

Oxygen Cylinders Monitoring System in Hospital Warehouse Using CNN



Pranav Suryadevara and S. Muthulakshmi

Abstract In recent years, a shortage of oxygen cylinders was observed in hospitals causing numerous deaths. Thus, a need for efficient monitoring of the quantity of the oxygen cylinders in a hospital storage facility becomes the need of the hour. To address this issue, we have come up with a solution of a surveillance system to keep track of the quantity of the oxygen cylinders. A surveillance system typically uses a camera to spy and observe a specific target. The process of monitoring can be done in both real time (live video-based) and in offline, where the surveillance can happen at a later time on a recorded video. This project employs the real time tracking of the oxygen cylinders in hospital store house using convolution neural network.

Keywords Oxygen cylinders · Surveillance · Real time tracking · Convolution neural network

1 Introduction

Currently, there is not an automated system in hospitals to check the number of oxygen cylinders in a hospital storage room. The process of monitoring the number of oxygen cylinders in a hospital is done manually, and the count is entered into spreadsheets electronically to maintain record. This process is prone to human error and may prove to be catastrophic when unrecognized in times of shortage. This is what happened in Gorakhpur, India in the year 2017 which led to the death of 60 infants in a hospital [1]. Also, considering the current covid-19 pandemic situation, it is very much essential to maintain enough stock of oxygen cylinders to serve the need of severely affected patients who are in need of oxygen [2]. Hence, we propose an oxygen cylinder monitoring system which can be considered as a solution for the said problem.

P. Suryadevara · S. Muthulakshmi (✉)
School of Electronics Engineering (SENSE), Chennai, India
e-mail: Muthulakshmi.s@vit.ac.in

P. Suryadevara
e-mail: Suryadevara.pranav2017@vitstudent.ac.in

Oxygen cylinder monitoring system is a low-cost solution for alarming issues such as deaths due to shortage of essential necessities like oxygen cylinders. This can be implemented on a commercial scale in an economic and simple to use manner in hospital storage facilities [3]. It can also be extended to monitor other important medical gases like Carbon Dioxide, Nitrogen, Nitrous Oxide and commercially used gases like Medical Air (which would be indicating to the clean supply of compressed air used for driving the ventilators, as a carrier for anesthesia agents and also incubators in hospitals).

Our project mainly focusses on providing the count of the oxygen cylinders present in a storage at any point of time. The device is semi-autonomous which requires the human user to input the number of oxygen cylinders in case of newly added cylinders. It is proposed to use a communication module so that the oxygen cylinder supplier can directly receive the message which reduces the notification delay due to human fallacies. Also, a mobile or a web application would have an even more effective impact where it could be monitored by Health Officials. Convolutional neural networks (CNNs) are used in designing oxygen monitoring system (OMS), since it can be used in analyzing image data.

This paper is organized as follows. Section 2 discusses the hardware components and software tools used in implementing this work. This chapter further discusses the working principle of the hardware prototype and also the work flow of the software in detail. Result of this work is explained in Sect. 3.

2 Methods and Materials

2.1 Hardware

2.1.1 Components

The main components for this oxygen cylinders monitoring system are Raspberry Pi and a web camera (Fig. 1).

Raspberry Pi 4B

The Raspberry Pi 4B is a very compact, cost-effective, a powerful, and a miniaturized computer that is capable of being integrated with a standard computer. It comes in different models offering a wide range of specifications, namely, such as the Raspberry Pi 2, Raspberry Pi 3, Raspberry Pi 3B, Raspberry Pi 3B+, Raspberry Pi 4, Raspberry Pi 4B launched by the Raspberry Pi foundation [4].

USB Webcam

A webcam is typically a small video camera capable of recording a video that can feed or streams a video in a real time to a computer or through a computer to a computer network such as the Internet [5]. They are usually small and can be attached in any



Fig. 1 Hardware setup of the proposed system

desired location. It is interfaced to the Raspberry Pi's USB port through a USB hence making it easy to use.

2.2 Software

2.2.1 Programming Language and Platforms

Python: This is a general-purpose, high level, interpreted programming language used to program the Raspberry Pi. The execution command is written using this language is interpreted by the processor which in turn produces the desired results.

2.2.2 Operation System

The operating systems which was used in this prototype of Raspberry Pi is Raspbian (now Raspberry Pi OS). It is the operating system officially supported by the Raspberry Pi foundation for the series of Raspberry Pi. It comes in hand with ample amounts of software already installed for educational, programming and general-purpose uses. It has Scratch, Python, Java, Sonic pi and many more installed.

2.2.3 Machine Learning Algorithms

The machine learning algorithm used in this prototype is a convolution neural network (CNN) [6]. Typically, a feed forward neural network or a multiple hidden layer perceptron model with many layers in an artificial neural network (ANN) is usually known as deep neural networks (DNNs). CNN is an efficient algorithm for recognition which is extensively employed in the fields like recognition of patterns and

processing of images. It has many advantageous aspects which include simplicity of the structure, requirement of less training parameters as well as examples and finally adaptability. It improves the model by reducing the complexity of the network by taking only a few features which are predominant for the identification and reducing associated weights [7].

Typically, the structure of a convolution neural network consists of two types of layers. They are a feature extraction layer (layers where filters are used to extract the data) and a feature mapping layer (a pooling layer where the image size is reduced by keep the features intact) [8]. The feature extraction layer is responsible for extracting local features from the input of every neuron connected to the local receiving fields of the previous layers. Once the features are extracted, the relationship between the extracted feature and other features would be determined. The other layer is responsible for the mapping of the features where every computing layer of the neural network composes a plurality of the feature map. Each feature map is a plane, the weight of the neurons in the plane is equal which helps the network learn in parallel. This is a major advantage attributed to a convolution neural network. The activation function used in a neural network is a sigmoid function which makes a feature map have full shift invariance. Every convolution neural network has a convolution layer which would be followed by a pooling layer (also known as computing layer). The pooling layer is used to calculate the local average and the second is to extract, and this distinct dual feature extraction structure reduces the size of the data (here resolution since it is an image). The process of the CNN is depicted in Fig. 2 which depicts the process the input being subjected to convolution filters and max-pooling layers and finally the output from the fully connected neurons.

A convolution neural network algorithm consists of two very essential processes which would be convolution and pooling. The process of convolution uses a set of standard filters in start so that it can derive features which can be used to develop the trainable filters [9]. This convoluted image is now further subjected to a process called pooling which reduces the size of the image keeping the features intact. This process helps a lot as the model can now be trained faster and would more effective. The convolutions used here were of the size 3X3 filters which make the input of size

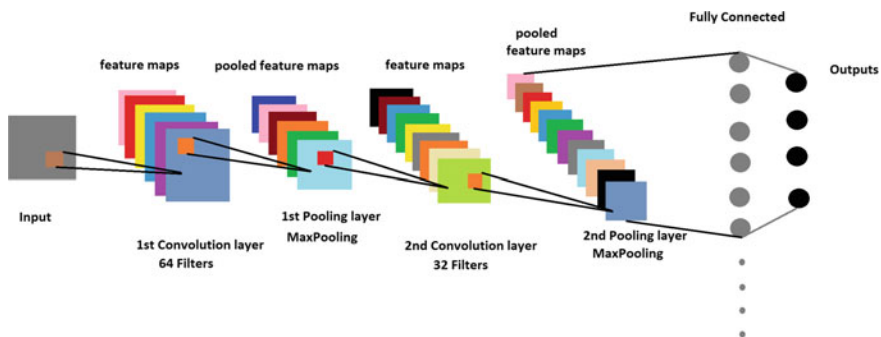


Fig. 2 Shows a representation of a multi-class classification convolution neural network

MXN pixels to result in an image of size $M-2 \times N-2$ (since the edge pixels could not have a convolution output thus canceling layers on all the 4 sides). The pooling layer helps us in reducing the size by grouping pixels and returning a pixel that represents the group of selected pixels (the pooling layer used here was a max-pooling layer which takes a set of 4 pixels and returns the max value as the pixel represented.) thereby reducing the size of the image (data) which also has the features enhanced [10].

2.2.4 Python Libraries

TensorFlow is a free and open-source software library for dataflow and differentiable programming useful across a range of tasks. It is a symbolic math library used for machine learning applications such as neural networks [11].

OpenCV (Open-Source Vision Library) is an open-source computer vision and machine learning software library. It was built to provide a common infrastructure for computer vision applications [12].

H5py package is a python interface to the HDF5 binary data format. It lets you store amounts of numerical data and easily manipulate that data from Numpy. A HDF5 (Hierarchical Data Format) file is a container for two kinds of objects: datasets which are array-like collections of data and groups which are folder-like containers that hold the datasets and other groups [13].

Scipy is a python based ecosystem of open-source software for mathematics, science and engineering used for scientific and technical computing. It contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering. It has core packages that include Numpy, Pandas, Matplotlib [11].

Numpy is a library for the python programming language which is predominantly used in the data science community for handling data for machine learning implementations. It helps extensively by adding support for large, multi-dimensional arrays and matrices (as mainly most data is stored in these data structures), along with a large collection of high-level mathematical functions to operate on these arrays. From a company point of view, this library finds its main uses in cleaning the data before it is further processed and refined [11].

Pillow is a python imaging library is a free and open-source library for the python programming language that provides support for opening, manipulating and saving many different image file formats [11].

2.3 Working Methodology

2.3.1 Circuit and Setup

The proposed OMS is designed by interfacing USB web camera with a Raspberry Pi. The working model is shown in Fig. 1 The task of powering the raspberry pi can be done by using power bank (as done in this prototype, also as a backup) but it is suggested to use it via an electrical socket so that it could be switched on at all times leaving no room for fallacies. Adhesives are used to secure the components in the desired locations, and care is taken so that the USB camera is properly fixed.

2.3.2 Working

This system setup has a camera placed near the exit door of a facility which helps in detecting the oxygen cylinders when being taken for use. The main focus of the project is that the camera sends a live image taken every 3 min. This 3-min time delay is considered because the model takes this specified time to execute and return a result. During this time interval, the image is processed and check for the oxygen cylinders in the image. If the system detects the oxygen cylinders, then the number of oxygen cylinders in the image are counted. If the total number of cylinders drops below, a threshold value then a message is displayed to notify the issue.

This implementation assumes that oxygen cylinders as shown in Figs. 3 and 7 are distinguishable on the basis of a color (which is Green in this use case) apart from other medical supply cylinders in the storage facilities. When an image consisting

Fig. 3 Green color marked oxygen cylinder on a stand support



```

tion of 54681600 exceeds 10% of system memory.
2019-11-03 22:33:45.621054: W tensorflow/core/framework/allocator.c
tion of 54681600 exceeds 10% of system memory.
Cylinder : 95.25291323661804
1
pi@raspberrypi:~/Desktop/Praveen $

```

Fig. 4 Output of the machine leaning model in the console when input as in Fig. 3 is given

of a cylinder(s) with the appropriate color is obtained as an input then this process of detection, as in Figs. 4, and 8 would take place.

Code Summary

The flow diagram of the proposed work is given in Fig. 5. At first, the data has been collected, i.e., in this case, the input data is the images of the cylinders in a stock house. The entire dataset has been divided into training, validation and test sets. Now after dividing the data into the training, validation and test sets, the data has been reshaped and was normalized (to make all the features to be of the same scale which makes it easier for processing). Now using the keras library, we use a 2-D convolutional neural network to process the data [14]. We use max-pooling, followed by flattening and finally a dense layer to calculate the weights thus training the machine learning model. We can use different parameters to define the model. We have used an Adam optimizer, loss calculation using a binary categorical cross-entropy with the metric being used is accuracy [15]. The model learns the features from the training images to detect the desired objects from the set of images given as input [16]. We finally fit the model using these metrics and the data. We define a function to stop the model evaluation when the accuracy is reached to a desired threshold level (95% in this case) on the test data. Once the desired accuracy is obtained, we stop the model and it is capable of detecting a new test image on an unseen images thereby detecting the required object by using the feature learnt from the training data [17]. The code also draws a contour around the required object to be detected thus making it easier to cross verify the results as in the case of Figs. 6 and 9. Since a CNN used was a 2-D network, it can only detect the cylinders that are shown (clearly depicted) completely. When the multiple cylinders with slight colored to differentiate, Fig. 7, as oxygen cylinders are given as input [18], then we see a slight decrease in probability as the training had a few images of that sort hence the result obtained as in Fig. 8.

3 Results and Discussions

The prototype proposed detects all the oxygen cylinders present in an input image. It also gives a probability of the model predicting that the object would be an oxygen cylinder. The oxygen cylinders are detected based on the color which is green as in the use case presented here. It was also tested in various shades of lighting to observe the effects of lighting with the model always predicting the right output. It

Fig. 5 Flowchart for the oxygen monitoring system using convolution neural networks (CNNs)

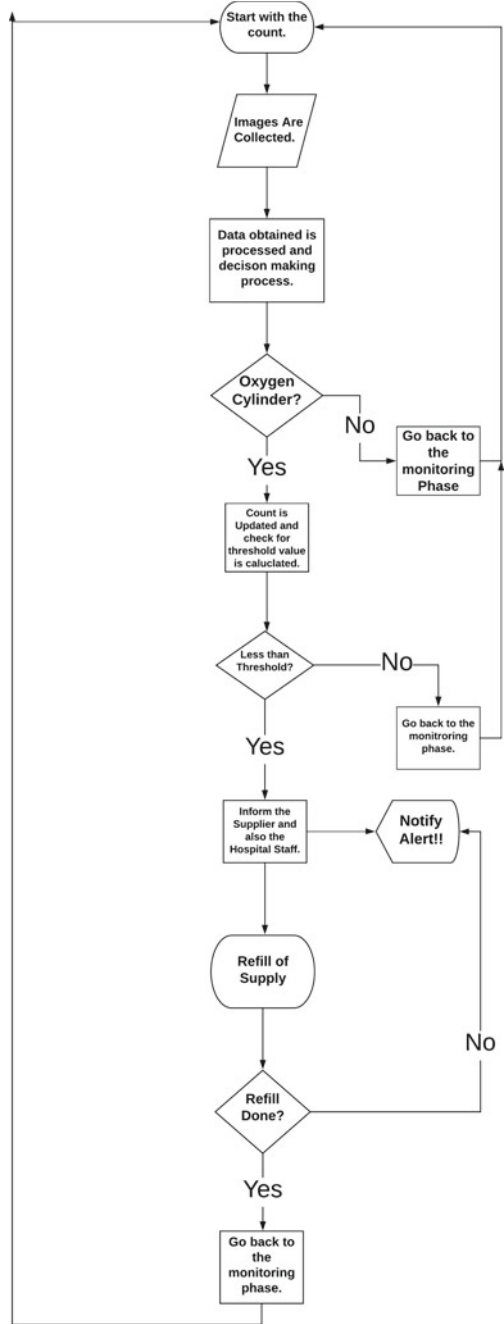


Fig. 6 The contour around the oxygen cylinder to represent the detected object



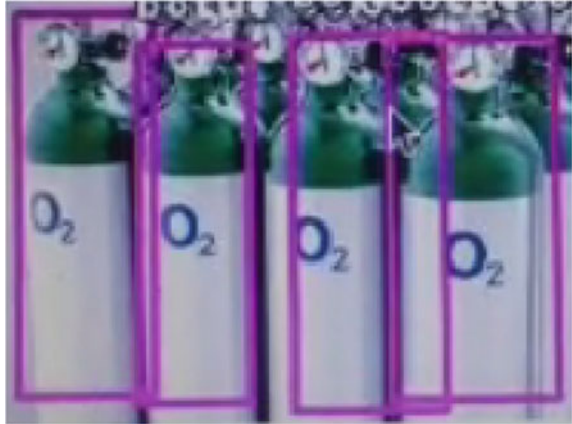
Fig. 7 Multiple oxygen cylinders in a plane surface



```
tion of 63272960 exceeds 10% of system memo  
Cylinder : 75.79192519187927  
Cylinder : 66.5608286857605  
Cylinder : 57.77403712272644  
Cylinder : 63.29222917556763  
4
```

Fig. 8 Output of the machine leaning model in the console when input is a single-oxygen cylinder as in Fig. 3

Fig. 9 The contour across each and every cylinder detected to represent detected objects



```

Threshold Set is 3 Cylinders.
Current Number of Oxygen Cylinders: 4.
Cylinder : 95.25291323661804
1
1 Cylinders are detected.
Cylinders are at Threshold value!!!
Please notify the Supplier.
  
```

Fig. 10 Output in the console when the threshold limit is exceeded

also maintains the count and updates it as soon as an oxygen cylinders are detected. The count when decreases beyond a previously set threshold it notifies the vendor (Fig. 10) once the count falls beyond the threshold. The prototype is robust and would have the capabilities to operate in incase of a power cut as it uses a portable power bank as a backup supply.

4 Conclusion

This project can be further developed by employing the use of a communication module like a GSM (Global System for Mobile Communications) that would be helpful for direct communication with oxygen cylinder supplier so as to reduce any human fallacy. Also, a website with a database which could see all the entries being logged in would be very helpful for health officials monitoring the quantity of all cylinders in all hospitals.

Acknowledgements The authors wish to thank everyone who has contributed for the project study.

References

1. [https://en.wikipedia.org/wiki/2017_Gorakhpur_hospital_deaths#:~:text=Acute%20encephalitis%20syndrome%20\(AES\)%20was,major%20cause%20for%20avoidable%20deaths](https://en.wikipedia.org/wiki/2017_Gorakhpur_hospital_deaths#:~:text=Acute%20encephalitis%20syndrome%20(AES)%20was,major%20cause%20for%20avoidable%20deaths)
2. Cheng RZ, Kogan M, Davis D (2020) Ascorbate as prophylaxis and therapy for COVID-19—update from Shanghai and U.S. Medical Institutions. *Glob Adv Health Med*. <https://doi.org/10.1177/2164956120934768>
3. Rammohana Reddy E, Sankara K (2018) Internet of things based home automation control system using Raspberry Pi. *Int J Sci Res Comput Sci, Eng Inf Technol*, 3(4):34–38
4. https://en.wikipedia.org/wiki/Raspberry_Pi
5. Patrick A, Seidel, Böhnke K (2009) Camera-based control for industrial robots using OpenCV libraries. In: International conference on research and education in robotics, pp 144–157
6. Liu T, Fang S, Zhao Y, Wang P, Zhang J Implementation of training convolutional neural networks. *Arxiv.org* (in press)
7. Bosse S, Maniry D, Müller K, Wiegand T, Samek W (2018) Deep neural networks for no-reference and full-reference image quality assessment. *IEEE Trans Image Process* 27(1):206–219. <https://doi.org/10.1109/TIP.2017.2760518>
8. He K, Zhang X, Ren S, Sun J Deep residual learning for image recognition. *Arxiv.org* (in press)
9. Simonyan K, Zisserman A Very deep convolutional networks for large-scale image recognition. *Arxiv.org* (in press)
10. <https://machinelearningmastery.com/object-recognition-with-deep-learnin>
11. <https://www.coursera.org/learn/introduction-tensorflow/home/welcome>
12. Budiharto W (2015) Intelligent surveillance robot with obstacle avoidance capabilities using neural network. Hindawi Publishing Corporation Computational Intelligence and Neuroscience, vol 2015
13. Ziganshin E, Numerov M, Vygllov S (2010) UWB baby monitor. In: 2010 5th international conference on ultrawideband and ultrashort impulse signals, pp 159–161
14. https://www.tensorflow.org/guide/keras/sequential_model
15. Taqi AM, Awad A, Al-Azzo F, Milanova M (2018) The impact of multi-optimizers and data augmentation on tensorflow convolutional neural network performance. In: 2018 IEEE conference on multimedia information processing and retrieval (MIPR), Miami, FL, 2018, pp 140–145. <https://doi.org/10.1109/MIPR.2018.00032>.
16. <https://www.coursera.org/specializations/deep-learning>
17. Singh P, Deepak BBVL, Sethi T, Murthy MDP (2015) Real-time object detection and tracking using color feature and motion. In: International conference on communication and signal processing, pp 1252–1257
18. Oh SW, Kim SJ (2017) Approaching the computational color constancy as a classification problem through deep learning. *Pattern Recogn* 61:405–416