

CoviExpert: COVID-19 detection from chest X-ray using CNN

A. Arivoli^{*}, Devdatt Golwala, Rayirth Reddy

School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India



ARTICLE INFO

Keywords:
COVID-19
Deep learning
CNN
X-ray
CoviExpert
CT scan

ABSTRACT

COVID-19 continues to threaten the world with its impact and severity. This pandemic has created a sense of havoc and shook the world stretching the medical fraternity to an unimaginable extent, who are now facing fatigue and exhaustion. Due to the rapid increase in cases all across the globe demanding extensive medical care, people are hunting for resources like testing facilities, medical drugs and even hospital beds. Even people with mild to moderate infection are panicking and mentally giving up due to anxiety and desperation. To combat these issues, it is necessary to find an inexpensive and faster way to save lives and bring about a much-needed change. The most fundamental way through which this can be achieved is with the help of radiology which involves examination of Chest X rays. They are primarily used for the diagnosis of this disease. But due to panic and severity of this disease a recent trend of performing CT scans has been observed. This has been under scrutiny since it exposes patients to a very high level of radiation known to increase the probability of cancer. As quoted by the AIIMS Director, one CT scan is equivalent to around 300–400 Chest X-rays. Also, it is relatively a much costlier testing method. Hence, in this report, we have presented a Deep learning approach which can detect covid 19 positive cases from Chest X ray images. It involves creation of a Deep learning based Convolutional Neural Network (CNN) using Keras (python library) and integrating the model with a front-end user interface for ease of use. This leads up to the creation of a software which we have named as CoviExpert. It uses the sequential Keras model which is built layer by layer. All the layers are trained independently to make independent predictions which are then combined to give the final output. 1584 images of Chest X-rays of both COVID-19 positive and negative patients have been used as training data. 177 images have been used as testing data. The proposed approach gives a classification accuracy of 99%. CoviExpert can be used on any device by any medical professional to detect Covid positive patients within a few seconds.

1. Introduction

The world has undergone a complete transformation in the past one year in all aspects whether it be economic, social or political due to the COVID-19 Pandemic. The only way to combat and defeat this pandemic is by preventing its spread and testing people as soon as they are symptomatic. Currently, the rapid antigen testing kit is considered to be the best possible instant testing method which is being used in homes and clinics, unfortunately it is not so accurate, also, the turn-around times for COVID-19 test results range from 3 to more than 48 h. Another drawback of these testing kits is that they might not be accessible everywhere around the world. Our COVID-19 detection web-based app runs on a deep learning based Convolutional Neural Network (CNN). This deep learning model analyses the X-Ray images fed to the web app by the user and in turn gives an output of whether the patient is

infected with COVID-19 or not. The model learns its detection feature using a sample space of 1584 images of chest X-Rays, in a layer-by-layer format as expected in a sequential Keras model. Each layer is independently trained to make accurate predictions by gaining knowledge for the discrete role that is considered. This model gains and applies learned knowledge on a sample to give a prediction. This ML prediction model is employed in a webapp named CoviExpert. The webapp is simple and easy to use. The user needs to upload a picture of their specific chest X-Ray, and the CoviExpert detector returns a positive or negative result. Such an application is platform independent and flexible, and in turn can be used anywhere, anytime. Also, the application is linked with the patient's history so as to aid the doctors in analysing his/her condition. This paper presents a technical view of our software covering all aspects of software design and engineering which includes Systems analysis [5], design, process specification, test plan and finally the results. The main

* Corresponding author.

E-mail addresses: arivoli.a@vit.ac.in (A. Arivoli), rushichandra.golwala2019@vitstudent.ac.in (D. Golwala), rayirthreddy.pakala2019@vitstudent.ac.in (R. Reddy).

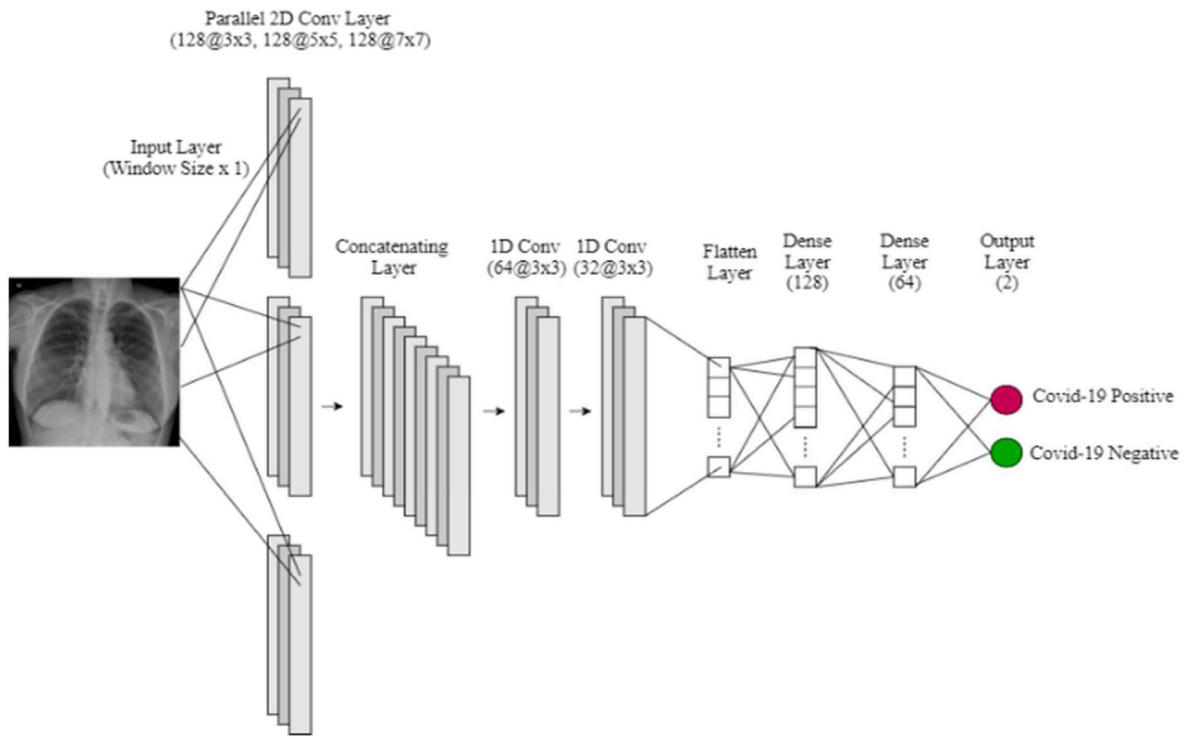


Fig. 1. System architecture.

aim of CoviExpert is to provide accurate, efficient and fast testing mechanism for all sets of people, especially those who have mild to moderate infection and can be treated at home without panicking. It is a great means of providing faith and relief to those who are doomed by misinformation, and also for critical patients as well, since it provides quick results thereby saving valuable time.

2. System analysis

2.1. Existing system: literature survey

1. The article in Ref. [1] compares various X-Ray testing methods implemented around the world and datasets developed on websites like Kaggle.

Gap: It doesn't have firm practical evidence as it is survey based.

2. The work in Ref. [2] provides a strong baseline for the problem of multi-source multi-target domain adaptation and generalization in medical imaging.

Gap: It is only limited to a small amount of data pertaining to diseases like pneumonia and other chest infections.

3. Researchers in Ref. [3] provided great knowledge about CNN and its use in radiology which would not only aid researchers in detecting covid, but also provide new patterns and recognisable links to a variety of symptoms.

Gap: It is still in its early stages of prototyping with low accuracy levels and limited support to mutations.

4. A solution to automatically classify COVID-19 cases in chest x-ray images and visualization using heat maps is depicted by the authors in Ref. [4].

Gap: It primarily focuses on the visualization of the images rather than proper training and testing of the dataset.

5. The Authors in Ref. [6] predicts the presence of COVID 19 from chest X-Ray images using a variety of algorithms like CNN, Random Forest (RF) and support vector machine (SVM)

Gap: The classification accuracy is 95.2% which is less as compared to our proposed accuracy of 99.12%. Also, this research does not deploy a web application interface which decreases ease of usability for a non-technical professional.

2.2. Proposed system

The primary objective of CoviExpert is to deploy an effective and sustainable solution for the healthcare and pharmaceutical industry. The solution encompasses the prediction of COVID-19 results by analysing the chest X-ray of the patient using deep learning. The deep learning algorithm implemented is CNN (Convolutional Neural Networks) as shown in Fig. 1. It is used to train the deep learning model which can predict whether a person is having COVID-19 or not. This solution recognizes and analyses complex patterns and connections in X-ray images across various layers aiding in efficient prediction thereby aiding radiologists and medical professionals to take critical decisions during diagnosis. The primary client of this system will be able to identify themselves as covid positive or negative with a probability of either result. Following the medical norms, the patient if chooses to undergo an X-ray [7,8] as a part of the diagnosis, will be further tested for COVID-19 using this software. The patient or radiologist in that case will upload the X-ray on to the web application and get the result along with its probability.

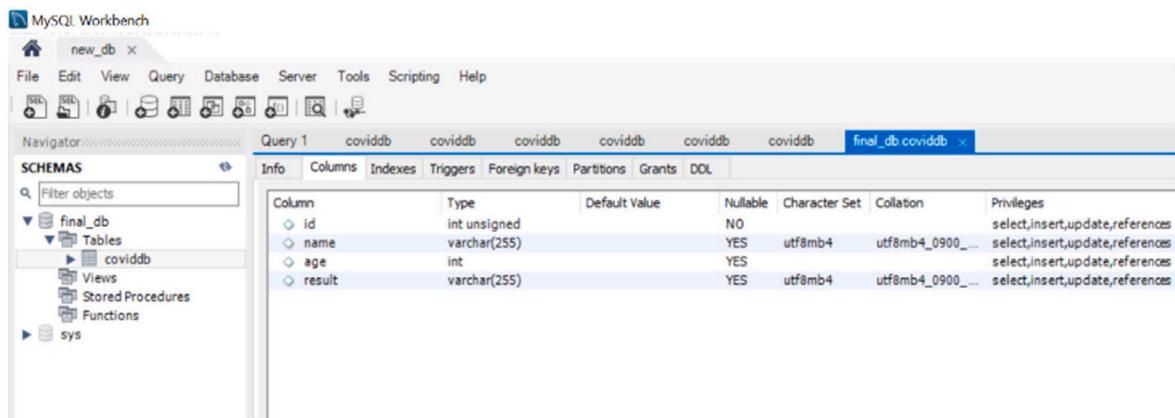


Fig. 2. Database schema.

3. System design

3.1. Detailed design

A convolutional neural network is classified as a subset of deep neural networks. It is extensively used in image processing and analysis. Drawing parallelism from the human neural network, CNN comprises of a networked layer of artificial neurons. These neurons perform mathematical operations on a set of inputs layer by layer and finally outputs possibility score which is then compared with the prediction label. They are mostly applied in identification procedures like decoding facial recognition, image and video processing, natural language processing [9,10], recommender systems, image classification, Image segmentation, medical image analysis, financial trends analysis, human-computer interaction. CNNs analyse and use the hierarchical pattern in data. The input data which is mostly in the form of an image is in vector format and as the complexity increases, feature selection increases which is the reason where the layer-by-layer training approach of CNN comes in handy using simple patterns present in the filters in Fig. 1.

- The sequential model of keras is used to train our model.
- The model represents the actual neural network and allows us to create it layer by layer
- To split the dataset into training and testing data, we use the scikit learn python library.
- After splitting the data, our model is finally trained using keras. Keras creates the model in the form of weights and trains it for a fix number of epochs (iterations on a dataset, which is 20 iterations in our case)
- Checkpoints are initialised after each iteration and history is monitored and saved.
- The process is used to train the model and finally generate validation loss, training loss, training accuracy and validation accuracy.
- At the end we evaluate the model on the test data and finally get the accuracy. A testing loss of 1% and testing accuracy of 99% is expected at the end of model training.

3.2. Database design

4. System implementation modules

4.1. Data pre-processing

The first step in creation of any deep learning model is filtering the data which is to be given as input. Data preprocessing module involves encoding data into machine understandable format. It helps to clean, format and transform the raw data into efficient datasets, thereby

making it ready for Machine Learning models. The steps involved in data preprocessing are:

- The number of input variables are reduced which is known as dimensionality reduction.
- Categorical labels COVID-19 positive and negative are created.
- The images are reduced to NumPy arrays so that mathematical computations for model training can be performed.
- Data and metadata are encoded from csv files into arrays using NumPy and pandas.

4.2. Training the convolutional neural network

- i. In the form of NumPy arrays the data and target data created is loaded.
- ii. Keras:
 - o The keras library is used to develop deep learning model.
 - o Shaping the data: Images in the form of NumPy array (data and target data) are converted to matrix form which is called shaping the data.
 - o Weight is assigned to all data.
 - o Five Keras layers which are Input Layer, Concatenating Layer, Flatten Layer, Dense Layer and Output Layer are used to break down the model.
 - o Trainable and non-trainable parameters are given by the output layer.
- iii. Scikitlearn and tensorflow:
 - o Using the tensorflow and scikitlearn python library data is split into training and Testing data.
- iv. Matplotlib:
 - o To evaluate the model graphs are plotted which show training and validation loss.
- v. Finally, the model is evaluated and its prediction accuracy is calculated.

4.3. Web application

The backend is implemented in Python and the front end is created using html, CSS and JavaScript. In python, the application links all the above-mentioned modules using their respective functions so that they are executed in a sequential manner:

- i. Data preprocessing
- ii. Index
- ii. Predict

Table 1

List of python modules.

Sr no.	Python Module	Description
1.	Flask (Python)	To implement the backend, we have used an easy-to-use microweb framework called flask in python.
2.	Keras	Python library for deep learning
3.	Jupyter	A web-based interface for creating and sharing documents.
4.	Pillow	An open-source Python programming language library which contains image processing tools to create, save and edit images.
5.	scikit-learn	A machine learning library for predictive data analysis.
6.	NumPy	It is used for manipulating arrays and data.
7.	TensorFlow	It is an open-source library for machine learning. It particularly focuses on inference and training of deep neural networks. It helps to easily build and deploy machine learning models.
8.	Matplotlib	It is a plotting library in python which provides object-oriented API for inserting plots into applications using GUI toolkits.
9.	OpenCV	It is widely used in image processing useful in reading and analysing image files in python.

4.4. Database connection

MySQL relational database will be used to store the patient details in a schematic manner in Fig. 2. Real time updation of the database will be done as soon as the report is generated and assessed by the medical professional.

5. Python modules used

Python modules and frameworks used to build this project were Flask, Keras, Jupyter, Pillow, NumPy scikit-learn, TensorFlow, matplotlib. These were useful as data science frameworks and also helped in integrating python with request response model in database and Java Script in Table 1.

6. Software test plan

6.1. STEP-BY-STEP instructions to use the interface

1. The user of CoviExpert opens the webapp and navigates to the top of the screen which contains all menu items
2. Click on ABOUT so as to land up on the testing page.
3. Start entering the patient details
 - a. First enter the patients ID (unique – primary key)
 - b. Then enter the Full name of the patient
 - c. Finally, enter the Age of the patient
4. Choose the X-ray file of the patient which needs to be uploaded so as to arrive at a prediction by clicking up on the choose file button
5. Check whether the correct file has been uploaded

6. If not, then you can re-upload the file again by clicking the choose file button
7. Once, the image of the X-ray has been uploaded, click on Predict button
8. The user will be able to see the Prediction of whether he/she is suffering from covid 19 along with probability of the prediction.
9. Now, the user can select the radio buttons depending upon what the result is.
 - a. If It is COVID-19 Positive, click on Covid positive
 - b. If it is COVID-19 negative, click on Covid negative
10. Click on submit button to store all the details of this result onto the database for future reference.
11. Now, after the main testing has been done: The user can view the main menu banner which will be on top of the screen and access additional features like
 - a. Service: Clicking on this will provide a brief overview about the services which our webapp offers
 - b. Frequently Asked Questions: clicking on this, the user will be able to see what are the FAQ's in general by previous research and users
 - c. Contact us: The user can give us their feedback and mailing them to us with the contact us section where they will be asked to enter their Full name, E-mail and feedback. The facility wherein we are located is also provided so that if anyone wants to physically meet us for queries or collaboration, can easily reach out to us
 - d. Features section: by clicking the drop down menu, the users can access the features section which informs the users about the key features of our software.

We have implemented a total of 11 major test cases in our project divided into 3 categories:

- A. Responsive User Interface Module
- B. Prediction Module

6.2. Responsive user interface

The first set of test cases is Responsive User interface which cover aspects of responsive web design and make an overall interactive experience for the user using the software. In Fig. 3, we performed this manually by taking a sample user go and interact with UI. In case of invalid inputs we observed what happens.

6.3. Prediction module

These set of test cases cover the major checkpoints of our Deep learning Based modelling and finally predicting accurate results so as to classify a patient as covid positive or negative in Fig. 4.

TEST CASE ID	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_UI_001_1	Web-app opening with all its contents and integrations	1. Need a valid hosting server and internet connection	1. Open Website link 2. Hover to the database module	<Valid flask and front end integration> <Valid Css code>	Successful Opening of the web app	As expected	Pass
TC_UI_001_2	Animations working when hovered to.	1. Need a valid animation scheme	1. Hover to different webapp sections 2. Have interactive buttons	<Valid css code> <Valid Css code>	Animations perform efficiently across all platforms (phone, laptop, tablets)	As expected	Pass
TC_UI_001_3	Required Fields	1. Need valid information about Patients	1. Enter the field marked with * 2. Click Submit	<Valid field information(not empty)>	Data is accepted successfully	As expected	Pass
TC_UI_001_4	Error Message popups	1. Need a valid reason for them to show up	1. Enter any invalid data 2. Click Submit	<Invalid data field(eg. Invalid image)>	A message corresponding to the error will be shown	As expected	Pass

Fig. 3. Response user Interface.

TEST CASE ID	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_PRED_003_1	Train and Test the CNN model	1. Need A valid training and testing dataset	1. Enter Training data 2. Enter Testing data 3. Train and test model	<Valid training data> <Valid testing data>	A highly accurate model	As expected (Accuracy of 99.11%)	Pass
TC_PRED_003_2	Verify correct prediction-with valid image	Need a valid jpeg/png image file of the xray	1. Choose Image 2. Press predict button 3. Check Probability	<Valid Image of the patient>	Able to see the prediction with a probability of greater than 95% (COVID +ve or -ve)	As expected	Pass
TC_PRED_003_3	Verify correct prediction-with valid image	Need a valid jpeg/png image file of the xray	1. Choose Image 2. Press predict button 3. Check Probability	<Invalid User Name>	Inavlid image type	As expected	Pass

Fig. 4. Prediction module.

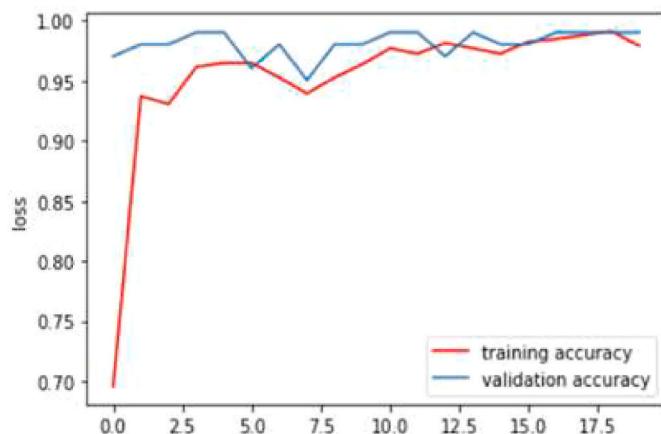


Fig. 5. Training and validation Accuracy.

7. Result and discussions

After training the keras sequential model for 20 epochs (iterations) we get the final weighted model. Fig. 5 depicts Graph of training accuracy and validation accuracy versus number of iterations (This is not in percentage, it is decimal. So as we can see, the accuracies becomes almost close to 100% at the end of the 20th iteration). Fig. 6 depicts Graph of Training loss and validation loss versus the number of iterations (the losses minimise to less than 10% till the final iteration). Fig. 7 depicts the prediction accuracy outcomes of the proposed system. Figs. 8-13 depicts the application deployment of COVIEXPERT.

Performance is dependent on factors like accuracy, precision, recollect, and efficiency, as shown in the following formule as shown in Eqs (1)-(3) and Eqn (4).

True Positive - TP: suggests that if the example is positive, it has been detected normal

False Negative - FN: suggests that if the example is positive it's detected as negative.

True Negative - TN: suggests that if the example is negative it's detected as negative.

False Positive - FP: suggests that if the example is negative it's detected as positive

```
In [21]: from matplotlib import pyplot as plt
plt.plot(history.history['loss'], 'r', label='training loss')
plt.plot(history.history['val_loss'], label='validation loss')
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```

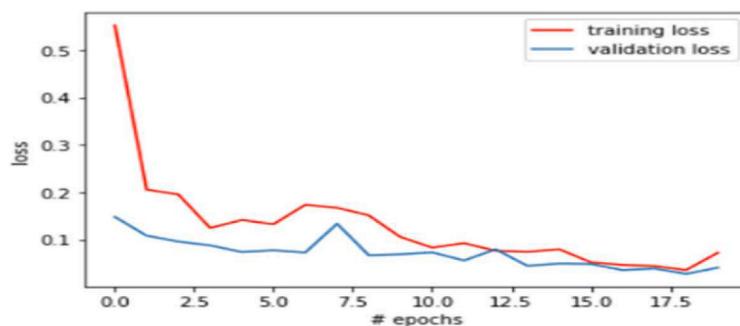
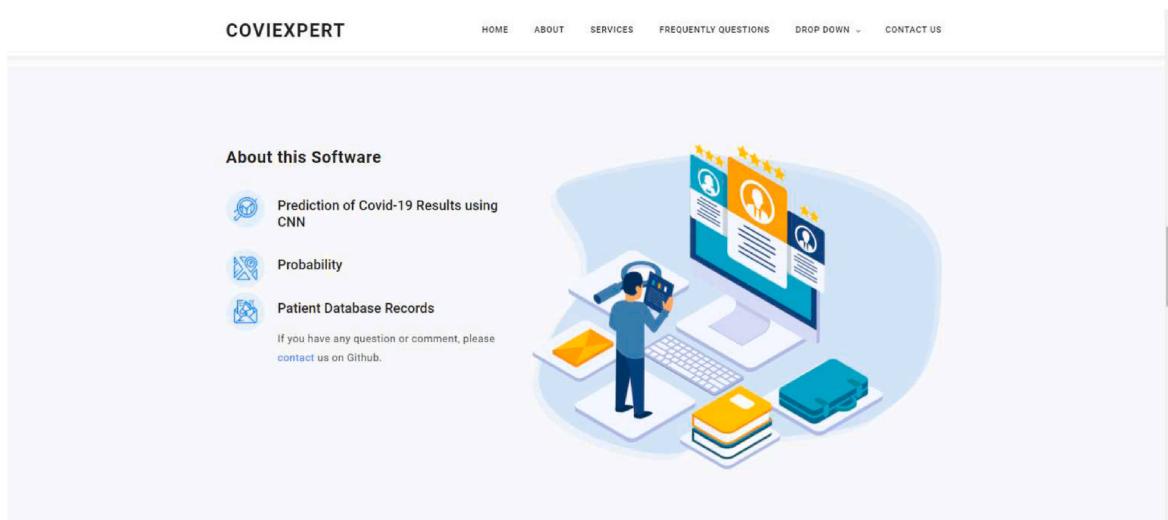


Fig. 6. Training and validation loss.

```
In [23]: print(model.evaluate(test_data,test_target))
```

```
113/113 [=====] - 0s 2ms/step
[0.028494504000167405, 0.991150438785553]
```

Fig. 7. Final accuracies.

**Fig. 8.** Testing interface 1.**Fig. 9.** Testing interface 2.**Fig. 10.** About this software – interface.

The screenshot shows the COVIEXPERT website's FAQ section. The header includes links for HOME, ABOUT, SERVICES, FREQUENTLY QUESTIONS (which is red), DROP DOWN, and CONTACT US. The main content area has a yellow background. On the left, a box titled "Curious?" contains text about misconceptions and misinformation, and an email address for contact. On the right, a list of frequently asked questions is shown, each with a blue arrow icon:

- How is this software accurate?
- Second Question
- Third Question for you
- Fourth Question Asked
- Fifth Ever Question
- Sixth Sense Question

Fig. 11. Frequently asked questions.

The screenshot shows the COVIEXPERT website's contact us page. The header is identical to Fig. 11. The main content area features a map of Vellore Institute of Technology and its surroundings on the left, and a large blue contact form on the right. The contact form includes fields for Full Name, E-mail, and Your Message, along with a Send it! button and social media sharing icons at the bottom.

Fig. 12. Contact us using iframes

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Efficiency = \frac{2 * Recall * Precision}{Recall + Precision}$$

$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN} \quad (4)$$

This is the final model accuracy after training and testing, it is 99.12%.

8. Conclusion and future work

As the corona virus is mutating and evolving quickly, new strains from many regions have started showing up. We never know which mutation can cause more harm or can spread more rapidly, so we need

- (1) to stay alert and develop quicker methods to promote rapid detection of the virus. This research paves way for the recognition of Artificial Intelligence and machine learning algorithms as capable and potential tools for helping mankind. In this research, CNN was used to accurately predict the results of a COVID-19 patient. A classification accuracy of 99% has been achieved on the proposed model. This model is integrated with a GUI to be a full-fledged application using flask and bootstrap. The advantage of having this is that it will aid doctors and radiologists in their diagnosis and help the affected patients within a few seconds. CoviExpert aims at bridging the gap between medical science and artificial intelligence, ultimately contributing towards developing advanced medical facilities.

The greatest challenge in this study are the new variants and strains emerging from various parts of the world. It always objects the accuracy of this model and hence in the future we would like to have a wider exposure to world data, a greater number of x-ray samples so as to implement a real time training model which constantly trains itself with newer data. We would also like to implement this rapid testing method

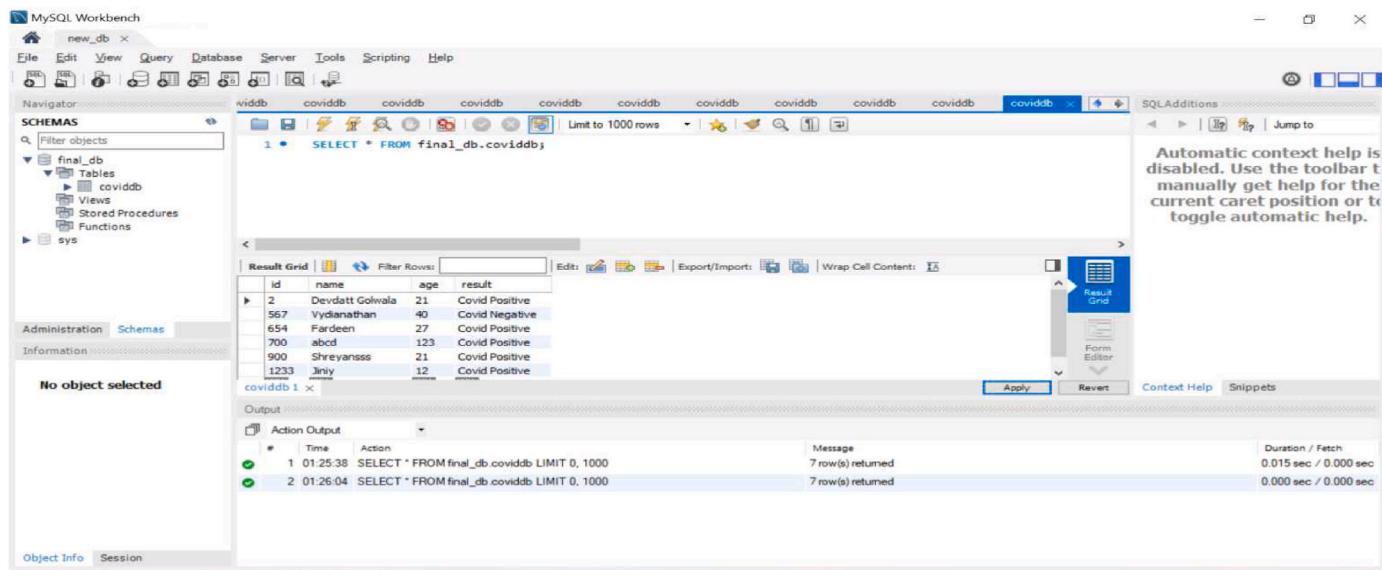


Fig. 13. Database connection.

wherever possible for instance at airports, offices, immigration centers and other bureaucratic institutions so as to help wider set of people. An improved interpretation of CNN shall be performed in the future so as to classify data on the basis of gender, age and comorbidities for improving the intelligence of our model as well as aiding the analysis of doctors during diagnosis. We hope that further study in this domain would lay out more information about the usage of machine learning and deep learning concepts with COVID-19 related data, thereby helping the world combat the pandemic in the best possible manner.

CRediT authorship contribution statement

Arivoli A. : conceptualization, supervision, project administration, writing : review and editing, validation. Devdatt Golwala: Methodology, Software, formal analysis , methodology, writing formal draft , validation. Rayirth Reddy: methodology, data curation , writing formal draft, validation

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] G. Maguolo, L. Nanni, A critic evaluation of methods for covid-19 automatic detection from x-ray images, *Inf. Fusion* 76 (2021) 1–7.
- [2] L. Yao, J. Prosky, B. Covington, K. Lyman, A Strong Baseline for Domain Adaptation and Generalization in Medical Imaging, 2019 arXiv preprint arXiv: 1904.01638.
- [3] L. Wang, Z.Q. Lin, A. Wong, Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images, *Sci. Rep.* 10 (1) (2020) 1–12.
- [4] W. Kusakunniran, S. Karnjanaprechakorn, T. Sirapisith, P. Borwarginn, K. Sutassananon, T. Tongdee, P. Saiviroonporn, COVID-19 detection and heatmap generation in chest x-ray images, *J. Med. Imag.* 8 (S1) (2021), 014001.
- [5] M. Young, *The Technical Writer's Handbook*, University Science, Mill Valley, CA, 1989.
- [6] K. Chellamuthu, B. Kiran Bala, A Novel approach for the age detection using bone X-Ray images, *Bull. Env. Pharmacol. Life Sci.* 10 (2) (January 2021) 184–187.
- [7] I. Infant Raj, B. Kiran Bala, A novel approach for infected lungs by using different transformations, *Bull. Env. Pharmacol. Life Sci.* 10 (2) (January 2021) 180–183.
- [8] Yousef Methkal Abd Algani, K. Boopalan, G. Elangovan, D. Teja Santosh, K. Chanthirasekaran, Indrajit Patra, N. Pughazendi, B. Kiranbala, R. Nikitha, M. Saranya, Autonomous service for managing real time notification in detection of COVID-19 virus, *Comput. Electr. Eng.* 101 (2022), <https://doi.org/10.1016/j.compeleceng.2022.108117>, 108117.
- [9] S. Sareen, S.K. Sood, S.K. Gupta, IoT-based cloud framework to control Ebola virus outbreak, *J. Ambient Intell. Hum. Comput.* 9 (3) (2018) 459–476.
- [10] S. Tuli, et al., HealthFog: an ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments, *Future Generat. Comput. Syst.* 104 (2020) 187–200, <https://doi.org/10.1016/j.future.2019.10.043>.