves a room, even when more than one is inside it. The criticalities are mainly two: (1) the difficulty in defining an efficient threshold for cosine similarity, since the method adopted is susceptible to sudden changes in the person's position (such as front and rear vision of the person); (2) the influence of the quality of masks produced by Mask R-CNN on the re-identification network. A sudden change in the portion of the image taken into consideration (even without sudden movements of the subject) can reduce the cosine similarity.

Cosine similarity can be a powerful tool for guiding the re-identification task: limiting the search to the people inside the room and using the cosine similarity always leads to correct identifications. Nevertheless, the weaknesses listed above heavily reduce its effectiveness when it is necessary to recognize a person who had previously left the corridor (without entering any room) and who reappears later on. Indeed, the choice of a high threshold (i.e., ≥ 0.9) makes it difficult to assign the same ID in the situation under analysis, because usually, the person will reappear in a completely different pose (for example, from behind and not the front) which will reduce the value of the cosine similarity. In this case, there will be no ID switches between different people, but each time one reappears in the scene it will be assigned a new ID.

On the contrary, lowering the threshold facilitates the ID switches, creating some cascading problems in the framework (an ID already assigned - even if incorrectly - to a person will not be re-assigned as long as the person is in the scene, not even if the one it was originally assigned to reappears). However, these problems do not affect the recognition of people leaving the rooms: the identifier produced by the Re-ID network and the similarity computed with the cosine similarity is sufficient for the correct attribution of the ID. Compared to the baseline (Re-ID network trained on the original MARS dataset), it can be observed that the cosine similarity of the same person in two different situations (frames) is greater (by 1-2%) when assessed with our method.

As a final benchmark, the accuracy of the logs (seen as the ratio of the logs equal to ones of the ground truth over the total number of them) produced in the tests is equal to 50%. The accuracy goes up to 84% if we also include those logs with labels "in" and "out" that differ only in the person ID from the ground truth (but only if that ID is a new one, and therefore if there is no ID switch with a previously known identity). When a person enters a room, the relative log at the exit is always correct, as already mentioned above. As for performance, an Nvidia Tesla K80 is capable of processing 1.4-1.5 frames per second.

We also ran a test in a setup with slightly different specifications. In fact, the recording device was placed at eye level, tilted almost parallel to the floor and with an image ratio of 16:9. The results obtained are comparable to those indicated above, although tracking people in areas very distant from the camera (and therefore at lower resolution) turns out to be more critical. Under these conditions, it is quite easy for two different subjects to appear very similar even to the human eye. An example is shown in Figure 9. Ultimately, the framework is most effective when the distance to the doors is not excessively large.

Figure 9: Those shown in the figure are two different people, who however visually appear practically identical. Their appearance descriptor is therefore very similar and this leads the framework to a wrong ID attribution when one of the two leaves the room.

6. Conclusion

DPPL Hallway Tracker turns out to be a good starting point for developing a framework capable of tracking people entering and leaving multiple rooms. The use of a re-ID network that exploits the masks produced in the detection and segmentation phase leads, even in the tests performed, to improvements in identification.

A project extension might be able to address some of the remaining issues: (1) the enrichment of the datasets of people and doors could lead to better detection in several more challenging contexts: for example, as discussed above, the detection and segmentation of doors "thinned" from perspective remains difficult; (2) using a dynamic threshold and investigating complementary solutions to the re-identification network could alleviate the difficulty of assigning the same ID to a person who reappears in the corridor without leaving a room. The study of solutions for tracing people entering and leaving the rooms is of great importance for the application developments that it can have. It not only allows contact tracing in the event of pandemics but it can be also used for other contexts, as for the analysis of the movements of patients and medical operators and the optimization of hospital wards.

References

[1] V. Alfano, S. Ercolano, The efficacy of lockdown against covid-19: a cross-country panel analysis, Applied health economics and health policy 18 (2020) 509–517.

[2] S. Pepe, S. Tedeschi, N. Brandizzi, S. Russo, L. Iocchi, C. Napoli, Human attention assessment using a machine learning approach with gan-based data augmentation technique trained using a custom dataset, OBM Neurobiology 6 (2022). doi:10.21926/obm.neurobiol.2204139.

[3] V. Ponzi, S. Russo, A. Wajda, R. Brociek, C. Napoli, Analysis pre and post covid-19 pandemic rorschach test data of using em algorithms and gmm models, volume 3360, 2022, pp. 55 – 63.

[4] V. Marcotrigiano, G. D. Stingi, S. Fregnan, P. Magarelli, P. Pasquale, S. Russo, G. B. Orsi, M. T. Montagna, C. Napoli, C. Napoli, An integrated control plan in primary schools: Results of a field investigation on nutritional and hygienic features in the apulia region (southern italy), Nutrients 13 (2021). doi:10.3390/nu13093006.

[5] G. De Magistris, M. Romano, J. Starczewski, C. Napoli, A novel dwt-based encoder for human pose estimation, volume 3360, 2022, pp. 33 – 40.

[6] M. Bano, C. Arora, D. Zowghi, A. Ferrari, The rise and fall of covid-19 contact-tracing apps: when nfrs collide with pandemic, in: 2021 IEEE 29th International Requirements Engineering Conference (RE), 2021, pp. 106–116. doi:10.1109/RE51729.2021.00017.

[7] N. Wojke, A. Bewley, D. Paulus, Simple online and realtime tracking with a deep association metric, in: 2017 IEEE international conference on image processing (ICIP), IEEE, 2017, pp. 3645–3649.

[8] A. Alfarano, G. De Magistris, L. Mongelli, S. Russo, J. Starczewski, C. Napoli, A novel convmixer transformer based architecture for violent behavior detection 14126 LNAI (2023) 3 – 16. doi:10.1007/978-3-031-42508-0_1.

[9] B. Yang, R. Nevatia, Multi-target tracking by online learning of non-linear motion patterns and robust appearance models, in: 2012 IEEE Conference on

Computer Vision and Pattern Recognition, 2012, pp. 1918–1925. doi:10.1109/CVPR.2012.6247892.

[10] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, 2016 IEEE International Conference on Image Processing (ICIP) (2016). URL: http://dx.doi.org/10.1109/ICIP.2016.7533003. doi:10.1109/icip.2016.7533003.

[11] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28 (2015) 91–99.

[12] R. E. Kalman, A new approach to linear filtering and prediction problems (1960).

[13] V. Rabaud, S. Belongie, Counting crowded moving objects, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 1, 2006, pp. 705–711. doi:10.1109/CVPR.2006.92.

[14] C. Labit-Bonis, J. Thomas, F. Lerasle, F. Madrigal, Fast tracking-by-detection of bus passengers with siamese cnns, in: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2019, pp. 1–8. doi:10.1109/AVSS.2019.8909843.

[15] C.-H. Chen, Y.-C. Chang, T.-Y. Chen, D.-J. Wang, People counting system for getting in/out of a bus based on video processing, in: 2008 Eighth International Conference on Intelligent Systems Design and Applications, volume 3, 2008, pp. 565–569. doi:10.1109/ISDA.2008.335.

[16] J.-W. Perng, T.-Y. Wang, Y.-W. Hsu, B.-F. Wu, The design and implementation of a vision-based people counting system in buses, in: 2016 International Conference on System Science and Engineering (ICSSE), 2016, pp. 1–3. doi:10.1109/ICSSE.2016.7551620.

[17] S. A. Velastin, R. Fernández, J. E. Espinosa, A. Bay, Detecting, tracking and counting people getting on/off a metropolitan train using a standard video camera, Sensors 20 (2020). URL: https://www.mdpi.com/1424-8220/20/21/6251. doi:10.3390/s20216251.

[18] S. D. Pore, B. F. Momin, Bidirectional people counting system in video surveillance, in: 2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT), 2016, pp. 724–727. doi:10.1109/RTEICT.2016.7807919.

[19] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[20] F. Bonanno, G. Capizzi, S. Coco, C. Napoli, A. Laudani, G. L. Sciuto, Optimal thicknesses determination in a multilayer structure to improve the spp efficiency for photovoltaic devices by an hybrid fem - cascade neural network based approach, 2014, pp. 355 – 362. doi:10.1109/SPEEDAM.2014.6872103.

[21] F. Bonanno, G. Capizzi, G. L. Sciuto, C. Napoli, Wavelet recurrent neural network with semi-parametric input data preprocessing for micro-wind power forecasting in integrated generation systems, 2015, pp. 602 – 609. doi:10.1109/ICCEP.2015.7177554.

[22] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[23] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, 2017. arXiv:1612.03144.

[24] J. Ramôa, V. Lopes, L. Alexandre, S. Mogo, Real-time 2d–3d door detection and state classification on a low-power device, SN Applied Sciences 3 (2021). doi:10.1007/s42452-021-04588-3.

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.

[26] N. Wojke, A. Bewley, Deep cosine metric learning for person re-identification, in: IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018. URL: https://elib.dlr.de/116408/.

[27] MARS: A Video Benchmark for Large-Scale Person Re-identification, Springer, 2016.

[28] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, Q. Tian, Scalable person re-identification: A benchmark, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015.

[29] P. Pathak, A. E. Eshratifar, M. Gormish, Video person re-id: Fantastic techniques and where to find them, 2019. arXiv:1912.05295.

[30] Y. Yan, J. Qin1, J. Chen, L. Liu, F. Zhu, Y. Tai, L. Shao, Learning multi-granular hypergraphs for video-based person re-identification, 2021. arXiv:2104.14913.