ADRAS: Airborne Disease Risk Assessment System for Closed Environments*

Wilber Rojas[1]✉, Edwin Salcedo[1], and Guillermo Sahonero[1,2]

[1] Department of Mechatronics Engineering
Universidad Católica Boliviana "San Pablo", La Paz, Bolivia
{wrojas,esalcedo}@ucb.edu.bo
https://www.ucb.edu.bo/
[2] Institute for Biological and Medical Engineering,
Pontificia Universidad Católica de Chile, Santiago, Chile
gsahonero@uc.cl

Abstract. Airborne diseases are easy to spread in the population. The advent of COVID-19 showed us that we are not prepared to control this type of disease. The pandemic has drastically posed challenges to the daily functioning of public and private establishments. In general, while there have been several approaches to reduce the potential risk of spreading the virus, many of them rely on the commitment that people make, which - unfortunately - cannot be constant, e.g., wearing a facemask in closed environment at all times or social distancing. In this work, we propose a computer-vision system to determine the risk of airborne disease spread in closed environments. We modify and implement the Wells-Riley epidemiological equation. We also evaluate the Openvino Models Zoo for people detection with mAP, precision, recall and F1-Score. For mask detection, we applied transfer learning and obtained the best performance for a model based on MobileNetV2. The generated data from several devices is visible in a web platform to monitor multiple areas and locations. Finally, an OAK-D camera and a Jetson device are embedded in a end device meant to monitor a closed environment and send spread risk data continually to the web platform. The results obtained are promising and suggest that such a system is beneficial to control, measure and prevent airborne contagion.

Keywords: Airborne Disease · Risk Assessment · Stereo Vision · Edge Computing.

1 Introduction

In the past few years, society has become more conscious of airborne diseases due to the disruption caused by the COVID-19 pandemic. Different from other illnesses transmitted between people, viruses or bacteria of airborne diseases can

* This research was supported by the OpenCV Foundation through an award provided at the OpenCV AI Competition 2021.

stay in the air mixed with dust particles and respiratory droplets for longer times. Then, these particles are eventually inhaled by other people and cause a spread of the disease. Airborne diseases comprise Measle, Tuberculosis, Chickenpox, Influenza, Pertussis, SARS-CoV-1, and SARS-CoV-2, among others. Even though the contagious rate and symptoms vary between them, their control and prevention are similar and consist mainly of the installation of isolation rooms for infected people, use of protective clothes such as Personal Protective Equipment (PPE), facemasks, and gloves, better ventilation for closed environments, and stricter practices of sanitation and hygiene. In-depth investigations on SARS-CoV-2 spread proved that the closer or denser a group of people, the higher the risk of airborne disease contagion [1].

The current research aims to propose a computer vision-based system to monitor the spread risk of an airborne disease. We considered Coronavirus 2019 as a study case for the investigation. This disease, also known as COVID-19, is caused by the SARS-CoV-2 virus and has different symptoms such as fever, breathing difficulties, fatigue, tiredness, cough, and loss of taste and smell [1]. So far, it has spread through 591 million people and caused 6,4 million deaths worldwide as of August 19, 2022 according to [2]. Although there is a greater percentage of healthy people that have not got reportedly infected, this disease has completely affected the lifestyle of the entire world population. While many developing countries chose to close borders and declare quarantines to stop the spread, others decided not to dictate strict measures to avoid economic slowdown and job instability [1]. Nevertheless, all developing countries were obligated to return to the new normality. Consequently, since a rapid increase of infected people with new variants can collapse any healthcare system, there is a constant research need for new control and prevention methods.

The paper is structured as follows. We first review some related works in Section 2. Then, we adapt an epidemiological model by considering several concepts from Wells-Riley estimation model in Section 3. Later, we explain the details of the computer vision models for person and mask detection in Section 4 which is complemented with the description of the distance estimation in Section 5. Moreover, we explain how the monitoring system works in Section 6 and present our results discussing their implications in Section 7. Finally, we conclude the paper in Section 8.

2 Related Works

By the end of 2019, the initial breakthrough of the SARS-CoV-2 virus led to a massive number of deaths and the declaration of a worldwide pandemic. This encouraged many computer vision developers and researchers to collaborate towards the development of new ideas to prevent the virus spread. Since keeping physical distance of at least one meter from others has proved to be one of the most effective measure against SARS-CoV-2 [2], extensive research has been carried out for the development of Visual Social Distance Monitoring Systems (VSDMS) [3]. Nevertheless, other projects complemented the distance measuring idea with face masks detection, or face-hand interaction [4] [5] [6] for controlling that people comply with measures against a contagion.

Computer Vision approaches are influenced by how images are taken, which, by default, is linked to the camera perspective and point-of-view. For example, authors in [4] used an overhead perspective while researchers in [7] used a top view, both to control a wider scene of an open area. Also, other studies propose distinct fields of application, from robots that continuously navigate through a room to monitor social distance [8], to headsets and wearables that prevent people from getting too close to each other by activating sound alarms [9].

The state-of-the-art proposals to measure physical contact among people can be classified as either 2D based or 3D based. The former commonly uses a sequence of methods to recognize people: a) image processing; b) image segmentation; c) shape extraction; d) object recognition. This last step might vary from classical computer vision-methods [10] to deep learning-based methods [11], [4], [12], [5], [6]. Focusing on deep learning methods, researchers usually implement object detectors based on Convolutional Neural Networks (CNNs) such as YOLOv4, Yolov5, MobileNet, SSD, or R-CNN [13] to measure social distance by detecting pedestrians. Then, the investigations regularly calculate the pairwise Euclidean distances among the centroids of the detected bounding boxes [4], [3]. These models are commonly applied to video processing given that monitoring an environment requires continuous surveillance. Consequently, in order to reduce the high computational cost, state-of-the-art proposals combine the person detectors with object tracking algorithms such as DeepSORT, SORT, and StrongSort [14], [15]. Given a set of bounding boxes enclosing people found by an object detector, object trackers focus on estimating or predicting the position of the bounding boxes in each consecutive video frame.

Even though camera-based surveillance systems have been developed up to the point of becoming commercialized solutions [9] [16] [17], this brief literature review let us note that 3D vision has been less explored despite its higher accuracy for distance calculation. A recent study has proven that 3D visual information can be estimated using monocular cameras to monitor social distance [18]. To the authors' best knowledge, researchers in [19] have been the first to publish about the application of stereo-vision in a VSDMS in April 2022. There, researchers describe a new VSDMS, which implements stereo and monocular cameras. Moreover, they explain their deployment process in an indoor hospital environment and conclude that stereo-vision cameras were superior than regular cameras. Specifically, they use a Zed M Camera and an MSI laptop equipped with NVIDIA GTX 1060 3GB GPU, with which they stream videos and obtain depth maps. Beyond the published work in indexed venues, developers in [20] and [21] apply stereo-vision cameras for social distance monitoring in September 2020 and January 2021, respectively.

So far, most of the previously mentioned research focused on detecting physical distance among people and face masks in order to notify authorities about protective measures noncompliance. Nevertheless, only a few works explore further implications and contagion risk assessment by leveraging the extracted visual information of the protective measures compliance. For instance, investigators in [15] contributed with a remarkable online infection risk assessment scheme for open environments named DeepSocial, which considers people's moving trajectories and the rate of social distancing violations to calculate the contagion risk. Moving on, researchers in [22] propose BEV-NET to assess social distancing compliance and probability of infection in closed environments using a monocular camera from a top perspective. Additionally, this investigation proposes the COVID-19 infection risk assessment for each individual present in a scene and a general risk assessment for the complete ambient. In both projects, the closer or denser a group of people is, the higher the risk of contagion [22], [15].

In contrast to COVID-19, the monitoring of infection risks of other airborne diseases such as MERS, Rhinovirus, or Adenovirus by recognising the protective rule's compliance is less investigated, however, their risk quantification can also be implemented using the approaches proposed for COVID-19. The multiple waves of SARS-CoV-2 have taught us that airborne diseases can severely affect the economy and normal functioning of an entire country, therefore, there is a shortage of research to have better tools that monitor and inform the risk of contagion in an ambient. All these to provide confidence to citizens and users in public and closed environments.

3 Epidemiological Model

This section describes the definition of the equation for estimating the spread risk of a airborne disease in a closed environment. We know that the classification of risk prediction models can be split into Wells-Riley based and Dose-Response based models [23]. We analyze the first one and modify it so that the implementation is feasible in a computer vision monitoring environment.

Wells developed an equation to estimate the risk of infection in a close environment [24]. Riley proposed an improvement to the equation, considering the ventilation of the room as a parameter [25]. The Wells-Riley equation generalizes the infectivity of pathogens with a new infectious dose unit called *quanta*. A *quanta* is the number of infectious particles required to infect a person [26]. So a *quanta* of influenza would infect the same number of people as a *quanta* of tuberculosis or COVID-19. If the disease is more contagious, the infected person would have a higher *quanta* emission rate.

The Wells-Riley equation is defined by Equation 1:

$$P_i = 1 - exp\left(\frac{I * q * p * t}{Q}\right) \tag{1}$$

Where, P_i is the probability of infection, I is the number of infected people, q is the quanta emission rate, p is the pulmonary ventilation rate of a person, t is the exposure time, and Q is the ventilation rate of the room [26]. For this investigation, each of these parameters were obtained as follows:

- Number of infected people (I):
  Wells-Riley estimates the risk of infection based on a certain number of infected people [26], however, we cannot know this information with entire certainty. We calculate the probability of infected people based on the population percentage of cases in a region:

  $$p_c = \frac{c}{P_r} \tag{2}$$

  Where, p_c is the percentage of cases, c is the number of cases in a region, and P_r is the total population of the region. If we multiply the percentage of cases by the number of people in a room, we obtain the probability of infected people in that room. So, our value for I is:

  $$I = N * p_c \tag{3}$$

  Where N is the number of people.

- Quanta emission rate (q):
  The quanta generation rate is the only parameter that contains the infectivity of the virus [26] [14], so each pathogen has its own value of q. Mikszewski et al. [27], analyzed the quanta generation rate for the most common airborne diseases, including SARS-CoV1, SARS-CoV2, MERS, Tuberculosis, and Influenza. These values are constants in the monitoring system and are described in Table 1.

Table 1: Quanta emission rate values $[q * h^{-1}]$ [27].

Pathogen	Resting, oral breathing	Standing, Speaking	Light activity, speaking loudly
SARS-CoV-1	0.0084	0.042	0.71
MERS	0.011	0.056	0.96
Tuberculosis (On Treatment)	0.020	0.098	1.7
Influenza	0.035	0.17	3.0
Coxsackievirus	0.062	0.31	5.2
Rhinovirus	0.21	1.0	18
SARS-CoV-2	0.55	2.7	46
Tuberculosis (Untreated)	0.62	3.1	52
Adenovirus	0.78	3.9	66
Measles	3.1	15	260

Li et al. determined that the viral load, therefore also the quanta, is almost the same between presymptomatic, asymptomatic, and symptomatic subjects [28]. In the case of the advent of a new airborne disease or actual disease variant, the value of q should be calculated using Equation 4:

$$q = c_v * c_i * p * v_d \tag{4}$$

where, c_v is the viral load, c_i is a conversion factor between a quanta and the infectious dose, p is the inhalation rate, and v_d is the volume of droplets expelled by a person [29].

- Pulmonary ventilation rate (p):
  Adams [30], conducted a study where he empirically determined the average person inhalation rate for different activities. Table 2 shows the values obtained from this study, which are used in Equation 1.

Table 2: Inhalation rate values [30].

Activity	Inhalation Rate [m3 h-1]
laying down	0.49
stand	0.54
very light exercise	0.72
light exercise	1.38
moderate exercise	2.35
heavy exercise	3.30

- Ventilation rate of the room (Q):
  It refers to the ACH (Air Changes per Hour) value of a close environment. To obtain this parameter, first we determined the air flow rate.

$$A_{FR} = w * A_v \tag{5}$$

In Equation 5, A_{FR} is the air flow rate, w is the window area, and A_v is the air velocity. Finally, the ACH is the Air flow rate divided by room volume (V_R)[31].

$$ACH = \frac{A_{FR}}{V_R} \tag{6}$$

In case of having artificial ventilation, the ACH value can be obtained from the specifications of the machine, and should be added to the natural ventilation calculated in the equation 6.

– Exposure time (t):
This parameter refers to the time that "I" number of infected people will remain in the closed environment. Note that the Wells-Riley formula requires the total exposure time as a parameter. And, in this research, the objective is to implement it in a monitoring system, so the number of infected people for a specific time is variable and will be calculated in real time.

We can represent the Wells-Riley equation with the use of integrals for the time variation:

$$P_i = 1 - exp\left(\int_0^{\infty} \frac{I * q * p}{Q} dt\right) \tag{7}$$

– Facemask detection:
Wells-Riley does not consider if people wear facemasks, which is required for estimating the spread of a disease in a closed environment. Therefore, we implement an additional parameter to the equation considering that we will monitor the presence of facemasks on the detected people's faces.

It is well-known that the worst type of facemasks are made of cloth, so we consider it as the default type of facemask used by everybody detected in a scene. It is worth noting that the use of cloth facemask reduces the contagion risk of an airborne disease by half [32]. From the Wells-Riley equation, we know that if we double the ventilation rate, we also halve the risk of spread. Consequently, the final equation is:

$$P_i = 1 - exp\left(\int_0^{\infty} \frac{I * q * p}{Q * (1 + M)} dt\right) \tag{8}$$

Where M is the percentage, in range $[0, 1]$, of people wearing a mask in a closed environment.

4 Person and Mask Detection

In the proposed system, two CNN-based object detection models localize person and facemask appearances in the frames coming from the OAK-D camera. The collected datasets, the object detection models, and most importantly, the used metrics are presented in the following subsections.

4.1 Dataset and Preprocessing

We collected the dataset of person and mask instances separately through three different means: web scrapping, video processing, and public datasets. First, we developed a Python script to download images from Google images. The terms used to find people images were: "pedestrians", "people in room", and "meeting". Once we obtained 377 images, we needed to review their quality due to some unrelated images downloaded by mistake. Second, we used the Computer Vision Annotation Tool (CVAT) [33] to obtain image samples of people and facemasks independently by labelling video frames. This let us get 1,084 instances of people and 337 instances of facemasks. Finally, we obtained 756 images by combining public datasets [34]. All these subsets were sorted to create two subsets: one with instances of people without facemasks, and another with images with facemask instances. Since the subsets were small, we needed to implement data augmentation techniques, such as horizontal flip, brightness change, and grayscale, by using the Roboflow platform [35].

Table 3: Datasets

Subset	Collected	Augmented	Total
Person Instances	1084	2277	3361
Facemask Instances	337	606	943

4.2 Object Detection

First, instead of creating and training object detection models from scratch, the person detection models that we tried were based on pre-trained architectures provided by Intel OpenVino [36]. Specifically, we tested the models person-detection-0200, person-detection-0201, person-detection-0202, person-detection-0203, and person-detection-0302 carefully to achieve a good performance with our collected dataset. In addition, it is worth mentioning that OpenVino models can be easily deployed on OAK-D devices by using the MiryiadX Blob Converter [37], which made us decide to use them. Secondly, for facemask detection, we applied transfer learning to re-train the models Yolov3, Yolov3 Tiny, and Mobilenet v2, which had been previously trained with bigger datasets. It is important to note that several developers who worked with OAK-D devices defined Mobilenetv2-based models as the best object detectors to deploy on them [36].

4.3 Metrics

Intersection Over Union The main tool to evaluate the human and facemask detection models with respect to each localized bounding box was the Intersection Over Union (IoU). This metric, also known as the Jacquard Index, measures the overlap area between the ground-truth bounding boxes and the predicted bounding boxes, and ranges between 0 and 1. For object detection tasks, it is recommended to set different IoU thresholds to take a detected bounding-box as true positive, true negative, false positive, or false negative. In the current project, we calculated used IoU@50%, which means we defined 0.50 as the threshold to calculate the the metrics.

Precision and Recall To evaluate the general performance of the object detection models with respect of a subset, precision and recall values were used. Precision shows the percentage of correct predictions among all the positive-predicted images while recall describes the percentage of actual positives that was identified correctly. Both are better defined in Equation 9 and Equation 10.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{9}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{10}$$

Mean Average Precision Commonly, precision and recall can be plotted against each other to obtain the precision-recall curve, and the Average Precision will be the area under this curve. This metric is defined in Equation 11, where r represents recall, p represents precision as a function of r. Therefore, $p(r)$ means "precision at recall r".

$$AP = \int_0^1 p(r)\ dr \tag{11}$$

Given that the person and facemask detection models were applied separately, Mean Average Precision (mAP) in our case is the same as Average Precision (AP). However, it is worth mentioning that mAP is the mean of Average Precisions of all individual classes for multi-class detection tasks and should be calculated as in Equation 12.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{12}$$

Here, mAP is Mean Average Precision, N is the number of class labels, and AP_i is the Average Precision for the i^{th} class. We considered and calculated mean average precision for different IoU thresholds: mAP@50% IoU, mAP@75% IoU, mAP@50%:5%:95% IoU.

F1-Score Finally, both precision and recall were used to calculate the F1-Score metric provided in Equation 13. The benefit of this metric is that it considers the number of prediction errors that the model makes and also the type of errors that are made. We calculated this metric with an IoU threshold of 0.5.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{13}$$

5 Distance Estimation

To calculate the distance among people, we need to estimate the relative position of the objects detected with respect of the camera. Stereo-vision helped us estimate these distances and positions by obtaining a three-dimensional view of a scene through the OAK-D camera and its binocular vision. Stereo-vision can be applied to calculate the depth of an object by making use of the angle of convergence.

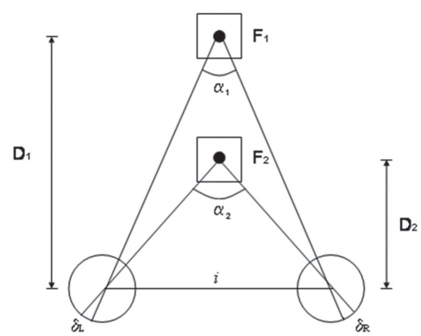


Fig. 1: Stereo-vision definition, where α is the convergence angle, D is the distance between the camera and the detected object, and i is the distance between cameras. F_1 and F_2 are different objects, which have different distances, and therefore, different angles [19].

As shown in the Figure 1, the convergence angle α_2 is in the middle of the two monocular cameras that capture an object located on the front [19]. Also, this angle varies depending on the distance of the object detected, this allows a precise approximation of its real distance. We used the OAK-D device built-in functions to calculate this distance. Specifically, the function used was "Spatial Location Calculator" [38]. In order to control social distancing, we calculated the Euclidean distance between each person, defined by Equation 14, where d is the distance between person p_1 and person p_2, and x, y, z refers to the positions in the three dimensional plane.

$$d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{14}$$

6 Monitoring System

Combining the result of Wells-Riley equation and the positions of the people determined by the stereo camera, we can analyze the sectors with the highest risk of infection in the monitoring area. We implement a Gaussian analysis to distribute the concentration of infectious particles. The Fig. 2 shows an analysis of a room monitoring COVID-19 for a period of 3 hours.

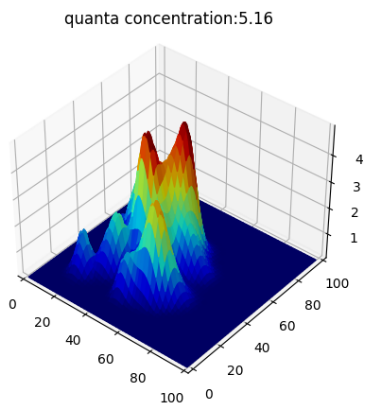
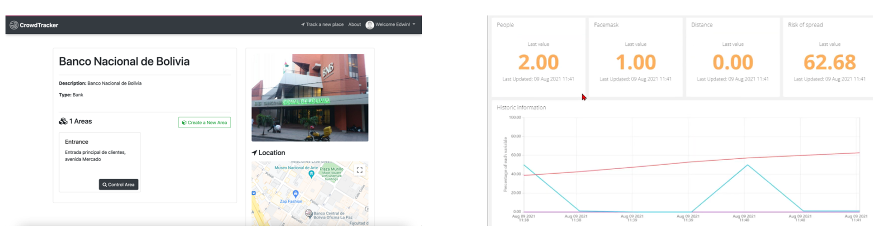


Fig. 2: Gaussian analysis of quanta concentration

In Fig. 2, the highest points represent the sectors where people remained for the longest time. Additionally, we send the data to an online monitoring system shown in Fig. 3. This system was developed as a web application and was deployed on a cloud server, which collected the airborne spread risk calculated from multiple end-devices. If the risk in an environment exceeded a threshold, the system sent notifications via Telegram to alert the authorities. Specifically, Fig. 3 (a) shows an example of the interface in which a bank can register multiple areas for real-time surveillance. The system was developed to monitor multiple areas by collecting data from end in several public places. For instance, the interface shown in Fig. 3 (b) was the detail view that helped us monitor a specific closed environment every 30 seconds.

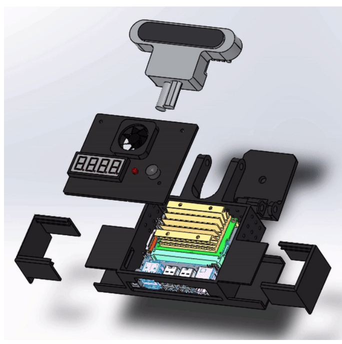


(a) Web application to monitor multiple places.

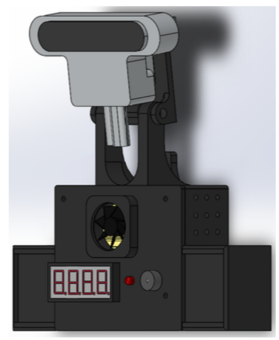
(b) Detail view of a closed environment.

Fig. 3: Online Monitoring System.

About the end-device, this was designed using SolidWorks, assembled with 3D printing and installed to monitor a closed environment continually. Internally, the device consisted of a Jetson Nano computer and an OAK-D camera. The final design is shown in Fig 4. The device has a screen that shows the risk of spread, as well as, activates an alert when the calculated risk exceeds a safety threshold.



(a) Disassembled.



(b) Assembled.

(c) Real Device.

Fig. 4: Monitoring Device.

7 Results and Discussion

In the validation of our Wells-Riley model, the results shows that our equation obtains a value close to the study of [39]. We used the same input in the equation 8 and the result obtained was 9.88 which is similar to the result obtained by the reference study (10.0). This value was expected because the equation was not modified in essence, and the structure was only slightly changed to allow different input values. We can conclude that the small difference between results in [39] and our result is because the authors did not report their result with decimal values.

For the person detection models, we evaluated several architectures provided on the OpenVino public repository [36]. These tests were performed with the total subset named "Person Instances", described in Table 3. As mentioned above, we mainly decided to use these models because they were already trained with bigger datasets for person detection. The resulting metrics are shown in Table 4, which shows that the the model person-detection-0203 obtained the best result for the mAP metric. In this table, the column Complexity defines the number of computational operations to pass a frame through the model. Size define the footprint of the memory needed for each prediction. Given that person-detection-0203 obtained the best performing results, it was deployed in the OAK-D device for continuous tracking.

Table 4: Results of the tested models for person detection from [36]. Complexity column is in GFLOPS. Size column is in Mp. mAP, Precision, Recall, and F1-Score are in percentage.

Model	**Complexity**	**Size**	**mAP**	**Precision**	**Recall**	**F1-Score**
person-detection-0200	0.786	1.817	70.23	71	59	64.46
person-detection-0201	1.768	1.817	67.83	63.86	52.3	57.50
person-detection-0202	3.143	1.817	71.39	68.34	55.6	61.31
person-detection-0203	6.519	2.392	73.02	72.1	62.1	66.7
person-detection-0302	370.208	51.164	71.23	69.85	56	62.2

Using the obtained datasets, we train a model with Tensorflow in Google Colab. This task was performed using the Luxonis [40] training instructions. Transfer learning was used with MobileNetV2 and the results of the training are described in the table 5.

Table 5: Results of the trained model with MobileNetV2 for facemask detection.

Image size	mAP
Large	0.49738526
Medium	0.25847965
Small	0.06959111
IoU=.50	0.57984364

The table 5 shows that the model is not good at making inferences on small or distant objects. However, in the implementation the model proved to have excellent inferences.

8 Conclusions

In this paper, we proposed the implementation of epidemiological models in computer vision systems, with the aim of reduce the spread of airborne diseases. We develop a new version of the Wells-Riley equation capable of calculate the risk of infection in real time. We use object detection models to determine the number of people in an environment, as well as the number of people wearing masks. We implement stereo-cameras and Gaussian mathematical models to obtain a three-dimensional map of the sections with the highest risk of infection. All information is sent to an online monitoring system, with which multiple environments can be monitored. An important limitation is that our percentage of infected (I) is an estimate and it cannot be guaranteed that it is the real value of infected in the environment. An oversize can be added to this value in order to prevent false security values.

References

1. S. E. Park, "Epidemiology, virology, and clinical features of severe acute respiratory syndrome -coronavirus-2 (SARS-CoV-2; Coronavirus Disease-19)," *Clinical and experimental pediatrics*, vol. 63, no. 4, pp. 119–124, apr 2020. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32252141
2. "WHO: Coronavirus (COVID-19) Statistics." [Online]. Available: https://covid19.who.int/
3. Y. Himeur, S. Al-Maadeed, N. Almaadeed, K. Abualsaud, A. Mohamed, T. Khattab, and O. Elharrouss, "Deep visual social distancing monitoring to combat COVID-19: A comprehensive survey," *Sustainable Cities and Society*, vol. 85, p. 104064, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210670722003821
4. I. Ahmed, M. Ahmad, J. J. P. C. Rodrigues, G. Jeon, and S. Din, "A deep learning-based social distance monitoring framework for COVID-19," *Sustainable Cities and Society*, vol. 65, p. 102571, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210670720307897
5. M. Razavi, H. Alikhani, V. Janfaza, B. Sadeghi, and E. Alikhani, "An automatic system to monitor the physical distance and face mask wearing of construction workers in covid-19 pandemic," *CoRR*, vol. abs/2101.0, 2021. [Online]. Available: https://arxiv.org/abs/2101.01373
6. F. I. Eyiokur, H. K. Ekenel, and A. Waibel, "A Computer Vision System to Help Prevent the Transmission of COVID-19," *undefined*, 2021.
7. A. A. A. Shareef, P. L. Yannawar, A. S. H. Abdul-Qawy, and Z. A. Ahmed, "Yolov4-based monitoring model for covid-19 social distancing control," *3rd International Conference on Smart IoT Systems: Innovations and Computing, SSIC 2021*, vol. 235, pp. 333–346, 2022. [Online]. Available: https://doi.org/10.1007/978-981-16-2877-1_31
8. A. J. Sathyamoorthy, U. Patel, Y. A. Savle, M. Paul, and D. Manocha, "COVID-Robot: Monitoring Social Distancing Constraints in Crowded Scenarios," *CoRR*, vol. abs/2008.0, 2020. [Online]. Available: https://arxiv.org/abs/2008.06585
9. V. Kamalasanan and M. Sester, "Living with Rules: An AR Approach," *Adjunct Proceedings of the 2020 IEEE International Symposium on Mixed and Augmented Reality, ISMAR-Adjunct 2020*, pp. 213–216, nov 2020.

10. N. Petrovic and D. Kocić, *IoT-based System for COVID-19 Indoor Safety Monitoring*, sep 2020.
11. S. Degadwala, D. Vyas, H. Dave, and A. Mahajan, "Visual Social Distance Alert System Using Computer Vision Deep Learning," *Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2020*, pp. 1512–1516, nov 2020.
12. D. Yang, E. Yurtsever, V. Renganathan, K. A. Redmill, and Ü. Özgüner, "A Vision-Based Social Distancing and Critical Density Detection System for COVID-19," *Sensors 2021, Vol. 21, Page 4608*, vol. 21, no. 13, p. 4608, jul 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/13/4608/htmhttps://www.mdpi.com/1424-8220/21/13/4608
13. O. Karaman, A. Alhudhaif, and K. Polat, "Development of smart camera systems based on artificial intelligence network for social distance detection to fight against COVID-19," *Applied Soft Computing*, vol. 110, p. 107610, oct 2021.
14. J. Li and Z. Wu, "The application of yolov4 and a new pedestrian clustering algorithm to implement social distance monitoring during the covid-19 pandemic," *Journal of Physics: Conference Series*, vol. 1865, no. 4, apr 2021. [Online]. Available: https://doi.org/10.1088/1742-6596/1865/4/042019
15. M. Rezaei and M. Azarmi, "DeepSOCIAL: Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemic," *Applied Sciences*, vol. 10, no. 21, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/21/7514
16. "Delta variant: 8 things you should know | Coronavirus | UC Davis Health." [Online]. Available: https://health.ucdavis.edu/coronavirus/covid-19-information/delta-variant.html
17. "Covid-19 vaccine tracker: View vaccinations by country." [Online]. Available: https://edition.cnn.com/interactive/2021/health/global-covid-vaccinations/
18. L. Bertoni, S. Kreiss, and A. Alahi, "Perceiving humans: From monocular 3d localization to social distancing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7401–7418, July 2022.
19. M. L. M.D., N. Varble, B. T. M.D., S. Xu, and B. J. W. M.D., "Camera-based distance detection and contact tracing to monitor potential spread of COVID-19," in *Medical Imaging 2022: Image Perception, Observer Performance, and Technology Assessment*, C. R. Mello-Thoms and S. Taylor-Phillips, Eds., vol. 12035, International Society for Optics and Photonics. SPIE, 2022, p. 120351D. [Online]. Available: https://doi.org/10.1117/12.2612846
20. ibaiGorordo, "Ibaigorordo/social-distance-feedback-for-the-blind: A social distancing feedback system for the blind using the oak-d camera." [Online]. Available: https://github.com/ibaiGorordo/Social-Distance-Feedback-For-The-Blind
21. R. Kanjee, "Social distance detection system — using raspberry pi and opencv ai kit." [Online]. Available: https://medium.com/augmented-startups/social-distance-detection-system-using-raspberry-pi-and-opencv-ai-kit-97fd68ff8dd4
22. Z. Dai, Y. Jiang, Y. Li, B. Liu, A. B. Chan, and N. Vasconcelos, "Bev-net: Assessing social distancing compliance by joint people localization and geometric reasoning," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2021, pp. 5381–5391.
23. R. Mittal, C. Meneveau, and W. Wu, "A mathematical framework for estimating risk of airborne transmission of covid-19 with application to face mask use and social distancing," *Physics of Fluids*, vol. 32, p. 101903, 10 2020.
24. W. Wells, "Airborne contagion and air hygiene. an ecological study of droplet infections." *Cambridge MA*, 1955.

25. E. Riley, G. Murphy, and R. Riley, “Airborne spread of measles in a suburban elementary school,” *American Journal of Epidemiology*, vol. 107, 1978.
26. G. N. S. To and C. Y. H. Chao, “Review and comparison between the wells-riley and dose-response approaches to risk assessment of infectious respiratory diseases,” *Indoor Air*, vol. 20, 2 2010.
27. A. Mikszewski, L. Stabile, G. Buonanno, and L. Morawska, “The airborne contagiousness of respiratory viruses: A comparative analysis and implications for mitigation,” *Geoscience Frontiers*, p. 101285, 8 2021.
28. J. Li, Z. Cheng, Y. Zhang, N. Mao, S. Guo, Q. Wang, L. Zhao, and E. Long, “Evaluation of infection risk for sars-cov-2 transmission on university campuses,” *Science and Technology for the Built Environment*, vol. 27, pp. 1165–1180, 10 2021.
29. G. Buonanno, L. Morawska, and L. Stabile, “Quantitative assessment of the risk of airborne transmission of sars-cov-2 infection: Prospective and retrospective applications,” *Environment International*, vol. 145, p. 106112, 12 2020.
30. W. Adams, “Measurement of breathing rate and volume in routinely performed daily activities,” 1993.
31. R. Teppner, B. Langensteiner, W. Meile, G. Brenn, and S. Kerschbaumer, “Air change rates driven by the flow around and through a building storey with fully open or tilted windows: An experimental and numerical study,” *Energy and Buildings*, vol. 80, pp. 570–583, 9 2014.
32. S. E. Eikenberry, M. Mancuso, E. Iboi, T. Phan, K. Eikenberry, Y. Kuang, E. Kostelich, and A. B. Gumel, “To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the covid-19 pandemic,” *Infectious Disease Modelling*, vol. 5, 2020.
33. “Computer Vision Annotation Tool,” 2022. [Online]. Available: https://cvat.org/
34. GotG, “How to train an object detector using mobilenet ssd v2,” 2020. [Online]. Available: https://github.com/GotG/test_object_detection_demo/tree/master/data/medmask_voc
35. “Roboflow,” 2022. [Online]. Available: https://roboflow.com/
36. “Overview of OpenVINO™ Toolkit Intel’s Pre-Trained Models,” 2022. [Online]. Available: https://docs.openvino.ai/latest/omz_models_group_intel.html
37. Luxonis, “Luxonis mytiadx blob converter.” [Online]. Available: http://blobconverter.luxonis.com
38. Luxonis, “Spatial location calculator.” [Online]. Available: https://docs.luxonis.com/projects/api/en/latest/components/nodes/spatial_location_calculator/
39. F. Velarde, R. Rub´, R. Mamani-Paco, and M. Andrade-Flores, “Estimation of the probability of contagion of covid-19 by aerosols in closed environments: Applications to cases in the city of la paz, bolivia,” vol. 37, pp. 22–30, 2020. [Online]. Available: http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1562-38232020000200004&lng=es&nrm=iso
40. GotG, “How to train an object detector using mobilenet ssd v2,” 2020. [Online]. Available: https://github.com/GotG/test_object_detection_demo