Social distance monitoring using YoloV4 on aerial drone images

Ali El Habchi[1], Kaoutar Baibai[1], Younes Moumen[2], Ilham Zerouk[1], Wassim Khiati[1], Nourdine Rahmoune[1], Jamal Berrich[1] and Toumi Bouchentouf[1]*

[1] Mohammed First University, ENSAO, AIRES laboratory, Oujda, Morocco
[2] ATLAN space, Casablanca, Morocco

Abstract. Monitoring social distancing in public spaces plays a crucial role in controlling and slowing the spread of the coronavirus during the COVID-19 pandemic. Using camera-equipped drone, the system presented in this paper detect unsafe social distance between people by applying deep learning algorithms namely the YoloV4 CNN algorithm to detect persons in images, in combination with trans-formation equations to calculate the real world position of each person, and finally calculate the distance between each pair in order to determine whether it is safe. We show also the results of training and testing a model using YoloV4 algorithm, and test the system for social distance calculation.

1 Introduction

The 2019 coronavirus pandemic (COVID-19) which was reported by China in Wuhan, Hubei province on December 31, 2019 [1], caused a global crisis by its lethal spread around the world. Medical experts, scientists and researchers have worked intensively to search for the vaccine and the effective drug for this deadly virus. To limit the spread of the infectious virus, the world community is looking for precautions and alternative measures.

Currently, since vaccination rates are still low on poor areas and due to newer strains that are highly contagious, social distancing is still considered an adequate precaution against the spread of this virus. Research has shown that social distancing measures are more effective. This reduced the average number of infections by 92% and 24% in mid-2020 and at the end of 2020, respectively [2]. This distancing involves maintaining a distance of 1.5 m between people, which can prevent the spread of most infectious respiratory diseases.

Current research offers effective solutions to measure this distance using surveil-lance videos as well as approaches based on computer vision, machine learning and deep learning. In this article, a solution based on deep learning is proposed. This solution is based on a camera-equipped drone, which essentially consists of two parts the first is the detection of individuals gathered in a public place using convolution net-works followed by the calculation of the distance between these individuals.

2 Related works

Recent studies highlighted the effectiveness of social distancing practices during the COVID-19 pandemic. Most of this research focuses on techniques based on computer vision, because it offers a cheaper and more attractive alternative to smartphones for safety assessment or pedestrian detection [6].

For object detection, two main approaches are used. The first is a region-based approach, such as Fast-RCNN, Faster-RCNN [10], Mask RCNN, and RetinaNet, which detects humans from images in two steps. The work [9] uses the RCNN [14], Fast RCNN [13], and Faster RCNN algorithm to detect masks and social distancing in real time. Despite the high detection precision of these approaches, their applications could be limited due to their great complexity. The second approach is called the single stage approach which includes the You Only Look Once (YOLO) [3] model, improved versions of this YOLO3 [4] YOLO4 [5] models, as well as the Multibox Single Shot detector (SSD). These approaches combine the pixels of the image with bounding boxes and class probabilities in order to detect people or objects; they are generally faster than region-based methods.

By combining these computer vision techniques with distance approximation methods, it can be determined whether a group of people meets social distancing requirements, which helps reveal patterns of human contact and the effectiveness of social distance policies.

Among the most recent work that has been conducted to investigate the problem of visual social distancing for COVID-19, the work [11] that uses the SSD300 model to detect people in a video or image, implementing surveillance of social distance in real time. The work [7] proposes a new technique for detecting and tracking people by deep learning, which exploits the YOLOv2 approach, using thermal images. The distance calculated using the Euclidean distance between two bounding boxes. The work [8] developed an autonomous drone-based model for social distance monitoring. They trained

* Corresponding author: a.elhabchi@ump.ac.ma

the YOLOv3 model with a custom dataset. Calculations made by measuring the centroid distance between pedestrians. The YOLOv4 model was used for the detection and tracking of people using a single time-of-flight (ToF) camera [12].

3 Methodology

The main problem that we are trying to solve in this project is the detection of people who do not follow the measures of social distancing using a camera-equipped drone. The problem can be divided into two axes, the first is the detection of people by a detection algorithm using deep neural networks, which gives as a result the locations of the objects/people detected in the image, this result is then used in the second stage to estimate the position of persons in ground and then calculate the social distance using the output of the first stage combined with camera position and orientation, to detect if people are respecting the social distancing rules as is shown in the figure 2.

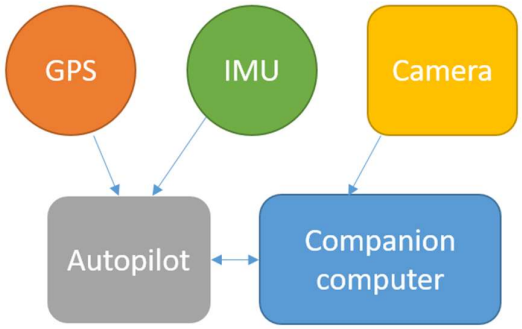

Fig. 1. System components and sensors

To solve this we are using a drone equipped with an RGB camera, a GPS sensor for the location capture, the IMU sensor for the orientation of the camera, this is then forwarded to the companion computer embedded in the drone (see figure 1) through the autopilot.

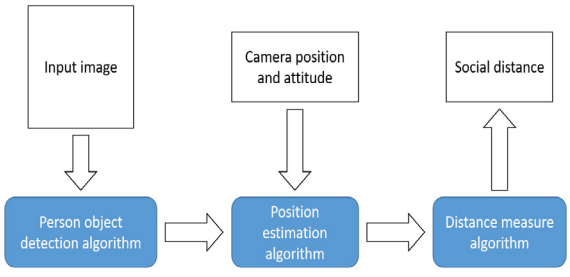

Fig. 2. Flow chart diagram of the whole system

3.1 Person detection

1.1.1 Darknet

For the Object detection task, we used multiple architectures all based on the darknet deep learning framework [15], and the YOLO algorithm [3]. The first architecture that we used is the YoloV3 [4], which is an improved version of the original YOLO algorithm and the second version YoloV2 [16]. In this architecture, they improved the training performance and accuracy, and they added scale output to deal with different objects sizes, we used also 2 different resolutions, 416 pixel by 416 and 608 by 608. After that we tested training also on YoloV4 [5] which is the newest version that is based also on darknet framework, in this version they added a lot of new features that can be used to decrease the training time, like new augmentation techniques (Mixup, Cutout, cutmix) and many others techniques.

1.1.1 Dataset

The dataset that we used in the beginning was a general purpose 'Visdrone-DET' dataset used originally as challenge for object detection for vehicles, persons, and other classes of aerial images captured from a drone in cities [18]. We selected from the dataset images that contains a majority of persons in them, we arrived at 1735 images, divided to either 608 by 608 or 416 by 416 depending on the architecture used, and contains around 38762 person/bounding box in them. After seeing the results, we augmented the dataset with another dataset for person detection 'Okutama-action' [17] to improve further the performance and generalization of our model; this dataset consists of 43 minutes of image sequences of aerial footage of persons with annotations, to add to 3491 images.

3.2 Social distance measurement

To measure the distance between people in the image, first we need the camera intrinsic parameters, like the sensor size of the camera, the focal length, field of view, etc. we need also extrinsic camera parameters that we capture using external sensors, GPS for position of the camera/drone, and IMU for attitude of camera. We also need the position of objects in the image using the detection algorithm results. Then we use triangulation equations to calculate the real world position of persons based on all those information, so then we can calculate the distance between each pair of persons. The figure 3 shows a scenario where the drone is hovering on persons and shows the frames of reference of position and orientation.

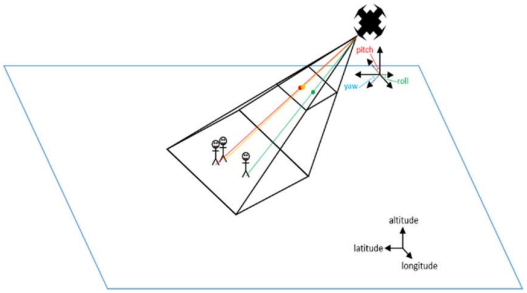

Fig. 3. Shooting scenario, and frames of reference

3.3 Algorithm

The Social distance-monitoring algorithm (see Figure 4) consists of an outer while loop where we capture every camera frame alongside the camera position from the GPS, the camera orientation from the IMU, and pass it to the object detection algorithm, which gives us as a result the bounding boxes of the objects/persons in the image. Then we loop through those bounding boxes after we get their center bottom coordinates, and we calculate the geolocation for each pair of the objects, so finally we can measure the Euclidian distance between the pairs, and decide whether the distance is safe of not.

```
Algorithm 1: Social Distance Monitoring
Result: Distances
while notFinished do
    cameraOrientation ← getIMUReadings()
    cameraPosition ← getGPSReadings()
    image ← Camera.capture()
    Bboxes ← ImageDetector.detect(image, config, weights)
    for Object1Pos in ObjectsPos do
        for Object2Pos in ObjectsPos do
            if Object1Pos!=Object2Pos then
                position1 ← getTargetLatLong(Object1Pos,
                cameraPosition,cameraOrientation, cameraParams)
                position2 ← getTargetLatLong(Object2Pos,
                cameraPosition, cameraOrientation, cameraParams)
                distance ← getDistance(position1, position2)
                if distance > safeSocialDistance then
                    Distances[Object1][Object2] ← distance
                end
            end
        end
    end
end
```

Fig. 4. Social distance monitoring algorithm

4 Results and discussion

The training and testing stage of the person detection algorithm, was done on a Google Colab virtual machine which gives a handful of GPUs, ranging from a Tesla T4, Tesla P4, Tesla K80, and even P100 [19]. The results of tests below are all done on a Tesla P4 GPU as seen in (table 1).

Table 1. Different architectures performance results

Architecture	mAP@0.50	precision	recall	F1-score	Average IoU	Inference time
YoloV3 608 coco	38.78	0.49	0.32	0.39	35.56	54 ms
YoloV3 416	44.57	0.59	0.52	0.55	40.68	**28 ms**
YoloV3 608	63.04	**0.75**	0.66	0.70	**54.27**	47 ms
YoloV4 608	**74**	0.70	**0.72**	**0.71**	51.22	54 ms

We began testing with the YoloV3 with the original weights of COCO dataset [20], where we got a mAP of 38.78, after that we trained the YoloV3 on 2 resolutions, the first is the default resolution with 608 by 608; that reached 63.04 AP after 6773 iteration (see figure 5), and the second resolution of 416 by 416 reached an AP of 44.57, we trained also the YoloV4 with 608 resolution and we reached 74% of AP after 16474 iteration (see figure 6).

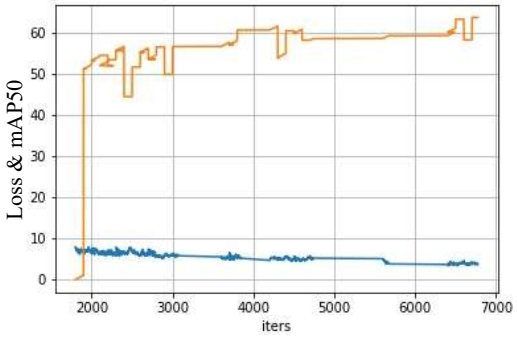

Fig 5. YoloV3 608 training loss and mAP50 charts

For the performance side you can see the inferences times on (Table 1). YoloV3 with the input resolution of 416 pixel by 416 had the fastest time on the Tesla T4 with an average time of 28 ms, followed by the same architecture with an input resolution of 608 by 608 of 47ms. The original architecture with coco dataset weights had an inference time of 54 ms considering the bigger output shape of 80 classes instead of the single person class output. Finally, the YoloV4 had a similar inference time of 54 ms, which is good considering the bigger architecture and all the added features and training time advantage and precision over the old one.

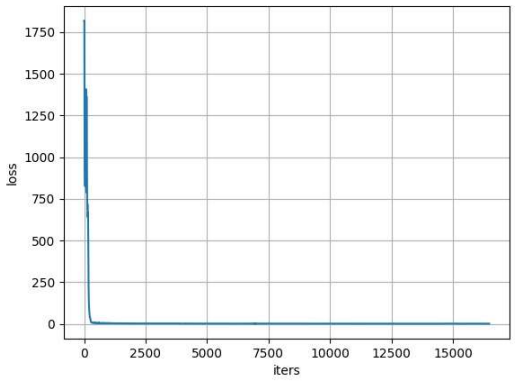

Fig. 6. YoloV4 training loss chart

To test the social distance method, we have done a static test with an android phone camera due to lack of proper equipment, where we captured images from a high vantage point, and the position and orientation of the phone. As first results, we got an average error of **40** cm, which is not very bad considering the accuracy of phone measurements.

5 Conclusion & Future works

In this paper, we presented our work on creating a system for person detection for monitoring crowd social distance. In the first part, we used deep learning approach for the object detection where we trained a model that reached good performances for the task. For the social distance and location calculation, we have developed a method based on triangulation equations to transform from the image pixel space to 3D world space and calculating the distance between each pair of persons, and then we did a proof of concept test using phone camera and measurements. The results are good considering the accuracy of measurements. Regarding future works, we are waiting for testing equipments, and we are building a proper dataset for social distance calculation benchmark.

References

1. Gralinski, L. E., & Menachery, V. D. Return of the coronavirus: 2019-nCoV. Viruses. 2020; 12: 135. Google Scholar.
2. Prem, K., Liu, Y., Russell, T. W., Kucharski, A. J., Eggo, R. M., Davies, N., ... & Klepac, P. (2020). The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China: a modelling study. The Lancet Public Health, 5(5), e261-e270.
3. REDMON, Joseph, DIVVALA, Santosh, GIRSHICK, Ross, et al. You only look once: Unified, real-time object detection. In : Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 779-788.
4. REDMON, Joseph et FARHADI, Ali. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.
5. BOCHKOVSKIY, Alexey, WANG, Chien-Yao, et LIAO, Hong-Yuan Mark. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020.
6. ZUO, Fan, GAO, Jingqin, KURKCU, Abdullah, et al. Reference-free video-to-real distance approximation-based urban social distancing analytics amid COVID-19 pandemic. Journal of Transport & Health, 2021, vol. 21, p. 101032.
7. Saponara, S., Elhanashi, A. & Gagliardi, A. Implementing a real-time, AI-based, people detection and social distancing measuring system for Covid-19. J Real-Time Image Proc (2021).
8. AHMED, Imran, AHMAD, Misbah, RODRIGUES, Joel JPC, et al. A deep learning-based social distance monitoring framework for COVID-19. Sustainable Cities and Society, 2021, vol. 65, p. 102571.
9. MEIVEL, S., DEVI, K. Indira, MAHESWARI, S. Uma, et al. Real time data analysis of face mask detection and social distance measurement using Matlab. Materials Today: Proceedings, 2021.
10. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in Proceedings of the 28th International Conference on Neural Information Processing Systems, Ser. NIPS'15, pp. 91–99, MIT Press, Cambridge, MA, USA, 2015.
11. QIN, Jingchen et XU, Ning. Reaserch and implementation of social distancing monitoring technology based on SSD. Procedia Computer Science, 2021, vol. 183, p. 768-775.
12. Rahim A, Maqbool A, Rana T (2021) Monitoring social distancing under various low light conditions with deep learning and a single motionless time of flight camera.
13. Girshick R. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 1440–1448.
14. Girshick R, Donahue J, Darrell T, Malik J. Region-based convolutional networks for accurate object detection and segmentation. IEEE transactions on pattern analysis and machine intelligence. 2015;38(1):142–158.
15. Joseph Redmon, . "Darknet: Open Source Neural Networks in C." http://pjreddie.com/darknet/. (2013–2016).
16. Redmon, Joseph, and Ali, Farhadi. "YOLO9000: Better, Faster, Stronger".*arXiv preprint arXiv:1612.08242* (2016).
17. Barekatain, M., Miquel Martí, Hsueh-Fu Shih, Samuel Murray, K. Nakayama, Y. Matsuo and H. Prendinger. "Okutama-Action: An Aerial View Video Dataset for Concurrent Human Action Detection." 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2017): 2153-2160.
18. Du, Dawei, et Al. "VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results." 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW) (2019): 213-226.
19. Google colab, https://research.google.com/colaboratory/faq.html, last accessed 2021/07/16
20. Lin, Tsung-Yi, M. Maire, Serge J. Belongie, James Hays, P. Perona, D. Ramanan, Piotr Dollár and C. L. Zitnick. "Microsoft COCO: Common Objects in Context." ECCV (2014).