U.Porto Journal of Engineering, 9:3 (2023) 140-157
ISSN 2183-6493
DOI: 10.24840/2183-6493_009-003_001636

Received: 4 June, 2022
Accepted: 10 September, 2022
Published: 28 April, 2023

IsoCore – An efficient model to aid rapid forecasting of SARS-CoV-2 infection from biomedical imagery

Faraz Bagwan[1], Nitin Pise[2]

[1]Research scholar, School of Computer Engineering and Technology, Dr. Vishwanath Karad MIT World Peace University, Pune, India. (farazbagwan11@gmail.com) ORCID 0000-0002-0018-9590

[2]Professor, School of Computer Engineering and Technology, Dr. Vishwanath Karad MIT World Peace University, Pune (nitin.pise@mitwpu.edu.in) ORCID 0000-0001-5496-3038

Abstract

Combating the covid19 scourge is a prime concern for the human race today. Rapid diagnosis is critical to identify the infection accurately. Due to the prevalence of public health crisis, reaction-based blood tests are the customary approach for identifying covid19. As a result, scientists are analyzing screening methods like deep layered machine learning on chest radiographs. Despite their usefulness, these approaches have large computational costs, rendering them unworkable in practice. This study's main goal is to establish an accurate yet efficient method for predicting SARS-CoV-2 infection (Severe Acute Respiratory Syndrome CoronaVirus 2) using chest radiography pictures. We utilized and enhanced the graph-based family of neural networks to achieve the stated goal. The IsoCore algorithm is trained on a collection of X-ray images separated into four categories: healthy, Covid19, viral pneumonia, and bacterial pneumonia. The IsoCore model has 5 to 10 times fewer parameters than the other tested designs. It attains an overall accuracy of 99.79%. We believe the acquired results are the most ideal in the deep inference domain at this time. This proposed model might be employed by doctors via phones.

Author Keywords. Covid19, Deep Learning, IsoCore, Pneumonia, Chest Radiography

Type: Research Article
Open Access Peer Reviewed CC BY

1. Introduction

The Sars Cov2 pathogen is responsible for the covid19 infection, which may induce flu-like symptoms before advancing to a respiratory disorder. The graveness of the disease prompted a worldwide public health campaign to curb the advancement of the virus by detecting it early (A. H. AU Davarpanah et al. 2020). An efficient and conclusive distinction of covid19 can be made by using reactionary blood testing methods like the polymerase tests. However, such tests are time consuming and sensitive to the environment leading to high chances of incorrect diagnosis. Factors such as inadequate cell material in the sample or improper filtration methods are observed to interact with these tests (J. d. A. B. Araujo-Filho et al. 2020). Consequently, numerous negative tests are required to conclude the absence of COVID-19 infection in a person, which may lead to a scarcity of test kits (A. C. of Radiology et al. 2020). Chest X-Rays (CXR) are becoming more important as COVID-19 spreads across the globe (P. Huang et al. 2020; T. Ai et al. 2020; M. Y. Ng et al. 2020). An extensive body of evidence proves the ability of deep layered automatic learning networks to predict existence of covid19 by utilizing chest X-rays of patients. To properly identify covid19 infection, radiographic competence is essential to understand the nitty-gritties. As a result of the lack of expert radiologists pertaining to the chest region, particularly in developing countries, it is difficult to provide accurate interpretations of complicated chest exams. Techniques deeply influenced by the brain's anatomy like the artificial neural networks promotes the detection of primary patterns. Experimentation with neural networks especially, convolving networks have shown superior performance in many processing applications (Y. LeCun et al. 2015; H. Touvron et al. 2019; P. Rajpurkar et al. 2017), thus making deep learning methods an appropriate choice for the processing of x-rays. Pneumonia along with other illnesses have previously been detected and classified using deep learning (P. Rajpurkar et al. 2017; X. Wang et al. 2017; A. K. Jaiswal et al. 2019). Deep layered architectures possess substantive potential for identifying hidden minuscule infected regions from lung X-rays pictures. Low computational cost models are in demand because they enable the usage of input pictures with significantly greater resolutions without increasing processing time. Graph Neural Network (GNN) (Scarselli F. et al. 2009) is among the inference models that are friendly for developing lite applications with low hardware resource utilization thus making it compatible with different smart phone systems. This goal necessitates that the models have a small footprint and fast inference time, which means that they can be incorporated with smart devices extensively, thereby allowing it to be used seamlessly by individuals and also at hospitals. Moreover, GNN also deals with the overfitting and class imbalance problem which is difficult to attain with Convolutional networks. Experimental findings in other domain show that GNN-based models are typically fast (Mondal R. et al. 2019). Multi layered deep architectures like GNN are renowned for solving graph categorization challenges. As a result, the graph data structure is required for GNN input data. On the other hand, the input to every 2D-CNN (Two Dimensional - Convolution Neural Metwork) model is just a 2D picture matrix. A good method is needed to transfer picture classification to a graph classification. In order to turn a picture into graph data, we used an appropriate pre-processing approach. For this reason, we have built our suggested IsoCore (a Graph Isomorphic Network – GIN (Xu K. et al. 2018) based model) named IsoCore, which is a specific category of GNN.

Experimental data shows that our suggested model works very well in terms of time consumed by the model. Because of the one-to-one mapping characteristic of the aggregate functionality, our design has also performed well in datasets with a large class imbalance. The design is able to appropriately transfer various graphs into various embedding space representations. As a result, the suggested model is capable of accurately classifying images with a lower count. A total of four publicly accessible datasets were used into our analysis: the SARS Cov2 CT-scan dataset (Soares E. et al. 2020), the covid19 CT-scan dataset (Yang X. et al. 2003), the datasets from CMS-678-ML-Proj. (Jamdade V. et al. 2020) (3-4 class), and a mixture of the two datasets i.e., covid19 chest x-ray dataset accessible on Github (Cohen et al. 2020), and the pneumonia dataset of x-rays from Kaggle (Mooney P et al. 2018).

There are six parts to the rest of this article. In the next section, we'll look at some relevant literature. Section 3 explains the approach and dataset. Section 4, lays forth the proposed system i.e, Isocore model. An extensive collection of computational experiments is discussed in Section 5. Ideas for further study and conclusion are discussed in the final section.

2. Related Literature

In (E. E. D. Hemdan et al. 2020), a comparative study of seven renowned and conventional multi layered deep neural architectures was published to address Covid19. With just 50 photos belonging to normal category and covid19 infected individuals to work with, the researchers conducted their studies with a tiny dataset. Pre-trained models used the ImageNet dataset (a collection of radiographic images), which contained above 12 million pictures was trained. The VGG19 (K. Simonyan et al. 2019) (Visual Geometry Group19) and DenseNET201 (Y. Huang et al. 2019) designs performed the best in their tests.

CXR pictures are classified in (L. Wang et al. 2003) belonging to 3 classes: healthy, pneumonia-viral, and Covid19 are classified using a novel CNN architecture called COVID-net. There are 182 covid19 patients and over 13 thousand X-ray scans belong to healthy cases The total quantity of residing images in the data is just below 13 thousand, which is much greater than the prior study's dataset. For Covid19 detection, the researchers claim an overall accurate rate of 92.4% and the model's ability to identify the covid19 infection was 89.50%. For the four-categorization issue of radiographic chest dataset divided into healthy, covid19, pneumonia-bacterial, and pneumonia-viral pneumonia, the ResNet50 (Elaziz M. A. et al. 2020) is fine-tuned in (M. Farooq et al. 2003). Compared to COVID-net model, this architecture accomplished an astounding 97.10 percent on comprehensive accuracy and was proved to be 100% sensitive in detecting the covid19 infection. Although (M. Farooq et al. 2003) includes an additional class, it is vital to point out that the dataset it uses is a subset of [19's dataset]. Radiographs of 68 Covid19 infected cases, around twelve hundred normal images, over nine hundred pneumonia-bacterial scans and over 650 patients with no Covid19 but showing patterns of pneumonia- viral are included in the dataset in (M. Farooq et al. 2003).

They also used hierarchical analysis for COVID-19 pattern detection on CXR (Chest Xray) images (R. M. Pereira et al. 2022). Datasets from various public datasets were combined to create one with 1,144 X-ray pictures. Only 90% of the photos belonged to the COVID-19 classification, with the rest belonging to five distinct forms of pneumonitis along with a normal ordinary category. A deep convolutional network approach was one of several tools utilized to extract information from the photos (Inception-V3 (C. Szegedy et al. 2016)). To classify data, the authors looked at classifiers of the likes of support vectors, supervised random forests, K nearest neighbors, multi layered neurons, and decision-making using trees, among other options. The COVID-19 class's F1-Score is reported to be 0.89. In spite of the significant connection to the current study, we note that a head-on comparative analysis is not feasible because of the distinct nature of data utilized in both the studies.

An algorithm known as CoroNet, developed by the authors in (A. I. Khan et al. 2020), automates the identification of corona infection from x-rays. (F. Chollet et al. 2017) This particular network is built and used to process the ImageNet datastore using the Xception CNN architecture (J. Deng et al. 2009). There are two public image datasets (covid19 chest x-rays kaggle pneumonia viral chest x-rays) from which CoroNet was trained and tested. With a four-layered cross-validation technique, CoroNet architecture obtained an accuracy of 89.7 percent for covid19 cases, supporting precision and recall rates of 92.98 percent and 97.90 percent for four categories (covid19 bacterial versus Pneumonitis viral vs healthy). In addition, the authors tested the model in question, with yet another dataset, even when the dataset seems to include the previously used covid19 pictures for processing.

3. Materials and Methods

In this part, the datasets employed are discussed in detail. The foundational approach based on Isomorphic Networks is further explained in this section.

3.1. Dataset description

The research in this paper was carried out using the four datasets listed below.

1. The "covid19 ct-scan data" (Soares E. et al. 2020), developed and maintained on a data platform named kaggle.[31]

2. Yang et al., "SARS cov2 dataset," a 2-category dataset available on Kaggle. (Yang X. et al. 2003)
3. Three-class chest X-ray pictures used from two distinct repositories
   i. "Covid19-chestxray-dataset", (Cohen et al. 2020), present on a source control management platform, named Github.
   ii. Radiographic pictures of the lungs (pneumonia) collected by Mooney and colleagues in the "Covid-chestxray-dataset," (Mooney P et al. 2018)
4. CXR datasets for three and four classes of the "CMSC 678 ML Project" are available on GitHub (Jamdade V. et al. 2020).

All the datasets have two classes in common i.e., COVID19 and normal, whereas one dataset (Covid19-Chest-X-ray-dataset) has an additional Pneumonia class. The CMSC-678-ML Project's four-class dataset categorizes pneumonia into two subcategories: bacterial and viral. The Computed tomographic and radiographic pictures of the chest are shown in (Figure 1). (Table 1) displays the characteristics of each of these collections. (Figure 1) depicts sample images from the datasets used as input to the model.

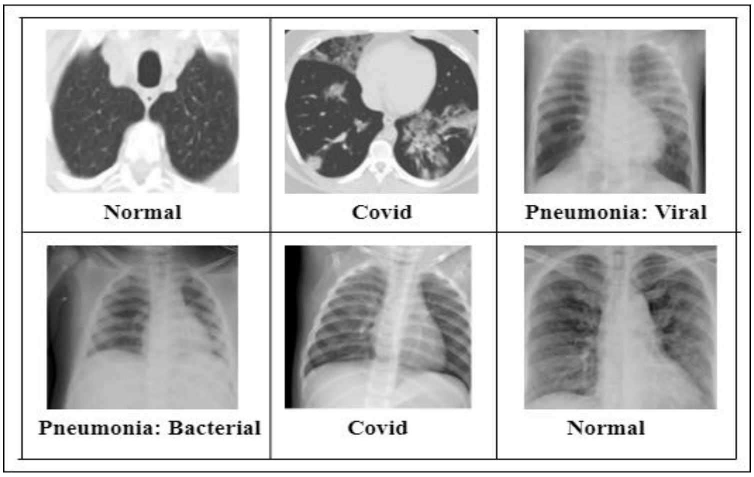


Figure 1: Sample CXR images and CT scans from the dataset.

Dataset Used	No. of classes	Image type	Total images		
			Healthy	Covid19	Pneumonia
Covid19-CT-data	2	CT	407	348	-
SARS-Cov2-CT-data	2	CT	1229	1252	-
Covid19 + Pneumonia-Chest-X-ray-data	3	CXR	1593	503	4343
CMS_678_ML-Proj. data	3	CXR	79	69	79
	4	CXR	79	69	Viral-79, Bacterial-79

Table 1: Quantitative overview of all experimental datasets.

3.2. Input Preprocessing

The Image cleaning, noise reduction, outlier removal, and other pre-processing procedures are all possible options to keep your images clean and clear. Pre-processing for the IsoCore model is limited to edge-tracking and graph-construction.

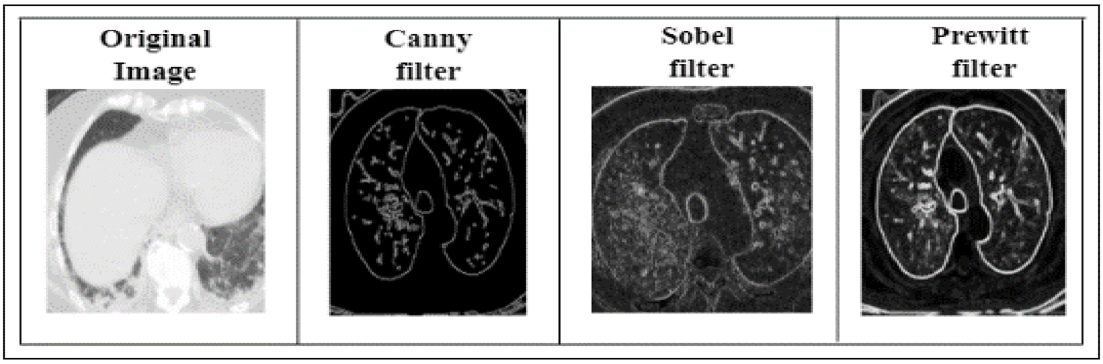
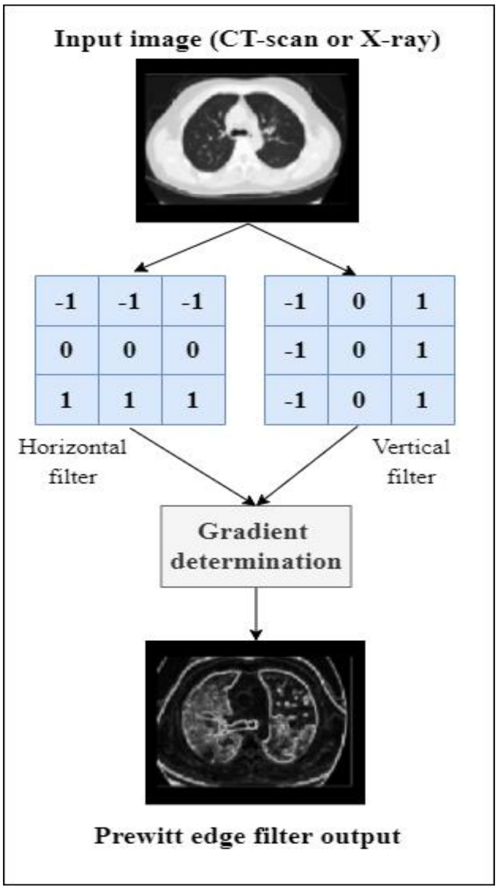


Figure 2: Edge-tracking using different image-filtering techniques

A. **Edge-tracking**: If an image's edge indicates where the intensity changes locally, then the image's edge area will have a local maximum value or a minimum value in terms of intensity change. In order to make the edges stand out, the source picture must be processed properly. For both horizontal and vertical edge detection, we used a 3*3 Prewitt filter (Prewitt et al. 1970) to concatenate the original picture matrix. Because it is simple to construct and finds edges relatively efficiently (Priyam P. et al. 2016), we have chosen the Prewitt operator for this experiment. Canny, Sobel, and Prewitt are illustrated in (Figure 2) as a comparison of the three most common edge filters. (Figure 2) shows that the picture produced by the Sobel filter is the noisiest, while the image produced by the Canny filter is the least noisy. Prewitt produces a noisier picture than Canny, but unlike Canny, all edges of the image have varying pixel intensities. For Prewitt filter, it would be better to use pixel values as a feature.



$$\begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

Figure 3: Edge-tracking process.

The gradient of each 3x3 sub-matrix has been calculated after convolution with both horizontal and vertical filters. All pictures are rendered in black and white, hence we assumed that each pixel in an edge had to be more than or equal to 128 in gradient magnitude for it to be called an edge. (Figure 3) provides a clearer picture of the edge-tracking procedure.

B. **Graph-construction**: Using the necessary procedures, the edge maps are turned into a graph dataset. Using the Prewitt filter (Prewitt et al. 1970), each image is turned into a graph. a Preparation of the graphs involves the following three steps:

1. Grayscale intensity values are greater than or equal to 128 for nodes and vertex points. Nodes appear to be restricted to the most visible edges of the edge picture, based on this finding. The feature of a node is the pixel's grayscale intensity.

2. Every unit from the original picture correlates to an exact neighboring vertex.

3. Every one of the photos are accompanied by a corresponding graph. In other words, all of the nodes and edges created from a single image are all part of the same network.

Only grayscale values are used for graph-based normalization of node attributes. The next step is to normalize the data by subtracting the original value from the average of all variables presented on an undirected graph and dividing the result by the standard deviation. Preparation of such data requires less memory since nodes are created just from edges present in a picture, rather than the complete picture. For Pneumonia scans like COVID19, the margins of the graph and the structure of the graph will be changed because of a cloudy region for coughing. This discrepancy in classification may prove useful at a later time. We constructed five datasets to constitute the equivalent graphical data from all filters:

1. The *node-set* contains the values of each node attribute (here, the normalized gray-scale intensity).

2. The *graph-data-id* comprises a collection of the node IDs.

3. *Set-of-node-labels* contains the class label information for every single node. The nodes in the same graph will have the same label because this classification is at the graph level, in actuality it is the class label of the related graph.

4. Each node in the same graph will have the same label because this classification is at the graph level, which is really the class label of the related graph. There is a class label for each graph in the *graph-set*.

5. For all graphs, an *Adjacency-Set* is included. The adjacency sparse-matrices of all the graphs are kept here.

(Figure 4) shows the whole graph construction process.

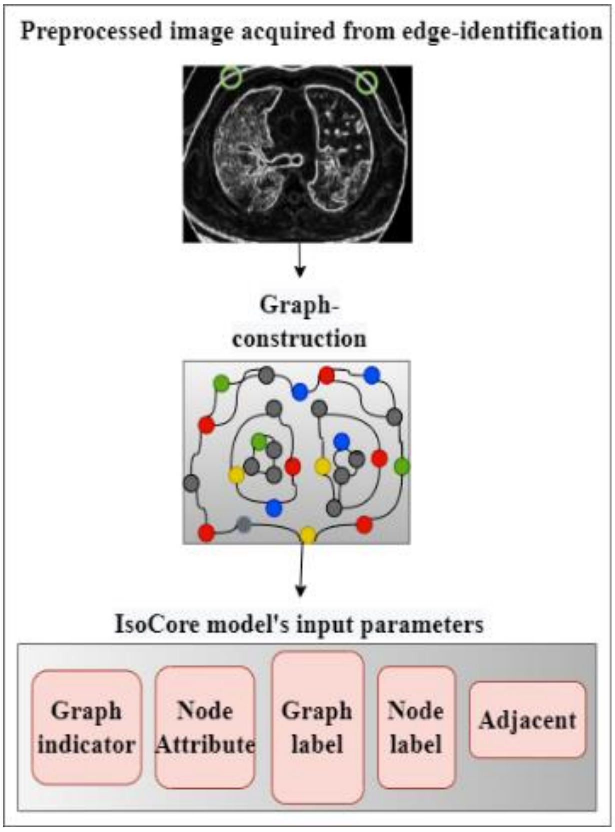

Preprocessed image acquired from edge-identification
Graph-construction
IsoCore model's input parameters
Graph indicator
Node Attribute
Graph label
Node label
Adjacent

Figure 4: Graph construction process

3.3. *Graph Neural Network (GNN) and Graph Isomorphic Network (GIN)*

A. **GNN**: The collection of vertices *V* and edges *E* that make up a graph *G* may be used to characterize the graph. Graph neural network is an algorithm that works directly with the graph structure. Node categorization is a common use of GNN. Predicting the labels of nodes in the network without relying on any ground truth is the ultimate goal. Each node v is described by its characteristic *x_v* and paired with a ground-truth label *t_v* in the node classification problem setting. To forecast the labels of the unlabeled nodes in a partly labelled graph *G*, the objective is to use the labelled nodes. To represent each node, it builds a *d*-dimensional vector (state) *h_v* that includes data about its immediate surroundings.

$$H_v = f(x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]}) \tag{1}$$

In (Formula 1), $h_{ne[v]}$ signifies the embedding of the nearby nodes of *v*, whereas $x_{ne[v]}$ provides the characteristics of *v*'s immediate neighbours. Inputs are transformed into *d*-dimensional space using the transition function. Message forwarding or neighbourhood aggregation are two actions that are then performed.

B. **GIN:** Graphs encountered in real-world applications are a mix of continuous and discrete structures (node and edge-features and connectedness, respectively). GNNs may be able to discriminate between various kinds of graph structures, according to one theory. The graph isomorphism issue, a classic in graph theory, asks whether two graphs are topologically similar. The sole difference between two isomorphic networks is the order in which their nodes are arranged. The graph isomorphism issue was thought to have a polynomial-time solution in the form of the Weisfeiler-Lehman (WL) test. Hashing is used at each phase to create unique new-nodes from the nodes that have been aggregated from their neighbours. Upon obtaining node stability, the algorithm comes to an end. WL, on the other hand, was unable to tell the difference between basic graph configurations. In an attempt to produce neural networks comparable to the WL method, Xu developed Graph Isomorphism Networks, which comprised of aggregate and update functions.

4. Proposed System

GIN is utilized to train the model to recognize Covid19 from chest x-rays or ct-scans using an innovative approach called IsoCore. As a result, the embedding space for a graph in GNN is represented by the symbol h_v, which is dimensioned by d. Embedded space places nodes that are connected to one another or that have common neighbors in close proximity. A feature vector-point f_{vec} and a neighborhood embedding vector-point h_{necv} are used to determine the embedding vector-point h_{vec} for each node. GNNs are used to learn a node's representation vector, which incorporates the graph's topology and the properties of its nodes. Nodes have their own neural network architecture (Scarselli F. et al. 2009) determined by the context in which they operate. This is seen in (Figure 5).

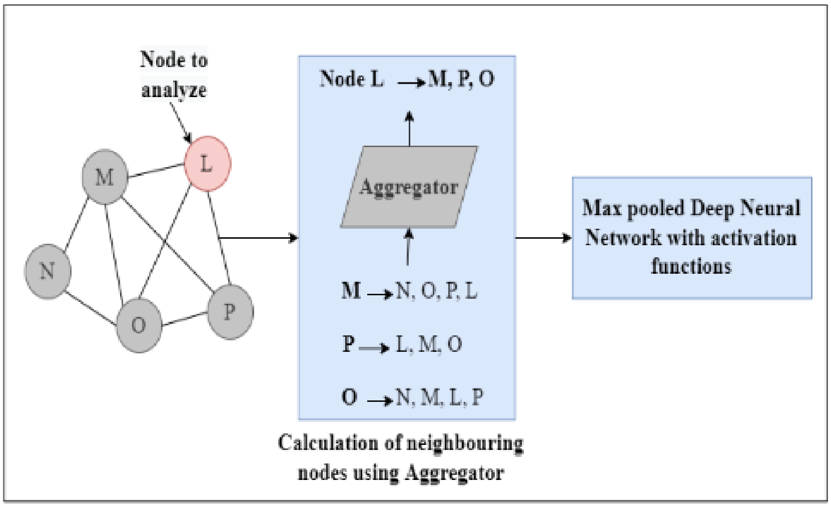

Node to analyze
L
M
N
O
P
Node L →M, P, O
Aggregator
M →N, O, P, L
P → L, M, O
O →N, M, L, P
Calculation of neighbouring nodes using Aggregator
Max pooled Deep Neural Network with activation functions

Figure 5: Graph processing

There is no restriction on the model length for each node. By continually updating the representation of each node, the knowledge about its near neighbors is constantly being added to its representation. All node's layers have their own unique embeddings. GNN's Aggregation algorithm iterations supply the k^{th} layer of GNN with the following.

$$a_{vec}^{k} = Aggregate^{(k)}[h_u^{(k-1)} : u \, \epsilon \, N(vec)], Combine^{(k)} \, (h_{vec}^{(k-1)}, a_{vec}^{k}) \qquad \textbf{(2)}$$

In (Formula 2), $h_v^{(k)}$ is the feature vector-point of node *vec* at the K^{th} level and $h_a^{(k)}$ is the averaged-out message from its neighbors $N_{(vec)}$ is the name given to the group of nodes that surround *vec*. Graph Convolutionalized Networks (GCN) (Berg et al. 2017) use element basis mean pooling as opposed to aggregation and combination processes.

A GINConv layer leverages Multi-Layer Perception in our design (MLP). The collection of neighborhood data will be handled by MLP in the future stages of our design. In the MLP, a second linearity layer follows linear rectified unit i.e., ReLU layer. The GIN convolutional MLP design is shown in (Figure 6)

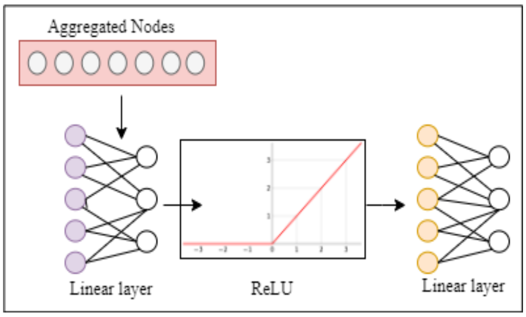


Figure 6: Multi layered perceptron network

The GIN convolutions accept two primary inputs.

1. Vertices from the graph have dimensions *V*dim*, which is equal to the entire number of vertices in the graph multiplied by *dim*, which gives us an initial feature matrix, *x*.
2. There are *L* vertices in the graph, with *v1* and *v2* connected by a single edge. The edge point *E* has a size of *2*L*, and this is its dimension.

Non-linearity is added to the GINConv layer's output after a dropout of 0.5 and a normalization (norm) layer are applied to the mini-batch of inputs. In another block, the output of another GINConv ReLU-drop-out-norm layer, *out2*, is supplied (*out1*). See (Figure 8). A global mean pooling layer is then placed on the cover of GINConv-ReLU-dropout layers that have been applied to this *out2*. After a 50% dropout rate, the next two layers have dropout rates that are half as high as the previous layer's dropout rate, and so on. Finally, it was decided to use a Log SoftMax to construct the final probability vector *(z).* We used the negative log likelihood function to classify the data in this study.

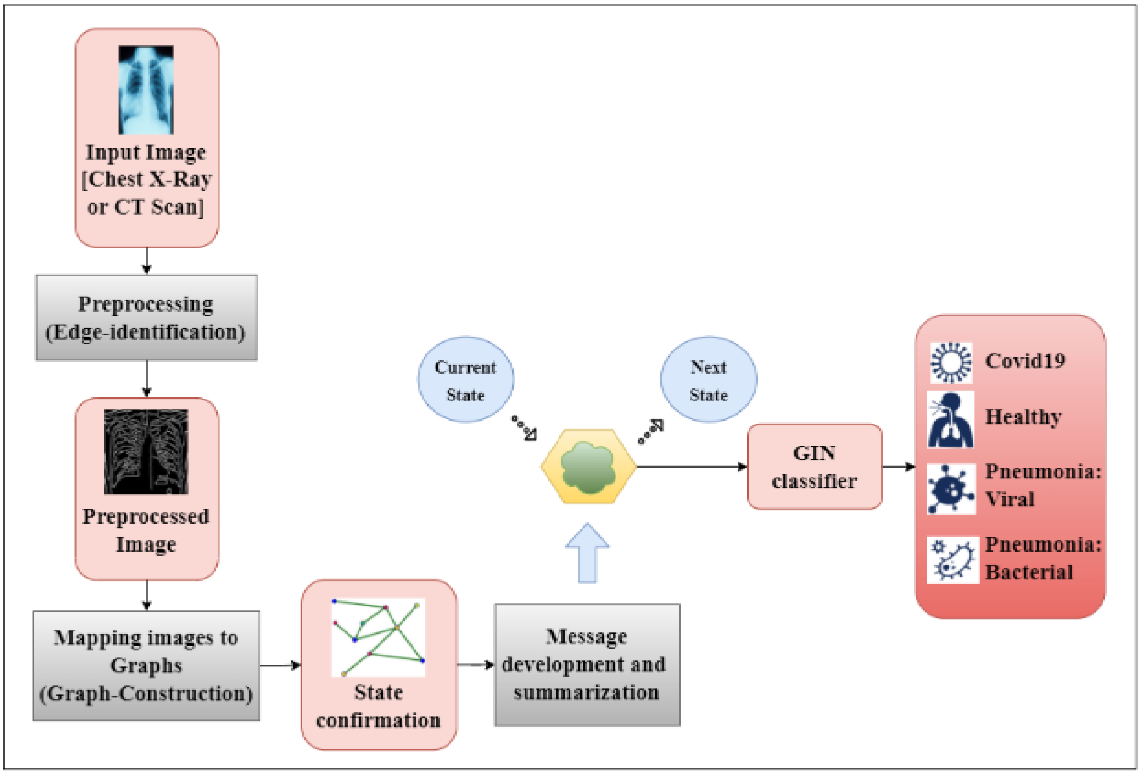

Input Image [Chest X-Ray or CT Scan]
Preprocessing (Edge-identification)
Preprocessed Image
Mapping images to Graphs (Graph-Construction)
State confirmation
Message development and summarization
Current State
Next State
GIN classifier
Covid19
Healthy
Pneumonia: Viral
Pneumonia: Bacterial

Figure 7: Proposed System i.e., The IsoCore model architecture

(Figure 7) shows a diagrammatic representation of the approach we proposed i.e., the Isocore model

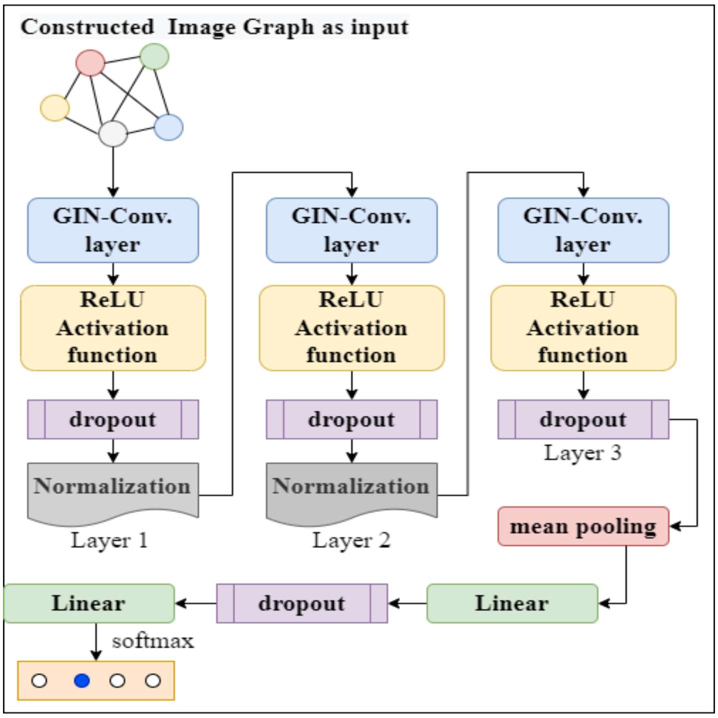


Figure 8: Proposed System internal layering

5. Results

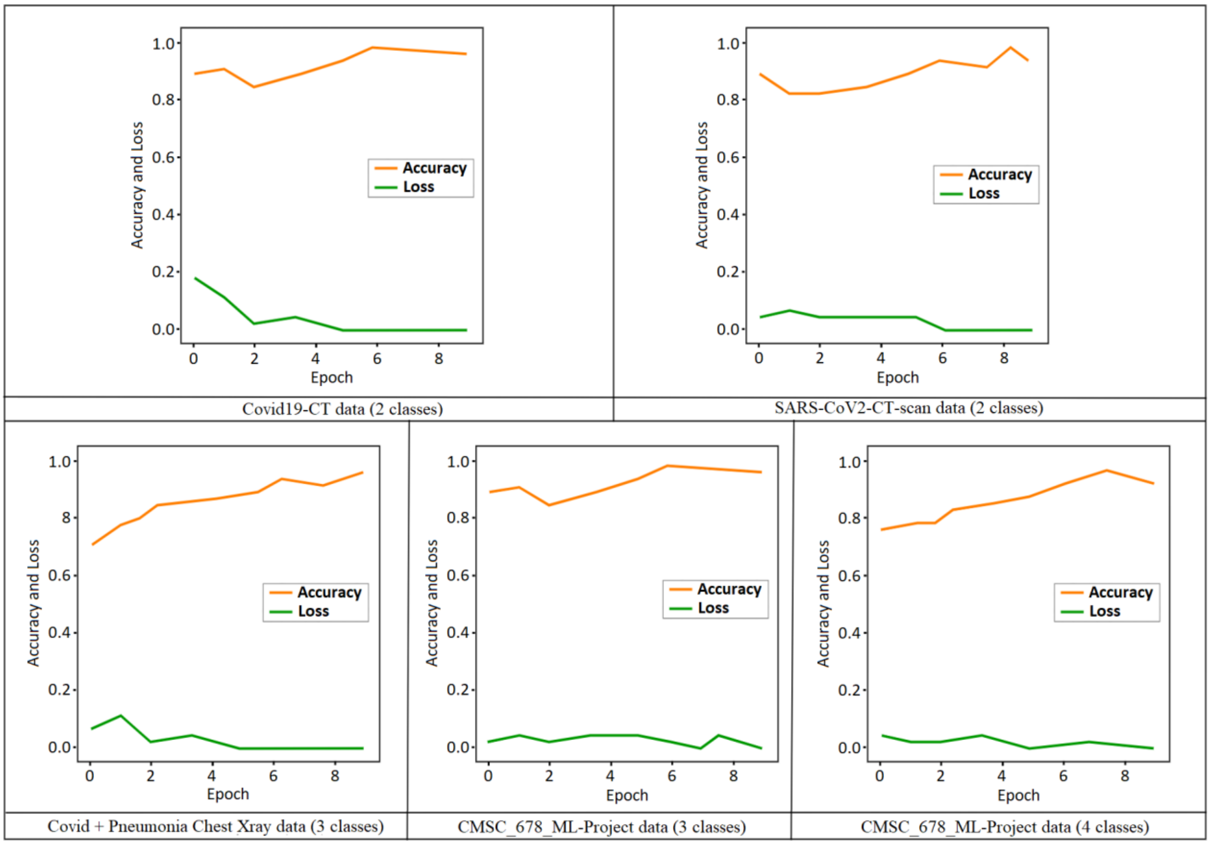
We utilized a five timed cross validation in our experiments to test the network's accuracy. There are ten training cycles for each fold. A Stochastic Gradient Descent (SGD) algorithm with a processing percentage of 0.01 was utilized to train our network, coupled with Adam's optimizer. According to established metrics, including Accuracy and the ROC curve, our model has performed well. (Table 2) depicts the results of our proposed IsoCore network performance along with the mean time spent on training and testing in each iteration for all four datasets.

Name of data set	Class	Accuracy	Training time in secs	Testing time in secs
Covid19-CT data	2	99.51	342.586	2.328
SARS-Cov2-CT data	2	99.45	146.365	1.151
Covid + Pneumonia Chest Xray data	3	99.55	971.29	7.138
CMS-678-ML-Proj. data	3	99.84	66.923	0.6
	4	99.79	73.697	0.612

Table 2: Assessment measures

For all datasets, the IsoCore model's accuracy is demonstrated in (Table 2) to be at least 99%, with a maximum accuracy of 99.84% for the 3-class dataset. According to our model, as the quantity of classes grows, the efficiency of the model's predictions drops from 99.85% to 99%. To add to the impressiveness of our suggested model, it accurately identified 99.84% of the covid19 and Pneumonia database patients in a significantly class-unbalanced data collection of patients. It is intuitively obvious that GNNs may map two nodes to the same place if the respective nodes have similar sub branches possessing the same properties. Subtree structures are built recursively using node neighbors. All that has to be decided is whether or not a GNN tracks two zones i.e., two multisets, to the same rendering or representation. GNNs cannot be used to map a single representation to numerous neighborhoods, or multiple sets of feature vectors. Injective aggregation is thus required. As a consequence, a sophisticated GNN aggregation approach can express injective multi-set functions.

The proposed IsoCore model contains a GIN network that can transform any two unique graphs into a variety of embeddings in order to solve the difficult graph isomorphism problem. Non-isomorphic and non-isomorphic graphs must be represented in different ways when it comes to isomorphic graphs It is possible to use this strategy to datasets that have a significant degree of class imbalance due to the following reasons: For both training and testing, (Table 2) data demonstrates that our proposed approach needs just 1–10 minutes. The fact that there are fewer epochs might potentially be a factor in the decreased training time. While this is true, as previously said, the initial loss of training is quite little. As a consequence, a large number of epochs is not required for training purposes. Fig. 9 shows that the training loss is rather small.



Covid19-CT data (2 classes)

SARS-CoV2-CT-scan data (2 classes)

Covid + Pneumonia Chest Xray data (3 classes)

CMSC_678_ML-Project data (3 classes)

CMSC_678_ML-Project data (4 classes)

Figure 9: Accuracy and loss

First epoch results in 99.90% of datasets being properly recognized with a loss of only 0.3 per dataset, as seen in the graph. To top it all off, when the period size increases, the classification accuracy either remains the same or improves somewhat. However, as can be seen in (Figure 9), accuracy seems to be constant due to the loss change being more apparent than the overall accuracy change. Because of appropriate pre-processing methods, the proposed network possesses the ability to completely comprehend the input graphs. Thus, training may be completed within 12 epochs, with a low initial loss. Each of the datasets portrayed in (Figure 10) was processed through a renowned characteristics operator curve also known as the ROC curve so that we could verify the correctness of our classification model. There is a 11% discrepancy between the two levels of instruction and testing in our tests. As an added visual aid, (Figure 11) displays the training and testing performance in terms of accuracy rate in relation to training to testing ratio for every dataset.

Training data	Testing data	Classes	Accuracy	Recall
Covid19-CT-data	SARS-Cov2-CT-data	2	99.51	99.42
SARS-Cov2-CT-data	Covid19-CT-data	2	99.45	99.40
Covid + Pneumonia Chest X-ray-data	CMS-678-ML-Proj.data	3	99.55	99.38
CMS-678-ML-Proj.data	Covid + Pneumonia Chest X-ray-data	3	99.84	99.45

Table 3: Model performance train-test figures, in detail, with respect to accuracy and recall

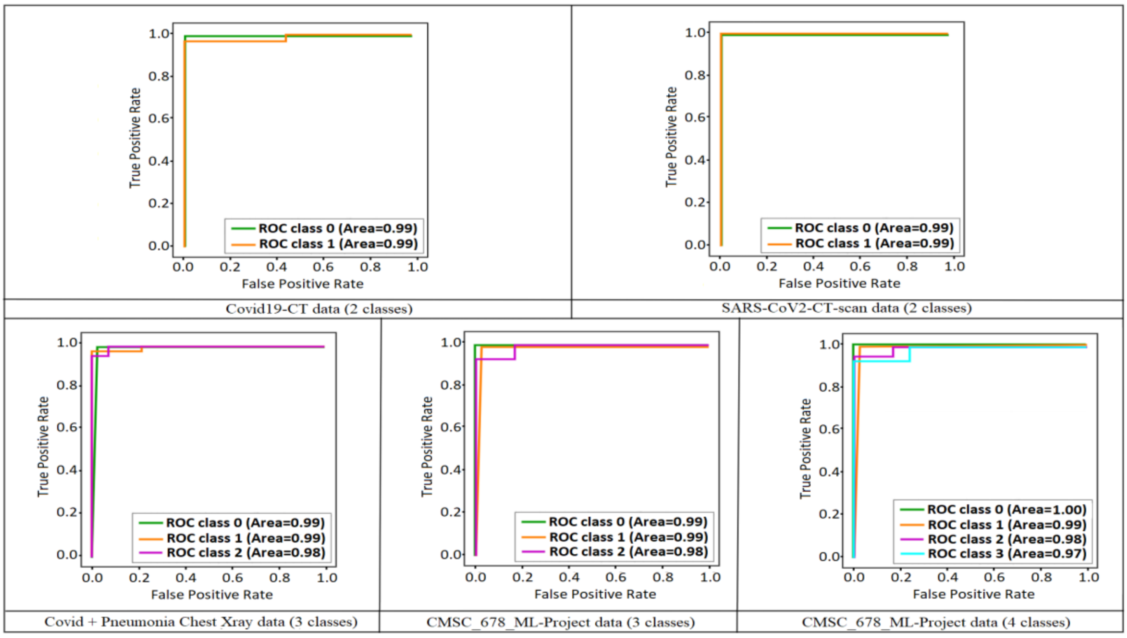


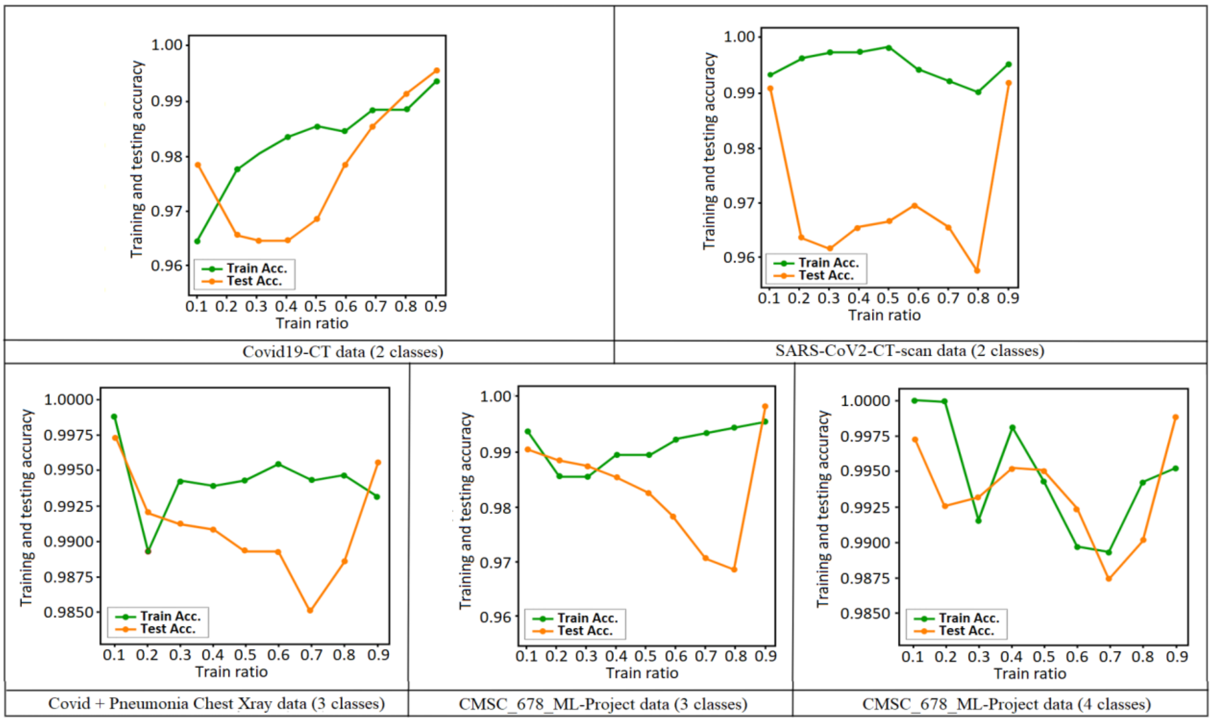
Figure 10: ROC and AUC

At least 96% of the samples are correctly predicted for all training-to-testing ratios in (Figure 11), which demonstrates the robustness of the IsoCore model. Area covered under the arch for every one of the ROCs in (Figure 10) is at least 0.98 units, demonstrating that the classifier is competent.

One can notice from their perfect ROC curves, the AUCs for both 2-class datasets are 0.990. With the IsoCore paradigm, any dataset of two classes may be handled. We conducted trials on datasets with the same quantity of categories for all training and testing process. Outcomes of all combinations of train and test are shown in (Table 3)

Despite the fact that the training and testing images originate from two distinct sources, the suggested model's accuracy is over 99 percent, as shown in (Table 5). Using this information, IsoCore's credibility has been boosted significantly. Pre-trained CNN models like VGG19 (Simonyan K. et al. 2014), ResNet152 (He K. et al. 2016), DenseNet201 (Huang G. et al. 2017), Xception (Szegedy C.et al. 2019), and MobileNetV2 (Chollet F. et al. 2017) have also been compared to our proposed model to ensure that it is superior to our model in both raw and filtered edge-mapped images. Accuracy (percent) is provided in (Table 4) based on the CNN models.

(Tables 2) and (Table 5) indicate that IsoCore outperforms all of the typical CNN models, indicating the sturdiness of our model's construction. We've also compared our proposed IsoCore model's outputs to those of other research that employed the same datasets. The results of these comparisons are shown in (Table 5). That our technique is more accurate than any of the other methods reviewed here is evident in (Table 5).



Covid19-CT data (2 classes)

SARS-CoV2-CT-scan data (2 classes)

Covid + Pneumonia Chest Xray data (3 classes)

CMSC_678_ML-Project data (3 classes)

CMSC_678_ML-Project data (4 classes)

Figure 11: Training and testing accuracy

Our model, IsoCore, outperforms the previous ones, even though some of them were done on a different or even larger dataset. This dataset (Jamdade V. et al. 2020) has not been utilized in any previous study, to the extent of our knowledge. Very limited quantity of studies has looked at a 4-class covid19 classification database thus far.

Existing models	Covid + Pneumonia Chest Xray data		CMS-678-ML-Proj. (3 class data)		CMS-678-ML-Proj. (4 class data)		SARS-Cov2-CT data		Covid19-CT data	
	Image	Filter image	Image	Filter image	Image	Filter image	Image	Filter image	Image	Filter image
Xception	96.74	99.22	82.61	86.96	82.15	83.97	83.30	81.79	82.01	87.58
ResNet152	98.68	97.82	91.31	91.40	86.13	85.88	77.87	84.58	86.65	87.97
Inception_ResNet-V2	98.22	98.05	82.61	91.3	77.56	86.45	77.85	80.08	74.35	78.95
VGG19	98.45	96.50	86.96	97.83	79.65	92.2	78.27	82.55	79.60	84.27
MobileNet-V2	98.76	98.52	93.48	84.74	81.45	82.25	77.46	80.48	78.18	76.97
DenseNet-201	99.07	97.35	95.65	96.13	88.65	90.44	75.86	85.69	89.11	90.21

Table 4. Original image vs. equivalent filtered edge image accuracy comparison % for different models.

We were compelled to document the findings of the CMS-678-ML-Proj. GitHub dataset (Jamdade V. et al. 2020) analysis. In addition, any deep learning network, such as the CMS-678-ML-Proj. Github dataset (Jamdade V. et al. 2020), is typically difficult to accomplish exceptional accuracy with a limited number of input samples. (Table 2) shows that IsoCore properly forecasts with 99.84 percent and 99.79 percent performance rate for its 3-class and 4-class scenarios accordingly. Even when working with little datasets, the model we've constructed is able to perform effectively. Our proposed model, as compared to other models already in use, is very precise and robust.

Data used	Researchers	Model used	Accuracy	Precision	Recall	F1-score
Covid19-CT data	(Silva P. et al. 2020)	EfficientNet with transfer learning	98.99	99.20	98.80	99
	(Soares E. et al. 2020)	xDNN	97.38	99.16	95.53	97.31
	Proposed	IsoCore	99.51	99.39	99.42	99.45
SARS-Cov2-CT data	(Yang X. et al. 2003)	Segmentation masks with CSSL	89.1	–	–	89.6
	(Silva P. et al. 2020)	EfficientNet with transfer learning	87.68	93.98	79.59	86.19
	Proposed	IsoCore	99.45	99.25	99.40	99.38
Covid + Pneumonia Chest Xray data	(Zhong Y. et al. 2007)	VGG16 based CNN model	87.3	89.67	84.4	86.96
	(Oh Y. et al. 2020)	DenseNet103 for segmentation + ResNet19	88.9	83.4	85.9	84.4
	(Chandra T. et al. 2020)	SVM, KNN, Decision trees	93.41	–	–	–
	(Makris A. et al. 2020)	VGG16 and VGG19 with transfer learning	95.88	Covid-96 Normal-95 Pneumonia-95	Covid-96 Normal-100 Pneumonia-91	Covid-98 Normal-98 Pneumonia-98
	(Elaziz M. A. et al. 2020)	MRFO + KNN	96.09	98.75	98.75	98.75
	(Turkoglu M. et al. 2020)	Alex network with support vectors	99.18	99.48	99.13	99.30
	(Toğaçar M. et al. 2020)	CNN for feature extraction+ SVM	98.97	–	89.39	96.72
	(Hemdan E. et al. 2020)	VGG19 or DenseNet201	90	Covid-83 Normal-100	Covid-100 Normal-80	Covid-91 Normal-89
	Proposed	IsoCore	99.55	99.30	99.38	99.29

Table 5. Comparative assessment of IsoCore model with previous models

6. Conclusions

To identify any abnormalities produced by COVID-19 in chest radiography images, we adopt an effective Graph Neural Network architecture. It was tested on four distinct datasets, utilizing the graph Isomorphism technique to construct a classification model that performed well. In spite of the lack of COVID-19 photos in the datasets, model training was unaffected by the imbalance due to the graph-based structure of the IsoCore model. Prewitt filter[41], used in the preprocessing stage, is used to construct the picture's borders. Consequently, our proposed technique utilizes processing resources and space more efficiently as compared to standard deep CNN networks. The architecture was trained on 11 epochs. In terms of evaluation, the suggested method was found to be 99.79% accurate, with a computational efficiency more than 20 times more than the baseline effort. In our opinion, the present approach is a strong contender for incorporation into medical devices or even the cell phones of doctors. Physicians may use the IsoCore to help them rapidly find covid19 in radiographic pictures.

References

"ACR Recommendations for the Use of Chest Radiography and Computed Tomography (CT) for Suspected COVID-19 Infection." 2020. A. C. of Radiology. 2020. https://www.acr.org/Advocacy-and-Economics/ACR-Position-Statements/Recommendations-for-Chest-Radiography-and-CT-for-Suspected-COVID19-Infection.

Ai, Tao, Zhenlu Yang, Hongyan Hou, Chenao Zhan, Chong Chen, Wenzhi Lv, Qian Tao, Ziyong Sun, and Liming Xia. 2020. "Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases." Radiology 296 (2): E32–40. https://doi.org/10.1148/radiol.2020200642.

Angelov, Plamen, Eduardo Soares, Biaso, S., Higa Froes, M., and Kanda Abe, D. 2020. "SARS-CoV-2 CT-Scan Dataset: A Large Dataset of Real Patients CT Scans for SARS-CoV-2 Identification." Health Informatics. https://doi.org/10.1101/2020.04.24.20078584.

Arimateia Batista Araujo-Filho, Jose de, Marcio Valente Yamada Sawamura, André Nathan Costa, Giovanni Guido Cerri, and Cesar Higa Nomura. 2020. "COVID-19 Pneumonia: What Is the Role of Imaging in Diagnosis?" Jornal Brasileiro de Pneumologia 46 (2): e20200114–e20200114. https://doi.org/10.36416/1806-3756/e20200114.

Berg, Rianne van den, Thomas N. Kipf, and Max Welling. 2017. "Graph Convolutional Matrix Completion." https://doi.org/10.48550/ARXIV.1706.02263.

Chandra, Tej Bahadur, Kesari Verma, Bikesh Kumar Singh, Deepak Jain, and Satyabhuwan Singh Netam. 2021. "Coronavirus Disease (COVID-19) Detection in Chest X-Ray Images Using Majority Voting Based Classifier Ensemble." Expert Systems with Applications 165 (March): 113909. https://doi.org/10.1016/j.eswa.2020.113909.

Chollet, François. 2016. "Xception: Deep Learning with Depthwise Separable Convolutions." https://doi.org/10.48550/ARXIV.1610.02357.

Cohen, Joseph Paul, Paul Morrison, Lan Dao, Karsten Roth, Tim Q Duong, and Marzyeh Ghassemi. 2020. "COVID-19 Image Data Collection: Prospective Predictions Are the Future." https://doi.org/10.48550/ARXIV.2006.11988.

Davarpanah, Amir H., Arash Mahdavi, Ali Sabri, Taraneh Faghihi Langroudi, Shahram Kahkouee, Sara Haseli, Mohammad Ali Kazemi, et al. 2020. "Novel Screening and Triage Strategy in Iran During Deadly Coronavirus Disease 2019 (COVID-19) Epidemic: Value of Humanitarian Teleconsultation Service." Journal of the American College of Radiology 17 (6): 734–38. https://doi.org/10.1016/j.jacr.2020.03.015.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. "ImageNet: A Large-Scale Hierarchical Image Database." In 2009 IEEE Conference on Computer Vision and Pattern Recognition, 248–55. https://doi.org/10.1109/CVPR.2009.5206848.

Elaziz, Mohamed Abd, Khalid M. Hosny, Ahmad Salah, Mohamed M. Darwish, Songfeng Lu, and Ahmed T. Sahlol. 2020. "New Machine Learning Method for Image-Based Diagnosis of COVID-19." Edited by Robertas Damasevicius. PLOS ONE 15 (6): e0235187. https://doi.org/10.1371/journal.pone.0235187.

Farooq, Muhammad, and Abdul Hafeez. 2020. "COVID-ResNet: A Deep Learning Framework for Screening of COVID19 from Radiographs." https://doi.org/10.48550/ARXIV.2003.14395.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. "Deep Residual Learning for Image Recognition." https://doi.org/10.48550/ARXIV.1512.03385.

Hemdan, Ezz El-Din, Marwa A. Shouman, and Mohamed Esmail Karar. 2020. "COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images." https://doi.org/10.48550/ARXIV.2003.11055.

Huang, Gao, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2016. "Densely Connected Convolutional Networks." https://doi.org/10.48550/ARXIV.1608.06993.

Huang, Peikai, Tianzhu Liu, Lesheng Huang, Hailong Liu, Ming Lei, Wangdong Xu, Xiaolu Hu, Jun Chen, and Bo Liu. 2020. "Use of Chest CT in Combination with Negative RT-PCR Assay for the 2019 Novel Coronavirus but High Clinical Suspicion." Radiology 295 (1): 22–23. https://doi.org/10.1148/radiol.2020200330.

Huang, Yanping, Youlong Cheng, Ankur Bapna, Orhan Firat, Mia Xu Chen, Dehao Chen, HyoukJoong Lee, et al. 2018. "GPipe: Efficient Training of Giant Neural Networks Using Pipeline Parallelism." https://doi.org/10.48550/ARXIV.1811.06965.

Huijbregts, L. A., and M. Schreurs. 1975. "Evaluation of Amplitude and Frequency Changes of Miniature Potentials with a Poor Signal-to-Noise Ratio." Comparative Biochemistry and Physiology. C: Comparative Pharmacology 52 (1): 11–16. https://doi.org/10.1016/0306-4492(75)90005-2.

Jaiswal, Amit Kumar, Prayag Tiwari, Sachin Kumar, Deepak Gupta, Ashish Khanna, and Joel J.P.C. Rodrigues. 2019. "Identifying Pneumonia in Chest X-Rays: A Deep Learning Approach." Measurement 145 (October): 511–18. https://doi.org/10.1016/j.measurement.2019.05.076.

Jamdade, Vaishnavi. (2020) 2023. "CMSC-678-ML-Project." Jupyter Notebook. https://github.com/vj2050/Transfer-Learning-COVID-19.

Khan, Asif Iqbal, Junaid Latief Shah, and Mohammad Mudasir Bhat. 2020. "CoroNet: A Deep Neural Network for Detection and Diagnosis of COVID-19 from Chest x-Ray Images." Computer Methods and Programs in Biomedicine 196 (November): 105581. https://doi.org/10.1016/j.cmpb.2020.105581.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep Learning." Nature 521 (7553): 436–44. https://doi.org/10.1038/nature14539.

Makris, Antonios, Ioannis Kontopoulos, and Konstantinos Tserpes. 2020. "COVID-19 Detection from Chest X-Ray Images Using Deep Learning and Convolutional Neural Networks." Preprint. Radiology and Imaging. https://doi.org/10.1101/2020.05.22.20110817.

Mondal, Riktim, Debadyuti Mukherjee, Pawan Kumar Singh, Vikrant Bhateja, and Ram Sarkar. 2021. "A New Framework for Smartphone Sensor-Based Human Activity Recognition Using Graph Neural Network." IEEE Sensors Journal 21 (10): 11461–68. https://doi.org/10.1109/JSEN.2020.3015726.

Mooney, P. n.d. "Chest X-Ray Images (Pneumonia)." Accessed March 9, 2023. https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia.

Ng, Ming-Yen, Elaine Y. P. Lee, Jin Yang, Fangfang Yang, Xia Li, Hongxia Wang, Macy Mei-sze Lui, et al. 2020. "Imaging Profile of the COVID-19 Infection: Radiologic Findings and Literature Review." Radiology: Cardiothoracic Imaging 2 (1): e200034. https://doi.org/10.1148/ryct.2020200034.

Oh, Yujin, Sangjoon Park, and Jong Chul Ye. 2020. "Deep Learning COVID-19 Features on CXR Using Limited Training Data Sets." https://doi.org/10.48550/ARXIV.2004.05758.

Pereira, Rodolfo M., Diego Bertolini, Lucas O. Teixeira, Carlos N. Silla, and Yandre M.G. Costa. 2020. "COVID-19 Identification in Chest X-Ray Images on Flat and Hierarchical Classification

Scenarios." Computer Methods and Programs in Biomedicine 194 (October): 105532. https://doi.org/10.1016/j.cmpb.2020.105532.

Prewitt, J. M. S. 1970. Object Enhancement and Extraction. B. Lipkin and A. Rosenfeld, Eds. Picture Processing and Psychopictorics. New York: Academic Press.

Priyam, Diganta Dey, and Dipanjan Polley Shreya. 2016. "Edge Detection by Using Canny and Prewitt." International Journal of Scientific & Engineering Research 7 (4): 251–54. https://www.ijser.org/researchpaper/Edge-Detection-by-Using-Canny-and-Prewitt.pdf.

Rajpurkar, Pranav, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, et al. 2017. "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning." https://doi.org/10.48550/ARXIV.1711.05225.

Rohrmann, C. A., H. J. Ansel, R. L. Protell, F. E. Silverstein, S. E. Silvis, and J. A. Vennes. 1979. "Significance of the Nonopacified Gallbladder in Endoscopic Retrograde Cholangiography." AJR. American Journal of Roentgenology 132 (2): 191–95. https://doi.org/10.2214/ajr.132.2.191.

Scarselli, F., M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini. 2009. "The Graph Neural Network Model." IEEE Transactions on Neural Networks 20 (1): 61–80. https://doi.org/10.1109/TNN.2008.2005605.

Silva, Pedro, Eduardo Luz, Guilherme Silva, Gladston Moreira, Rodrigo Silva, Diego Lucio, and David Menotti. 2020. "COVID-19 Detection in CT Images with Deep Learning: A Voting-Based Scheme and Cross-Datasets Analysis." Informatics in Medicine Unlocked 20: 100427. https://doi.org/10.1016/j.imu.2020.100427.

Simonyan, Karen, and Andrew Zisserman. 2014. "Very Deep Convolutional Networks for Large-Scale Image Recognition." https://doi.org/10.48550/ARXIV.1409.1556.

Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. 2016. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning." https://doi.org/10.48550/ARXIV.1602.07261.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. "Rethinking the Inception Architecture for Computer Vision." https://doi.org/10.48550/ARXIV.1512.00567.

Toğaçar, Mesut, Burhan Ergen, and Zafer Cömert. 2020. "COVID-19 Detection Using Deep Learning Models to Exploit Social Mimic Optimization and Structured Chest X-Ray Images Using Fuzzy Color and Stacking Approaches." Computers in Biology and Medicine 121 (June): 103805. https://doi.org/10.1016/j.compbiomed.2020.103805.

Touvron, Hugo, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. 2019. "Fixing the Train-Test Resolution Discrepancy." https://doi.org/10.48550/ARXIV.1906.06423.

Turkoglu, Muammer. 2021. "COVIDetectioNet: COVID-19 Diagnosis System Based on X-Ray Images Using Features Selected from Pre-Learned Deep Features Ensemble." Applied Intelligence 51 (3): 1213–26. https://doi.org/10.1007/s10489-020-01888-w.

Wang, Linda, and Alexander Wong. 2020. "COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images." https://doi.org/10.48550/ARXIV.2003.09871.

Wang, Xiaosong, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. 2017. "ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases." In 2017

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3462–71. https://doi.org/10.1109/CVPR.2017.369.

Xu, Keyulu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. "How Powerful Are Graph Neural Networks?" https://doi.org/10.48550/ARXIV.1810.00826.

Yang, Xingyi, Xuehai He, Jinyu Zhao, Yichen Zhang, Shanghang Zhang, and Pengtao Xie. 2020. "COVID-CT-Dataset: A CT Scan Dataset about COVID-19." https://doi.org/10.48550/ARXIV.2003.13865.

Zhong, Yi. 2020. "Using Deep Convolutional Neural Networks to Diagnose COVID-19 From Chest X-Ray Images." https://doi.org/10.48550/ARXIV.2007.09695.