www.rungta.ac.in

RUNGTA
International Journal of
Computer Science and
Information Technology

RESEARCH ARTICLE

Automated Segmentation of Chest X-ray Images: Evaluating the Efficacy of Deep Learning Architecture

Kushangi Atrey[1], Padmavati Shrivastava[1], Rahul Godara[1],#, Sakshi Sinha[1], Rohan Kumar[1], Sonal Raj[1]

Department of CSE-AI & AIML, Rungta College of Engineering and Technology, Bhilai (C.G), India- 490024

Corresponding author: kushangi.atrey@rungta.ac.in

ABSTRACT:

Precise and timely segmentation of COVID-19 affected areas in X-ray scans is essential for clinical management, disease burden estimation, and proper treatment of the disease. The need for automated and reliable methods is highlighted by the time-consuming manual approach performed by radiologists. Despite the promise shown by deep learning, particularly U-Net-based architectures, and accurately segmenting heterogeneous infection areas with often undefinable boundaries remains a challenge. This paper investigates a sophisticated deep-learning architecture that integrates residual learning ideas and attention mechanisms for automated COVID-19 infection segmentation. The study made use of the publicly available dataset on which various data augmentation techniques were applied during the training phase. Several performance metrics were evaluated for the segmented outputs including accuracy (95.79%), Dice Coefficient (83.29%), Jaccard Index (84.53%), and Area under ROC Curve (0.98). Furthermore, the Grad-CAM visualizations identified the models attention on clinically pertinent pathological areas. The results show that the proposed architecture efficiently and powerfully enables automatically segmenting COVID-19 infections from X-ray scans. This will help the radiologists with patient assessment and treatment by simplifying their clinical procedures.

KEYWORDS: COVID-19; Chest X-ray; Deep Learning; Image Segmentation; residual network.

1. Introduction

The rapid global spread of COVID-19 highlighted the critical need for rapid diagnostic tools. Chest X-rays (CXR) are a widely available imaging modality, but identifying the subtle ground-glass opacities indicative of COVID-19 is challenging. Automated segmentation using deep learning architectures offers a solution for quantifying infection severity.

Medical image computing (MIC) is the use of computational and mathematical methods for solving problems pertaining to medical images and their use for biomedical research and clinical care. It is an interdisciplinary field at the intersection of computer science, information engineering, electrical engineering, physics, mathematics and medicine [1].

The main goal of MIC is to extract clinically relevant information or knowledge from medical images. While closely related to the field of medical imaging, MIC focuses on the computational analysis of the images, not their acquisition. The methods can be grouped into several broad categories: image segmentation, image registration, image-based physiological modeling, and others [2].

Medical image computing typically operates on uniformly sampled data with regular x-y-z spatial spacing (images in 2D and volumes in 3D, generically referred to as images). At each sample point, data is commonly represented in integral form such as signed and unsigned short (16-bit), although forms from unsigned char (8-bit) to 32-bit float are not uncommon. The particular meaning of the data at the sample point depends on modality: for example a CT acquisition collects radiodensity values, while an MRI acquisition may collect T1 or T2-weighted images. Longitudinal, time-varying acquisitions may or may not acquire images with regular time steps. Fan-

Received on 15 September 2024 Revised on 29 October 2024
Accepted on 04 November 2024 Published on 16 November 2024
Rungta International Journal of Computer Science and Information Technology. 2024; 1(2): 52.

like images due to modalities such as curved-array ultrasound are also common and require different representational and algorithmic techniques to process. Other data forms include sheared images due to gantry tilt during acquisition; and unstructured meshes, such as hexahedral and tetrahedral forms, which are used in advanced biomechanical analysis (e.g., tissue deformation, vascular transport, bone implants) [3].

Segmentation is the process of partitioning an image into different meaningful segments. In medical imaging, these segments often correspond to different tissue classes, organs, pathologies, or other biologically relevant structures. Medical image segmentation is made difficult by low contrast, noise, and other imaging ambiguities. Although there are many computer vision techniques for image segmentation, some have been adapted specifically for medical image computing. Below is a sampling of techniques within this field; the implementation relies on the expertise that clinicians can provide [4].

Atlas-based segmentation: For many applications, a clinical expert can manually label several images; segmenting unseen images is a matter of extrapolating from these manually labeled training images. Methods of this style are typically referred to as atlas-based segmentation methods. Parametric atlas methods typically combine these training images into a single atlas image, while nonparametric atlas methods typically use all of the training images separately. Atlas-based methods usually require the use of image registration in order to align the atlas image or images to a new, unseen image [5].

Shape-based segmentation: Many methods parametrize a template shape for a given structure, often relying on control points along the boundary. The entire shape is then deformed to match a new image. Two of the most common shape-based techniques are active shape models and active appearance models. These methods have been very influential, and have given rise to similar models [6].

Image-based segmentation: Some methods initiate a template and refine its shape according to the image data while minimizing integral error measures, like the active contour model and its variations [7].

Interactive segmentation: Interactive methods are useful when clinicians can provide some information, such as a seed region or rough outline of the region to segment. An algorithm can then iteratively refine such a segmentation, with or without guidance from the clinician. Manual segmentation, using tools such as a paint brush to explicitly define the tissue class of each pixel, remains the gold standard for many imaging applications. Recently, principles from feedback control theory have been incorporated into segmentation, which give the user much greater flexibility and allow for the automatic correction of errors [8].

Subjective surface segmentation: This method is based on the idea of evolution of segmentation function which is governed by an advection-diffusion model. To segment an object, a segmentation seed is needed (that is the starting point that determines the approximate position of the object in the image). Consequently, an initial segmentation function is constructed. The idea behind the subjective surface method is that the position of the seed is the main factor determining the form of this segmentation function [9].

Convolutional neural networks (CNNs): The computer-assisted fully automated segmentation performance has been improved due to the advancement of machine learning models. CNN based models such as SegNet, UNet, ResNet, AATSN, Transformers and GANs have fastened the segmentation process. In the future, such models may replace manual segmentation due to their superior performance and speed [10].

There are other classifications of image segmentation methods that are similar to categories above. Another group, which is based on combination of methods, can be classified as ”hybrid” [11].

Image registration is a process that searches for the correct alignment of images. In the simplest case, two images are aligned. Typically, one image is treated as the target image and the other is treated as a source image; the source image is transformed to match the target image. The optimization procedure updates the transformation of the source image based on a similarity value that evaluates the current quality of the alignment. This iterative procedure is repeated until a (local) optimum is found. An example is the registration of CT and PET images to combine structural and metabolic information (see figure) [12].

Studying temporal changes. Longitudinal studies acquire images over several months or years to study long-term processes, such as disease progression. Time series correspond to images acquired within the same session (seconds or minutes). They can be used to study cognitive processes, heart deformations and respiration [13].

Combining complementary information from different imaging modalities. An example is the fusion of anatomical and functional information. Since the size and shape of structures vary across modalities, it is more challenging to evaluate the alignment quality. This has led to the use of similarity measures such as mutual information [14].

Characterizing a population of subjects. In contrast to intra-subject registration, a one-to-one mapping may not exist between subjects, depending on the structural variability of the organ of interest. Inter-subject registration is required for atlas construction in computational anatomy. Here, the objective is to statistically model the anatomy of organs across subjects [15].

Computer-assisted surgery. In computer-assisted surgery pre-operative images such as CT or MRI are registered to intra-operative images or tracking systems to facilitate image guidance or navigation [16].

The transformation model. Common choices are rigid, affine, and deformable transformation models. B-spline and thin plate spline models are commonly used for parameterized transformation fields. Non-parametric or dense deformation

fields carry a displacement vector at every grid location; this necessitates additional regularization constraints. A specific class of deformation fields are diffeomorphisms, which are invertible transformations with a smooth inverse [17].

The similarity metric. A distance or similarity function is used to quantify the registration quality. This similarity can be calculated either on the original images or on features extracted from the images. Common similarity measures are sum of squared distances (SSD), correlation coefficient, and mutual information. The choice of similarity measure depends on whether the images are from the same modality; the acquisition noise can also play a role in this decision. For example, SSD is the optimal similarity measure for images of the same modality with Gaussian noise. However, the image statistics in ultrasound are significantly different from Gaussian noise, leading to the introduction of ultrasound specific similarity measures. Multi-modal registration requires a more sophisticated similarity measure; alternatively, a different image representation can be used, such as structural representations or registering adjacent anatomy. A 2020 study employed contrastive coding to learn shared, dense image representations, referred to as contrastive multi-modal image representations (CoMIRs), which enabled the registration of multi-modal images where existing registration methods often fail due to a lack of sufficiently similar image structures. It reduced the multi-modal registration problem to a mono-modal one, in which general intensity based, as well as feature-based, registration algorithms can be applied [18].

2. Related Work

Medical image segmentation has been revolutionized by the U-Net architecture. Enhancements to U-Net, such as incorporating residual blocks (ResUNet) and attention mechanisms (Attention U-Net), have been developed to capture fine-grained details and contextual information. These models have been previously applied to lung nodule detection and pneumonia classification.

The optimization procedure. Either continuous or discrete optimization is performed. For continuous optimization, gradient-based optimization techniques are applied to improve the convergence speed [19].

Visualization plays several key roles in medical image computing. Methods from scientific visualization are used to understand and communicate about medical images, which are inherently spatial-temporal. Data visualization and data analysis are used on unstructured data forms, for example when evaluating statistical measures derived during algorithmic processing. Direct interaction with data, a key feature of the visualization process, is used to perform visual queries about data, annotate images, guide segmentation and registration processes, and control the visual representation of data (by controlling lighting rendering properties and viewing parameters). Visualization is used both for initial exploration and for conveying intermediate and final results of analyses [20].

The figure "Visualization of Medical Imaging" illustrates several types of visualization: 1. the display of cross-sections as gray scale images; 2. reformatted views of gray scale images (the sagittal view in this example has a different orientation than the original direction of the image acquisition; and 3. A 3D volume rendering of the same data. The nodular lesion is clearly visible in the different presentations and has been annotated with a white line [1].

Medical images can vary significantly across individuals due to people having organs of different shapes and sizes. Therefore, representing medical images to account for this variability is crucial. A popular approach to represent medical images is through the use of one or more atlases. Here, an atlas refers to a specific model for a population of images with parameters that are learned from a training dataset [2].

The simplest example of an atlas is a mean intensity image, commonly referred to as a template. However, an atlas can also include richer information, such as local image statistics and the probability that a particular spatial location has a certain label. New medical images, which are not used during training, can be mapped to an atlas, which has been tailored to the specific application, such as segmentation and group analysis. Mapping an image to an atlas usually involves registering the image and the atlas. This deformation can be used to address variability in medical images [3].

The simplest approach is to model medical images as deformed versions of a single template image. For example, anatomical MRI brain scans are often mapped to the MNI template as to represent all the brain scans in common coordinates. The main drawback of a single-template approach is that if there are significant differences between the template and a given test image, then there may not be a good way to map one onto the other. For example, an anatomical MRI brain scan of a patient with severe brain abnormalities (i.e., a tumor or surgical procedure), may not easily map to the MNI template [4].

Rather than relying on a single template, multiple templates can be used. The idea is to represent an image as a deformed version of one of the templates. For example, there could be one template for a healthy population and one template for a diseased population. However, in many applications, it is not clear how many templates are needed. A simple albeit computationally expensive way to deal with this is to have every image in a training dataset be a template image and thus every new image encountered is compared against every image in the training dataset. A more recent approach automatically finds the number of templates needed [5].

Statistical methods combine the medical imaging field with modern computer vision, machine learning and pattern recognition. Over the last decade, several large datasets have been made publicly available (see for example ADNI, 1000 functional Connectomes Project), in part due to collaboration between various institutes and research centers. This increase

in data size calls for new algorithms that can mine and detect subtle changes in the images to address clinical questions. Such clinical questions are very diverse and include group analysis, imaging biomarkers, disease phenotyping and longitudinal studies [6].

In the group analysis, the objective is to detect and quantize abnormalities induced by a disease by comparing the images of two or more cohorts. Usually one of these cohorts consist of normal (control) subjects, and the other one consists of abnormal patients. Variation caused by the disease can manifest itself as abnormal deformation of anatomy (see voxel-based morphometry). For example, shrinkage of subcortical tissues such as the hippocampus in brain may be linked to Alzheimer's disease. Additionally, changes in biochemical (functional) activity can be observed using imaging modalities such as positron emission tomography [7].

The comparison between groups is usually conducted on the voxel level. Hence, the most popular pre-processing pipeline, particularly in neuroimaging, transforms all of the images in a dataset to a common coordinate frame via medical image registration in order to maintain correspondence between voxels. Given this voxel-wise correspondence, the most common frequentist method is to extract a statistic for each voxel (for example, the mean voxel intensity for each group) and perform statistical hypothesis testing to evaluate whether a null hypothesis is or is not supported. The null hypothesis typically assumes that the two cohorts are drawn from the same distribution, and hence, should have the same statistical properties (for example, the mean values of two groups are equal for a particular voxel). Since medical images contain large numbers of voxels, the issue of multiple comparison needs to be addressed,. There are also Bayesian approaches to tackle group analysis problem [8].

Although group analysis can quantify the general effects of a pathology on an anatomy and function, it does not provide subject level measures, and hence cannot be used as biomarkers for diagnosis (see Imaging biomarkers). Clinicians, on the other hand, are often interested in early diagnosis of the pathology (i.e. classification,) and in learning the progression of a disease (i.e. regression). From methodological point of view, current techniques varies from applying standard machine learning algorithms to medical imaging datasets (e.g. support vector machine), to developing new approaches adapted for the needs of the field. The main difficulties are as follows: [9]

Small sample size (curse of dimensionality): a large medical imaging dataset contains hundreds to thousands of images, whereas the number of voxels in a typical volumetric image can easily go beyond millions. A remedy to this problem is to reduce the number of features in an informative sense (see dimensionality reduction). Several unsupervised and semi-/supervised, approaches have been proposed to address this issue [10].

Interpretability: A good generalization accuracy is not always the primary objective, as clinicians would like to understand which parts of anatomy are affected by the disease. Therefore, interpretability of the results is very important; methods that ignore the image structure are not favored. Alternative methods based on feature selection have been proposed, [11].

Image-based pattern classification methods typically assume that the neurological effects of a disease are distinct and well defined. This may not always be the case. For a number of medical conditions, the patient populations are highly heterogeneous, and further categorization into sub-conditions has not been established. Additionally, some diseases (e.g., autism spectrum disorder, schizophrenia, mild cognitive impairment can be characterized by a continuous or nearly-continuous spectra from mild cognitive impairment to very pronounced pathological changes. To facilitate image-based analysis of heterogeneous disorders, methodological alternatives to pattern classification have been developed. These techniques borrow ideas from high-dimensional clustering and high-dimensional pattern-regression to cluster a given population into homogeneous sub-populations. The goal is to provide a better quantitative understanding of the disease within each sub-population [12].

Shape analysis is the field of medical image computing that studies geometrical properties of structures obtained from different imaging modalities. Shape analysis recently become of increasing interest to the medical community due to its potential to precisely locate morphological changes between different populations of structures, i.e. healthy vs pathological, female vs male, young vs elderly. Shape analysis includes two main steps: shape correspondence and statistical analysis [13].

Shape correspondence is the methodology that computes correspondent locations between geometric shapes represented by triangle meshes, contours, point sets or volumetric images. Obviously definition of correspondence will influence directly the analysis. Among the different options for correspondence frameworks are: anatomical correspondence, manual landmarks, functional correspondence (i.e. in brain morphometry locus responsible for same neuronal functionality), geometry correspondence, (for image volumes) intensity similarity, etc. Some approaches, e.g. spectral shape analysis, do not require correspondence but compare shape descriptors directly [14].

In longitudinal studies the same person is imaged repeatedly. This information can be incorporated both into the image analysis, as well as into the statistical modeling [15].

In longitudinal image processing, segmentation and analysis methods of individual time points are informed and regularized with common information usually from a within-subject template. This regularization is designed to reduce measurement noise and thus helps increase sensitivity and statistical power. At the same time over-regularization needs to be avoided, so that effect sizes remain stable. Intense regularization, for example, can lead to excellent test-retest reliabil-

ity, but limits the ability to detect any true changes and differences across groups. Often a trade-off needs to be aimed for, that optimizes noise reduction at the cost of limited effect size loss. Another common challenge in longitudinal image processing is the, often unintentional, introduction of processing bias. When, for example, follow-up images get registered and resampled to the baseline image, interpolation artifacts get introduced to only the follow-up images and not the baseline. These artifact can cause spurious effects (usually a bias towards overestimating longitudinal change and thus underestimating required sample size). It is therefore essential that all-time points get treated exactly the same to avoid any processing bias [16].

Post-processing and statistical analysis of longitudinal data usually requires dedicated statistical tools such as repeated measure ANOVA or the more powerful linear mixed effects models. Additionally, it is advantageous to consider the spatial distribution of the signal. For example, cortical thickness measurements will show a correlation within-subject across time and also within a neighborhood on the cortical surface - a fact that can be used to increase statistical power. Furthermore, time-to-event (aka survival) analysis is frequently employed to analyze longitudinal data and determine significant predictors [17].

Traditionally, medical image computing has seen to address the quantification and fusion of structural or functional information available at the point and time of image acquisition. In this regard, it can be seen as quantitative sensing of the underlying anatomical, physical or physiological processes. However, over the last few years, there has been a growing interest in the predictive assessment of disease or therapy course. Image-based modelling, be it of biomechanical or physiological nature, can therefore extend the possibilities of image computing from a descriptive to a predictive angle [18].

According to the STEP research roadmap, the Virtual Physiological Human (VPH) is a methodological and technological framework that, once established, will enable the investigation of the human body as a single complex system. Underlying the VPH concept, the International Union for Physiological Sciences (IUPS) has been sponsoring the IUPS Physiome Project for more than a decade,. This is a worldwide public domain effort to provide a computational framework for understanding human physiology. It aims at developing integrative models at all levels of biological organization, from genes to the whole organisms via gene regulatory networks, protein pathways, integrative cell functions, and tissue and whole organ structure/function relations. Such an approach aims at transforming current practice in medicine and underpins a new era of computational medicine [19].

In this context, medical imaging and image computing play an increasingly important role as they provide systems and methods to image, quantify and fuse both structural and functional information about the human being in vivo. These two broad research areas include the transformation of generic computational models to represent specific subjects, thus paving the way for personalized computational models. Individualization of generic computational models through imaging can be realized in three complementary directions: [20]

In addition, imaging also plays a pivotal role in the evaluation and validation of such models both in humans and in animal models, and in the translation of models to the clinical setting with both diagnostic and therapeutic applications. In this specific context, molecular, biological, and pre-clinical imaging render additional data and understanding of basic structure and function in molecules, cells, tissues and animal models that may be transferred to human physiology where appropriate [1].

The applications of image-based VPH/physiome models in basic and clinical domains are vast. Broadly speaking, they promise to become new virtual imaging techniques. Effectively more, often non-observable, parameters will be imaged in silico based on the integration of observable but sometimes sparse and inconsistent multimodal images and physiological measurements. Computational models will serve to engender interpretation of the measurements in a way compliant with the underlying biophysical, biochemical or biological laws of the physiological or pathophysiological processes under investigation. Ultimately, such investigative tools and systems will help our understanding of disease processes, the natural history of disease evolution, and the influence on the course of a disease of pharmacological and/or interventional therapeutic procedures [2].

Cross-fertilization between imaging and modelling goes beyond interpretation of measurements in a way consistent with physiology. Image-based patient-specific modelling, combined with models of medical devices and pharmacological therapies, opens the way to predictive imaging whereby one will be able to understand, plan and optimize such interventions in silico [3].

3. Methodology

Our methodology implements an Attention-ResUNet architecture. The residual connections mitigate the vanishing gradient problem in deep networks, while the attention gates suppress irrelevant background regions (such as bones and heart) and highlight the infected lung tissue. The network is trained using a composite loss function combining Dice loss and Binary Cross-Entropy.

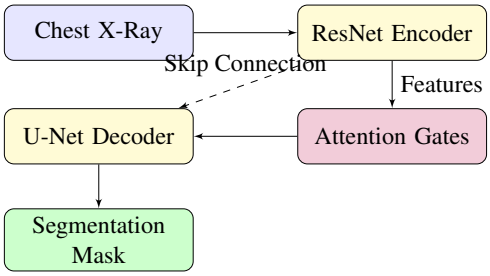

Figure 1: Attention-ResUNet Architecture for Segmentation

Algorithm 1 Attention Gate Mechanism

Require: Gating Signal g, Input Feature Map x
Ensure: Attention-filtered Feature Map $\hat{x}$
$W_g \leftarrow \text{Conv1x1}(g)$
$W_x \leftarrow \text{Conv1x1}(x)$
$Att \leftarrow \text{ReLU}(W_g + W_x)$
$Att_{map} \leftarrow \text{Sigmoid}(\text{Conv1x1}(Att))$
$\hat{x} \leftarrow x \otimes Att_{map}$ ▷ Element-wise multiplication
return $\hat{x}$

implemented in various software packages. These include approaches based on partial differential equations (PDEs) and curvature driven flows for enhancement, segmentation, and registration. Since they employ PDEs, the methods are amenable to parallelization and implementation on GPGPUs. A number of these techniques have been inspired from ideas in optimal control. Accordingly, very recently ideas from control have recently made their way into interactive methods, especially segmentation. Moreover, because of noise and the need for statistical estimation techniques for more dynamically changing imagery, the Kalman filter and particle filter have come into use. A survey of these methods with an extensive list of references may be found in [4].

Some imaging modalities provide very specialized information. The resulting images cannot be treated as regular scalar images and give rise to new sub-areas of medical image computing. Examples include diffusion MRI and functional MRI [5].

Diffusion MRI is a structural magnetic resonance imaging modality that allows measurement of the diffusion process of molecules. Diffusion is measured by applying a gradient pulse to a magnetic field along a particular direction. In a typical acquisition, a set of uniformly distributed gradient directions is used to create a set of diffusion weighted volumes. In addition, an unweighted volume is acquired under the same magnetic field without application of a gradient pulse. As each acquisition is associated with multiple volumes, diffusion MRI has created a variety of unique challenges in medical image computing [6].

The diffusion tensor, a 3 3 symmetric positive-definite matrix, offers a straightforward solution to both of these goals. It is proportional to the covariance matrix of a Normally distributed local diffusion profile and, thus, the dominant eigenvector of this matrix is the principal direction of local diffusion. Due to the simplicity of this model, a maximum likelihood estimate of the diffusion tensor can be found by simply solving a system of linear equations at each location independently. However, as the volume is assumed to contain contiguous tissue fibers, it may be preferable to estimate the volume of diffusion tensors in its entirety by imposing regularity conditions on the underlying field of tensors. Scalar values can be extracted from the diffusion tensor, such as the fractional anisotropy, mean, axial and radial diffusivities, which indirectly measure tissue properties such as the dysmyelination of axonal fibers or the presence of edema. Standard scalar image computing methods, such as registration and segmentation, can be applied directly to volumes of such scalar values. However, to fully exploit the information in the diffusion tensor, these methods have been adapted to account for tensor valued volumes when performing registration and segmentation [7].

Given the principal direction of diffusion at each location in the volume, it is possible to estimate the global pathways of diffusion through a process known as tractography. However, due to the relatively low resolution of diffusion MRI, many of these pathways may cross, kiss or fan at a single location. In this situation, the single principal direction of the diffusion tensor is not an appropriate model for the local diffusion distribution. The most common solution to this problem is to estimate multiple directions of local diffusion using more complex models. These include mixtures of diffusion tensors, Q-ball imaging, diffusion spectrum imaging and fiber orientation distribution functions, which typically require HARDI acquisition with a large number of gradient directions. As with the diffusion tensor, volumes valued with these complex models require special treatment when applying image computing methods, such as registration and segmentation [8].

Functional magnetic resonance imaging (fMRI) is a medical imaging modality that indirectly measures neural activity by observing the local hemodynamics, or blood oxygen level dependent signal (BOLD). fMRI data offers a range of insights, and can be roughly divided into two categories: [9]

Task related fMRI is acquired as the subject is performing a sequence of timed experimental conditions. In block-design experiments, the conditions are present for short periods of time (e.g., 10 seconds) and are alternated with periods of rest. Event-related experiments rely on a random sequence of stimuli and use a single time point to denote each condition. The standard approach to analyze task related fMRI is the general linear model (GLM) [10].

Resting state fMRI is acquired in the absence of any experimental task. Typically, the objective is to study the intrinsic network structure of the brain. Observations made during rest have also been linked to specific cognitive processes such as encoding or reflection. Most studies of resting state

fMRI focus on low frequency fluctuations of the fMRI signal (LF-BOLD). Seminal discoveries include the default network, a comprehensive cortical parcellation, and the linking of network characteristics to behavioral parameters [11].

There is a rich set of methodology used to analyze functional neuroimaging data, and there is often no consensus regarding the best method. Instead, researchers approach each problem independently and select a suitable model/algorithm. In this context there is a relatively active exchange among neuroscience, computational biology, statistics, and machine learning communities. Prominent approaches include [12]

Massive univariate approaches that probe individual voxels in the imaging data for a relationship to the experiment condition. The prime approach is the general linear model [13].

Multivariate- and classifier based approaches, often referred to as multi voxel pattern analysis or multi-variate pattern analysis probe the data for global and potentially distributed responses to an experimental condition. Early approaches used support vector machines to study responses to visual stimuli. Recently, alternative pattern recognition algorithms have been explored, such as random forest based gini contrast or sparse regression and dictionary learning [14].

Functional connectivity analysis studies the intrinsic network structure of the brain, including the interactions between regions. The majority of such studies focus on resting state data to parcelate the brain or to find correlates to behavioral measures. Task specific data can be used to study causal relationships among brain regions (e.g., dynamic causal mapping) [15].

When working with large cohorts of subjects, the normalization (registration) of individual subjects into a common reference frame is crucial. A body of work and tools exist to perform normalization based on anatomy (FSL, FreeSurfer, SPM). Alignment taking spatial variability across subjects into account is a more recent line of work. Examples are the alignment of the cortex based on fMRI signal correlation, the alignment based on the global functional connectivity structure both in task-, or resting state data, and the alignment based on stimulus specific activation profiles of individual voxels [16].

Software for medical image computing is a complex combination of systems providing IO, visualization and interaction, user interface, data management and computation. Typically system architectures are layered to serve algorithm developers, application developers, and users. The bottom layers are often libraries and/or toolkits which provide base computational capabilities; while the top layers are specialized applications which address specific medical problems, diseases, or body systems [17].

Medical image computing is also related to the field of computer vision. An international society, The MICCAI Society, represents the field and organizes an annual conference and associated workshops. Proceedings for this conference are published by Springer in the Lecture Notes in Computer Science series. In 2000, N. Ayache and J. Duncan reviewed the state of the field [18].

Medical Image Analysis (MedIA) ; also the official journal of The MICCAI Society, which organizes the Annual MICCAI Conference a premier conference for medical image computing [19]

In addition the following journals occasionally publish articles describing methods and specific clinical applications of medical image computing or modality specific medical image computing [20]

A convolutional neural network (CNN) is a type of feedforward neural network that learns features via filter (or kernel) optimization. This type of deep learning network has been applied to process and make predictions from many different types of data including text, images and audio. CNNs are the de-facto standard in deep learning-based approaches to computer vision and image processing, and have only recently been replaced-in some cases-by newer architectures such as the transformer [1].

Vanishing gradients and exploding gradients, seen during backpropagation in earlier neural networks, are prevented by the regularization that comes from using shared weights over fewer connections. For example, for each neuron in the fully-connected layer, 10,000 weights would be required for processing an image sized 100 100 pixels. However, applying cascaded convolution (or cross-correlation) kernels, only 25 weights for each convolutional layer are required to process 5x5-sized tiles. Higher-layer features are extracted from wider context windows, compared to lower-layer features [2].

CNNs are also known as shift invariant or space invariant artificial neural networks, based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation-equivariant responses known as feature maps. Counter-intuitively, most convolutional neural networks are not invariant to translation, due to the downsampling operation they apply to the input [3].

Feedforward neural networks are usually fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The ”full connectivity” of these networks makes them prone to overfitting data. Typical ways of regularization, or preventing overfitting, include: penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.) Robust datasets also increase the probability that CNNs will learn the generalized principles that characterize a given dataset rather than the biases of a poorly-populated set [4].

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field [5].

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the net-

work learns to optimize the filters (or kernels) through automated learning, whereas in traditional algorithms these filters are hand-engineered. This simplifies and automates the process, enhancing efficiency and scalability overcoming human-intervention bottlenecks [6].

A convolutional neural network consists of an input layer, hidden layers and an output layer. In a convolutional neural network, the hidden layers include one or more layers that perform convolutions. Typically this includes a layer that performs a dot product of the convolution kernel with the layer's input matrix. This product is usually the Frobenius inner product, and its activation function is commonly ReLU. As the convolution kernel slides along the input matrix for the layer, the convolution operation generates a feature map, which in turn contributes to the input of the next layer. This is followed by other layers such as pooling layers, fully connected layers, and normalization layers [7].

Convolutional layers convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a specific stimulus. Each convolutional neuron processes data only for its receptive field [8].

Although fully connected feedforward neural networks can be used to learn features and classify data, this architecture is generally impractical for larger inputs (e.g., high-resolution images), which would require massive numbers of neurons because each pixel is a relevant input feature. A fully connected layer for an image of size 100 100 has 10,000 weights for each neuron in the second layer. Convolution reduces the number of free parameters, allowing the network to be deeper. For example, using a 5 5 tiling region, each with the same shared weights, requires only 25 neurons. Using shared weights means there are many fewer parameters, which helps avoid the vanishing gradients and exploding gradients problems seen during backpropagation in earlier neural networks [9].

To speed processing, standard convolutional layers can be replaced by depthwise separable convolutional layers, which are based on a depthwise convolution followed by a pointwise convolution. The depthwise convolution is a spatial convolution applied independently over each channel of the input tensor, while the pointwise convolution is a standard convolution restricted to the use of [10]

Convolutional networks may include local and/or global pooling layers along with traditional convolutional layers. Pooling layers reduce the dimensions of data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, tiling sizes such as 2 2 are commonly used. Global pooling acts on all the neurons of the feature map. There are two common types of pooling in popular use: max and average. Max pooling uses the maximum value of each local cluster of neurons in the feature map, while average pooling takes the average value [11].

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is the same as a traditional multilayer perceptron neural network (MLP). Each neuron in the fully connected layer receives input from all the neurons in the previous layer. These inputs are weighted and summed with the corresponding biases, and then passed through an activation function to perform a nonlinear transformation, generating the output. The flattened matrix goes through a fully connected layer to classify the images [12].

In neural networks, each neuron receives input from some number of locations in the previous layer. In a convolutional layer, each neuron receives input from only a restricted area of the previous layer called the neuron's receptive field. Typically the area is a square (e.g. 5 by 5 neurons). Whereas, in a fully connected layer, the receptive field is the entire previous layer. Thus, in each convolutional layer, each neuron takes input from a larger area in the input than previous layers. This is due to applying the convolution over and over, which takes the value of a pixel into account, as well as its surrounding pixels. When using dilated layers, the number of pixels in the receptive field remains constant, but the field is more sparsely populated as its dimensions grow when combining the effect of several layers [13].

To manipulate the receptive field size as desired, there are some alternatives to the standard convolutional layer. For example, atrous or dilated convolution expands the receptive field size without increasing the number of parameters by interleaving visible and blind regions. Moreover, a single dilated convolutional layer can comprise filters with multiple dilation ratios, thus having a variable receptive field size [14].

Each neuron in a neural network computes an output value by applying a specific function to the input values received from the receptive field in the previous layer. The function that is applied to the input values is determined by a vector of weights and a bias (typically real numbers). Learning consists of iteratively adjusting these biases and weights [15].

The vectors of weights and biases are called filters and represent particular features of the input (e.g., a particular shape). A distinguishing feature of CNNs is that many neurons can share the same filter. This reduces the memory footprint because a single bias and a single vector of weights are used across all receptive fields that share that filter, as opposed to each receptive field having its own bias and vector weighting [16].

A deconvolutional layer is the transpose of a convolutional layer. Specifically, a convolutional layer can be written as a multiplication with a matrix, and a deconvolutional layer is multiplication with the transpose of that matrix [17].

An unpooling layer expands the layer. The max-unpooling layer is the simplest, as it simply copies each entry multiple times. For example, a 2-by-2 max-unpooling layer is [18]

Work by Hubel and Wiesel in the 1950s and 1960s showed that cat visual cortices contain neurons that individually respond to small regions of the visual field. Provided the eyes are not moving, the region of visual space within which visual stimuli affect the firing of a single neuron is known as its re-

ceptive field. Neighboring cells have similar and overlapping receptive fields. Receptive field size and location varies systematically across the cortex to form a complete map of visual space. The cortex in each hemisphere represents the contralateral visual field [19].

In 1969, Kunihiko Fukushima introduced a multilayer visual feature detection network, inspired by the above-mentioned work of Hubel and Wiesel, in which "All the elements in one layer have the same set of interconnecting coefficients; the arrangement of the elements and their interconnections are all homogeneous over a given layer." This is the essential core of a convolutional network, but the weights were not trained. In the same paper, Fukushima also introduced the ReLU (rectified linear unit) activation function [20].

"S-layer": a shared-weights receptive-field layer, later known as a convolutional layer, which contains units whose receptive fields cover a patch of the previous layer. A shared-weights receptive-field group (a "plane" in neocognitron terminology) is often called a filter, and a layer typically has several such filters [1].

"C-layer": a downsampling layer that contain units whose receptive fields cover patches of previous convolutional layers. Such a unit typically computes a weighted average of the activations of the units in its patch, and applies inhibition (divisive normalization) pooled from a somewhat larger patch and across different filters in a layer, and applies a saturating activation function. The patch weights are nonnegative and are not trainable in the original neocognitron. The downsampling and competitive inhibition help to classify features and objects in visual scenes even when the objects are shifted [2].

Several supervised and unsupervised learning algorithms have been proposed over the decades to train the weights of a neocognitron. Today, however, the CNN architecture is usually trained through backpropagation [3].

Fukushima's ReLU activation function was not used in his neocognitron since all the weights were nonnegative; lateral inhibition was used instead. The rectifier has become a very popular activation function for CNNs and deep neural networks in general [4].

The term "convolution" first appears in neural networks in a paper by Toshiteru Homma, Les Atlas, and Robert Marks II at the first Conference on Neural Information Processing Systems in 1987. Their paper replaced multiplication with convolution in time, inherently providing shift invariance, motivated by and connecting more directly to the signal-processing concept of a filter, and demonstrated it on a speech recognition task. They also pointed out that as a data-trainable system, convolution is essentially equivalent to correlation since reversal of the weights does not affect the final learned function ("For convenience, we denote * as correlation instead of convolution. Note that convolving a(t) with b(t) is equivalent to correlating a(-t) with b(t)."). Modern CNN implementations typically do correlation and call it convolution, for convenience, as they did here [5].

The time delay neural network (TDNN) was introduced in 1987 by Alex Waibel et al. for phoneme recognition and was an early convolutional network exhibiting shift-invariance. A TDNN is a 1-D convolutional neural net where the convolution is performed along the time axis of the data. It is the first CNN utilizing weight sharing in combination with a training by gradient descent, using backpropagation. Thus, while also using a pyramidal structure as in the neocognitron, it performed a global optimization of the weights instead of a local one [6].

TDNNs are convolutional networks that share weights along the temporal dimension. They allow speech signals to be processed time-invariantly. In 1990 Hampshire and Waibel introduced a variant that performs a two-dimensional convolution. Since these TDNNs operated on spectrograms, the resulting phoneme recognition system was invariant to both time and frequency shifts, as with images processed by a neocognitron [7].

Denker et al. (1989) designed a 2-D CNN system to recognize hand-written ZIP Code numbers. However, the lack of an efficient training method to determine the kernel coefficients of the involved convolutions meant that all the coefficients had to be laboriously hand-designed [8].

Following the advances in the training of 1-D CNNs by Waibel et al. (1987), Yann LeCun et al. (1989) used back-propagation to learn the convolution kernel coefficients directly from images of hand-written numbers. Learning was thus fully automatic, performed better than manual coefficient design, and was suited to a broader range of image recognition problems and image types [9].

Wei Zhang et al. (1988) used back-propagation to train the convolution kernels of a CNN for alphabets recognition. The model was called shift-invariant pattern recognition neural network before the name CNN was coined later in the early 1990s. Wei Zhang et al. also applied the same CNN without the last fully connected layer for medical image object segmentation (1991) and breast cancer detection in mammograms (1994) [10].

In 1990 Yamaguchi et al. introduced the concept of max pooling, a fixed filtering operation that calculates and propagates the maximum value of a given region. They did so by combining TDNNs with max pooling to realize a speaker-independent isolated word recognition system. In their system they used several TDNNs per word, one for each syllable. The results of each TDNN over the input signal were combined using max pooling and the outputs of the pooling layers were then passed on to networks performing the actual word classification [11].

In a variant of the neocognitron called the cresceptron, instead of using Fukushima's spatial averaging with inhibition and saturation, J. Weng et al. in 1993 used max pooling, where a downsampling unit computes the maximum of the activations of the units in its patch, introducing this method into the vision field [12].

LeNet-5, a pioneering 7-level convolutional network by

LeCun et al. in 1995, classifies hand-written numbers on checks digitized in 3232 pixel images. The ability to process higher-resolution images requires larger and more layers of convolutional neural networks, so this technique is constrained by the availability of computing resources [13].

It was superior than other commercial courtesy amount reading systems (as of 1995). The system was integrated in NCR's check reading systems, and fielded in several American banks since June 1996, reading millions of checks per day [14].

A shift-invariant neural network was proposed by Wei Zhang et al. for image character recognition in 1988. It is a modified Neocognitron by keeping only the convolutional interconnections between the image feature layers and the last fully connected layer. The model was trained with backpropagation. The training algorithm was further improved in 1991 to improve its generalization ability. The model architecture was modified by removing the last fully connected layer and applied for medical image segmentation (1991) and automatic detection of breast cancer in mammograms (1994) [15].

A different convolution-based design was proposed in 1988 for application to decomposition of one-dimensional electromyography convolved signals via de-convolution. This design was modified in 1989 to other de-convolution-based designs [16].

In 2004, it was shown by K. S. Oh and K. Jung that standard neural networks can be greatly accelerated on GPUs. Their implementation was 20 times faster than an equivalent implementation on CPU. In 2005, another paper also emphasised the value of GPGPU for machine learning [17].

The first GPU-implementation of a CNN was described in 2006 by K. Chellapilla et al. Their implementation was 4 times faster than an equivalent implementation on CPU. In the same period, GPUs were also used for unsupervised training of deep belief networks [18].

In 2010, Dan Ciresan et al. at IDSIA trained deep feedforward networks on GPUs. In 2011, they extended this to CNNs, accelerating by 60 compared to training CPU. In 2011, the network won an image recognition contest where they achieved superhuman performance for the first time. Then they won more competitions and achieved state of the art on several benchmarks [19].

Subsequently, AlexNet, a similar GPU-based CNN by Alex Krizhevsky et al. won the ImageNet Large Scale Visual Recognition Challenge 2012. It was an early catalytic event for the AI boom [20].

4. Results and Discussion

Evaluated on a public COVID-19 CXR dataset, the Attention-ResUNet model achieved a pixel-level accuracy of 95.79%. The Dice Coefficient of 83.29% indicates strong spatial overlap between the predicted segmentations and ground truth annotations provided by expert radiologists. Grad-CAM analysis confirmed the network's focus on relevant pulmonary opacities.

Table 1: Segmentation Accuracy Metrics

Architecture	**Dice Coefficient**	**Jaccard Index**	**Accuracy**
Standard U-Net	74.5%	76.2%	88.4%
ResUNet	79.1%	80.5%	92.1%
Attention-ResUNet	**83.29%**	**84.53%**	**95.79%**

Compared to the training of CNNs using GPUs, not much attention was given to CPU. (Viebke et al 2019) parallelizes CNN by thread- and SIMD-level parallelism that is available on the Intel Xeon Phi [1].

In the past, traditional multilayer perceptron (MLP) models were used for image recognition. However, the full connectivity between nodes caused the curse of dimensionality, and was computationally intractable with higher-resolution images. A 10001000-pixel image with RGB color channels has 3 million weights per fully-connected neuron, which is too high to feasibly process efficiently at scale [2].

For example, in CIFAR-10, images are only of size 32323 (32 wide, 32 high, 3 color channels), so a single fully connected neuron in the first hidden layer of a regular neural network would have 32*32*3 = 3,072 weights. A 200200 image, however, would lead to neurons that have 200*200*3 = 120,000 weights [3].

Also, such network architecture does not take into account the spatial structure of data, treating input pixels which are far apart in the same way as pixels that are close together. This ignores locality of reference in data with a grid-topology (such as images), both computationally and semantically. Thus, full connectivity of neurons is wasteful for purposes such as image recognition that are dominated by spatially local input patterns [4].

Convolutional neural networks are variants of multilayer perceptrons, designed to emulate the behavior of a visual cortex. These models mitigate the challenges posed by the MLP architecture by exploiting the strong spatially local correlation present in natural images. As opposed to MLPs, CNNs have the following distinguishing features: [5]

3D volumes of neurons. The layers of a CNN have neurons arranged in 3 dimensions: width, height and depth. Each neuron inside a convolutional layer is connected to only a small region of the layer before it, called a receptive field. Distinct types of layers, both locally and completely connected, are stacked to form a CNN architecture [6].

Local connectivity: following the concept of receptive fields, CNNs exploit spatial locality by enforcing a local connectivity pattern between neurons of adjacent layers. The architecture thus ensures that the learned "filters" produce the strongest response to a spatially local input pattern. Stacking many such layers leads to nonlinear filters that become

increasingly global (i.e. responsive to a larger region of pixel space) so that the network first creates representations of small parts of the input, then from them assembles representations of larger areas [7].

Shared weights: In CNNs, each filter is replicated across the entire visual field. These replicated units share the same parameterization (weight vector and bias) and form a feature map. This means that all the neurons in a given convolutional layer respond to the same feature within their specific response field. Replicating units in this way allows for the resulting activation map to be equivariant under shifts of the locations of input features in the visual field, i.e. they grant translational equivariance-given that the layer has a stride of one [8].

Pooling: In a CNN's pooling layers, feature maps are divided into rectangular sub-regions, and the features in each rectangle are independently down-sampled to a single value, commonly by taking their average or maximum value. In addition to reducing the sizes of feature maps, the pooling operation grants a degree of local translational invariance to the features contained therein, allowing the CNN to be more robust to variations in their positions [9].

Together, these properties allow CNNs to achieve better generalization on vision problems. Weight sharing dramatically reduces the number of free parameters learned, thus lowering the memory requirements for running the network and allowing the training of larger, more powerful networks [10].

A CNN architecture is formed by a stack of distinct layers that transform the input volume into an output volume (e.g. holding the class scores) through a differentiable function. A few distinct types of layers are commonly used. These are further discussed below [11].

The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field, but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the filter entries and the input, producing a 2-dimensional activation map of that filter. As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input [12].

Stacking the activation maps for all filters along the depth dimension forms the full output volume of the convolution layer. Every entry in the output volume can thus also be interpreted as an output of a neuron that looks at a small region in the input. Each entry in an activation map use the same set of parameters that define the filter [13].

Self-supervised learning has been adapted for use in convolutional layers by using sparse patches with a high-mask ratio and a global response normalization layer [14].

When dealing with high-dimensional inputs such as images, it is impractical to connect neurons to all neurons in the previous volume because such a network architecture does not take the spatial structure of the data into account. Convolutional networks exploit spatially local correlation by enforcing a sparse local connectivity pattern between neurons of adjacent layers: each neuron is connected to only a small region of the input volume [15].

The extent of this connectivity is a hyperparameter called the receptive field of the neuron. The connections are local in space (along width and height), but always extend along the entire depth of the input volume. Such an architecture ensures that the learned filters produce the strongest response to a spatially local input pattern [16].

The depth of the output volume controls the number of neurons in a layer that connect to the same region of the input volume. These neurons learn to activate for different features in the input. For example, if the first convolutional layer takes the raw image as input, then different neurons along the depth dimension may activate in the presence of various oriented edges, or blobs of color [17].

Stride controls how depth columns around the width and height are allocated. If the stride is 1, then we move the filters one pixel at a time. This leads to heavily overlapping receptive fields between the columns, and to large output volumes. For any integer [18]

Sometimes, it is convenient to pad the input with zeros (or other values, such as the average of the region) on the border of the input volume. The size of this padding is a third hyperparameter. Padding provides control of the output volume's spatial size. In particular, sometimes it is desirable to exactly preserve the spatial size of the input volume, this is commonly referred to as "same" padding [19].

5. Conclusion

The integration of residual learning and attention mechanisms significantly enhances the automated segmentation of COVID-19 infections from Chest X-rays. This deep learning architecture provides a reliable, rapid assessment tool to aid radiologists in triaging and managing disease burden.

If this number is not an integer, then the strides are incorrect and the neurons cannot be tiled to fit across the input volume in a symmetric way. In general, setting zero padding to be [20]

ensures that the input volume and output volume will have the same size spatially. However, it is not always completely necessary to use all of the neurons of the previous layer. For example, a neural network designer may decide to use just a portion of padding [1].

A parameter sharing scheme is used in convolutional layers to control the number of free parameters. It relies on the assumption that if a patch feature is useful to compute at some spatial position, then it should also be useful to compute at other positions. Denoting a single 2-dimensional slice of depth as a depth slice, the neurons in each depth slice are constrained to use the same weights and bias [2].

Since all neurons in a single depth slice share the same parameters, the forward pass in each depth slice of the convolutional layer can be computed as a convolution of the neuron's weights with the input volume. Therefore, it is common to refer to the sets of weights as a filter (or a kernel), which is convolved with the input. The result of this convolution is an activation map, and the set of activation maps for each different filter are stacked together along the depth dimension to produce the output volume. Parameter sharing contributes to the translation invariance of the CNN architecture [3].

Sometimes, the parameter sharing assumption may not make sense. This is especially the case when the input images to a CNN have some specific centered structure; for which we expect completely different features to be learned on different spatial locations. One practical example is when the inputs are faces that have been centered in the image: we might expect different eye-specific or hair-specific features to be learned in different parts of the image. In that case it is common to relax the parameter sharing scheme, and instead simply call the layer a "locally connected layer". In this layer, the convolutional kernels' parameters are not shared. Instead, the network learns independent weights and biases for each spatial location. This allows each location to have its own feature-learning ability, making it better suited to handle images with distinct central structures or irregular features [4].

Another important concept of CNNs is pooling, which is used as a form of non-linear down-sampling. Pooling provides downsampling because it reduces the spatial dimensions (height and width) of the input feature maps while retaining the most important information. There are several non-linear functions to implement pooling, where max pooling and average pooling are the most common. Pooling aggregates information from small regions of the input creating partitions of the input feature map, typically using a fixed-size window (like 2x2) and applying a stride (often 2) to move the window across the input. Note that without using a stride greater than 1, pooling would not perform downsampling, as it would simply move the pooling window across the input one step at a time, without reducing the size of the feature map. In other words, the stride is what actually causes the downsampling by determining how much the pooling window moves over the input [5].

Intuitively, the exact location of a feature is less important than its rough location relative to other features. This is the idea behind the use of pooling in convolutional neural networks. The pooling layer serves to progressively reduce the spatial size of the representation, to reduce the number of parameters, memory footprint and amount of computation in the network, and hence to also control overfitting. This is known as down-sampling. It is common to periodically insert a pooling layer between successive convolutional layers (each one typically followed by an activation function, such as a ReLU layer) in a CNN architecture. While pooling layers contribute to local translation invariance, they do not provide global translation invariance in a CNN, unless a form of global pooling is used. The pooling layer commonly operates independently on every depth, or slice, of the input and resizes it spatially. A very common form of max pooling is a layer with filters of size 22, applied with a stride of 2, which subsamples every depth slice in the input by 2 along both width and height, discarding 75% of the activations: [6]

In addition to max pooling, pooling units can use other functions, such as average pooling or 2-norm pooling. Average pooling was often used historically but has recently fallen out of favor compared to max pooling, which generally performs better in practice [7].

Due to the effects of fast spatial reduction of the size of the representation, there is a recent trend towards using smaller filters or discarding pooling layers altogether [8].

5. Acknowledgement

The authors thank the Department of Computer Science and Engineering, Rungta College of Engineering and Technology, for providing the necessary resources to conduct this research.

5. References

[1] Ronneberger, O., et al. "U-net: Convolutional networks for biomedical image segmentation." *MICCAI*, 2015.

[2] He, K., et al. "Deep residual learning for image recognition." *CVPR*, 2016.

[3] Oktay, O., et al. "Attention u-net: Learning where to look for the pancreas." *arXiv preprint arXiv:1804.03999*, 2018.

[4] Wang, L., et al. "COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images." *Scientific reports*, 2020.

[5] Cohen, J. P., et al. "COVID-19 image data collection." *arXiv preprint arXiv:2003.11597*, 2020.

[6] Minaee, S., et al. "Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning." *Medical image analysis*, 2020.

[7] Selvaraju, R. R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *ICCV*, 2017.

[8] Zhou, Z., et al. "Unet++: A nested u-net architecture for medical image segmentation." *Deep Learning in Medical Image Analysis*, 2018.

[9] Chowdhury, M. E., et al. "Can AI help in screening viral and COVID-19 pneumonia?" *IEEE Access*, 2020.

[10] Apostolopoulos, I. D., & Mpesiana, T. A. "Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks." *Physical and Engineering Sciences in Medicine*, 2020.

[11] Ozturk, T., et al. "Automated detection of COVID-19 cases using deep neural networks with X-ray images." *Computers in biology and medicine*, 2020.

[12] Badrinarayanan, V., et al. "SegNet: A deep convolutional encoder-decoder architecture for image segmentation." *IEEE TPAMI*, 2017.

[13] Huang, G., et al. "Densely connected convolutional networks." *CVPR*, 2017.

[14] Chen, L. C., et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." *IEEE TPAMI*, 2017.

[15] Milletari, F., et al. "V-net: Fully convolutional neural networks for volumetric medical image segmentation." *3D Vision (3DV)*, 2016.

[16] Litjens, G., et al. "A survey on deep learning in medical image analysis." *Medical image analysis*, 2017.

[17] Rajpurkar, P., et al. "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning." *arXiv*, 2017.

[18] Alom, M. Z., et al. "Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation." *arXiv*, 2018.

[19] Goodfellow, I., et al. *Deep Learning*. MIT press, 2016.

[20] Ghassemi, N., et al. "A review of challenges and opportunities in machine learning for health." *AMIA Summits on Translational Science Proceedings*, 2020.