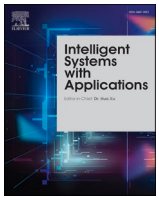
Intelligent Systems with Applications 16 (2022) 200147

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications



An effective approach to address processing time and computational complexity employing modified CCT for lung disease classification

Inam Ullah Khan [a], Sami Azam [b,*], Sidratul Montaha [a], Abdullah Al Mahmud [a], A.K.M. Rakibul Haque Rafid [a], Md. Zahid Hasan [a], Mirjam Jonkman [b]

[a] Health Informatics Research Laboratory (HIRL), Department of Computer Science and Engineering, Daffodil International University, Dhaka, 1341, Bangladesh
[b] College of Engineering, IT and Environment, Charles Darwin University, Casuarina, 0909, NT, Australia

ARTICLE INFO

Keywords:
COVID-19
Chest X-rays
Image Preprocessing
Modified compact convolutional transformer
Deep convolutional GAN, Hyper-parameter Tuning

ABSTRACT

Early identification and adequate treatment can help prevent lung disorders from becoming chronic, severe, and life-threatening. X-ray images are commonly used and an automated and effective method involving deep learning techniques can potentially contribute to quick and accurate diagnosis of lung disorders. However, in the study of medical imaging using deep learning, two obstacles limit interpretability. One is an insufficient and imbalanced number of training samples in most medical datasets. The other is excessive training time. Although training time can be reduced by decreasing the number of pixels in the images, training with low resolution images tends to result in poor performance. This study represents a solution to overcome these impediments by balancing the number of images and reducing overall processing time while preserving accuracy. The dataset used in this research contains an unequal number of images in the different classes. The quantity of data in the classes is balanced by creating synthetic images based on the patterns and characteristics of the original images, using a Deep Convolutional Generative Adversarial Network (DCGAN). Unwanted regions are removed from the X-ray images, the brightness and contrast of the images are enhanced, and the abnormalities are highlighted by using different artifact removal, noise reduction, and enhancement techniques. We propose a Modified Compact Convolutional Transformer (MCCT) model using 32×32 sized images for the categorization of lung disorders into four classes. An ablation study of eleven cases is employed to adjust several hyper parameters and layer topologies. This reduces training time while preserving accuracy. Six transfer learning models, VGG19, VGG16, ResNet152, ResNet50, ResNet50V2, and MobileNet are applied with the same image size the performance is compared with the proposed MCCT model. Our MCCT model records the greatest test accuracy of 95.37%, requiring a short training time, 10-12 s/epoch, whereas the other models only reach near-moderate performance with accuracies ranging from 43% to 79% and training times of 80-90 s/epoch. The robustness of the model with regards to the number of training samples is validated by training the model multiple times reducing the number of training images gradually from 49621 images to 6204 images. Results suggest that even with a smaller dataset, the performance is sustained. Our proposed approach may contribute to an effective CAD based diagnostic system by addressing the issues of insufficient and imbalanced numbers of medical images, excessive training times and low-resolution images.

1. Introduction

Deep learning-based methods, specifically deep convolutional neural networks (DCNNs), have led to noteworthy breakthroughs in medical image categorization and segmentation (Zhang et al., 2019). Due to the advances of deep learning in Computer Aided Diagnosis (CAD) systems, these are now widely used in studies of CAD systems for different medical imaging techniques. Although these approaches have the potential to be more reliable and accurate than traditional feature-based methods, the drawbacks of deep learning models include the requirement of large numbers of training images with the associated long training time and complexity (Mamalakis et al., 2021). Introducing transfer learning could address the concern of requiring large datasets, however other concerns remain, including extensive computational

* Corresponding author.
E-mail address: sami.azam@cdu.edu.au (S. Azam).

https://doi.org/10.1016/j.iswa.2022.200147
Received 30 June 2022; Received in revised form 12 October 2022; Accepted 3 November 2022
Available online 8 November 2022

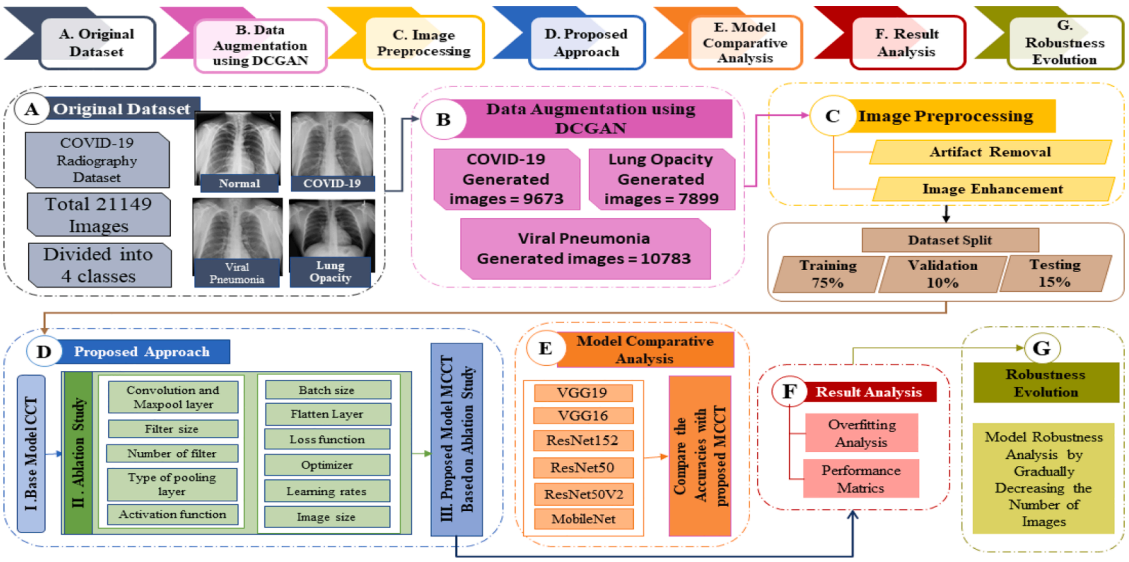


Fig. 1. The process to classify Chest X-ray (CXR) images into four classes using MCCT. Each phase represented by a block.

requirements and training times, generalization capability, performance consistency and robustness of the model (Alhasan and Hasaneen, 2021; Hussain et al., 2021). Lung disease is regarded a major health challenge worldwide and is one of the most common causes of death. COVID-19 has become a major threat to global health and the pandemic has severely affected healthcare systems, the global economy, education, workplaces, tourism etc. (Rahman et al., 2021; Vocaturo et al., 2021). Even though the first outbreak of the virus occurred years ago, it is still a serious threat because the number of deaths and new cases is rising every day (Ayris et al., 2022). An accurate prediction of risk factors can help to prevent lung diseases from becoming chronic, severe, and life-threatening conditions. In this regard, timely diagnosis and proper treatment planning can prevent spread of infections and deterioration of lung diseases, thus reducing the mortality rate. Research focusing on the detection of lung diseases, including COVID-19, from chest X-rays indicates that X-ray images contain meaningful information regarding the progression of the disease (Borghesi et al., 2020; Cozzi et al., 2020). An automated and reliable approach based on deep learning techniques using X-ray images could be a promising solution in the diagnosis of lung diseases. CNN models trained on standard chest radiography images to detect and categorize lung disorders require extensive computational resources and time (Zumpano et al., 2021; Sarv Ahrabi et al., 2021). These impediments are compounded by the problem of small medical datasets with an imbalanced number of images in the different classes. However, successful classification models addressing the computational complexity issue can be built without convolutions. In this regard, transformers have become a key focus of Machine Learning (ML) research. The most notable work in this area is Vision Transformer (ViT), which implements a pure self-attention-based model on sequences of image patches and achieves competitive performance when compared to CNNs. In terms of computational efficiency and accuracy, ViT models outperform CNNs by almost a factor of four and achieve better accuracies on large datasets with less training time (Paul & Chen, 2022). Because self-attention layers are faster than recurrent layers (Vaswani et al., 2017), transformers are also substantially more efficient than Recurrent Neural Networks (RNNs) if we take into consideration the computational complexity of the "Sequential operations". ViTs can address the issue of training time, but due to the architecture of the transformer models, they are data-hungry and require massive amounts of data to provide satisfactory results. In medical research, collecting a large amount of annotated image data is often challenging, time consuming and costly. To resolve this, Hassani et al. (2021) introduced Compact Convolutional Transformer (CCT) by adding simple convolutional blocks to the tokenization step of the vision transformer. This resulted in reducing training time with noteworthy performance improvements. In this study, the COVID-19 Radiography dataset is used for the automatic detection and classification of lung diseases into COVID-19, Normal, Lung Opacity, and Viral Pneumonia. In this context, the issues of training time and complexity, small medical datasets, an imbalanced number of images and low resolution images are addressed with noteworthy performance. The main contributions can be summarized as follows:

i The dataset used for the experiment, contains imbalanced numbers of images in the different classes which might lead to poor performance of the model. Therefore, the dataset is balanced by producing synthetic images using Generative Adversarial Network (GAN) for the classes that contain fewer images.

ii A number of image preprocessing techniques, Morphological Opening, Gamma Correction, CLAHE, Bilateral, and Spectrum, are applied to remove artifacts and enhance the quality of the images.

iii Several statistical analysis PSNR, SSIM, MSE, RMSE measures are employed to ensure that the image processing techniques do not degrade image quality.

iv To address the issues of high training time and low numbers of images, we propose a model named MCCT by modifying an original CCT model for the automatic classification of lung diseases. The tokenization step of the vision transformer is performed using convolutional blocks, reducing model training time significantly while achieving good accuracy even with low resolution images.

v An ablation study is performed by changing different hyperparameters and the layer architecture of the proposed model to further improve the performance and reduce the number of parameters and the time complexity.

vi Several transfer learning models, including VGG19, VGG16, ResNet152, ResNet50, ResNet50V2, MobileNet, are applied to our dataset to compare the performance of the proposed MCCT model in terms of accuracy and training time with images of pixel size 32×32.

vii To evaluate the generalization capability and sustainability of our model further with regards to the volume of the training dataset, the model is trained four times, gradually decreasing the number of images. Results suggests that, even for a lower number of

Table 1
Dataset Description

Name	Description
Total Number of Images	21149
Dimension	299 × 299
Images type	X-ray
Covid-19	3616
Normal	10192
Lung Opacity	6012
Viral Pneumonia	1345

images, the model yields satisfactory performance validating the robustness of MCCT model.

viii Several performances metrics, such as accuracy, precision, sensitivity, recall, F1-score, and MCC are evaluated to compare the performance of transfer learning models with the proposed MCCT model. It is found that our proposed MCCT model outperforms the transfer learning models in terms of accuracy and training time while using 32 × 32 sized images as training data.

The MCCT model records the highest test accuracy of 95.37% requiring a training time of 10-12 sec/epoch while VGG19, VGG16, ResNet152, ResNet50, ResNet50V2, and MobileNet yield test accuracies of 79.51%, 76.97%, 53.39%, 67.77%, 65.35% and 43.42% respectively requiring a training time of 10-12 sec/epoch on average. Moreover, while decreasing the number of images gradually from 49621 to 6204, the performance is sustained with accuracies in a range of 91%-95%. This study may help to address the issues of computational complexity, training time, data imbalance, and data inadequacy.

The remainder of this paper is organized as followsSection 2 describes the details of the proposed methodology. Section 3 describes the dataset. Section 4 presents details of GAN and its architecture. Section 5 provides an overview of image preprocessing techniques. The models and experimental setup are summarized in Section 6. Section 7 discusses the ablation study and results. Section 8 gives a brief overview and comparison with related work. Section 9 concludes the paper.

2. Methodology

The study is conducted by introducing deep learning approaches to classify Chest X-ray (CXR) images into four classes. Fig. 1 illustrates the process.

The Covid19 Radiography Chest X-ray dataset is used for all the experiments in this study. The Data augmentation technique DCGAN is introduced to deal with the data imbalance problem by generating new images. Afterwards, several image preprocessing algorithms are applied on the augmented balanced dataset to remove artifacts and enhance the images. Statistical analysis methods, Peak signal-to-noise ratio (PSNR), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the structural similarity index measure (SSIM) are used to assess and ensure that image quality is not reduced. The preprocessed dataset is split into training, validation and testing before feeding the images into the deep learning models. We propose a MCCT model using an image size of 32 × 32 by modifying the layer architecture and hyper parameters of an original CCT model. An ablation study of eleven cases is done to ensure the best performance while addressing the time complexity. The performance of the MCCT model is compared with five deep learning models, VGG16, VGG19, ResNet152, ResNet50, ResNet50V2 and MobileNet, in terms of training time and accuracy for an image size of 32 × 32, Several performance metrics are evaluated and the possible occurrence of overfitting is assessed. Further evaluation of the model's robustness is conducted by testing its performance with decreasing numbers of images. All processes are described briefly in the sections and sub-sections below.

3. Dataset

In our research, we evaluated the suggested model on a publicly available COVID-19 Radiography Dataset, obtained from "Kaggle" ("COVID-19 Radiography Database") comprising a total of 21149 Chest X-ray (CXR) images. The dataset contains four classes. The COVID-19 class has 3616 images, the Lung Opacity class has 6012 images, the Normal class has 10192 images and the Viral Pneumonia class has 1345 images. All images are 299 × 299 pixels in grayscale format. A summary of the dataset is given in Table 1.

Fig. 2 depicts an example of each of the four classes of this dataset.

4. Data augmentation using DCGAN

In computer vision, the performance of a neural network greatly depends on the availability of a sufficient number of labeled data. This is one of the biggest challenges in medical imaging. To overcome a data shortage, the training dataset is often expanded artificially by simple image transformations and color adjustment methods, such as scaling, flipping, converting, enhancing contrast or brightness, blurring and sharpening, white balance, and so on, (Krizhevsky et al., 2017). However, these augmentations are designed to turn an existing sample into a slightly altered sample. The modifications are limited and do not create a completely plausible alternative to unseen data (Motamed et al., 2021). A new, advanced augmentation approach that overcomes the limitations of traditional data augmentation methods is synthetic data augmentation.

4.1. Deep convolutional GAN

GAN is an effective deep learning based generative model that generates synthetic images, without supervision, using a min-max scheme. Synthetic data obtained using a generative model have more variability and enrich the dataset to improve the system training process. GANs capture the training data distribution and create new examples based on the same distribution. This leads to an improved generalization ability of CNN models and consequently prevents overfitting (Bowles et al., 2018).

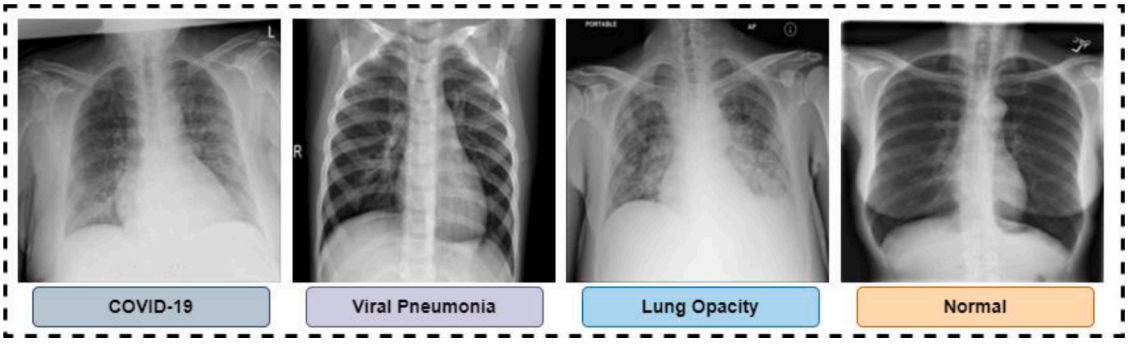


Fig. 2. Images from each class of the dataset

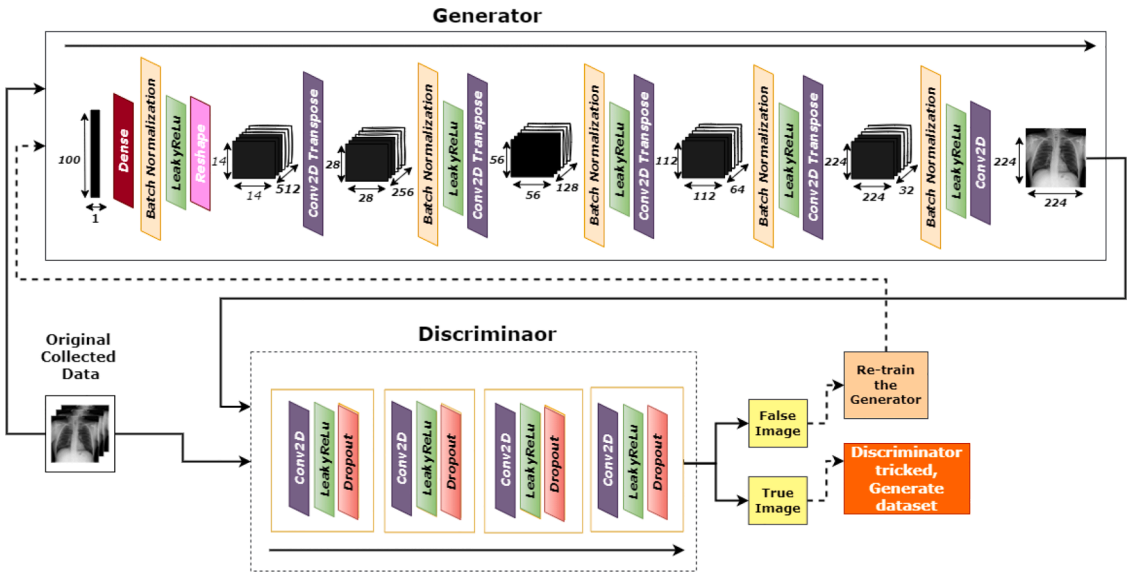


Fig. 3. Architecture of DCGAN

GANs combine two neural networks, named generator and discriminator, which create new data instances by minimizing the probability distribution distance between the original and generated data. The task of the generator is to generate new fake (artificial) data instances that look like the original training data. The discriminator network then distinguishes the fake (artificially generated) and real data. If the discriminator can recognize the fake data, it sends the data back to the generator and the generator upgrades the fake data, sending it again to the discriminator to recognize. Before applying DCGAN, all the images from dataset have been resized to 224 × 224. During training, the generator network improves its ability to generate artificial samples by minimizing the loss function. The discriminator, on the other hand, learns to become better at discriminating between original and fake samples by maximizing a similar loss function. Some limitations of a basic GAN are supervised learning, inability to detect overfitting, instability when used in small datasets (Jin et al., 2020). For this reason, we use DCGAN which combines GAN with deep CNN while ensuring a stable architecture through modification (Salehinejad et al., 2018). The architecture and function of DCGAN is similar to the original GAN, except that both the discriminator and the generator networks employ convolutional and convolutional-transpose layers. The following equation is used to train the generator and discriminator networks (Kora Venu and Ravula, 2020).

$$\underset{N}{min}\underset{M}{max}V_{GAN}(M,N) = \mathbb{E}_{x\sim Pdata^{(x)}}[\log M(x)] + \mathbb{E}_{z\sim Pz^{(z)}}[\log(1-M(N(z)))] \quad (1)$$

where M is the Discriminator and N is the Generator, $\mathbb{E}_{x\sim Pdata^{(x)}}$ and $\mathbb{E}_{z\sim Pz^{(z)}}$ are the expected values of overall real and fake instances, $N(z)$ is the generator function that maps to the data space. x denotes original data and $M(x)$ is the probability that x came from the original data distribution rather than from the generated data distribution. $P_z(z)$ is the random noise variable sampled from a standard normal distribution, Fig. 3 represents the detailed architecture of the generator network used in this study.

Initially, the generator takes a random 100 × 1 noise vector as an input, which is fed into the dense layer and reshaped to 14 × 14 × 512. We use four convolution2D transpose and one conv2D layer in this architecture to up sample an image size representation from 14 × 14 × 512 to a size of 224 × 224 × 3.

Data of size 14 × 14 × 512 pass through the first Convolutional2D Transpose and are reshaped into the image size of 28 × 28 × 256. In the second, third and fourth layer, the architecture is same. The output from the first Conv2D transpose layer forwards through the batch normalization layer, the activation function LeakyReLu and Conv2D transpose and is reshaped to respectively 56 × 56 × 128, 112 × 112 × 64 and 224 × 224 × 32. In the final layer, using the conv2D layer, we obtain an output with an image size of 224 × 224 × 3. Batch normalization (Ioffe and Szegedy, 2015) is used to stabilize the learning process and the input is normalized to have a zero mean and unit variance.

The discriminator takes the generated images of the generator network and the real images of the source dataset as input. This input then goes through a combination of convolution layers of four blocks. Each convolution block of the discriminator network contains Conv2D, LeakyReLu as activation function and a dropout layer. After passing through four blocks the discriminator recognizes the image as real or fake. The discriminator works as a binary classifier that predicts real of fake images. Therefore, binary cross-entropy is employed as the loss function, as stated in Equation 2 (Kora Venu and Ravula, 2020):

$$J_{BCE^{(\theta)}} = -\frac{1}{N}\sum_{n=1}^{N}[y_n \times \log(h_\theta(x_n)) + (1-y_n) \times \log(1-h_\theta(x_n))] \quad (2)$$

Here, N is the number of training samples, y_n is the target label for training sample n (the label for an original image is 1 and for a fake image is 0), x_n is the input for training sample n, and h_θ is the model with neural network weights θ.

If the generated image is very similar to a real image, the discriminator gets tricked into thinking it is a real image and identifies the fake image as real. If on the other hand, the generator produces a fake image that does not resemble the original image, the discriminator identifies it as fake data and gradients are acquired which update the weights of the generator through backpropagation. The generator with updated weights produces better fake images and keeps trying to trick the discriminator into identifying fake images as real. Through these cycles of generating and discriminating, a robust generator can be obtained which is capable of producing fake images which closely resemble real images and can be used to increase the number of images of a particular dataset.

4.2. Training strategy and augmented dataset generation

As stated before, our dataset contains four classes having an imbalanced number of samples for the different classes. The highest number of images (10192) is found in the Normal class. We have balanced other three classes by creating image numbers close to the Normal class. For training DCGAN, the resized (224 × 224) dataset is used. As the Normal

Table 2
Number of Original and Generated Data using DCGAN

Class	Original Images	DCGAN Training Images	DCGAN Generated Images	Total Images
COVID-19	3616	1800	9673	13289
Lung Opacity	6012	1872	7899	13911
Normal	10192	-	-	10192
Viral Pneumonia	1345	1345	10783	12128
	TOTAL= 21165		TOTAL=28355	TOTAL=49520

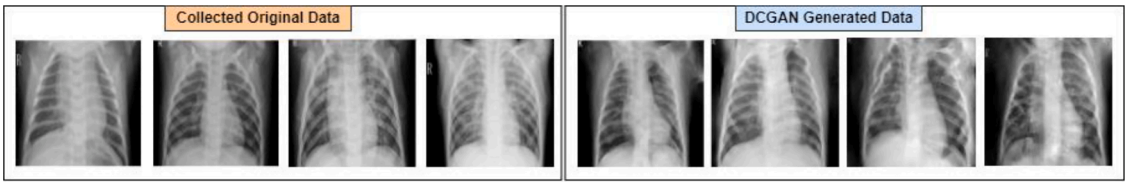


Fig. 4. Original Image and DCGAN Generated Image

class is considered as threshold, DCGAN is applied to the remaining three classes. The model is trained using optimizer Adam, with learning rate: 0.0008, batch size 128 and the loss function 'binary cross-entropy'. We base the epoch number on number of images in the original dataset using the training time and output. Therefore, for the class Lung Opacity, the model is trained for 200 epochs as the original number of images (6012) was sufficient to generate enough transformed images within these epochs. However, for the classes COVID and Viral Pneumonia, the number of images was not sufficient to generate the required number of transformed samples in 200 epochs. For these classes, the number of epochs was set to 250. After augmentation the dataset is enlarged from 21165 images to 49520 images. However, we did not create an equal number of images for all classes, as the interpretation capability of our classification model might not be evaluated effectively using such a completely balanced dataset. Table 2 shows the number of images in the original dataset, the number of generated datasets using DCGAN, and the total number of images after augmentation.

Fig. 4 depicts original images and DCGAN generated images. It can be seen that generated images are very similar to the real images.

5. Image preprocessing techniques

Image preprocessing, before feeding the images into a neural network, is one of the most important steps to ensure that the model's performance and computation time are both optimized. Image preprocessing in this study includes artifact removal and image enhancement through several commonly used algorithms. The chest X-ray images of this dataset have several artifacts, noise and low contrast. First artifacts are removed from the images by applying morphological opening (Breuel, 2007). Subsequently, gamma correction (Dhar et al., 2021) and CLAHE (Hassan et al., 2021) are applied to improve the brightness and contrast of the images. Bilateral (Tomasi and Manduchi, 2002) filter is employed to smooth the pixels while preserving the edges of ROIs. Finally, a filter for ImageJ software named 'Spectrum' (Beeravolu et al., 2021) is applied to highlight the abnormality. Statistical evaluation with PSNR, MSE, RMSE, and SSIM is done to ensure that the image quality is not reduced due to these processing algorithms.

5.1. Artifact removal

As artifacts can affect the performance of the model, artifact removal is an important step in image preprocessing. This is done by morphological opening.

5.1.1. Morphological opening

To apply morphological opening, the image is first converted to binary format using binary thresholding (Breuel, 2007). After converting to binary format, small noises become more visible. Morphological opening is applied on the binary image using a kernel. This kernel's shape and size are determined based on the characteristics of the artifacts to be erased. A structural element is a matrix that identifies and defines each pixel and its neighborhood. After experimenting with several kernel shapes and sizes, a rectangular kernel of size 5×5 is applied as for this kernel the artifacts are removed successfully while preserving the necessary information. Thus, a noise-free binary mask is achieved which is later merged with the original image using a bitwise AND function.

5.2. Image enhancement

Chest X-ray details are often hard to interpret due to their complex characteristics and hidden information, which makes it challenging for a model to distinguish the classes. To achieve optimal performance, suitable image enhancement techniques may aid in improving the visual distinction of the Regions of Interest (ROIs) from the background.

5.2.1. Gamma correction

Using a nonlinear transformation, gamma correction modifies the overall brightness and contrast of an image (Dhar et al., 2021). In this study, gamma correction is used to improve the distribution of light and dark areas with the aim to highlight the ROI against a dark background. The algorithm is applied using the following equation:

$$O = I \wedge (1 / G) \tag{3}$$

where I is the input image, G is the gamma value and O the output image.

The correction of brightness and contrast depends on the gamma value G where $G < 1$ causes the pixels appear darker and $G > 1$ makes them appear lighter. A suitable gamma value is determined after experimenting with several gamma values for our dataset. A gamma value of 1.2 is found to result in an optimally enhanced image.

5.2.2. CLAHE

CLAHE is applied to balance the overall contrast by correcting over-amplification of contrast levels. Rather than working with the entire image, the algorithm divides the image into small regions called tiles and operates on the individual tiles (Hassan et al., 2021). To apply CLAHE, two parameters are used, cliplimit and tilegridsize, where cliplimit is the threshold contrast value to be applied and tilegridsize is the size of tile in each row and column. These parameter values are determined after a few experiments with different values on our dataset, resulting in a cliplimit of 40 and a tilegridsize of 8×8.

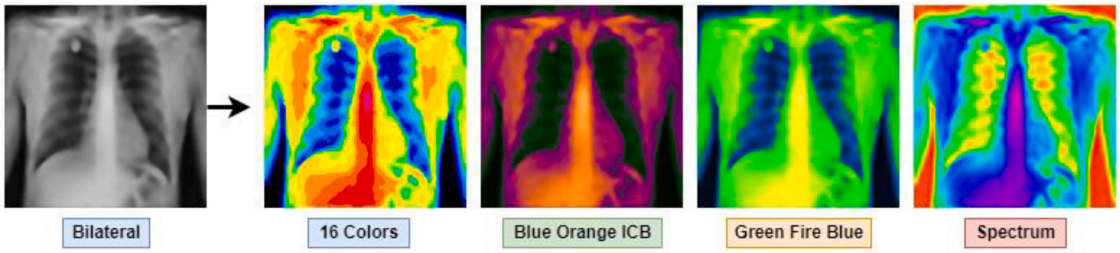


Fig. 5. Output of various ImageJ filters

Table 3
Selected parameter values for all pre-processing algorithms

Process	Algorithm	Parameter Value
Artifact Removal	Morphological opening	Structuring element = rectangular Kernel Size =5 × 5
Image enhancement	Gamma correction	Value =1.2
	CLAHE	ClipLimit=1.5, TileGridSize=8 × 8
	bilateral Filter	diameter =9, sigmaColor=75, sigmaSpace=75

5.2.3. Bilateral

The bilateral filter is a way to smooth the pixels of an image while preserving the edges. A weighted average of nearby pixel values is used in Gaussian smoothing (Tomasi and Manduchi, 2002). The filter applies a tonal weight to pixel values that are closer to the pixel value in the center, weighting them more heavily than pixel values that are more dissimilar. Because of this tonal weighting, the bilateral filter can preserve edges while smoothing flatter sections. To apply the algorithm, the parameters diameter, sigmaColor, and sigmaSpace are used. Diameter is the pixel size of each neighborhood, sigmaColor is the color space value of sigma. As the value increases, nearby colors start to blend with each other. Sigma's coordinate space value is sigmaSpace.

5.2.4. Spectrum

ImageJ's "Look Up Tables (LUTs)" software tool is used to complete the final enhancement (Montaha et al., 2021). A multi-color separation is achieved using this filter, which is applied to the image to show the affected area and surrounding cells separately. LUT has a number of filters. To determine the best filter for our dataset, we experimented with several LUT filters named '16 Colors', 'Blue Orange ICB', 'Green Fire Blue' and 'Spectrum'. The LUT filter that worked best for our dataset was 'Spectrum'. Fig. 5 represents the state of the data after applying each LUTs filter.

The best parameter values associated with the image processing methods are chosen after running multiple tests on the dataset. Table 3 represents the parameter values for all the applied image processing techniques.

Fig. 6 represents the entire image pre-processing process from artifacts removal to image enhancement.

5.3. Verification

Finally, statistical analysis is done to show that the image quality does not deteriorate due to the algorithms (Montaha et al., 2022). The equations for these verification methods are given below.

MSE is probably the simplest and most prevalent loss function. To determine the MSE, difference between the predictions made by the model and the actual data is squared and then averaged throughout the entire dataset. MSE is defined mathematically by the following equation:

$$MSE = \frac{1}{pq}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(O(m,n), -P(m,n))^2 \tag{4}$$

where O is the original image, P is the processed image, p and q indicate the pixels of O and P, and m, n indicate the rows of the pixels p, q. The MSE value ranges from 0 to 1, with a value close to 0 indicating good image quality. If the value is greater than 0.5, the quality has deteriorated. A value of 0 indicates that the image is completely free of noise.

PSNR calculates the signal-to-noise ratio between two pictures. This ratio is used as a measure of image quality between the original and the compressed version. The greater the PSNR, the higher the image quality. The mathematical expression of PSNR is:

$$PSNR = 20\log_{10}\left(\frac{(MAX)}{\sqrt{MSE}}\right) \tag{5}$$

Here, MAX denotes the image's maximum pixel value (i.e., 299). A good PSNR value for an 8-bit image is usually between 30 and 50 dB.

SSIM is a metric which measures the image quality loss caused by image processing. It needs two images: a reference image and a processed image with the same image origin. The equation for SSIM:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{6}$$

Here, μx and μy are the Gaussian window averages of the two pictures (x, y). The variance is denoted σ_x^2 *and* σ_y^2, while the covariance of the pictures is denoted by σ_{xy}. C1 and C2 are the two variables used to stabilize the division, where c1 is $(0.01 \times 255)^2$ and c2 = $(0.03 \times 255)^2$, with default values of 0.01 and 0.03. The SSIM ranges from 0 to 1, with 1 denoting 'perfect structural similarity' and 0 denoting 'no structural similarity'.

RMSE is a commonly used metric for comparing values predicted by

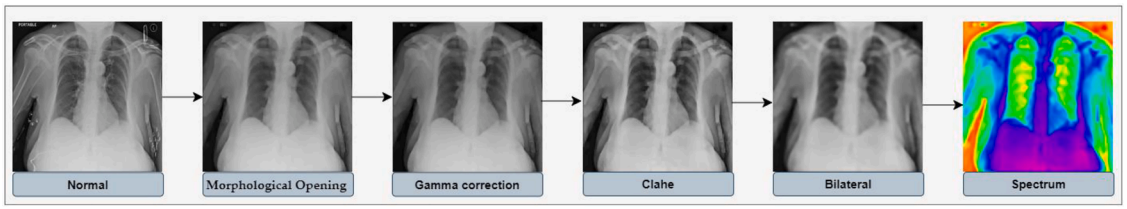


Fig. 6. Flowchart of Image Preprocessing and Its Results

Table 4
MSE, PSNR, SSIM, and RMSE values for ten images

	PSNR	SSIM	MSE	RMSE
Image_1	31.68	0.9930	0.44	0.66
Image_2	32.19	0.9954	0.39	0.62
Image_3	31.98	0.9936	0.41	0.64
Image_4	32.86	0.9916	0.33	0.57
Image_5	31.56	0.9924	0.45	0.67
Image_6	32.84	0.9955	0.33	0.57
Image_7	32.24	0.9941	0.38	0.61
Image_8	33.22	0.9950	0.30	0.54
Image_9	32.09	0.9944	0.40	0.63
Image_10	32.01	0.9939	0.40	0.63

a model or estimate to values actually observed. It represents the Euclidean distance between the measured true values and the predictions. RMSE can be expressed as:

$$RMSE = \left[\sum_{j=1}^{N} \left(m_{fi} - m_d\right)^2 \Big/ N\right]^{\frac{1}{2}} \tag{7}$$

Where $\sum$ indicates summation, $(m_{fi} - m_d)^2$ is the square of differences, and N is the dataset size. A lower RMSE, especially near 0, suggests fewer errors and better image quality. The computed PSNR, MSE, SSIM, and RMSE value of ten random images are presented in Table 4.

It can be observed from Table 4 that the PSNR values of the images are larger than 31, the SSIM values are larger than 0.99, the MSE values are larger than 0.33 and the RMSE values are larger than 0.54 which indicates a good quality of the preprocessed images. The values of PSNR, SSIM, MSE and RMSE for the rest of the images in the dataset are close to this range which validates the effectiveness of our image preprocessing algorithms.

6. Proposed model

Vision transformers (Vit) can simultaneously process numerous sequential data and can detect long-range relationships between sequential pieces using their self-attention mechanism. This makes them exceptionally robust in image classification tasks (Huang et al., 2022; Islam, 2022; Khan et al., 2022). Nevertheless, most real-world medical datasets are insufficient to train ViTs for satisfactory performance. CCT, a hybrid compact ViT with convolution, solves this problem. CCT models use CNN blocks as patching blocks with a local receptive field that maintains local image information. The self-attention mechanism collects relationships between image patch pieces and combines pertinent information.

6.1. Compact convolutional transformer (CCT)

There are two main blocks in the CCT architectures, Convolutional Tokenization and Transformer with sequential pooling. Fig. 7 shows the detailed mechanism of CCT.

The Convolutional Tokenization block is used to generate patches for input images (Cubuk et al., 2018). The dimension of the augmented images are H × W × C where H is the height, W is the width, and C is the number of channels. These images are divided into patches and turned into a sequence of length m. For a given image x with dimensions H × W × C the operations of Convolutional Tokenization will be:

$$x_0 = \mathbf{MaxPool}(\mathbf{ReLU}(\mathbf{Conv2D}(\mathbf{x}))) \tag{8}$$

where, the convolutional layer (Conv2d) has 64 filters with strides 2 equipped with the ReLU activation function. The maxpool layer then downscales the resultant feature maps of Conv2D. The convolutional tokenization block can take input images of any size. As a result, CCT models do not require all image patches to be of equal sizes. The CNN layers help the model to retain local spatial information because of these convolutional patches.

Afterwards, the resultant image patches from the first block go to the transformer-based backbone where a Multihead self-attention (MSA) layer and a Multilayer perceptron (MLP) head make up the encoder block. Layer normalization (LN), GELU activation, and dropout are used by the transformer encoder. Layer normalization is applied after positional embedding in CCT models where the positional embeddings are learnable.

The resultant output of the transformer backbone is pooled through the sequence pooling layer where sequence pooling is used as an alternative to applying a class token to map sequential outputs to a single class [10]. This sequence pooling enables the network to weigh latent spaces' sequential embeddings created by the transformer encoder and improve data correlation for the input data. The entire sequence of data is pooled by the sequence pooling layer as it comprises relevant information from various portion of the input images. This method can be called Mapping transformation and is denotes as $T: \mathbb{R}^{(b\times n\times d)} \rightarrow \mathbb{R}^{(b\times d)}$

The operation can be described as:

$$x_L = f(x_0) \in \mathbb{R}^{(b\times n\times d)} \tag{9}$$

where L is a layer transformer encoder and its output is x_L or $f(x_0)$. Furthermore, a mini-batch size denoted by b, d is considered as the embedding dimension and n denoted the sequence length. Afterwards, x_L is fed to a linear layer $g(x_L) \in \mathbb{R}^{(d\times 1)}$ and the softmax activation function (Eq. 10) is applied.

$$x_L^{'} = soft\max\left(g(x_L)^T\right) \in \mathbb{R}^{(b\times 1\times n)} \tag{10}$$

The output can be computed as:

$$z = x_L^{'} x_L = soft\max\left(g(x_L)^T\right) \times x_L \in \mathbb{R}^{(b\times 1\times d)} \tag{11}$$

After pooling of the second dimension, $z \in \mathbb{R}^{(b\times d)}$ is achieved as an output. This then goes through a linear classification layer and the images are classified.

6.2. Base model architecture

In this study a modified version of a CCT model (MCCT) is proposed, which is achieved by conducting ablation studies on a base CCT model. Fig. 8 shows the Base Model architecture of CCT.

The base CCT architecture comprises of multiple modules and layers

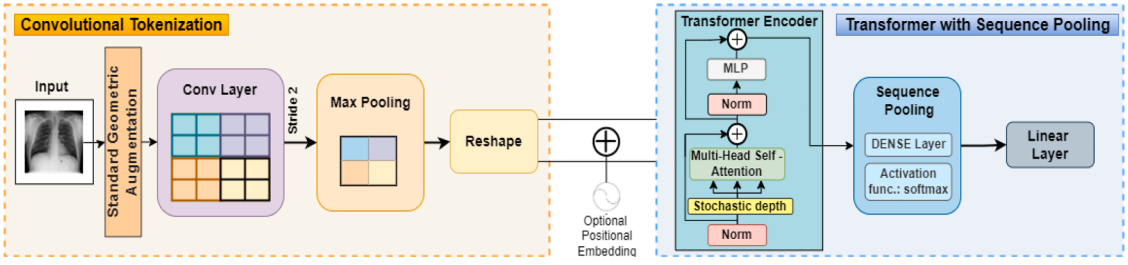


Fig. 7. Structure of CCT

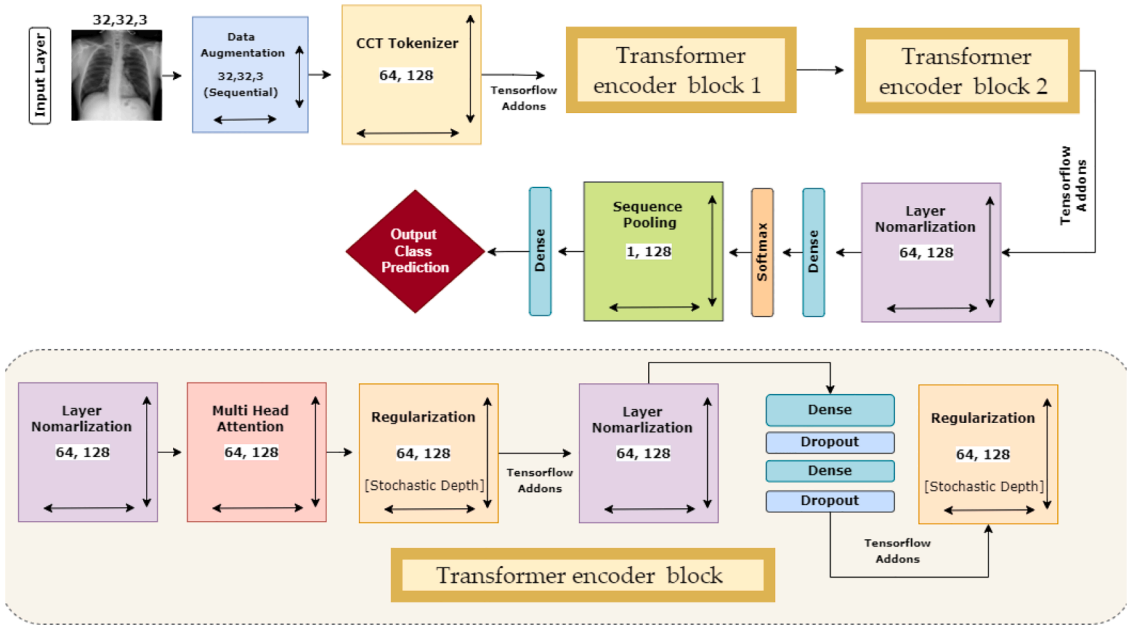


Fig. 8. Base Model architecture of CCT

including the input layer, the data augmentation layer, the CCT Tokenizer, multi-head attention layers, a regularization (stochastic depth) layer, pooling layers, dropout layers, dense layers and output dense layers equipped with the softmax activation function. The model takes images of dimensions $32 \times 32 \times 3$ as input and the data augmentation layer performs various geometric augmentations on the input images. The augmented images are fed to the CCT Tokenizer block and the output image data is reshaped into dimensions 64×128. Initially, the convolutional layer of CCT Tokenizer block contains strides of size 2 and kernels of size 4 and a pooling layer kernel size of 4. After tokenization, the data passes through tensorflow addons and then to the transformer encoder block. This block comprises of several layers in a specific sequence: layer normalization (1), multi-head attention, regularization, layer normalization (2), followed by two pairs of dense and dropout layers with a dropout factor of 0.1. Another regularization layer is attached at the end of the transformer encoder block. The output of this layer is of dimension 64×128 and is regularized once again with the Regularization layer, followed by another transformer encode block, identical to the first one. The output of the second transformer encoder block goes through a regularization layer and a normalization layer. The normalized output then passes through a dense layer and a softmax layer that produces output data of dimensions 64×1. This is forwarded to a sequence pooling layer which results in output data with a dimension of 1×128. Finally, a linear classification layer classifies the chest X-ray images into four classes.

Furthermore, as a loss function, Categorical Crossentropy is selected and the Adam optimizer is used with a learning rate of 0.001. The model is run for 100 epochs with a batch size of 128.

6.3. Ablation study

As stated, we have performed an ablation study on the base CCT model by altering the layer architecture and tuning hyper parameters in order to achieve the best possible performance. Eleven ablation studies were conducted, including adding or decreasing the number of transformer encoder blocks, changing activation functions and pooling layer

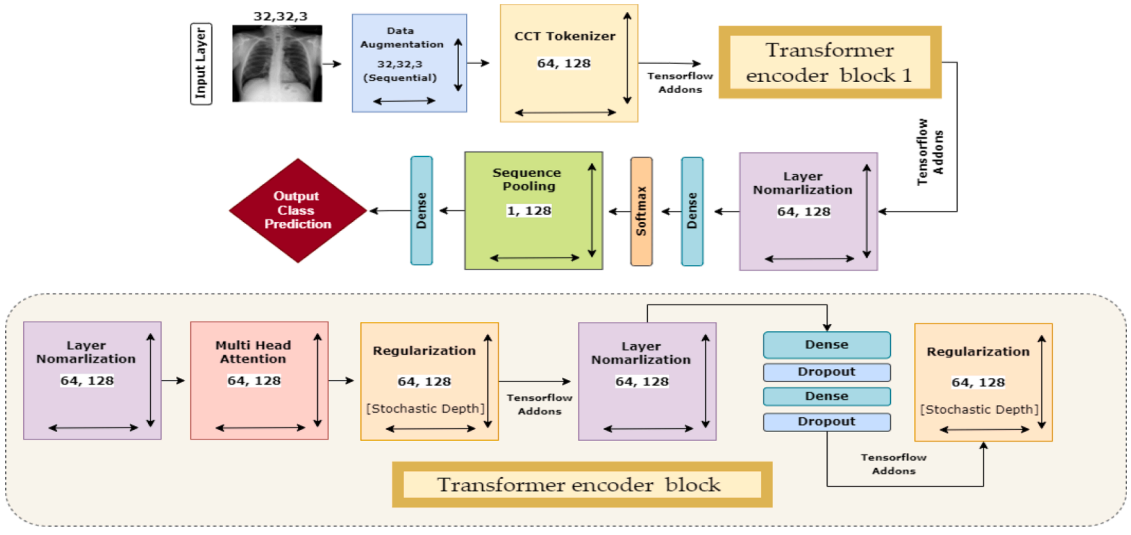


Fig. 9. Proposed Model MCCT Architecture

types, and experimenting with stride sizes, kernel sizes, pooling layer kernel sizes, loss functions, batch sizes, optimizers, and learning rates. After completion of all ablation studies, the proposed MCCT model has a more robust design with improved classification accuracy and reduced processing time. The results of the ablation study can be found in section: 7.2.

6.4. Proposed MCCT architecture

To minimize time complexity and training times and optimize the performance, the proposed MCCT architecture is made shorter and more robust. The resultant MCCT architecture after ablation studies has a close resemblance to the base CCT model with fewer transformer encoder blocks. The base CCT architecture contains two transformer encoder blocks whereas the MCCT model contains just one transformer encoder block making the model smaller and offering faster training times. The rest of the architecture is kept the same with some changes in model hyper parameters, including the stride size and kernel size (Fig. 9).

The model does not require positional encoding, unlike transformer-based models, which helps in maintaining a low computational complexity. Self-attention has a computational complexity of $O(n^2.d)$ where the input sequence length is n and the dimensionality of the vector representation is denoted as d. With introduction of positional encoding, the computational complexity increases $(O(n^2.d + n.d^2))$(Vaswani et al., 2017). As positional encoding is not necessary in the MCCT model and the transformer backbone is purely based on the self-attention mechanism, the training and testing phase of the proposed model requires fewer resources and is faster. This further increases the effectiveness of the model.

6.5. Training strategy

To train the models, the batch size was set to 128 and the maximum number of epochs was set to 100. The Adam optimizer was utilized with a learning rate of 0.001. In multiclass cases, the default loss function is 'categorical cross-entropy' (Lorencin et al., 2021). As previously mentioned, 'Relu activation' is utilized to predict the probability for each class. The dataset split ratio was 75% for training, 10% for validation and 15% for testing. We used three PCs, each with an Intel Core i5-8400 processor, NVidia GeForce GTX 1660 GPU, 16 GB of memory, and a 256 GB DDR4 SSD for storage, while we experimented with various models and setups.

6.6. Model comparison

Six deep learning models are compared with MCCT in terms of accuracy and training time using image size of 32 × 32.

6.6.1. VGG16

VGG16 is a state-of-the-art transfer learning model which consists of sixteen weighted layers. The model obtained 92.7% accuracy for the top five test results in the ImageNet dataset. It also won the Large-Scale Visual Recognition Challenge (ILSVRC) competition which was organized by the Oxford Visual Geometry Group. The model can help the kernel learn more complex features, because the VGG model has more depth.

6.6.2. VGG19

VGG19 is a variant of the VGG model with 19 weighted layers, In addition to the VGG16 model, there are three additional FC layers with a total of 4096, 4096, and 1000 neurons. Also, there are five maxpool layers as well as a Softmax classification layer. The ReLU activation function is used in the convolutional layers.

6.6.3. ResNet50

The ResNet50 architecture uses a combination of convolution filters of different sizes to deal with the deterioration of CNN models and reduce the training time. This architecture consists of 48 convolutional layers in total, as well as a maxpool and an average pool layer. There are about 23 million trainable parameters in this model.

6.6.4. ResNet152

ResNet152 is another ResNet model which contains 152 layers. The fundamental innovation of ResNet152 was that it enabled successful training of very deep neural networks with more than 150 layers. ResNet is thought to be a good deep learning architecture because it is easy to optimize and achieves good results. However, as there are many layers in the network architecture, it has a high time complexity.

6.6.5. ResNet50V2

ResNet50V2 is a modified version of the original ResNet50. When evaluated on the ImageNet dataset, ResNet50V2 outperforms both the original ResNet50 and ResNet101. The propagation concept of the connections between blocks in ResNet50V2 was changed.

6.6.6. MobileNet

MobileNet is a considerably faster and smaller CNN design that makes use of a new type of convolutional layer, called Depth wise Separable Convolution. MobileNet models are regarded particularly useful for implementation on mobile and embedded devices due to their modest size.

7. Results and discussion

The results of this study are presented and discussed in this section, including results of the various ablation studies and model evaluation metrics. A discussion regarding the confusion matrix, accuracy loss curves, performance evaluation with reduced number of images is also included in this section to further evaluate the effectiveness of the proposed MCCT model.

7.1. Evaluation metrics

To assess the performance of the proposed classification model, several metrics are computed. A true positive (TP) is a result in which the model classifies the positive class accurately. A true negative (TN) is a result in which the model accurately predicts the negative class. A false positive (FP) is an outcome in which the model forecasts the positive class inaccurately and a false negative (FN) is an outcome in which the model forecasts the negative class incorrectly. Accuracy (ACC) is the proportion of correct predictions:

$$\mathbf{ACC} = \frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}} \tag{12}$$

Precision refers to the percentage of all positive predictions that are actually positive. Recall is the ratio of correctly predicted positive results to all positive predictions.

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}} \tag{13}$$

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}} \tag{14}$$

Specificity is determined by dividing the number of accurate negative predictions by the total number of negative predictions. F1 Score is the harmonic mean of precision and recall.

$$\mathbf{Specificity} = \frac{\mathbf{TN}}{\mathbf{TN} + \mathbf{FP}} \tag{15}$$

Table 5
Ablation study on various ImageJ image enhancement filters

Filter Name	Image size	No. of Parameter	Epoch x training time	Test accuracy (%)	Finding
Spectrum	32 × 32	0.41M	100 × 21s	90.4%	Highest accuracy
Green Fire Blue	32 × 32	0.41M	100 × 21s	90.18%	Near highest accuracy
Blue Orange icb	32 × 32	0.41M	100 × 21s	89.98%	Lower accuracy
16 Colors	32 × 32	0.41M	100 × 21s	89.01%	Lower accuracy

$$\mathbf{F}_1 = 2\,\frac{\mathbf{precision} * \mathbf{recall}}{\mathbf{precision} + \mathbf{recall}} \tag{16}$$

Some other metrics which can be calculated with TP, TN, FP, FN are false positive rate (FPR), false negative fate (FNR), false discovery rate (FDR) and the negative predicted value (NPV):

$$\mathbf{FPR} = \frac{\mathbf{FP}}{\mathbf{FP} + \mathbf{TN}} \tag{17}$$

$$\mathbf{FNR} = \frac{\mathbf{FN}}{\mathbf{FN} + \mathbf{TP}} \tag{18}$$

$$\mathbf{FDR} = \frac{\mathbf{FP}}{\mathbf{TP} + \mathbf{FP}} \tag{19}$$

Table 6
Ablation study on changing transformer layer, activation function, pooling layer, stride size.

Configuration No.	Study 1: changing transformer layer Number of transformer encoder block	No. of Parameters	Epoch x training time	Total time	Test accuracy (%)	Findings
1	3	0.57M	100 × 32s	53-60 minutes	90.63%	High time, High accuracy
2	2	0.41M	100 × 21s	35-40 minutes	90.4%	Medium time, High accuracy
3	1	0.24M	100 × 11s	18-20 minutes	90.24%	Lower time, Near High accuracy
Study 2: changing the activation function						
Configuration No.	Activation function	No. of parameters	Epoch x training time	Test accuracy (%)	Findings	
1	Tanh	0.24M	100 × 10s	90.81%	Lower accuracy	
2	relu	0.24M	100 × 10s	92.06%	Highest accuracy	
3	elu	0.24M	100 × 11s	91.38%	Near highest accuracy	
4	softsign	0.24M	100 × 10s	90.4%	Lower accuracy	
5	softplus	0.24M	100 × 10s	84.97%	Lower accuracy	
Study 3: changing the pooling layer						
Configuration No.	Type of pooling layer	No of parameters	Epoch x training time	Test accuracy (%)	Findings	
1	Max	0.24M	100 × 10s	92.97%	Highest accuracy	
2	Average	0.24M	100 × 10s	92.06%	Lower accuracy	
Study 4: changing the stride size						
Configuration No.	No. of strides	No. of Parameters	Epoch x training time	Test accuracy (%)	Findings	
1	1	0.24M	100 × 10s	94.57%	Highest accuracy	
2	2	0.24M	100 × 5s	93.16%	Near Highest accuracy	
3	3	0.24M	100 × 5s	92.97%	Lower accuracy	
4	4	0.24M	100 × 5s	88.43%	Lower accuracy	

Table 7
Ablation study on changing kernel size, pooling layer kernel size, loss function, batch size

Study 5: changing the kernel size					
Configuration No.	No. of kernel size	No. of Parameter	Epoch x training time	Test accuracy (%)	Finding
1	4	0.3M	100 × 11s	94.33%	Near highest accuracy
2	3	0.24M	100 × 10s	94.57%	Highest accuracy
3	2	0.2M	100 × 12s	93.12%	Lower accuracy
4	1	0.17M	100 × 13s	86.3%	Lower accuracy
Study 6: changing the pooling layer kernel size					
Configuration No.	No. of pooling kernel size	No. of Parameter	Epoch x training time	Test accuracy (%)	Finding
1	5	0.24M	100 × 11s	94.57%	Near highest accuracy
2	4	0.24M	100 × 11s	94.72%	Near highest accuracy
3	3	0.24M	100 × 10s	94.80%	Highest accuracy
4	2	0.24M	100 × 10s	93.62%	Lower accuracy
5	1	0.24M	100 × 10s	92.97%	Lower accuracy
Study 7: changing the loss function					
Configuration No.	Loss Function	No. of Parameter	Epoch x training time	Test accuracy (%)	Finding
1	Binary Crossentropy	0.24M	100 × 10s	94.57%	Near highest accuracy
2	Categorical Crossentropy	0.24M	100 × 10s	94.80%	Highest accuracy
3	Mean Squared Error	0.24M	100 × 10s	94.68%	Near highest accuracy
4	Mean absolute error	0.24M	100 × 10s	93.93%	Lower accuracy
5	Mean squared logarithmic error	0.24M	100 × 10s	26.76%	Lower accuracy
Study 8: changing the batch size					
Configuration No.	Batch size	No. of Parameter	Epoch x training time	Test accuracy (%)	Finding
1	256	0.24M	100 × 9s	94.09%	Lower accuracy
2	128	0.24M	100 × 10s	94.8%	Highest accuracy
3	64	0.24M	100 × 14s	94.56%	Near highest accuracy
4	32	0.24M	100 × 20s	94.3%	Near highest accuracy

Table 8
Ablation study on changing optimizer, learning rate, image size

Study 9: changing the optimizer					
Configuration No.	Optimizer	No. of Parameter	Epoch x training time	Test accuracy (%)	Finding
1	Adam	0.24M	100 × 10s	95.2%	Highest accuracy
2	Nadam	0.24M	100 × 10s	84.52%	Lower accuracy
3	SGD	0.24M	100 × 10s	93.4%	Lower accuracy
4	Adamax	0.24M	100 × 10s	94.18%	Near highest accuracy
5	RMSprop	0.24M	100 × 10s	94.8%	Near highest accuracy
Study 10: changing the learning rate					
Configuration No.	Learning rate	No. of Parameter	Epoch x training time	Test accuracy (%)	Finding
1	0.01	0.24M	100 × 10s	88.12	Lower accuracy
2	0.006	0.24M	100 × 10s	90.42	Lower accuracy
3	0.001	0.24M	100 × 10s	95.37%	Highest accuracy
4	0.0008	0.24M	100 × 10s	94.8%	Near highest accuracy
Study 11: changing the image size					
Configuration No.	Image size	No. of Parameter	Epoch x training time	Test accuracy (%)	Finding
1	64	0.24M	100 × 35s	95.45%	Highest accuracy
2	32	0.24M	100 × 10s	95.37%	Highest accuracy
3	28	0.24M	100 × 9s	94.17%	Lower accuracy
4	16	0.24M	100 × 5s	93.43%	Lower accuracy

$$\mathbf{NPV} = \frac{\mathbf{TN}}{\mathbf{TN} + \mathbf{FN}} \tag{20}$$

The Matthews correlation coefficient (MCC) is a more dependable statistical metric that yields a high score only if the model performed well in all four confusion matrix areas (TP, TN, FP, FN).

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{21}$$

7.2. Results of the ablation study

This section describes all the ablation studies conducted in this research. First, to determine the best image enhancement technique, a total of four ImageJ image filters are explored. The base model is trained with all four image enhancement filters and test accuracies are recorded in Table 5.

It is evident that Spectrum filter outperforms the other ImageJ filters, acquiring a test accuracy of 90.4%. Spectrum ImageJ filter highlights the ROI regions of the chest X-rays, thus increasing the classification performance of the base model. This filter is chosen for further ablation studies.

Various experiments are performed by modifying the model's components and analyzing the model's performance. A classification model's performance may be improved by modifying a number of its components. In this study, a total of 11 studies are performed. The results of these ablation studies are recorded in Tables 6–8.

- Study 1: changing the transformer layer

For this study, the configuration of the transformer layers of the base model is altered by adding or subtracting transformer encoded blocks in order to achieve highest accuracy. Table 6 shows the results of various configurations of the proposed model with different numbers of transformer encoded blocks. The best performance is achieved with configuration 3 (one transformer encoded block) where the model is able to achieve nearly the highest accuracy of 90.24% with a significantly lower training time. The other two configurations had training times of 35-40 minutes and 53-60 minutes whereas configuration 3 took only 18-20 minutes. As configuration 3 contains significantly less trainable parameters, 0.24 million, this configuration has the lowest training time per epoch and is less time consuming. Configuration 3 is therefore chosen for further ablation studies.

- Study 2: changing the activation function

Various activation functions have a different impact on the performance of a classification model. Selecting an optimal activation function can be an effective way to increase the performance of a model. A total of six activation functions: Tanh, Exponential Linear Units (ELU), ReLU, SoftSign and SoftPlus are explored (Table 6). ReLU demonstrates the best performance with a test accuracy of 92.06% and 10 seconds per epoch (Table 6). Thus, the ReLU activation function is chosen for further ablation studies.

- Study 3: changing the type of pooling layer

Experiments with two types of pooling layers: maxpooling and average pooling are conducted (Table 6). Max pooling layer increased the test accuracy from 92.06% to 92.97%. The maxpooling layer is therefore chosen for further ablation studies.

- Study 4: changing the stride size

This study explores various stride sizes in the transformer layers of the modelFour stride sizes: 1, 2, 3 and 4 are tried and the results are shown in Table 6. The performance of the model is increased to 94.57% while maintaining per epoch training time of 10 seconds with a stride size of 1. Thus, a stride size of 1 is chosen to move for further ablation studies.

- **Study 5: changing the kernel size**

Various kernel sizes of the transformer layers are explored in this study. Kernel sizes of 4, 3, 2 and 1 are experimented with and a kernel size of 3 showed the highest test accuracy of 94.57% (Table 7). Furthermore, with this kernel size, the model also had the lowest training time per epoch of 10 seconds, which contributes to reducing the overall training time. A kernel size of 3 for the transformer layers is therefore chosen for further ablation studies.

- **Study 6: changing the pooling layer kernel size**

Similar to the previous study, various kernel sizes of the pooling layers are experimented with. Pooling layer kernel sizes of 4, 5, 3, 2 and 1 are tried (Table 7). A kernel size of 3 outperformed the others with a 94.8% test accuracy (Table 7). For further ablation studies, a kernel size of 3 is chosen for the pooling layers of the model.

- **Study 7: changing the loss function**

Experiments using six different loss functions, namely Binary Crossentropy, Categorical Crossentropy, Mean Squared Error, Mean Absolute Error, Mean Squared Logarithmic Error and Kullback Leibler

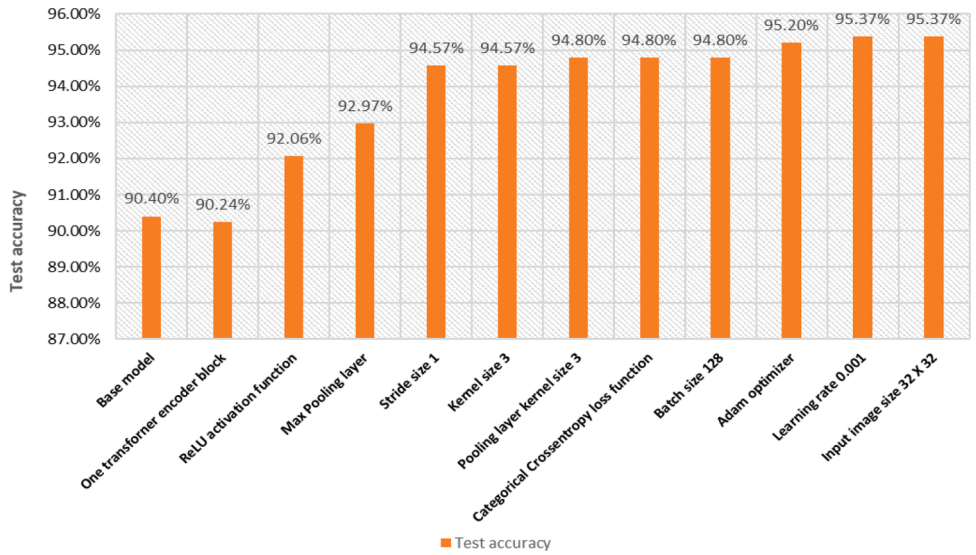


Fig. 10. Improvement in test accuracy over 11 ablation studies.

Divergence are conducted in this study to achieve maximum performance from the model. Table 7 showcases the results of various loss functions. The model equipped with the Categorical Crossentropy loss function is seen to retain the highest test accuracy of 94.8% (Table 7), whereas the accuracy is reduced for other loss functions. Thus, Categorical Crossentropy is selected.

- **Study 8: changing the batch size**

Exploring various batch sizes is necessary as the performance of a classification model may vary with different batch sizes. Experimentation with batch sizes of 256, 128, 64 and 32 is conducted (Table 7). The findings indicate that training the model with a batch size of 128 results in the highest test accuracy, 94.8%, while maintaining a training time per epoch of 10 seconds. The accuracy drops when training with other batch sizes (Table 7). The batch size 128 is therefore chosen for further ablation studies.

- **Study 9: changing the optimizer**

A total of five optimizers, namely Adam, Nadam, SGD, Adamax and RMSprop are experimented with in this study. The learning rates of the optimizers are set to 0.001. Table 8 shows that the highest test accuracy, of 95.2%, is recorded with the Adam optimizer. The Adam optimizer is selected for further ablation studies.

- **Study 10: changing the learning rate**

Further experimentation with the Adam optimizer and various learning rates (0.01, 0.006, 0.001 and 0.0008) is done and the results are recorded in Table 8. The best performance is obtained with learning rate 0.001, resulting in a test accuracy of 95.37% while maintaining a training time of 10 seconds per epoch. Hence, this learning rate is chosen for further ablation studies.

- **Study 11: changing the image size**

For the last study, experiments with the model's input layer image dimensions (image height and width) are conducted. Image sizes of 64 × 64, 32 × 32, 28 × 28 and 16 × 16 are tried. Table 8 contains the results of this study. The highest accuracy (95.45%) is achieved for an image size of 64 × 64 with a per epoch training time of 35 seconds. However, the model managed to achieve nearly the highest testing accuracy, of 95.37%, with an image size of 32 × 32 while significantly lowering the per epoch training time, from 35 seconds to just 10 seconds. As our goal is to build a model with good performance while also keeping time complexity in mind, an image size of 32 × 32 is chosen for the input image dimension because this consumes less training time while maintaining good performance. Fig. 10 visualizes the gradual increase in test accuracy with all ablation studies conducted on the base model.

The final configuration of the MCCT model is summarized in Table 9.

Table 9
Configuration of proposed MCCT architecture after ablation study

Configuration	Value
Image size	32 × 32
Epochs	100
Optimization function	Adam
Learning rate	0.001
Batch size	128
Kernel size	3
Activation function	relu
Loss Function	Categorical Crossen-tropy
pooling layer kernel size	3
stride size	1
pooling layer	Max pooling
projection_dim	128
stochastic_depth_rate	0.1
weight_decay	0.0001

7.3. Performance analysis of proposed model

By conducting ablation studies on the base model, a final MCCT

Table 10
Various matrices computed for performance evaluation of MCCT model.

Measure	Value
Recall	95.44%
Specificity	98.43%
Precision	95.34%
F1 Score (F1)	95.39%
Fall-out or False Positive Rate (FPR)	0.015%
Miss Rate or False Negative Rate (FNR)	0.045%
False Discovery Rate (FDR)	0.046%
Negative Predictive Value (NPV)	98.48%
Matthews Correlation Coefficient (MCC)	0.93%

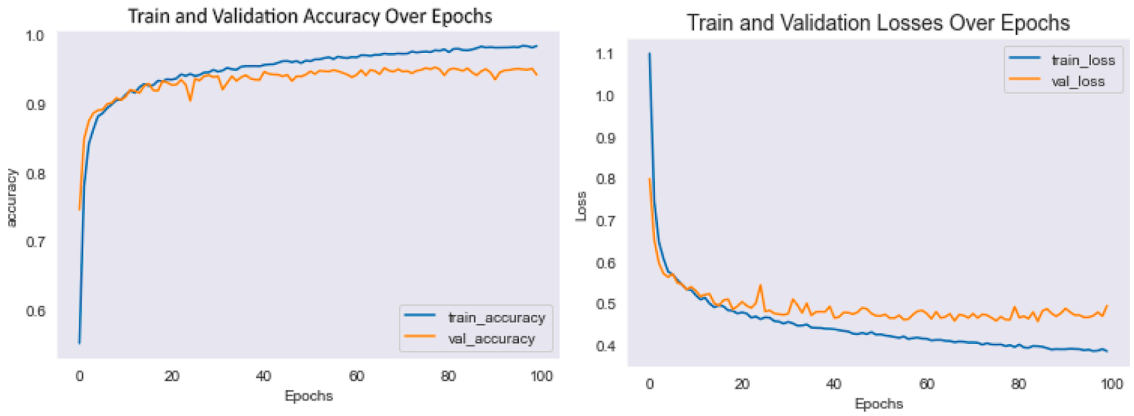


Fig. 11. Loss curve and accuracy curve of CCT model after ablation study.

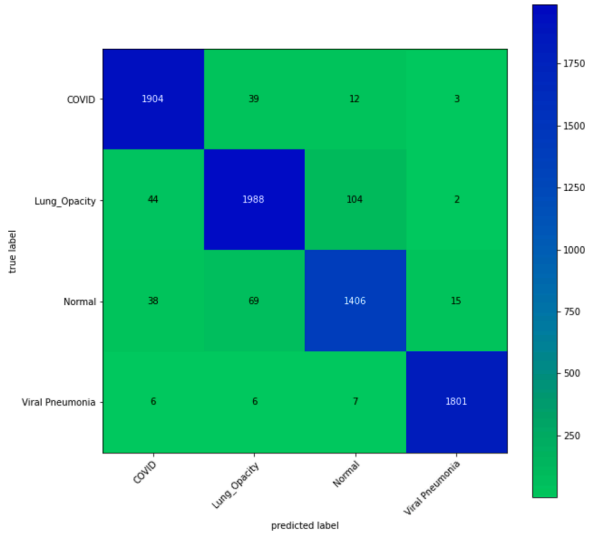


Fig. 12. Confusion matrix of proposed MCCT model after ablation study.

model can be achieved which significant increases the classification performance. This is achieved through alterations and various configurations of the model. Some evaluation metrics including statistical analysis for the proposed MCCT model are shown in Table 10.

While evaluating the proposed MCCT model with the test set, the model achieves an F1 score of 95.39% while it scored 95.44% and 98.43% in terms of recall and specificity respectively, with a precision of 95.34%. FPR and FNR values are also computed, resulting in values of 0.015% and 0.045% respectively. The model managed to keep FDR score quite low (0.046%) while maintaining an NPV value of 98.48%. The MCC value of the model is 0.93% where 1 is considered a perfect MCC score (Houssein et al., 2022).

Fig. 11 visualizes the accuracy and loss curves for the proposed MCCT model. The training and validation curves are found to converge quite efficiently without showing major gaps between the curves, indicating no sign of overfitting during the training process of the model. Similarly, the loss curves show (Fig. 11) steady convergence from the start to the final epoch. It can be concluded that no occurrence of over fitting or underfitting can observed during the training phase of the model.

Fig. 12 shows the confusion matrix generated from the MCCT model. Row values indicate the true labels of the test images. The labels predicted by the model on the test set images are represented by column values. The diagonal values in the confusion matrix (Fig. 12) represent the number of correctly predicted test images by the model. It can be

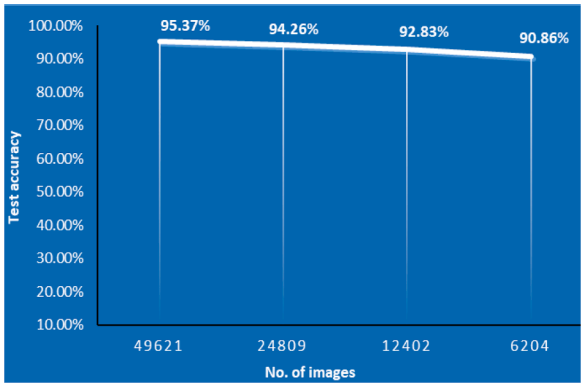


Fig. 13. Evaluation of proposed model with reduced number of images

seen that the model is not biased to one or multiple classes and does not predict any particular class much better than the others. In fact, the model gives near equal numbers of correct predictions across all classes which further demonstrates the robustness of the model.

Experiments are conducted by reducing the number of input images to evaluate the performance consistency of the proposed MCCT model. The MCCT model is again trained and evaluated multiple times. In each step the number of images in the dataset is reduced to nearly half of the previous number of images. Results are shown in Fig. 13.

Fig. 13 shows that while the model is trained with half the number images (24809 images) of the original dataset, the accuracy drops only about 1%. Further decreasing the number of images to 12402, still results in a high-test accuracy, of 92.83%. Training and evaluating the model with only 6204 images, results in moderate performance with a test accuracy of 90.86%. Six thousand images are a very low number for traditional CNN and ViT models. However, even with such a small number of images (six thousand), the proposed MCCT model is able to produce a good result while maintaining low training times. This demonstrates the performance consistency of the model.

7.4. Comparison with transfer learning models

For evaluation, the proposed model is compared with six state of the art transfer learning CNN models. All six models are trained and tested on the same dataset as the proposed model and the input image dimensions are kept at 32 × 32 pixels. The Categorical crossentropy loss function and the ReLu activation function are utilized for all transfer learning models. The models are tested with the Adam optimizer with a learning rate of 0.001 and a batch size of 128. All models are trained for 100 epochs. The findings of this experiment are shown in Table 11.

These findings show that among the six transfer learning models,

Table 11
Performance comparison with six states of the art transfer learning CNN models.

Model	Number of params	epochs	Total time (min)	Per epoch time	Optimizer	Batch size	Image size	Learning rate	Accuracy
VGG19	20026436	100	100-120	61-63s	Adam	128	32	0.001	76.97%
VGG16	14716740	100	100-120	61-63s	Adam	128	32	0.001	79.51%
ResNet152	58379140	100	100-120	61-63s	Adam	128	32	0.001	53.39%
ResNet50	23595908	100	100-120	61-63s	Adam	128	32	0.001	67.77%
ResNet50V2	23572996	100	100-120	61-63s	Adam	128	32	0.001	65.35%
MobileNet	3232964	100	100-120	61-63s	Adam	128	32	0.001	43.42%
MCCT	241861	100	18-20	11-12s	Adam	128	32	0.001	95.37%

VGG16 achieved the highest accuracy, 79.51%, while the other models had accuracies in the range of 43% to 77% (Table 11). It can be seen that CNN models are struggling to achieve even moderate performance when trained with input image of small dimensions like 32 × 32 pixels. In contrast, MCCT can be seen to be very robust as it outperformed all six CNN models with a top accuracy of 95.37% (Table 11) for 32 × 32-pixel input images. Training with such small sized images results in lower training times and requires less storage space which can be important for large datasets. The MCCT model has 241,861 trainable parameters (Table 11) which is quite low compared to the CNN models (Table 11). The small number of parameters contributes to shorter training periods, of 10–11 seconds per epoch. CNN models with larger parameter numbers require more than 60 seconds per epoch for training (Table 11). The total training time for 49621 images is reduced from nearly two hours (for traditional CNN models) to just 18-20 minutes (for our model). This is a significant improvement in terms of time complexity. In addition, acquiring near optimal performance with smaller sized images requires less memory and storage space, making the model less resource hungry and contributing to reduced space complexity.

8. Related work

In recent years, a growth of Machine Learning and Deep Learning approaches are observed across a variety of applications in lung disease detection and classification.

Kong and Cheng (2022) proposed a model based on DenseNet, VGG16 feature fusion and used the attention mechanism to extract deep features and classify lung images into three classes. To deal with data imbalance and unequal distribution, they developed the fine-tuned global attention block (GAB) and a category attention block (CAB) while ResNet was used to segment image data. Their model achieved an accuracy of 98% for binary classification and 97.3% for multi classification. The processing time required was 104 minutes for DenseNet201, 90 minutes for VGG16, 103 minutes for DenseNet169, 100 minutes for Xception and 110 minutes for their proposed model. Xu et al. (2021) proposed a two-stage prediction model to classify lung diseases into five classes. Their proposed MANet model uses the mask attention mechanism (MA) as spatial attention maps for all CXR images in order to extract the lung regions in the first stage and applies a CNN to classify the segmented images in the second stage. MA improved the test classification accuracy of ResNet34, ResNet50, VGG16, and Inception v3 where ResNet50 with MA achieved the greatest average test accuracy of 96.32%. However, the average training time of the evaluated models was 2 to 7 hours. Other authors (Umer et al., 2022), used CNN to extract features of CXR images and classify them into four classes using VGG16 and AlexNet. They generated 1000 images from only 264 original ones using ImageDataGenerator. The prediction accuracies were 97.21% for binary class and 84.76% for multi-class prediction. Regarding training time, their proposed model required 1.5 hours, VGG16 3.25 hours and AlexNet 2.5 hours. Narin et al. (2021) used five pre-trained models, ResNet50, ResNet152, ResNet101, Inception-ResNetV2 and InceptionV3, to detect lung disease using CXR images. Three different datasets were used to evaluate the performance of the proposed models. ResNet50 performed best with accuracies of 96.1% for Dataset-1, 99.5% for Dataset-2 and 99.7% for Dataset-3. Training times for all models were high. Bhattacharyya et al. (2022) detected lung disease using a three step Deep learning based approach. A conditional generative adversarial network (C-GAN) was used in the first step to segment the raw X-ray data in order to acquire the lung images. They then fed the segmented lung images into a unique pipeline that combined key point extraction methods and trained deep neural networks (DNN) for discriminatory feature extraction. In the last stage, several machine learning (ML) models were used to categorize COVID-19, pneumonia, and normal lungs. They obtained the best accuracy of 96.6% using the VGG-19 for binary class. In the study of Wang et al. (2020), a hybrid model of deep and machine learning was developed to classify lung disease. Experimenting with five pre trained models including VGG16, InceptionV3, ResNet50, Xception and DenseNet121, Xception was found to perform best. Afterwards, they developed a hybrid Xception and SVM model and achieved an accuracy of 99.33%. Zebin and Rezvy (2021) applied pre-trained VGG16, ResNet50, and EfficientNetB0 to classify lung images into three classes. They used 802 CXR images and trained a generative adversarial framework (CycleGAN) to generate minority class data. The classification accuracy was 90%, 94.3%, and 96.8% for VGG16, ResNet50, and EfficientNetB0 respectively. Other authors (Akter et al., 2021), applied 11 existing CNN models to classify lung diseases, with some image\s preprocessing techniques and some modification to the models. A robust model, MobileNetV2, acquired an accuracy of 98%. The lowest processing time of these models was 2.5 hours. Ismael and Şengür (2021) also used pre-trained ResNet18, ResNet50, ResNet101, VGG16, and VGG19 models and SVM with various kernel functions to classify lung diseases. With the Linear kernel function, the deep features collected from the ResNet50 model and SVM classifier yielded an accuracy of 94.7%. Toraman et al. (2020) proposed an artificial neural network system based on Convolutional CapsNet to classify chest X-rays into three classes, where each class contained same number of 1050 images. To evaluate the performance, they used a 10-fold cross-validation, resulting in an accuracy of 97.21% for binary class classification, and 97.24% and 84.22% for multi-class classification. The processing time required was 72 s/epoch with data augmentation, and 16 s/epoch without data augmentation. However, an improved accuracy can be achieved when using a completely balanced dataset. A limitation of this study is the high processing time. Jin et al. (2021) proposed a three-step ensemble model which includes a feature extractor, a feature selector, and a classifier to classify lung images into three classes. Comparing the performance of five existing models, their proposed AlexNet+ReliefF+SVM obtained the best accuracy of 98.64%. They used AlexNet as feature extractor which had a running time of (5.991 s). Marques et al. (2020) proposed an automated lung disease diagnostic system introducing an EffecientNet pipeline for classifying chest X-ray images into three classes. The proposed EfficientNet model recorded accuracies of 99.62% and 96.70% for binary and multi-class classification respectively. The model training time was 111.83 minutes for multi-class and 79.16 minutes for binary class classification with a total of 17M parameters. A balanced dataset was used where each class contained 404 X-ray images. Duong et al. (2021) proposed a Hybrid model of modified EfficientNet and modified original Vision Transformer to detect tuberculosis from chest X-ray images. The authors

Table 12
Comparative analysis of the existing models and the proposed model

Paper	Model	Number of Image	Image Size	Training Time	Limitation
Kong et al. (2022)	DenseNet201, VGG16, DenseNet169, Xception, Proposed model	6518	512 × 512	104 min, 90 min, 103 min, 100 min, 110 min	1. Limited amount of data 2.High processing time and low precision. 3.No image preprocessing techniques used.
Xu et al. (2021)	ResNet34+MA, ResNet50+MA, VGG16+MA, Inceptionv3+MA	6792	512 × 512	132.6 min, 148.2 min, 453.6 min, 153.6 min	1. Limited amount of data 2. Higher processing time 3. No image preprocessing techniques used.
Umer et al. (2021)	Proposed model, VGG16, AlexNet	264	-	90 min, 195 min, 150 min	1. Limited amount of data. 2.High processing time and low multiclass classification accuracy.
Narin et al. (2021)	ResNet50, ResNet101, ResNet152, InceptionV3, Inception-ResNetV2	8088	224 × 224 to 229 × 299	Minimum 243.6 minutes to maximum 381 minutes	1.Very high training times. 2. Limited amount of data 3. No image preprocessing techniques used.
Bhattacharyya et al. (2022)	VGG-16, VGG-19, DenseNet-169, DenseNet-201, sCNN	1030	-	17.43 min, 21.48 min, 13.52 min, 38.3 min, 3.33 min	1. Limited amount of data. 2. No image preprocessing techniques used.
Wang et al. (2020)	VGG16, InceptionV3, ResNet50, Xception, DenseNet121,	1105	224 × 224	0.65 min/epoch, 0.73 min/epoch, 0.68 min/epoch, 0.7 min/epoch, 0.73 min/epoch	1. Limited amount of data. 2. No image preprocessing techniques used.
Zebin et al. (2020)	VGG16, ResNet50, EfficientNetB0	802	224 × 224	15 min, 15 min, 15 min	1. Limited amount of data. 2. No image preprocessing techniques used.
Akter et al. (2021)	VGG16, VGG19, MobileNetv2, InceptionV3, NFNet, ResNet50, ResNet101, DenseNet, EfficientNetB7, AlexNet, GoogLeNet	13808	299 × 299	310.8 min, 376.2 min, 325.2 min, 456 min, 366 min, 272.4 min, 370.8 min, 387 min, 385.2 min, 374.4 min, 150 min	1.High processing time with binary class prediction.
Ismael et al. (2021)	ResNet50 Features + SVM, Fine-tuning of ResNet50, End-to-end training of CNN, BSIF + SVM	380	224 × 224	-	1. Limited amount of data 2. No image preprocessing techniques used.
Toraman et al. (2020)	Convolutional CapsNet	3150	128 × 128	With Data augmentation 1.2 min/epoch, Without data augmentation 0.27 min/epoch	1.Limited amount of data
Jin et al. (2021)	AlexNet + ReliefF + SVM	1743	227 × 227 × 3	-	1. Limited amount of data 2. No image preprocessing techniques used.
Marques et al. (2020)	EffecientNet	404	-	111.83 minutes	1. Limited amount of data 2. Higher processing time 3. No image preprocessing techniques used.
Duong et al.	ViT_Base_EfficientNet_B1_224	28672	384 × 384 × 3	720 min	1.Higher processing time
Tangudu et al. (2022)	GoogleNet, InceptionResNet, ResNet50, MobileNet, MNRSC	5184	224 × 224, 128 × 128	0.85 min/epoch, 1.95 min/epoch, 0.98 min/epoch, 0.75 min/epoch, 0.78 min/epoch	1.Fails to maintain its performance in noisy and low-quality datasets with imbalanced data
Mamalakis et al. (2021)	DenResCov-19	6696	-	-	1. Limited amount of data. 2.Imbalanced Dataset 3. No image preprocessing techniques used
Proposed Study	**MCCT**	21165	32 × 32	18-20 min	**Our study addressed these limitations:** 1.A large amount of data 2.Dataset balanced 3.Image preprocessing techniques used 4. Low processing time 5. Maintain model performance in noisy and low-quality dataset

experimented with 14 configurations of the hybrid model and ViT_Base_EfficientNet_B1_224 achieved the maximum accuracy of 97.72%. The proposed model's parameter size was 94M and the training phase took 12 hours to complete. In the study of Tangudu et al. (2022) a model was proposed based on MobileNet and residual separable convolution blocks for the detection of lung diseases from chest X-rays, having 3.626 M parameters and requiring 89.9 seconds training time. The model achieved 99% accuracy in binary classification. Mamalakis et al. (2021) established a transfer learning pipeline that consists of DenseNet-121 and ResNet-50 with an extra CNN block. The author proposed DenResCov-19 to classify lung images into three classes using four datasets. Their achieved F1 values were 98.21, 87.29, 76.09, 83.17%.

In the majority of the studies discussed above and included in the literature table (Table 12), similar shortcomings can be observed such as an imbalanced dataset or a lack of data, long processing times, and a lack of image processing techniques.

Image resolution is an important factor in developing common deep learning models for medical imaging (Lakhani, 2020). In a study, Sabottke and Spieler (2020) examined well-known deep CNN models to identify radiographs at image resolutions ranging from 32 × 32 pixels to 600 × 600 pixels. The study showed that when the pixel size of the image is reduced, it removes the information that CNNs need for classificationand the accuracy drops. Our study overcomes this limitation by proposing a robust model that employs low-resolution images (32 × 32 pixel) and achieves high accuracy.

9. Conclusion

A lung CAD system is developed in this work to classify lung images into four categories. DCGAN is applied to the dataset to balance the number of images and image pre-processing methods are utilized to eliminate noise and artifacts and to enhance the visibility of the ROIs of the images. We propose a MCCT model based on the original CCT model, which outperforms CNNs in computational efficiency and accuracy. Our proposed network is subjected to ablation research in order to assess and improve the model's robustness, resulting in a training accuracy of 95.37%, a precision of 95.34%, a recall of 95.44% and an F1-score of 95.39%. The proposed model is compared with six transfer learning models, VGG19, VGG16, ResNet152, ResNet50, ResNet50V2, and MobileNet in terms of accuracy and training time. For the same set of 32 × 32-pixel images, the other models have accuracies ranging from 43% to 77%, whereas our model has an accuracy of 95.37%. The results of this study show that our proposed technique produced an accurate lung disease classification model, when trained with 32 × 32-pixel input images. When training with large datasets, smaller size images might result in shorter training times and less storage space, which can be very advantageous. Our model also minimizes the number of trainable parameters, reducing the training time. The proposed model may help to correctly classify lung X-ray images in a very short time. Finally, we intend to evaluate the MCCT in a variety of datasets in order to assess its generalization and robustness in different medical image classification tasks.

CRediT authorship contribution statement

Inam Ullah Khan: Methodology, Software, Validation, Data curation, Writing – original draft. **Sami Azam:** Conceptualization, Supervision, Writing – review & editing, Project administration. **Sidratul Montaha:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **Abdullah Al Mahmud:** Visualization, Validation, Methodology, Writing – original draft, Writing – review & editing. **A.K.M. Rakibul Haque Rafid:** Writing – original draft, Writing – review & editing, Visualization, Validation. **Md. Zahid Hasan:** Supervision, Writing – original draft. **Mirjam Jonkman:** Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

Akter, S., Shamrat, F. M. J. M., Chakraborty, S., Karim, A., & Azam, S. (2021). COVID-19 detection using deep learning algorithm on chest X-ray images. *Biology (Basel), 10*, 1174.

Alhasan, M., & Hasaneen, M. (2021). Digital imaging, technologies and artificial intelligence applications during COVID-19 pandemic. *Computerized Medical Imaging and Graphics*. https://doi.org/10.1016/j.compmedimag.2021.101933

Ayris, D., Imtiaz, M., Horbury, K., Williams, B., Blackney, M., Hui See, C. S., & Shah, S. A. A. (2022). Novel deep learning approach to model and predict the spread of COVID-19. *Intelligent Systems with Applications, 14*, Article 200068. https://doi.org/10.1016/J.ISWA.2022.200068

Beeravolu, A. R., Azam, S., Jonkman, M., Shanmugam, B., Kannoorpatti, K., & Anwar, A. (2021). Preprocessing of breast cancer images to create datasets for deep-CNN. *IEEE Access, 9*, 33438–33463.

Bhattacharyya, A., Bhaik, D., Kumar, S., Thakur, P., Sharma, R., & Pachori, R. B. (2022). A deep learning based approach for automatic detection of COVID-19 cases using chest X-ray images. *Biomedical Signal Processing and Control, 71*, Article 103182.

Borghesi, A., Zigliani, A., Golemi, S., Carapella, N., Maculotti, P., Farina, D., & Maroldi, R. (2020). Chest X-ray severity index as a predictor of in-hospital mortality in coronavirus disease 2019: A study of 302 patients from Italy. *International Journal of Infectious Diseases, 96*, 291–293.

Bowles C, Chen L, Guerrero R, Bentley P, Gunn R, Hammers A, et al. GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks. 2018. Available: http://arxiv.org/abs/1810.10863.

Breuel TM. Efficient Binary and Run Length Morphology and its Application to Document Image Processing. CoRR. 2007;abs/0712.0121. Available: http://arxiv.org/abs/0712.0121.

Cozzi, D., Albanesi, M., Cavigli, E., Moroni, C., Bindi, A., Luvarà, S., … Miele, V. (2020). Chest X-ray in new Coronavirus Disease 2019 (COVID-19) infection: findings and correlation with clinical outcome. *La Radiologia Medica, 125*, 730–737.

Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V. and Le, Q.V., 2018. Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501.

Dhar, P. (2021). A method to detect breast cancer based on morphological operation. *International Journal of Education and Management Engineering, 11*, 25–31.

Duong, L. T., Le, N. H., Tran, T. B., Ngo, V. M., & Nguyen, P. T. (2021). Detection of tuberculosis from chest X-ray images: Boosting the performance with vision transformer and transfer learning. *Expert Systems with Applications, 184*, Article 115519.

Hassan, N., Ullah, S., Bhatti, N., Mahmood, H., & Zia, M. (2021). The Retinex based improved underwater image enhancement. *Multimedia Tools and Applications, 80*, 1839–1857.

Hassani, A., Walton, S., Shah, N., Abuduweili, A., Li, J. and Shi, H., 2021. Escaping the big data paradigm with compact transformers. arXiv preprint arXiv:2104.05704.

Houssein, E. H., Abohashima, Z., Mohamed, Elhoseny, & Mohamed, W. M. (2022). Hybrid quantum-classical convolutional neural network model for COVID-19 prediction using chest X-ray images. *Journal of Computational Design and Engineering, 9*, 343–363.

Huang, X., Bi, N., & Tan, J. (2022). *Visual transformer-based models: A survey, in: Pattern Recognition and Artificial Intelligence* (pp. 295–305). Cham: Springer International Publishing.

Hussain, E., Hasan, M., Rahman, M. A., Lee, I., Tamanna, T., & Parvez, M. Z. (2021). CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images. *Chaos, Solitons & Fractals, 142*, Article 110495.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *InInternational conference on machine learning*. PMLR.

Islam, K., 2022. Recent Advances in Vision Transformer: A Survey and Outlook of Recent Work. arXiv preprint arXiv:2203.01536.

Ismael, A. M., & Şengür, A. (2021). Deep learning approaches for COVID-19 detection based on chest X-ray images. *Expert Systems with Applications, 164*, Article 114054.

Jin, L., Tan, F., & Jiang, S. (2020). Generative adversarial network technologies and applications in computer vision. *Computational Intelligence and Neuroscience, 2020*, Article 1459107.

Jin, W., Dong, S., Dong, C., & Ye, X. (2021). Hybrid ensemble model for differential diagnosis between COVID-19 and common viral pneumonia by chest X-ray radiograph. *Computers in Biology and Medicine, 131*, Article 104252.

Kaggle. COVID-19 radiography Database, Available. https://www.kaggle.com/.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). *Transformers in vision: A survey*. ACM Comput. Surv.

Kong, L., & Cheng, J. (2022). Classification and detection of COVID-19 X-Ray images based on DenseNet and VGG16 feature fusion. *Biomedical Signal Processing and Control, 77*, Article 103772.

Kora Venu, S., & Ravula, S. (2020). Evaluation of deep convolutional generative adversarial networks for data augmentation of chest X-ray images. *Future Internet, 13*, 8.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the Acm, 60*, 84–90.

Lakhani, P. (2020). The Importance of Image Resolution in Building Deep Learning Models for Medical Imaging. *Radiology Artificial Intelligence, 2*. https://doi.org/10.1148/RYAI.2019190177

Lorencin, I., Šegota, S. B., Andelïc, N., Mrzljak, V, Cabov, T., Španjol, J., & Car, Z. (2021). On urinary bladder cancer diagnosis: Utilization ´of deep convolutional generative adversarial networks for data augmentation. *Biology (Basel)*, 10.

Mamalakis, M., Swift, A. J., Vorselaars, B., Ray, S., Weeks, S., Ding, W., Clayton, R. H., Mackenzie, L. S., & Banerjee, A. (2021). DenResCov-19: A deep transfer learning network for robust automatic classification of COVID-19, pneumonia, and tuberculosis from X-rays. *Computerized Medical Imaging and Graphics, 94*. https://doi.org/10.1016/j.compmedimag.2021.102008

Marques, G., Agarwal, D., & de la Torre D\'\iez, I. (2020). Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network. *Applied Soft Computing, 96*, Article 106691.

Montaha, S., Azam, S., Rafid, A. K. M. R. H., Ghosh, P., Hasan, M. Z., Jonkman, M., & de Boer, F. (2021). BreastNet18: A high accuracy fine-tuned VGG16 model evaluated using ablation study for diagnosing breast cancer from enhanced mammography images. *Biology (Basel), 10*, 1347.

Montaha, S, Azam, S, Rafid, AKMRH, Islam, S, Ghosh, P, & Jonkman, M (2022). A shallow deep learning approach to classify skin cancer using down-scaling method to minimize time and space complexity. *Plos One, 17*(8), Article e0269826. https://doi.org/10.1371/journal.pone.0269826

Motamed, S., Rogalla, P., & Khalvati, F. (2021). Data augmentation using Generative Adversarial Networks (GANs) for GAN-based detection of Pneumonia and COVID-19 in chest X-ray images. *Informatics in Medicine Unlocked, 27*, Article 100779.

Narin, A., Kaya, C., & Pamuk, Z. (2021). Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. *Pattern Analysis and Applications, 24*, 1207–1220.

Paul, S., & Chen, P. Y. (2022, June). Vision transformers are robust learners. *Proceedings of the AAAI Conference on Artificial Intelligence, (Vol. 36, No. 2,*, 2071–2081.

Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Kashem, Abul, bin, S., Islam, M. T., al Maadeed, S., Zughaier, S. M., Khan, M. S., & Chowdhury, M. E. H. (2021). Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. *Computers in Biology and Medicine, 132*, Article 104319.

Sabottke, C. F., & Spieler, B. M. (2020). The effect of image resolution on deep learning in radiography. *Radiology Artificial Intelligence, 2*. https://doi.org/10.1148/RYAI.2019190015

Salehinejad, H., Valaee, S., Dowdell, T., Colak, E., & Barfett, J. (2018). Generalization of deep neural networks for chest pathology classification in x-rays using generative adversarial networks. In *In2018 IEEE International Conference on Acoustics, Speech and signal Processing* (pp. 990–994). IEEE.

Sarv Ahrabi, S., Scarpiniti, M., Enzo, Baccarelli, & Momenzadeh, A. (2021). An accuracy vs. Complexity comparison of Deep Learning architectures for the detection of COVID-19 disease. *Computation (Basel), 9*, 3.

Tangudu, V. S. K., Kakarla, J., & Venkateswarlu, I. B. (2022). COVID-19 detection from chest x-ray using MobileNet and residual separable convolution block. *Soft Computing, 26*, 2197–2208.

Tomasi, C., & Manduchi, R. (2002). Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. Narosa Publishing House.

Toraman, S., Alakus, T. B., & Turkoglu, I. (2020). Convolutional capsnet: A novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks. *Chaos, Solitons & Fractals, 140*, Article 110122.

Umer, M., Ashraf, I., Ullah, S., Mehmood, A., & Choi, G. S. (2022). COVINet: a convolutional neural network approach for predicting COVID-19 from chest X-ray images. *Journal of Ambient Intelligence and Humanized Computing, 13*, 535–547.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Vocaturo, E., Zumpano, E., & Caroprese, L. (2021). Convolutional neural network techniques on X-ray images for Covid-19 classification. In *Proceedings - 2021 IEEE International Conference on Bioinformatics and Biomedicine* (pp. 3113–3115). BIBM. https://doi.org/10.1109/BIBM52615.2021.9669784, 2021.

Wang, D., Mo, J., Zhou, G., Xu, L., & Liu, Y. (2020). An efficient mixture of deep and machine learning models for COVID-19 diagnosis in chest X-ray images. *Plos One, 15*, Article e0242535.

Xu, Y., Lam, H.-K., & Jia, G. (2021). MANet: A two-stage deep learning method for classification of COVID-19 from Chest X-ray images. *Neurocomputing, 443*, 96–105.

Zebin, T., & Rezvy, S. (2021). COVID-19 detection and disease progression visualization: Deep learning on chest X-rays for classification and coarse localization. *Applied Intelligence, 51*, 1010–1021.

Zhang, J., Xie, Y., Wu, Q., & Xia, Y. (2019). Medical image classification using synergic deep learning. *Medical Image Analysis, 54*, 10–19.

Zumpano, Ester, Fuduli, Antonio, Vocaturo, Eugenio, & Avolio, Matteo (2021). Viral pneumonia images classification by Multiple Instance Learning: preliminary results. In *Proceedings of the 25th International Database Engineering & Applications Symposium (IDEAS '21)* (pp. 292–296). Association for Computing Machinery. https://doi.org/10.1145/3472163.3472170.