Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Graph Based COVID-19 Detection Using Conditional Generative Adversarial Network

IMRAN IHSAN[1], AZHAR IMRAN[1] TAHIR SHER[1], MAHMOOD BASIL A. AL-RAWI[2], MOHAMMED EL-MELIGY [3,4], MUHAMMAD SALMAN PATHAN[5],

[1]Department of Creative Technologies, Faculty of Computing & AI, Air University, Islamabad, 44000, Pakistan (e-mail: imranihsan@au.edu.pk; azhar.imran@au.edu.pk; tahir2011.iiui@gmail.com)
[2]Department of Optometry, College of Applied Medical Sciences, King Saud University, Riyadh, Saudi Arabia (e-mail: malrawi@ksu.edu.sa)
[3]Applied Science Research Center, Applied Science Private University, Amman, Jordan (e-mail: mohammedali2000@gmail.com)
[4]Jadara University Research Center, Jadara University, PO Box 733, Irbid, Jordan
[5]School of Computing, Dublin City University, Collins Ave Ext, Whitehall, Dublin 9, Ireland (e-mail: muhammad.salman@dcu.ie)

Corresponding author: Muhammad Salman Pathan (e-mail: muhammad.salman@dcu.ie)

The authors extend their appreciation to King Saud University for funding this work through Researchers Supporting Project number (RSP2024R378), King Saud University, Riyadh, Saudi Arabia.

ABSTRACT Coronavirus (SARS-CoV-2) is a novel global pandemic, which requires rapid and accurate identification techniques to curb its spread. COVID-19, the disease induced by the virus, causes severe respiratory complications, necessitating advanced diagnostic tools for early detection. Recent research indicates the potential of radiographic imaging in unravelling critical insights into the characteristics of this formidable pathogen. Leveraging the advancements in Computer Vision (CV) and deep learning methodologies, an automated system can be devised to discern respiratory anomalies from X-ray images, enhancing conventional diagnostic methods. In this study, we propose a pioneering approach for COVID-19 diagnosis utilizing chest radiographs. The proposed methodology encompasses four distinct phases: initial segmentation of raw chest radiographs employing Conditional Generative Adversarial Networks (CGAN), followed by feature extraction through a tailored pipeline integrating both manual computer vision algorithms and pre-trained Deep Neural Network (DNN) models. Subsequently, a graph-based feature reconstruction technique amalgamates these extracted features across the network, culminating in a comprehensive representation. These reconstructed features serve as input to a classification module, comprising a multi-layer neural network, GCN, adept at processing graph-structured data, alongside conventional machine learning classifiers such as Support Vector Machine (SVM), Extreme Gradient Boosting (XGBoost), and Random Forest (RF), facilitating categorization of chest X-ray images into COVID-19, pneumonia, and normal cases. Furthermore, we conduct an exhaustive evaluation of the selected DNN architectures to ascertain the efficacy of our proposed models vis-à-vis existing research, thus ensuring the deployment of the most robust diagnostic framework.

INDEX TERMS COVID-19, Image segmentation, C-GAN, Deep neural network (DNN), Key point extraction, Classification Models.

I. INTRODUCTION

During the ongoing global crisis caused by the highly contagious respiratory disease [1, 2] resulting from the CORONA virus, the importance of swift detection, accurate diagnosis, and effective treatment cannot be overstated. COVID-19, an ongoing pandemic for over four years now, continues to exact a toll, spreading rapidly and claiming millions of lives worldwide. The virus, initially identified in Wuhan, China, poses significant challenges due to its rapid transmission through direct contact and respiratory droplets. Experts emphasize the need for affordable, efficient diagnostics to curb its spread.

In light of the exponential surge in COVID-19 cases, both viral nucleic acid tests (VNATs) and imaging techniques have become indispensable for prompt diagnosis. While VNATs, particularly the RT-PCR test, are renowned for their sensitivity [3, 4], imaging techniques such as chest radiology imaging and CT scans offer alternative diagnostic avenues.

However, chest radiology, while practical, exhibits lower sensitivity compared to RT-PCR and CT scans, highlighting the need for enhanced diagnostic methodologies [5].

X-ray imaging emerges as a cost-effective alternative for COVID-19 detection [6], albeit with limitations in clinical relevance and interpretational challenges. To address these limitations and enhance diagnostic accuracy, the development of a computer-assisted approach for automatic detection and identification becomes imperative. Such a system would not only aid medical professionals in early disease detection but also assist radiologists in distinguishing COVID-19 from other respiratory conditions, thereby facilitating timely clinical decisions and treatment planning.

In alignment with the World Health Organization's recommendations [6, 7], this study aims to present a comprehensive methodology for image segmentation, feature extraction, and graph-based representation to enable precise categorization of chest X-ray images. The key contributions of this study include:

- Conducting a critical analysis of existing image segmentation techniques to identify their strengths and limitations in the context of COVID-19 detection.
- Proposing a robust deep learning-based feature extraction method that is effective, structured, and practical for identifying key features indicative of COVID-19 infection.
- Outlining a systematic approach for incorporating graph representation to facilitate feature reconstruction, considering the spatial relationships between features within the image.
- Developing a methodical strategy for feature extraction and picture categorization using the pertinent features identified, enabling accurate classification of chest X-ray images into distinct categories, including COVID-19, pneumonia, and normal cases.

By addressing these objectives, this research endeavors to contribute to the advancement of diagnostic approaches for COVID-19, aiding in its timely identification and containment.

II. LITERATURE REVIEW

Advancements in deep learning have significantly enhanced COVID-19 detection and analysis via medical imaging, particularly chest X-rays (CXR) and computed tomography (CT) scans. Convolutional Neural Networks (CNNs) such as COVID-Net, COVIDX-Net, and transfer learning models like MobileNetV2 have demonstrated high accuracies in binary and multi-class classifications. Generative Adversarial Networks (GANs) have emerged as a critical tool for augmenting datasets and improving diagnostic performance. Techniques integrating deep learning with GANs and hybrid frameworks, including federated and blockchain technologies, have addressed challenges like dataset limitations, image quality issues, and data privacy concerns. Studies reveal promising accuracies exceeding 90% across various models, with methodologies incorporating transfer learning, segmentation, and region-of-interest (ROI) prioritization, optimizing classification and resource efficiency. These advancements underscore the transformative potential of AI in COVID-19 diagnostics and the broader field of medical imaging. Following is the extensive literature review related to our study shown in Table 1.

III. METHODOLOGY

In every research project, the primary step involves identifying the problem, while the crucial step in improving upon previous work lies in proposing a method to address the identified problem. Extensive research has focused on utilizing images for COVID-19 detection. It's essential to acknowledge that implementing our idea effectively necessitates adhering to specific protocols for each method. The entire system development concept comprises six components, as illustrated in the process flowchart in Figure 1.

A. DATASETS

The initial stage involves selecting and preparing a dataset for future analysis. In this process, COVID-19 was distinguished from pneumonia and normal images using chest X-ray (CXR) images sourced from various freely available online resources. The datasets utilized to train segmentation and classification models in this framework are:

1) Segmentation Dataset

For training the segmentation network, the CXR images dataset for segmentation [34] was utilized. This dataset comprises frontal view posterior anterior (PA) chest images and facilitates the segmentation of anatomical parts like the lungs, heart, and clavicles.

2) Classification Dataset

The chest X-ray images represented in Figure 2 constitute the dataset used for categorization. We utilized a publicly accessible dataset consisting of COVID-19 chest X-ray (CXR) images provided by [35]. Additionally, another dataset comprised images of pneumonia and typical individual chest X-rays (CXR), known as the pneumonia dataset [36]. Similar to the segmentation dataset, all chest X-ray (CXR) images are frontal views.

B. IMAGE PRE-PROCESSING

Preprocessing, which involves applying various transformation techniques to enhance features and make the data more manageable for analysis, is a crucial step in ensuring the quality of a dataset.

1) Segmentation Dataset Pre-processing

The segmentation dataset underwent preprocessing by creating three folders: train, test, and validation, to divide the dataset. Lung masks (images) for the left and right lungs were stored in separate folders from the original source images.

TABLE 1: Literature Review of Deep Learning Techniques for COVID-19 Detection

Study	Method/Model	Dataset	Techniques/Features	Accuracy	Use Case	Limitations
[8]	General DL advancements	Radiography datasets	Deep learning algorithms	x	COVID-19 detection	Limited dataset details
[9, 10]	DL-based imaging	Radiography datasets	Treatment and monitoring	x	Aid for practitioners	Lacks model specifics
[11, 12]	CNNs for classification	Various datasets	CNN architectures	x	Pattern recognition	Unspecified metrics
[13]	DarkCovidNet	CXR images	Binary/multi-class classification	98.08% (binary)	Radiologist support	Dataset details missing
[14]	COVID-Net	CXR images	Multi-objective optimization	93.3%	COVID diagnosis	High computational cost
[15]	COVIDX-Net	X-ray images	Seven CNNs tested	90%	Binary classification	Small dataset
[16]	GNet	Public datasets	Feature reconstruction	99%	Transfer learning	Dataset limitations
[17]	MobileNetV2, VGG-19	COVID, pneumonia, control images	Transfer learning	96.78% (binary)	Multi-class tasks	Dataset imbalance
[18]	Xception	COVID, bacterial pneumonia, control	Class balance correction	94% (three-class)	Multi-class categorization	Overfitting risks
[19]	VGG16	Radiographs (COVID, pneumonia, normal)	Pre-trained networks	96.5%	Multi-class detection	Limited generalization
[20]	SVM, LR, NB, DT, KNN	Normal, pneumonia, COVID cases	ML and transfer learning	98.5%	Comprehensive classification	Limited model variety
[21]	MobileNetV2, SqueezeNet	Fuzzy datasets	Feature engineering (SMO)	x	COVID classification	Accuracy not detailed
[22]	MobileNetV2, AlexNet	CXR images	Transfer learning	x	COVID diagnosis	Results missing
[23]	GAN	Augmented datasets	Semi-supervised detection	x	Data augmentation	Unquantified results
[24]	GAN with WOA	SARS-CoV-2 CT scans	Hyperparameter tuning	99.22%	Synthetic image generation	Limited real-world tests
[25]	FedGAN	Federated datasets	Privacy-preserving GANs	x	Collaborative diagnostics	High resource needs
[26]	GAN	Synthetic CT dataset	Image generation	40% (prelim.)	Data augmentation	Early-stage results
[27]	DT, RF, KNN, NB	COVID radiography	Supervised learning	98.42%	Early detection	Limited model variety
[28]	DCNN + GAN	Chest X-ray dataset	Feature integration	x	COVID diagnostics	High computation cost
[29]	SD-GAN	Four X-ray datasets	Style normalization	x	Cross-dataset classification	Generalization challenges
[30]	R2C-GAN	Noisy X-ray images	Restoration + classification	>90%	Low-quality image handling	Limited validation
[31]	CGAN + Transfer Learning	X-ray images	Data augmentation	96%	Multi-class detection	Overfitting risks
[32]	GAN for quantification	CT scans	Pneumonia quantification	x	Outcome prediction	Limited validation
[33]	GAN for lesion localization	CT scans	Weakly supervised GANs	x	Lesion identification	High input quality

These lung masks (images) were then vertically integrated with their respective source photos, initially overlapped to form a single mask image comprising both lungs. The images were cropped and shuffled to introduce variation before input to the masking model.

2) Classification Dataset Pre-processing

The classification dataset was organized into two distinct folders for training and testing purposes. Images in the dataset were masked using a previously trained mask model. However, before masking, low-contrast images underwent preprocessing techniques such as Histogram equalization and CLAHE to enhance contrast. Additionally, thresholding was applied during the masking process to facilitate easier analysis of the images.

C. IMAGE SEGMENTATION

Segmentation techniques are employed to isolate the regions of interest in images, particularly in the case of COVID-19 detection where the lungs are primarily affected. However, certain regions of the lung tissue, especially those towards the edges, may introduce noise due to their radio density values

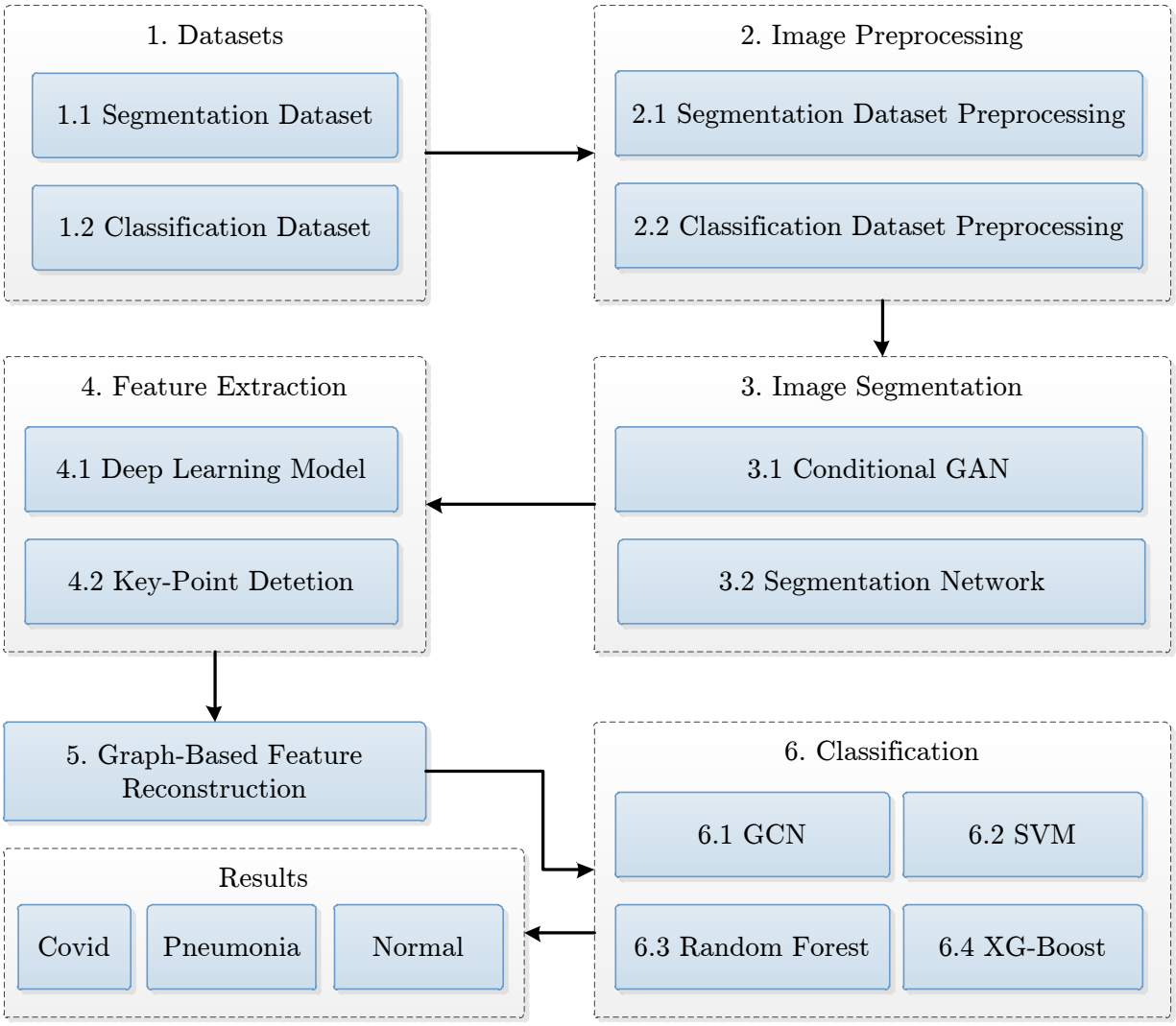

FIGURE 1: Proposed Methodology

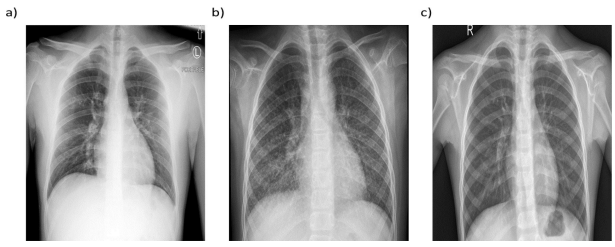

FIGURE 2: (a) Pneumonia CXR images (b) Normal CXR images (c) COVID-19 Infected CXR images.

falling outside the normal range. To address this issue, a U-net architecture conditional generative adversarial segmentation network will be utilized to filter out noise effectively.

1) Conditional-GAN

The conditional generative adversarial network (C-GAN) is a powerful tool commonly used in image processing [37] (Figure 3 and 4). In our system, the C-GAN consists of a generator and a discriminator. The generator is tasked with creating novel images based on a set of input samples, while the discriminator is trained to distinguish between real and fake images. The two networks compete during training to enhance performance. Specifically, in our setup, the generator takes chest X-rays (CXRs) as input and produces accurate lung masks. These masks are then evaluated by the discriminator alongside ground truth masks and input CXR images. The discriminator's role is to distinguish between genuine and fake pairs, and its weights are updated accordingly to improve performance.

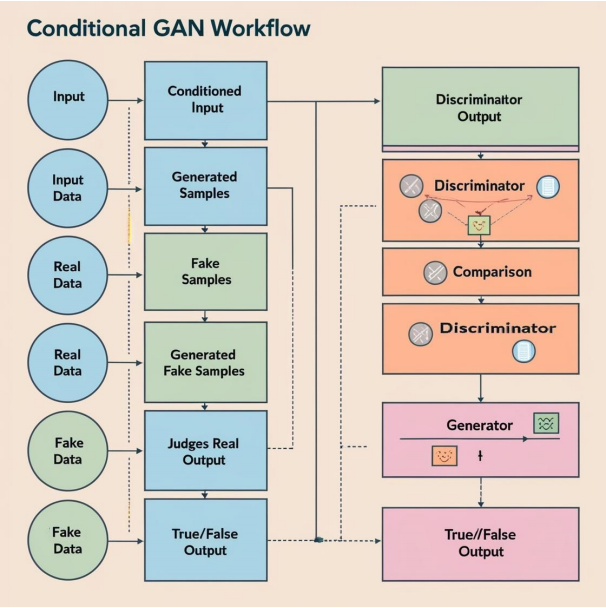

FIGURE 3: C-GAN Flow diagram

2) Training Segmentation Network

A U-net architecture, trained on [38], is employed for image segmentation. This involves training the generator using a pix-to-pix approach [39] and the discriminator using a classifier similar to Patch GAN. Each neural network competes to outperform the other during training. The generator loss decreases as the discriminator loss increases. Once trained, the generator produces accurate mask images for input CXR images. These generated masks, along with ground truth masks representing true segmentation, are then applied to the raw CXR images to create segmented lung images.

This approach allows for the extraction of meaningful features and enhances the quality of the dataset for subsequent stages of the analysis pipeline..

D. FEATURES EXTRACTION

Identification of relevant image features is crucial for effective classification, as each pixel in an image carries valuable information that can be utilized for analysis. Feature extraction involves measuring and extracting these values to represent the image or object digitally. In our framework, feature extraction from segmented lung images plays a vital role in distinguishing between COVID-19 cases and pneumonia (Figure 5).

Historically, manual feature extraction methods have been successful, but the recent surge in Deep Learning (DL) approaches have revolutionized feature extraction, offering superior performance by automatically detecting hidden patterns in data.

Our framework for feature extraction comprises two key components:

1) Deep Learning Models

Various transfer learning strategies, including DenseNet-169, DenseNet-201, VGG-16, VGG-19, Inception-ResNet, NasNetLarge, Xception, and a customized simple CNN (sCNN) model, will be employed in this research. DenseNet architectures are preferred due to their advantages, such as direct layer-to-layer interaction and support for feature reuse [40]. VGG models with 16–19 weight layers have demonstrated superior performance in terms of precision and validity [12]. Inception-ResNet [41] was chosen for its ability to learn both local and global features while maintaining acceptable training dynamics. NASNet-Large [42] was selected for its accuracy and computational efficiency, and Xception citeReference33 has shown promising results in feature extraction applications. Additionally, a customized sCNN model has been developed to compare its effectiveness in extracting distinctive characteristics from segmented lung images with transfer learning methods. Once trained, features are computed from the FC3 layer of the neural network as shown in Figure 6.

2) Key-Points Detection

The second component involves using computer vision algorithms, namely SIFT and BRISK, to extract significant details from the images. These algorithms identify key points or blobs resembling local characteristics in the segmented lung images. SIFT [43] is commonly used to extract rotation and scale-invariant features, while BRISK [44] is known for its fast processing and high precision. By applying the SIFT/BRISK algorithm to lung images, key regions are identified and separated into groups using the k-means clustering method. This results in a final 384-dimensional feature vector derived from the segmented lung images, combining information from DL models and key-points detection algorithms.

This comprehensive approach to feature extraction ensures that relevant information is captured effectively, enhancing the performance of the classification pipeline as shown in Figure 7.

E. GRAPH-BASED FEATURE RECONSTRUCTION

The next step involves reconstructing the features using the graph data obtained from the extracted features. This process aims to enhance individual features by incorporating information from their neighboring features. The term "features" here refers to the characteristics extracted by the trained CNNs and transfer learning models.

Each extracted feature contributes to the creation of a graph of features, where each node represents a feature. These features are grouped into batches of equal sizes to expedite computation. The total number of batches "n" can be determined using the formula ($n = \left\lceil D/N \right\rceil$) for a feature set $F \in \mathbb{R}^{D \times M}$, where "D" is the number of features in the dataset, "M" denotes the feature dimension, and "N" is the batch size.

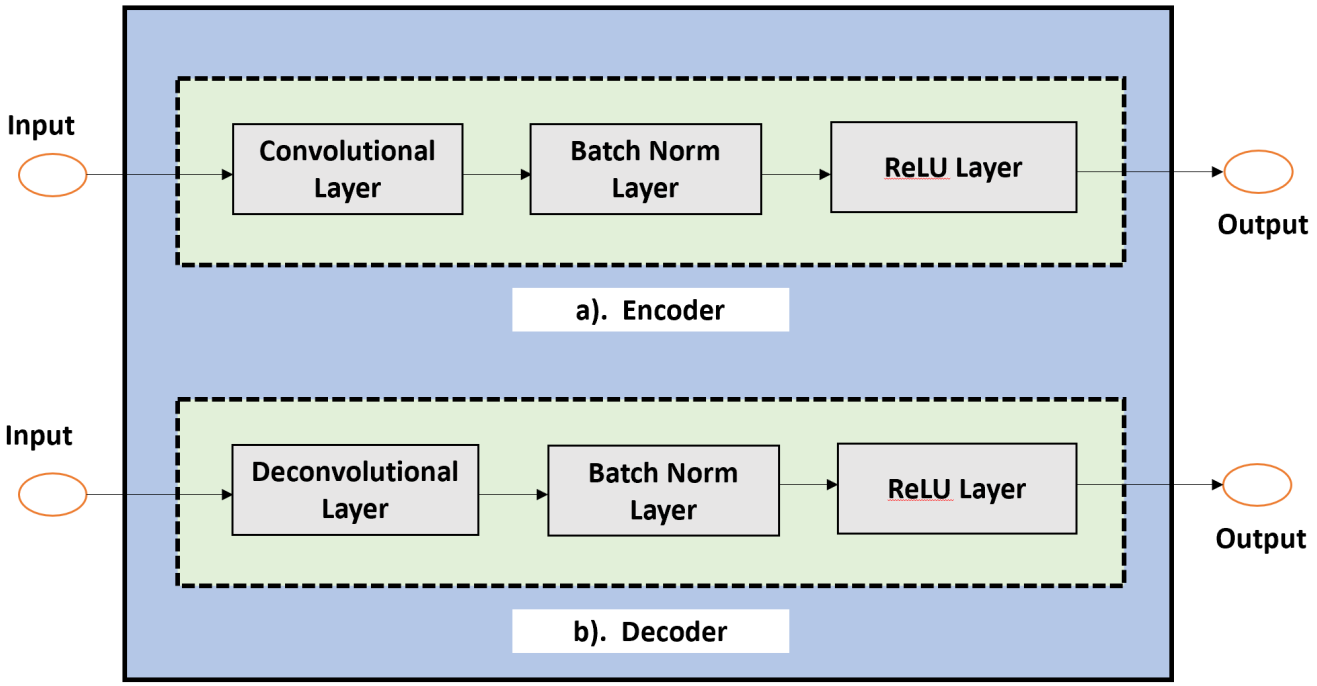

FIGURE 4: (a) The Encoder Block. (b) The Decoder Block.

Using paired variables (V_i, E_i), where V_i represents the nodes constituting each feature in the batch and E_i denotes the edges or connections between the nodes, a corresponding graph $G_i \in \mathbb{R}^M$ is generated for each F_i to depict the inherent relationships among nodes. Each node is connected to its k nearest neighbors by edges, determined by the shortest k value of Euclidean distance. The adjacency matrix $A_i \in \mathbb{R}^{S\times S}$ is crucial for constructing the graph G_i.

For each feature batch F_i, a graph G_i is initially created. Subsequently, the distance between each feature and the remaining features in the batch is computed, leading to the creation of a distance matrix Distance $\in \mathbb{R}^{S\times S}$. After sorting each row of the distance matrix in ascending order, an index matrix Index $\in \mathbb{R}^{S\times S}$ is generated to record the index of the closest k features in the batch F_i.

When one of the closest "k" neighbors is identified at a location, the value of one place in each row of A_i is set to 1. Finally, each feature batch F_i is multiplied by the normalized adjacency matrix A_i to produce a reconstructed feature batch for classification purposes. Refer to Figure 8 for a visual depiction of the feature reconstruction methods.

F. IMAGE CLASSIFICATION

We sought to deploy a GCN in conjunction with multiple machine learning models, including Support Vector Machine (SVM), Extreme Gradient Boosting (XG-Boost), and Random Forest (RF), which are frequently used as the classification module, in order to perform a comparison analysis. These classifiers will be trained and evaluated using the features extracted from the feature extraction workflow and feature reconstruction module.

1) Graph Convolutional Networks (GCN)

GCN is a multi-layer graph neural network that applies convolutional neural network theory and uses nodes from graphs as input. This kind of network is frequently employed in a number of applications. The system's design consists of an input layer and an output layer, the latter of which generates output signals in response to input signals received by the input layer. Due to their various layers, GCNs are able to capture more complicated information than GNETs, but they are typically more computationally expensive [17]. [16].

2) Support Vector Machine (SVM)

The Support Vector Machine (SVM) technique is used for classification, which separates training data into distinct classes using an ideal hyperplane. The discrimination and assimilation of the training data are key components of this supervised learning approach.

3) Random Forest (RF)

Random Forest, a supervised machine learning method, is frequently employed for classification tasks. It has proven to

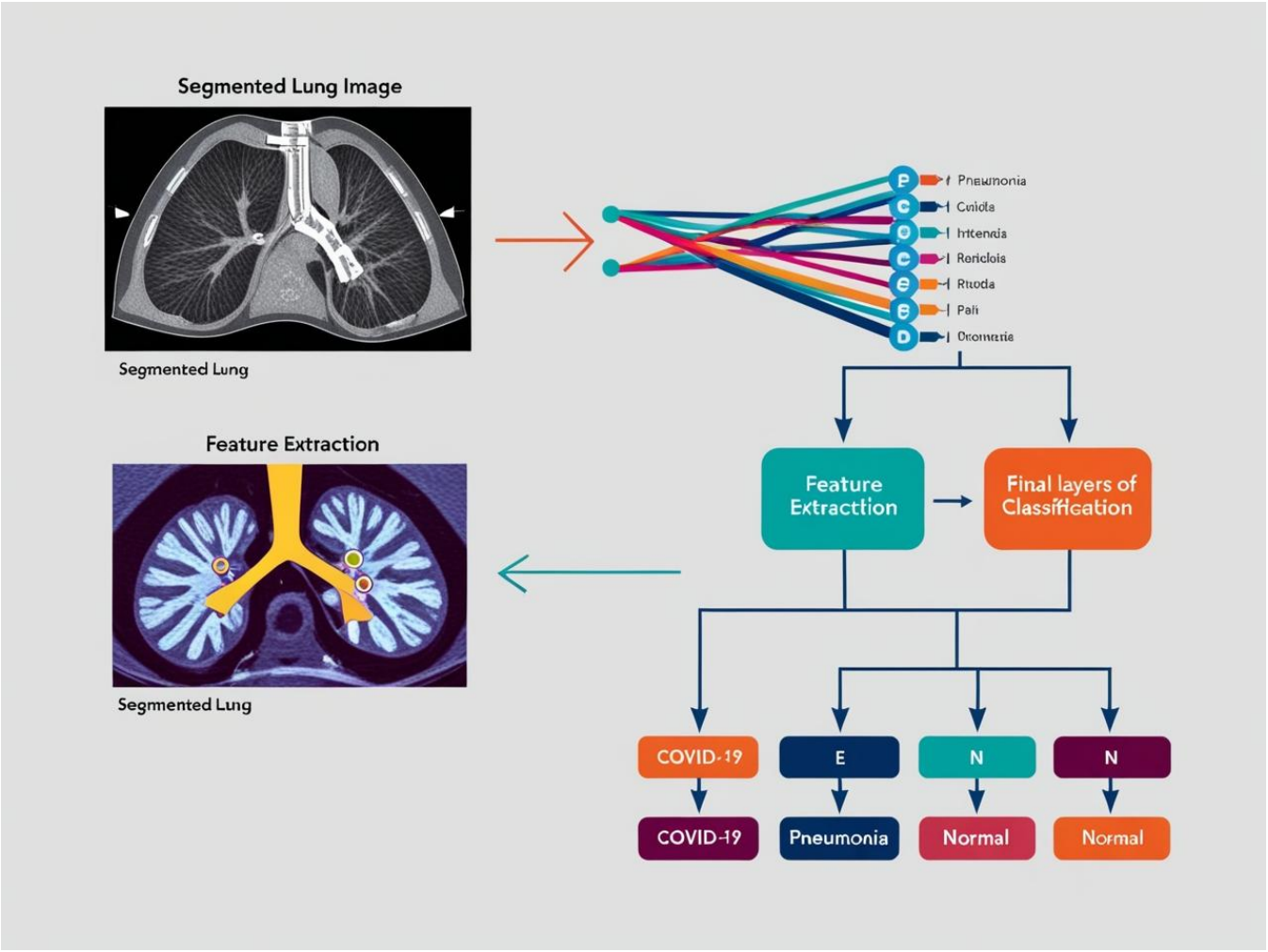

FIGURE 5: A condensed flowchart representing the suggested pipeline

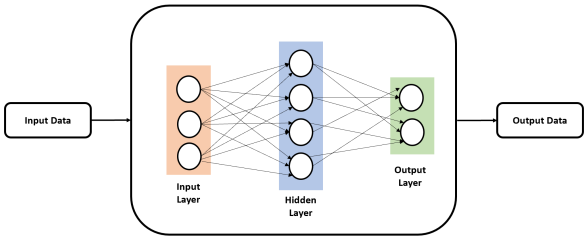

FIGURE 6: The computational model of the neural network

be adept at managing huge datasets and producing accurate forecasts.

4) Extreme Gradient Boosting (XG-Boost)

The Extreme Gradient Boosting method, often known as XG-Boost, is one of the classification modules used. It is an improved version of the gradient boosting algorithm that makes use of both tree-based and linear model-based learning

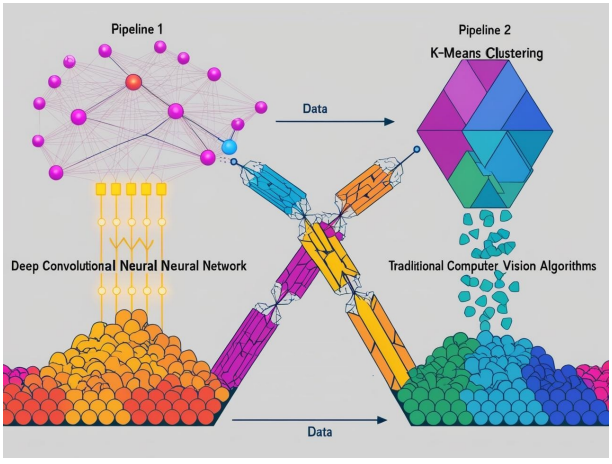

FIGURE 7: A diagrammatic representation that depicts the constituents present in the Feature Extraction unit

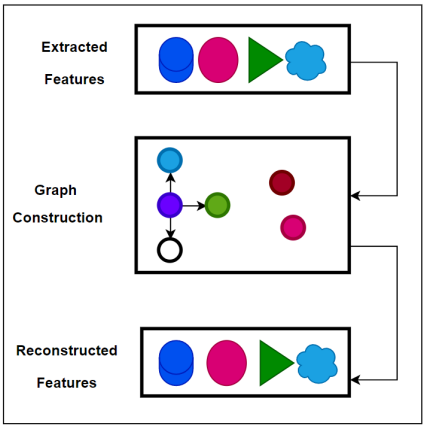

FIGURE 8: Feature reconstruction

techniques. As a result of its ability to perform parallel processing on a single machine, XG-Boost is a quick and efficient method. An algorithm for predictive modelling is improved as a result of the combination of these techniques.

IV. EXPERIMENTAL RESULTS

The actual experiments are explained in 7 smaller sections. The datasets utilised are described in Section A, along with their size, number of classes, and any data preparation methods applied. The experimental setup, including the parameters for the algorithm, the specifications for the hardware and software, and other pertinent information, is described in Section B. Section C presents the training times for all the models, followed by Section D's graph of model accuracy, Section E's graph of loss model, and Section F's description of the metrics used to gauge the success of the suggested strategy, such as average precision, average recall, average F1-score, and accuracy. The base classifiers ROC curves are shown in section G to make it easier to evaluate how well they work.

A. DATASETS

We used two openly accessible datasets to carry out segmentation and classification tasks. The segmentation network was trained using X-ray images (CXR) from the Segmentation Chest Radiographs (SCR) dataset [45]. The main dataset was made up of 988 photos, of which 48 were used for testing and the remaining 940 were split into two subsets for training and validation with a roughly 90/10 split.

A dataset of 700 images from the front view that were recognised as showing COVID-19 infection and had metadata suggesting views from both the posterior-anterior (PA) and anterior-posterior (AP) were also chosen for the COVID-19 classification. Additionally, the CXR images were used to derive the normal and pneumonia in the pneumonia dataset [36]. In order to maintain class balance, 645 CXR images of normal condition and 655 CXR images of pneumonia were randomly chosen, producing a total of 2000 images for categorization. Basic affine transformations were used to supplement the training images because the number of training images used was insufficient, bringing the total number of training images to 3000. Last but not least, the training images were split into training and validation datasets using an 80/20 split.

B. EXPERIMENT SETTINGS

An experimental setup was made up of a computer with 32 GB of RAM. Two different parameter sets were employed for the sequential training of transferred networks and GCN in order to prevent system problems brought on by memory restrictions. Training for the features extraction process took place over the course of 100 epochs with a set batch size of 32. The batch size "N" and the number of neighbours "k" in each batch were carefully chosen and maintained across all experiments to guarantee the best graph reconstruction results. The parts that follow will provide more information.

1) Training Conditional GAN

The Conditional Generative Adversarial Network (CGAN) approach, which focuses on image-to-image translation, was utilised during the training of the masking model to transform one type of image into another [39]. Generative Adversarial Network (GAN) subtype CGAN may generate images depending on specified input data. The model was trained using 830 training images over 250 epochs, and the results were satisfactory. The loss functions are optimised in relation to the number of epochs, as seen in the graphs below (Figure 9).

Discriminator Loss: Figure 9 depicts the variance in discriminator loss as the number of epochs increases. This is explained by the fact that as time passes, the generator loss likewise gets better, leading to an increase in the discriminator loss. However, the fact that the all-over loss function keeps getting smaller as the number of epochs increases suggests that the discriminator is getting better at telling the difference between real and fake masks.

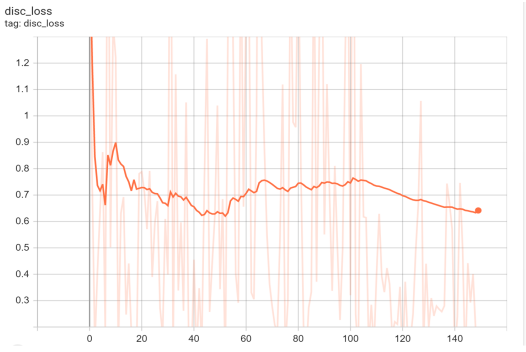

FIGURE 9: Discriminator loss function for CGAN during training process

Generator Loss: The Generator model receives the majority of attention during optimisation and training when using

C-GAN to translate images from one type to another. We looked at and examined numerous graphs that the training process produced.

- **Generator GAN Loss:** The discriminator's inability to distinguish between real and fake images at the start of the process, which led to a large generator GAN loss, is obvious from the graph shown in Figure 10. However, as the training went on, the discriminator got better at identifying the difference between the two kinds of images, which led to a reduction in the generator GAN loss. This shows that the fake images being created by the generator were becoming more convincing, and the quality of the generated masks as a whole was rising.
- **Generator L1 Loss:** Figure 11 illustrates the generator L1 loss, which is the difference between the generated output and the desired image. The generator's L1 loss lowers as the number of iterations rises, demonstrating the generator's capacity to produce images that closely mimic the real world. As the generator becomes familiar with the main characteristics of the lung mask, the loss initially decreases quickly. However, the loss swings between high and low values, demonstrating a progressive decline over time, as the model tries to learn the finer details, such as the margins of the mask or the boundaries of the lungs in the chest X-ray (CXR).
- **Generator Total Loss:** According to the graph in Figure 12, the generator total loss consists of two different types of losses and fluctuates between high and low levels. Due to the discriminator's initial inability to accurately distinguish between genuine and created images, the overall loss of the generator soon decreases, enabling the generator to swiftly capture the key characteristics of the mask. As the discriminator's performance improves, the loss fluctuates and gradually decreases.

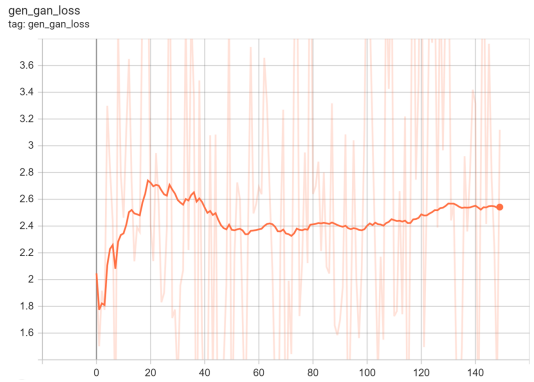

FIGURE 10: Generator GAN Loss function during training process

This occurs because the generator has trouble learning the micro-features that correspond to the lungs' borders or margins on chest x-rays (CXR). Figure 13 shows the segmented images, which show how the generative model performed.

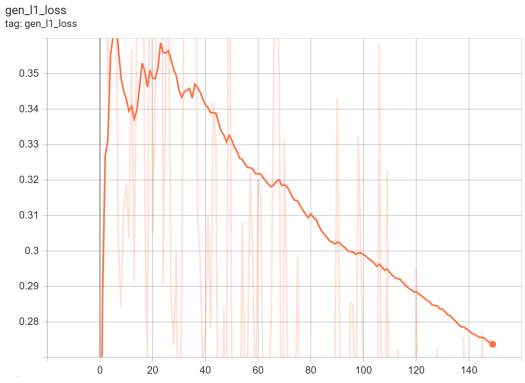

FIGURE 11: Generator L1 Loss during training process

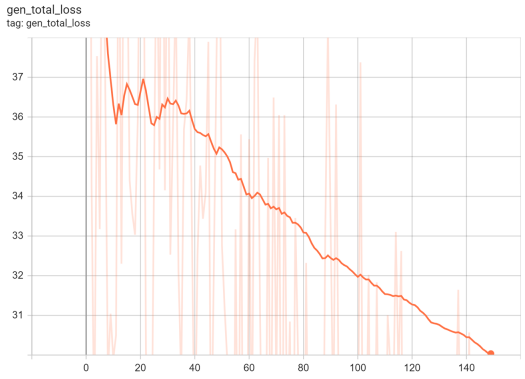

FIGURE 12: Generator Total Loss during training process

2) CLASSIFICATION PIPELINE IMPLEMENTATION

With computer vision feature extraction algorithms (SIFT and BRISK), deep learning classifier Graph Convolutional Network (GCN), and widely used machine learning methods for classification like Support Vector Machine, Random Forest, and XG-Boost for comparative analysis, we implemented 8 different deep learning approaches in all conceivable combinations. In order to plot the comparison graph and use the trained model for classification using machine learning classifiers, we also used Keras callbacks in TensorFlow to save the trained model in a .h5 file format. The resulting .h5 file can then be loaded to make predictions on new data or to continue training the model.

Deep Learning Models:

Below are descriptions of several deep learning models while the experimental setup is outlined in Table 2.

- **Dense-Net:** The foundation of Dense-Net models is the idea of "dense connectivity" between layers. The output of the layer before it serves as the input to the subsequent layer in a conventional CNN. However, in the case of Dense-Net, the feature maps produced by all of the earlier layers serve as the input that is supplied to each layer. With greater feature propagation and stronger gradients as a result of this dense connectivity, model performance may be enhanced[40]. Due to their advantages, Dense-Net designs were selected for our

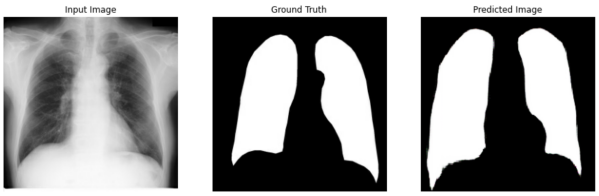

FIGURE 13: One example of a chest X-ray image with its corresponding original and predicted lung masks is shown in the image on the left, center, and right, respectively.

TABLE 2: DL Models

Experimental Setup	
Model	DenseNet , VGG , Inception, Resnet, NasNet, CNN
Variants	DenseNet169, DenseNet201, VGG16, VGG19, Inception-Resnet, NasNetLarge, Extreme Inception, sCNN
Maximum Training Epochs	100
Batch Size	32
Initial learning rate	0.001
Technique for optimization	Adam
Randomizing the training set	Every epoch

study.With our dataset, we have applied two of the variants: Dense-net169 and Dense-net201. with 169 and 201 layers respectively.

- **VGG:** There are several convolutional neural network (CNN) designs known as the VGG (Visual Geometry Group). It was developed in 2014 by the Visual Geometry Group at the University of Oxford, and there are other variations of it [12], including VGG-16 and VGG-19, which, according to [12], have 16 and 19 layers, respectively. We used both models to get precise results.
- **Inception-ResNet:** Convolutional neural network (CNN) designs integrate ideas from the Inception and ResNet architectures to create the Inception-ResNet architecture [41]. We chose this model because it can retain acceptable training dynamics while learning both local and global features.
- **NASNet Large:** In a research [42] published in 2018 by Barret Zoph and Quoc V. Le from Google Brain titled "Learning Transferable Architectures for Scalable Image Recognition," the NASNet (Neural Architecture Search Network) family of models—including NASNet-Large—was first described. It is created using a neural architecture search method based on reinforcement learning. Since NASNet-Large is intended to be both extremely accurate and computationally efficient, we employed it.
- **Xception:** A convolutional neural network design called Xception was introduced by The Extreme Inception, also known as Xception [46]. Xception has demonstrated impressive results when utilised as a feature extractor in transfer learning applications.
- **Simple CNN:** Self-customized simple CNN (sCNN model), unlike transfer learning methods, was built from the ground up. The FC3 layer's features are computed once a learnt deep learning model has been trained.

Key Points Detection Methods Key point detection algorithms such as SIFT and BRISK are widely used in computer vision and image processing applications to extract critical information from images and videos. Figure 14 illustrates clearly how these algorithms identify significant points.

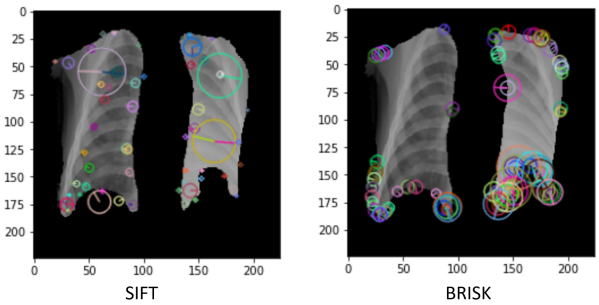

FIGURE 14: Sample lung images visually illustrate the detected features through the utilization of SIFT and BRISK algorithms

Graph Construction: Since one of Stellar Graph's primary characteristics is that it supports deep learning models, including Graph Convolutional Networks (GCNs), We used it to create a graph from the retrieved data, with relevant labels treated as edges, and the train and test characteristics are taken as nodes. The generated graph is displayed in Figure 15.

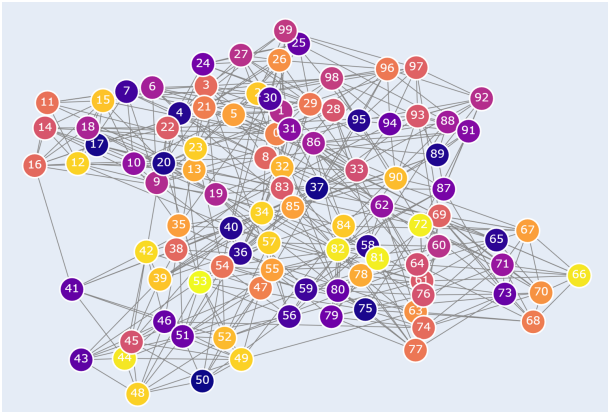

FIGURE 15: Graph Representation of Features

The adjacency matrix A is generated after the graph is created, and it produces a sparse matrix representation of the graph. The edges are unweighted when the weighted option is set to False. The identity matrix with ones on the diagonal is then added, adding self-connections to the adjacency matrix A. In order to meet the requirements of some graph convolutional neural network (GCN) architectures, it is ensured that each node has a self-loop by doing this. The

"diags" function from the sparse module is then used to compute the degree matrix to the power of -1/2. This matrix is used to normalise the adjacency matrix, which improves the GCN's stability and convergence. Finally, The adjacency matrix is normalized by left-multiplying it by the degree matrix and right-multiplying it by its transpose, resulting in the normalized adjacency matrix. To create the normalised adjacency matrix, the adjacency matrix is left-multiplied by the degree matrix and right-multiplied by its transpose.

Classifiers: In this investigation, we employed two sets of classifiers:

- **DL Classifier:** GCN (Graph Convolutional Network), a neural network designed to process graph-structured data, was the neural network used in our study. Convolutional filters are used on a graph in the primary principle of GCN, similar to how convolutional filters are used on images in conventional convolutional neural networks (CNNs) [47]. We encoded the labels and mapped the nodes to their indices in the adjacency matrix before providing the graph as an input to GCN for the purpose of graph classification. Hyper-parameters used throughout the experiment are listed in Table 3 below.
- **ML Classifiers:** We experimented using Support Vector Machines (SVM), Random Forest (RF), and XGBoost on both the features extracted directly from deep learning (DL) models, key detection algorithms, and the reconstructed features from GCN in addition to using GCN as a standalone classification model.

TABLE 3: GCN Parameters

Experimental Setup	
Models	GCN
Layers	2
Epochs	50
Batch Size	32
Dropout	0.5
Activation Function	Relu
Optimizer	Adam

Using the trained GCN, embeddings are extracted for all nodes in the graph. Utilising the Model class of Keras, we created a neural network model that receives the same inputs as the original model but outputs the second-to-last layer of the model, which contains the node embedding. Figure 16 shows an illustration of the Uniform Manifold Approximation and Projection (UMAP) dimensionality reduction technique for visualising the embedding.

The embeddings array are flattened and reshaped to a 2-dimensional matrix where each row corresponds to a node and its features. The idea that each node is represented by a vector of features is highlighted when the embeddings are reshaped into a 2D matrix. For the data to be fed into machine learning models, this is essential. These features are then utilised as input features to train ML classifiers like SVM, RF, and XG-Boost. The pre-trained deep learning (DL) models' extracted features were used as training and testing inputs for popular ML classifiers.

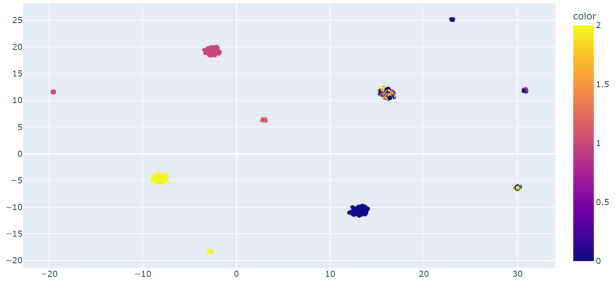

FIGURE 16: Nodes Embeddings Visualization using UMAP (Uniform Manifold Approximation and Projection)

We were able to contrast the performance of GCN as a solo model and its performance when integrated with conventional ML classifiers by employing these two alternative methodologies. We were able to choose the best technique for our specific categorization target by weighing the benefits and drawbacks of each approach.

C. TRAINING TIME

Table 4 displays the training time for the different CNN architectures taken into consideration. The NASNet Large model had the longest training duration (53432 s), averaging 7.55 s per step. The Simple CNN model, on the other hand, required the least amount of time to train (3222 s), with an average training time per step of 0.5372 s on Core i-5, 256 SSD with 8GB RAM.

TABLE 4: Training Time of DL Models

S. No.	Model Name	Training time	Average Time Per Step
1.	VGG-16	23860 s	4.19 s
2.	VGG-19	27214 s	4.54 s
3.	Simple Conv. Net	3222 s	0.5372 s
4.	DenseNet-169	13405 s	2.24 s
5.	DenseNet-201	14171 s	2.1 s
6.	Inception Resnet	23744 s	3.82 s
7.	NasNetLarge	53432 s	7.55 s
8.	Xception	16936 s	2.52 s

D. VISUAL DEPICTION OF MODELS ACCURACY

The accuracy graphs in Figure 17 show how various models performed over the course of training. The graph's y-axis represents the accuracy score, while the x-axis represents the number of epochs. The graph shows an orange line for the validation accuracy and a blue line for training accuracy. The accuracy scores of several models can be analysed and contrasted with the help of this visual depiction.

E. GRAPHICAL REPRESENTATION OF MODEL LOSS

Figure 18's loss graphs, which depict the model losses throughout training, have an x-axis for the number of training epochs and a y-axis for the loss score. The blue line shows the loss during training, while the orange line shows the loss during validation.

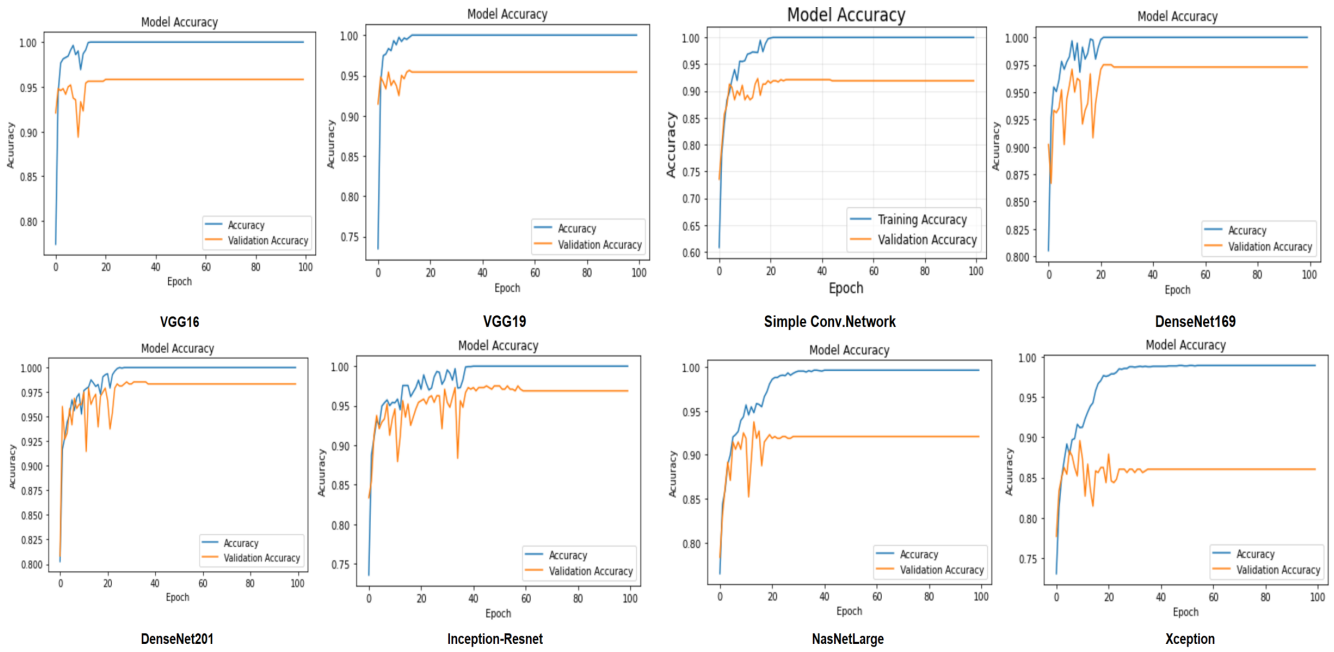

FIGURE 17: Training and Validation Accuracy curves displaying the respective accuracies during training and validation.

F. EXPERIMENTAL RESULTS FROM THE CLASSIFICATION PIPELINE

The proposed methodology involves training all the models, and at each stage, performance parameters based on the confusion matrix are calculated. The method combines both traditional feature extraction techniques and deep learning approaches, and it has produced successful results. Performance indicators include parameters like accuracy (Acc), sensitivity (Recall), precision, false positive rate (FPR), false negative rate (FNR), and F1-Score.

The rates of correctly identified positive and negative cases, as well as falsely identified positive and negative cases, are represented by the four components of the confusion matrix. The COVID-19, normal, and pneumonia categorization outcomes of the segmented X-ray images are shown in the confusion matrix, denoted as $C \in \mathbb{R}^{3\times3}$, in terms of their expected and actual categories.

The proposed methodology's performance is assessed by calculating the performance parameters based on the confusion matrix at each stage and averaging them across the three classes under consideration. How well different deep transfer learning models, including the VGG family (VGG-16 and VGG-19), the DenseNet family (DenseNet-169 and DenseNet-201), Inception-Resnet, Xception, and NasNet-Large, as well as a straightforward CNN, perform when using deep features is outlined in a tabular format. In the final layer of the deep learning models, Softmax (which is the final layer of the models), SVM, RF, and XGBoost are used to classify the feature matrices that these models produce. This approach is suggested for locating COVID-19 in CXR images, and the effectiveness of the selected deep neural network is assessed using these methods.

1) Accuracy

The accuracy ranges of classification using algorithms for computer vision feature extraction and then common machine learning (ML) classifiers like SVM, XG-Boost, and RF are provided in Table 5 as shown below.

TABLE 5: Accuracy of ML Classifiers with (SIFT & BRISK)

Model Name	SIFT	BRISK
SVM	79%-88%	62%-87%
XG-Boost	79%-87%	60%-86%
Random Forest	75%-86%	57%-85%

Table 6 shows the accuracy of the selected deep neural network (DNN) models during the pipeline's earliest stages, including the accuracy of DL transfer learning models both on their own and in combination with ML classifiers like SVM, XG-Boost, and RF.

TABLE 6: Accuracy of DL Models with ML Classifiers

Model Name	Model Accuracy	SVM	XG-Boost	Random-Forest
VGG-16	95%	97%	97%	97%
VGG-19	95%	95%	96%	96%
Simple Conv. Net	90%	92%	93%	93%
DenseNet-169	97%	98%	99%	99%
DenseNet-201	98%	99%	98%	99%
Inception ResNet	97%	95%	96%	95%
NasNetLarge	91%	94%	94%	94%
Xception	85%	89%	85%	85%

Table 6 displays the models' accuracy when used with key point detection techniques like the SIFT/BRISK algorithms. Table7 display the model's accuracy without a Graph Convolutional Network, while Table8 displays the models' accu-

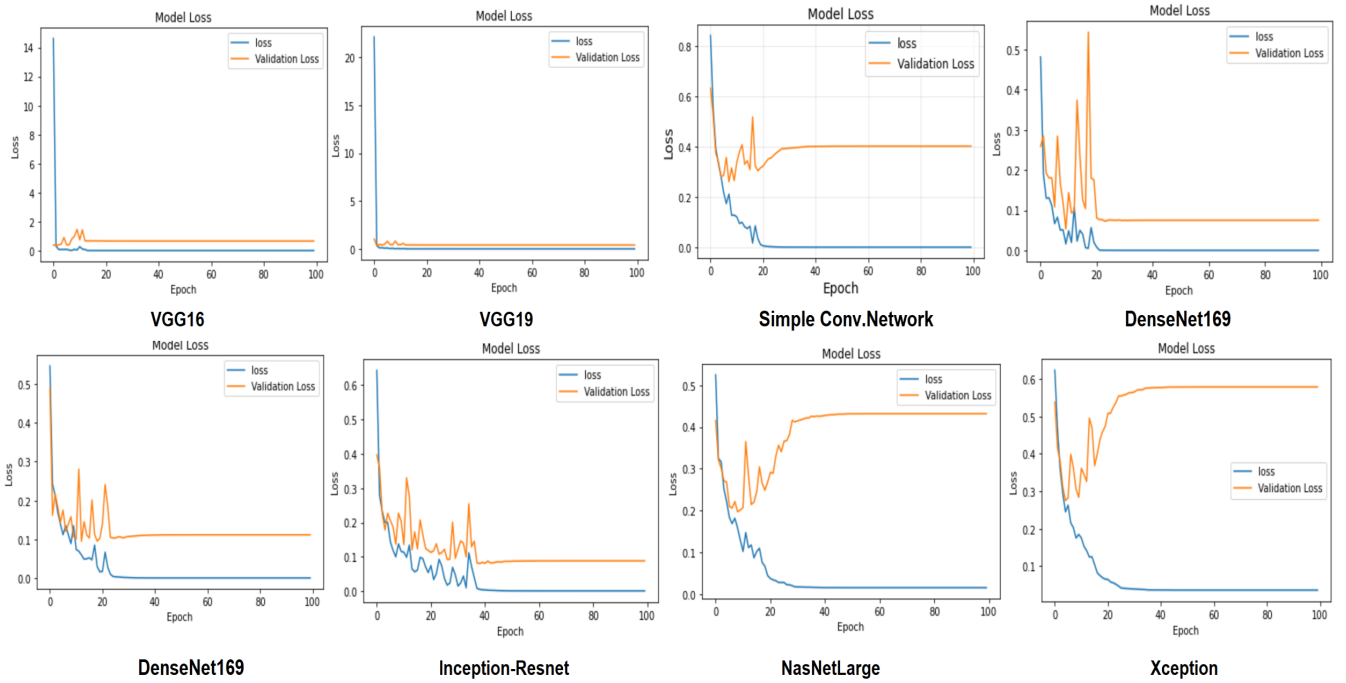

FIGURE 18: Convergence curves of training and validation loss are presented. The terms 'Loss' and 'Validation Loss' are referred to the convergence curves for training and validation loss.

racy when used with a graph convolutional network (GCN) and critical point detection techniques like the SIFT/BRISK algorithms.

2) Average F1-Score

Following the use of SVM, XGBoost, and Random Forest, Table 9 displays the average F1 scores for classification that are exclusively dependent on feature extraction methods in computer vision.

Table 10 shows the average F1 score of the selected models during the pipeline's initial stages, including the average F1 scores of DL transfer learning models used both alone and in conjunction with ML classifiers like SVM, XG-Boost, and RF.

The F1 score of models combined with critical point detection methods like the SIFT/BRISK algorithms is shown in Table 11.

Table 12 displays the F1 score of various models when used in conjunction with a graph convolutional network (GCN) and critical point detection techniques like the SIFT/BRISK algorithms.

3) Average Precision

Following the usage of SVM, XG-Boost, and Random Forest, Table 13 displays the average precision range for classification when computer vision feature extraction algorithms are applied.

Table 14 displays the average precision of the models used in the pipeline's earliest stages, including the average precision of DL transfer learning models used both alone and in conjunction with ML classifiers like SVM, XG-Boost, and RF.

The average precision of models when used with critical point detection techniques like SIFT/BRISK algorithms is shown in Table 15.

The average precision of the models when combined with the graph convolutional network (GCN) and critical point detection techniques like the SIFT/BRISK algorithms is shown in Table 16.

4) Average Recall

The average recall range for classification simply based on computer vision feature extraction algorithms and application of popular machine learning (ML) classifiers as SVM, XG-Boost, and RF is provided in Table 17.

The average recall of models at the beginning of the pipeline, including DL transfer learning models as independent classifiers and in combination with ML classifiers like SVM, XG-Boost, and RF, are shown in Table 18.

The average recall of the models when combined with critical point detection methods like SIFT/BRISK algorithms is shown in Table 19.

The average recall of the models when combined with the graph convolutional network (GCN) and critical point detection techniques like the SIFT/BRISK algorithms is shown in Table 20.

According to the aforementioned tables, the proposed methodology tested a total of twenty different combinations using eight different deep learning (DL) models, two important feature extraction algorithms (SIFT, BRISK), one

TABLE 7: Accuracy of DL Models without GCN, BRISK/SIFT & ML Classifiers

Model Name	Model Accuracy		SVM		XG-Boost		Random Forest	
	SIFT	BRISK	SIFT	BRISK	SIFT	BRISK	SIFT	BRISK
VGG-16	95%	96%	96%	95%	96%	96%	96%	96%
VGG-19	95%	95%	95%	94%	95%	94%	95%	94%
Simple Conv. Net	90%	94%	92%	92%	92%	92%	93%	93%
DenseNet-169	97%	86%	97%	86%	96%	85%	97%	85%
DenseNet-201	97%	88%	97%	87%	97%	88%	98%	87%
Inception ResNet	90%	67%	90%	68%	89%	66%	89%	66%
NasNetLarge	86%	62%	86%	61%	85%	62%	85%	61%
Xception	93%	60%	93%	62%	93%	58%	93%	61%

TABLE 8: Accuracy of DL Models with GCN, BRISK/SIFT & ML Classifiers

Model Name	GCN		GCN + SVM		GCN + XG-Boost		GCN + Random-Forest	
	SIFT	BRISK	SIFT	BRISK	SIFT	BRISK	SIFT	BRISK
VGG-16	79%	77%	81%	81%	81%	81%	81%	81%
VGG-19	77%	78%	81%	81%	81%	81%	81%	81%
Simple Conv. Net	77%	75%	81%	81%	81%	81%	81%	81%
DenseNet-169	80%	79%	81%	81%	81%	81%	81%	81%
DenseNet-201	76%	79%	80%	81%	81%	81%	81%	81%
Inception ResNet	79%	78%	83%	83%	83%	83%	83%	83%
NasNetLarge	79%	87%	83%	83%	83%	83%	83%	83%
Xception	80%	77%	81%	81%	81%	81%	81%	81%

TABLE 9: Average F1 Score of ML Classifiers with (SIFT & BRISK)

Model Name	SIFT	BRISK
SVM	78%-88%	57%-87%
XG-Boost	77%-87%	56%-86%
Random Forest	73%-85%	46%-84%

TABLE 10: Average F1 Score of DL Classifiers with ML Classifiers

Model Name	Model F1 Score	SVM	XG-Boost	Random-Forest
VGG-16	95%	97%	97%	96%
VGG-19	95%	95%	96%	96%
Simple Conv. Net	90%	92%	93%	93%
DenseNet-169	97%	98%	98%	99%
DenseNet-201	98%	99%	98%	99%
Inception ResNet	97%	95%	96%	95%
NasNetLarge	82%	89%	89%	89%
Xception	83%	85%	83%	83%

DL classifier, and three widely used machine learning (ML) classifiers. One can observe that, in terms of mean accuracy, recall, precision, and F1- score, transfer learning techniques have outperformed the standard CNN model.

Additionally, the random forest classifier outperformed the other machine learning classifiers examined in terms of classification performance measures. We have two models, DenseNet-169 and DenseNet-201, that perform quite well. When using features collected from deep neural network models like the DenseNet-169 and DenseNet-201 models, the classifier based on the random forest algorithm has the highest classification performance with 99% average accuracy, F1-score, precision, and recall. The simple customised CNN model that used a machine learning model such as XG-Boost in the last layer demonstrated overall average accuracy, F1-score, precision, and recall of 93%, 93%, 92%, and 93%, respectively. Other transfer learning models have been found to have comparable performance to that of top-performing DenseNet-169 and DenseNet-201 models in several other combinations.

Additionally, in accordance with the aforementioned tables, the performance parameters were assessed by fusing the deep features with the features manually retrieved using the SIFT and BRISK algorithms. It was found that the categorization performance indicators did not significantly change, Evidently, by combining SIFT-based features with deep features from the DenseNet-201 models with classification layer employs a machine learning classifier such as Random Forest, the average accuracy, F1-score, precision, and recall achieved was 98% again out performing all other models. However the statistics shows a considerable decline in classification performance when key-point features based on BRISK were merged with features extracted from DenseNet-169 and DenseNet-201 models.

In contrast, machine learning classifiers like RF showed considerable gains in classification performance. After combining BRISK-based features with deep features from VGG-16 with RF as ML classifier showed 96% average accuracy, F1-score, precision, and recall which is the highest of all for this specific combination.

In addition, we discovered that feeding deep features from Inception Resnet and NasNet-Large models and SIFT-BRISK based features to GCN produced superior results than other models under consideration. The best accuracy of 87% was obtained in a combination when deep features from the NasNet-Large model were combined with BRISK-based features and sent to the GCN classifier. While the best accuracy of 83% was obtained in a combination when reconstructed features from Inception Resnet and NasNet-Large models and SIFT-BRISK have been fed to ML classifiers like SVM, XG-Boost and Random Forest.

TABLE 11: Average F1 Score of DL Models, ML Classifiers and SIFT/BRISK

S. No.	Model Name	Model F1 Score	SVM	XG-Boost	Random-Forest				
		SIFT	BRISK	SIFT	BRISK	SIFT	BRISK	SIFT	BRISK
1.	VGG-16	95%	96%	96%	96%	96%	96%	96%	96%
2.	VGG-19	95%	95%	95%	94%	95%	94%	96%	94%
3.	Simple Conv. Net	91%	94%	92%	92%	92%	92%	93%	92%
4.	DenseNet-169	97%	86%	97%	86%	97%	85%	97%	85%
5.	DenseNet-201	97%	88%	97%	87%	97%	88%	98%	87%
6.	Inception ResNet	90%	62%	89%	66%	88%	63%	88%	63%
7.	NasNetLarge	86%	59%	86%	58%	84%	59%	84%	58%
8.	Xception	93%	59%	93%	62%	92%	58%	93%	61%

TABLE 12: F1 Score of DL Models, GCN, ML Classifiers & BRISK/SIFT

Model Name	GCN		GCN + SVM		GCN + XG-Boost		GCN + Random-Forest	
	SIFT	BRISK	SIFT	BRISK	SIFT	BRISK	SIFT	BRISK
VGG-16	79%	77%	81%	81%	81%	81%	81%	81%
VGG-19	77%	79%	81%	81%	81%	81%	81%	81%
Simple Conv. Net	77%	75%	81%	81%	81%	81%	81%	81%
DenseNet-169	80%	79%	81%	81%	81%	81%	81%	81%
DenseNet-201	76%	80%	80%	81%	81%	81%	81%	81%
Inception ResNet	78%	76%	81%	81%	81%	81%	81%	81%
NasNetLarge	78%	86%	81%	81%	81%	81%	81%	81%
Xception	80%	77%	81%	81%	81%	81%	81%	81%

TABLE 13: Average Precision of ML Classifiers with (SIFT & BRISK)

Model Name	SIFT	BRISK
SVM	80%-88%	61%-86%
XG-Boost	79%-87%	59%-86%
Random Forest	76%-86%	58%-85%

TABLE 14: Average Precision of DL Classifiers with ML Classifiers

Model Name	Model Precision	SVM	XG-Boost	Random-Forest
VGG-16	95%	97%	97%	97%
VGG-19	95%	95%	95%	96%
Simple Conv. Net	90%	92%	92%	92%
DenseNet-169	97%	99%	99%	99%
DenseNet-201	98%	99%	99%	99%
Inception ResNet	97%	95%	96%	95%
NasNetLarge	84%	90%	90%	90%
Xception	83%	91%	83%	83%

Figure 19 presents the classification report for the DenseNet-169 model's optimal pipeline level and displays the model's performance.

Figure 20, which displays the classification report for the DenseNet-201 model's ideal pipeline level, shows how well the model performed.

To showcase the performance of the proposed feature extraction models in each class, Figure 21 illustrates the confusion matrices for a combined set of 240 test images. This provides specific evidence of how the models perform for each class. Each cell of the matrix contains a numerical value representing the count of images that belong to the corresponding category based on classification. The matrix sum can provide the overall classification accuracy for a given model. The experimental outcomes for classification accuracy are shown in Figure 21.

The DenseNet-201 model with RF in the last layer demonstrated an overall classification accuracy of 99%, with correct classification of 82 COVID-19, 91 normal, and 64 pneumonia images, as depicted in Figure 21 (a). The other models, including DenseNet-169 model with RF, VGG16-BRISK and RF, Inception-Resnet-SIFT-GCN and SVM, NasNetLarge-BRISK- GCN and XG-Boost, also exhibited similar class-specific performance, as indicated in (b), (c), (d), and (e) of the Figure 21.

Finally, it is noteworthy to mention that In this research, the DenseNet-169 and DenseNet-201 architectures, used with RF or combined with SIFT and RF, exhibited the most exceptional performance compared to all other models suggested.

G. ROC CURVE

Better performance is indicated by classifiers that show curves closer to the upperleft corner. A random classifier is expected to produce outcomes that lie on the diagonal line, where the true positive rate (TPR) is equal to the false positive rate (FPR). Figure 22 shows the ROC curves of the DenseNet-201 model with SVM/XGBoost/RF, DenseNet-169 model with SVM/XG-Boost/RF, VGG16-BRISK and SVM/XG-Boost/RF, Inception-Resnet-SIFT-GCN and SVM/XG-Boost/RF, NasNetLarge-BRISKGCN and SVM/XG-Boost/RF.

V. RESULTS AND DISCUSSION

A. COMPARISON WITH EXISTING STATE-OF-THE-ART METHODS

We provide a comparison in Table 21 of our top-performing models against other advanced DL techniques used for identifying COVID-19 in CXR images. According to a source [48],

TABLE 15: Average Precision of DL Models, ML Classifiers, and SIFT/BRISK

S. No.	Model Name	Model Precision		SVM		XG-Boost		Random-Forest	
		SIFT	BRISK	SIFT	BRISK	SIFT	BRISK	SIFT	BRISK
1.	VGG-16	96%	96%	96%	95%	96%	96%	96%	96%
2.	VGG-19	95%	95%	95%	94%	95%	94%	96%	94%
3.	Simple Conv. Net	91%	94%	92%	92%	92%	92%	93%	92%
4.	DenseNet-169	97%	86%	97%	86%	97%	85%	97%	84%
5.	DenseNet-201	97%	88%	97%	87%	97%	88%	98%	87%
6.	Inception Resnet	91%	69%	91%	67%	87%	64%	88%	64%
7.	NasNetLarge	87%	59%	86%	59%	83%	59%	84%	58%
8.	Xception	93%	62%	92%	62%	92%	59%	92%	61%

TABLE 16: Average Precision of Models with GCN, BRISK/SIFT & ML Classifiers

Model Name	GCN		GCN + SVM		GCN + XG-Boost		GCN + Random-Forest	
	SIFT	BRISK	SIFT	BRISK	SIFT	BRISK	SIFT	BRISK
VGG-16	85%	82%	82%	82%	82%	82%	82%	82%
VGG-19	85%	83%	82%	82%	82%	82%	82%	82%
Simple Conv. Net	80%	81%	82%	82%	82%	82%	82%	82%
DenseNet-169	83%	85%	82%	82%	82%	82%	82%	82%
DenseNet-201	82%	81%	82%	82%	82%	82%	82%	82%
Inception Resnet	80%	81%	91%	91%	91%	91%	91%	91%
NasNetLarge	82%	94%	91%	91%	91%	91%	91%	91%
Xception	83%	82%	82%	82%	82%	82%	82%	82%

FIGURE 19: Categorization report of the DenseNet-169 model

TABLE 17: Average Recall of ML Classifiers with (SIFT & BRISK)

Model Name	SIFT	BRISK
SVM	77%-88%	56%-86%
XG-Boost	76%-87%	56%-86%
Random Forest	71%-85%	48%-84%

TABLE 18: Average Recall of DL Classifiers with ML Classifiers

Model Name	Model Recall	SVM	XG-Boost	Random-Forest
VGG-16	94%	97%	97%	97%
VGG-19	95%	96%	96%	96%
Simple Conv. Net	90%	92%	93%	93%
DenseNet-169	97%	98%	99%	99%
DenseNet-201	98%	99%	98%	99%
Inception ResNet	97%	96%	96%	96%
NasNetLarge	82%	89%	89%	89%
Xception	83%	83%	83%	83%

by combining the deep features computed from Gaussian filtered images using Inception V3 architecture with local binary patterns (LBP) features, an average accuracy, sensitivity, and specificity of 95.11%, 93.15%, and 96.5% have

TABLE 19: Average Recall of DL Models, ML Classifiers and SIFT/BRISK

S. No.	Model Name	Model Recall		SVM		XG-Boost		Random-Forest	
		SIFT	BRISK	SIFT	BRISK	SIFT	BRISK	SIFT	BRISK
1.	VGG-16	96%	95%	96%	96%	96%	96%	96%	96%
2.	VGG-19	95%	95%	95%	94%	94%	94%	95%	94%
3.	Simple Conv. Net	90%	93%	92%	92%	92%	92%	93%	92%
4.	DenseNet-169	97%	86%	96%	86%	96%	85%	97%	85%
5.	DenseNet-201	97%	88%	97%	87%	97%	88%	98%	87%
6.	Inception Resnet	90%	61%	89%	65%	88%	63%	88%	62%
7.	NasNetLarge	86%	60%	86%	58%	84%	59%	85%	58%
8.	Xception	93%	60%	92%	63%	92%	59%	92%	61%

TABLE 20: Average Recall of DL Models with GCN, BRISK/SIFT & ML Classifiers

Model Name	GCN		GCN + SVM		GCN + XG-Boost		GCN + Random-Forest	
	SIFT	BRISK	SIFT	BRISK	SIFT	BRISK	SIFT	BRISK
VGG-16	79%	78%	82%	82%	82%	82%	82%	82%
VGG-19	76%	79%	82%	82%	82%	82%	82%	82%
Simple Conv. Net	78%	76%	82%	82%	82%	82%	82%	82%
DenseNet-169	81%	79%	82%	82%	82%	82%	82%	82%
DenseNet-201	76%	79%	81%	82%	82%	82%	82%	82%
Inception Resnet	83%	80%	77%	77%	77%	77%	77%	77%
NasNetLarge	81%	82%	77%	77%	77%	77%	77%	77%
Xception	81%	78%	82%	82%	82%	82%	82%	82%

FIGURE 20: Categorization report of the DenseNet-201 model

been achieved respectively.

The usage of feature extraction from MobileNet v2 with transfer learning documented in [17] yielded an average classification accuracy of 93.33%, along with a recall of 90.66% and specificity of 95.23%. The performance of two different models has been evaluated for identifying lung infections in X-ray images. ResNet-34 based features were used in the transfer learning model, which resulted in an average accuracy of 95.29%, recall of 92.97%, and specificity of 96.46% [22].

Another model, proposed in [49], combined ResNet-50 with SVM, demonstrating an average classification accuracy, recall, and specificity of 93.33%, 90.41%, and 95.07%, respectively. According to [50], an AlexNet model that was fine-tuned achieved an average classification accuracy, recall, and specificity of 95.72%, 93.59%, and 96.78%, respectively.

However, in our proposed framework, which incorporates DenseNet201 and DenseNet-169 models, method for extracting features based on SIFT and RF classifier, the table demonstrates that we have surpassed the performance of the currently available techniques for identifying COVID-19 and pneumonia. Additionally, our proposed framework takes segmented lung images as input, unlike methods that extract features from raw CXR images.

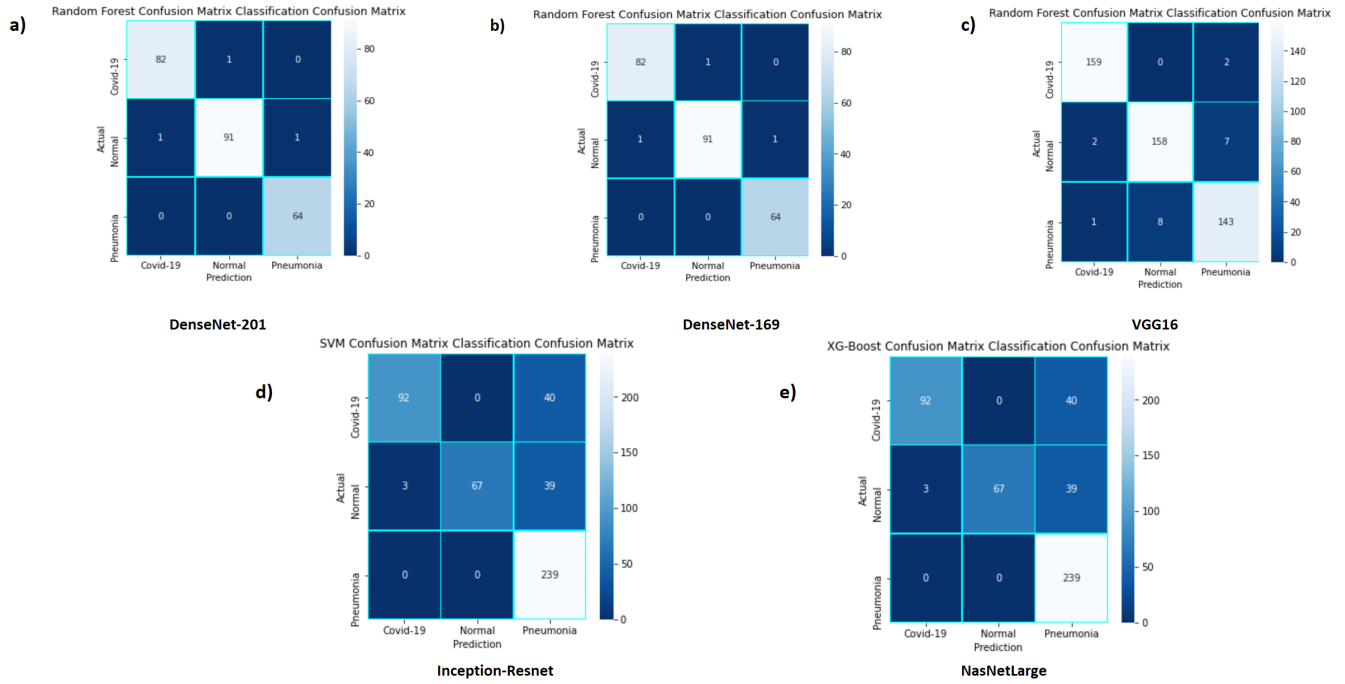

FIGURE 21: A set of confusion matrices generated for various feature extraction networks proposed in the study: (a) DenseNet-201 with RF (b) DenseNet-169 model with RF (c) VGG16-BRISK and RF (d) Inception-Resnet-SIFT-GCN and SVM (e) NasNetLarge-BRISK-GCN and XG-Boost.

TABLE 21: Performance comparison of the proposed method for CXR image classification

Authors	Method	Avg Acc (%)	Avg Sens (%)
Shankar and Perumal [48]	LBP and Inception V3	95.11%	93.15%
Apostolopoulos and Mpesiana [17]	MobileNet v2	93.33%	90.66% %
Nayak et al [22]	ResNet-34	95.29%	92.97%
Sethy and Behera [49]	ResNet-50 and SVM	93.33%	90.41%
Pham [50]	AlexNet	95.72%	93.59%
Our Proposed Method	DenseNet-169 DenseNet-201 with RF/SIFT and RF	99/98%	99/98%

B. DISCUSSION

This paper presents a novel deep learning approach to detect COVID-19 by incorporating graph-based feature reconstruction through the processes of segmenting images, extracting features, and classification.

Our hypothesis is that reconstructed features will be more meaningful if they are reconstructed by neighboring data points. Therefore, we developed a graph-based reconstruction technique. Although our approach involves generating graphs and reconstructing features, these are essential components of our method, we examined all stages of image classification, comprises of various stages, which include segmenting the image, extracting its features, reconstructing the features, and ultimately classifying it.

Regarding segmentation, the C-GAN model produced the most favorable results compared to other supervised learning methods tested. A feature extraction pipeline was built using a combination of deep CNN models and algorithms for key point extraction. According to a study [51], key point descriptor algorithms are effective in acquiring intensity information from objects in images. Therefore, to aid in the categorization of segmented lung images, we employed key point descriptors, namely SIFT and BRISK, to extract significant key intensity points. As previously noted, the model utilizes CNN architectures to extract deep features. These models have demonstrated remarkable accuracy and improved convergence with an increase in the quantity of layers. The features derived from the last layer are subsequently categorized using both a DL classifier and several ML techniques. The segmentation and classification methods proposed in this study have undergone training using datasets that are openly accessible.

The study demonstrated a comprehensive evaluation of the framework's performance, including the impact of intensity key point features and feature reconstruction on the results. The experimental findings suggest that in all cases, the deep transfer learning models outperformed the simple customized CNN model. The findings show that the technique is highly effective in distinguishing CXR images of corona infected individuals, pneumonia infected individuals, and normal in-

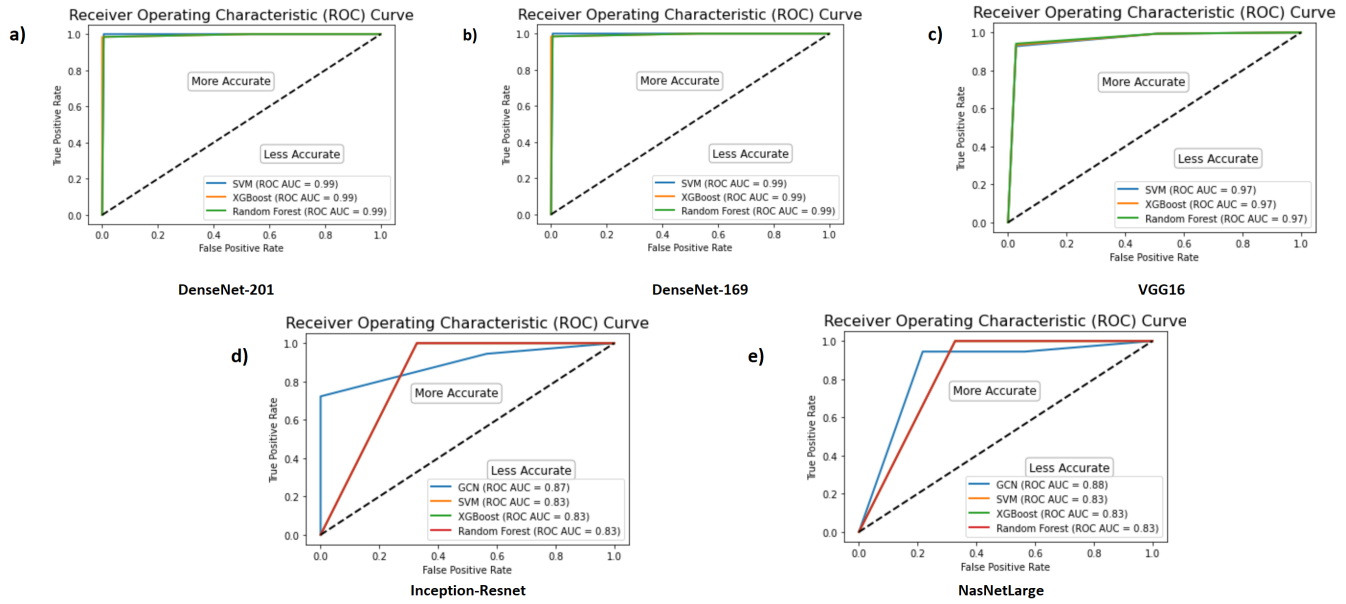

FIGURE 22: ROC Curves for the different proposed feature extraction networks: (a) DenseNet-201 model with SVM/XG-Boost/RF (b) DenseNet-169 model with SVM/XG-Boost/RF (c) VGG16-BRISK and SVM/XG-Boost/RF (d) Inception-Resnet-SIFT-GCN and SVM/XG-Boost/RF (e) NasNetLarge-BRISK-GCN and SVM/XG-Boost/RF

dividuals which can be a valuable diagnostic tool for radiologists.

To ensure the efficiency of our proposed framework, our future objectives involve validating it on a more extensive dataset that contains a greater quantity of CXR images that correspond to patients with COVID-19. Furthermore, our intention is to train the proposed model on a dataset that comprises CT scans of individuals with COVID-19, and then assess its efficacy against a model that has been trained using X-ray images.

VI. CONCLUTION AND FUTURE DIRECTIONS

A. CONCLUSION

This study presents a novel approach for detecting COVID-19, which combines graph-based feature reconstruction with image segmentation, feature extraction, and classification to increase the accuracy of identifying COVID-19, pneumonia, and normal CXR images through the use of deep learning. We propose a graph-based reconstruction method that enhances features using the information from neighboring pixels. The graph construction and feature reconstruction are critical components of our method. The segmentation of lung images was carried out using a conditional GAN algorithm's pix-to-pix approach, that has been trained utilizing CXR images having ground truth masks.

The proposed method in this study involved using a trained segmentation network for lung image segmentation, followed by a feature extraction pipeline comprised of several Deep Neural Networks (DNN), including VGG-16, VGG-19, and DenseNet-169, DenseNet 201, Inception-Resnet, simple customized CNN, NasNetLarge, and Xception and techniques for detecting key points, for instance, algorithms like SIFT and BRISK. The derived features were then reconstructed utilizing a graph-based method, which aggregated features throughout the graph. The extracted features were then fed into GCN and certain ML classifiers like softmax, RF, SVM, and XG Boost, to classify the different classes of images. The combination of DenseNet-169 and DenseNet201 models with RF obtained the maximum average classification accuracy of 99%, while VGG-16 model with BRISK and RF achieved 97% accuracy. Our approach to identifying COVID-19 by analyzing CXR images demonstrated superior performance compared to existing methods.

B. RESEARCH CONTRIBUTIONS

With the urgency to find precise and effective techniques for identifying COVID-19 on the rise, deep learning models have shown to have promising potential in COVID-19 detection. The study highlights the following contributions:

- Our framework suggests utilizing CXR images to identify COVID-19 through a segmentation model, a classification pipeline and features reconstruction technique and achieves high accuracy in detecting COVID-19.
- We were able to segment chest X-ray images efficiently by utilizing a conditional generative adversarial network (C-GAN).
- Further, we developed a feature extraction pipeline that incorporates both deep transfer learning models and

key-point features detectors like SIFT and BRISK.
- We used Stellar-Graph for creating graph from the extracted features.
- For feature reconstruction, node embedding is used for dimensionality reduction and feature reconstruction.
- Dataset of radiographs have been classified into three classes such as COVID19, pneumonia, and normal classes utilizing a DL model GCN for and various machine learning models in the classification module such as SVM, XG-Boost and RF

C. FUTURE WORK

The proposed framework provides a promising approach for COVID-19 detection, there are several potential areas for future improvement. Here are a few possibilities:

- It could benefit from a larger and more diverse dataset, which could help enhance the model's capacity to adapt to novel cases.
- While our work mainly focuses on chest X-ray images, there may be additional clinical data that could improve COVID-19 detection. For example, including information on symptoms, comorbidities, and laboratory tests could help improve the accuracy of the model.
- Our framework has shown great promise for image classification, but it can be difficult to interpret and may not provide insight into the specific features or biomarkers that are driving classification decisions. Future work could explore techniques for explainability and interpretability, to help understand what features the model is using to make its predictions.
- The potential of a method utilizing deep learning (DL) for the identification of COVID-19 in clinical settings will determine its success. Future work could focus on developing a user-friendly interface for the model and integrating it into electronic health record systems to facilitate practitioners.

References

[1] Yun Liu et al. "Rethinking computer-aided tuberculosis diagnosis". In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, pp. 2646–2655.
[2] Kai-Cai Liu et al. "CT manifestations of coronavirus disease-2019: a retrospective analysis of 73 cases by disease severity". In: European journal of radiology 126 (2020), p. 108941.
[3] Payam Behzadi, Reza Ranjbar, and Seyed Moayed Alavian. "Nucleic acidbased approaches for detection of viral hepatitis". In: Jundishapur Journal of Microbiology 8.1 (2015).
[4] Yicheng Fang et al. "Sensitivity of chest CT for COVID-19: comparison to RTPCR". In: Radiology 296.2 (2020), E115–E117.
[5] Tao Ai et al. "Correlation of chest CT and RT-PCR testing for coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases". In: Radiology 296.2 (2020), E32–E40.
[6] X Chen et al. "PIN92 pediatric bacterial pneumonia classification through chest x-rays using transfer learning". In: Value in Health 22 (2019), S209–S210.
[7] World Health Organization et al. Laboratory testing strategy recommendations for COVID-19: interim guidance, 21 March 2020. Tech. rep. World Health Organization, 2020.
[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: nature 521.7553 (2015), pp. 436–444.
[9] Dinggang Shen, Guorong Wu, and Heung-Il Suk. "Deep learning in medical image analysis". In: Annual review of biomedical engineering 19 (2017), pp. 221– 248.
[10] Geert Litjens et al. "A survey on deep learning in medical image analysis". In: Medical image analysis 42 (2017), pp. 60–88.
[11] Ken Chatfield et al. "Return of the devil in the details: Delving deep into convolutional nets". In: arXiv preprint arXiv:1405.3531 (2014).
[12] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: arXiv preprint arXiv:1409.1556 (2014).
[13] Tulin Ozturk et al. "Automated detection of COVID-19 cases using deep neural networks with X-ray images". In: Computers in biology and medicine 121 (2020), p. 103792.
[14] Linda Wang, Zhong Qiu Lin, and Alexander Wong. "Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images". In: Scientific Reports 10.1 (2020), pp. 1–12.
[15] Ezz El-Din Hemdan, Marwa A Shouman, and Mohamed Esmail Karar. "Covidxnet: A framework of deep learning classifiers to diagnose covid-19 in x-ray images". In: arXiv preprint arXiv:2003.11055 (2020).
[16] Xiang Yu, Shui-Hua Wang, and Yu-Dong Zhang. "CGNet: A graph-knowledge embedded convolutional neural network for detection of pneumonia". In: Information Processing & Management 58.1 (2021), p. 102411.
[17] Ioannis D Apostolopoulos and Tzani A Mpesiana. "Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks". In: Physical and engineering sciences in medicine 43.2 (2020), pp. 635– 640.
[18] Asif Iqbal Khan, Junaid Latief Shah, and Mohammad Mudasir Bhat. "CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images". In: Computer methods and programs in biomedicine 196 (2020), p. 105581.
[19] Mohammad Khalid Pandit et al. "Automatic detection of COVID-19 from chest radiographs using deep learning". In: Radiography 27.2 (2021), pp. 483–489.
[20] Boran Sekeroglu and Ilker Ozsahin. "<? covid19?> Detection of COVID-19 from Chest X-Ray Images Using Convolutional Neural Networks". In: SLAS TECHNOLOGY: Translating Life Sciences Innovation 25.6 (2020), pp. 553–565.
[21] Mesut Tog˘açar, Burhan Ergen, and Zafer Cömert. "COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches". In: Computers in biology and medicine 121 (2020), p. 103805.
[22] Soumya Ranjan Nayak et al. "Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: A comprehensive study". In: Biomedical Signal Processing and Control 64 (2021), p. 102365.
[23] Saman Motamed, Patrik Rogalla and Farzad Khalvati. "Data augmentation using Generative Adversarial Networks (GANs) for GAN-based detection of Pneumonia and COVID-19 in chest X-ray images". In: Informatics in medicine unlocked, 27 (2021), p. 100779.
[24] Tripti Goel, R Murugan, Seyedali Mirjalili and Deba Kumar Chakrabartty. "Automatic screening of covid-19 using an optimized generative adversarial network". In: Cognitive computation, 16(4), (2024), p. 1666-1681.
[25] Dinh C Nguyen, Ming Ding, Pubudu N Pathirana, Seneviratne, Aruna and Albert Y Zomaya. "Federated learning for COVID-19 detection with generative adversarial networks in edge cloud computing". In: IEEE Internet of Things Journal, 9(12), (2021), p. 10257-10271.
[26] Prerak Mann, Sahaj Jain, Saurabh Mittal and Aruna Bhat. "Generation of covid-19 chest CT scan images using generative adversarial networks". *in procedings of international conference on intelligent technologies (CONIT)*,(2021), p. 1-5.
[27] Tahir Sher, Abdul Rehman and Dongsun Kim. "COVID-19 Outbreak Prediction by Using Machine Learning Algorithms," In: Computers, Materials & Continua, 75(1), 2023.
[28] Ku Muhammad Naim Ku Khalif etal. "Integrated generative adversarial networks and deep convolutional neural networks for image data classification: A case study for covid-19". In: Information, 15(1), (2024), p. 58.
[29] Tasleem Kausar, Yun Lu, Adeeba Kausar, Mustajab Aliand Adnan Yousaf. "SD-GAN: A style distribution transfer generative adversarial network for Covid-19 detection through X-ray images". In: IEEE Access, 15, (2023), p. 24545-24560.
[30] Mete Ahishali etal. "R2C-GAN: Restore-to-Classify Generative Adversarial Networks for blind X-ray restoration and COVID-19 classification". In: Pattern Recognition, 156, (2024), p. 110765.

[31] Haneen Majid and Khawla Hussein Ali. "Automatic Diagnosis of Coronavirus Using Conditional Generative Adversarial Network (CGAN)". In: Iraqi Journal of Science, 156, (2023), p. 4542-4556.
[32] Seung-Jin Yoo etal. "Generative adversarial network for automatic quantification of Coronavirus disease 2019 pneumonia on chest radiographs". In: European Journal of Radiology, 164, (2023), p. 110858.
[33] Xiaojie Li etal. "CAGAN: Classifier-augmented generative adversarial networks for weakly-supervised COVID-19 lung lesion localisation". In: IET Computer Vision, 18(1), (2024), p. 1-14.
[34] Bram Van Ginneken, Mikkel B Stegmann, and Marco Loog. "Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database". In: Medical image analysis 10.1 (2006), pp. 19–40.
[35] Joseph Paul Cohen, Paul Morrison, and Lan Dao. "COVID-19 image data collection". In: arXiv preprint arXiv:2003.11597 (2020).
[36] Paul Mooney. "Chest x-ray images (pneumonia)". In: kaggle, Marzo (2018).
[37] Ian Goodfellow et al. "Generative adversarial nets". In: Advances in neural information processing systems 27 (2014).
[38] Phillip Isola et al. "Image-to-image translation with conditional adversarial networks". In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, pp. 1125–1134.
[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: International Conference on Medical image computing and computer-assisted intervention. Springer. 2015, pp. 234– 241.
[40] Gao Huang et al. "Densely connected convolutional networks". In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, pp. 4700– 4708.
[41] Yunfeng Chen et al. "Classification of lungs infected COVID-19 images based on inception-ResNet". In: Computer Methods and Programs in Biomedicine 225 (2022), p. 107053.
[42] Barret Zoph et al. "Learning transferable architectures for scalable image recognition". In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, pp. 8697–8710.
[43] François Chollet. "Xception: Deep learning with depthwise separable convolutions". In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, pp. 1251–1258.
[44] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: International journal of computer vision 60.2 (2004), pp. 91–110.
[45] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. "BRISK: Binary robust invariant scalable keypoints". In: 2011 International conference on computer vision. Ieee. 2011, pp. 2548–2555.
[46] Junji Shiraishi et al. "Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules". In: American Journal of Roentgenology 174.1 (2000), pp. 71–74.
[47] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: arXiv preprint arXiv:1609.02907 (2016).
[48] K Shankar and Eswaran Perumal. "A novel hand-crafted with deep learning features based fusion model for COVID-19 diagnosis and classification using chest X-ray images". In: Complex & Intelligent Systems 7.3 (2021), pp. 1277–1293.
[49] Prabira Kumar Sethy and Santi Kumari Behera. "Detection of coronavirus disease (covid-19) based on deep features". In: (2020).
[50] Tuan D Pham. "Classification of COVID-19 chest X-rays with deep learning: new models or fine tuning?" In: Health Information Science and Systems 9 (2021), pp. 1–11.
[51] Jin Xie et al. "Effective texture classification by texton encoding induced statistical features". In: Pattern Recognition 48.2 (2015), pp. 447–457

DR. IMRAN IHSAN received a PhD degree in Knowledge Graphs and Explainable AI from the Capital University of Science and Technology, Islamabad, in 2020. He has over 25 years of academic (teaching, project supervision, and academic administration) and industry (software, web, graphics, animation, and game development and operational management) experience. Since 2013, he has been working as an Associate Professor with the Department of Creative Technologies, Faculty of Computing and AI, Air University, Islamabad. He has authored or coauthored research work in various conferences and journals. His research interests include knowledge engineering, semantic computing, natural language processing, and computational linguistics.

DR. AZHAR IMRAN is an Assistant Professor at the Department of Creative Technologies, Faculty of Computing & Artificial Intelligence, Air University, Islamabad, Pakistan. He has completed his doctoral degree in Software Engineering from the Beijing University of Technology, China, and his master's degree in Computer Science from the University of Sargodha, Pakistan. He worked as a Senior Lecturer at the Department of Computer Science, University of Sargodha, Pakistan, from 2012 to 2017. He is a renowned expert in Image Processing, Healthcare Informatics, and Social Media Analysis. He is the Senior Member of IEEE and has contributed with 70+ research articles in well-reputed international journals and conferences. He is the editorial member and reviewer of various journals, including IEEE Access, MDPI Cancers, Applied Sciences, Mathematics, Springer Visual Computer, Talyor and Francis: Biomedical Imaging and Visualization, Multimedia Media Tools & Applications, IGI Global and Journal of Imaging, etc. Dr. Azhar has over 12 years of national and international academic experience as a Full-Time Faculty, teaching Software Engineering and core computing courses. Dr. Azhar has delivered guest talks and conducted seminars and trainings at numerous national and international forums in the past. He has contributed to multiple international conferences in diverse roles (keynote speaker, technical/ committee member, registration, speaker, etc.). His research interests include Image Processing, Social Media Analysis, Medical Image Diagnosis, Machine Learning, and Data Mining. He aims to contribute to interdisciplinary research of computer science and human-related disciplines.

MR. TAHIR SHER acquired Master of Science in Data Science from Air, University, Pakistan. He is Lecturer at Air University Islamabad, Pakistan. His career includes several academic experiences. He received his Bachelor's degree in Mathematics **(Gold Medalist)** from the International Islamic University, Islamabad, Pakistan. He also achieved **Distinction and Position Certificate** in his Bachelor's. He served in multiple positions at different training institutes as an instructor of Mathematics & statistics. Mr. Tahir Sher teaches the courses in Mathematics like Calculus, Ordinary/ Partial Differential Equations, Numerical Analysis, Linear Algebra, and Statistical & Mathematical Methods for Data Science to the students of Pakistan and other allied countries. He is also a research scholar in Explainable AI research Group at Air University, Pakistan. His research interests include the Internet of Things, Social Internet of Things, Machine Learning, Deep Learning, Natural Language Processing, Data Analysis, Time Series Analysis, Mathematical modeling for decision-making, Social Media Analysis, Federated Learning, and Computer vision. He aims to contribute to interdisciplinary research of computer science and human-related disciplines.

MAHMOOD BASIL A. AL-RAWI Msc in pharmacology and PhD in clinical pharmacy currently working in the Department of Optometry, College of Applied Medical Sciences, King Saud University, Riyadh, Saudi Arabia as an assistant professor, he has published more than 40 articles in Web of science journals. In addition to his academic duties .

MOHAMMED A. ELMELIGY received the B.Sc. degree in information technology from Menoufia University of Egypt, in 2005. He has been a Software Engineer with King Saud University, Riyadh, Saudi Arabia, since 2009. His research interests include Petri nets, supervisory control of discrete event systems, database software, and network administration

MUHAMMAD SALMAN PATHAN is an Assistant Professor at School of Computing, Dublin City University, Ireland. He received his PhD degree From Beijing University of Technology in 2015. He was a Marie Curie Research Scientist at University College Dublin in 2021. He has authored more than 40+ publication in leading journals and conferences. His research interests include machine learning, healthcare data, network security and sustainability.