**COVID-19 DETECTION WITH CHEST CT-SCAN IMAGES BY USING CNN**

**ABSTRACT**

SARS-CoV-2, also known as COVID-19 or Corona virus, is a viral contagious disease that is infected by a novel corona virus, and has been rapidly spreading across the globe. It is very important to test and isolate people to reduce spread, and from here comes the need to do this quickly and efficiently. For this project we collected data from kaggle. According to some studies, Chest-CT outperforms RT-PCR lab testing, which is the current standard, when diagnosing COVID-19 patients. Due to this, computer vision researchers have developed various deep learning systems that can predict COVID-19 using a Chest-CT scan correctly to a certain degree. We collected data from kaggle. The accuracy of these systems is limited since deep learning neural networks such as CNNs (Convolutional Neural Networks) need a significantly large quantity of data for training in order to produce good quality results. Since the disease is relatively recent and more focus has been on CXR (Chest XRay) images, the available chest CT scan image dataset is much less. Using a pre-built predictive model, we concluded that around 40% of the generated images are correctly predicted as COVID-19 positive. The dataset thus generated can be used to train a CNN-based classifier which can help determine COVID-19 in a patient with greater accuracy.

**Keywords:**Corona virus, Convolutional neural networks, Chest CT, Multi-objective, Differential evolution.

**CHAPTER 1**

**INTRODUCTION**

Pneumonia of unknown cause discovered in Wuhan, China, was published to the World Health Organization (WHO) office in China on 31st December 2019. It was consequently assigned to Severe Acute Respiratory Syndrome [Corona virus](https://www.sciencedirect.com/topics/medicine-and-dentistry/coronavirus) 2 (SARS-CoV-2) because of having similar genetic properties to the SARS outbreak of 2003. Therefore, on 11th February 2020, WHO termed that new disease as COVID-19 (Corona virus disease), which displays an upper respiratory tract and lung infection. The clinical characteristics of critical COVID-19 pandemic are [bronchopneumonia](https://www.sciencedirect.com/topics/medicine-and-dentistry/bronchopneumonia) that affects cough, fever, dyspnea, and detailed respiratory anxiety ailment. According to the WHO reports, COVID-19’s general indications are equivalent to that of ordinary flu, including fever, tiredness, dry cough, shortness of breath, aches, pains, and sore throat . Those shared signs turn it challenging to recognize the virus at an ancient step. Moreover, the aforementioned is a virus, which works on bacterial or [fungal infections](https://www.sciencedirect.com/topics/medicine-and-dentistry/mycosis)  with no possibility that antibiotics can restrict it. Besides, people suffering from medical complications, like diabetes and chronic respiratory and cardiovascular diseases, can undergo COVID-19. An explanatory statement of the Imperial College advised that the affection rate will be more than 90.0% of the world’s people, killing 40.6 million people if no reduction actions are grasped to combat the virus.

Advanced presumed discovery of COVID-19 is also a challenge for public health security and control of pandemics. The COVID-19 detection failure increases the mortality rate exponentially. The [incubation period](https://www.sciencedirect.com/topics/computer-science/incubation-period), which is a time between catching the virus and causing to have indications of the illness, is 1∼14 days, making it remarkably challenging to identify COVID-19 infection at a preliminary stage of an individual’s symptoms. The clinical screening test for the COVID-19 is [Reverse Transcription Polymerase Chain Reaction](https://www.sciencedirect.com/topics/medicine-and-dentistry/reverse-transcription-polymerase-chain-reaction) (RT-PCR), practicing respiratory exemplars. However, it is a manual, complicated, tiresome, and time-consuming fashion with an estimated true-positive rate of 63.0%. There is also a significant lack of RT-PCR kit inventory, leading to a delay in preventing and curing corona virus disease . Furthermore, the RT-PCR kit is estimated to cost around 120∼130 USD. It also requires a specially designed biosafety laboratory to house the PCR unit, each of which can cost 15,000∼90,000 USD . Nevertheless, using a costly screening device with delayed test results makes it more challenging to restrict the disease’s spread. Inadequate availability of screening workstations and measurement kits constitutes an enormous hardship to identify COVID-19 in this pandemic circumstance. In such a situation, speedy and trustworthy presumed COVID-19 cases are a massive difficulty for related personals.

However, it is observed that most of the COVID-19 incidents have typical properties on radiographic [Computed Tomography](https://www.sciencedirect.com/topics/medicine-and-dentistry/computer-assisted-tomography) (CT) and X-ray images, including bilateral, multi-focal, ground-glass opacities with a peripheral or [posterior distribution](https://www.sciencedirect.com/topics/computer-science/posterior-distribution), chiefly in the lower lobes and early- and late-stage pulmonary concentration . Those features can be utilized to build a sensitive Computer-aided Diagnosis (CAD) tool to identify COVID-19 pneumonia, which is deemed an automated screening tool. Currently, [deep Convolutional Neural Networks](https://www.sciencedirect.com/topics/computer-science/deep-convolutional-neural-networks) (CNNs) allow for building an end-to-end model without requiring manual and time-consuming feature extraction and engineering  demonstrating tremendous success in many domains of [medical imaging](https://www.sciencedirect.com/topics/computer-science/medical-imaging), such as arrhythmia detection skin [lesion segmentation](https://www.sciencedirect.com/topics/computer-science/lesion-segmentation) and [classification](https://www.sciencedirect.com/topics/computer-science/classification) breast cancer detection , brain disease segmentation and classification , pneumonia detection from chest X-ray images fundus [image segmentation](https://www.sciencedirect.com/topics/computer-science/image-segmentation) and lung segmentation . Most recently, various deep CNN-based methods have been published for identifying COVID-19 from X-rays and CT images, summarizing and bestowing, where the mixed datasets indicate that data have come from different open-sources.

Though the results obtained in the current articles are promising, they exhibit limited scope for use as a CAD tool, as most of the works, especially on x-ray images, have been based on data coming from different sources for two distinct classes (Covid Vs. Normal) . This brings inherent bias on the algorithms as the model tends to learn the distribution of the data source for [binary classification](https://www.sciencedirect.com/topics/computer-science/binary-classification) problems. Therefore, these models perform very low when used in practical settings, where the models have to adapt to data from different domains. Recently, the authors in launched a public chest [volumetric CT](https://www.sciencedirect.com/topics/medicine-and-dentistry/volume-ct) scan dataset with 1110 COVID-19 related studies (see details in Section). However, the published articles on this dataset consider only intra-slice spatial voxel information to isolate COVID-19 and regular healthy patients.

**1.1 Proposed System:**

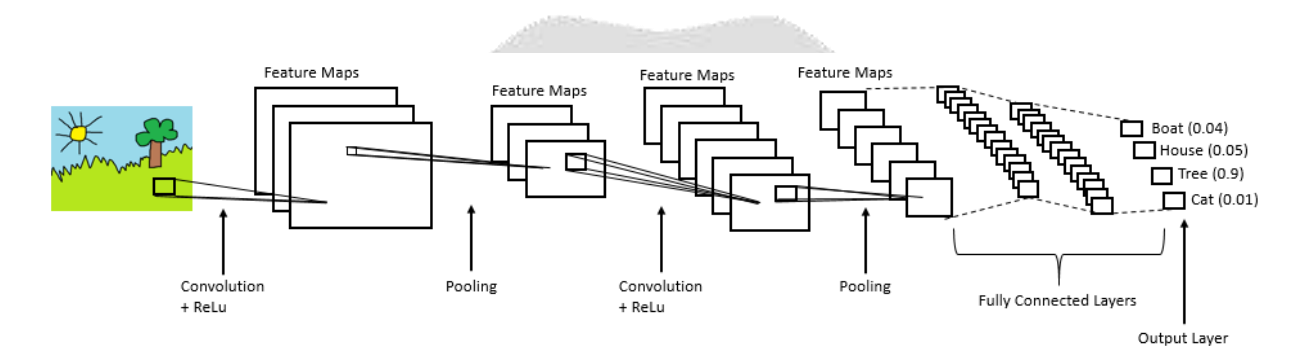
Convolutional Neural Network (CNN) is used to classify the data. CNN model is proposed for classification of dataset. Dataset is divided into three parts: training data, validation data and test data. Training data is used to train the model. The model uses the training images to train itself. Validation data is used to verify the training process and also determine the validation accuracy. The test data is used to determine the accuracy of the model; it is unknown data which is used to test the model. Initially, a CNN model is developed and dataset is imported. Then we finding the accuracy by using parameters such as precision, recall, F1 score and confusion matrix is calculated.

**1.2 OBJECTIVE:**

Our main objective is to detect the COVID-19 DETECTION WITH CHEST CT-SCAN IMAGES by using CNN. For this project we take CT scan images data. Finally, to evaluate the model we use classification report and confusion matrix.

**1.3 Convolutional Neural Network:-**

CNN is a deep learning algorithm which takes input image, assigns weights according to the features in the image and be able to differentiate one from another. CNN is used for image recognition, object classification and face recognition. Layers of CNN include Convolution layers with filters, pooling layers, fully connected layers. The block diagram of CNN is shown below:



**Fig1. Architecture of Convolutional Neural Network**

**1.3.1 Layers of CNN:**

**1.3.1.1 Convolutional layer:**

The objective of the convolutional operation is to extract the features such as color, edges, gradient, orientation, etc. from input image. Feature maps are used to extract the important features of the image. It is basically a matrix multiplication of feature map and input image. This layer reduces the dimension of the input image.

**1.3.1.2 Pooling layer:**

Similar to the convolutional layer, the pooling layer is used to reduce the dimension of the convolved feature. It is useful for extracting dominant features which are rotational and positional invariant. This helps in effective training of the model. There are two types of pooling: Max pooling and Average pooling. Max pooling returns the maximum value from the portion of the image covered by the kernel matrix. Whereas Average pooling returns the average of all the values.

**1.3.1.3 Flattening:**

Flattening layer reduces the three- dimensional matrix into one dimensional matrix so that it can be easily given as an input to the next layer.

**1.3.1.4 Fully connected layer:**

In this layer, all the inputs from one layer are connected to every activation unit of the next layer.

**1.4 Advantages:**

* Based on feature value, the classification output is generated and accuracy is calculated. The computation time is high and accuracy is low in SVM based covid19 positive and covid19 negative detection.
* To compare with other algorithms CNN is a best fit for this covid19 detection.
* By using CNN we get some good accuracy and predictions.

**CHAPTER-2**

**LITERATURESURVEY**

Individual Project Literature Review 6ET011

~ 7 ~

**Literature Review:**

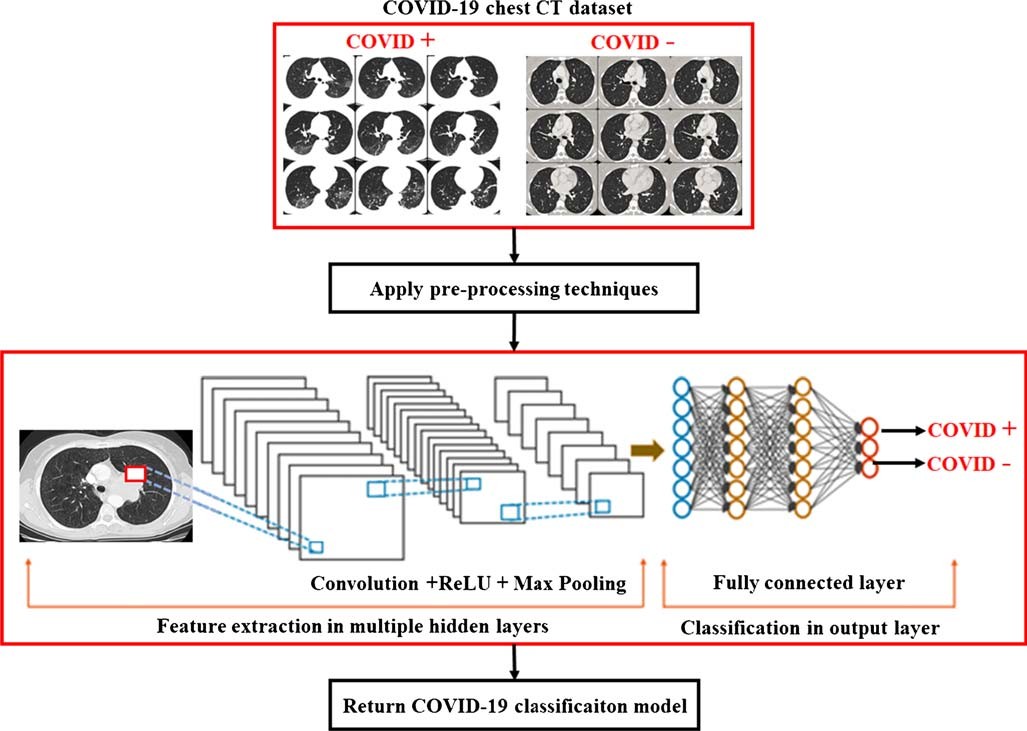
Recently, researchers have perceived the imaging patterns on chest CT for detecting the COVID-19 in chest CT Fang et al. studied the sensitivity of RT-PCR and chest CT during the detection of COVID-19. They analyzed the travel history and symptoms of 2 patients and found that the sensitivity of chest CT for detection of COVID-19 is much higher than RT-PCR. Xie et al. also reported that the 3% of 167 patients had negative RT-PCR for COVID-19 detection. However, chest CT has better sensitivity of detection of COVID-19 over RT-PCR. Bergheim et al. studied 121 infected patients’ chest CT from four different centers of China. The relationship between CT scan and symptom onset is established. They found that the severity of disease increased with time from onset of symptoms and designated the signs of disease. Recently, deep learning techniques have been widely used in detection of acute pneumonia in chest CT images

Li et al. developed a deep learning model named as COVNet to extract visual features from chest CT for detection of COVID-19. They used visual features to distinguish between community acquired pneumonia and other non-pneumonia lung diseases. However, COVNet is unable to categorize the severity of this disease. Gozes et al.developed an artificial intelligence-based CT analysis tool for detecting and quantification of COVID-19. The system extracted slice of opacities in the lungs automatically. The developed system achieved 98.2% sensitivity and 92.2% specificity. The output of system provides quantitative opacity measure and 3D volume display for opacities. The system is robust against pixel spacing and slice thickness Shan et al. developed a deep learning-based system named VB-net for automatic segmentation of all the lung and infection sites using chest CT. Xu et al.developed a prediction model to discriminate COVID-19 pneumonia and influenza-A viral pneumonia using deep learning techniques. The CNN model was used for prediction. The maximum accuracy obtained from prediction model was 86.7%. Wang et al. investigated the radiographic changes in CT images of infected patients. They developed a deep learning-based prediction model that utilizes the modified inception transfer learning technique. The features are extracted from CT images for prior diagnosis. The accuracy of 89.5% obtained from this method is better than Xu’s model and saved time for diagnosis. Narin et al. proposed an automatic deep convolution neural network–based transfer models for prediction of COVID-19 in chest X-ray images. They used InceptionV3, Inception-ResNetV2, and ResNet50 models for better prediction. The ResNet50 pre-trained model produced accuracy of 98%, which is higher than Sethy et al. developed a deep learning model for detecting COVID-19 from X-ray images. They extracted deep features and transferred them to support vector machine for classification. The accuracy of 95.38% obtained from the proposed model, which is better.

From the extensive review, it has been found that the chest CT images can be used for early classification of COVID-19-infected patients Therefore, in this paper; computational models are used to classify COVID-19 patients from chest CT images.

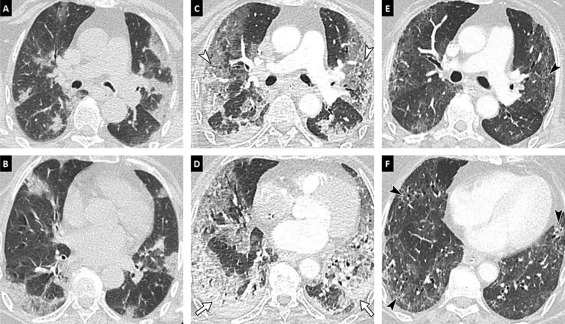
**CHAPTER-3**

**3.1BLOCK DIAGRAM:**



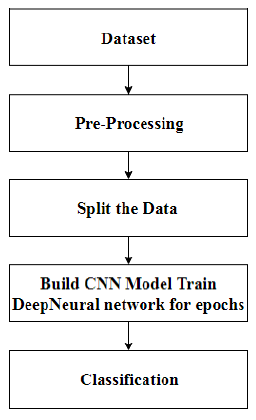
**Fig 2: Block Diagram**

**3.2 CT Scan Images Used:**



**Fig 3: Chest CT scans Images of Covid-19**

**3.3 Flow Chart:**

****

**Fig 4: Flow Chart**

**CHAPTER -4**

**HARDWARE AND SOFTWARE DESCRIPTION**

**4.1 Software Requirements:**

Operating System: Windows 8.

Front end: JUPYTER NOTEBOOK, GOOGLE COLAB, DJANGO.

Coding language: PYTHON.

**4.2 Hardware Requirements:**

Processor Type: Pentium i3.

Speed: 3.40GHZ.

Ram: 4GB DD2 RAM.

Hard disk: 500 GB.

Keyboard: 101/102 Standard Keys.

Mouse: Optical Mouse.

**4.3 Software Technologies Description:**

**JUPYTER NOTEBOOK:**

Jupyter Notebook is an open-source web app that lets you create and share documents with live code, equations, visualizations, and narrative text. Data cleansing and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and many other applications are all possible. After the analysis task, the software requirement specification is created. The function and performance allocated to software as part of system engineering are developed by establishing a complete information report as functional representation, a graphical representation of system behavior, a list of performance requirements and design constraints, and appropriate validation criteria.

**FEATURES OF JUPYTER NOTEBOOK:**

Code editing in the browser, with syntax highlighting, indentation, and tab completion/introspection. The ability to run code from the browser, with computation results attached to the code that generated them. Using rich media representations such as HTML, LaTeX, PNG, SVG, and others to display the result of the computation. For example, figures rendered by the matplotlib library that are suitable for publication can be included inline.

Rich text editing in the browser using the Markdown markup language, which can provide code commentary, isn't limited to plain text. The ability to use LaTeX to include a mathematical notation in markdown cells, which MathJax renders natively.

**NOTEBOOK DOCUMENTS:**

Notebook documents contain the inputs and outputs of an interactive session, as well as additional text that is intended to accompany the code but not be executed.

Notebook files can then be used as a complete computational record of a session, combining executable code with explanatory text, mathematics, and rich representations of the resulting objects. Internally, these documents are saved as JSON files with the.ipyCNN extension. They can be versioned and shared with colleagues because JSON is a plain text format. The nbconvert command can export notebooks to a variety of static formats, including HTML (for example, for blog posts), restructured Text, LaTeX, PDF, and slide shows.

The Jupyter Notebook Viewer can also share any. python notebook document that is accessible via a public URL (nb viewer). The notebook document is loaded from the URL and rendered as a static web page by this service. The results can then be shared with a colleague or published as a public blog post without requiring other users to install the Jupyter notebook. In effect, nbviewer is just a web service for the viewer, so you can use nbconvert to do your static conversions without relying on nbviewer.

**PYTHON:**

Python is a high-level programming language that is designed to be simple to read and use. It's free to use, even for commercial purposes, because it's open-source. Python is available for Mac, Windows, and Unix systems, as well as Java and .NET virtual machines.

Python, like Ruby or Perl, is a scripting language that is frequently used to create Web applications and dynamic Web content. Python is also supported by a variety of 2D and 3D imaging programs, allowing users to write custom plug-ins and extensions. GIMP, Inkscape, Blender, and Autodesk Maya are examples of applications that support a Python API. Python scripts (.PY files) can be parsed and executed right away. They can also be saved as compiled programs (.PYC files), which are commonly used as programming modules that other Python programs can reference.

**PYTHON FEATURES:**

Python has a variety of useful features that distinguish it from other programming languages. It supports object-oriented programming, procedural programming, and memory allocation that is dynamic. A few essential features are listed below.

**1) Easy to Learn and Use:**

Python is a simple programming language to learn when compared to other programming languages. Its syntax is simple and similar to that of the English language. The semicolon and curly brackets are not used; instead, the indentation defines the code block. For beginners, it is the recommended programming language.

**2) Expressive Language:**

Python is capable of performing complex tasks with just a few lines of code. For example, to run the hello world program, simply type print ("Hello World"). It will only require one line of code to run, whereas Java or C will require multiple lines.

**3) Interpreted Language:**

Python is an interpreted language, which means that each line of a Python program is executed separately. The benefit of being an interpreted language is that debugging is simple and portable.

**4) Cross-platform Language:**

Python can run on a variety of platforms, including Windows, Linux, UNIX, and Macintosh. As a result, we can say that Python is a portable programming language. It allows programmers to create software for multiple competing platforms by writing only one program.

**5) Free and Open Source:**

Python is a free programming language that anyone can use. On its official website, www.python.org, it is freely available. It has a large community all over the world working hard to create new Python modules and functions. The Python community welcomes contributions from anyone. "Anyone can download its source code without paying a penny," says open-source.

**6) Object-Oriented Language:**

Python supports object-oriented programming, which introduces the concepts of classes and objects. It allows for an inheritance, polymorphism, and encapsulation, among other things. The object-oriented method aids programmers in writing reusable code and developing applications with less code.

**7) Extensible:**

It means that other languages, such as C/C++, can be used to compile the code, allowing us to use it in our Python code. It converts the program to byte code, which can be run on any platform.

**8) Large Standard Library :**

It offers a diverse set of libraries for a variety of fields, including machine learning, web development, and scripting. Tensor flow, Pandas, Numpy, Keras, and Pytorch are just a few examples of machine learning libraries. Python web development frameworks include Django, Flask, and Pyramids.

**9) GUI Programming Support:**

For the development of a desktop application, a graphical user interface is used. The libraries used to develop the web application are PyQT5, Tkinter, and Kivy.

**10) Integrated:**

It's simple to integrate with languages like C, C++, and JAVA, among others. Python, like C, C++, and Java, executes code line by line. It makes it easy to debug the code.

**11) Embeddable:**

Other programming languages' code can be used in the Python source code. Python source code can also be used in other programming languages. It has the ability to embed other languages into our code.

**12) Dynamic Memory Allocation:**

We don't need to specify the variable's data type in Python. When we assign a value to a variable, the variable's memory is automatically allocated at run time. If the integer value 15 is assigned to x, we don't need to write int x = 15. Simply write x = 15 on a piece of paper.

**ANACONDA:**

Boa constrictor Cloud is a bundle of the executive's administration by Anaconda. Cloud makes it simple to find, access, store, and offer public note pads, conditions, and conda and PyPI bundles. Cloud additionally makes it simple to remain current with refreshes made to the bundles and conditions you are utilizing. Cloud has many helpful Python bundles, scratchpads, ventures, and conditions for a wide assortment of uses. You don't have to sign in, or even have a Cloud account, to look for public bundles, download, and introduce them.

You can assemble new condo bundles utilizing conda-fabricate, then, at that point, transfer the bundles to Cloud to rapidly impart to other people or access yourself from anyplace. The Anaconda Cloud order line interface (CLI), boa constrictor customer, permits you to deal with your record - including verification, tokens, transfer, download, eliminate and search. Connect to and deal with your Anaconda Cloud account. Transfer bundles you have made. Create access tokens to permit admittance to private bundles.

For engineers, Cloud is intended to make programming advancement, delivery, and upkeep simple by giving wide bundle the board support. Cloud considers free open bundle facilitating, just as bundle channels, giving adaptable and versatile assistance for gatherings and associations, everything being equal.

**APPLICATIONS PROVIDED IN ANACONDA DISTRIBUTION**

The Anaconda distribution comes with the following applications along with Anaconda Navigator.

* JupyterLab
* Jupyter Notebook
* Qt Console
* Spyder
* Glueviz
* Orange3
* RStudio
* Visual Studio Cod

**JupyterLab:**

This is an extensible workspace for intuitive and reproducible figuring, in light of the Jupyter Notebook and Architecture.

**Jupyter Notebook:**

This is an electronic, intuitive figuring journal climate. We can alter and run comprehensible docs while depicting the information investigation.

**Qt Console:**

It is the PyQt GUI that upholds inline figures, legitimate multiline altering with language structure featuring, graphical calltips and the sky is the limit from there.

**Spyder:**

Spyder is a logical Python Development Environment. It is an incredible Python IDE with cutting-edge altering, intuitive testing, investigating, and contemplation highlights.

**Versus Code:**

It is a smoothed out code manager with help for improvement activities like investigating, task running, and form control.

**Glueviz:**

This is utilized for multidimensional information representation across documents. It investigates connections inside and among related datasets.

**Orange 3:**

It is a part-based information mining structure. This can be utilized for information representation and information examination. The work processes in Orange 3 are exceptionally intelligent and give a huge tool compartment.

**Rstudio:**

It is a bunch of incorporated devices intended to assist you with being more useful with R. It incorporates R fundamentals and note pad.

**What is Colab?**

Google is quite aggressive in AI research. Over many years, Google developed AI framework called TensorFlow and a development tool called Colaboratory. Today TensorFlow is open-sourced and since 2017, Google made Colaboratory free for public use. Colaboratory is now known as Google Colab or simply Colab. Another attractive feature that Google offers to the developers is the use of GPU. Colab supports GPU and it is totally free. The reasons for making it free for public could be to make its software a standard in the academics for teaching machine learning and data science. It may also have a long term perspective of building a customer base for Google Cloud APIs which are sold per-use basis. Irrespective of the reasons, the introduction of Colab has eased the learning and development of machine learning applications.

**What Colab offers us?**

As a programmer, you can perform the following using Google Colab.

* Write and execute code in Python
* Document your code that supports mathematical equations
* Create/Upload/Share notebooks
* Import/Save notebooks from/to Google Drive
* Import/Publish notebooks from GitHub
* Import external datasets e.g. from Kaggle
* Integrate PyTorch, TensorFlow, Keras, OpenCV
* Free Cloud service with free GPU

**4.4 Code:**

**Importing required libraries:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import cv2

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout, MaxPooling2D

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.callbacks import EarlyStopping

import os

import sklearn.metric

**Loading Data:**

DIR = 'D:/sanjeev/covid\_cnn/'

SUBDIR\_POS = 'CT\_COVID/'

SUBDIR\_NEG = 'CT\_NonCOVID/'

print(f'Positive samples: {len(os.listdir(DIR + SUBDIR\_POS))}.')

print(f'Negative samples: {len(os.listdir(DIR + SUBDIR\_NEG))}.')

**Augmentation of Images**:

train\_datagen = ImageDataGenerator(

rescale=1./255,

horizontal\_flip=True,

rotation\_range=5,

width\_shift\_range=0.05,

height\_shift\_range=0.05,

shear\_range=0.05,

zoom\_range=0.05,

validation\_split=0.2)

train\_generator = train\_datagen.flow\_from\_directory(

DIR,

target\_size=(img\_height, img\_width),

batch\_size=BATCH\_SIZE,

class\_mode='binary',

color\_mode="grayscale",

subset='training')

validation\_generator = train\_datagen.flow\_from\_directory(

DIR,

target\_size=(img\_height, img\_width),

batch\_size=BATCH\_SIZE,

class\_mode='binary',

color\_mode="grayscale",

subset='validation')

**Visualization Data:**

plt.title('Accuracy')

plt.plot(hist.history['accuracy'])

plt.plot(hist.history['val\_accuracy'])

plt.ylabel('accuracy')

plt.xlabel('epoch')

plt.legend(['train', 'validation'], loc='upper left')

plt.show()

plt.title('Loss')

plt.plot(hist.history['loss'])

plt.plot(hist.history['val\_loss'])

plt.ylabel('loss')

plt.xlabel('epoch')

plt.legend(['train', 'validation'], loc='upper left')

plt.show()

plt.title('Recall')

plt.plot(hist.history['recall'])

plt.plot(hist.history['val\_recall'])

plt.ylabel('recall')

plt.xlabel('epoch')

plt.legend(['train', 'validation'], loc='upper left')

plt.show()

plt.title('Precision')

plt.plot(hist.history['precision'])

plt.plot(hist.history['val\_precision'])

plt.ylabel('recall')

plt.xlabel('epoch')

plt.legend(['train', 'validation'], loc='upper left')

plt.show()

**Model Building:**

def create\_model():

model = Sequential([

Conv2D(16, 1, padding='same', activation='relu', input\_shape=(img\_height, img\_width, 1)),

MaxPooling2D(),

Conv2D(32, 3, padding='same', activation='relu'),

MaxPooling2D(),

Conv2D(64, 5, padding='same', activation='relu'),

MaxPooling2D(),

Conv2D(64, 5, padding='same', activation='relu'),

MaxPooling2D(),

Flatten(),

Dense(128, activation='relu'),

Dropout(0.4),

Dense(64, activation='relu'),

Dropout(0.5),

Dense(8, activation='relu'),

Dropout(0.3),

Dense(1, activation='sigmoid')

])

model.compile(optimizer=OPTIMIZER,loss='binary\_crossentropy',metrics=['accuracy', 'Precision', 'Recall'])

return model

**Accuracy Finding:**

y\_pred = (model.predict\_generator(validation\_generator) > 0.5).astype(int)

y\_true = validation\_generator.classes

for name, value in zip(model.metrics\_names, model.evaluate\_generator(validation\_generator)):

print(f'{name}: {value}')

print(f'F1 score: {sklearn.metrics.f1\_score(y\_true, y\_pred)}')

**CHAPTER-5**

**ADVANTAGES AND APPLICATIONS**

**5.1 Advantages:**

* Based on feature value, the classification output is generated and accuracy is calculated. The computation time is high and accuracy is low in SVM based covid-19 positive and covid-19 negative detection.
* To compare with other algorithms CNN is a best fit for this covid-19 detection.
* By using CNN we get some good accuracy and predictions.

**5.2 Applications:**

* We can use this project in Health care Industries.

**CHAPTER-6**

**RESULTS AND CONCLUSION**

**6.1 Result:**

Here we detect the Covid-19 positive or negative by using CNN model. And we got accuracy 70.9%.

Additionally we do the predictions to this project.

**6.2 Conclusion:**

In this research the proposed system used CT scan image as an input to a multi-layered CNN model. We investigated the capabilities of CNN architectures by building them with small kernels, as opposed to standard deep CNN implementations that use shallow architectures with big filtering algorithms. We also discovered that shallow architectures performed worse even when employing a larger number of feature maps. The system apart from just classifying the brain tumor into yes or no categories, further classifies the covid-19 into two classes i.e. Covid positive and Covid negative. The average accuracy reached by the proposed methodology was 75%.

Accuracy of 70.9% was accomplished with the help of wide and diverse range of dataset containing more than images. The multi-layered CNN architecture containing convolution, max pool, dropout, fully connected and SoftMax layer.

[CT](https://www.sciencedirect.com/topics/medicine-and-dentistry/computer-assisted-tomography) imaging plays a key role in the management of COVID-19 pneumonia, particularly for early evaluation by allowing rapid triage of dyspneic patients, and also in the event of clinical worsening to detect complications, particularly [thromboembolic](https://www.sciencedirect.com/topics/medicine-and-dentistry/thromboembolism" \o "Learn more about thromboembolic from ScienceDirect's AI-generated Topic Pages) ones. Pulmonary CT extension correlates with clinical severity and should be assessed in a semi-quantitative manner by the radiologist. In a near future, AI techniques should make it possible to automate the positive diagnosis as well as the quantitative assessment of lesions, and perhaps allow the extraction of biomarkers to predict the outcome of COVID-19 patients.

In this section few seed points have been added which can be used by researchers to work further on this topic. An automatic classification system can be built to classify types of tumors; reinforcement learning can be used instead of the supervised CNN model which will remove the need to update the model every time a new covid is detected. Moreover, the model can be further developed for commercial purposes along with providing more features and privileges for the user.

**6.3 Future Scope:**

The future possibility could be to use a larger dataset to control the occurrence of over fitting. Additionally, extensive tuning of the hyper parameters such as optimizer and loss on the models can be performed to result in a high accuracy for Medical Image Classification with Deep Learning.

**6.4 References:**

1. World Health Organization,Novel Coronavirus(2019-nCoV) Situation Report-11. 2020. <https://www.who.int/docs/default-source/coronaviruse/situationreports/20200131-sitrep-11-ncov.pdf?sfvrsn=de7c0f7_4>. Accessed 24 March 2020

2. World Health Organization, Novel Coronavirus(2019-nCoV) Situation Report-30. <https://www.who.int/docs/default-source/coronaviruse/situationreports/20200219-sitrep-30-covid-19.pdf?sfvrsn=6e50645_2>. Accessed 24 March 2020

3. Worldometer Coronavirus. <https://www.worldmeters.info/coronavirus/countries-where-coronavirus-has-spread/>. Accessed 01 Apr 2020

4. WHO-China joint mission on coronavirus disease 2019 report, 2020

5. Zhang Y (2020) The epidemiological characteristics of an outbreak of 2019 novel coronavirus (COVID-19)-China CCDC. Zhonghua liu xing bing xue za zhi=Zhonghua liuxingbingxue zazhi 41(2):145 [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/32064853)]

6. Xie Z. Pay attention to SARS-CoV-2 infection in children. Pediatr Invest. 2020;4(1):1–4. doi: 10.1002/ped4.12178. [[PMC free article](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7331291/)] [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/32851334)] [[CrossRef](https://dx.doi.org/10.1002%2Fped4.12178" \t "_blank)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Pediatr+Invest&title=Pay+attention+to+SARS-CoV-2+infection+in+children&author=Z+Xie&volume=4&issue=1&publication_year=2020&pages=1-4&doi=10.1002/ped4.12178&)]

7. Ali T et al (2020) Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases. Radiology. 10.1148/radiol.2020200642 [[PMC free article](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7233399/)] [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/32101510)]

8. Xu X, Jiang X, Ma C, Du P, Li X, Lv S, Yu L, Chen Y, Su J, Lang G, Li Y, Zhao H, Xu K, Ruan L, Wu W (2020) Deep learning system to screen coronavirus disease 2019 pneumonia. arXiv preprintarXiv:2002.09334, 1–29

9. Wang S, Kang B, Ma J, Zeng X, Xiao M, Guo J,Cai M, Yang J, Li Y, Meng X, Xu B (2020) A deep learning algorithm using CT images to screen for corona virus disease (COVID-19). medRxiv preprint. 10.1101/2020.02.14.20023028, 1–26

10. Sethy PK, Behera SK Detection of coronavirus disease (COVID-19) based on deep features. Preprints 2020, 2020030300. 10.20944/preprints202003.0300.v1

11. Narin A, Kaya C, Pamuk Z (2020) Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural network. arXiv preprint arXiv:2003.10849

12. Wang D, Hu B, Hu C et al (2020) Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus–infected pneumonia in Wuhan, China. Jama 323(11):1061–1069 [[PMC free article](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7042881/)] [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/32031570)]

13. Xie X, Zhong Z, Zhao W, Zheng C, Wang F, Liu J (2020) Chest CT for typical 2019-nCoV pneumonia: relationship to negative RT-PCR testing. Radiology. 10.1148/radiol.2020200343 [[PMC free article](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7233363/)] [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/32049601)]

14. Fang Y, Zhang H, Xu Y, Xie J, Pang P, Ji W (2020) CT manifestations of two cases of 2019 novel coronavirus (2019-nCoV) pneumonia. Radiology 295(1):208–209 [[PMC free article](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7233358/)] [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/32031481)]

15. Song F, Shi N, Shan F et al (2020) Emerging coronavirus 2019-nCoV pneumonia. Radiology 295(1)210–217 [[PMC free article](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7233366/)] [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/32027573)]

16. Ng M, Lee E, Yang J et al (2020) Imaging profile of the COVID-19 infection: radiologic findings and literature review. Radiol Cardiothorac Imaging 2(1):e200034

17. Kong W, Agarwal P (2020) Chest imaging appearance of COVID-19 infection. Radiol Cardiothorac Imaging. 10.1148/ryct.2020200028

18. Kay F, Abbara S (2020) The many faces of COVID-19: spectrum of imaging manifestations. Radiol Cardiothorac Imaging. 10.1016/B978-0-12-814551-7.00038-6

19. Venugopal VK, Mahajan V, Rajan S, Agarwal VK, Rajan R, Syed S, Mahajan H (2020) A systematic meta-analysis of CT features of COVID-19: lessons from radiology. medRxiv. 10.1101/2020.04.04.20052241

20. Li X, Zeng X, Liu B, Yu Y (2020) COVID-19 infection presenting with CT halo sign. Radiol Cardiothorac Imaging. 10.1148/ryct.2020200026

21. Chung M, Bernheim A, Mei X et al (2020) CT imaging features of 2019 novel coronavirus (2019-nCoV). Radiology 295(1):202–207 [[PMC free article](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7194022/)] [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/32017661)]

22. Yijiu X China’s Hubei reports jump in new cases of COVID-19 after diagnosis criteria revision. National Health Commission of the People’s Republic of China website. [www.en.nhc.gov/cn/2020-02/13/c\_76515.htm](http://www.en.nhc.gov/cn/2020-02/13/c_76515.htm). Accessed 24 March 2020

23. Bernheim A, Mei X, Huang M et al (2020) Chest CT findings in coronavirus disease-19 (COVID-19): relationship to duration of infection 10.1148/radiol.2020200463 [[PMC free article](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7233369/)] [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/32077789)]

24. Li L et al (2020) Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT. Radiology. 10.1148/radiol.2020200905

25. Gozes O et al (2020) Rapid AI development cycle for the coronavirus (COVID-19) pandemic: initial results for automated Detection & patient monitoring using deep learning CT image analysis. arXiv preprint arXiv:2003.05037

26. Shan F, Gao Y, Wang J, Shi W, Shi N, Han M, Xue Z, Shi Y (2020) Lung infection quantification of COVID-19 in CT images with deep learning. arXiv preprint arXiv:2003.04655, 1–19, 2020

27. Liu K-C, Xu P, Lv W-F, Qiu X-H, Yao J-L, Jin-Feng G (2020) CT manifestations of coronavirus disease-2019: a retrospective analysis of 73 cases by disease severity. Eur J Radiol 108941. 10.1016/j.ejrad.2020.108941 [[PMC free article](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7118536/)] [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/32193037)]

28. Moeskops P, Viergever MA, Mendrik AM, De Vries LS, Benders MJNL, Išgum I. Automatic segmentation of MR brain images with a convolutional neural network. IEEE Trans Med Imaging. 2016;35(5):1252–1261. doi: 10.1109/TMI.2016.2548501. [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/27046893)] [[CrossRef](https://dx.doi.org/10.1109%2FTMI.2016.2548501" \t "_blank)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=IEEE+Trans+Med+Imaging&title=Automatic+segmentation+of+MR+brain+images+with+a+convolutional+neural+network&author=P+Moeskops&author=MA+Viergever&author=AM+Mendrik&author=LS+De+Vries&author=MJNL+Benders&volume=35&issue=5&publication_year=2016&pages=1252-1261&pmid=27046893&doi=10.1109/TMI.2016.2548501&)]

29. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, et al. Recent advances in convolutional neural networks. Pattern Recogn. 2018;77:354–377. doi: 10.1016/j.patcog.2017.10.013. [[CrossRef](https://dx.doi.org/10.1016%2Fj.patcog.2017.10.013" \t "_blank)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Pattern+Recogn&title=Recent+advances+in+convolutional+neural+networks&author=J+Gu&author=Z+Wang&author=J+Kuen&author=L+Ma&author=A+Shahroudy&volume=77&publication_year=2018&pages=354-377&doi=10.1016/j.patcog.2017.10.013&)]

30. Matsugu M, Mori K, Mitari Y, Kaneda Y. Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural Netw. 2003;16(5–6):555–559. doi: 10.1016/S0893-6080(03)00115-1. [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/12850007)] [[CrossRef](https://dx.doi.org/10.1016%2FS0893-6080(03)00115-1" \t "_blank)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Neural+Netw&title=Subject+independent+facial+expression+recognition+with+robust+face+detection+using+a+convolutional+neural+network&author=M+Matsugu&author=K+Mori&author=Y+Mitari&author=Y+Kaneda&volume=16&issue=5%E2%80%936&publication_year=2003&pages=555-559&pmid=12850007&doi=10.1016/S0893-6080(03)00115-1&)]

31. Krizhevsky A, Sutskever I, Hinton GE. Advances in neural information processing systems. 2012. Imagenet classification with deep convolutional neural networks; pp. 1097–1105. [[Google Scholar](https://scholar.google.com/scholar_lookup?title=Advances+in+neural+information+processing+systems&author=A+Krizhevsky&author=I+Sutskever&author=GE+Hinton&publication_year=2012&)]

32. Jin KH, McCann MT, Froustey E, Unser M. Deep convolutional neural network for inverse problems in imaging. IEEE Trans Image Process. 2017;26(9):4509–4522. doi: 10.1109/TIP.2017.2713099. [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/28641250)] [[CrossRef](https://dx.doi.org/10.1109%2FTIP.2017.2713099" \t "_blank)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=IEEE+Trans+Image+Process&title=Deep+convolutional+neural+network+for+inverse+problems+in+imaging&author=KH+Jin&author=MT+McCann&author=E+Froustey&author=M+Unser&volume=26&issue=9&publication_year=2017&pages=4509-4522&pmid=28641250&doi=10.1109/TIP.2017.2713099&)]

33. Zbontar J, LeCun Y. Stereo matching by training a convolutional neural network to compare image patches. J Mach Learn Res. 2016;17(1):2287–2318. [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=J+Mach+Learn+Res&title=Stereo+matching+by+training+a+convolutional+neural+network+to+compare+image+patches&author=J+Zbontar&author=Y+LeCun&volume=17&issue=1&publication_year=2016&pages=2287-2318&)]

34. Lawrence S, Giles CL, Tsoi AC, Back AD. Face recognition: a convolutional neural-network approach. IEEE Trans Neural Netw. 1997;8(1):98–113. doi: 10.1109/72.554195. [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/18255614)] [[CrossRef](https://dx.doi.org/10.1109%2F72.554195" \t "_blank)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=IEEE+Trans+Neural+Netw&title=Face+recognition:+a+convolutional+neural-network+approach&author=S+Lawrence&author=CL+Giles&author=AC+Tsoi&author=AD+Back&volume=8&issue=1&publication_year=1997&pages=98-113&pmid=18255614&doi=10.1109/72.554195&)]

35. Pannu HS, Singh D, Malhi AK. Improved particle swarm optimization based adaptive neuro-fuzzy inference system for benzene detection. CLEAN–Soil, Air, Water. 2018;46(5):1700162. doi: 10.1002/clen.201700162. [[CrossRef](https://dx.doi.org/10.1002%2Fclen.201700162" \t "_blank)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=CLEAN%E2%80%93Soil,+Air,+Water&title=Improved+particle+swarm+optimization+based+adaptive+neuro-fuzzy+inference+system+for+benzene+detection&author=HS+Pannu&author=D+Singh&author=AK+Malhi&volume=46&issue=5&publication_year=2018&pages=1700162&doi=10.1002/clen.201700162&)]

36. Pannu HS, Singh D, Malhi AK. Multi-objective particle swarm optimization-based adaptive neuro-fuzzy inference system for benzene monitoring. Neural Comput & Applic. 2019;31:2195–2205. doi: 10.1007/s00521-017-3181-7. [[CrossRef](https://dx.doi.org/10.1007%2Fs00521-017-3181-7" \t "_blank)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Neural+Comput+&+Applic&title=Multi-objective+particle+swarm+optimization-based+adaptive+neuro-fuzzy+inference+system+for+benzene+monitoring&author=HS+Pannu&author=D+Singh&author=AK+Malhi&volume=31&publication_year=2019&pages=2195-2205&doi=10.1007/s00521-017-3181-7&)]

37. Kaur M, Gianey HK, Singh D, Sabharwal M. Multi-objective differential evolution based random forest for e-health applications. Mod Phys Lett B. 2019;33(05):1950022. doi: 10.1142/S0217984919500222. [[CrossRef](https://dx.doi.org/10.1142%2FS0217984919500222" \t "_blank)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Mod+Phys+Lett+B&title=Multi-objective+differential+evolution+based+random+forest+for+e-health+applications&author=M+Kaur&author=HK+Gianey&author=D+Singh&author=M+Sabharwal&volume=33&issue=05&publication_year=2019&pages=1950022&doi=10.1142/S0217984919500222&)]

38. Kaur M, Singh D, Sun K, Rawat U. Color image encryption using non-dominated sorting genetic algorithm with local chaotic search based 5D chaotic map. Futur Gener Comput Syst. 2020;107:333–350. doi: 10.1016/j.future.2020.02.029. [[CrossRef](https://dx.doi.org/10.1016%2Fj.future.2020.02.029" \t "_blank)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Futur+Gener+Comput+Syst&title=Color+image+encryption+using+non-dominated+sorting+genetic+algorithm+with+local+chaotic+search+based+5D+chaotic+map&author=M+Kaur&author=D+Singh&author=K+Sun&author=U+Rawat&volume=107&publication_year=2020&pages=333-350&doi=10.1016/j.future.2020.02.029&)]

39. Storn R, Price K (1995) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces (Tech. Rep.), Berkeley, CA. TR-95-012

40. Zhabitskaya E, Zhabitsky M. Mathematical Modeling and Computational Science. 2012. Asynchronous differential evolution; pp. 328–333. [[Google Scholar](https://scholar.google.com/scholar_lookup?title=Mathematical+Modeling+and+Computational+Science&author=E+Zhabitskaya&author=M+Zhabitsky&publication_year=2012&)]

41. Zhang J, Sanderson AC. JADE: adaptive differential evolution with optional external archive. IEEE Trans Evol Comput. 2009;13(5):945–958. doi: 10.1109/TEVC.2009.2014613. [[CrossRef](https://dx.doi.org/10.1109%2FTEVC.2009.2014613" \t "_blank)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=IEEE+Trans+Evol+Comput&title=JADE:+adaptive+differential+evolution+with+optional+external+archive&author=J+Zhang&author=AC+Sanderson&volume=13&issue=5&publication_year=2009&pages=945-958&doi=10.1109/TEVC.2009.2014613&)]

42. Vaishali, Sharma TK. Proceedings of Fifth International Conference on Soft Computing for Problem Solving. Singapore: Springer; 2016. Asynchronous differential evolution with convex mutation; pp. 915–928. [[Google Scholar](https://scholar.google.com/scholar_lookup?title=Proceedings+of+Fifth+International+Conference+on+Soft+Computing+for+Problem+Solving&author=+Vaishali&author=TK+Sharma&publication_year=2016&)]

43. Ilonen J, Kamarainen JK, Lampinen J. Differential evolution training algorithm for feed-forward neural networks. Neural Process Lett. 2003;17(1):93–105. doi: 10.1023/A:1022995128597. [[CrossRef](https://dx.doi.org/10.1023%2FA%3A1022995128597" \t "_blank)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Neural+Process+Lett&title=Differential+evolution+training+algorithm+for+feed-forward+neural+networks&author=J+Ilonen&author=JK+Kamarainen&author=J+Lampinen&volume=17&issue=1&publication_year=2003&pages=93-105&doi=10.1023/A:1022995128597&)]

44. Storn R (1996) On the usage of differential evolution for function optimization. In: Fuzzy Information Processing Society, 1996. NAFIPS. Biennial Conference of the North American, pp 519–523. IEEE

45. Hancer E, Xue B, Zhang M. Differential evolution for filter feature selection based on information theory and feature ranking. Knowl-Based Syst. 2018;140:103–119. doi: 10.1016/j.knosys.2017.10.028. [[CrossRef](https://dx.doi.org/10.1016%2Fj.knosys.2017.10.028" \t "_blank)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Knowl-Based+Syst&title=Differential+evolution+for+filter+feature+selection+based+on+information+theory+and+feature+ranking&author=E+Hancer&author=B+Xue&author=M+Zhang&volume=140&publication_year=2018&pages=103-119&doi=10.1016/j.knosys.2017.10.028&)]

46. Kaur M, Kumar V, Li L. Color image encryption approach based on memetic differential evolution. Neural Comput & Applic. 2019;31(11):7975–7987. doi: 10.1007/s00521-018-3642-7. [[CrossRef](https://dx.doi.org/10.1007%2Fs00521-018-3642-7" \t "_blank)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Neural+Comput+&+Applic&title=Color+image+encryption+approach+based+on+memetic+differential+evolution&author=M+Kaur&author=V+Kumar&author=L+Li&volume=31&issue=11&publication_year=2019&pages=7975-7987&doi=10.1007/s00521-018-3642-7&)]