

# FluNet: An AI-Enabled Influenza-like Warning System

Ryan Ward <sup>1</sup>, Fred Jjunju <sup>2</sup>, Elias Griffith <sup>2</sup>, Alan Marshall <sup>2</sup>, Stephen Taylor <sup>2</sup>, Isa Kabenge <sup>2</sup>, Noble Banadda <sup>2</sup>, and Rhoda Wanyenze <sup>2</sup>

<sup>1</sup>University of Liverpool

<sup>2</sup>Affiliation not available

October 30, 2023

## Abstract

Influenza, an acute viral respiratory disease that is currently causing severe financial and resource strains worldwide. With the recent COVID-19 pandemic exceeding 153 million cases worldwide, there is a need for a low-cost and contactless surveillance system to detect symptomatic individuals, more so in counties with limited healthcare resources. As with many diseases, there are bio-clinical signals relating to the physical symptoms. The main objective of this study was to develop FluNet, a novel, proof-of-concept, low-cost and contactless device for the detection of high-risk individuals. The system passively conducts face detection in the longwave infrared domain with a precision rating of 0.9798 and mean intersection over union of 0.7386 while sequentially taking the temperature trend of faces with a thermal accuracy of  $\pm 1$  K. While in parallel determining if someone in audible proximity is coughing by using a custom deep convolutional neural network with a precision rating of 0.9519. In addition to presenting FluNet, two datasets have been constructed, one for face detection in the longwave infrared domain consisting of 250 images of 20 participants' faces at various rotations and coverings, including face masks. The other for the real-time detection of cough patterns comprised of a sizeable dataset of 40,482 cough / not cough sounds, coupled with a new lightweight artificial neural network architecture for the classification of cough spectrograms. These findings could be helpful for future low-cost edge computing applications for influenza-like monitoring.

# FluNet: An AI-Enabled Influenza-like Warning System

Ryan J. Ward<sup>1</sup>, Fred P.M. Jjunju<sup>1</sup>, Isa Kabenge<sup>2</sup>, Rhoda Wanyenze<sup>3</sup>, Elias J. Griffith<sup>1</sup>, Noble Banadda<sup>2</sup>, Stephen Taylor<sup>1</sup> and Alan Marshall<sup>1</sup>

<sup>1</sup> Department of Electrical and Electronic Engineering, University of Liverpool, Liverpool, 9 Brownlow Hill, L69 7ZX

<sup>2</sup> Department of Agricultural and Biosystems Engineering Makerere University, P.O. Box 7062, Kampala, Uganda

<sup>3</sup> School of Public Health Makerere University, P.O. Box 7062, Kampala Uganda.

## Abstract

Influenza, an acute viral respiratory disease that is currently causing severe financial and resource strains worldwide. With the recent COVID-19 pandemic exceeding 153 million cases worldwide, there is a need for a low-cost and contactless surveillance system to detect symptomatic individuals, more so in counties with limited healthcare resources. As with many diseases, there are bio-clinical signals relating to the physical symptoms. The main objective of this study was to develop FluNet, a novel, proof-of-concept, low-cost and contactless device for the detection of high-risk individuals. The system passively conducts face detection in the longwave infrared domain with a precision rating of 0.9798 and mean intersection over union of 0.7386 while sequentially taking the temperature trend of faces with a thermal accuracy of  $\pm 1$  K. While in parallel determining if someone in audible proximity is coughing by using a custom deep convolutional neural network with a precision rating of 0.9519. In addition to presenting FluNet, two datasets have been constructed, one for face detection in the longwave infrared domain consisting of 250 images of 20 participants' faces at various rotations and coverings, including face masks. The other for the real-time detection of cough patterns comprised of a sizeable dataset of 40,482 cough / not cough sounds, coupled with a new lightweight artificial neural network architecture for the classification of cough spectrograms. These findings could be helpful for future low-cost edge computing applications for influenza-like monitoring.

## Introduction

Influenza is an acute viral respiratory disease with an estimated 35.5 million cases, 16.5 million medical visits, 490,000 hospitalizations, and a death toll of 34,200 between 2018 - 2019 in the American flu season [1]. With the arrival of the notable coronavirus disease (COVID-19) exceeding 153 million cases worldwide and over 3.2 million confirmed deaths [2], there is a need to limit the spread of the highly infective virus, even more so in countries where health care resources are sparse [3], [4]. Symptoms of COVID-19 typically include; elevated temperatures  $> 37.8^{\circ}\text{C}$ , coughing, shortness of breath, fatigue, muscle and body aches, headaches, loss or altered perception of taste and/or smell, sore throat, nose congestion, nausea or vomiting and diarrhea [5]. Some of the biosignals relating to infected individuals can be detected using sensors, including a thermal camera to detect elevated temperatures and a microphone array to capture cough sounds. The measured biosignals need to be coupled with artificial intelligence (AI) for real-time prediction. AI systems, such as machine learning, deep learning and convolutional neural networks has crucial part to play in the large-scale screening, detection, monitoring and reduction of workload in mitigating COVID-19. However, the use of AI for many bio-clinical applications requires an increased amount of data sharing, especially towards detecting infected individuals and the early detection of pandemics to

enable reliable and precise predictions [6], [7]. The use of AI for the early detection of diseases, be it infectious or otherwise, is essential for preemptive treatment and consequently saving lives [8]. Towards this end, we have rendered two datasets, one for face detection in the LongWave Infrared Domain (LWIR) domain using a low-cost thermal camera and the other for the real-time detection of coughs. Most sensors for patient motoring require direct contact with the subject [9], however due to the high transmissibility of COVID-19, contactless detection would be desirable. Research in using machine vision in the field of thermography-based monitoring is still at an early stage [9].

Recent work by Quer *et al.* indicated that out of 54 COVID-19 positive subjects,  $\approx 42\%$  reported having a fever and  $\approx 64\%$  reporting a cough [10], which means that temperature readings alone are not sufficient for the detection of influenza-like illnesses and that coughing has the highest likelihood of being a symptom. Monto *et al.* findings revealed that the best multivariable predictors for influenza infections were fever and cough with a predictive value of 79% [11]. Therefore, a two-pronged approach was considered to detect influenza-like symptoms, the first being elevated temperature and the second being coughs. Prior work using machine learning and artificial intelligence for the COVID-19 pandemic and focused on predicting the death rate and other abnormalities, rather than identifying high risk individuals [12].

In this work, the signals directly related to the

physical symptoms of influenza-like diseases are detected using our proof-of-concept device “FluNet”. This system may be useful for future contactless vital sign measurements. The device consists of two sensing modules: (i) a thermal camera for automated temperature checks and (ii) a microphone array for cough detection and direction of arrival estimation. Bespoke software was developed using artificial intelligence and deep learning for data fusion, classification and interpretation, including predictive algorithms that use COVID-19 indicators (elevated temperature and coughing) for potential symptomatic COVID-19 cases. We use the term artificial intelligence to refer to the automation of detecting influenza-like symptoms. We use the term deep learning to refer to deep convolutional neural network used for detection cough patterns.

The main contributions of this paper are threefold; (1) a novel proof-of-concept system for the detection of influenza-like symptoms. (2) the presentation of two datasets, one for face detection in the thermal domain and another for real-time cough detection, to aid in developing various clinical applications that enable more reliable and precise predictions and (3) a new lightweight architecture for the classification of cough spectrograms.

## Related Works

The literature has focused its efforts towards the detection and monitoring of the COVID-19 virus in a medical setting (i.e., using deep convolutional neural networks for the classification of X-ray [13] or CT [14] images). Al Hossain *et al.* created a surveillance system for influenza-like illnesses for hospital waiting areas using edge-computing and artificial intelligence. This system can reliably measure crowd density and continuously measure speech and cough sounds in hospital waiting areas for future pandemic predictions [15]. Allam *et al.* reviewed the role AI can have in the early detection of a pandemic. They conclude that using AI for this task requires an increased amount of data sharing to render a sufficiently large dataset for precise predictions [6]. Maghdid *et al.* presented a framework for the detection of COVID-19 using smartphone embedded sensors. They proposed using the phone’s camera to take photos of CT images which is then passed to an artificial neural network, the microphone to measure coughs, the use of the accelerometer to measure fatigue, and the use of the fingerprint reader to measure temperature [16]. Lyra *et al.* used infrared thermography to measure respiratory related chest movements and skin temperature trend measurements [9]. The developed algorithm could extract the respiratory rate with a

mean absolute error of 2.69 beats per minute and estimate the temperature trend with a mean squared error of 1.34 K. See [7], [17], [18] for reviews on using artificial intelligence for COVID-19 detection.

Recent studies regarding cough modeling have explored audio-based cough recognition algorithms. Most recently studies have started exploring the use of different convolutional neural network (CNN) architectures [15], [19]. Cough recognition models have been developed using Spectrogram based features [15], [19], [20] all of which show promising results in terms of accuracy, low false positives, privacy and mobility. Hidden Markov Models coupled with Mel-frequency cepstral coefficient [21], [22] are commonly used to train cough recognition models. Additionally, models have been created using a variety of different acoustic features, such as linear predictive coding with filter banks [23] and gammatone cepstral coefficients [24].

## Materials & Methods

### Experimental Design

This study aims to provide proof-of-concept for an influenza-like warning system to help reduce the spread of influenza-like diseases (COVID-19), to increase the amount of data available for clinical applications and improve upon the reliability and precision of AI in this context. The FluNet framework consists of a low-cost thermal imaging camera, a high-quality camera, a microphone array, and an Intel® Neural Compute Stick 2, all situated in a custom 3D printed case. Fig. 1 shows a photo of the FluNet system and the conceptualization of the software components. Figure S1 shows the schematic setup of the hardware components. The Raspberry Pi and the thermal imaging pipeline are programmed using C++. The image processing was done using the OpenCV [25] library for C++. The ReSpeaker was controlled using a custom Python 3.7 script. The following list details the hardware used in the FluNet system.

- Seek Thermal Compact Pro [26]: A thermal camera with a 320 x 240 resolution, a 32° field of view, 1,800 ft detection distance and has a focusable lens.
- ReSpeaker Microphone Array v2.0 [27]: A high-performance chipset with an array of 4 microphones.
- Raspberry Pi High Quality Camera [28]: A 12.3 megapixel camera with an adjustable back focus and support for a C- and CS-mount lenses.
- Raspberry Pi High Quality Camera Lens [29]: A 6mm, 3 megapixel wide-angle camera lens with a 63° field of view.

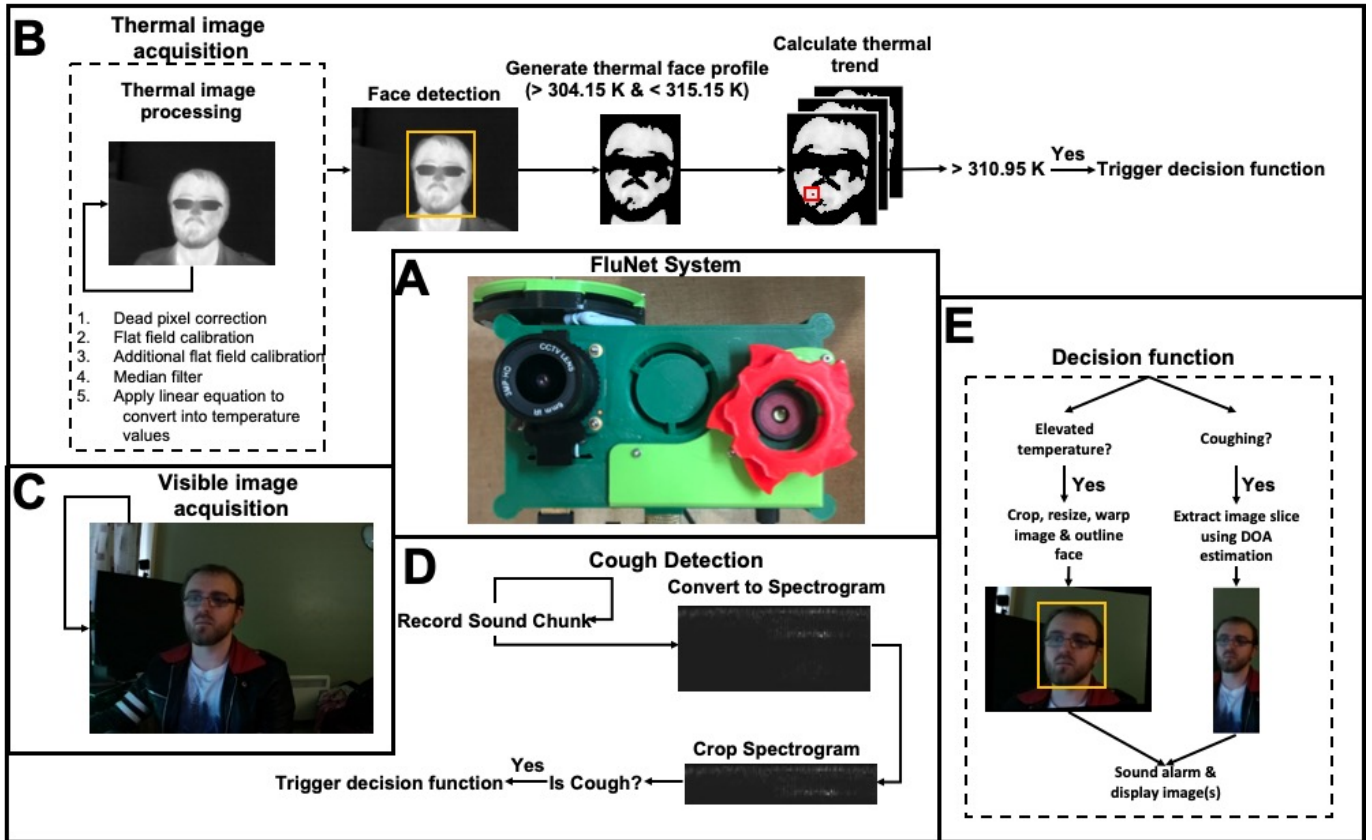


Fig. 1. (A) Photo of the FluNet System. (B - E) Conceptualisation of the software components. (B) Automated temperature checks pipeline. (C) Visible image acquisition. (D) Cough detection pipeline and (E) decision function pipeline triggered if the is a cough and/or elevated temperature is detected.

- Intel® Neural Compute Stick 2 [30]: A compact deep learning processing module for inference on the edge.
- Raspberry Pi 4 [31]: A control unit with a small form factor to synchronize the above devices and sensors.

## Temperature & Audio Processing

A Seek Thermal Compact Pro [26] was used to collect information from the LWIR domain. This information is passed to the Raspberry Pi for real-time processing consisting of face detection and temperature checks. To ensure the thermal camera was reporting as close to the correct temperature as possible, dead pixel correction, flat field calibration, additional flat field calibration, a 5x5 median filter, and a linear equation were conducted to translate the values produced by the thermal camera into temperature values. The flat field calibration consists of a thirty second calibration period when the software is launched. The thermal cameras shutter was manually closed during this process and consisted of taking an averaged frame used as an offset when processing the thermal images. This was done to compensate for the temperature variation on the lens of the thermal camera. The additional flat field calibration consists of taking a frame when the thermal cameras shutter is closed to compensate for temperature changes over time during

operation. To reduce the influence of thermal drift, the room temperature was estimated by taking the median value from a 5x5 region of interest (ROI) in the bottom right corner. The background temperature was used to perform the emissivity correction and was only applied after calibration but before any temperature values were reported. A ReSpeaker Microphone Array v2.0 [27] was used to collect audio data. The recordings are one second in length and are converted to spectrograms before being transmitted sent to the Neural Compute Stick for classification. Additionally, the spectrograms were supplied to the deep convolutional neural network for training, testing and classification. The convolutional neural network was designed, trained and tested in MATLAB R2018b and ported over to OpenVINO for inference on the Neural Compute Stick.

## Influenza-like Detection

To determine if subjects showed symptoms of influenza, a two-pronged approach was employed. In parallel, the system automatically checks the temperature of people passing and checks if anyone in audible proximity of the microphone array is coughing. The temperature checks are done by conducting face detection in the thermal domain; from this ROI, a thermal facial profile was constructed where any pixel values within range of 304.15 K and 315.15 K were kept. Fifteen of these images were stored in a buffer

and used for the temperature trend calculation. To reduce the influence of thermal drift, a 5x5 ROI from the bottom right hand corner was used to estimate background temperature. To ensure more reliable temperature readings, temperature trend estimation was considered; this consisted of taking the mean of the maximum temperatures over 1-second intervals. The cough checks were done by using a deep convolutional neural network running on the Intel® Neural Compute Stick 2. Successive one second sound chunks were recorded. The sound chunks were then converted into spectrograms and passed to the Intel® Neural Compute Stick 2 for classification. If an elevated temperature or coughing were detected, then in that case, an alarm is sounded, and optionally, a photo of the person's face with the elevated temperature is shown, and/or a 15° slice was taken from the visible image, centered at the direction of arrival of the detected cough.

### Multi-Camera Calibration

An interactive graphical user interface was created using C++ and OpenCV to align the infrared camera with the visible light camera. This interface allowed for the real-time visualization of the aligned image with the thermal image superimposed onto it, allowing us to visually determine the transformation that best aligned the two images. To create the homography matrix, two photos were taken simultaneously using the thermal and visible light cameras; these photos were of a metal box containing a complex pattern. As the two cameras have different fields of view, the visible light image was cropped to remove the features of the image that was not in the thermal image and the final image was resized to be 320 x 240 (the same size as the thermal image). Next, both photos were converted to greyscale, and their Oriented FAST and rotated BRIEF (ORB) features [32] were calculated. ORB was chosen as it was fast, efficient, resistant to image noise, rotation invariant and multi-scale. The number and quality of features were interactable via the interface. These parameters were adjusted to determine the homography matrix that aligned these two images the best; this matrix was then saved and used as input into an affine transformation warping the visible light image to align with the thermal image by translation, rotation and scaling. This was pre-computed opposed to real-time to maximize the efficiency, consistency and accuracy.

### Direction of Arrival Camera Calibration

To align the direction of arrival obtained from the microphone array with the video feed from the visible light camera, we first convert a musical excerpt into stereo format. This was done to make the sound coming from the speakers from the mobile phone

identical and thereby, reducing the influence of different sound and/or volume coming from either speaker of the mobile phone. This excerpt was then played from the mobile phone mounted on a tripod and was visually centered to the video feed from the visible light camera. We played this musical excerpt five times and recorded the direction of arrival. We manually adjusted the rotation of the microphone array until the direction of arrival averaged to be 90° (directly in front of the system).

## Results & Discussion

### Thermal Processing

To determine how well FluNet could measure temperature,  $\approx 500$  ml of water was boiled, placed in a bowl 1m vertically below the thermal camera; we took 28 photos of the water as it cooled down and took the absolute temperature of the water using a Salter instant-read digital thermometer with a precision of 0.1 K. A linear equation was then applied to the temperature of the water in Kelvin and the corresponding value received from the thermal camera, as shown in, Fig. 2, this resulted in an  $R^2 = 0.99$ . Emissivity correction was applied before any temperature values were used to correct for radiation and produce more reliable temperature values. Subtracting the absolute temperature from the reported temperature values revealed a thermal accuracy of  $\pm 1.5$  K.

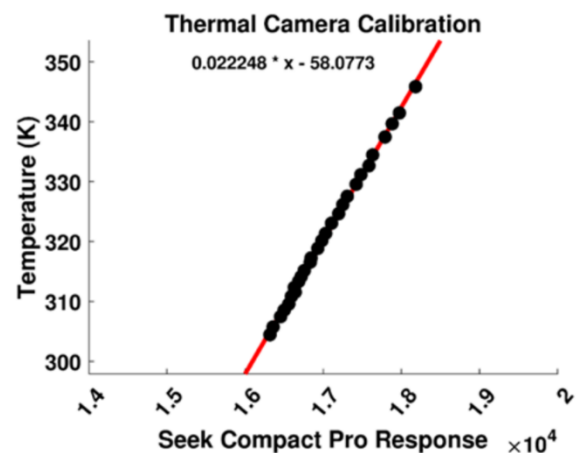


Fig. 2. Linear equation fitted to the temperature in Kelvin and values produced by the thermal camera after preprocessing.

These findings show that the system can measure the temperature of objects with an emissivity close to that of human skin, however, with a margin of error of  $\pm 1.5$  K.

### Temperature Trend Estimation

In order to decrease the error of the temperature measurements a time series analysis was considered. The recordings were conducted in the same manner as reported in the section above, fifteen frames were

logged ( $\approx$  one second of frames) for eleven different recordings resulting in one hundred and ten frames of data used for this analysis. First the maximum temperature values from a ROI were taken, along with the median value from a 5x5 ROI in the bottom right hand corner for background temperature estimation. The resulting vectors were then smoothed using a three-point moving average filter. Using the mean value from both of these arrays coupled the linear equation uncovered in the Thermal Processing section these values were then converted to temperature values. Emissivity correction using the estimated background temperature was then applied to the resulting temperature value. Subtracting the absolute temperature from the reported temperature values revealed that the time series temperature estimation provided a better result increasing the thermal accuracy from  $\pm 1.5$  K to  $\pm 1$  K and therefore demonstrating the stability of the temperature readings, as shown in Fig. 3.

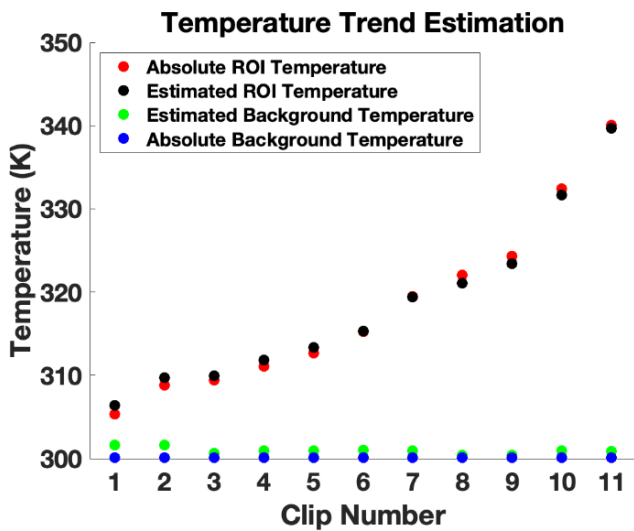


Fig. 3. Temperature trend estimation showing the estimated temperature (black and green markers) and their absolute temperature values (red and blue markers) for both background temperature and a specified ROI temperature.

## Face Detection

### Data Sources

The face detection is conducted in the LWIR (Longwave Infrared) domain. A dataset was constructed consisting of two parts (positive instances and negative instances). The positive instances comprised of 250 images of 20 participants at five different approximate facial rotations ( $-90^\circ$ ,  $-45^\circ$ ,  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ), with varying face coverings (face masks, glasses, no coverings) and looking downwards (see Fig. 4). We opted to take these photos to allow the face detection algorithm to detect people who are wearing face masks. Additionally, we merged the data we had collected with 976 images from TUFTS thermal face dataset [33].

### Data Argumentation

Data argumentation was then conducted to ensure the algorithm could still detect faces when there are hot objects (i.e., hot coffee) being detected by the thermal camera. This consisted of relative temperature changes to the thermal images (0.25, 0.5, .75 and 1); this made a dataset consisting of 4154 images for the positive instances. All positive instances consist of a manually drawn ROI around the face of the subject to indicate the part of the image to use in training. The negative instances comprised of 1,156 new photos taken with the thermal camera, none of which included a face. Additionally, we took a maximum of 10 random patches from each positive instance image using a minimum of 50 x 50 pixels and only from outside of the ROI as not to include any faces in the negative instances, creating 31,070 images; the final negative instances set contains 32,226 images.

### Modelling

The algorithm chosen for face detection is Viola-Jones [34] using local binary patterns [35]. This particular method was selected as it has been shown to work well in the thermal domain [36] for computational efficiency and accuracy, allowing for accurate and real-time face detection on the edge. However, YOLOv4-Tiny algorithm also performs well in the thermal domain [9] but we opted for Viola-Jones over this as

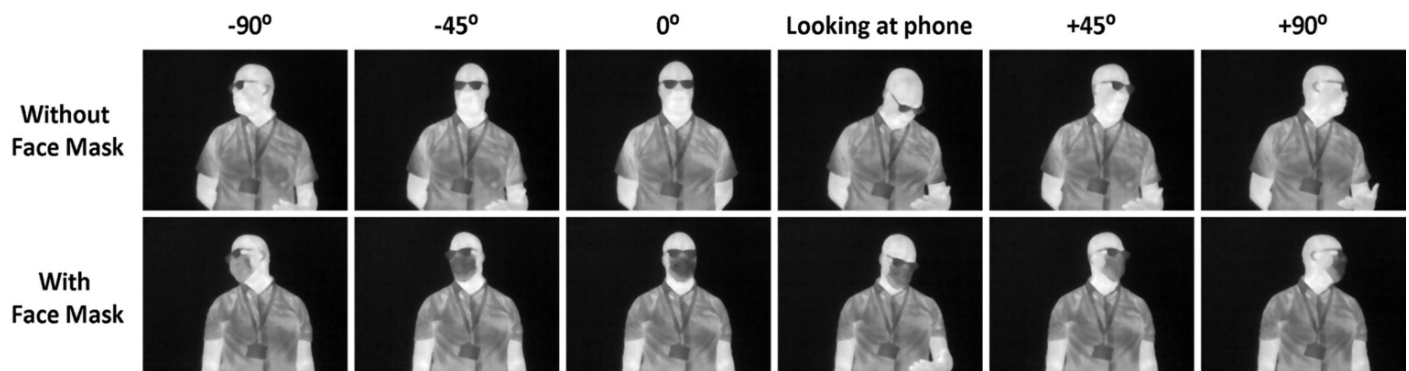


Fig. 4. Sample photos collected to train the face detector consistent of 5 facial rotations and looking downwards to a phone.



the YOLO algorithm would need to be delegated to the Neural Compute Stick and we wanted the inference for the cough detection to be done as soon as possible after the recording.

### Testing

To test the accuracy of the face detection algorithm, we took 204 new images; 97 of these images included faces with face coverings and different facial rotations using people unseen during the training stage, we also took 107 background images. We then supplied these images to the classifier with manually labelled regions of interest for where the faces are located within the photos. If the detected face was not outside the ROI, the face inference was labelled as being correct. Whereas, if any faces were present in the background images, it was labelled as being incorrect. The resulting confusion matrix is shown in Fig. 4. The classifier archived a 95.59% accuracy rating, a precision of 0.9798, a recall of 0.9065 and an F score of 0.9565.

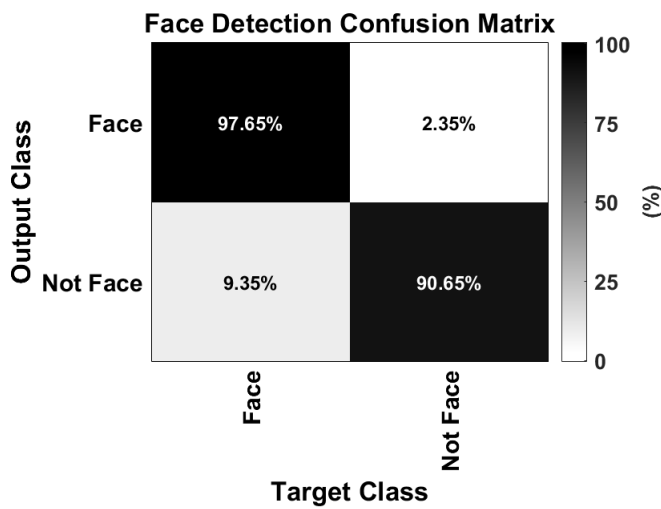


Fig. 5. Confusion matrix showing the percentages of class wise correct vs. incorrect prediction for the face detector.

Next to determine how well the face detection was detecting the faces the mean intersection over union (bounding box overlap ratio of the area of intersection) of all “Face” test instances was calculated. All the faces in the “Face” set had a manually defined ROI this was then compared to the ROI determined by the face detection algorithm this revealed a mean intersection over union of 0.7386.

## Cough Detection

### Data Sources

A series of clips from YouTube™ containing people coughing was downloaded (see supplementary material). The video element of these clips was removed to only contain audio. Any clips that had a sample rate that was not 44100 Hz was then discarded. Each clip was linearly split into  $N$  one-second clips; a dataset was then constructed consisting of the clips that contained coughs, manually labelled by a human.

Clips that did not include a cough were discarded. This resulted in a dataset consisting of 1,557 cough samples. We also collected a “not cough” dataset consisting of background noise, office, music, airport, coffee shop, harbor, nightclub, and simulated turbulence sounds. Again, these were trimmed to be 1 second in length and at a sample rate of 44100 Hz. A total of 23,890 background noise samples were collected.

### Data Argumentation

Volume, background noise, turbulence and time argumentation were performed. Data argumentation was conducted to ensure that the model would recognize coughs in different environments and at different distances. Manually recorded noises from a highly windy environment were performed to simulate the turbulence, with no wind but rapid movements of the microphone array and both highly windy and rapid movements. These were then added to the background noise samples. To simulate different distances away from the microphone array and varying background noises, we merged a random background noise into the cough samples at a random ratio (0.25, 0.5, 0.75, 1, 2 and 4) where, 0.25 simulates a quiet cough with loud background noise. To simulate different coughs at different times, the training images were also flipped the audio clips on the x-axis. After data argumentation, the final dataset consisted of 40,482 cough samples at a random volume with random background noise and 23,890 “Not Cough” samples.

### Modelling

A custom two-layer convolutional neural network was developed to classify cough sounds; the architecture is shown in Fig. 6. We originally started with the convolutional neural network architecture for a sparse resource environment [37]. We then modified this neural network to optimize for classification of our spectrograms. The dataset was then converted into spectrograms with a window shift and window size of 10ms and 30ms, respectively, and contained frequencies ranging from 0 to 4000 Hz resulting in an image size of 121 x 275 x 1. These particular parameters were chosen as they provided the best classification accuracy for coughs in [15]. The spectrograms are consequently fed into the convolutional neural network for classification. Additionally, we only kept the first 50 rows of each spectrogram because, based on visual inspection, they didn’t contain any information relating to the cough sounds resulting in a final image size of 50 x 275 x 1. Details to replicate the artificial neural network can be found in Table S1. The training progress of the artificial neural network is shown in Fig. S2.

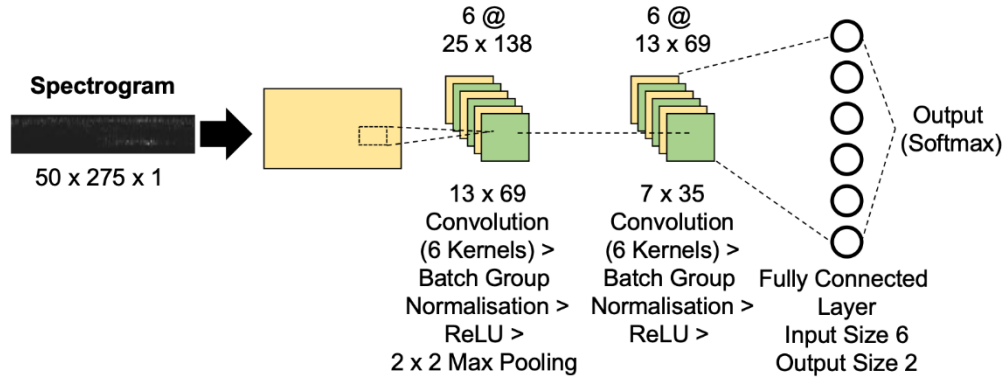


Fig. 6. Graphical illustration of the neural network created and used for cough detection.

## Testing

To test our convolutional neural network, we collected an additional 374, 1-second cough samples with random volume and background noise augmentation (train station ambience or people talking). These new samples were entirely unseen in the training stage neural network. We also collected 600, 1-second background noise samples consisting of music, train station ambience and simulated turbulence. For a total of 974 samples for testing, we classified the testing data using the trained neural network after converting the recordings to spectrograms. The resulting confusion matrix is shown in Fig. 7A and has an accuracy rating of 92.42%, a precision of 0.9519, a recall of 0.9233, an F score of 0.9374 and an AUC of

0.9806 (Fig. 7B). The audio recordings used to test the classifier were not included in the training set.

## Direction of Arrival

We used the direction of arrival provided by the ReSpeaker Microphone Array v2.0 as it is capable of producing accurate and real-time direction of arrival estimates [15] and offloaded some of the required computations off the Raspberry Pi. To test accuracy of the direction of arrival algorithm a musical excerpt was converted to be mono instead of stereo. This was done so the same sound was coming from both of the speakers on the mobile phone. Two VIVE trackers [38] were used to track the actual angle between the microphone array and the mobile phone; one was placed directly over the microphone array, the other was attached to a mobile phone. The sound clip was played from a mobile phone whilst being moved around the environment. The direction of arrival from the microphone array was recorded along with the angle between the two VIVE trackers. As the direction of arrival is provided continuously, only estimates that were not the same as the last estimate was logged. Next, we compared the set of angles (direction of arrival vs computed angle). This revealed an error of  $\pm 4.78$  from 80 different angles ranging from  $\approx 90^\circ$  to  $-90^\circ$  from the front center of the microphone array (see Fig. 8). The direction of arrival was used to extract a  $15^\circ$  slice from the Raspberry Pi High-Quality Camera video feed from  $\pm 7.5^\circ$  from where the cough was detected, as the potential error was only  $\pm 4.78^\circ$  we added an additional  $2.72^\circ$  either side to make sure the full face could be captured while compensating for the maximum potential error at a distance of 1 meter. To determine the location of the two VIVE trackers, Unity 2018.03 was used to trace and log the angle between the two trackers along with the DOA estimation from the microphone array.

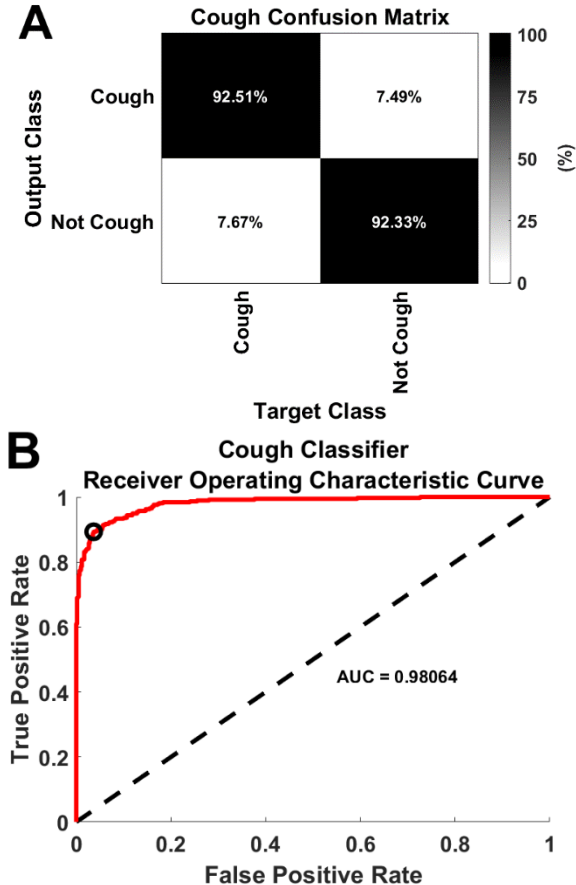


Fig. 7. (A) Confusion matrix showing the percentages of class wise correct vs. incorrect prediction for the cough detector. (B) ROC curve for the cough classifier the black circular marker denotes the operating point.



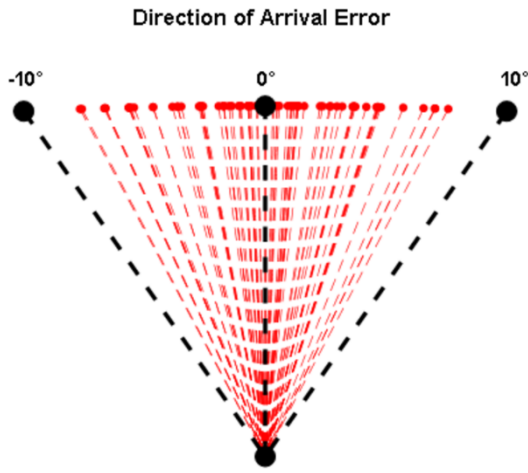


Fig. 8. Red lines indicate error from expected angle. Centre black line is the perfect angle (no error). Left and right black lines mark  $-10^\circ$  and  $10^\circ$  from the microphone array (no error).

## Conclusion & Future Work

A novel, proof-of-concept influenza-like warning system has been developed, consisting of automated temperature checks and cough detection. The device triggers an audio alarm under three conditions. (1) a ROI deemed to be the face is between a certain temperature threshold. (2) subsequent coughing is detected, and (3) both (1) and (2) have been detected. Each one of the three conditions produces a unique audible alarm. We have gathered two datasets one to train and test the face detector consisting of 250 images from 20 participants at a variety of different angles ( $-90^\circ$ ,  $-45^\circ$ ,  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ) and facial coverings, including face masks. The developed classifier has a precision and recall of 0.9798 and 0.9065 respectively and a mean intersection over union of 0.7386. The other dataset was used to train and test a custom deep artificial neural network to determine if someone in audible proximity was coughing. This dataset consisted of a sizable 1,557 cough samples and 23,890 background noise samples. Our architecture uses the frequency of audio signals in the form of a spectrogram as input and has a precision and recall of 0.9667 and 0.9183, respectively. The quality and potential of these datasets have been exemplified in the development of FluNet. Our findings offer the potential to enhance face detection in the thermal domain and real-time cough detection through an increased amount of data and will prove useful for future low-cost edge computing applications for influenza-like monitoring. Real-time monitoring is a passive form of surveillance that does not require direct human contact, which shows potential to continuously monitor potential symptomatic individuals. It is worth mentioning that not all infected individuals will exhibit symptoms, be it temperature, coughing or otherwise.

Other physiological factors, such as respiratory rate [39], [40], could be taken into consideration. Moreover, the device could be converted into a

wearable sensory argumentation system, allowing for the detection of influenza-like symptoms on the go. Although we have shown that the system's individual components perform well, user trials will need to be conducted to determine how well the system works together. These tests will need to be conducted on individuals with and without a form of influenza.

## Datasets

The datasets collected for this work have been made freely available at [41].

## References

- [1] "Estimated Influenza Illnesses, Medical visits, Hospitalizations, and Deaths in the United States — 2018–2019 influenza season | CDC." [Online]. Available: <https://www.cdc.gov/flu/about/burden/2018-2019.html>. [Accessed: 30-Aug-2020].
- [2] WHO, "Coronavirus disease (COVID-19)." [Online]. Available: <https://www.who.int/emergencies/diseases/new-covid-19>. [Accessed: 30-Aug-2020].
- [3] K. Lumpur, "Informal consultation on influenza pandemic preparedness in countries with limited resources."
- [4] H. Oshitani, T. Kamigaki, and A. Suzuki, "Major issues and challenges of influenza pandemic preparedness in developing countries," *Emerging Infectious Diseases*, vol. 14, no. 6. Centers for Disease Control and Prevention (CDC), pp. 875–880, 2008, doi: 10.3201/eid1406.070839.
- [5] "Symptoms of Coronavirus | CDC." [Online]. Available: <https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html>. [Accessed: 30-Aug-2020].
- [6] Z. Allam, G. Dey, and D. S. Jones, "Artificial Intelligence (AI) Provided Early Detection of the Coronavirus (COVID-19) in China and Will Influence Future Urban Health Policy Internationally," *Ai*, vol. 1, no. 2, pp. 156–165, 2020, doi: 10.3390/ai1020009.
- [7] A. Alimadadi, S. Aryal, I. Manandhar, P. B. Munroe, B. Joe, and X. Cheng, "Artificial intelligence and machine learning to fight covid-19," *Physiol. Genomics*, vol. 52, no. 4, pp. 200–202, 2020, doi: 10.1152/physiolgenomics.00029.2020.
- [8] S. Lalmuanawma, J. Hussain, and L. Chhakchhuak, "Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review," no. January, 2020.

- [9] S. Lyra *et al.*, "A deep learning-based camera approach for vital sign monitoring using thermography images for ICU patients," *Sensors*, vol. 21, no. 4, pp. 1–18, 2021, doi: 10.3390/s21041495.
- [10] G. Quer *et al.*, "Wearable sensor data and self-reported symptoms for COVID-19 detection," *Nat. Med.*, 2020, doi: 10.1038/s41591-020-1123-x.
- [11] A. S. Monto, S. Gravenstein, M. Elliott, M. Colopy, and J. Schweinle, "Clinical signs and symptoms predicting influenza infection," *Arch. Intern. Med.*, vol. 160, no. 21, pp. 3243–3247, 2000, doi: 10.1001/archinte.160.21.3243.
- [12] S. Kushwaha *et al.*, "Significant applications of machine learning for covid-19 pandemic," *J. Ind. Integr. Manag.*, vol. 5, no. 4, pp. 453–479, Dec. 2020, doi: 10.1142/S2424862220500268.
- [13] F. M. Salman, S. S. Abu-Naser, E. Alajrami, B. S. Abu-Nasser, and B. A. M. Ashqar, "COVID-19 Detection using Artificial Intelligence," *Int. J. Acad. Eng. Res.*, vol. 4, no. 3, pp. 18–25, 2020.
- [14] O. Gozes, M. Frid, H. Greenspan, and D. Patrick, "Rapid AI Development Cycle for the Coronavirus ( COVID-19 ) Pandemic : Initial Results for Automated Detection & Patient Monitoring using Deep Learning CT Image Analysis," *arXiv:2003.05037*, 2020.
- [15] F. Al Hossain, A. A. Lover, G. A. Corey, N. G. Reich, and T. Rahman, "FluSense: A Contactless Syndromic Surveillance Platform for Influenza-Like Illness in Hospital Waiting Areas," *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 4, no. 1, pp. 1–28, 2020, doi: 10.1145/3381014.
- [16] H. S. Maghdid, K. Z. Ghafoor, A. S. Sadiq, K. Curran, and K. Rabie, "A Novel AI-enabled Framework to Diagnose Coronavirus COVID 19 using Smartphone Embedded Sensors: Design Study," pp. 1–7, 2020.
- [17] F. Shi *et al.*, "Review of Artificial Intelligence Techniques in Imaging Data Acquisition, Segmentation and Diagnosis for COVID-19," *IEEE Rev. Biomed. Eng.*, pp. 1–13, 2020, doi: 10.1109/RBME.2020.2987975.
- [18] R. Vaishya, M. Javaid, I. Haleem Khan, and A. Haleem, "Artificial Intelligence (AI) applications for COVID-19 pandemic," *Diabetes Metab Syndr.*, vol. 14(4), no. Jul-Aug, pp. 337–339, 2020.
- [19] J. Amoh and K. Odame, "Deep Neural Networks for Identifying Cough Sounds," *IEEE Trans. Biomed. circuits systms*, vol. 10, no. 5, pp. 1003–1011, 2016.
- [20] E. C. Larson, T. Lee, S. Liu, M. Rosenfeld, and S. N. Patel, "Accurate and privacy preserving cough sensing using a low-cost microphone," *UbiComp'11 - Proc. 2011 ACM Conf. Ubiquitous Comput.*, no. 3, pp. 375–384, 2011, doi: 10.1145/2030112.2030163.
- [21] S. Matos, S. S. Birring, I. D. Pavord, and D. H. Evans, "Detection of Cough Signals in Continuous Audio Recordings Using Hidden Markov Models," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 6, pp. 1078–1083, 2006.
- [22] C. Zhu, L. Tian, X. Li, H. Mo, and Z. Zheng, "Recognition of cough using features improved by sub-band energy transformation," *Proc. 2013 6th Int. Conf. Biomed. Eng. Informatics, BMEI 2013*, vol. 1, no. Bmei, pp. 251–255, 2013, doi: 10.1109/BMEI.2013.6746943.
- [23] S. J. Barry, A. D. Dane, A. H. Morice, and A. D. Walmsley, "The automatic recognition and counting of cough," *Cough*, vol. 2, p. 8, 2006, doi: 10.1186/1745-9974-2-8.
- [24] J. Liu *et al.*, "Cough signal recognition with gammatone cepstral coefficients," in *China Summit & International Conference on Signal and Information Processing (ChinaSIP)*, 2013, pp. 160–164.
- [25] G. Bradski, "The OpenCV Library," *Dr. Dobb's J. Softw. Tools*, 2000.
- [26] "Thermal Cameras for your Smartphone - Seek Thermal | Affordable Infrared Thermal Imaging Cameras." [Online]. Available: <https://www.thermal.com/compact-series.html/>. [Accessed: 31-Aug-2020].
- [27] "ReSpeaker Mic Array v2.0 - Seeed Wiki." [Online]. Available: [https://wiki.seeedstudio.com/ReSpeaker\\_Mic\\_Array\\_v2.0/](https://wiki.seeedstudio.com/ReSpeaker_Mic_Array_v2.0/). [Accessed: 31-Aug-2020].
- [28] "Buy a Raspberry Pi High Quality Camera – Raspberry Pi." [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-high-quality-camera/>. [Accessed: 18-Mar-2021].
- [29] "Raspberry Pi High Quality Camera Lens– The Pi Hut." [Online]. Available: <https://thepihut.com/products/raspberry-pi-high-quality-camera-lens?variant=31811254190142>. [Accessed: 18-Mar-2021].
- [30] "Intel® Neural Compute Stick 2." [Online]. Available: <https://software.intel.com/content/www/us/en/develop/hardware/neural-compute-stick.html>. [Accessed: 31-Aug-2020].
- [31] "Raspberry Pi." [Online]. Available: <https://www.sonymencoed.co.uk/case-studies/raspberry-pi/>. [Accessed: 31-Aug-2020].
- [32] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT

- or SURF," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2564–2571, 2011, doi: 10.1109/ICCV.2011.6126544.
- [33] K. Panetta *et al.*, "A Comprehensive Database for Benchmarking Imaging Systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 509–520, 2020, doi: 10.1109/TPAMI.2018.2884458.
- [34] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2001, doi: 10.1109/cvpr.2001.990517.
- [35] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002, doi: 10.1109/TPAMI.2002.1017623.
- [36] K. Reese, Y. Zheng, and A. Elmaghraby, "A Comparison of Face Detection Algorithms in the Visible and Thermal Spectrums," *Int'l Conf. Adv. Comput. Sci. Appl.*, 2012.
- [37] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2015-Janua, pp. 1478–1482, 2015.
- [38] "VIVE Tracker (3.0) | VIVE United Kingdom." [Online]. Available: <https://www.vive.com/uk/accessory/tracker3/>. [Accessed: 18-Mar-2021].
- [39] C. E. Goldfine, M. F. T. Oshim, S. P. Carreiro, B. P. Chapman, D. Ganesan, and T. Rahman, "Respiratory rate monitoring in clinical environments with a contactless ultra-wideband impulse radar-based sensor system," in *Proceedings of the... Annual Hawaii International Conference on System Sciences. Annual Hawaii International Conference on System Sciences*, 2020, vol. 2020, p. 3366.
- [40] Y. Cho, S. J. Julier, N. Marquardt, and N. Bianchi-Berthouze<sup>1</sup>, "Robust tracking of respiratory rate in high- dynamic range scenes using mobile thermal imaging," *arXiv*, vol. 8, no. 10, pp. 1565–1588, 2017, doi: 10.1364/boe.8.004480.
- [41] R. J. Ward *et al.*, "COVID-19 Thermal Face & Cough Dataset." Zenodo, May-2021, doi: 10.5281/zenodo.4739682.

# Supplementary Material

## FluNet: An AI-Enabled Influenza-like Warning System

Ryan J. Ward<sup>1</sup>, Fred P.M. Jjunju<sup>1,\*</sup>, Isa Kabenge<sup>2</sup>, Rhoda Wanyenze<sup>3</sup>, Elias J. Griffith<sup>1</sup>, Noble Banadda<sup>2</sup>, Stephen Taylor<sup>1</sup> and Alan Marshall<sup>1</sup>

<sup>1</sup>University of Liverpool, Department of Electrical Engineering & Electronics, Liverpool, L69 3GJ, United Kingdom.

<sup>2</sup>Makerere University, Department of Agricultural and Biosystems Engineering, Kampala, P.O. Box 7062, Uganda.

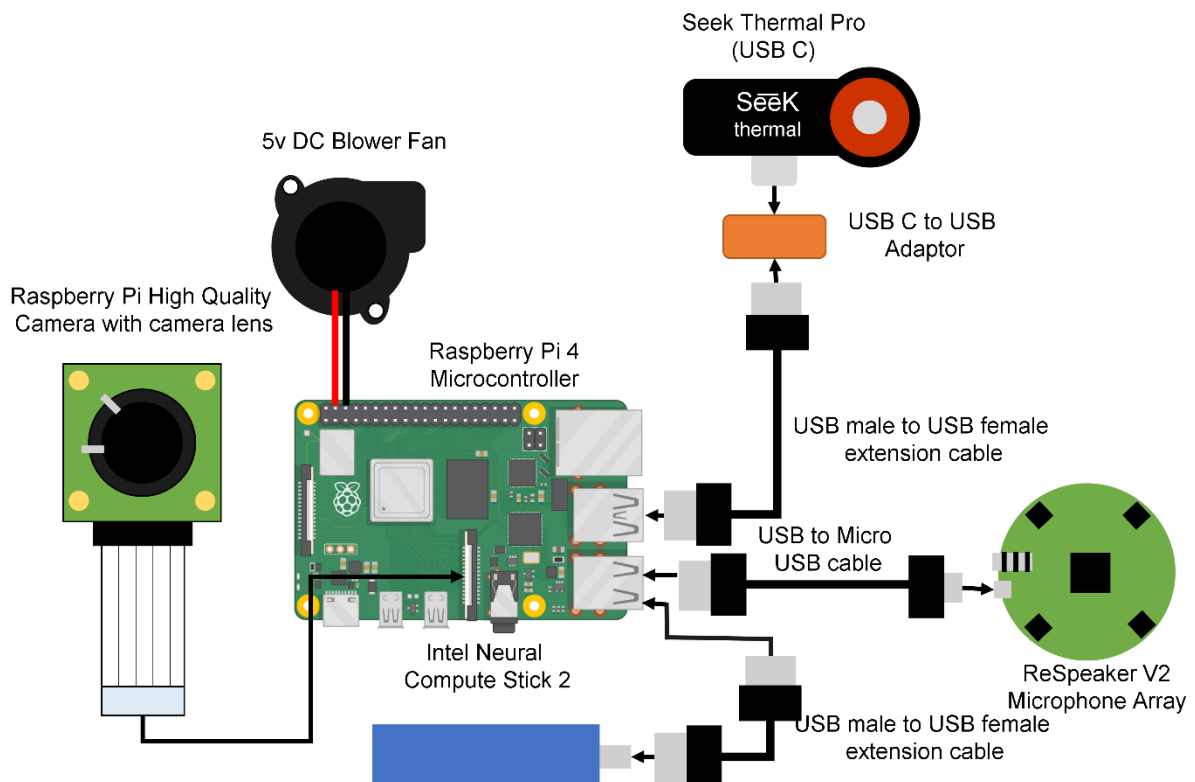
<sup>3</sup> Makerere University, School of Public Health, Kampala, P.O. Box 7062, Uganda.

\*[fjjunju@liverpool.ac.uk](mailto:fjjunju@liverpool.ac.uk)

## Table of contents

Section	Page number
Methods (Experimental Design)	2
Face Detection	5
Cough Detection (Data Sources)	6
Cough Detection (Modelling)	6-7

## Methods (Experimental Design)



**Fig S1.** Hardware setup for the FluNet system.

## Face Detection

Link to TUFTS thermal face dataset; <https://www.kaggle.com/kpvisionlab/tufts-face-database-thermal-td-ir>

This dataset was used in conjunction with our collected data to train the face detector.

## Cough Detection (Data Sources)

The list below indicates the YouTube clips where the cough sounds were taken from.

Clips used for training & validation

<https://www.youtube.com/watch?v=jxYNLCYTzwZQ>  
<https://www.youtube.com/watch?v=qfpJg179YNk>  
<https://www.youtube.com/watch?v=d2wkdrScerU>  
<https://www.youtube.com/watch?v=ct3tHDfNKiQ>  
[https://www.youtube.com/watch?v=Dc\\_aoUCqw2E](https://www.youtube.com/watch?v=Dc_aoUCqw2E)  
<https://www.youtube.com/watch?v=tvHytTIGs0M>  
<https://www.youtube.com/watch?v=0QQxKN-KC1U>  
<https://www.youtube.com/watch?v=DYfjPnty2Ho>  
<https://www.youtube.com/watch?v=q6WsoL3J8U8>  
<https://www.youtube.com/watch?v=CTSLdNxN1cc>  
<https://www.youtube.com/watch?v=tfc5cXiXMDc>  
[https://www.youtube.com/watch?v=rkF\\_uMizqoc](https://www.youtube.com/watch?v=rkF_uMizqoc)



<https://www.youtube.com/watch?v=5905FxxZ9dl>  
<https://www.youtube.com/watch?v=lzPMblll3LE>  
<https://www.youtube.com/watch?v=h2FLCKMcEX0>  
<https://www.youtube.com/watch?v=2Mw-s5jngXU>  
<https://www.youtube.com/watch?v=diuuEXKzNB8>  
<https://www.youtube.com/watch?v=dg-l9j76-t8>  
<https://www.youtube.com/watch?v=TK4CveeCWfY>  
<https://www.youtube.com/watch?v=4k0ziD0j5BI>  
<https://www.youtube.com/watch?v=CsDXlt7Ei1c>  
[https://www.youtube.com/watch?v=7Ez5Wc\\_esBg](https://www.youtube.com/watch?v=7Ez5Wc_esBg)  
<https://www.youtube.com/watch?v=NaOVmYoljbs>  
<https://www.youtube.com/watch?v=XrpB4DTNQZw>  
<https://www.youtube.com/watch?v=h-GtQfDCoaE>  
<https://www.youtube.com/watch?v=u2KMBD5-oCg>  
<https://www.youtube.com/watch?v=A5s2ZgwQ1VM>  
<https://www.youtube.com/watch?v=ekqLlw-Xe68>  
<https://www.youtube.com/watch?v=6LK6yHtlung>  
<https://www.youtube.com/watch?v=tZtJaS2ZtME>  
<https://www.youtube.com/watch?v=AQOeIVbhFm4>  
<https://www.youtube.com/watch?v=zjd4HrJbc8o>  
<https://www.youtube.com/watch?v=1UDFq2InljM>  
<https://www.youtube.com/watch?v=LkxvBb2VXbs>  
[https://www.youtube.com/watch?v=i\\_hl3gVoRr0](https://www.youtube.com/watch?v=i_hl3gVoRr0)  
[https://www.youtube.com/watch?v=3id3zRRZBVM&list=PLKUX6UbAKeUC5\\_4rOY0W1Q6RjyYLHoWU](https://www.youtube.com/watch?v=3id3zRRZBVM&list=PLKUX6UbAKeUC5_4rOY0W1Q6RjyYLHoWU)  
[https://www.youtube.com/watch?v=3gFShyvw8iU&list=PLKUX6UbAKeUC5\\_4rOY0W1Q6RjyYLHoWU&index=2](https://www.youtube.com/watch?v=3gFShyvw8iU&list=PLKUX6UbAKeUC5_4rOY0W1Q6RjyYLHoWU&index=2)  
[https://www.youtube.com/watch?v=lH6ym1UrInE&list=PLKUX6UbAKeUC5\\_4rOY0W1Q6RjyYLHoWU&index=3](https://www.youtube.com/watch?v=lH6ym1UrInE&list=PLKUX6UbAKeUC5_4rOY0W1Q6RjyYLHoWU&index=3)  
[https://www.youtube.com/watch?v=4WyFn2L44m0&list=PLKUX6UbAKeUC5\\_4rOY0W1Q6RjyYLHoWU&index=5](https://www.youtube.com/watch?v=4WyFn2L44m0&list=PLKUX6UbAKeUC5_4rOY0W1Q6RjyYLHoWU&index=5)  
[https://www.youtube.com/watch?v=GkWQbUPBDry&list=PLKUX6UbAKeUC5\\_4rOY0W1Q6RjyYLHoWU&index=7](https://www.youtube.com/watch?v=GkWQbUPBDry&list=PLKUX6UbAKeUC5_4rOY0W1Q6RjyYLHoWU&index=7)

Clips used for background sounds

<https://www.youtube.com/watch?v=fuwGT88P-RU>  
<https://www.youtube.com/watch?v=BOdLmxy06H0>  
<https://www.youtube.com/watch?v=8trgESN1Eks>  
<https://www.youtube.com/watch?v=Hx3vzG8Uahk>  
<https://www.youtube.com/watch?v=Hym8S0CqtW0>  
<https://www.youtube.com/watch?v=sJU14CH4f-c>  
<https://www.youtube.com/watch?v=ZSrVznkaMEM>  
<https://www.youtube.com/watch?v=Uj0DuD92Tik>  
<https://www.youtube.com/watch?v=ttgBegSSyTs>

Clips used for the background merged into cough sounds

<https://www.youtube.com/watch?v=IKB3Qiglyro>  
<https://www.youtube.com/watch?v=dZHnol-qDyA>

The list below indicates the YouTube clips used to test the artificial neural network;

Background sounds merged into the cough sounds

<https://www.youtube.com/watch?v=ZhQx8z5BCzY&t=21s>

<https://www.youtube.com/watch?v=w1yTt5wkUwI>

<https://www.youtube.com/watch?v=ZD47Mhzm-CU>

<https://www.youtube.com/watch?v=zAQacLrItNY>

Cough sounds

<https://www.youtube.com/watch?v=MV9TaoN7cHw>

<https://www.youtube.com/watch?v=BwfscnYTnNA>

<https://www.youtube.com/watch?v=JzbPkywYfh0>

<https://www.youtube.com/watch?v=gZcFgmlqX6k>

<https://www.youtube.com/watch?v=TcmaHbc2xbE>

<https://www.youtube.com/watch?v=jjwWAtmYYtw>

<https://www.youtube.com/watch?v=ZD47Mhzm-CU>

<https://www.youtube.com/watch?v=KkEWHls-tHY>

<https://www.youtube.com/watch?v=86yCdiE3rYA&list=PLWMgRXX-UehKekLdKDuCVM3AwPoI5pnbK&index=12>

Background sounds

<https://www.youtube.com/watch?v=nYWaDjr8sfs>

<https://youtubetomp3music.com/en4/download?url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DOMaHyZmbqdc>

## Cough Detection (Modelling)

Table S1. Neural network training parameters.

Training Option	Parameter
Momentum	0.9
Initial learn rate	1.0000e-04
Learn rate drop factor	0.1
Learn rate schedule	None
Learn rate drop period	10
L2 Regularization	1.0000e-04
Gradient threshold method	L2norm
Gradient threshold	INF
Max epochs	15
Mini batch size	200
Validation frequency	300
Validation patience	INF
Shuffle	Every epoch
Execution environment	GPU



Fig S2. Training progress of the artificial neural network used for the classification of coughs

