A monitoring system for evaluation of COVID-19 infection risk

Jevgenis Telicko, Dagis Daniels Vidulejs and Andris Jakovičs
University of Latvia, Institute of Numerical Modelling, Jelgavas 3, Riga, Latvia
E-mail: jevgenij.s.telicko@gmail.com

Abstract.
Monitoring systems allow operators to accomplish the greatest comfort indoors, but, as a rule, the available parameters are not enough to analyse the epidemiological threat in buildings. Due to the pandemic and increasing incidence of the disease, there is a need for monitoring systems that can provide the necessary information to analyse the risk of infection. With timely notification of people about the risks, such a system could not only increase safety in buildings, but also save crucial resources such as the work of medical personnel. This paper presents an example of real-world implementation of a cheap and scalable system to indicate risks and inform people inside. To achieve this, an appropriate set of sensors and communication protocols was selected, and processing of indirect measurements with artificial intelligence (AI) algorithms was carried out on an embedded Jetson Nano computer. Based on the experiments and a review of the literature, the necessary parameters for measurements were selected. Detailed analysis of measured data for risk evaluation is provided in [1].

1. Introduction
The Covid-19 pandemic has had a significant impact on the global economy and on people’s lives. As respiratory infections such as Covid-19 can be transmitted by aerosol, the risk of infection indoors is much higher [2]. In addition to this, according to statistics, people spend 86.9% of their time indoors [3]. Current monitoring systems can maintain high thermal comfort inside buildings and prevent mold growth inside structures [4]. It is known that some parameters such as relative humidity, temperature and amount of CO₂ can affect infection risk [5]. However, as a rule, standard building management system (BMS) measurement parameters are not enough to analyze the epidemiological threat in buildings, as the risk of infection depends on potentially dangerous particles in air and the number of people. Not all buildings have built-in monitoring systems.

There are some solutions such as the Aranet wireless measurement system, but, unfortunately, their measurement parameters are not sufficient in order to evaluate the risk of infection - at the very least, one must know how many people are inside a given room. Thus, there was a need for a portable monitoring system that could provide all of the necessary information. With timely notification of people about the risks, such a system could not only increase safety in buildings, but also save on the labour of medical personnel. As an additional benefit, such a system could give additional information to the BMS, if one exists, to change the ventilation parameters and decrease the risk of infection. It is hard to detect virus in air as its size is measured on a nanometer scale[6] - to evaluate the risk, therefore, indirect measurement techniques are necessary.

This article provides a real-world example of the implementation of a portable system to indicate risks and inform people inside. One of the goals was to create a cheap, scalable system. For this, an...
appropriate set of sensors and communication protocols were selected, as well as processing of indirect measurements with neural networks was carried out on an embedded computer Jetson Nano. Based on a review of the available literature, it was found that humans can produce potentially dangerous amounts of particles (aerosol) containing SARS-CoV-2 during coughing, speaking, sneezing etc. [7, 8]. These kinds of actions are usually accompanied by specific sounds which may be detected using most microphones and classified by specific processing algorithms proposed in this article. As aerosols produced by people contain salt, which in theory could stay in the air after the aerosol dries up, an additional dust sensor was added to the standard suite of measurements. To detect and count people, an additional AI sensor was used.

The University of Latvia and Pauls Stradiņš Clinical University Hospital in Riga, Latvia were selected as the benchmark locations for the system. Detailed analysis of the measured data for risk evaluation is provided in [1].

2. Set of sensors with direct measurements and experiments
Measures such as CO₂, humidity and dust can be affected by several factors such as nearby heaters and ventilation ducts. Therefore, the decision was made to create a separate board due to the high likelihood that the optimal location for camera installation could be close to a ventilation duct or another object that might influence the measurements.

As the base for the sensor board was chosen Arduino Nano with the Atmega328P microcontroller. It is a very compact board with I2C, SPI and UART communication protocols, which are required for communication with other components. To make the installation simple it was decided to transmit data to Jetson Nano wirelessly. Since data transmission will not be carried out over long distances, other modules such as Lora could be replaced with cheaper alternatives. A nrf24 radio module, priced at less than 2$ per unit, was selected for wireless data transmission. According to our measurements, nrf24 with a power amplifier could require more than 100 mA current at the beginning of transmission, which is why an additional 3.3V stabilizer was added to the circuit. Because Arduino logic operates at 5V, but nrf24 operates at 3.3V, an additional logic level converter was used.

![Figure 1. PM measurements with humidifier filled with salt solution and distilled water. The interval between the red lines marks when the humidifier was in operation.](image)

Initial research and several experiments were performed to determine which measurement parameters could provide more information about the state of the premises. One unusual measurement parameter is the dust sensor. The idea is to measure solid particles which may remain from the aerosol exhaled by a person. To confirm this theory, an experiment was carried out in which distilled water and a 5% solution of water with salt were poured into an air humidifier powered by a piezoelectric device. Particulate matter (PM) values PM10 and PM2.5 were measured in 3 different states - humidifier switched off, working with distilled water, and working with the 5% salt solution. PM values were measured in a 6x7 m room...
with the SDS011 sensor positioned approximately 7 m away from the humidifier. When the humidifier was switched off, the values were the same as when it vaporised distilled water (PM10=10, PM2.5=7).

When the humidifier was loaded with a 5% salt solution, upon switching on the humidifier, values started to significantly grow and in 400 seconds reached levels of over 100 each (see figure 1). This sensor was included in the final board.

For CO₂ measurements SCD30 with NDIR measurement technology was chosen. Compared to cheaper sensors which provide CO₂ measurements, it is less affected by other gases and it will be easier to debug the model in [1]. Because the sensor comes as a module, there is a much lower chance of miscalibration during soldering as opposed to SMD modules. In addition to this, the module has built-in relative humidity and temperature sensors, useful for sensor reading correction and reachable by Arduino on I2C bus. The disadvantage of SCD30 and SDS011 is that both require quite a lot of current - 20 mA and 70 mA, respectively. A power supply was chosen for the power source, since a battery solution would be expensive and have large dimensions.

After some tests, it became clear that the built-in temperature/relative humidity sensor in SCD30 is affected by the CO₂ measuring part. The measured temperature was elevated by around 2°C. To solve this problem, a separate temperature and relative humidity sensor SHT30 was added. Unfortunately, more involved measurement analysis in [1] did not show dust measurements correlating with other measured parameters, but this sensor was still useful in detecting rapid changes in the amount of particles in air - usually caused by changing the ventilation system mode or opening a window. This information can be useful in debugging risk evaluation models.

3. People-counting sensor

Many supermarkets use optical interrupt sensors to count people. The disadvantage of such systems is that if there is only one sensor, how many people have entered and how many have left cannot be accurately determined. By supplementing such a system with a second sensor to determine the trajectory of a person, in theory, is possible to get the exact number of people. However, if people enter and exit at the same time, or move in groups, an error is likely to occur and the system will require recalibration.

An alternative solution is to use a camera with further image processing based on an AI algorithm. For neural networks (NNs) to work correctly, it is necessary to create an architecture and train the network using a number of examples. A number of difficulties arise here, since in this case, saving photographs entails bureaucratic difficulties associated with personal data processing which take time to resolve; furthermore, obtaining and labelling the required amount of data can be time-consuming. To implement the project in due time, ready-made architectures and weightings with image pre-processing were used in the development of the sensor.

In addition to its low price and included GPU, Jetson Nano was chosen because the operating system already contains the CUDA package, and the installation of libraries such as the full version of TensorFlow is also possible. During the experiments, the lack of the board was detected under load - unless additional cooling is used, the throttle is triggered and performance drops dramatically.

During initial tests, a web camera was connected to the Jetson Nano via USB. For object detection and semantic segmentation, convolutional neural networks (CNN) [9, 10] are often used. In the first round of experiments an open-source framework TensorFlow Object Detection API with “SSD with Mobilenet” model from [11] was chosen and tested in a laboratory room (see figure 2).
As seen in the photo, the camera’s field of view is not enough to capture the entire room. Therefore, a camera with a 170-degree angle (IMX219-170) was chosen in order to capture the whole room. As a bonus, this module is smaller, has better resolution up to 3280 × 2464, and with a MIPI CSI-2 connection, the price is lower than that of USB webcams, which have greatly increased in price during the pandemic. The new camera provides an overview of the whole room but has a fisheye effect. Because the model training dataset had no fisheye effect, the neural network’s accuracy decreased. To overcome this problem, an OpenCV fisheye correction module was used. To compute K and D coefficients, several photos were taken with a calibration chess board, with the result shown in figure 3.

Recognition accuracy improved, but the system was poor at recognizing people at the far table. Consequently, several other NN models were tested such as YOLO v3, CSRNet and Mask R-CNN [12, 13, 14]. The best results were obtained from Mask R-CNN with mask_rcnn_coco weights. This NN was chosen for all future experiments. Unfortunately, even with the more complicated network, it was difficult to detect people positioned far away from the camera. It is known that small object detection is a difficult task for computer vision [15]. A decision was made to install a camera with a top-down viewing angle in the middle of the rooms; this way, the maximum distance to the camera is reduced.

When lighting was turned off at night and in the morning, the network often gave errors, indicating that there were several people in the room. To eliminate the error, an additional check and filter was added. Firstly, the camera calculates the mean value of pixels and checks if it is higher than threshold - identifying if the image is completely dark, in which case the number of people is assumed to be 0. It is more difficult in the morning when lighting is not switched on but there might be people in the rooms. After a detailed examination of the pictures taken in low light, it became clear that the camera produces color artifacts in such conditions. Another threshold was therefore defined for removing Gaussian noise. For that purpose fastNLMeansDenoisingColored from OpenCV was used.
It was observed that the NN almost never gave a false positive result, but quite often would indicate fewer people than there were. An additional buffer of 5 images was therefore created, rotated by the FIFO principle, and the response of the NN was the maximum number detected within that buffer. After all modifications, the NN was able to successfully count people inside the office, but unfortunately if people in the big laboratory room were close to each other, the NN still would give wrong values, indicating fewer people than actually present.

4. Sound sensor

The purpose of this system is to accurately and reliably predict the semantic label of an auditory event, in this case to distinguish coughing and sneezing sounds. Modern machine learning models capture the temporal nature of the audio signal. Two of the most promising types of models for this problem are Recurrent Neural Networks (RNNs) and CNNs.[16]

An example of a successful CNN is Google’s large-scale Audioset model, classifying over 30,000 different labels and trained on 5.4 M hours of audio.[17] Recently, leading results with CNN for auditory event detection have been achieved with transfer learning from image recognition CNNs.[18] However, a lighter architecture was chosen to compute in real-time on the embedded computer.

Encouraged by results in [19], an RNN BLSTM architecture was adopted. Firstly, the suggested architecture is resilient to noise and has exceptionally high accuracy on the binary classification task - 92.15% to 88.11% at different sound to noise ratios (SNR). Secondly, the problem of anomalous sound detection strongly resembles the problem of cough and sneeze detection, since these auditory events are both rare and short in duration.

A recent comparative study for acoustic cough detection in [20] shows some popular models and input features, using area under the receiver operating characteristic curve (AUC) as the performance metric. The semantic label count is the same as in this article, however a single metric is too ambiguous for definite comparisons, and some conditions are different. Otherwise, this model ranks among the best models in the study and has a training AUC of 0.9742 and a test AUC of 0.9568.

In the middle of 2020, there were very scarce resources of correctly labeled cough and sneezing sounds available. Similarly to other studies on cough recognition, the database was collected from multiple public datasets, often samples were picked by hand from online resources [21]. The resources and datasets used include, but are not limited to ESC-50: Dataset for Environmental Sound Classification [22], the Speech Accent Archive, Freesound.org [23] user-submitted sounds and the research dataset from the COVID-19 Sounds App by the University of Cambridge [24].

The training dataset duration for sound classes is as follows: 47 min. of sneezing, 67 min. of coughing, 77 min. of laughing, 179 min. of speaking and 296 min. of noise. Using augmentation techniques, the durations were all extended to 320 minutes. Each training sample was cut into 2.4 second segments and segments shorter than 1.7 seconds were discarded.

The dataset was augmented to resemble real-life conditions and statistical distributions. This in turn increased the accuracy, robustness and generalization of the AI model. One type of augmentation introduces different types of auditory noise into the samples, other techniques are pitch shifting, time shifting and amplitude-based segment repetition [25, 26].

In testing it was found that condenser microphones are necessary in order to capture sound from the whole room, because of high sensitivity retention at longer distances compared to dynamic microphones. The microphone used in the final product is the Blue Snowball omni-directional USB condenser microphone, but other microphones could also be used such as conference room condenser microphones.

The methods in the audio processing pipeline are inspired by automatic speech recognition, see figure 4 for an overview. The prevalent input feature for audio tasks in machine learning is the spectrogram, containing time and frequency axes. To create a spectrogram, the audio signal is divided into overlapping frames for which the fast Fourier transform is computed to obtain the frequency spectrum. The spectrogram is then transformed into the nonlinear mel-scale which resembles the human frequency perception curve, before taking the log of the magnitude. To combat the problem of stationary auditory
noise, an experimental method of convolutional denoising is applied to the spectrogram, described further in this article. Studies show that robust spectrogram-based features outperform mel-frequency cepstral coefficients (MFCC) under noisy conditions [27]. Nevertheless the MFCC is computed after denoising in this case, reducing the input dimensions and the BLSTM network complexity.

The practical consideration for the approach described in [19] is the high robustness to noise, which is needed to support different ambient noise conditions and room sizes as well as to reduce variance due to different microphones. The results can be attributed to the auto-encoder network approach, extracting a deep audio representation from three different input features. In practice, it was found that any single fully connected neural network auto-encoder is more computationally expensive than the main BLSTM network, which generates the classification predictions by itself - this was problematic due to the limited resources on the embedded computer.

This study employed a different approach for increasing the reliability of MFCC features under noisy conditions, namely the denoising convolutional autoencoder as described in [28]. Although CNN is less frequently used on embedded devices, this auto-encoder is not demanding on computation or memory. It contains just 4 convolutional layers, the first layer contains (11x11) gabor filters as kernels and the other are (5x5) learned kernels. The size of the spectrogram is also small at (200x54). See figures 5 and 6 for spectrogram comparison.

**Figure 4.** Sound data processing pipeline.

**Figure 5.** Spectrogram before denoising.  
**Figure 6.** Spectrogram after denoising.

It is important to reduce false positive results for coughs and sneezes. The approach taken was to assign a larger weight to the correct or incorrect classification of noise. The trade-off is lower sensitivity of the model. The inter-class similarity of coughing, sneezing and laughing also adds to the difficulty of the problem. In figure 7 it can be seen that the confusion matrix between coughing and sneezing is mutually less discerned.

It must be noted that the model is monophonic, meaning it can only classify the dominant sound at any moment. This is problematic when, for example, there is a person speaking and another person coughs at the same time. The prediction then will be either speech or cough, but not both.
generally machine learning algorithms require specifically standardized input data to work correctly outside of testing conditions. There was a problem with the sound sensor when the microphone produced a waveform with values outside the input range, which caused random output predictions. The problem can be solved with a simple algorithm to dynamically find the optimal volume level for the microphone at any given time.

Another problem arose with different room impulse responses (RIRs) from those in the dataset. Room size and damping by walls have a large impact on acoustics, which leads to performance drawbacks with the machine learning model. To solve this problem, the training data can be augmented to simulate reverb by convolving the sounds with different RIRs.

5. Communication and people notification
To provide a lightweight and easy-to-use connection to many independent sensors in different rooms the MQTT protocol with Transport Layer Security was selected. For this purpose an MQTT broker on Jetson Nano was created. To streamline the system, a communication topology was created, in which each building had its own unique topic. Rooms in buildings are sub-topics of the respective buildings. Each measurement type is a sub-topic of its room and each sensor has a unique ID which is a sub-topic of the measurement type. This way it is easy to debug the system and analyse data. To inform people an LED tower indicator was created which, according to room name, is subscribed to the risk value subtopic.

6. Summary
Thanks to the use of embedded computers, it was possible to create the monitoring system at low cost. For mass installations, the price can be further reduced by using a single powerful computer in combination with cheaper nodes such as ESP32 for data re-transmission from multiple rooms. However, this may require additional regulations from the government related to private data processing. Wireless communications can greatly simplify the insulation of the devices. The developed system allows counting the number of people, analyzing sound information and measuring the necessary environmental parameters to provide data for evaluating the risk of infection according to the model in [1]. Measurements of parameters such as the number of people in a room can be significantly improved by using an additional pre-processing and supplementing the network with training data from a specific measurement location. The provided solution is able to easily distinguish human vocalisations and respiratory sounds from noise, although quite some ambiguity exists between the sounds of sneezing and coughing. With increasing demand for COVID-19 risk warning systems, institutions are collecting auditory samples, which enables AI sound monitoring systems to vastly improve.

Acknowledgement
This study was conducted with the financial support of project “New Technologies for Targeted Tracing, Testing and Treatment of COVID-19 Patients”, No. VPP-COVID-2020/1-0025.
References

[1] Virbulis J, Sjomkane M, Surovovs M and Jakovics A 2021 Numerical model for prediction of indoor covid19 infection risk based on sensor data (Preprint 8th International Building Physics Conference in Copenhagen)

[2] Asadi S, Bouvier N, Wexler A and Ristenpart W 2020 Aerosol Science and Technology 0 1–4

[3] Klepeis N, Nelson W, Ott W and Robinson J 2001

[4] Yang S, Wan M, Wanyu C, Ng B and Dubey S 2020 Applied Energy 271 115147

[5] Ahlawat A, Wiedensohler A and Mishra S 2020 Aerosol and Air Quality Research 20 1856–61

[6] Neuman B, Kiss G, Kunding A, Bhella D, Baksh M, Connelly S, Droese B, Klaus J, Makino S, Sawicki S, Siddell S, Stamou D, Wilson I, Kuhn P and Buchmeier M 2011 J Struct Biol 174 11–22

[7] Lindsley W, Pearce T, Hudnall J, Davis K, Davis S, Fisher M, Khakoo R, Palmer J, Clark K, Celik I, Coffey C, Blachere F and Beezhold D 2012 Journal of occupational and environmental hygiene 9 443–9

[8] Eiche T and Kuster M 2020 International Journal of Environmental Research and Public Health 17 9088

[9] Girshick R 2015 IEEE International Conference on Computer Vision

[10] Long J, Shelhamer E and Darrell T 2015 pp 3431–40

[11] Ozlu A 2018 Tensorflow object counting api URL https://github.com/ahmetozlu/tensorflow_object_counting_api

[12] Li Y, Zhang X and Chen D 2018 Csnet: Dilated convolutional neural networks for understanding the highly congested scenes

[13] Redmon J and Farhadi A 2018 Yolov3: An incremental improvement

[14] He K, Gkioxari G, Dolla’r P and Girshick R 2017 IEEE International Conference on Computer Vision (ICCV) pp 2980–88

[15] Pang J, Li C, Shi J, Xu Z and Feng H 2019 IEEE Transactions on Geoscience and Remote Sensing 57

[16] Yu D and Li J 2018 Recent progresses in deep learning based acoustic models (updated) (Preprint 1804.09298)

[17] Hershey S, Chaudhuri S, Ellis D P W, Gemmeke J F, Jansen A, Moore R C, Plakal M, Platt D, Saurous R A, Seybold B, Slaney M, Weiss R J and Wilson K 2017 Cnn architectures for large-scale audio classification (Preprint 1609.09430)

[18] Palanisamy K, Singhania D and Yao A 2020 Rethinking cnn models for audio classification

[19] Li Y, Li X, Zhang Y, Liu M and Wang W 2018 IEEE Access 6 58043–55

[20] Miranda I D S, Diac1 A H and Niesler T R 2019 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) pp 2601–05

[21] Gheorghe Pop Horia Cucu D B and Burileanu C 2020 ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY 23 S77–88

[22] Piczak K J 2015 ESC: Dataset for Environmental Sound Classification

[23] Fonseca E, Pons J, Favory X, Font F, Bogdanov D, Ferraro A, Oramas S, Porter A and Serra X 2017 Proceedings of the 18th International Society for Music Information Retrieval Conference pp 486–93

[24] Brown C, Chauhan J, Grammenos A, Han J, Hasthansasombat A, Spatis D, Xia T, Cicuta P and Mascolo C 2020 International Conference on Knowledge Discovery and Data Mining

[25] Salamon J and Bello J P 2016 CoRR abs/1608.04363 (Preprint 1608.04363)

[26] Prayitno B A and Suyanto S 2019 Procedia Computer Science 157 420–26 ISSN 1877-0509 the 4th International Conference on Computer Science and Computational Intelligence (ICCSCI 2019) : Enabling Collaboration to Escalate Impact of Research Results for Society

[27] Shahrin M and Huzaifah M 2017 Comparison of time-frequency representations for environmental sound classification using convolutional neural networks

[28] Kayser M and Zhong V 2015 Denoising convolutional autoencoders for noisy speech recognition