
# Transformers for COVID-19 detection for the multiclass classification problem utilizing X-ray images

Submitted as Research Report / Honours / Master Dissertation in
SIT723

SUBMISSION DATE
T2-2022

Rahul Kumar
STUDENT ID 221045868
COURSE - Master of Applied Artificial Intelligence (Professional) (S737)

# ACKNOWLEDGEMENT

It gives me immense pleasure presenting research thesis report on **"Transformers for COVID-19 detection for the multiclass classification problem (COVID-19, Pneumonia and Normal cases) utilizing X-ray images".** Throughout the process I have been blessed enough by the excellent guidance and support from my supervisor **Prof. Bahareh Nakisa**. Her expertise in the field of applied AI & computer vision with years of experience, insight and direction has helped me to explore the ongoing research in computer vision domain and find shortfalls to overcome in coming future. With every session I found myself motivated enough to continue the hard work in this field.

I would, hereby, acknowledge the support provided to me by my unit chair ***Prof Yong Xiang*** who taught me various strategies about how to approach the research thesis. Furthermore, ***Dr. Asef Nazari, Dr. Duc Thanh Nguyen, Dr. Chandan Karmakar & Dr. Wei Luo*** for teaching me the mathematics behind the utilization of applied AI & laying the groundwork for computer vision, machine learning & deep learning. In addition, the Deakin University online library tool assisted me in compiling my study resources and provided the knowledge I needed for my research thesis. The extensive collection of past research papers, journals, articles, and books proved to be an asset in educating me to get sufficient knowledge in the conducted research area.

RAHUL KUMAR

221045868

# LIST OF ABBREVIATIONS

FNN – Feed Forward Neural Network

RNN – Recurrent Neural Network

LSTM – Long Short-Term Memory (LSTM) networks

ANN – Artificial Neural Network

MLP – Multilayer Perceptron

CNN – Convolution Neural Network

ReLU – Rectified Linear Unit

FC – Fully Connected

SARS – Severe Acute Respiratory Syndrome

COVID-19 – Coronavirus Disease 2019

CXR – Chest X-ray

CT – Computed Tomography

RT-PCR – Reverse Transcription Polymerase Chain Reaction

AI – Artificial Intelligence

NLP – Natural Language Processing

SVM – Support Vector Machine

ViT – Vision Transformer

DeiT – Data Efficient Image Transformer

PVT – Pyramid Vision Transformer

TNT – Transformer In Transformer

AUC – Area Under the Curve

NPV – Negative Predictive Values

AUROC – Area Under the Receiver Operating Characteristic Curve

HL – Hamming Loss

LRAP – Label Ranking Average Precision

EANet – External Attention Transformer

CCT – Compact Convolutional Transformer

GCP – Google Cloud Platform

# ABSTRACT

The Coronavirus Disease 2019 (COVID-19) pandemic keeps on spreading across the globe and the expeditious spread of COVID-19 since its outbreak has impelled many nations healthcare frameworks & economy to the edge of breakdown. Therefore, to suppress the spread of the disease and minimize the ongoing expenditure on the healthcare system, accurate identification & isolation of COVID-19 positive individuals for treatment is vital. Albeit accurate and adequate, the fundamental Coronavirus screening test, RT-PCR, has a lengthy turnaround time. Various research on COVID-19 detection and treatment based on emerging technology is still ongoing. In the last few years, researchers have exhibited utilization of Deep Learning (DL) methods like Convolutional Neural Network (CNN) on chest X-ray (CXR) for COVID-19 detection which speeds up the COVID-19 diagnostic process, but due to CNN's inherent image-specific inductive bias, CNN methods fail to capture the global context. Therefore, a need to develop advanced Deep Learning model using Transformers for COVID-19 detection arises.

# LIST OF FIGURES

# LIST OF TABLES

# TABLE OF CONTENTS

# 1. INTRODUCTION

## 1.1 MOTIVATION

A fatal viral pandemic known as the Spanish flu, or influenza pandemic, caused by the H1N1 virus, struck one-third of the world's population almost 102 years ago and killed at least 50 million people worldwide [44, 45]. After the Spanish flu, there has not been a significant global pandemic until 2019. In December 2019, the Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2) triggered a global pandemic & gave rise to a viral respiratory disease known as Coronavirus Disease 2019 (COVID-19) which was first debatably appeared in the city of Wuhan (Province of Hubei, China). As of 24 September 2022, a total of 615 million COVID-19 cases & 6.54 million deaths worldwide have been recorded [46]. The global spread of the Coronavirus Disease 2019 (COVID-19) pandemic has pushed several countries' healthcare systems and economies to the brink of collapse. People with COVID-19 have reported experiencing a wide range of symptoms, from mild symptoms to serious sickness. Fever/chills, cough, exhaustion, shortness of breath, sore throat, loss of taste or smell, and muscle aches are some of the symptoms of COVID-19 that are frequently noted [47]. The onset of symptoms might range from 2 to 14 days following viral exposure. After experiencing symptoms for around 8 days, people can still continue to spread the virus. COVID-19 can be lethal in patients debilitated by age or another chronic illness and is communicated by close contact with infected individuals [47]. Therefore, it's crucial to accurately identify and isolate COVID-19 positive individuals for treatment in order to suppress the disease's spread and reduce the ongoing costs to the healthcare system. Due to different variant of the virus in various countries, COVID-19 is proved to be more contagious & difficult to analyze with standard methods.

### 1.1.1 TRADITIONAL METHODS

The healthcare sector has greatly progressed throughout recent years, making it possible for medical specialists to aid and serve more efficiently in the diagnosis and treatment of illness. Real-time Reverse Transcription Polymerase Chain Reaction (RT-PCR) has been the gold standard for diagnosing COVID-19 due to its accuracy & authenticity as it demonstrates high specificity & sensitivity [50]. A RT-PCR is utilized to detect SARS-CoV-2 by means

of collected respiratory samples such as nose or throat swabs [49]. In order to extract only the RNA present in the sample, it is treated with a number of chemical solutions that eliminate other components including proteins and fats. Reverse transcription of RNA to DNA, or RT, is an additional step in RT-PCR that enables amplification. RT-PCR can provide a valid diagnosis in just three hours, although laboratories typically require between six to eight hours [48]. However, it is expensive, requires a laboratory, trained personnel, and RT-PCR kit for COVID-19 detection & analysis, which is laborious, time-consuming & exhibits poor sensitivity as greater false negative values in RT-PCR tests can have serious repercussions. Furthermore, in order to validate whether an individual is effected with COVID-19 with more precision, he had to undergo Computed Tomography (CT) (or Chest CT scan) or Chest X-rays (CXR) using medical image analysis. Afterwards, a radiologist specialist reviews the scans to determine whether the disease is present in the chest region [53]. With the innovation of automation in the medical sector, medical professionals no longer required to manually gather data from each individual patient or manually examine the disease within a constrained timeframe, which could lead to human error during testing. Therefore, it is believed that improved medical image analysis may support medical professionals if it were effectively addressed. In this view, medical data are of utmost importance. It can be represented with Unidimensional or bidimensional signals [55, 56], an image stack [57], or a significant quantity of data [58]. Furthermore, it is possible to make use of multimodal data sources [59]. On the other hand, medical imaging methods like X-ray and CT-based screening are more broadly accessible, faster, and generally safe.

### 1.1.2 CURRENT METHODS

CT scans have also been helpful in the evaluation of pulmonary conditions & exhibited sound promise in assisting early identification of COVID-19 with improved image details [51] [52]. However, it necessitates moving the patient to the CT department, performing a platform-sterilization procedure prior to carrying out the test, and requiring the assistance of experts both before and after. In general, CT scanners require expensive equipment & maintenance. In contrast to CT imaging, X-ray imaging has been used frequently for COVID-19 screening because it is more widely available in most healthcare systems (even in remote areas [54]), is less expensive, and the portability of the CXR system reduces the risk of COVID-19 transmission by performing the exams inside the isolation room, which

is not possible with the fixed CT scanners. Significantly, CXR empowers rapid triaging of associated COVID-19 cases in the majority of impacted nations, including USA, Spain, and Italy, where they have exhausted their RT-PCR testing capacity and supplies [60]. The accuracy of a COVID-19 infection diagnosis via chest imaging is heavily dependent on radiological competence due to the complex morphological patterns of lung involvement, which can change in severity and appearance over time.

## 1.2 AIM & OBJECTIVES

The following list summarizes the primary aim and objective of the research thesis with detailed subsections that provides more in-depth scope of our research project.

### 1.2.1 Aim: To develop an advanced Deep Learning model using Transformers for COVID-19 detection for the multiclass classification problem (COVID-19, Pneumonia and Normal cases) utilizing X-ray images.

In order to identify COVID-19 patients with the least amount of time spent and maximum amount of effectiveness, deep learning and applied artificial intelligence are therefore required. By examining thousands of medical images and identifying irregularities that support or refute a diagnosis, Deep Learning-based models can reduce the workload of medical specialists. Applied AI aids in the earlier detection of the underlying cases and the organization of a more viable management of medical personnel and equipment to deal with the pandemic's peak. Additionally, it might make the disease's diagnosis and treatment simpler.

According to the findings of the conducted literature review, adopting transformers in computer vision is a major leap towards research in this field. Thereby, we can reduce the amount of time, money, and labour required in the medical industry for COVID-19 detection; additionally, we can improve the performance and efficiency of the models by combining transformers with convolutions.

#### 1.2.1.1 To investigate the benefits of the architectural design of the conducted CvT

Based on the conducted CvT model, we will investigate the design benefits in terms of model optimization, performance & computational cost.

#### 1.2.1.2 To investigate the design modifications of the conducted CvT

Based on the conducted CvT model, we will investigate the architectural design modification & how it’s impacting the performance of the model.

## 1.3 STRUCTURE

The conducted research thesis is structured as follows: Section 2 conducts a literature review of the related works on deep learning models for COVID-19 detection using X-ray images, to discover fundamental concepts and address gaps in study. Section 3 demonstrates the background & scope of the proposed method, along with its advantages & disadvantages. Section 4 demonstrates the research design & methodology of the proposed method for achieving the aims and objectives of the research thesis. Section 5 explains the approach and technical specifics of how a software artefact was developed to evaluate this method. Section 6 details the empirical evaluation of the conducted method. Section 7 presents the evaluation & threats to validity of the proposed method. Finally, Section 8 concludes the report with the discussion of results and recommendations for future works.

# 2. LITERATURE REVIEW

The global spread of the Coronavirus Disease 2019 (COVID-19) pandemic, which has been active since its emergence, has pushed several countries' healthcare systems and economies to the brink of collapse. Therefore, it's crucial to accurately identify and isolate COVID-19 positive individuals for treatment in order to stop the disease's spread and reduce the ongoing costs to the healthcare system.

**This research review discusses novel methods to curb the problem of COVID-19 cases using Transformers. We first discuss about the current methods employed in COVID-19 detection. Afterwards, we will discuss about the use of CNNs in COVID-19 detection and later we will discuss various transformer models, tools & their optimization techniques applied in detecting COVID-19 cases. Finally, we will provide conclusion.**

Reverse Transcription Polymerase Chain Reaction (RT-PCR) has been the golden standard for diagnosing COVID-19 due to its accuracy & authenticity but is expensive and requires trained professionals, laboratory, and RT-PCR kit for COVID-19 detection & analysis which is time-consuming & arduous. Medical images such as chest X-rays (CXR) are crucial to confirming a COVID-19 diagnosis, as they provide accurate laboratory test as visual evidence to physicians and radiologists while being readily accessible in most healthcare systems. In the last few years, researchers have exhibited utilization of Deep Learning (DL) methods like Convolutional Neural Network (CNN) on chest X-ray (CXR) for COVID-19 detection [61, 62. 63, 64] which speeds up the COVID-19 diagnostic process, but CNN methods fail to capture the global context due to their inherent image-specific inductive bias [65]. Additionally, in order for CNNs to capture long-range dependencies, a large receptive field is required, necessitating the design of enormously large kernels or deep networks, which results in a complex model that is difficult to train.

Therefore, an advanced Deep Learning model, Vision transformers using transformer architecture for images was proposed by Dosovitskiy et al. (2021) [66] based on original

self-attention-based architecture, in particular Transformers (Vaswani et al., 2017). Vision Transformer used self-attention mechanism for modelling the pixel dependency among pixels [67] as it enhances local related features and performs feature recalibration, which inadvertently lessens the importance of features at the spatial and channel levels [68]. The rise of Vision Transformers has likewise given a significant establishment to the development of vision models. ViT models presents a new state-of-the-art on Image Recognition with a model that, even fully depending on self-attention, is able to present performances on par with current state-of-the-art models.

The following review of research literature is based on detecting & classifying COVID-19 patients using Chest Radiography (X-ray) images, using different transformers conducted over the last two years, from 2020-2022 and provide a summary and statistical data analysis on the basis of performance metrics & results with previously implemented state-of-the-art methods using Transformers. This review of research literature excludes the use of transformers for Natural language processing (NLP) tasks as we are mainly dealing with applications utilizing transformers in computer vision domain for COVID-19 detection specifically.

Convolution Neural Network (CNN) are quite popular among the research community of AI in medicine as they produced best classification accuracy as compared to classification techniques like Artificial Neural Network (ANN), Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) [69] for COVID-19 detection using X-ray images. A work by Maram Mahmoud A. Monshi (2021) [70] proposed CovidXrayNet, an optimized CNN model based on EfficientNet-B0 [72] utilizing data augmentation & CNN hyperparameter tuning for detecting COVID-19 from CXRs in terms of validation accuracy. The author achieved state-of-the-art accuracy of 95.82% on the COVIDx dataset in the three-class classification task (COVID-19, normal or pneumonia) with only 30 epochs of training. They further classified that CovidXrayNet is still at a research stage and currently not suitable for direct clinical diagnosis.

Notwithstanding, existing Deep Convolutional Neural Network (CNN) methods neglect to capture the global context because of their inherent image-specific inductive bias. As of late, different investigations utilized the Vision Transformer (ViT) models for COVID-19 detection rather than CNN architectures. The challenge in medical images for ViT approaches comes in form of long-range dependencies and multi-modality. In Arnab Kumar Mondal et al. (2021) [74], they proposed a vision transformer based deep neural classifier, xViTCOS (Explainable Vision Transformer Based COVID-19 Screening Using Radiography). They employed a multi-stage transfer learning approach to address the problem of need for large-scale data & Gradient Attention Rollout algorithm [75] (For Explanability). They demonstrated the viability of the proposed structure in distinguishing COVID-19 positive cases from non-COVID-19 Pneumonia and Normal control utilizing both chest CT scan and X-ray methodology, through several trials on benchmark datasets where they achieved an overall accuracy of 98.1% & recall (Sensitivity) of 96%. But xViTCOS-CT fails to predict the ground truth non-COVID-19 Pneumonia with confidence.

The image classification utilizing transformers can be optimized via transfer learning, data augmentation & hyperparameter tuning of the models. In Mohamed Chetoui & Moulay A. Akhloufi (2022) [79], several ViT models (ViT-B16, ViTB32, and ViT-L32) were fine-tuned for the multiclass classification problem (COVID-19, Pneumonia and Normal cases). Data augmentation was applied during training, hyperparameter tuning of ViT models were done & visualized the signs detected by ViT by using the attention map of the best model (ViT-B32). Furthermore, demonstrated that the obtained results outperformed comparable state-of-the-art models for detecting COVID-19 on CXR images using CNN architectures. They report an accuracy of 96% & Recall (Sensitivity) of 96%. While they concluded that the recommended model would have an even better performance if compared to a manual interpretation by radiologists. Koushik Sivarama Krishnan & Karthik Sivarama Krishnan (2021) [82] proposed a fine-tuned vision transformer (ViT-B/32) for detecting COVID-19 on chest X-rays. Noise is a key factor in radiography that affects the model's performance. For distinguishing COVID-19 cases from Viral Pneumonia, Lung Opacity, and Healthy chest X-rays, they acquired an accuracy score of 97.61%, Precision score of 95.34%, Recall (Sensitivity) score of 93.84% & F1-Score of 94.58% using ViT-B/32 transformer [84].

Debaditya Shome et al. (2021) [87], used a constructed three-class data set of 30 K chest X-ray pictures for COVID-19 detection using Vision Transformer (ViT L-16) for healthcare. They designed a COVID-19 detection pipeline utilizing the Vision Transformer model and fine-tuned it on their dataset with a custom MLP block, resizing & interpolation of images, data augmentation techniques (random rotation, width shift, height shifts, and flipping). For better model interpretability and simplicity of diagnosis, they created Grad-CAM-based visualizations of COVID-19 movement in the lungs, which helps the diagnosis process for medical services. They report high performance with accuracy and AUC score as high as 98% and 99%, respectively for classifying COVID-19 from normal chest X-rays in the binary classification task. Furthermore, distinguishes COVID-19, normal, and pneumonia patient's X-rays with an accuracy of 92% and AUC score of 98% in the multi-class classification task. The only disadvantage with their model was that the model results in overfitting beyond 2 dense layers. The conducted study might be very important in regions where quick testing is inaccessible, and it might likewise be utilized as a second screening method after the standard RT-PCR test to check that any true negative or false positive cases don't happen.

The self-attention transformer-based approach is of fundamental importance for the methods intent to analyze the COVID-19 in CT scan images. In Fozia Mehboob et al. (2022) [93] proposed a novel architecture of vision transformers using self-attention Transformer based diagnosis approach for the diagnosis of COVID-19 using 3D CT Slices. The authors employed image segmentation using transformer encoder/ decoder architecture, layer normalization & data augmentation (image flipping (horizontal), resizing (image size) and rotation) for optimizing image classification. They compared the performances of self-attention transformer-based approach with CNN and Ensemble classifiers for diagnosis of COVID-19 using Brazilian dataset (SARS-COV2 CT scan dataset [92]) & COVID-19

(HUST-19 [94]) CT scan dataset. Upon evaluation, the authors achieved an accuracy of 98% on Brazilian dataset & 99.7% on multi-class Hust19 CT scan dataset. The proposed transformer vision approach can predict the quantification of COVID-19 based on the pixel values in the long-range relation-based maps which can provide valuable assistance to specialists in decision making with respect to the assessment of the severity of the COVID-19.

Swin Transformer (Ze Liu et al., 2021) [96] outstandingly refreshed past records on object detection and semantic segmentation benchmarks, providing the community prominent certainty that the Transformer structure will turn into the new standard for visual modeling. In Juntao Jiang & Shuyi Lin (2021) [95], the authors proposed a method which combined Swin Transformer (Swin-B) [96] and Transformer in Transformer (small) [97] for classifying COVID-19, Pneumonia and Normal (healthy) cases using chest X-ray images [98] and did model ensemble by using weighted average method. The authors utilized data augmentation (horizontal flipping, and the rotation or translation), applied label smoothing in loss & performed hyperparameter tuning of the model for optimizing the model. The authors report achieving an accuracy of 94.75% for detecting COVID-19 cases.

ViT does not generalize well when trained on insufficient amounts of data while Data-Efficient Image Transformer (DeiT) almost follows the same architecture as ViT but follows a teacher-student strategy with distillation token & provides better performance as compared to competitive convnet models and most state-of-the-art ViT models. In Mohamad Mahmoud Al Rahhal et al. (2022) [91], presented a novel architecture based on a Data-Efficient Image Transformer (DeiT) architecture which is an improved version of Vision Transformer (ViT) for COVID-19 detection using CT (SARS-CoV-2 CT [92]) & X-ray images (COVIDx [73]). The authors employed a Siamese encoder that implements a distillation technique to classify original and augmented images. Heat maps were utilized which showed the progression of focus areas over network layers, similar to X-ray or CT images. Upon evaluation, the authors achieved an accuracy of 94.62% on CXR dataset & 99.13% accuracy on CT dataset which showed the model's robustness under limited training data.

Pyramid Vision Transformer (PVT) proposed by Wenhai Wang et al. (2021) [100] overcomes the challenges of porting Transformer to different dense prediction tasks. PVT can be trained on dense partitions of the image to accomplish high output resolution and can reduce computations of large feature maps using a progressive shrinking pyramid structure. In Xiaoben Jiang et al. (2022) [99] proposed a new variant of pyramid vision Transformer (MXT) for multi-label chest X-ray image classification which can capture both short and long-range visual information through self-attention. The authors employed multi-layer overlap patch (MLOP) embedding, class token Transformer block and multi-label attention (MLA) for more effective processing of multi-label classification. The authors evaluated MXT on a large-scale CXR dataset (Chest X-ray14 [101]) with 14 disease pathologies & Catheter dataset which resulted the highest mean AUC score of 83.0% on the Chest X-ray14 dataset and 94.6% on the Catheter dataset [102]. While the heatmaps generated from the

Model-D were discrete and blurry (i.e., CXR image with Atelectasis, Pneumothorax), and location of the lesion regions is in error which presented the limitations of the presented model. Overall, MXT can help radiologists in diagnoses of lung diseases and check the position of catheters, which can lessen the work strain of medical staff.

**Table 1 Literature Review**

| Reference (Authors, papers etc.) | Type of Transformers | Database Used | Methodology Used | Performance/Accuracy of Model |
|---|---|---|---|---|
| 1. **xViTCOS: Explainable Vision Transformer Based COVID-19 Screening Using Radiography** [74] | Vision Transformers (ViT-B/16, ViT-B/32, ViT-L/16 and ViT-L/32) | CT Scan Dataset (COVIDx CT-2A [76] dataset), Chest X-ray (CXR) Dataset (COVIDx CXR-2) [77], CheXpert [78] | Multi-stage transfer learning approach, Gradient Attention Rollout algorithm (For Explanability) | xViTCOS-CT: Accuracy- 98.1%, Recall (Sensitivity)- 96%, Precision- 96%, F1-Score- 96.1%, Specificity- 98.8% & NPV- 98.8% for COVID-19 cases<br><br>xViTCOS-CXR: Accuracy- 96%, Recall (Sensitivity)- 100%, Precision- 99%, F1-Score- 99.5%, Specificity- 99.7% & NPV- 100% for COVID-19 cases |
| 2. **CovidXrayNet: Optimizing data augmentation and CNN hyperparameters for improved COVID-19 detection from CXR** [70] | VGG-16, VGG-19 [71], ResNet-18, ResNet-34, ResNet-50, and EfficientNet-B0 [72]. | COVIDx [73], COVIDcxr | Data augmentation on CXR ( resizing, flipping, rotating, zooming, warping, lighting, and normalizing) | CovidXrayNet: Accuracy- 95.82%, Recall (Sensitivity)- 95.43%, Precision- 96.93%, F1-Score- 96.16%, MCC- 92.24% for COVID-19 cases |
| 3. **Explainable Vision Transformers and Radiomics for COVID-19 Detection in Chest X-rays** [79] | Vision Transformers (ViT-B16, ViT B32,and ViT-L32) | SIIM-FISABIO-RSNA COVID-19 [80], RSNA [81] | Data augmentation was applied during training, Hyperparameter tuning of ViT models & visualized the | ViT-B32: Accuracy- 96%, Recall (Sensitivity)- 96%, Specificity- 96% & AUC- 99% for COVID-19 cases |

| | | | | |
|---|---|---|---|---|
| | | | signs detected using attention map of the best model (ViT-B32). | |
| 4. **Vision Transformer based COVID-19 Detection using Chest X-rays** [82] | Vision Transformer (ViT-B/32 [84]) | COVID19 X-ray database dataset [85, 86], COVID19 Pneumonia Normal Chest X-ray PA Dataset | Transfer Learning, Image augmentation, Image enhancement methods [83] & Hyperparameter tuning of Vision Transformers | ViT-B/32: Accuracy- 97.61%, Precision- 95.34%, Recall (Sensitivity)- 93.84% & F1-Score- 94.58% for COVID-19 cases |
| 5. **COVID-Transformer: Interpretable COVID-19 Detection Using Vision Transformer for Healthcare** [87] | Vision Transformer (ViT L-16) | Constructed a three-class data set of 30 K chest X-ray pictures (Qi et al. [90], El-Shafai et al. [88] & Sait et al. [89]) | Used a custom MLP block, Resizing & Interpolation of images, Data Augmentation techniques (random rotation, width shift, height shifts, and flipping) | COVID-Transformer: Accuracy- 92%, Precision- 93%, Recall (Sensitivity)- 89%, F1-Score- 91% & AUC- 98% for COVID-19 cases |
| 6. **COVID-19 Detection in CT/X-ray Imagery Using Vision Transformers** [91] | Vision Transformers (DeiT Architecture [108]) | COVIDx [73], SARS-CoV-2 CT [92] | Model is based on a Data-Efficient Image Transformer (DeiT) Architecture, Token & Distiller classifiers are used, Siamese encoder is used | For CXR: Accuracy- 94.62%, Precision- 96.77%, Recall (Sensitivity)- 96.77%, F1-Score- 96.77% & Specificity- 99.65% for COVID-19 cases<br><br>For CT: Accuracy- 99.13%, Precision- 99.46%, Recall (Sensitivity)- 98.82%, F1-Score- 99.13% & Specificity- 99.47% for COVID-19 cases |

| | | | | |
|---|---|---|---|---|
| 7. **COVID-19 Detection in Chest X-ray Images Using Swin-Transformer and Transformer in Transformer** [95] | Swin Transformer [96] (Swin-B) + Transformer in Transformer [97] (small) | Challenge of Chest XR COVID-19 detection (Dataset) [98] | Data Augmentation (horizontal flipping, and the rotation or translation), applied label smoothing in loss & Hyperparameter tuning of model | Results for CXR: Accuracy- 94.75%, Recall (Sensitivity)- 94.75% & Specificity- 95.09% for COVID-19 cases |
| 8**. MXT: A New Variant of Pyramid Vision Transformer for Multi-label Chest X-ray Image Classification** [99] | Pyramid vision Transformer (PVT) [100] | Chest X-ray14 & Catheter dataset [101, 102] | Multi-layer overlap patch (MLOP) Embedding, class token Transformer block and multi-label attention (MLA) are utilized, Downsampling spatial reduction attention (DSRA) & Dynamic position feed forward with zero paddings is utilized | For Chest X-ray14: AUC- 83%, HL- 6.9% & LRAP- 81%<br><br>For Catheter: AUC- 94.6%, OE- 14.5%, HL- 5.9% & LRAP- 90.1% |
| 9. **Towards robust diagnosis of COVID-19 using vision self-attention transformer** [93] | Vision Transformer | Brazilian dataset (SARS-COV 2 CT scan dataset [92]) & COVID-19 (HUST-19 [94]) | Segmentation of image is performed using transformer encoder/ decoder architecture, Layer normalization is implemented & Data augmentation (image flipping (horizontal), resizing (image | Brazilian Data-set: Accuracy- 98%, AUROC- 99.6%<br><br>Hust19 Data-set: Accuracy- 99.7%, AUROC- 99.7% |

| | | | | |
|---|---|---|---|---|
| | | | size) and rotation) | |
| 10. **Vision transformer and explainable transfer learning models for auto detection of kidney cyst, stone and tumor from CT-radiography** [104] | Vision Transformer (EANet [106], CCT [107], and Swin transformers [96]) | CT KIDNEY DATASET [105] | Scaled, resized, randomized images for various deep learning & transformer models. Image augmentation, normalization & Transfer learning | EANet: Accuracy- 77.02%<br><br>Swin Transformers: Accuracy- 99.30%<br><br>CCT: Accuracy- 96.54% |

After reviewing the relevant literature, Transformer's extraordinary success in NLP has been explored in the computer vision domain & presents a crucial step in the research direction. Model architectures in the related work of adopting transformer in computer vision very well may be utilized in a pure Transformer manner or in a hybrid manner by combining with CNNs. Utilizing convolutions with transformers can enhance the performance & efficiency of the models. In spite of the viability of deep learning-based frameworks in COVID-19 identification, models that are trained on small datasets will lead to improper generalization because of which the model could perform inadequately in real world scenario and having a small test set could bring about missing out on false positives or negatives. Transformers are still profoundly data-dependent and is undoubtedly a significant step towards research in computer vision domain through which we can save a lot of time, cost & labour in medical industry for COVID-19 detection.

# 3. BACKGROUND & SCOPE

In this section, we will discuss the theoretical & practical background of the emergence of transformers along with its advantages over convolution neural networks (CNN). Furthermore, we will also discuss the types of transformer architectures used in computer vision domain for image segmentation and detection using various methodologies.

## 3.1 Recurrent Neural Networks (RNNs)

Recurrent neural networks (RNNs) are feed-forward neural networks that are rolled out over time in the form of a graph with directed cycles. Information moves from layer to layer with existing loops in the network, as opposed to feedforward neural networks (FNNs), causing each state to be influenced by its previous ones. Due to the inclusion of loops, RNNs have

the memory capacity to store past computations and display dynamic temporal behavior [1] [2] [3].

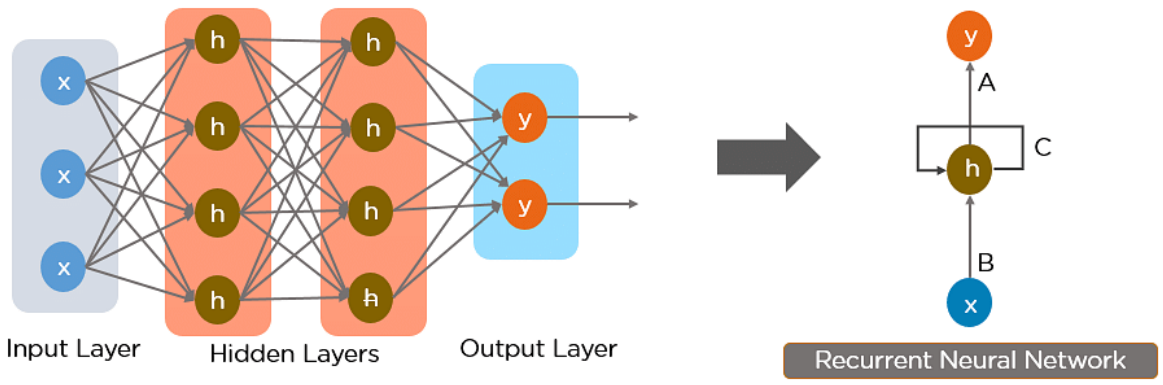



**Figure 3.1 Conversion of a FNN into a simple RNN [5]**

In this case, the input layer is represented as "x," hidden layer as "h," and the output layer as "y." The network parameters A, B, and C are used to enhance the model's output. At any given interval t, the current input is an integration of input at x(t) and x(t-1). The output at any given time is called back to the network to enhance the output [5].

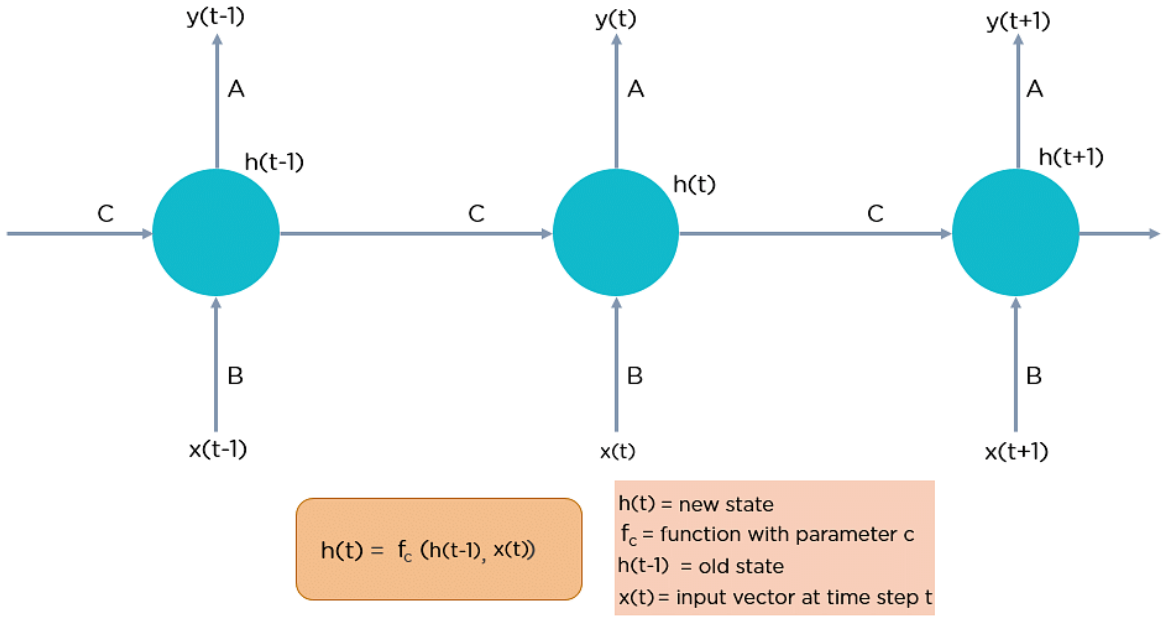



**Figure 3.2 Fully connected Recurrent Neural Network (RNN) [5]**

An RNN has two major disadvantages:

1. **Vanishing gradient and exploding gradient**

The issues are categorized by the gradient's size which is the slope of the loss function along the error curve. When the gradient is too small, it keeps getting smaller (close to 0), making it harder for the network to retain or learn words further along in the sequence (long-term dependencies) [4], and thus forces it to rely its predictions solely on the most recent data due to insufficient weight adjustments [1]. Similarly, when the gradient is too large, the model weights increase immensely, causing very massive updates to the neural network model weights during training [1] [4].

2. **Slow to Train**

   RNNs have Longer training time which slows down the training process and results in poor performance with bad accuracy overall [1] [4].

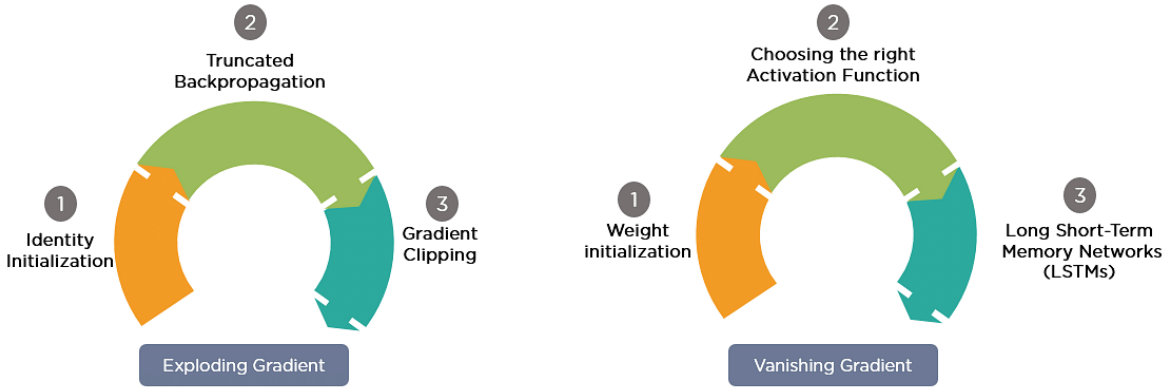



**Figure 3.3 Solution to gradient problem [5]**

## 3.2 Long Short-Term Memory (LSTM)

A special sort of RNN designed specifically for overcoming vanishing gradient problems is the long short-term memory (LSTM). They have the capacity to learn long-term dependencies. Selectively remembering or forgetting relevant and less significant information is possible using LSTMs [1] [5] [6].

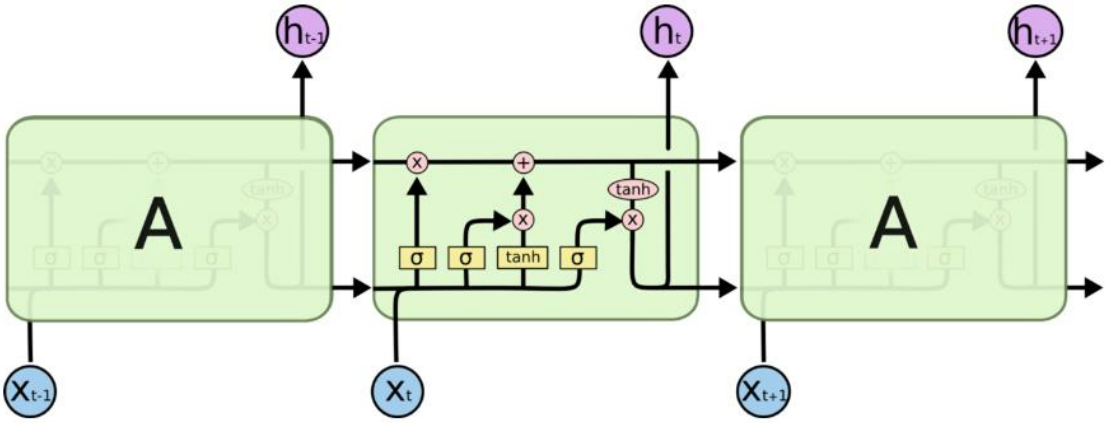



**Figure 3.4 Standard LSTM**

While learning is improved by LSTM recurrent units, they are far more complex than RNNs and demand more computational resources.

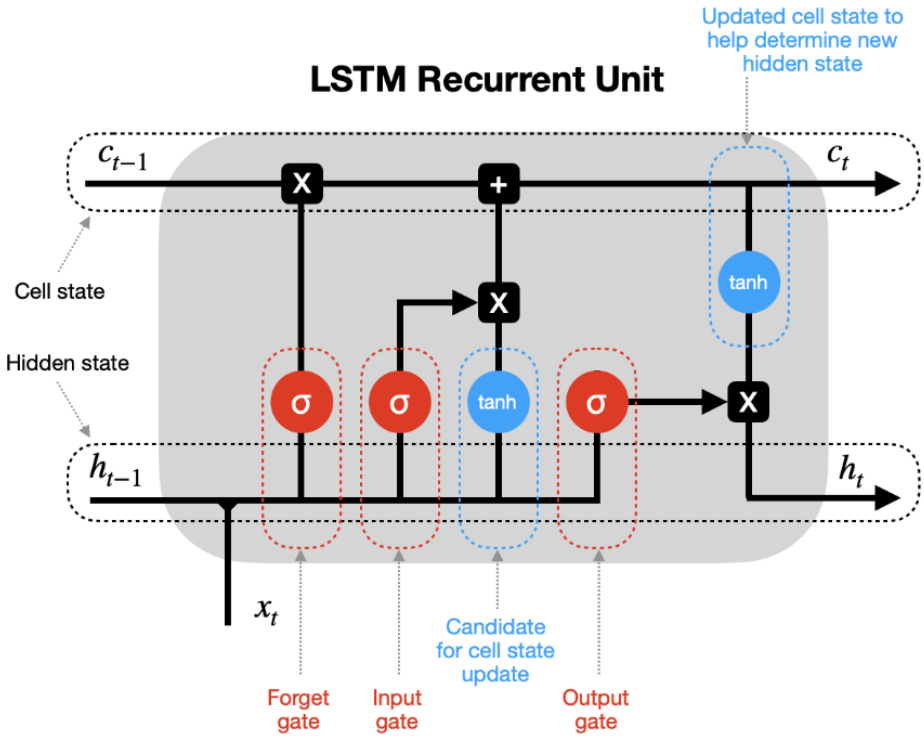



**Figure 3.5 Working of LSTM [8]**

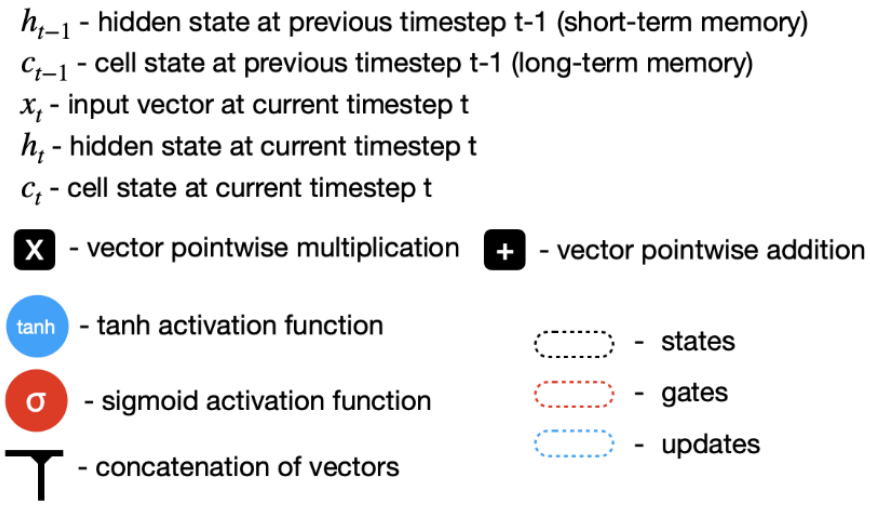



**Figure 3.6 Representation of symbols in LSTM process [8]**

The LSTM operates in a three-step process. The LSTM operation's working process is outlined below:

1. ***Determine how much historical data it should retain***

The first stage in LSTM is to determine which type of information should be excluded from the cell in that particular time-step. This is determined by the 'sigmoid function'. It computes the function while taking into account the current input (x)t and the prior state (ht-1) (presented mathematically in equation 3.1) [5][8].

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right) \tag{3.1}$$

where, $ft$ = forget gate decides what type of insignificant information to exclude from previous time-step.

2. ***Determines the amount of information contributes to the current state***

Second layer consists of two parts: sigmoid function and tanh function. Sigmoid function determines which values to pass through (0 or 1) while tanh function provides weightage to the values that are allowed by determining their level of significance (-1 to 1) (presented mathematically in equation 3.2) [5] [8].

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right) \tag{3.2}$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

where, $it$ = input gate decides which information to pass through according to its importance in the current time-step. The input gate examines the crucial information with the current input at x(t).

***3. Determines Which part of the Current Cell State Is Included in the Output***

The third step deals about the output. We first implement a sigmoid layer to determine which components of the cell state makes it to the output. Afterwards, the sigmoid gate output is multiplied by the cell state after being passed through tanh to push the values to be between -1 and 1 (presented mathematically in equation 3.3).

$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right) \tag{3.3}$$
$$h_t = o_t * \tanh\left(C_t\right)$$

where, $Ot$ = Output gate permits the information that is passed to affect the output in the current time-step.

**Drawbacks of LSTM**

- LSTM is even slower to train as compared to RNN.
- Vanishing gradient problem still persists.
- Sequential computation inhibits parallelization.
- Distance between positions is linear

Due to the recursive nature of LSTMs (in order to get the current state, we must first find out the prior state), they cannot be trained in parallel. LSTMs are constrained when faced with more difficult modern issues, such as multilingual machine translation or text generation, it becomes impossible to distinguished from human-written text [9].

## 3.3 Convolutional Neural Networks (CNN)

The most popular model in computer vision applications is convolutional neural networks (also known as convnets or CNN). When the input dimension is relatively small, they perform admirably as task layers. But images are the inputs for computer vision applications. In such applications, convolutional networks perform significantly better than densely connected networks achieving better accuracy [14].

Densely connected layers are replaced with convolution layers in convnets, which are designed to capture spatial patterns in an image. Convolutional layers can capture spatial information with increasing levels of abstraction when they are stacked one on top of the other [14].

A convnet has three parts, just as other networks: ***preprocessing, feature, and task layers***.

There are three different kinds of layers in CNNs: ***Convolutional, pooling, and fully-connected layers***. A CNN architecture is created once these layers are stacked. Figure 3.8 shows a comprehensible CNN architecture for MNIST classification.

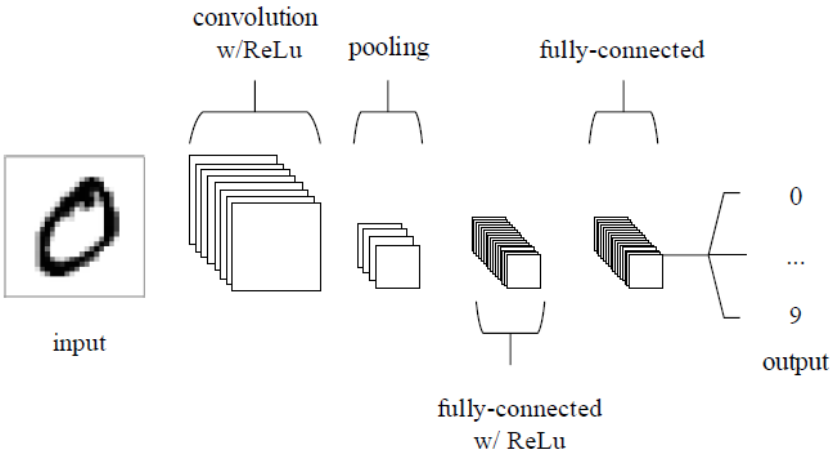



**Figure 3.7 A simple CNN architecture containing five layers [11]**

The typical CNN architectures stack a few convolutional layers (followed by a ReLU layer), then a pooling layer, then another few more convolutional layers (+ReLU), then another pooling layer, and so on. The image gets smaller and smaller as it moves through the network, but it also usually gets deeper and deeper that is with more feature maps, because of the convolutional layers (figure 3.9). A typical feedforward neural network made up of a few fully connected layers (+ReLUs) is added to the top of the stack, and the prediction is outputted by the last layer (e.g., a softmax layer that outputs estimated class probabilities) [10] [15].

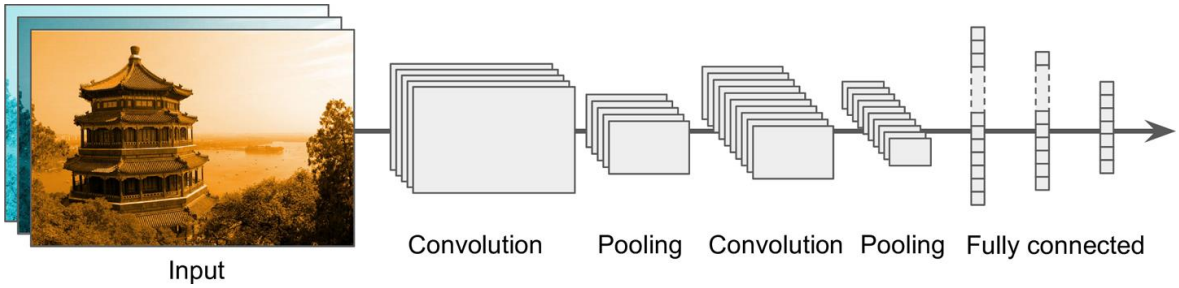



**Figure 3.8 Typical CNN architecture [15]**

### Convolution Layer Operation

Figure 3.10 illustrates how Convolution layers learn local patterns while Dense layers in their input feature space learn global patterns (e.g., in a MNIST digit, patterns involving all pixels) [14].

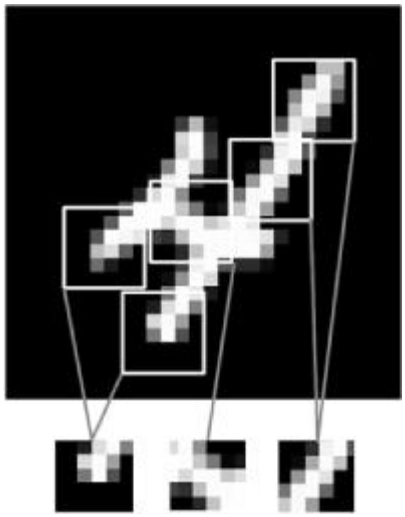


**Figure 3.9 Edges, textures, and other local patterns can be used to deconstruct an image [14]**

Convnets have the following two characteristics while working with image data:

1. They are ***able to learn spatial hierarchies of patterns***, which allows them to put together more complicated patterns using simpler ones as the visual world is basically spatially hierarchical. Edges and other small local patterns will be learned by a first convolution layer. A second convolution layer will afterwards learn larger patterns assembled by the characteristics of the first layer, and so forth [2] [12] [14].

2. Since the ***patterns they learn are translation invariant***, the network decision is unaffected by the precise positioning of an object within an image. The visual world is basically translation-invariant. A learnt pattern may emerge in several parts of a

picture, and the convnet may detect it anywhere. Convolution achieves filter parameter sharing, data efficiency on perceptual problems, and avoids custom feature engineering by sliding a filter over the input feature map, halting at every feasible region to capture local translation-invariant features [11] [14] [15].

Convolutions work using feature maps, which are operated over 3D tensors having two spatial axes (***height and width***) and a ***depth axis*** (also called the ***channels axis***). The convolution process extracts patches from its input feature map and transforms them all using the same transformation, thereby creating an ***output feature map***. This 'output feature map' still retains a width and a height, making it a 3D tensor. Since the output depth is a layer parameter and the different channels in that depth axis no longer represent distinct colours as in RGB input but rather ***filters***, its depth is arbitrary. At a high level, a single filter could encode the concept "presence of an ear in the input," for example. Filters encode distinctive characteristics of the input data [14] [15].

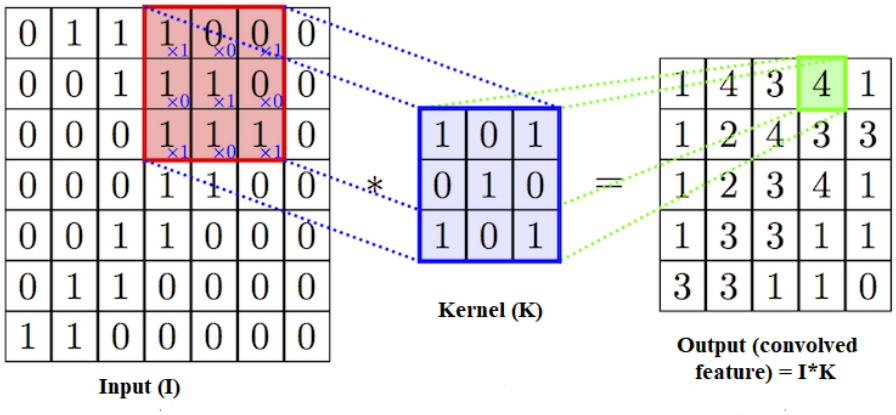



**Figure 3.10 Convolution Layer [13]**

There are two crucial factors that constitute convolutions:

1. ***Size of the patches that are extracted from the inputs***, which are generally 3x3 or 5x5 in size (also called as filter).
2. ***The depth of the output feature map*** which represents the quantity of convolutionally computed filters.

When doing a convolution, the 3x3 or 5x5 windows (or filter) are slid over the 3D input feature map (of an image), halting at each potential region to extract the 3D patch of surrounding features. Then, each of these 3D patches is converted into a 1D vector using a

convolution kernel. The final step is to spatially reconstruct all of these vectors into a 3D output map. The output feature map's spatial locations are identical to every spatial location in the input feature map (for instance, information about the lower-left corner of the output accommodates information about the lower-left corner of the input) [14] [16].

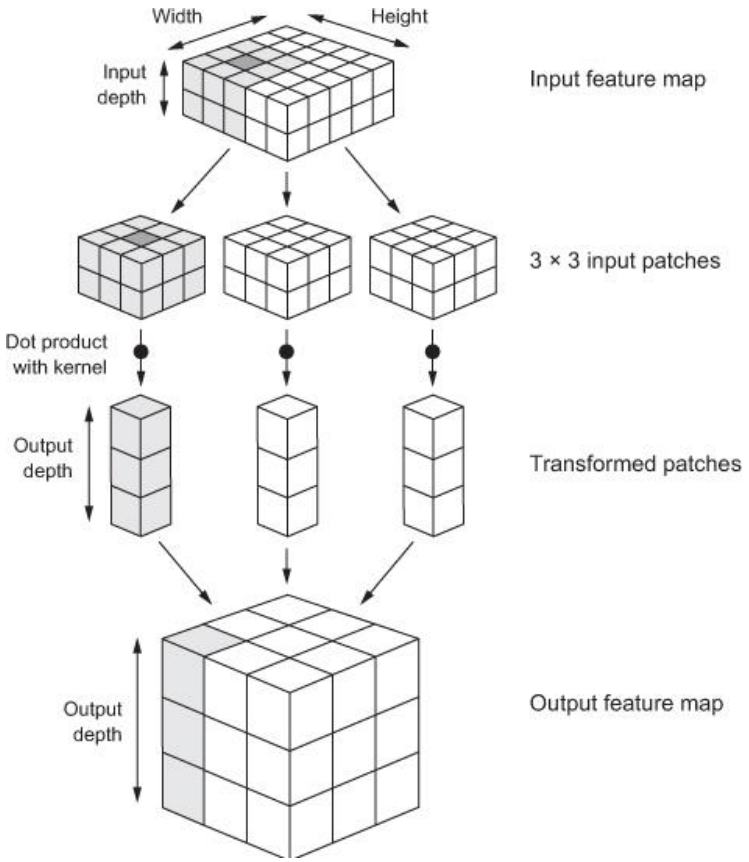



**Figure 3.11 Full convolution operation [14]**

The previous explanations are condensed into equation 3.4, which demonstrates how to compute the output of a given neuron in a convolutional layer. It represents the calculation of the weighted sum of all the inputs, including the bias term [15].

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_{n'}-1} x_{i',j',k'} \cdot w_{u,v,k',k} \quad \text{with} \begin{cases} i' = i \times s_h + u \\ j' = j \times s_w + v \end{cases} \tag{3.4}$$

Throughout the training process, a CNN learns the values of these filters on its own (however, prior to the training process, we still need to set parameters like the number of filters, filter size, the network's architecture, and so on). Our network gets better at identifying patterns on unseen images as we add more filters because more image features are extracted as we add more filters [16].

**Rectified Linear Unit (ReLU)**

Rectified Linear Unit, or ReLU, refers to a non-linear operation. Since the majority of the real-world data, we would like our ConvNet to learn would be non-negative linear values, ReLU's objective is to bring non-linearity into our ConvNet. The ReLU function's primary advantage over other activation functions is that it does not simultaneously activate all of the neurons. As a result, the neurons won't stop activating till the outcome of the linear transformation is smaller than 0. The ReLU function, in contrast to the sigmoid and tanh functions, is much more computationally efficient because only a small subset of neurons are activated [7] [16] [17].

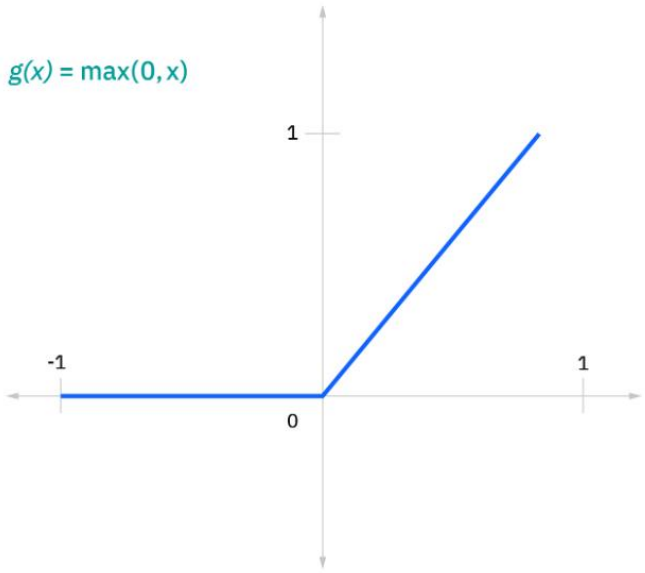



**Figure 3.12 ReLU activation function**

**Max-pooling or sub-sampling layer**

The process of max pooling entails extracting windows from the input feature maps in order to output the maximum value for each channel. Although conceptually similar to convolution, it transforms local patches using a hardcoded max tensor operation rather than a learned linear transformation (the convolution kernel). Max pooling typically uses 2x2

window and stride 2, downsampling the feature maps by a factor of 2, which is a significant difference from convolution. On the other hand, convolution is normally performed with 3x3 window and no stride (stride 1) [14] [19].

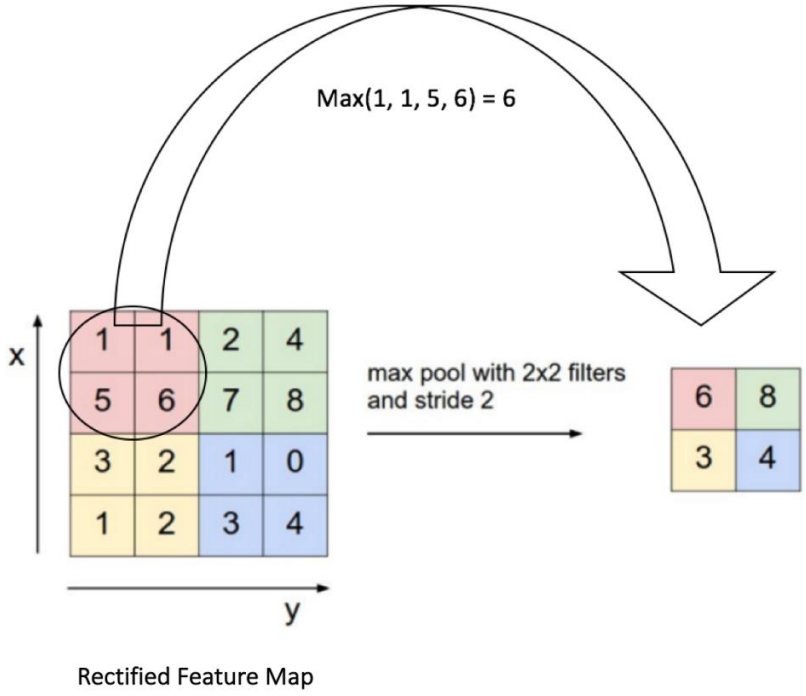



**Figure 3.13 Max Pooling operation with a 2x2 window on a Rectified Feature map [18]**

Most frequently, feature maps are downsampled in order to increase the effective receptive field using a max-pooling (or average pooling) layer. You can think of max-pooling as a particular kind of convolution (with the nonlinear max tensor operation). Max pooling eliminates all irrelevant features, preserving only the strongest ones, giving the following layers a cleaner signal to work with. Max pooling also requires a little less computation than average pooling and provides stronger translation invariance [13] [14].

**Fully Connected Layer (or FC Layer)**

This layer takes inputs from every neuron in the previous layer and utilizes each neuron in the current layer to perform operations to generate the output. Since a dense network requires a 1D array of features for each instance, we flatten our matrix into a vector and feed it into a fully connected layer, much like a neural network. These features were merged using fully connected layers to construct a model, which was then used to classify the outputs using an activation function like softmax or sigmoid [7] [15] [20].

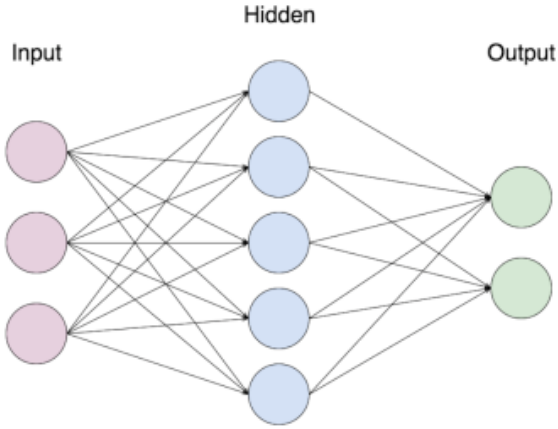



**Figure 3.14 Fully Connected Layer [13]**

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) was launched in 2012, and a sizeable deep convolution neural network developed by Krizhevsky et al., called AlexNet [21], performed exceptionally well on it [22]. In the years that followed, several CNN models, including ZFNet [23], VGGNet [24], GoogleNet [25], ResNet [26], DenseNet [27], CapsNet [28], SENet [29], etc., were inspired by AlexNet's performance.

**LeNet-5 (1998)**

LeCun et al. developed the CNN to classify handwritten digits in 1998. LeNet-5 [30] is the name of the CNN model (figure 3.16), and it comprises seven weighted (trainable) layers. Three convolutional layers (C1, C3, C5), two average pooling layers (S2, S4), one fully connected layer (F6), and one output layer make up the layers. Before performing a pooling procedure, nonlinearity was included using a sigmoid function. The output layer classified 10 digits using Euclidean Radial Basis Function units (RBF) [31].

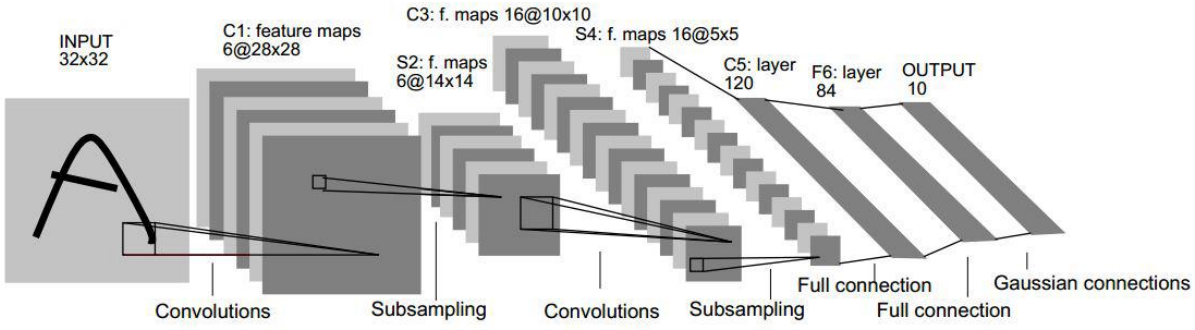



**Figure 3.15 LeNet-5's Architecture [30]**

Table-1 represents the LeNet-5 architecture details of different layers.

**Table 2 LeNet-5 architecture details of different layers**

| Layer | Type | Maps | Size | Kernel Size | Stride | Activation |
|---|---|---|---|---|---|---|
| Out | Fully connected | - | 10 | - | - | RBF |
| F6 | Fully connected | - | 84 | - | - | tanh |
| C5 | Convolution | 120 | 1 × 1 | 5 × 5 | 1 | tanh |
| S4 | Avg pooling | 16 | 5 × 5 | 2 × 2 | 2 | tanh |
| C3 | Convolution | 16 | 10 × 10 | 5 × 5 | 1 | tanh |
| S2 | Avg pooling | 6 | 14 × 14 | 2 × 2 | 2 | tanh |
| C1 | Convolution | 6 | 28 × 28 | 5 × 5 | 1 | tanh |
| In | Input | 1 | 32 × 32 | - | - | - |

**AlexNet (2012)**

Krizhevky et al. created a sizable deep CNN named AlexNet [21] to classify data from ImageNet [32] in 2012. Similar to LeNet-5 in architecture, but significantly larger. There are eight trainable layers in it. There are 3 fully connected layers and 5 convolutional layers among them. After convolutional and FC layers, rectified linear unit (ReLU) [33] non-linearity allowed their model to be trained more quickly than similarly constructed networks using tanh units. Table 2 demonstrates the architecture of AlexNet.

**Table 3 AlexNet architecture**

| Layer | Type | Maps | Size | Kernel Size | Stride | Padding | Activation |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Out | Fully connected | - | 1000 | - | - | - | Softmax |
| F10 | Fully connected | - | 4096 | - | - | - | ReLU |
| F9 | Fully connected | - | 4096 | - | - | - | ReLU |
| S8 | Max pooling | 256 | 6 × 6 | 3 × 3 | 2 | valid | - |
| C7 | Convolution | 256 | 13 × 13 | 3 × 3 | 1 | same | ReLU |
| C6 | Convolution | 384 | 13 × 13 | 3 × 3 | 1 | same | ReLU |
| C5 | Convolution | 384 | 13 × 13 | 3 × 3 | 1 | same | ReLU |
| S4 | Max pooling | 256 | 13 × 13 | 3 × 3 | 2 | valid | - |
| C3 | Convolution | 256 | 27 × 27 | 5 × 5 | 1 | same | ReLU |
| S2 | Max pooling | 96 | 27 × 27 | 3 × 3 | 2 | valid | - |
| C1 | Convolution | 96 | 55 × 55 | 11 × 11 | 4 | valid | ReLU |
| In | Input | 3 (RGB) | 227 × 227 | - | - | - | - |

**VGGNet (2014)**

The deeper configuration of AlexNet [21] was exploited by Simonyan and Zisserman, and they proposed VGGNet [24]. In order to make the network deeper while keeping other parameters fixed, they used tiny filters of size 3x3 for each layer. Six different CNN configurations were used: A, A-LRN, B, C, D (VGG16) and E (VGG19) with 11, 11, 13, 16, 16, and 19 weighted layers, respectively. In order to increase non-linearity, the authors added three 1x1 filters to model C's sixth, ninth, and twelfth convolution layers. Additionally, the effective receptive field of a pack of three 3x3 convolution layers (with stride 1) is equal to that of a single 7x7 convolution layer. This update reduces the number

of parameters in the network and increases non-linearity by substituting a single 7x7 layer for a group of three 3x3 convolution layers [13].

**GoogLeNet (2015)**

Szegedy et al. proposed the architecture of GoogLeNet [25] which is different from traditional CNN. Utilizing parallel filters (called inception module [34]) of size 1x1, 3x3, and 5x5 in each convolution layer, they have increased the number of units in each layer (conv layer). The number of layers has also been raised to 22. The model is applicable to embedded and mobile systems. To address varied scales, they incorporated a number of weighted Gabor filters [35] of various sizes in the inception architecture. Instead of using the naïve form of inception module, they employed inception module with dimensionality reduction to make the architecture computationally efficient. Despite having 22 layers, GoogLeNet uses 12 times less parameters than AlexNet, but its accuracy is noticeably higher. ReLU non-linearity is used in every convolution, reduction, and projection layer. Instead of using fully connected layers, they employed average pooling layers. To mitigate the vanishing gradient problem and overfitting, they have deployed auxiliary classifiers, which are essentially smaller CNNs, on top of some inception modules [13].

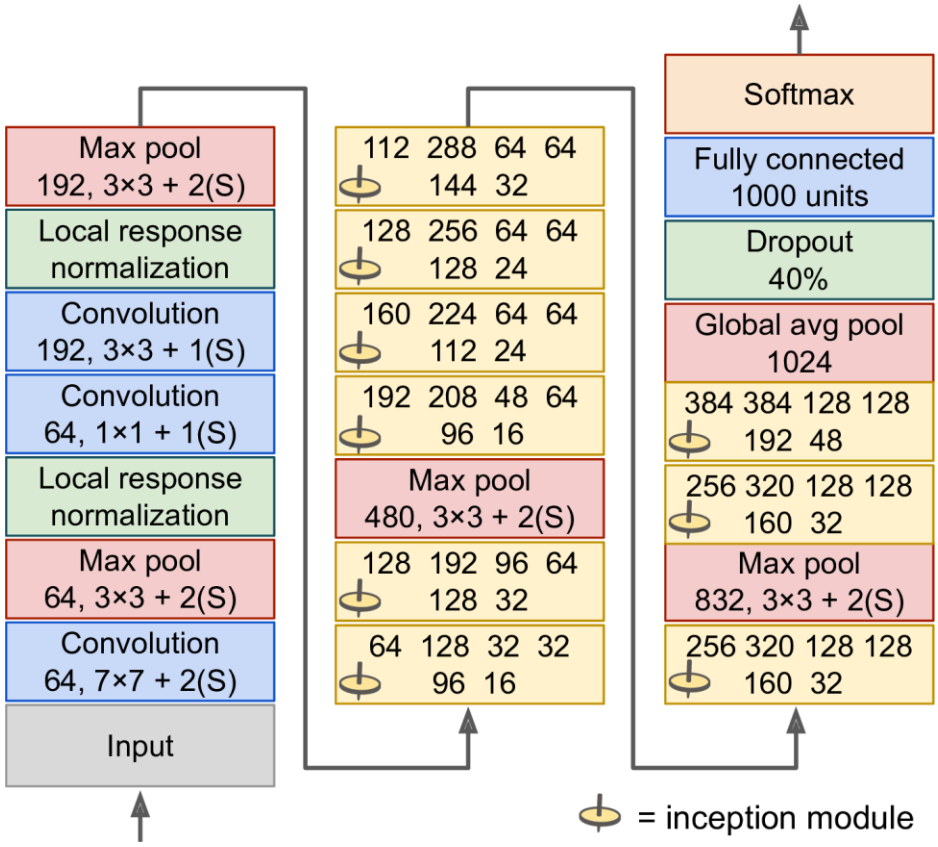



**Figure 3.16 Architecture of GoogLeNet [15]**

**ResNet (2015)**

With a top-five error rate under 3.6%, Kaiming He et al. used a Residual Network (or ResNet [26]) to win the ILSVRC 2015 challenge. The winning strategy made use of a 152-layer, exceptionally deep CNN (other variants had 34, 50, and 101 layers). The general trend of deeper and deeper models with fewer and fewer parameters was validated. Using skip connections, also known as shortcut connections, allows you to train such a deep network since they allow you to add the signal coming into one layer to the output of a layer that is slightly higher up the stack [15].

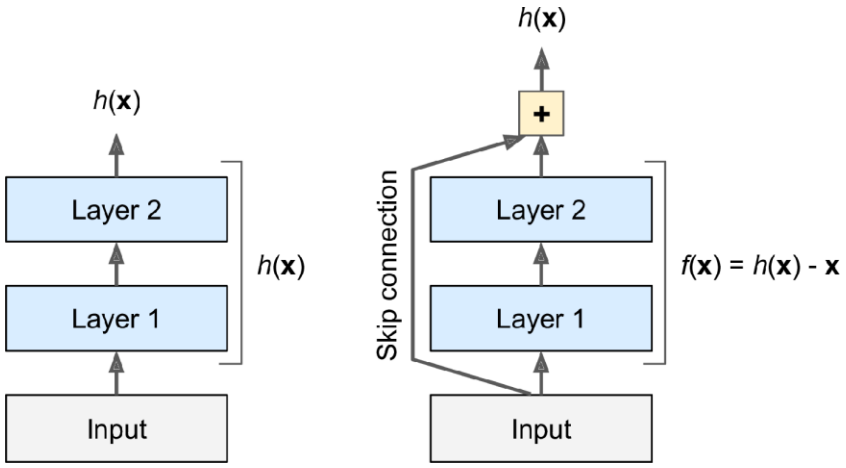



**Figure 3.17 Residual Learning [15]**

Stacked residual blocks of 3x3 convolutional layer make up the ResNet architecture. They have employed a stride of 2 and have periodically doubled the number of filters. They have employed a 7x7 convolutional layer as the initial layer. At the very end, they didn't employ any fully connected layers. In the ILSVRC-2014 competition, different ResNet depths (34, 50, 101, and 152) were used. For CNNs with depths greater than 50, the "bottleneck" layer was utilized to reduce dimensionality and increase efficiency, similar to GoogLeNet. Three convolution layers 1x1, 3x3, and 1x1, make up their bottleneck architecture. Despite being 8 times deeper than VGG nets, the 152 Layer ResNet has lower complexity [13].

## 3.4 Choosing a correct activation function & its type

Data can be mapped into dimensions more effectively with an appropriate activation function [42] [43]. We employ the appropriate activation function and value range to provide the desired output:

- We employ a dense layer for ***regression*** (with unbounded output) without explicitly applying an activation function.
- When dealing with a ***binary classification problem (two-class classification)***, we often utilize a dense layer with a single output followed by the sigmoid function (also known as the logistic sigmoid function), which reduces the output to a value between 0 and 1, encoding a probability.
- We often employ a dense layer with outputs, followed by the softmax function, for ***multi-class classification***. Both the sigmoid and softmax algorithms produce

probabilities as values between 0 and 1. However, the softmax function ensures that the outputs add to 1 and accepts a vector as input before outputting a normalised vector.

**Types of activation functions:**

A. ***Sigmoid function***: In machine learning, the sigmoid function [38] serves as an activation function that is used to introduce non-linearity into a model; in other words, it determines which values should be passed as output and which should not. The logistic function is another name for this entity. In the output node of a classification model with two classes, this function is frequently employed. In deep learning models (like the LSTM model), it is also employed as the "gate" function to simulate Boolean logic gates. The function is differentiable. Therefore, we can determine the sigmoid curve's slope between any two points. A neural network may become stuck during training as a result of the logistic sigmoid function [36] [37].

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.5}$$


**Figure 3.18 Sigmoid activation function [39]**

B. ***Tanh function (Hyperbolic Tangent)***: The Tanh function (or "tanh") is another name for the hyperbolic tangent activation function. It even shares the same S-shape

with the sigmoid activation function. This activation function, which solely maps negative input into negative quantities and has a range of -1 to 1, performs marginally better than the sigmoid function. Any real value may be used as an input, and the function returns values between -1 and 1. The larger the input, the closer the output value will be to 1, whereas the smaller the input, the closer the output will be to -1. While the derivative of the function is not monotonic, the function itself is [38,40].

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{3.6}$$

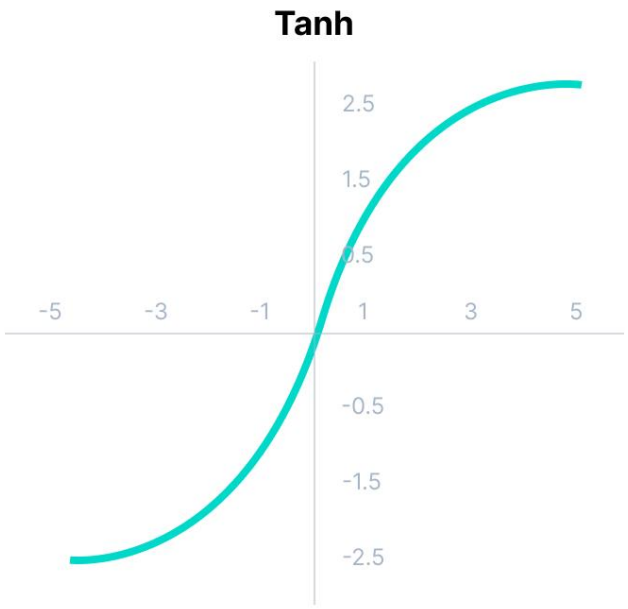



**Figure 3.19 Tanh Function [39]**

C. ***Rectified Linear Unit function (ReLU)***: One of the most popular activation functions is Rectified Linear Unit function (ReLU) [41], due to the fact that practically all convolutional neural networks or deep learning employ it. In networks with several layers, it aids in addressing the vanishing gradient issue. ReLU is a piecewise linear function that returns zero if the input is negative and the output the input directly if the input is positive. Since, the model that uses is simpler to train and frequently results in higher performance, it has evolved into the standard activation function for many different kinds of neural networks.

D. ***Softmax function:*** A vector of numbers is transformed into a vector of probabilities via the mathematical function known as Softmax, where the probability of every value is proportional to the relative scale of every value in the vector. Similar to how sigmoid activation operates, softmax is primarily utilized at the last layer, or output layer, for decision making. The softmax basically assigns values to the input variables based on their weights, and the sum of these weights ultimately equals one. Both sigmoid and softmax are equally comprehensible for binary classification, however when dealing with multi-class classification problems, we typically combine softmax and cross-entropy [17] [39].

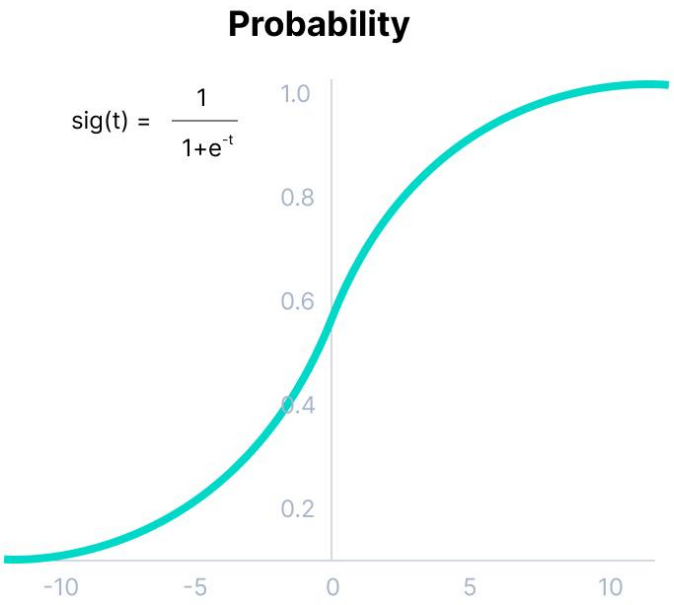



**Figure 3.20 Softmax Function [39]**

These are some of the advantages & disadvantages of an activation function in a model:

**Table 4 Advantages & disadvantages of activation functions**

| Type of Function | Advantages | Disadvantages |
|---|---|---|
| Sigmoid | • The function's nature is nonlinear & its combinations are also nonlinear. | • Gradients are saturated and eliminated by sigmoids which results in the issue of vanishing gradients. |

| | | |
|---|---|---|
| | • In contrast to the step function, it produces an analogue activation. | • The network either won't learn further or is extremely slow. |
| Tanh | • The derivatives are steeper because the gradient is greater for tanh than sigmoid. | • Tanh also deals with vanishing gradient issue. |
| ReLU | • The vanishing gradient issue is either avoided or resolved.<br>• ReLU requires fewer mathematical operations than tanh and sigmoid, making it less computationally expensive. | • It ought to only be applied in a neural network model's hidden layers.<br>• During training, some gradients can become brittle and even dissolve. It may result in a weight change that prevents future activation on any data point. Therefore, ReLU may possibly cause dead neurons. |

## 3.5 Vision Transformers (VITs)

Unlike the conventional CNN architectures, which typically use filters with a local receptive field the attention mechanism employed by the Vision Transformer [66] allows it to be used over different regions of the image and to integrate information across the entire image. Convolutional neural networks' requirement for integrating the global relationship among the pixels was one of the primary issues in the computer vision field. A vision transformer that used a self-attention mechanism to describe the pixel dependency among pixels was offered as a solution to this problem. When training on fewer datasets, Vision Transformer

(ViT) shows a generally lesser inductive bias compared to convolutional neural networks (CNN), which increases reliance on model regularization or data augmentation (AugReg).

### Vision Transformer (ViT) Architecture

The overall architecture of the vision transformer model is given as follows in a step-by-step manner:

1. Split an image into patches (fixed sizes)
2. Flatten the image patches
3. Create lower-dimensional linear embeddings from these flattened image patches
4. Include positional embeddings
5. Feed the sequence as an input to a state-of-the-art transformer encoder
6. Pre-train the ViT model with image labels, which is then fully supervised on a big dataset
7. Fine-tune the downstream dataset for image classification

Without the need for image-specific biases, the vision transformer model in computer vision leverages ***multi-head self-attention (MHSA)*** [66]. The transformer encoder processes the positional embedding patches that the model creates from the images. It accomplishes this in order to comprehend the features both locally and globally present in the image. Not to mention, the ViT requires less training time and has a higher precision rate on a big dataset.

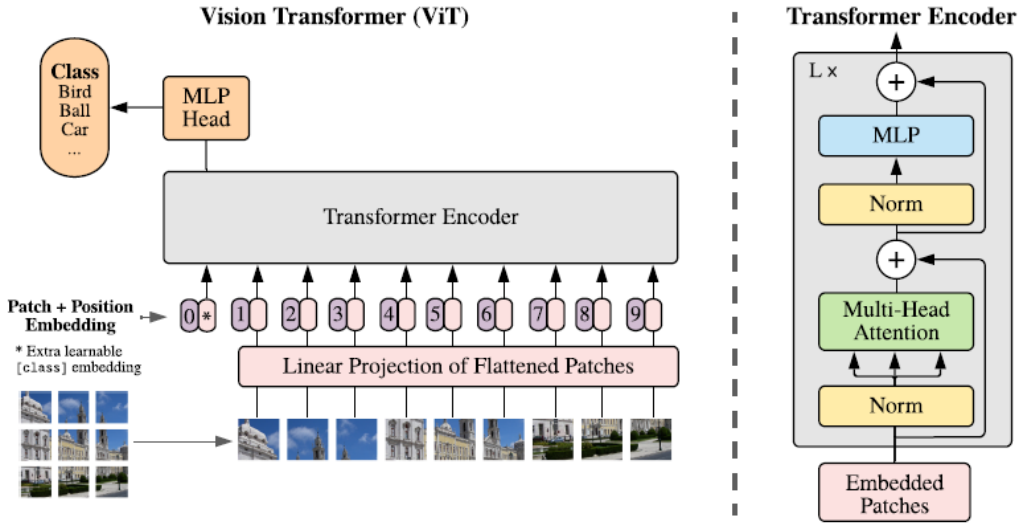



**Figure 3.21 Model Architecture [66]**

# 4. RESEARCH DESIGN & METHODOLOGY

In this study, we applied and adopted Convolutions to Vision Transformers (CvT) (Haiping Wu et al. 2021) [109], which enhanced the performance and effectiveness of Vision Transformer (ViT) [66] by incorporating convolutions into ViT to produce the best of both designs. The conducted model is achieved via two main modifications: a hierarchy of Transformers with a new convolutional token embedding and a convolutional Transformer block using a convolutional projection. According to [109], It is safe to eliminate positional encoding, a fundamental element of current Vision Transformers, simplifying the design for higher resolution vision tasks. While preserving the benefits of Transformers (i.e., dynamic attention, global context, and better generalization), these modifications bring useful convolutional neural network (CNN) properties to the ViT architecture, namely shift, scale, and distortion invariance. The model's incorporation of convolutions enables the capture of local context. ViT is deficient in a number of important characteristics that CNNs naturally possess and that make them particularly well-suited to tackle vision-related problems. Images, for instance, typically contain a strong 2D local structure with highly correlated spatially neighbouring pixels. By utilizing local receptive fields, shared weights, and spatial subsampling [112], the CNN architecture compels the capture of this local structure and, in doing so, partially achieves shift, scale, and distortion invariance. Additionally, convolutional kernels' hierarchical structure enables them to learn visual patterns with varied degrees of complexity, ranging from basic low-level edges and textures to more complex higher order semantic patterns, that incorporate local spatial context.

## *4.1 Convolutional Token Embedding*

This layer enables us to modify the token feature dimension and the number of tokens at each stage by changing the convolution operation's parameters. This way, we gradually decrease the token sequence (or feature resolution) in each stage while increasing the token width (or feature dimension). This gives the tokens the potential to depict increasingly complex visual patterns over larger and larger spatial footprints & allow for spatial Downsampling which is comparable to feature layers of CNN.

Provided a 2D image or a 2D output token map that has been reshaped from a previous stage $x_{i-1} \in \mathbb{R}^{Hi-1\ X\ Wi-1\ X\ Ci-1}$ as the input stage $i$, we learn a function $f(\cdot)$ that maps $x_{i-1}$ into

new tokens $f(x_{i-1})$ with a channel size $C_i$, where $f(\cdot)$ is a 2D convolution operation with kernel size $s\ X\ s$, stride $s - o$ and $p$ padding (in order to deal with boundary conditions). The new token map $f(x_{i-1}) \in \mathbb{R}^{Hi\ X\ Wi\ X\ Ci}$ contains height & width.

$$H_i = \left[\frac{H_{i-1}+2p-s}{s-o} + 1\right], W_i = \left[\frac{W_{i-1}+2p-s}{s-o} + 1\right] \quad (4.1)$$

## *4.2 Convolutional Projection layer*

In this study, this layer is formed by depth-wise separable convolutions in place of the original position-wise linear projection for Multi-Head Self-Attention (MHSA). The traditional position-wise linear projection used in ViT is replaced by convolutional projection for the embeddings of the query, key, and value, respectively. The classification token of the last stage output is then used to predict the class using an MLP (i.e., fully connected) Head.

### *4.2.1 Implementation Specifics*

The original position-wise linear projection employed in ViT (figure 4.1 (a)) & (figure 4.1 (b)) displayed our proposed $s\ X\ s\ \ Convolutional\ Projection$. A 2D token map is first created from the reshaped tokens (figure 4.1 (b)). Next, a depth-wise separable convolution layer with kernel size s is used to build a convolutional projection. The projected tokens are finally flattened into 1D for further processing. This can be stated as follows:

$$x_i^{q/k/v} = Flatten\ (Conv2d\ (Reshape2D(x_i);\ s)) \quad (4.2)$$

Where, $x_i^{q/k/v}$ is the token input for $Q/K/V$ matrices at layer $i, x_i$ is the undisturbed token prior to the Convolution Projection, $Conv2d$ is a depth-wise separable convolution implemented by: $Depth-wise\ Conv2d \rightarrow BatchNorm2d \rightarrow Point-wise\ Conv2d$, and $s$ refers to the convolution kernel size. The new Transformer Block that results with the Convolutional Projection layer is a generalisation of the initial Transformer Block framework. A convolution layer with a kernel size of 1x1 might be used to quickly implement the original position-wise linear projection layer.

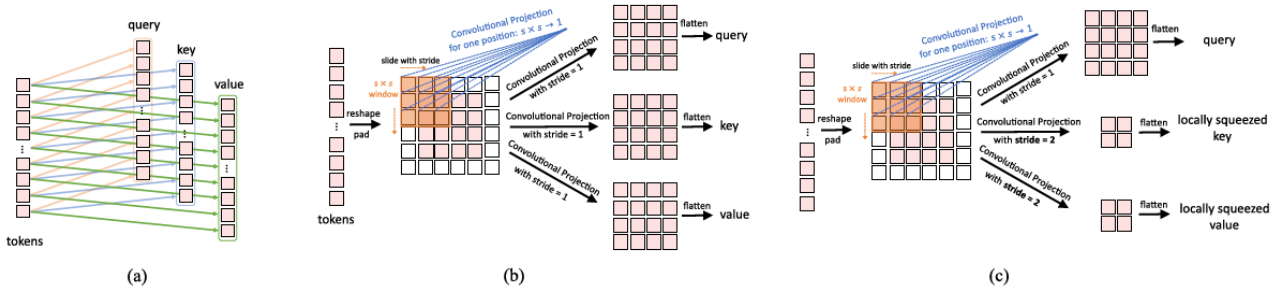



**Figure 4.1 (a) Linear projection in ViT. (b) Convolutional projection. (c) Squeezed convolutional projection. Unless otherwise stated, we use (c) Squeezed convolutional projection by default.**

**Table 5 Architectures for ImageNet classification**

| | Output Size | Layer Name | CvT-13 | CvT-21 | CvT-W24 |
|---|---|---|---|---|---|
| Stage1 | $56 \times 56$ | Conv. Embed. | $7 \times 7$, 64, stride 4 | | $7 \times 7$, 192, stride 4 |
| | $56 \times 56$ | Conv. Proj.<br>MHSA<br>MLP | $\begin{bmatrix} 3 \times 3, 64 \\ H_1 = 1, D_1 = 64 \\ R_1 = 4 \end{bmatrix} \times 1$ | $\begin{bmatrix} 3 \times 3, 64 \\ H_1 = 1, D_1 = 64 \\ R_1 = 4 \end{bmatrix} \times 1$ | $\begin{bmatrix} 3 \times 3, 192 \\ H_1 = 3, D_1 = 192 \\ R_1 = 4 \end{bmatrix} \times 2$ |
| Stage2 | $28 \times 28$ | Conv. Embed. | $3 \times 3$, 192, stride 2 | | $3 \times 3$, 768, stride 2 |
| | $28 \times 28$ | Conv. Proj.<br>MHSA<br>MLP | $\begin{bmatrix} 3 \times 3, 192 \\ H_2 = 3, D_2 = 192 \\ R_2 = 4 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 192 \\ H_2 = 3, D_2 = 192 \\ R_2 = 4 \end{bmatrix} \times 4$ | $\begin{bmatrix} 3 \times 3, 768 \\ H_2 = 12, D_2 = 768 \\ R_2 = 4 \end{bmatrix} \times 2$ |
| Stage3 | $14 \times 14$ | Conv. Embed. | $3 \times 3$, 384, stride 2 | | $3 \times 3$, 1024, stride 2 |
| | $14 \times 14$ | Conv. Proj.<br>MHSA<br>MLP | $\begin{bmatrix} 3 \times 3, 384 \\ H_3 = 6, D_3 = 384 \\ R_3 = 4 \end{bmatrix} \times 10$ | $\begin{bmatrix} 3 \times 3, 384 \\ H_3 = 6, D_3 = 384 \\ R_3 = 4 \end{bmatrix} \times 16$ | $\begin{bmatrix} 3 \times 3, 1024 \\ H_3 = 16, D_3 = 1024 \\ R_3 = 4 \end{bmatrix} \times 20$ |
| Head | $1 \times 1$ | Linear | 1000 | | |
| Params | | | 19.98 M | 31.54 M | 276.7 M |
| FLOPs | | | 4.53 G | 7.13 G | 60.86 G |

Table 4 deals with Input image size of $224\ X\ 224$ by default. Conv. Embed.: Convolutional Token Embedding. Conv. Proj.: Convolutional Projection. $H_i$ and $D_i$ is the number of heads and embedding feature dimension in the $ith$ MHSA module. $R_i$ is the feature dimension expansion ratio in the $ith$ MLP layer.

# 5. ARTEFACT DEVELOPMENT

The artefact was built in Python utilizing both open-source and proprietary libraries. $OS$, $argparse$, $logging$, $time$, $pickle$, and $pprint$ are the primary standard libraries used. The third-party libraries utilized are $pytorch$ & $tensorboardx$ for deep learning frameworks and utilities, $scikit-learn$ for model metrics and evaluation tools, pandas for working with datasets and their labels, and $numpy$ for advanced mathematics and array programming.

*Jupyter*, which enables code to be run in cells with an output directly below each cell, was used to build the code. Being able to observe outcomes and visualize the outputs of certain implementation components while they are being developed improves the development cycle. Additionally, *cells* can be performed in any sequence, which is helpful in adjusting hyperparameters before retraining a model and comparing the outcomes. Additionally, *markdown cells* can be used in notebooks to write comments and headings, improving the code's readability and presentation, and facilitating a more effective distribution of the concepts discussed in this thesis.

In this preliminary stage of development, datasets were chosen, and a data ingestion pipeline was constructed dividing it into training, validation, and test sets, as well as preprocessing the data. The dataset selected for this stage of the project was the 'COVID-19 Radiography Database [113]' which has 1200 COVID-19 positive images, 1341 normal images, and 1345 viral pneumonia images to work with. It is presented as the Winner of the COVID-19 Dataset Award by Kaggle Community which allowed many developers to implement their models using PyTorch or Keras.

The artefact is available in the following GitHub repository (forked from the original GitHub repository): https://github.com/Rahulk9520/CvT-main/tree/main/tools.

# 6. EMPIRICAL EVALUATION

This section provides details on the experimental setup that will be utilized to evaluate the conducted 'CvT' method stated in Section 4, utilizing the artefact implementation given in Section 5. To ascertain whether the goals and objectives of the research from Section 1.2 have been achieved, research questions are created, and experiments are carried out to provide answers to these questions.

## *6.1 Dataset*

The evaluation's chosen dataset is the 'COVID-19 Radiography Database [113]' which has 1200 COVID-19 positive images, 1341 normal images, and 1345 viral pneumonia images. The dataset is divided into patients having 'Covid, Normal & Viral Pneumonia' images. We prepared the dataset for training & testing according to the method provided (in the

documentation). Furthermore, followed the steps given in the documentation for running the conducted model in Google Cloud Platform (GCP). We have further divided our dataset into 70:30 (70% for training & 30% for validation) for training & validation.

For evaluation, we employ both the ImageNet dataset (1.3M images) with 1k classes and superset ImageNet-22k (14M images) with 22k classes. Moreover, we transfer the models pretrained on ImageNet-22k to subsequent tasks (for testing) including CIFAR-10/100 [115], Oxford-IIIT-Pet [116], Oxford-IIIT-Flower [117], following [118, 119].

## *6.2 Model Variants*

By altering the quantity of Transformer blocks at each stage and the hidden feature dimension applied, as shown in Table 2, we instantiate models with various parameters and FLOPs. It is modified in three steps. With 19.98M and 31.54M parameters, respectively, we define CvT-13 and CvT-21 as basic models. Convolutional vision Transformer with X total transformer blocks is known as CvT-X. In order to validate the suggested architecture's capacity to scale, we also experiment with a wider model called CvT-W24 (W stands for Wide), which has a higher token dimension for each stage and results in 298.3M parameters.

## *6.3 Efficiency Considerations*

Our Convolutional Projection layer's design offers ***two main efficiency advantages***.

We start by using effective convolutions. Convolutional Projection would require $s^2C^2$ parameters and $O(s^2C^2T)$ FLOPs if ordinary $s\ X\ s$ convolutions were used directly, where $C$ is the token channel dimension and $T$ is the number of tokens to process. The usual $s\ X\ s$ convolution was instead divided into a depth-wise separable convolution. With respect to the total parameters and FLOPs of the models, each proposed convolutional projection would only add $s^2C$ parameters and $O(s^2CT)$ FLOPs over the initial position-wise linear projection.

Second, we use the proposed Convolutional Projection to lower the MHSA operation's computation costs. By employing a stride greater than 1, the $s\ X\ s$ Convolutional Projection allows for a reduction in the amount of tokens. The key and value projection are subsampled by employing a convolution with a stride greater than 1, as shown in figure 4.1 (c), which

displays the convolutional projection. For key and value projection, we utilize a stride of 2, while query stride remains at 1. This results in a 4 times reduction in computational cost and a 4 times reduction in the number of tokens needed for the later MHSA operation for key and value. This has a little impact on performance because nearby pixels patches in images tend to have redundancy in appearance/semantics.

## *6.4 Training & Fine-tuning the model*

The optimizer we have used is AdamW [114] with the weight decay of 0.05 for our CvT-13, and 0.1 for our CvT-21 and CvT-W24. We train our models with an initial learning rate of 0.02 and a total batch size of 32 for 50 epochs, workers of 3, print frequency of 100 with a cosine learning rate decay scheduler & changed the image data format to 'png'. We used the same strategies for data augmentation and regularization as ViT [108]. All ImageNet models, unless otherwise specified, are trained using 224x224 input data.

We use the ViT's [108] fine-tuning approach. To fine-tune, an SGD optimizer with a momentum of 0.9 is employed. Similar to ViT [108], we pre-train our models at 224x224 resolution and fine-tune at 384x384 resolution. With a total batch size of 512, we fine-tune each model individually over 20,000 steps on ImageNet-1k, 10,000 steps on CIFAR-10 and CIFAR-100, and 500 steps on Oxford-IIIT Pets and Oxford-IIIT Flowers-102.

## *6.5 Research Questions (RQs)*

Based on the aims and objectives of the research outlined in Section 1.2.1, the following research questions have been identified:

***Q1. What benefits the architectural design of the conducted CvT model presents in terms of model optimization, performance & computational cost? (1.2.1.1)***

***Q2. How the modifications in the conducted CvT design impacts the overall performance of the model utilizing hyperparameter tuning of the model? (1.2.1.2)***

Aim 1.2.1 is also addressed by addressing all of these questions utilizing sub-aims 1.2.1.1 & 1.2.1.2. The experimentation covered in this section aims to yield findings that can be applied to these research questions. The study design and methodology's compliance with the aims and objectives of this thesis is thus determined by comparing the findings to the standards established by the research questions.

# 7. EVALUATION AND THREATS TO VALIDITY

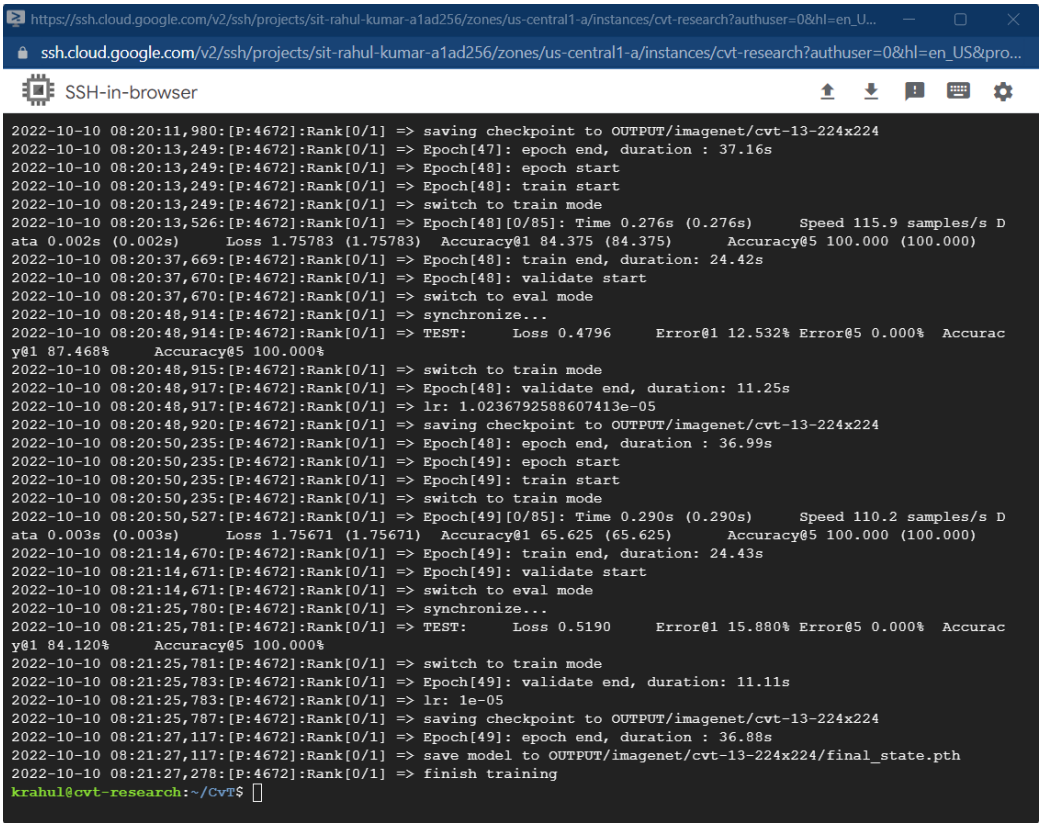
Using the software artefact conducted in Section 5 and the experimental setup outlined in Section 6, experiments were carried out. To ascertain if the aims and objectives of the research identified in Section 1.2.1 have been achieved, the results of these experiments are evaluated in this section using the standards of the research questions from Section 6.5.

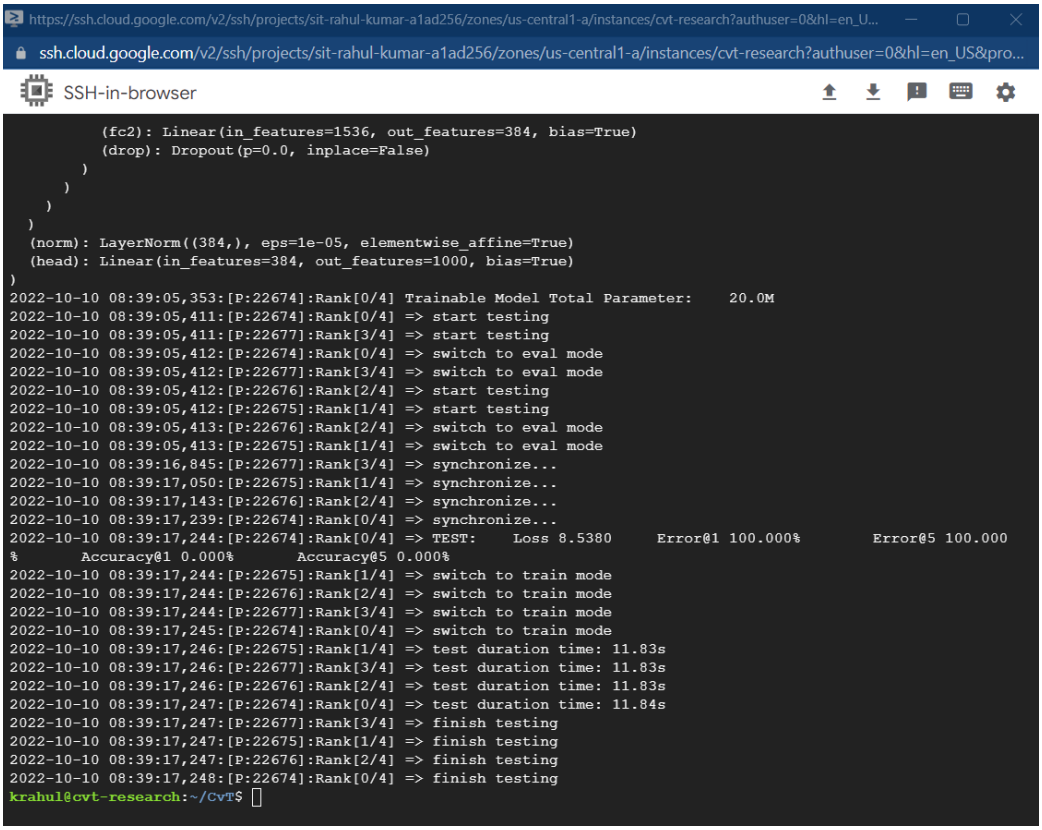
We can model local spatial relationships through the network by introducing Convolutional Projections for each Transformer block with Convolutional Token Embedding. As demonstrated by our research, this built-in trait enables eliminating the position embedding from the network without degrading performance, simplifying design for vision tasks with variable input resolution.

CvT model has the ability to achieve higher accuracy with fewer parameters and FLOPs than Transformer-based models due to its design modifications. Additionally, when we are dealing with more data, conducted wide model CvT-W24* pretrained on ImageNet-22k can outperform some of the previous best Transformer-based models with comparable numbers of model parameters and FLOPs. Convolutional Token Embedding enables CvT model spatial relationships without position embedding, in addition to improving performance.

For less computational expense and memory utilization, Convolutional Projection with stride 2 for key and value is used by default. Accuracy will increase if the proposed Convolutional Projection is used in place of the initial Position-wise Linear Projection. Additionally, as more stages employ the design, performance will keep getting better, demonstrating the efficacy of this approach as a modelling technique.

By training the CvT model on our dataset utilizing GCP we are able to achieve 84.12% accuracy overall.


https://ssh.cloud.google.com/v2/ssh/projects/sit-rahul-kumar-a1ad256/zones/us-central1-a/instances/cvt-research?authuser=0&hl=en_U...
ssh.cloud.google.com/v2/ssh/projects/sit-rahul-kumar-a1ad256/zones/us-central1-a/instances/cvt-research?authuser=0&hl=en_US&pro...
SSH-in-browser
2022-10-10 08:20:11,980:[P:4672]:Rank[0/1] => saving checkpoint to OUTPUT/imagenet/cvt-13-224x224
2022-10-10 08:20:13,249:[P:4672]:Rank[0/1] => Epoch[47]: epoch end, duration : 37.16s
2022-10-10 08:20:13,249:[P:4672]:Rank[0/1] => Epoch[48]: epoch start
2022-10-10 08:20:13,249:[P:4672]:Rank[0/1] => Epoch[48]: train start
2022-10-10 08:20:13,249:[P:4672]:Rank[0/1] => switch to train mode
2022-10-10 08:20:13,526:[P:4672]:Rank[0/1] => Epoch[48][0/85]: Time 0.276s (0.276s) Speed 115.9 samples/s D
ata 0.002s (0.002s) Loss 1.75783 (1.75783) Accuracy@1 84.375 (84.375) Accuracy@5 100.000 (100.000)
2022-10-10 08:20:37,669:[P:4672]:Rank[0/1] => Epoch[48]: train end, duration: 24.42s
2022-10-10 08:20:37,670:[P:4672]:Rank[0/1] => Epoch[48]: validate start
2022-10-10 08:20:37,670:[P:4672]:Rank[0/1] => switch to eval mode
2022-10-10 08:20:48,914:[P:4672]:Rank[0/1] => synchronize...
2022-10-10 08:20:48,914:[P:4672]:Rank[0/1] => TEST: Loss 0.4796 Error@1 12.532% Error@5 0.000% Accurac
y@1 87.468% Accuracy@5 100.000%
2022-10-10 08:20:48,915:[P:4672]:Rank[0/1] => switch to train mode
2022-10-10 08:20:48,917:[P:4672]:Rank[0/1] => Epoch[48]: validate end, duration: 11.25s
2022-10-10 08:20:48,917:[P:4672]:Rank[0/1] => lr: 1.0236792588607413e-05
2022-10-10 08:20:48,920:[P:4672]:Rank[0/1] => saving checkpoint to OUTPUT/imagenet/cvt-13-224x224
2022-10-10 08:20:50,235:[P:4672]:Rank[0/1] => Epoch[48]: epoch end, duration : 36.99s
2022-10-10 08:20:50,235:[P:4672]:Rank[0/1] => Epoch[49]: epoch start
2022-10-10 08:20:50,235:[P:4672]:Rank[0/1] => Epoch[49]: train start
2022-10-10 08:20:50,235:[P:4672]:Rank[0/1] => switch to train mode
2022-10-10 08:20:50,527:[P:4672]:Rank[0/1] => Epoch[49][0/85]: Time 0.290s (0.290s) Speed 110.2 samples/s D
ata 0.003s (0.003s) Loss 1.75671 (1.75671) Accuracy@1 65.625 (65.625) Accuracy@5 100.000 (100.000)
2022-10-10 08:21:14,670:[P:4672]:Rank[0/1] => Epoch[49]: train end, duration: 24.43s
2022-10-10 08:21:14,671:[P:4672]:Rank[0/1] => Epoch[49]: validate start
2022-10-10 08:21:14,671:[P:4672]:Rank[0/1] => switch to eval mode
2022-10-10 08:21:25,780:[P:4672]:Rank[0/1] => synchronize...
2022-10-10 08:21:25,781:[P:4672]:Rank[0/1] => TEST: Loss 0.5190 Error@1 15.880% Error@5 0.000% Accurac
y@1 84.120% Accuracy@5 100.000%
2022-10-10 08:21:25,781:[P:4672]:Rank[0/1] => switch to train mode
2022-10-10 08:21:25,783:[P:4672]:Rank[0/1] => Epoch[49]: validate end, duration: 11.11s
2022-10-10 08:21:25,783:[P:4672]:Rank[0/1] => lr: 1e-05
2022-10-10 08:21:25,787:[P:4672]:Rank[0/1] => saving checkpoint to OUTPUT/imagenet/cvt-13-224x224
2022-10-10 08:21:27,117:[P:4672]:Rank[0/1] => Epoch[49]: epoch end, duration : 36.88s
2022-10-10 08:21:27,117:[P:4672]:Rank[0/1] => save model to OUTPUT/imagenet/cvt-13-224x224/final_state.pth
2022-10-10 08:21:27,278:[P:4672]:Rank[0/1] => finish training
krahul@cvt-research:~/CvT$


https://ssh.cloud.google.com/v2/ssh/projects/sit-rahul-kumar-a1ad256/zones/us-central1-a/instances/cvt-research?authuser=0&hl=en_U...
ssh.cloud.google.com/v2/ssh/projects/sit-rahul-kumar-a1ad256/zones/us-central1-a/instances/cvt-research?authuser=0&hl=en_US&pro...
SSH-in-browser
(fc2): Linear(in_features=1536, out_features=384, bias=True)
(drop): Dropout(p=0.0, inplace=False)
)
)
)
)
(norm): LayerNorm((384,), eps=1e-05, elementwise_affine=True)
(head): Linear(in_features=384, out_features=1000, bias=True)
)
2022-10-10 08:39:05,353:[P:22674]:Rank[0/4] Trainable Model Total Parameter: 20.0M
2022-10-10 08:39:05,411:[P:22674]:Rank[0/4] => start testing
2022-10-10 08:39:05,411:[P:22677]:Rank[3/4] => start testing
2022-10-10 08:39:05,412:[P:22674]:Rank[0/4] => switch to eval mode
2022-10-10 08:39:05,412:[P:22677]:Rank[3/4] => switch to eval mode
2022-10-10 08:39:05,412:[P:22676]:Rank[2/4] => start testing
2022-10-10 08:39:05,412:[P:22675]:Rank[1/4] => start testing
2022-10-10 08:39:05,413:[P:22676]:Rank[2/4] => switch to eval mode
2022-10-10 08:39:05,413:[P:22675]:Rank[1/4] => switch to eval mode
2022-10-10 08:39:16,845:[P:22677]:Rank[3/4] => synchronize...
2022-10-10 08:39:17,050:[P:22675]:Rank[1/4] => synchronize...
2022-10-10 08:39:17,143:[P:22676]:Rank[2/4] => synchronize...
2022-10-10 08:39:17,239:[P:22674]:Rank[0/4] => synchronize...
2022-10-10 08:39:17,244:[P:22674]:Rank[0/4] => TEST: Loss 8.5380 Error@1 100.000% Error@5 100.000
% Accuracy@1 0.000% Accuracy@5 0.000%
2022-10-10 08:39:17,244:[P:22675]:Rank[1/4] => switch to train mode
2022-10-10 08:39:17,244:[P:22676]:Rank[2/4] => switch to train mode
2022-10-10 08:39:17,244:[P:22677]:Rank[3/4] => switch to train mode
2022-10-10 08:39:17,245:[P:22674]:Rank[0/4] => switch to train mode
2022-10-10 08:39:17,246:[P:22675]:Rank[1/4] => test duration time: 11.83s
2022-10-10 08:39:17,246:[P:22677]:Rank[3/4] => test duration time: 11.83s
2022-10-10 08:39:17,246:[P:22676]:Rank[2/4] => test duration time: 11.83s
2022-10-10 08:39:17,247:[P:22674]:Rank[0/4] => test duration time: 11.84s
2022-10-10 08:39:17,247:[P:22677]:Rank[3/4] => finish testing
2022-10-10 08:39:17,247:[P:22675]:Rank[1/4] => finish testing
2022-10-10 08:39:17,247:[P:22676]:Rank[2/4] => finish testing
2022-10-10 08:39:17,248:[P:22674]:Rank[0/4] => finish testing
krahul@cvt-research:~/CvT$

By using neural architecture search (NAS), we may further enhance the conducted CvT architecture designing in terms of model parameters and FLOPs. As we have only trained our dataset to the CvT model, it still requires testing in order to validate the results of our training. Performance can suffer if non-overlapping Patch Embedding is used in place of the convolutional token embedding. The accuracy may somewhat decrease when utilizing both location embedding and convolutional token embedding.

# 8. RESULTS AND FUTURE WORK

The development of a novel method utilizing Transformers for COVID-19 detection for the multiclass classification problem (COVID-19, Pneumonia, and Normal cases) using X-ray images was examined in detail by conducting an in-depth literature review in order to determine the current state-of-the-art in COVID-19 detection using X-ray images utilizing Transformers and to identify any limitations and research gaps that could be filled.

We have provided a thorough analysis of adding convolutions to the Vision Transformer architecture in order to combine the advantages of Transformers with CNNs for image recognition applications. Numerous studies have shown that the newly proposed convolutional token embedding and projection techniques, as well as the multi-stage network design made possible by convolutions, enable our implemented CvT architecture to achieve greater performance while maintaining computational efficiency. Additionally, because convolutions have built-in local context structure, CvT no longer needs a position embedding, which could be advantageous for adapting to a variety of vision tasks demanding different input resolutions.

For future work, this study on CvT model using our dataset only deals with research & training. It requires testing & calculating the performance metrics of the model for further validation in order to analyze the impact of Convolution Token Embedding & Convolution Projection. Furthermore, we can expand our dataset using ‘COVIDx’ dataset for further validation using GCP with proper distribution of images (according to the dataset division).

# BIBLIOGRAPHY


[1] Sherstinsky, A., 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, p.132306.

[2] Dhruv, P. and Naskar, S., 2020. Image classification using convolutional neural network (CNN) and recurrent neural network (RNN): a review. Machine learning and information processing, pp.367-381.

[3] Deepak Kumar Sharma, Mayukh Chatterjee, Gurmehak Kaur, Suchitra Vavilala, 3 - Deep learning applications for disease diagnosis, Editor(s): Deepak Gupta, Utku Kose, Ashish Khanna, Valentina Emilia Balas, Deep Learning for Medical Applications with Unique Data, Academic Press, 2022, Pages 31-51, ISBN 9780128241455, (https://www.sciencedirect.com/science/article/pii/B9780128241455000058)

[4] Hochreiter, S., 1998. Recurrent neural net learning and vanishing gradient. International Journal Of Uncertainity, Fuzziness and Knowledge-Based Systems, 6(2), pp.107-116.

[5] Avijeet. 2022. Recurrent Neural Network (RNN) Tutorial: Types, Examples, LSTM and More. https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn. September 5, 2022.

[6] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.

[7] Prabhu. 2018. Understanding of Convolutional Neural Network (CNN)—Deep Learning. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148. March 4, 2018.

[8] Saul. 2022. LSTM Recurrent Neural Networks—How to Teach a Network to Remember the Past. https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c2ff22e. September 5, 2022.

[9] Ye, A., 2020. Long Short-Term Memory Networks Are Dying: What's Replacing It?. Medium. https://medium.com/mlearning-ai/long-short-term-memory-networks-are-dying-whats-replacing-it-5ff3a99399fe. September 7, 2022.

[10] Albawi, S., Mohammed, T.A. and Al-Zawi, S., 2017, August. Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET) (pp. 1-6). Ieee.

[11] O'Shea, K. and Nash, R., 2015. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.

[12] Guo, T., Dong, J., Li, H. and Gao, Y., 2017, March. Simple convolutional neural network on image classification. In 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA) (pp. 721-724). IEEE.

[13] Sultana, F., Sufian, A. and Dutta, P., 2018, November. Advancements in image classification using convolutional neural network. In 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN) (pp. 122-129). IEEE.

[14] Chollet, F. (2017) Deep learning with python. New York, NY: Manning Publications.

[15] Geron, A. (2017) Hands-on machine learning with scikit-learn and TensorFlow. Sebastopol, CA: O'Reilly Media.

[16] ujjwalkarn (2016) An intuitive explanation of Convolutional Neural Networks, Ujjwal Karn. https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/. September 5, 2022.

[17] Roopa, C. M. (2022) Activation functions in artificial neural network - Roopa CM, Medium. https://medium.com/@roopa.usa.cm/activation-functions-in-artificial-neural-network-454897f859ba. September 10, 2022.

[18] CS231n Convolutional Neural Networks for Visual Recognition, Stanford

[19] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Computer Vision–ECCV 2014, pp. 818–833. Springer (2014).

[20] Nebauer, C.: Evaluation of convolutional neural networks for visual recognition. Neural Networks, IEEE Transactions on 9(4), 685–696 (1998).

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/ 4824-imagenet-classification-with-deep-convolutional-neural-networks. pdf.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale

visual recognition challenge," Int. J. Comput. Vision, vol. 115, no. 3, pp. 211–252, Dec. 2015. [Online]. Available: http://dx.doi.org/10.1007/s11263-015-0816-y.

[23] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in Computer Vision – ECCV 2014, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 818–833.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," CoRR, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[27] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," CoRR, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993.

[28] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," CoRR, vol. abs/1710.09829, 2017. [Online]. Available: http://arxiv.org/abs/1710.09829.

[29] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," CoRR, vol. abs/1709.01507, 2017. [Online]. Available: http://arxiv.org/abs/ 1709.01507.

[30] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[31] M. D. Buhmann, "Radial basis functions," Acta Numerica, vol. 9, p. 138, 2000.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in CVPR09, 2009.

[33] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in Proceedings of the 27th International Conference on International Conference on Machine Learning, ser. ICML'10. USA: Omnipress, 2010, pp. 807–814. [Online]. Available: http://dl.acm.org/citation.cfm?id=3104322.3104425

[34] M. Lin, Q. Chen, and S. Yan, “Network in network,” CoRR, vol. abs/1312.4400, 2013. [Online]. Available: http://arxiv.org/abs/1312.4400 [35]

[35] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, “Robust object recognition with cortex-like mechanisms,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 3, pp. 411–426, March 2007.

[36] Sharma, S., Sharma, S. and Athaiya, A., 2017. Activation functions in neural networks. towards data science, 6(12), pp.310-316.

[37] Wang, Y., Li, Y., Song, Y. and Rong, X., 2020. The influence of the activation function in a convolution neural network model of facial expression recognition. Applied Sciences, 10(5), p.1897.

[38] Bawa, V.S.; Kumar, V. Linearized sigmoidal activation: A novel activation function with tractable non-linear characteristics to boost representation capability. Expert Syst. Appl. 2019, 120, 346–356.

[39] Baheti, P. (2022) “Activation functions in neural networks [12 types & use cases],” V7labs.com. V7, 3 October. https://www.v7labs.com/blog/neural-networks-activation-functions. October 3, 2022.

[40] Lohani, H.K.; Dhanalakshmi, S.; Hemalatha, V. Performance Analysis of Extreme Learning Machine Variants with Varying Intermediate Nodes and Different Activation Functions. In Cognitive Informatics and Soft Computing; Springer: Singapore, 2019; pp. 613–623.

[41] Eckle, K.; Schmidt-Hieber, J. A comparison of deep networks with ReLu activation function and linear spline-type methods. Neural Netw. 2019, 110, 232–242.

[42] Maguolo, G.; Nanni, L.; Ghidoni, S. Ensemble of Convolutional Neural Networks Trained with Different Activation Functions. arXiv 2019, arXiv:1905.02473.

[43] Dubey, A.K.; Jain, V. Comparative Study of Convolution Neural Network’s ReLu and Leaky-ReLu Activation Functions. In Applications of Computing, Automation and Wireless Systems in Electrical Engineering; Springer: Singapore, 2019; pp. 873–880.

[44] Hagemann, H. (2020) “The 1918 flu pandemic was brutal, killing more than 50 million people worldwide,” NPR, 2 April. https://www.npr.org/2020/04/02/826358104/the-1918-flu-pandemic-was-brutal-killing-as-many-as-100-million-people-worldwide. September 15, 2022.

[45] 1918 pandemic (H1N1 virus) (2020) Cdc.gov. https://www.cdc.gov/flu/pandemic-resources/1918-pandemic-h1n1.html. September 15, 2022.

[46] WHO Coronavirus (COVID-19) dashboard (no date) Who.int. https://covid19.who.int/. September 15, 2022.

[47] CDC (2022) Symptoms of COVID-19, Centers for Disease Control and Prevention. https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html. September 15, 2022.

[48] How is the COVID-19 Virus Detected using Real Time RT-PCR? (2020) Iaea.org. https://www.iaea.org/newscenter/news/how-is-the-covid-19-virus-detected-using-real-time-rt-pcr. September 15, 2022.

[49] W. Wang, et al., Detection of SARS-CoV-2 in different types of clinical specimens, Jama 323 (18) (2020) 1843–1844.

[50] C.P. West, V.M. Montori, P. Sampathkumar, COVID-19 testing: the threat of falsenegative results, in: In Mayo Clinic Proceedings, vol. 95, Elsevier, 2020, pp. 1127–1129, 6.

[51] X. Xie, Z. Zhong, W. Zhao, C. Zheng, F. Wang, and J. Liu, "Chest CT for typical coronavirus disease 2019 (COVID-19) pneumonia: Relationship to negative RT-PCR testing," Radiology, vol. 296, no. 2, pp. E41-E45, Aug. 2020.

[52] A. Bernheim et al., "Chest CT findings in coronavirus disease-19 (COVID-19): Relationship to duration of infection," Radiology, vol. 295, no. 3, Jun. 2020, Art. no. 200463.

[53] M.-Y. Ng, et al., Imaging profile of the COVID-19 infection: radiologic findings and literature review, Radiology: Cardiothoracic Imag. 2 (1) (2020), e200034.

[54] Nayak, S.R.; Nayak, D.R.; Sinha, U.; Arora, V.; Pachori, R.B. Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: A comprehensive study. Biomed. Signal Process. Control 2021, 64, 102365.

[55] Zhang, C.; Eskandarian, A. A Survey and Tutorial of EEG-Based Brain Monitoring for Driver State Analysis. IEEE/CAA J. Autom. Sin. 2021, 8, 1222–1242.

[56] Rahhal, M.M.A.; Bazi, Y.; AlHichri, H.; Alajlan, N.; Melgani, F.; Yager, R.R. Deep Learning Approach for Active Classification of Electrocardiogram Signals. Inf. Sci. 2016, 345, 340–354.

[57] Kumar, A.; Kim, J.; Lyndon, D.; Fulham, M.; Feng, D. An Ensemble of Fine-Tuned Convolutional Neural Networks for Medical Image Classification. IEEE J. Biomed. Health Inform. 2017, 21, 31–40.

[58] Gopinath, K.; Sivaswamy, J. Segmentation of Retinal Cysts from Optical Coherence Tomography Volumes via Selective Enhancement. IEEE J. Biomed. Health Inform. 2019, 23, 273–282.

[59] Vásquez-Correa, J.C.; Arias-Vergara, T.; Orozco-Arroyave, J.R.; Eskofier, B.; Klucken, J.; Nöth, E. Multimodal Assessment of Parkinson's Disease: A Deep Learning Approach. IEEE J. Biomed. Health Inform. 2019, 23, 1618–1630.

[60] G.D. Rubin, et al., The Role of Chest Imaging in Patient Management during the COVID-19 Pandemic: a Multinational Consensus Statement from the Fleischner Society, Chest, 2020.

[61] A. I. Khan, J. L. Shah, and M. M. Bhat, "CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest X-ray images," Comput. Methods Programs Biomed., vol. 196, Nov. 2020, Art. no. 105581.

[62] A. A. Ardakani, A. R. Kana , U. R. Acharya, N. Khadem, and A. Mohammadi, "Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks," Comput. Biol. Med., vol. 121, Jun. 2020, Art. no. 103795.

[63] L.Wang, Z. Q. Lin, and A.Wong, "COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images," Sci. Rep., vol. 10, no. 1, 2020, Art. no. 19549.

[64] H. Gunraj, L. Wang, and A. Wong, "COVIDNet-CT: A tailored deep convolutional neural network design for detection ofCOVID-19 cases from chest CT images," Frontiers Med., vol. 7, Dec. 2020, Art. no. 608525.

[65] Chen J, Lu Y, Yu Q, Luo X, Adeli E, Wang Y, et al. Transunet: Transformers make strong encoders for medical image segmentation. 2021. arXiv preprint arXiv: 2102.04306.

[66] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in Proc. ICLR, 2021, pp. 1-22.

[67] Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M. (2021). Transformers in vision: A survey. arXiv preprint arXiv:2101.01169.

[68] Zhao, P., Zhang, J., Fang, W. & Deng, S. SCAU-Net: Spatial-channel attention U-net for gland segmentation. Front. Bioeng. Biotechnol. 8, 670. https:// doi. org/ 10. 3389/ fbioe. 2020. 00670 (2020).

[69] M. Ahsan, M. Based, J. Haider, M. Kowalski, COVID-19 detection from chest X-ray images using feature fusion and deep learning, Sensors 21 (4) (2021) 1480.

[70] Monshi, M.M.A., Poon, J., Chung, V. and Monshi, F.M., 2021. CovidXrayNet: Optimizing data augmentation and CNN hyperparameters for improved COVID-19 detection from CXR. Computers in biology and medicine, 133, p.104375.

[71] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014 arXiv preprint arXiv:1409.1556.

[72] M. Tan, Q.V. Le, "Efficientnet, Rethinking Model Scaling for Convolutional Neural Networks", 2019 arXiv preprint arXiv:1905.11946.

[73] L. Wang, Z.Q. Lin, A. Wong, Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images, Sci. Rep. 10 (1) (2020) 1–12.

[74] Mondal, A.K., Bhattacharjee, A., Singla, P. and Prathosh, A.P., 2021. xViTCOS: explainable vision transformer based COVID-19 screening using radiography. IEEE Journal of Translational Engineering in Health and Medicine, 10, pp.1-10.

[75] H. Chefer, S. Gur, and L. Wolf, "Transformer interpretability beyond attention visualization," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 782-791.

[76] H. Gunraj. (2021). COVIDx CT-2A: A Large-Scale Chest CT Dataset for COVID-19 Detection. [Online]. Available: https://www.kaggle.com/hgunraj/covidxct.

[77] A. Zhao. (2021). COVIDx CXR-2: Chest X-ray Images for the Detection of COVID-19. [Online]. Available: https://www.kaggle.com/andyczhao/covidx-cxr2.

[78] J. Irvin et al., "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in Proc. AAAI, 2019, pp. 590-597.

[79] Chetoui, M. and Akhloufi, M.A., 2022. Explainable Vision Transformers and Radiomics for COVID-19 Detection in Chest X-rays. Journal of Clinical Medicine, 11(11), p.3013.

[80] Society for Imaging Informatics in Medicine (SIIM). SIIM-FISABIO-RSNA COVID-19 Detection. Available online: https: //www.kaggle.com/c/siim-covid19-detection.

[81] Pan, I.; Cadrin-Chênevert, A.; Cheng, P.M. Tackling the radiological society of north america pneumonia detection challenge. Am. J. Roentgenol. 2019, 213, 568–574.

[82] Sivarama Krishnan, K. and Sivarama Krishnan, K., 2021. Vision Transformer based COVID-19 Detection using Chest X-rays. arXiv e-prints, pp.arXiv-2110.

[83] Alexander Buslaev et al. "Albumentations: Fast and Flexible Image Augmentations". In: Information 11.2 (2020). ISSN: 2078-2489. DOI: 10.3390/info11020125. URL: https://www.mdpi.com/2078-2489/11/2/125.

[84] Alexey Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. 2021. arXiv: 2010.11929 [cs.CV].

[85] Muhammad E. H. Chowdhury et al. "Can AI Help in Screening Viral and COVID-19 Pneumonia?" In: IEEE Access 8 (2020), pp. 132665–132676. DOI: 10.1109/ACCESS.2020.3010287.

[86] Tawsifur Rahman et al. "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images". In: Computers in Biology and Medicine 132 (2021), p. 104319. ISSN: 0010-4825. URL: https://www.sciencedirect.com/science/article/pii/S001048252100113X.

[87] Shome, D., Kar, T., Mohanty, S.N., Tiwari, P., Muhammad, K., AlTameem, A., Zhang, Y. and Saudagar, A.K.J., 2021. Covid-transformer: Interpretable covid-19 detection using vision transformer for healthcare. International Journal of Environmental Research and Public Health, 18(21), p.11086.

[88] El-Shafai,W.; Abd El-Samie, F. Extensive COVID-19 X-ray and CT Chest Images Dataset. Mendeley Data 2020, 3, 384.

[89] Sait, U.; Lal, K.G.; Prajapati, S.; Bhaumik, R.; Kumar, T.; Sanjana, S.; Bhalla, K. Curated Dataset for COVID-19 Posterior-Anterior Chest Radiography Images (X-rays). Mendeley Data 2020.

[90] Qi, X.; Brown, L.G.; Foran, D.J.; Nosher, J.; Hacihaliloglu, I. Chest X-ray image phase features for improved diagnosis of COVID-19 using convolutional neural network. Int. J. Comput. Assist. Radiol. Surg. 2021, 16, 197–206.

[91] Al Rahhal, M.M., Bazi, Y., Jomaa, R.M., AlShibli, A., Alajlan, N., Mekhalfi, M.L. and Melgani, F., 2022. Covid-19 detection in ct/x-ray imagery using vision transformers. Journal of Personalized Medicine, 12(2), p.310.

[92] Soares, E.; Angelov, P.; Biaso, S.; Froes, M.H.; Abe, D.K. SARS-CoV-2 CT-Scan Dataset: A Large Dataset of Real Patients CT Scans for SARS-CoV-2 Identification. medRxiv 2020.

[93] Mehboob, F., Rauf, A., Jiang, R., Saudagar, A.K.J., Malik, K.M., Khan, M.B., Hasnat, M.H.A., AlTameem, A. and AlKhathami, M., 2022. Towards robust diagnosis of COVID-19 using vision self-attention transformer. Scientific Reports, 12(1), pp.1-12.

[94] Xue, Y. (2020) HUST-19 for predicting COVID-19 clinical outcomes, Springer Nature. Available at: https://bioengineeringcommunity.nature.com/posts/hust-19-for-predicting-covid-19-clinical-outcomes.

[95] Jiang, J. and Lin, S., 2021. Covid-19 detection in chest x-ray images using swin-transformer and transformer in transformer. arXiv preprint arXiv:2110.08427.

[96] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, Baining Guo, Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, arXiv:2103.14030.

[97] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, Yunhe Wang, Transformer in Transformer, arXiv:2103.00112.

[98] Akhloufi, Moulay A. and Chetoui, Mohamed, Chest XR COVID-19 detection,https: //cxr-covid19.grand-challenge.org/,August,2021.

[99] Jiang, X., Zhu, Y., Cai, G., Zheng, B. and Yang, D., 2022. MXT: A New Variant of Pyramid Vision Transformer for Multi-label Chest X-ray Image Classification. Cognitive Computation, pp.1-16.

[100] Wang W, Xie E, Li X, Fan D-P, Song K, Liang D, et al. Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. 2021. arXiv preprint arXiv: 2102.12122.

[101] NIH. ChestX-ray14 dataset. 2017. https:// nihcc. app. box. com/v/Chest Xray-NIHCC.

[102] Kaggle. Catheter and Line Position Challenge. 2021. https://www.kaggle. com/c/ranzcr- clip-catheter-line-classification.

[103] Zhang M-L, Zhou Z-H, Engineering D. A review on multilabel learning algorithms. IEEE transactions on knowledge. 2013;26(8):1819–1837.

[104] Islam, M.N., Hasan, M., Hossain, M., Alam, M., Rabiul, G., Uddin, M.Z. and Soylu, A., 2022. Vision transformer and explainable transfer learning models for auto detection of kidney cyst, stone and tumor from CT-radiography. Scientific Reports, 12(1), pp.1-14.

[105] Islam, M. CT kidney dataset: Normal-cyst-tumor and stone 2021. [Online]. Available: https://www. kaggle.com/nazmu l0087/ctkidney-dataset- normal- cyst- tumor-and- stone.

[106] Guo, M.-H., Liu, Z.-N., Mu, T.-J. & Hu, S.-M. Beyond self-attention: External attention using two linear layers for visual tasks. arXiv preprintarXiv: 2105. 02358, (2021).

[107] Hassani, A., Walton, S., Shah, N., Abuduweili, A., Li, J. & Shi, H. Escaping the big data paradigm with compact transformers. arXiv preprintarXiv: 2104. 05704, (2021).

[108] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herv´e J´egou. Training data-efficient image transformers & distillation through attention. arXiv preprint arXiv:2012.12877, 2020.

[109] Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L. and Zhang, L., 2021. Cvt: Introducing convolutions to vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 22-31).

[110] Franc¸ois Chollet. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1251–1258, 2017.

[111] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[112] Yann Lecun, Patrick Haffner, L´eon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In Contour and Grouping in Computer Vision. Springer, 1999.

[113] M.E.H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z.B. Mahbub, K.R. Islam, M.S. Khan, A. Iqbal, N. Al-Emadi, M.B.I. Reaz, M. T. Islam, “Can

AI help in screening Viral and COVID-19 pneumonia?" IEEE Access, Vol. 8, 2020, pp. 132665 - 132676.

[114] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017.

[115] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[116] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[117] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In Indian Conference on Computer Vision, Graphics and Image Processing, Dec 2008.

[118] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. arXiv preprint arXiv:1912.11370, 6(2):8, 2019.

[119] Nico Engel, Vasileios Belagiannis, and Klaus Dietmayer. Point transformer. arXiv preprint arXiv:011.00931, 2020.