REPUBLIC OF TURKEY

ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY

INSTITUTE OF GRADUATE EDUCATION

DEPARTMENT OF COMPUTER ENGINEERING

A MOBILENET BASED CNN MODEL WITH A NOVEL FINE TUNING MECHANISM FOR COVID-19 INFECTION DETECTION

ERCAN GÜRSOY

MASTER OF SCIENCE

ADANA 2023

REPUBLIC OF TURKEY
ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY

INSTITUTE OF GRADUATE EDUCATION
DEPARTMENT OF COMPUTER ENGINEERING

A MOBILENET BASED CNN MODEL WITH A NOVEL FINE TUNING MECHANISM FOR COVID-19 INFECTION DETECTION

ERCAN GÜRSOY
MASTER OF SCIENCE

SUPERVISOR
ASSOC. PROF. DR. YASİN KAYA

ADANA 2023

ABSTRACT

A MOBILENET BASED CNN MODEL WITH A NOVEL FINE TUNING MECHANISM FOR COVID-19 INFECTION DETECTION

Ercan GÜRSOY

Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Yasin KAYA

January 2023, 41 pages

A new coronavirus called COVID-19, which emerged in Wuhan, China, in December 2019, is very dangerous because of its rapid global spread worldwide, causing severe acute respiratory syndrome. It has infected 349.641.119 people worldwide and caused over 5.592.266 deaths, as reported in January 2022. Although there are several different diagnostic methods, polymerase chain reaction (PCR) is considered the gold standard for laboratory diagnosis of the COVID-19 pathogen. However, test results are obtained within a few hours to two days, and this relatively late response is the main barrier to early intervention. Researchers have focused on alternative methods that use x-ray imaging to shorten the time to diagnose the disease. This study proposes a deep-transfer learning approach with novel fine-tuning mechanisms for detecting COVID-19 disease using chest X-ray images. The model is based on the MobileNetV2 architecture. To evaluate the proposed model, we used a combined dataset from two publicly available databases containing three classes: normal, COVID-19, and pneumonia X-ray images. In our approach, we proposed one classical and two new fine-tuning mechanisms to increase classification accuracy and achieved average accuracy rates of 95.62%, 96.10%, and 97.61% for 3-class cases with five-fold cross-validation. In addition, our third model reduced 81.92% of the total fine-tuning operations and achieved better results. The numerical results show that the proposed approach achieves promising results when raw data without complex preprocessing steps are used.

Keywords: COVID-19 Disease detection · Deep Transfer Learning · CNN · Fine-Tuning · MobileNet

ÖZET

COVİD-19 ENFEKSİYON TESPİTİ İÇİN YENİ BİR İNCE AYAR MEKANİZMASINA SAHİP MOBİLENET TABANLI CNN MODELİ

Ercan GÜRSOY

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Yasin KAYA

Ocak 2023, 41 sayfa

Aralık 2019'da Çin'in Wuhan kentinde ortaya çıkan COVID-19 adlı yeni bir koronavirüs, dünya çapında hızla yayılması ve ciddi akut solunum sendromuna neden olması nedeniyle çok tehlikeli bir hal aldı. Ocak 2022'de bildirildiği üzere, virüs dünya çapında 349.641.119 kişiye bulaştı ve 5.592.266'dan fazla ölüme neden oldu. Birkaç farklı teşhis yöntemi olmasına rağmen, polimeraz zincir reaksiyonu (PCR), COVID-19 patojeninin laboratuvar teşhisi için ana standart olarak kabul edilir. Ancak test sonuçları birkaç saat ile iki gün arasında elde edilmesi erken müdahalenin önündeki en büyük engeldir. Bu yüzden araştırmacılar, hastalığı teşhis etme süresini kısaltmak için röntgen görüntülerini kullanan alternatif yöntemlere odaklandılar. Bu çalışma, göğüs röntgeni görüntülerini kullanarak COVID-19 hastalığını saptamak için yeni ince ayar mekanizmalarına sahip bir derin aktarım öğrenme yaklaşımı önermektedir. Model, MobileNetV2 mimarisine dayanmaktadır. Önerilen modeli değerlendirmek için, üç sınıf içeren halka açık iki veritabanından birleştirilmiş bir veri seti kullandık: normal, COVID-19 ve pnömoni X-ray görüntüleri. Yaklaşımımızda, sınıflandırma doğruluğunu artırmak için bir klasik ve iki yeni ince ayar mekanizması önerdik ve beş kat çapraz doğrulama ile 3 sınıflı sınıflandırmada ortalama %95.62, %96.10 ve %97.61 doğruluk oranlarına ulaştık. Ayrıca üçüncü modelimiz, toplam ince ayar işlemlerinin %81.92'sini azalttı ve daha iyi sonuçlar elde etti. Sayısal sonuçlar, önerilen yaklaşımın, karmaşık ön işleme adımları olmaksızın ham veriler kullanıldığında umut verici sonuçlar elde ettiğini göstermektedir.

Anahtar Kelimeler: COVID-19 hastalık tespiti, derin transfer öğrenme, CNN, ince ayar, MobileNet

1. INTRODUCTION

COVID-19 (COronaVIrus Disease 2019 - COVID-19) is a contagious disease originated by coronavirus 2 (SARS-CoV-2) that causes severe acute respiratory syndrome (Ai et al., 2020). Coronaviruses are large enveloped, positive, single-stranded RNA (V'kovski et al., 2021) that affect various animals and humans (Pal et al., 2020). The World Health Organization (WHO) officially reported the disease on March 11, 2020 and declared a pandemic affecting more than 227 countries (Virtual Press Conference on COVID-19, 2020 March Howpublished = "Https://Www.Euro.Who.Int/En/Health-Topics/Health-Emergencies/Coronavirus-Covid19/News/News/2020/3/Who-Announces-Covid-19-Outbreak-a-Pandemic," n.d.). COVID-19 has infected over 349.641.119 individuals all over the world and caused over 5.592.266 deaths, as reported on January 23, 2022 (WHO Coronavirus (COVID-19) Dashboard, Howpublished = "Https://Covid19.Who.Int/," n.d.). Pneumonia was the first clinical manifestation of SARS-CoV-2 associated with COVID-19 disease, allowing case detection (Seibert et al., 2020). The gastrointestinal symptoms and asymptomatic infections, especially among young children, were also described in recent reports. In patients, symptoms of the disease, such as cough, pyrexia, nasal obstruction, fatigue, and upper respiratory infection, usually begin in less than a week (Velavan & Meyer, 2020). However, the infection found in about 75% of patients is fatal, as evidenced by computed tomography during hospitalization. (Feng et al., 2020). Despite the development of numerous successful vaccines against the disease, the epidemic could not be brought under control for a long time. The main factors in the rapid spread of the disease are the reduced effectiveness of vaccines with different mutations and inadequate medical testing (Kerr et al., 2021). Thus, prompt detection and isolation of positive cases are crucial to controlling the spread. Given the increase in COVID-19 cases worldwide, more attention needs to be paid to containing the outbreak. Using reverse transcription-polymerase chain reaction (RT-PCR) is the common method to detect the COVID-19 (Mallett et al., 2020). Rapid and accurate detection of the disease is critical for epidemic control (for Economic Co-operation & (OECD), 2020) because test results are available in a few hours to 2 days (Lan et al., 2020). Medical imaging (chest X-rays) and artificial intelligence (AI) have been shown to be beneficial in detecting and rapidly evaluating patients infected with COVID-19 virus (Borkowski et al., 2020). Because it takes a long time for test results to become available, image classification of COVID-19 and the development and distribution of AI tools are urgently needed to address the current situation (Hassanien et al., 2020). Some classification methods have been used to detect lung diseases (Chaunzwa et al., 2021), such as tuberculosis, cancer, and pneumonia, that may be related to COVID-19 in chest X-rays in previous studies using deep learning methodologies. The importance of medical imaging is recognized as a crucial source of information for rapid diagnosis (Song, Zheng, Li, Zhang, Zhang, Huang, Chen, Wang, Zhao, Zha, et al., 2021). A combination of AI methods and chest radiographs can help illustrate and detect COVID-19 complications (Afshar-Oromieh et al., 2021) X-ray images are essential in the rapid clinical evaluation of COVID-19 (Wong et al., 2020). If chest X-rays are normal, the patient can be sent home after quarantine prevention while waiting for the PCR test result. This can avoid hospital overcrowding. The presence or absence of pathologic findings on a chest x-ray is the basis for whether the patient remains in the hospital or is sent home. It plays a crucial role in making a quick decision. Thanks to the development of solutions based on computer imaging, the fight against the coronavirus epidemic have been accelerated (Ulhaq et al., 2020) More recently, medical imaging systems have used deep-learning techniques and helped medical personnel make rapid diagnoses (Suzuki, 2017). Researchers used the convolutional neural network (CNN) model to detect the disease on chest X-ray images (Maior et al., 2021). It is reported that classical computer imaging systems cannot solve complex problems such as image identification, organ recognition, bacterial colony classification, and disease detection (Abiyev & Ma'aitah, 2018). However, computer imaging systems that use deep learning models can solve image classification problems with a high success rate (Barstugan et al., 2020). Deep learning approaches require robust GPU systems and many training examples to train the models. To overcome this challenge, transfer learning can be applied to pre-trained CNN models for small datasets by redesigning the last few layers and fine-tuning the model (Aneja & Aneja, 2019). The CNN model can achieve high accuracy if appropriate hyperparameters are adjusted and efficient fine-tuning approaches are used. CNN, which involves computing weights and extracting features during training, was developed to handle multidimensional data such as time series or image data. The name "convolutional" comes from using a convolution operator to help solve complex operations. In addition, CNNs are most used in healthcare because they can automatically generate features from images (Navamani, 2019). CNNs can also learn from one task and transfer it to another using transfer learning and fine-tuning. This method has demonstrated its effectiveness in classification tasks (Tajbakhsh et al., 2016). Numerous deep learning models have been reported in the literature for detecting and classification of COVID-19 cases on chest X-ray images. These studies include several CNN models such as Resnet50, VGG16 (Shorfuzzaman & Masud, 2020), MobileNetV2 (Apostolopoulos & Mpesiana, 2020a), DenseNet (Iskanderani et al., 2021), and several customized new CNN models such as DarkCovidNet (Ozturk et al., 2020a), InstaCovidNet (Gupta et al., 2021a). Most of these studies, far from creating a new model, apply existing models to COVID data. The datasets used usually have a small amount of data. When these studies were published, there were few publicly available COVID X-ray image datasets. For this reason, the reliability of these models trained with few data could not be thoroughly tested. In our study, we tested five different CNN architectures, namely VGG16, DenseNet, MobileNetV2, InceptionV3, and ResNet, to find a suitable model for our problem. We trained the models for 100 epochs with 80% of the images, and the results show that MobileNetV2 obtained more successful results. In this study, a deep-transfer learning approach with novel fine-tuning mechanisms is proposed to detect COVID-19 diseases from chest X-rays. The model based on MobileNetV2 architecture was used. We created a relatively large dataset by combining two publicly available datasets containing COVID-19 and pneumonia X-ray images, namely Chest X-Ray Images (ChestX-RayImagesPneumonia, 2022) and COVID-19 Radiography Database (Covid-19RadiographyDatabase, 2022a), to evaluate the proposed model. The dataset consisted of three classes: normal, pneumonia, and COVID-19. Our approach integrated the training phases of the MobileNetV2 CNN architecture, such as data augmentation, transfer learning, and fine-tuning, into a single design. We proposed one classical and two new fine-tuning mechanisms to enhance the performance of the models. We also employed data augmentation methods such as zooming and rotation to preserve the diversity of the test data. Our theory is that during the training of the network, the last layers are trained more, which increases the ability to generalize to the new problem while simultaneously hiding the information learned from the previous training of the first layers, which leads to more successful results. In this study, 3 methods were used for comparative theory implementation. In the fine-tuning phase, the first approach was classical by opening the last 50 layers and freezing the 102 remaining layers of the MobileNetV2 model. The second approach used a step function to fine-tune the model. The last method used a predefined mathematical exponential function to fine-tune the model. We performed experimental tests with three models and compared the numerical results. The numerical results show that the proposed approach achieves promising results using raw data without complex preprocessing steps. The main contributions of this study can be summarized as follows: 1. A new deep transfer learning model based on MobileNetV2 was proposed to detect COVID-19 diseases. 2. Two novel fine-tuning mechanisms were implemented in the proposed model. 3. We combined two public COVID-19 X-ray databases to evaluate our model and trained our model with this dataset using two standard training methods, 30-70% random split training and cross-validation. 4. The proposed model reduced 81.92% of the total fine-tuning operations and achieved better results. The rest of the paper is organized as follows. Section 2 contains a literature review. Section 3 contains information about the datasets, the proposed CNN architecture, and its sublevels. Section 4 contains the numerical results obtained using the proposed approach. Section 5 discusses the proposed approach with the papers in the literature. Finally, Section 6 gives the conclusion and future work.

2. LITERATURE REVIEW

Deep learning algorithms and CNN models in image analysis and processing in the biomedical field have yielded successful results (Bala et al., 2019a). Moreover, various CNN-based deep neural networks can achieve remarkable results in ImageNet competition (Krizhevsky et al., 2012a). X-rays and CT are widely used in biomedical imaging systems. Pneumonia and early-stage cancer are both diagnosed with X-ray technology. However, scanning CT is a more advanced method based on X-ray technology that can detect organ changes. Soft tissue cannot be analyzed with X-rays and 2D imaging. CT uses 3D computer vision technology that takes multiple images of the organ from different angles. Although both X-rays and scans from CT can provide images of internal body structures, conventional X-rays tend to overlap.

In contrast, a scan of CT eliminates this overlap, highlights internal anatomy, and provides a clear picture of health status. This study provides a taxonomy for categorizing the diagnostic system of COVID-19. As shown in Figure 2.1, two different perspectives are used for deep learning techniques and imaging modalities, namely transfer learning and customized DL methods.

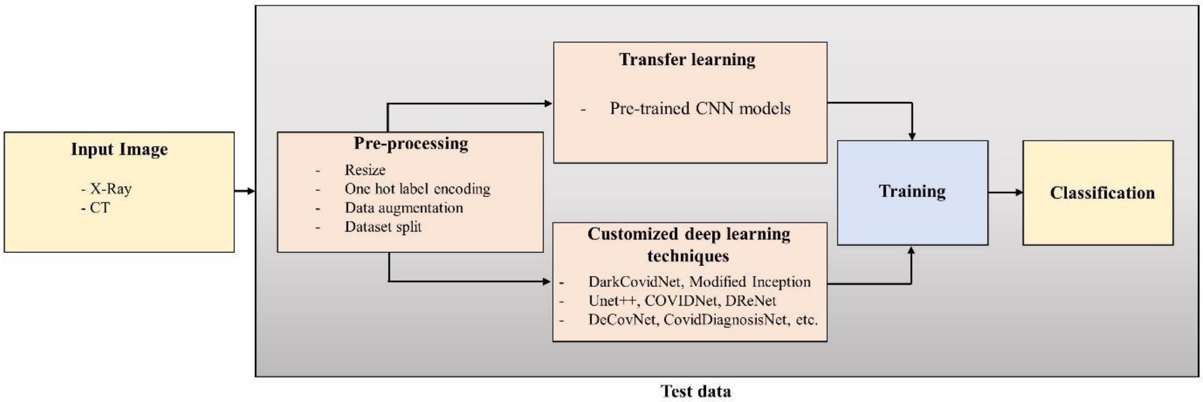

Figure 2.1 Overall workflow summary of the methods

A stored network previously trained on an extensive data set, especially for an image classification task, is called a pre-trained model. This model can be adapted to a specific task by using the previously trained weights and changing the top layers to fit the new task called transfer learning (TL) (Yildirim et al., 2019). Using a pre-trained model in transfer learning has many advantages. When a model is trained from scratch with large datasets, this takes a lot of time and computing power (Litjens et al., 2017). The pre-trained model allows faster convergence in transfer learning (Alom et al., 2019). In addition, we found that different models have been successfully applied in the investigation of recent studies. ResNet and its derivatives were mainly determined as the main architecture in these studies.

A COVID-19 recognition system based on DL with multiview fusion was presented by Wu et al. The system uses ResNet50, a pre-trained CNN model. Data were obtained from two local hospitals in China. For the experiment, 495 images were used, 368 of which had COVID-19 cases and 127 had other disease infections. For training, testing, and validation, the dataset was divided into three segments: 80%, 10%, and 10%. The input image used by the system was resized to 256 x 256 before the model was created. The system attained an accuracy of about %76 (Wu et al., 2020). In another study implementing a 3-class detection model for COVID-19, the authors used CT scans to detect the disease. They used ResNet18 with TL and achieved an 86.7% success rate with model-based segmentation procedures (Xu et al., 2020). In (Rehman et al., 2020), they developed ResNet101-based model to distinguish bacterial pneumonia, viral pneumonia, and normal images. They also applied their model to diagnose COVID-19, and the success rate of the model is 98.75%. Another COVID-19 diagnostic system was proposed by Jin et al. using ResNet152. They evaluated their model using 496 images with COVID-19 and 1385 healthy images and achieved an accuracy rate of 94.98% (C. Jin et al., 2020).

In another study that uses CT scans as input (Ardakani et al., 2020), the authors presented a system for diagnosing COVID-19 using pre-trained CNN models including AlexNet, MobileNetV2, ResNet18, ResNet101, VGG-16, VGG-19, SqueezeNet, GoogleNet, ResNet50 and Xception. They used 1020 CT healthy and COVID-19 CT scans in the proposed approach and experimental findings showed that the ResNet101 model achieves 99.51% accuracy using a 20-80% hold-out validation method. Furthermore, Narin et al. presented an approach to recognize COVID-19 using X-ray images implementing pre-trained CNN models. The dataset contains 100 X-rays that consist of 50 COVID-19 and 50 healthy images. They used the cross-validation method to evaluate the model and achieved 98% accuracy using the ResNet50-based model (Narin et al., 2021).

Finally, Bukhari et al. proposed another diagnostic model for COVID-19 using a pre-trained ResNet50 model. They evaluated their model using 278 chest X-rays from three different classes: 89 COVID-19, 93 healthy, and 96 pneumonia images. They used the hold-out method that divides the dataset into 80% for training and 20% for testing. Their model achieved 98.18%, 98.14%, 98.24%, 98.19% accuracy, precision, sensitivity, and F1 score, respectively (Bukhari et al., 2020).

Another common model which used in COVID-19 detection is VGG and its derivatives. In (Dansana et al., 2020), the authors implemented VGG-19 and InceptionV2 pre-trained models and applied fine-tuning to these models. They also used decision trees to evaluate the data. Using VGG-19, InceptionV2, and decision trees, they achieved success rates of 91%, 78%, and 60%, respectively. Moutounet-Cartan presented an approach to detect COVID-19 and lung infections from X-ray images. This approach uses VGG-16, VGG-19, InceptionResNetV2, Xception, and InceptionV3 pre-trained CNN architectures. A total of 327 images were collected to evaluate the model, including 152 healthy, 125 COVID-19, and 50 images of various lung diseases. They determined VGG-16 as the primary model with an accuracy of 84.1% (Moutounet-Cartan, 2020). Horry et al. presented a model for diagnosing COVID-19 in X-rays using deep TL models, including Inception, Xception, ResNet, and VGG. The dataset for testing included 200 healthy, 100 pneumonia, and 100 COVID-19 images. The precision, sensitivity, and F1-score were 83%, 80%, and 80%, respectively (Horry et al., 2020).

Another study that uses the VGG model we examined, the COVIDX-Net, was developed by Hemdan et al. to identify COVID-19 using X-ray images. They used 50 images consisting of 25 healthy and 25 COVID-19. According to their numerical results, DenseNet and VGG-19 achieved a successful classification rate of 90% and F1-score of 91% (Hemdan et al., 2020).

The AlexNet is another popular architecture that is used in medical image analysis. Cifci presented an approach to recognize COVID-19 from CT based on deep TL approaches. He used 5800 CT scans from a public repository, of which 80% were used for training and 20% for testing. In the experimental results, the AlexNet model performed better. The overall accuracy of AlexNet was 94.74%, and its specificity and sensitivity were 87.37% and 87.45%, respectively (Cifci, 2020). In another study, the authors presented a system that combines deep TL methods such as GoogleNet, ResNet18, AlexNet and Generative Adversarial Network. The authors utilized 307 images, including COVID-19, normal, bacterial, and viral. Googlenet had the highest accuracy rate of 80.6% for four-class classification. For three and two classes classification, Alexnet and Googlenet achieved 85.2% and 100% accuracies, respectively (Loey et al., 2020).

Sethy and Behra proposed a hybrid model that uses both Support Vector Machine (SVM) and pre-trained CNN models to detect COVID-19. They used pre-trained CNN models for feature extraction and SVM for classification using these features. They employed two datasets, the first of which included 25 COVID-19 X-ray images and 25 normal images and the second of which included 133 X-rays images that were both healthy and diseases such as Acute Respiratory Distress Syndrome, SARS, and Middle East Respiratory Syndrome. The proposed hybrid model achieved an accuracy rate of 95.38% (Sethy & Behera, 2020). Abbas et al. proposed another model that distinguished between COVID-19 and healthy images and used ResNet18 architecture as the main design. This model was trained using 196 images, including 80 from healthy individuals, 105 from COVID-19, and 11 images from SARS patients. They used 30-70% hold-out validation technique and they achieved 95.12% accuracy, 97.9% sensitivity, 91.87% specificity, and 93.36% precision (Maguolo & Nanni, 2021).

In addition to these popular pre-trained architectures, some studies are conducted with models such as DenseNet, Inception, MobileNet, UNet, and NasNetLarge. Apostolopoulos et al. proposed a model to distinguish between normal, viral, and COVID-19 cases. They used three different datasets and implemented a pre-trained MobileNetV2 as deep TL, and they achieved success rates of 98.66%, 96.46%, and 96.78%, for dataset1, dataset2, and dataset3, respectively. In (Rehman et al., 2020), they developed a CNN model to distinguish viral pneumonia, bacterial pneumonia, and healthy images. They also applied their model to diagnose COVID-19, and the model's success rate is 98.75%.

Yousefzadeh et al. presented a model using CT images called the COVID-AI system, which implements ResNet, EfficientNetB0, DenseNet, and Xception. They used a dataset that contained 2124 CT scans, including 1418 normal and 706 COVID-19 images. They implemented a 20-80 hold-out validation method to evaluate their model. The overall accuracy of 96.4% was achieved with the proposed model (Yousefzadeh et al., 2021). Gupta et al. used various pre-trained CNN models to classify COVID-19 and pneumonia by extracting features from X-rays. Their model had a 99.08% success rate (Gupta et al., 2021b).

Luz et al. also implemented a TL model that uses a pre-trained EfficientNet to analyze CT scans. Their experiments achieved an accuracy rate of 93.9 (Luz et al., 2021). In addition, Jin et al. proposed a COVID-19 identification model that uses DL methods, including 3D U-Net++ with a set of pre-trained CNN models containing DPN92, Inception-v3, ResNet50, and Attention ResNet50. The system was trained using 541 healthy and 850 COVID-19 samples. They achieved an accuracy rate of 97.1% (S. Jin et al., 2020).

In addition, Chen et al. also used UNet++ based method for identifying COVID-19 using CT scans. This work used 46,096 images from a hospital in China and achieved 98.85% accuracy. In another study, Javaheri et al. suggested a DL model, CovidCTNet, for diagnosing COVID-19 from CT scans. The system was developed using the U-Net architecture. The model used 89,145 CT scans, of which 32,230 samples had COVID-19, 25,699 samples had community-acquired pneumonia (CAP), and 31,216 samples had healthy scans. They achieved an accuracy, sensitivity, specificity, and AUC rates of 91.66%, 87.5%, 94.0%, and 95.0%, respectively (Javaheri et al., 2020).

Minaee et al. suggested a model to detect COVID-19 in X-rays using deep TL approaches, such as ResNet18, DenseNet-121, ResNet50, and SqueezeNet. A total of 5071 images were obtained from various publicly available datasets, and they used 3100 images (100 COVID-19 and 3000 healthy images) to test their model. They reported that SqueezeNet obtained the best results achieving 100% sensitivity and 98.00% accuracy (Minaee et al., 2020). In another study that uses ResNet, NASNetLarge, DenseNet169, Inception ResNetv2, and Inceptionv3 as TL architecture, Punn and Agarwal presented a diagnostic model for COVID-19 recognition. They used 1076 chest X-ray images to evaluate their model and the training, testing, and validation sets were assigned 80%, 10%, and 10% of the total dataset, respectively. Experimental findings demonstrated that NASNetLarge performs significantly better than other methods and achieved 98% accuracy, 88% precision, 91% sensitivity, 99% specificity, and 89% F1 score (Punn & Agarwal, 2021).

Deep learning algorithms and CNN models in image analysis and image processing in the biomedical field have produced successful results for years (Bala et al., 2019b). Moreover, different CNN-based deep neural networks can achieve remarkable results in the ImageNet competition (Krizhevsky et al., 2012b). Thus, various CNN-based deep neural networks have been widely used for medical image classification tasks these days. Diagnosis of COVID-19 by chest X-ray is associated with pneumonia symptoms. Image classification methods developed by researchers for COVID-19 or pneumonia are divided into the following categories: Machine learning (ML) methods (Barstugan et al., 2020),(Kwekha-Rashid et al., 2021),(Ambita et al., 2020), statistical approaches (Castillo & Melin, 2020),(Castillo & Melin, 2021), CNN architectures (Minaee et al., 2020), (L. Wang et al., 2020), (Varshni et al., 2019) complex CNN models (L. Wang et al., 2020), (Monshi et al., 2021), (Ulhaq et al., 2020) adversarial networks

(Shams et al., 2020), and transfer learning methods (Shorfuzzaman & Masud, 2020), (Pathak et al., 2020), (Apostolopoulos & Mpesiana, 2020b) .

In addition, Wang et al. employed a deep learning approach to calculate features and they achieved a correct detection rate of 79.3% on a database consisting of 325 COVID-19 and 740 pneumonia images (S. Wang et al., 2021). In another study, Apostolopoulos et al. proposed a model to distinguish between normal, viral, and COVID-19 cases. They implemented a model using transfer learning techniques and achieved success rates of 98.66%, 96.46%, and 96.78%, respectively (Apostolopoulos & Mpesiana, 2020b). Ucar and Korkmaz, the authors used a deep learning model to categorize coronavirus related infections using X-ray images. They applied Bayesian Optimization-based fine-tuning to their mçodel to calculate their results (Ucar & Korkmaz, 2020). In (Nour et al., 2020), the authors also proposed a classification model by optimizing the hyper-parameters of machine learning models using a Bayesian optimization algorithm and performed transfer learning. In another study that implemented a 3-class COVID-19 detection model, the authors used CT images to detect disease and they attained an 86.7% success rate with model segmentation-based procedures (Ozturk et al., 2020b). Khan et al. integrated pretrained Xception into a model to automatically classify COVID-19 images. Their study attained an accuracy rate of 87.5 (Khan et al., 2020). In (Rehman et al., 2020), a deep learning approach implementing a CNN model was developed to distinguish viral pneumonia, bacterial pneumonia, and normal images. They also applied their model to diagnose COVID-19 and the success rate of the model is 98.75%. Narin et. al. proposed a model using five different pre-trained CNN models namely, InceptionV3, ResNet50, ResNet101, ResNet152, and Inception-ResNetV2 on chest X-ray images. The proposed model attained a success rate of 98.00% (Narin et al., 2021). In (Toğaçar et al., 2020), the authors proposed MobileNetV2 based deep learning approach to detect COVID-19 infection from Xray images. They used stacked fuzzy colored and original images to increase their model's success and achieved an accuracy rate of 97.06%. They used Social Mimic Optimization algorithm to optimize their parameters. Gupta et al. used various pre-trained CNN models that classify COVID-19 and pneumonia by extracting features from healthy chest X-rays. The model had a 99.08% success rate (Gupta et al., 2021b). In another study that uses CNN to calculate features, the researchers developed a model by combining two CNN models to classify chest X-ray images and attained a classification success of 91.40% (Rahimzadeh & Attar, 2020). In (Dansana et al., 2020) the authors used the CNN model that implements VGG-19 and InceptionV2 models. They also employed decision trees to evaluate the data. They achieved success rates of 91%, 78%, and 60% using fine-tuned version VGG-19, InceptionV2, and decision tree, respectively. In (Mansour et al., 2021), the authors developed an unsupervised DL-based model to detect COVID-19 and they obtained accuracy rates of 98.7% and 99.2% for binary and multi-class classification, respectively. Luz et al. (Luz et al., 2021) presented an EfficientNet model to analyze X-ray images. They trained their model on 183 COVID-19 samples and achieved an accuracy rate of 93.9%. Punn and Agarwal introduced a DNN to detect coronavirus symptoms in another study. The method used 108 COVID-19 cases and obtained an average classification rate of 97% (Punn & Agarwal, 2021). The researchers also used pre-trained CNN models for transfer learning purposes to identify COVID 19 images (Hemdan et al., 2020), (Bukhari et al., 2020). The feature extraction and design of the deep network used in diagnosing COVID-19 infection are highly effective in the results.

Some reported approaches were hybrid models that implement classical machine learning algorithms with deep learning approaches and identify coronavirus-associated infections on chest images. Customized Deep Learning techniques enable architectural development with more accurate performance because they are tailored to the specific application. Combining a particular deep learning technique or algorithms with other AI fields such as data mining, machine learning, and nature-inspired algorithms (Hassanien et al., 2020), (Y. Li et al., 2021) customized models have been developed (E. Wang et al., 2019). Unlike pre-trained models, the network does not use prior biases and weights. Thus, it requires much computing power and execution time.

In (L. Li et al., 2020), the authors presented a novel approach, COVNet, based on ResNet50 to identify COVID-19 on CT scans. The dataset includes 4536 CT scans consisting of 1296 COVID-19, 1735 CAP, and 1325 other lung diseases. They split the dataset into training and test, including 90% and 10%, respectively. They achieved specificity, accuracy, and AUC rates of 96%, 90%, and 96%, respectively.

In another study, Ozturk et al. proposed a DNN to present a specialized version of DarkNet, DarkCovidNet, for detection of COVID-19 on chest x-rays. They achieved F1 score, precision, specificity, and accuracy of 96.51%, 98.03%, 95.13% and 98.08%, respectively (Ozturk et al., 2020b).

Moreover, to identify COVID-19 from CT scans, He et al. proposed a hybrid DL approach that reveals a new supervised learning technique combined with transfer learning called CRNet. This approach was evaluated using 746 CT scans, including 349 COVID-19 and 397 healthy images. The training, testing, and validation sets had weights of 60%, 25%, and 15%, respectively. They achieved an accuracy of 86%, an F1 score of 85%, and an AUC value of 94% (He et al., 2020).

Another hybrid method that combines CNNs and the Whale Optimization Algorithm (WOA) was proposed by Elghamrawy and Hassanien to identify COVID-19 from CT scans. In this method, CNNs were used for segmentation and predictions were made about the probability of patient response considering some factors. The dataset, which included 583 CT scans consisting of 432 COVID-19 images and 151 pneumonia images, was obtained from publicly available databases. They used 65-35% hold-out validation method and achieved a successful recognition rate of 96.40% (Elghamrawy & Hassanien, 2020). In another study, Khan et al. customized the Xception model to identify COVID-19 images automatically. Their study attained an accuracy rate of 87.5% (Khan et al., 2020).

In (S. Wang et al., 2021), the authors implemented Inception-based architecture, called Modified-Inception, for detecting COVID-19. They evaluated the proposed model using 1040 CT scans, of which 740 were categorized as COVID-19 and 325 as healthy. The experimental results show that their model had an accuracy of 79.3%, a sensitivity of 83%, specificity of 67%, and precision of 55%. In another study that used CT scans to diagnose COVID-19, Chen et al. recommended the Unet++ model based on the Unet model. The dataset used for this model included 51 COVID-19 and 55 other diseases. They achieved a 95.24% success rate (J. Chen et al., 2020).

Another diagnostic system utilizing a CNN model on CT scans was proposed by Liu et al. The proposed model, called COVIDNet, was based on DenseNet-264 and included 4 dense blocks. Every block holds numerous units, each of which had two connected stacks and received feature maps from the previous layers via dense block connections. These blocks included a normalization layer, a convolutional layer, and a ReLU activation. The dataset includes 1073 healthy and 920 COVID-19 CT scans. The data set is divided into training, testing, and validation groups of 60%, 20%, and 20%, respectively. The accuracy of the model was 94.3% (B. Liu et al., 2021).

Song et al. proposed a model, DeepPneumonia, based on DRE-Net. Image features were extracted by combining Feature Pyramid Network and ResNet50 model, which was used to develop the proposed system. A total of 1990 CT scans were included in the dataset, consisting of 708 healthy, 505 bacterial pneumonia, and 777 COVID-19 images. The proposed model achieved an accuracy of 94%, precision of 96%, F1 score of 94%, sensitivity of 93%, and AUC of 99% (Song, Zheng, Li, Zhang, Zhang, Huang, Chen, Wang, Zhao, Chong, et al., 2021).

Similarly, Zheng et al. developed a CNN model called DeCoVNet. They used three layers: the subsampling layer, the 3D convolutional layer, and the batch norm layer. They evaluated their model using 630 CT scans implementing an 80-20% hold-out validation method. They attained an accuracy of 90.1%, specificity of 91.1%, AUC of 95.9%, NPV of 84%, and sensitivity of 90.7% (Zheng et al., 2020).

In (Amyar et al., 2020), the authors proposed an approach that uses two decoders for segmentation, an MLP for classification, and an encoder for reconstruction. For each of the three tasks, a single encoder was used to input the sample data using CT scans, two decoders were used for reconstruction and segmentation, and the multilayer perceptron was used to classify the images based on the presence of COVID-19 cases. 1044 CT scans were used, including 449 COVID-19, 100 healthy, 98 lung cancer, and 397 other lung disease images. The proposed model achieved a sensitivity of 94%, specificity of 79%, an accuracy of 86%, and AUC of 93%.

In another study, Hasan et al. developed a Q-deformed entropy technique (QDE-DF) using DL features to identify COVID-19 from CT scans. Deep features were extracted using CNN and Q-deformed entropy, and LSTM was then used to categorize cases based on the deep features. They used 321 CT scans, including 118 COVID-19, 96 pneumonia, and 107 healthy images. The dataset was split into training and testing ratios of 70% and 30%. This model achieved a classification accuracy of 99.68% (Hasan et al., 2020).

Singh et al. presented a hybrid model to detect COVID-19 using CT scans. In this work, multicenter differential evolution was used to determine the initial parameters of the model. They implemented an artificial neural network (ANN) and an ANN with a fuzzy inference system (ANNFIS). 150 CT scans were included in the evaluation dataset, 75 COVID-19 and 75 healthy subjects. They tested Various validation ratios of the training and test for the experiment, including 20-80%, 30-70%, 40-60%, 50-50%, 60-40%, 70-30%, 80-20%, and 90-

10%. The sensitivity of the model was 90.70%, and the accuracy was 93.25% (Singh et al., 2020).

Ucar and Korkmaz proposed a model based on the Bayes SqueezeNet, COVIDiagnostic, to diagnose COVID-19. They used 1591 pneumonia, 45 COVID-19, and 1203 healthy chest x-rays. Experimental findings showed that their model achieved an accuracy rate of 98.26% and a specificity rate of 98.26% (Ucar & Korkmaz, 2020).

Farid et al. presented an innovative method to identify COVID-19 infections from CT images. They used a hybrid composite feature extraction technique to calculate the features, and the Stack Hybrid Classification (SHC) method was used to classify these extracted features. SHC combines multiple models, such as ensemble learning, to improve model performance. To validate the model, a 10-fold cross-validation scheme was performed. They achieved an accuracy of 94.11%, a precision of 99.4%, an F1-score of 94%, and an AUC value of 99.4% (Farid et al., 2020).

Rahimzadeh and Attar suggested a CNN model to recognize COVID-19 on chest X-ray images. They used ResNet50V2 and Xception CNN architectures as the main design and evaluated their model using 180 COVID-19, 6054 pneumonia images, and 8851 healthy images. A 5-fold cross-validation scheme was implemented, and a 99.50% success rate was achieved (Rahimzadeh & Attar, 2020).

In (Zhang, 2020), the authors developed a novel system using the CoroNet model and AutoEncoder to diagnose COVID-19 individuals on X-rays. They use semi-supervised learning and classify them using a user-defined network to extract the features. The problem of insufficient data was solved by the semi-supervised learning approach of the designed architecture. A total of 18529 images related to different diseases were used. Specifically, 9579 images were pneumonia, 99 images were COVID-19, and 8851 images were healthy people. 90% of the data were used for training and 10% for test sets. They achieved accuracy, precision, recall, and F1-score values of 93.50%, 93.63%, 93.50%, and 93.51%, respectively.

Farooq and Hafeez proposed a DL architecture, COVID-ResNet, based on the ResNet50 model to identify COVID-19. They trained it from scratch, applying parameter optimization and learning rate discrimination. They used 648 X-ray images, consisting of 239 bacterial pneumonia, 8 COVID-19, 234 healthy, and 149 viral pneumonia, and achieved an accuracy of 96.23% (Farooq & Hafeez, 2020).

Islam et al. developed a hybrid model for the identification of COVID-19 based on the hybridization of CNN with LSTM. The dataset of 4575 X-rays, including 1525 COVID-19, 1525 normal, and 1525 viral pneumonia images, was used to evaluate the effectiveness of the hybrid model. According to the numerical results, their model achieved 99.20% accuracy, 99.30% sensitivity, and 98.90% specificity (Islam et al., 2020).

In (Aslan et al., 2021), the authors employed two DL approaches to recognize COVID-19 cases from chest X-ray images. These proposed architectures use artificial neural networks for automatic lung segmentation (pre-processing) on CT scans and X-rays. AlexNet architecture is implied in both architectures as a TL. The second presented model has a hybrid structure that includes a Bidirectional Long Short-Term Memory (BiLSTM) layer that also considers temporal aspects. They used 2905 images, including 219 COVID-19, 1345 pneumonia, and 1341 healthy images, and the second hybrid architecture has a classification accuracy of 98.70%, while the first design achieved an accuracy rate of 98.14%.

Turkoglu introduced a hybrid model called COVIDetectioNet, which combines DL and classical ML methods such as AlexNet and SVM to identify COVID-19 using X-ray images. The researchers collected a dataset from GitHub and Kaggle that included 219 COVID-19, 4290 pneumonia, and 1583 normal images. COVIDetectioNet attained an accuracy of 99.18% (Turkoglu, 2021).

In (Tammina, 2022) the authors suggested a DL approach, namely CovidSORT, to detect COVID-19 using 5910 chest X-ray images from local hospitals in China to evaluate the performance of the proposed method. According to the numerical results, the accuracy of the CovidSORT model was 96.83%. Mahmud et al. designed a unique CNN method, CovXNeT, based on depth-stretched folding to identify COVID-19. CovXNeT was applied with a combined dataset containing 610, 610, and 5856 X-ray images from different sources. Experimental results show that CovXNeT attained an accuracy of 98.1% (Mahmud et al., 2020).

As shown in previous studies, the transfer learning and fine-tuning models of deep learning architectures are widely used to detect diseases such as COVID-19 and pneumonia. Most of works impelemented data augmentation operations such as image resizing. Although they yielded promising results, most of the works used minimal images to evaluate their models. In our approach, we used a balanced and broad dataset with 10518 training and 1248 test images with three classes.

3. MATERIAL AND METHODS

Before discussing the details of current methods for detecting COVID-19 diseases, it is essential to know the basics of Deep Learning. This section briefly discusses CNN architectures, datasets, data augmentation and preprocessing transfer learning, and fine-tuning approaches. Also, the proposed deep learning approaches are described in detail. A general block diagram of the study is shown in Figure 3.1. We used pre-trained CNN models, VGG16, MobileNetV2, InceptionV3, ResNet50, DenseNet, and EfficientNet, to find the most suitable design for our problem in this study. These models were trained on 80% of the entire dataset for 100 epochs. Table 3.1 shows the results of this test. As can be seen in Table 3.1, MobileNetV2 was the most effective architecture. Therefore, we chose MobileNetV2 as the main design.

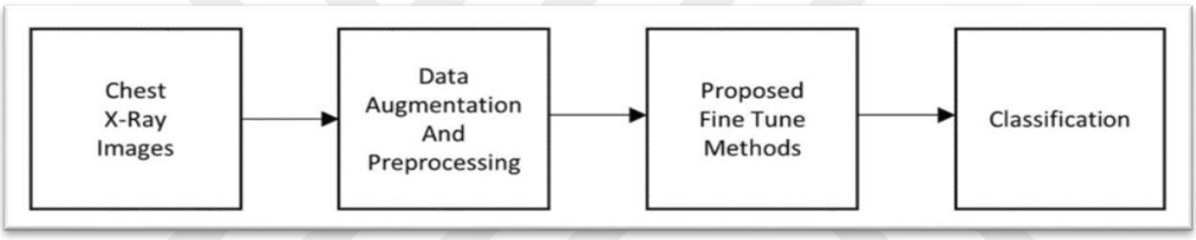

Figure 3.1 Block Diagram of proposed approach

Table 3.1 Achievements of pre-trained CNN models for COVID-19 disease classification before any fine-tune

Model	**Train Loss**	**Train Acc.**	**Val. Loss**	**Val. Acc.**
VGG16	0.9279	0.5938	0.9064	0.6250
InceptionV3	0.0857	0.9722	0.0730	0.9375
MobileNetV2	**0.0970**	**0.9657**	**0.1183**	**0.9549**
ResNet50	0.0445	0.9826	0.2045	0.9488
DenseNet	0.0598	0.9896	0.1710	0.9375
EfficientNet	0.3335	0.9890	0.8310	0.8346

3.1. Fundamentals of Deep Learning

DL and Deep Neural Networks (DNNs) have seen a significant upsurge in today's scientific research in recent years due to their excellent learning capabilities (X. Wang et al., 2020), (Gadekallu et al., 2020), (Roy et al., 2019). DL is used in various applications due to its ability to adapt to different types of data in numerous domains, such as classification problems, image recognition, and object recognition (Kamilaris & Prenafeta-Boldú, 2018).

Learning methods for DL fall into three categories such as unsupervised, semi-supervised, and supervised. In supervised learning, the training of the model is done with the input-output pair. Each detected layer generates an input vector and a control signal corresponding to the expected value. Based on the existing labels, the technique predicts the output labels (Kanavati et al., 2020). A strategy that lies between supervised and unsupervised learning is called semi-supervised learning. In semi-supervised learning, both unlabeled and labeled values form the training data. When used together with labeled data, unlabeled data can significantly increase learning accuracy (Zhu & Goldberg, 2009). Unsupervised learning analyses and groups unlabeled data using ML methods. These find hidden patterns or datasets without human intervention (Aldahiri et al., 2021). Anomaly detection and clustering are two examples of algorithms used in conjunction with these methods. Clustering involves finding patterns or anomalies in a dataset that are like each other (de Simone & Jacques, 2019). This unsupervised learning technique for anomaly detection is commonly used in security domains (Lima & Keegan, 2020). Figure 3.2 shows the different techniques used in DL.

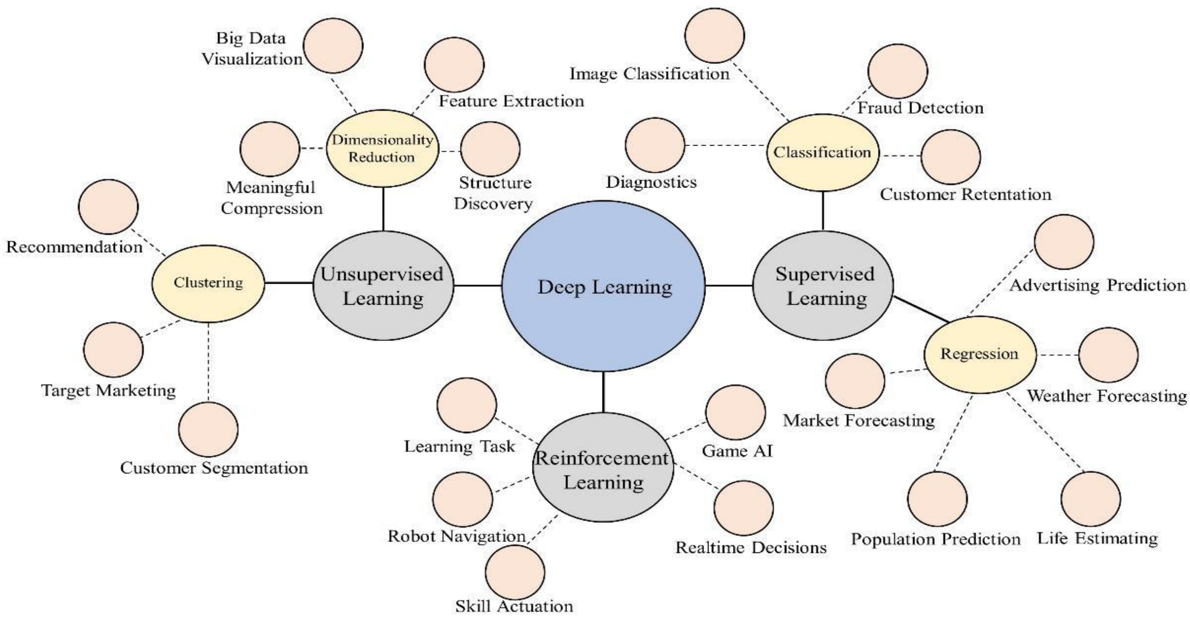

Figure 3.2 Applications and techniques of deep learning

3.2. CNN Architectures

A well-known deep-learning architecture, called the convolutional neural network (CNN), was modeled by the natural visual perception system of living things. Hubel & Wiesel (Hubel & Wiesel, 1968) discovered in 1959 that the cells of the visual cortex of animals are responsible for the perception of light in receptive fields. Kunihiko Fukushima proposed detection in 1980 (Fukushima & Miyake, n.d.), which can be considered the prototype of the CNN, and was motivated by this discovery. The current CNN system was introduced in 1990 in a seminal paper by Le-Cun et al. (Cun et al., n.d.) and was later further developed (Wei et al., 2019). LeNet-5, a multilayer artificial neural network they created, was able to classify handwritten digits. LeNet-5, like other neural networks, contains numerous layers and can be trained using the backpropagation technique. It can obtain accurate representations of the main image and allows recognition of visual patterns from the available raw pixels with little or no prior preparation.

CNN is a deep learning algorithm that can take an input image and assign weights to various features in the image. The preprocessing required in a CNN model is much less compared to other classification algorithms. While primitive methods design filters by hand with proper training, CNN can learn these filters/features. In addition, CNN can successfully capture spatial and temporal dependencies in an image by applying filters. The architecture is more suitable for the image dataset because the number of parameters involved can be reduced and the weights can be reused. Thus, the network can be trained to better understand the complexity of the image. The four components of CNN models are the convolutional layer, the pooling layer, the activation function, the fully connected layer, and the loss function.

Convolutional layers are the main component of CNN. In classification tasks, channels are treated as input to the convolutional layer, multiple 2D matrices are generated as output, and the input image is combined with these filters to create the output feature map.

Convolution is a linear operation used for feature extraction, where a number field called a kernel is applied to the input, a number field called a tensor. An element-by-element product between each element of the kernel and the input tensor is calculated at each position of the tensor and summed at the corresponding position of the output tensor to obtain the output value, which is called the feature map. Two hyperparameters that define the convolution operation are size and number of kernels. The former is usually 3×3, but sometimes 5×5 or 7×7. The second parameter is arbitrary and determines the depth of the feature maps output (Yamashita et al. 2018). This procedure is shown in Figure 3.3.

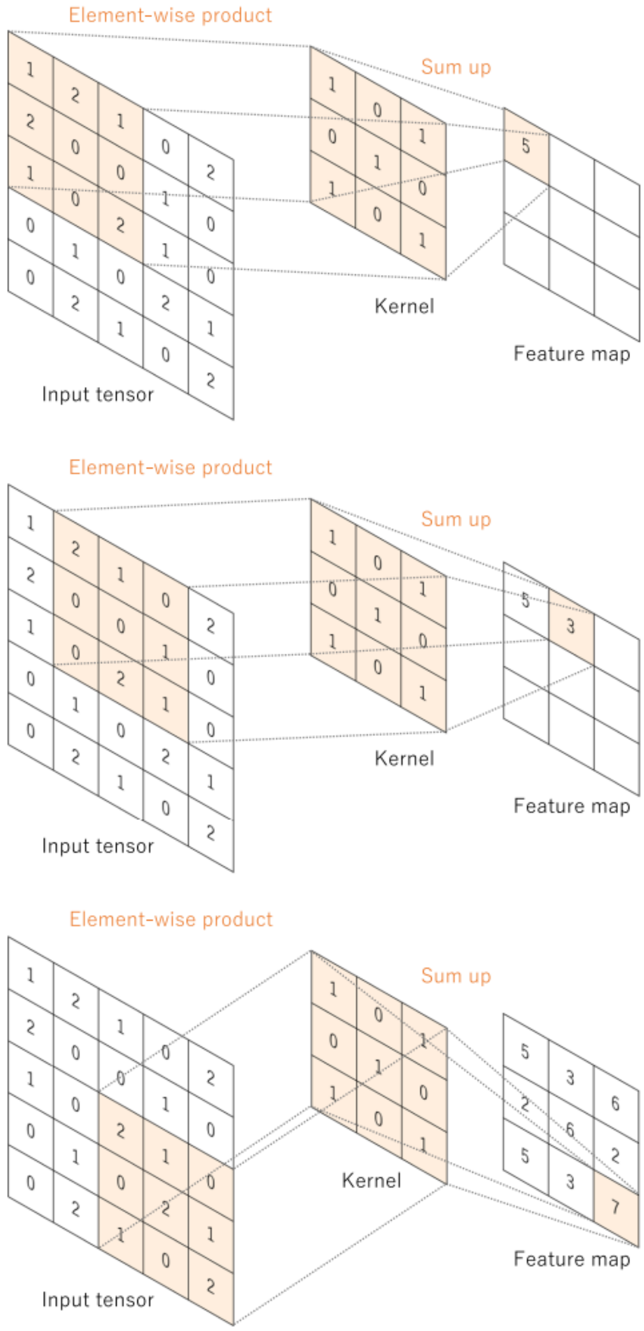

Figure 3.3 Convolution Operation with a Kernel Size (Yamashita et al. 2018)

The pooling layer plays an important role in the creation of feature maps. This layer adjusts large feature maps to create smaller feature maps. At the same time, the most important features are preserved at each step of the pooling stage. As with the convolution operation, the size of the stage and kernel is set before the pooling operation. Max, Min, and GAP are the most used pooling methods. Max and average pooling methods shown in Figure 3.4

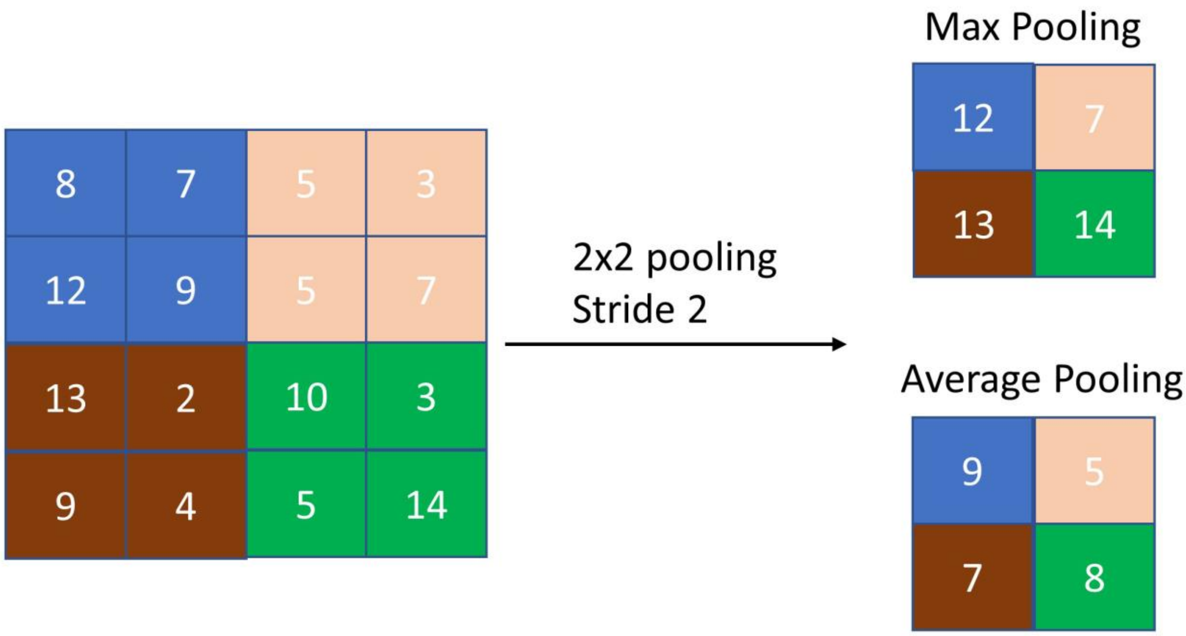

Figure 3.4 Max and Average Pooling

The (non-linear) activation function that maps input to output is the primary function of activation functions in CNN models. The main input value is defined by calculating the standard deviation of the weighted sum of all inputs. This calculation shows whether the activation function produces the appropriate output and decides whether the neuron will respond to a particular input.

The Fully Connected Layer is found at the end of the CNN architecture. This layer is connected to all neurons of the previous layer. It also follows the traditional multilayer neural perceptron method, as it is a feedforward ANN.

In the output layer, loss functions are used to calculate the estimated error that occurred during training. This error indicates the difference between the actual output and the predicted output. The loss function uses two parameters to calculate the error: the output predicted by the CNN and the actual output. There are several different error functions. The most used are Cross-Entropy, Softmax Loss Function and Euclidean Loss Function.

Since 2006, various models have been developed to circumvent the challenges associated with training deep CNNs. Most importantly, Krizhevsky et al. introduced a traditional CNN design and demonstrated significant advances over previous approaches to the image classification challenge. LeNet-5 and AlexNet-5 have similar overall architecture, but AlexNet has a deeper structure (Krizhevsky et al., 2012b). There are numerous variants of CNN architectures in the literature whose primary components are very similar. Since AlexNet in 2012, CNN has undergone many evolutions.

3.2.1. AlexNet

According to the first CNN-based model AlexNet, each subsequent architecture uses more layers in its network to reduce the error rate. This is effective with a smaller number of layers; when more layers are added, a significant DL problem known as the "vanishing/exploding gradient" occurs. That causes the gradient to become zero or too large, and the error rate in training and testing increases as the number of layers grows (M. Liu et al., 2021).

3.2.2. ResNets

According to the first CNN-based model AlexNet, each subsequent architecture uses more layers in its network to reduce the error rate. This is effective with a smaller number of layers; when more layers are added, a significant DL problem known as the "vanishing/exploding gradient" occurs. That causes the gradient to become zero or too large, and the error rate in training and testing increases as the number of layers grows (M. Liu et al., 2021). The ResNet architecture introduced the concept of residual blocks to solve the gradient problem. In this architecture, a technique called skip connections was proposed. Skip connections connect the activations of the first layer to other layers by skipping some intervening layers. ResNets are formed by stacking residual blocks, such as ResNet18, ResNet34, ResNet50, ResNet101, ResNet110, ResNet152, ResNet164, ResNet1202 (Mehmood et al., 2022). The winner of ILSVRC 2015, one of the most widely used architectures, is ResNets. In the ResNet family architecture, a residual block, a net within a network, is used. The architecture is defined in five steps with identity and convolutional blocks, and the input size is 224 x 224 (Najafabadi et al., 2015).

3.2.3. VGG16-19

Another model is the VGG architecture, which has been presented for use in image recognition software. Layers 16 and 19 are used by weighting them into VGG16 and VGG19 with a 3 x 3 convolutional filter size. The input image is 224 x 224 (W. Wang et al., 2019).

3.2.4. DenseNet

To solve the vanishing gradient problem, DenseNet connects each layer to the next layer of the network via a feedforward mechanism. In this way, the feature maps of each previous layer are sequentially transferred to all subsequent layers. Moreover, DenseNet combines the features of the previous layers instead of adding them to the model. However, DenseNet is parametrically overestimated due to the tight layer structure in addition to the increasing number.

3.2.5. Inception

ILSVR (ImageNet) participants in 2014 included the Inception model used for image classification. Participants use multiple filter sizes simultaneously for the input image rather than adding more layers to the model to make it deeper. The next inception block gets the chain of the inception block (Najafabadi et al., 2015). There are several variants of the Inception model, including InceptionV1 (Sam et al., 2019), InceptionV2 (Rahmaniar & Hernawan, 2021) and InceptionV3 (Cui et al., 2019), InceptionV4 (F. Chen et al., 2022) and InceptionResNet (Bharati & Pramanik, 2020). Each version of Inception is an iterative upgrade of the previous version. Understanding the upgrades can help develop optimized classifiers for speed and accuracy (Morid et al., 2021).

3.2.6. MobileNets

MobileNets are small architectures that can be used on both embedded and mobile devices. They have separable convolutional layers, and 2D convolutional layers are used in this architecture. This is associated with reducing the number of parameters, computations, training time, and memory requirements (Y. Li & Xia, 2020). In addition, MobileNets have 54 layers, and the input image size is 224 × 224. Capsule Networks can retrieve spatial information and other important features to avoid data loss during pooling operations. The architecture consists of six layers. The first three layers are called encoders, whose task is to convert the input image into vector form, and the last three layers are called decoders, which recover the image (Kalyani et al., 2021).

Examining the properties of these architectures, such as depth, robustness, and input size, is critical to selecting the right architecture for the task at hand. Table 3.2 provides a summary of the CNN architectures.

Table 3.2 Summary of CNN Models

Model	**Finding**	**Depth**	**Source**	**Input Size**
AlexNet	Relu and Utilize Dropout	8	ImageNet	227x227
CapsuleNet	Pays attention to special relationships between features	3	MNIST	28x28
DenseNet	Block of layers	201	CIFAR10, CIFAR 100, ImageNet	224x224
HRNetV2	High resolution representations	-	ImageNet	224x224
InceptionV3	Utilize small filter size	48	ImageNet	229x229
InceptionV4	Transform information concepts	70	ImageNet	229x229
MobileNetV2		53	ImageNet	224x224
ResNet152	Robust against overfitting due to symmetry mapping	152	ImageNet	224x224
Xception	A Depthwise convolution	71	ImageNet	229x229

VGG16	Increased depth, small filter size	16	ImageNet	224x224
VGG19	Increased depth, small filter size	19	Imagenet	224x224

3.3. Datasets

The biggest challenge in training and validating the proposed model is publically available labeled dataset deficiency. Most studies used small datasets in literature (Shorfuzzaman & Masud,2020), (Kaur et al., 2021), (Rehman et al., 2020), (Bukhari et al., 2020), which does not guarantee that the proposed model is fully trained. Thus, to increase the proposed model's generalization ability and create a more robust system, we wanted to work with a larger database. Most of the public COVID-19 datasets have two classes. We combined the two publicly available datasets to increase the number of images and add pneumonia as the third class in this work. The first dataset, the Chest X-Ray dataset, contains images in two-class as pneumonia and normal and consists of 5863 X-ray images (Covid Chest X-Ray Dataset, Howpublished = "Https://Github.Com/Ieee8023/Covid-Chestxray-Dataset", Note = Accessed: 2022-8-14, n.d.). The second dataset, the COVID-19 Radiography Database, includes normal, viral pneumonia, and COVID-19 images. The dataset consists of 1341 normal, 219 COVID-19, and 1345 viral pneumonia chest X-ray images (Covid-19RadiographyDatabase, 2022b). Some example images from these datasets are given in Figure 3.5.

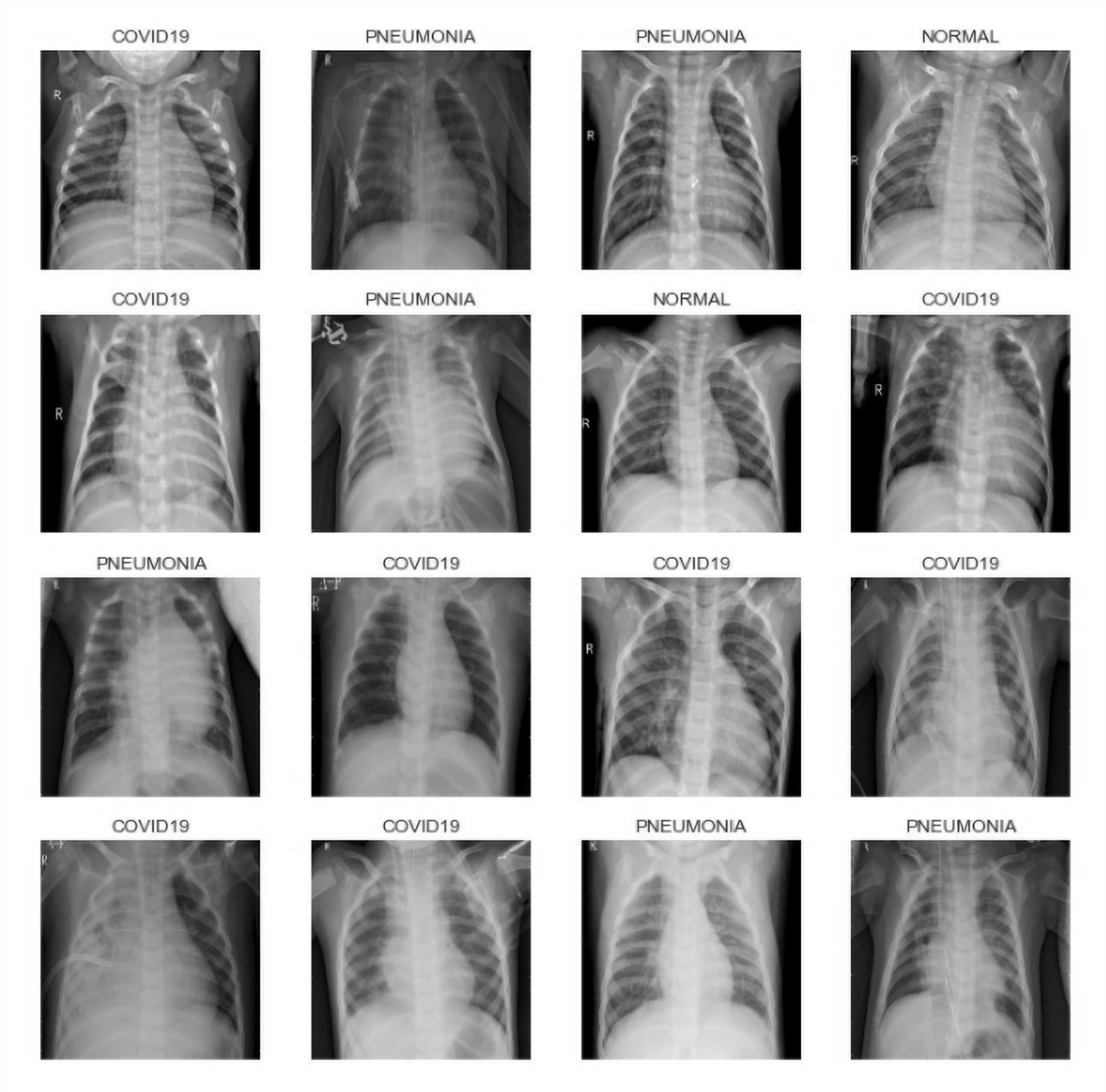

Figure 3.5 Sample Images

3.4. Data Augmentation and Pre-Processing

Deep CNNs have performed notably well on many image classification tasks; however, these models depend on big data to avoid over-fitting (Shorten & Khoshgoftaar, 2019). We need a large dataset to develop a robust and successful deep learning classification model, and this may not always be possible. Data augmentation methods, a suite of techniques that increase the size of training datasets are used. It also provides the diversity of classes in the dataset. Data augmentation was applied to the proposed model to increase data volume, avoid over-fitting, and create a more powerful model. These methods create new images by interacting with different variations of the visual properties of images that can significantly change them, such as image rotation, random horizontal reflection, and zooming.

The dataset contains chest X-ray images and some of them are 1007x1024 pixels in size. We need to resize the images to fit the input shape of the tested models. The input shape of the MobilenetV2 is 299x299 pixels. In this study, we only used resizing operator as pre-processing.

3.5. MobileNetV2 CNN Model

MobileNetV2 is a CNN architecture that aims to perform well on mobile devices. It is based on a different structure from other CNN structures where its connections are between the bottleneck layers. In addition, the intermediate expansion layer uses deep folds to filter out non-linear features. The MobileNetV2 architecture includes 32 layers of initial convolution followed by 19 bottleneck layers. The design details of the MobileNetV2 model are shown in Figure 3.6. In this study, we propose a MobileNetV2 design that uses two novel fine-tuning approaches to identify COVID-19 X-ray images (Sandler et al., 2018).

MobileNetV2 design has some advantages over the other deep learning designs. For small datasets, it hard to train the system and the visual classification task becomes prone to over-fitting. The MobileNetV2 architecture hinders this effect, preventing over-fitting, and it is a fast and successful architecture that optimizes memory consumption with a low margin of error. Moreover, MobileNetV2 design provides fast execution of transactions while making experimentation and parameter optimization.

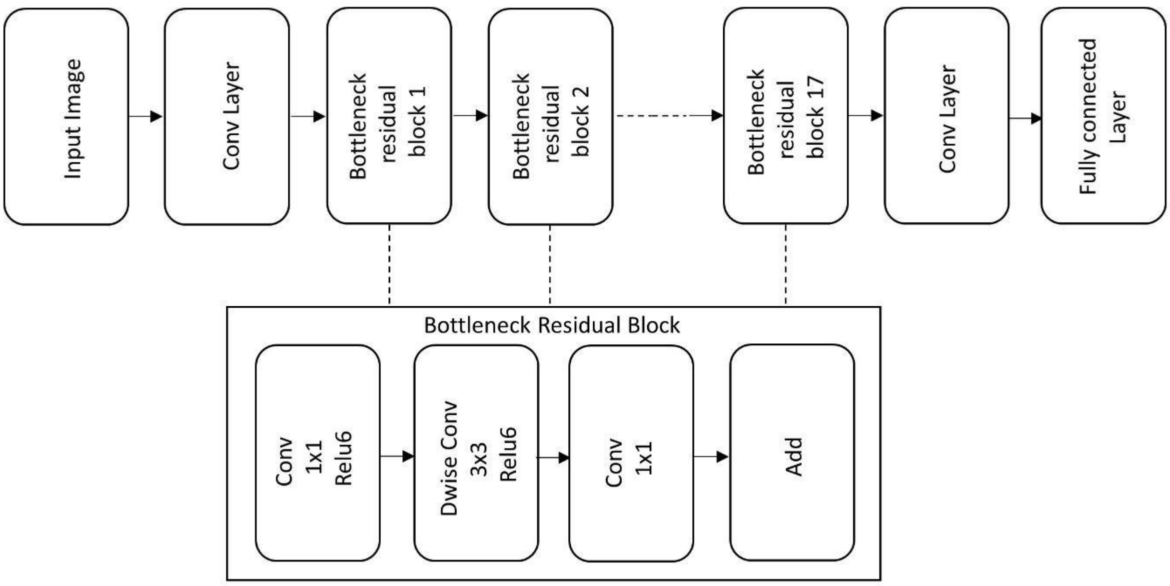

Figure 3.6 MobileNetV2 Architecture

3.6. Transfer Learning

Training a CNN model from scratch is an excessively time-consuming and computationally expensive process (Alzubaidi et al., 2021). Therefore, the common approach is to use pre-trained CNN models on datasets. Transfer learning is applying the gained knowledge. If training examples are insufficient in a classification problem, pre-trained examples of transfer learning are used (Marcelino, 2018). There are two methods of applying a pre-trained model in transfer learning. The first method is to use the pre-trained CNN model as a feature extractor and for classification; The last fully connected layer(s) is changed according to the number of classes in the data set. In the other method, fine-tune the pre-trained CNN model and its retraining of all or part of the layers with specific method (Pan & Yang, 2010). As a result, a changed architectural design is employed for the new classification task.

3.7. Fine Tuning

Fine-tuning, a transfer learning technique, focuses on saving the information obtained while solving a classification task and applying this knowledge to different approaches to a related problem (West et al., 2007). The major difference between fine-tuning and transfer learning is that only the weights of the newly added classifier layers are optimized in transfer learning, while the whole model is optimized in fine-tuning (Chollet, 2017).

In fine-tuning, we unfreeze some of the top layers of a frozen convolutional base model and simultaneously train the last layers of the convolutional base model and the newly added classification layer. Fine-tuning allows the higher order feature representations in the convolutional base model to become more relevant to the task at hand. In most convolutional networks, the final layers are more specialized than the initial layers. As you go deeper, the features of the pre-trained model become more specific to the pre-trained dataset. Fine-tuning aims to adapt these specific features to the new data, rather than overwriting the generic knowledge to fit the new classification task (Chollet, 2017). Thus, it is crucial to open how many of the last layers and to train them how many epochs. In this study, we proposed one classical and 2 novel fine-tuning mechanisms to train the model.

3.8. Proposed Fine-Tuned MobileNetV2 Model

In our study, we implemented the MobileNetV2 network to classify COVID-19 X-ray images. There are two kinds of blocks in MobileNetV2. One is the residual block with stride 1 and the other block is non-residual block with stride 2, which is used for downsizing (Sandler et al., 2018). Details of MobileNetV2 model are given as in Figure 3.7.

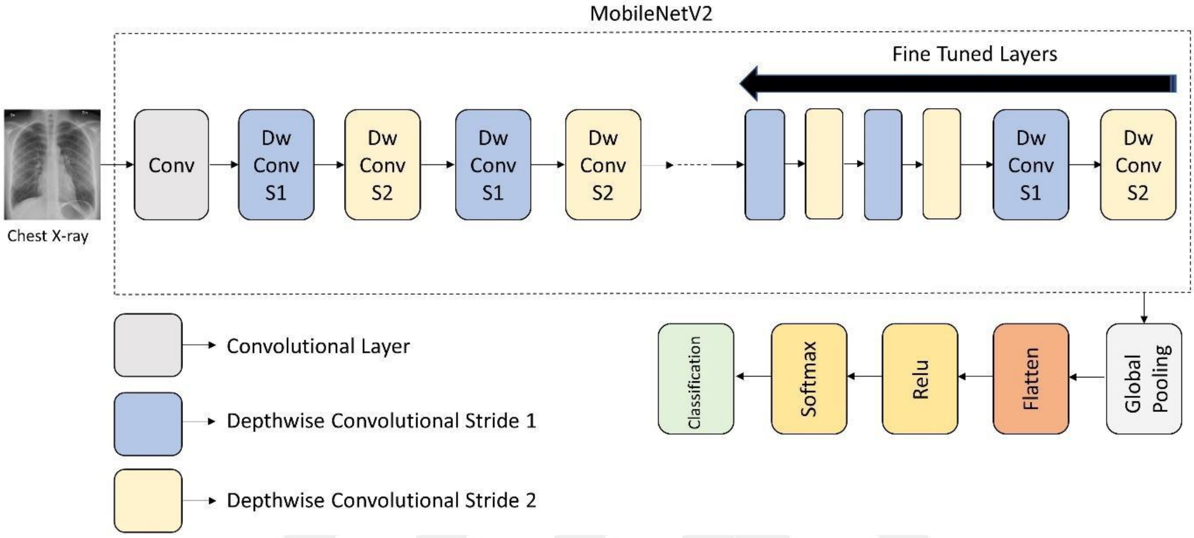

Figure 3.7 Fine-tuned MobileNetV2 Model

In our approach, we trained the whole model for 50 epochs before fine-tuning. In the first fine-tune approach, we opened the last 50 layers of the convolutional base model and created new training loops to train the whole model for 80 epochs, as seen in Figure 5 (yellow bars). In the second fine-tune approach, we started to open the last layers of the convolutional base model using a step function. In this approach, we have reduced the number of layers opened from the last 50 layers by five for every 8 cycles, as seen in Figure 3.8 (green bars).

In our last approach, unlike the other two approaches, we determined the number of epochs and which layers to be opened according to the result using a pre-defined exponential equation instead of performing the fine-tuning process in a certain decreasing or increasing order.
We used the following equation 1 to calculate these values:

$$y = \frac{e^{\frac{-2x}{8}} + 9}{2} + 1 \qquad \textbf{(1)}$$

where x is between 1 and 50 corresponding to defining unfrozen top layers of the base model and y is the calculated values for fine-tuning to define the number of epochs. This equation was determined by empirical observation. Therefore, another exponential decreasing equation can be used for that purpose.

According to this equation, the entire system was trained for 80 epochs in the last layers and decreased exponentially to 1 epoch in the 102nd layer, as shown in Figure 3.8 (blue bars). Thus, our model can preserve more pre-trained generic information. The last layers of CNN models may have specific learned properties and the first layers of CNN models often learn more about generic properties, such as edges, shapes, and textures. This equation provides us with an exponentially decreasing number of training cycles from the last layer to a given depth during the training process. As can be seen in Figure 3.8, in our three approaches, we fine-tuned the models with a total of 4000, 2200, and 723 training cycles for models 1, 2, and 3, respectively. We used the following training parameters: 0.0001 as the learning rate and Adam as the optimizer.

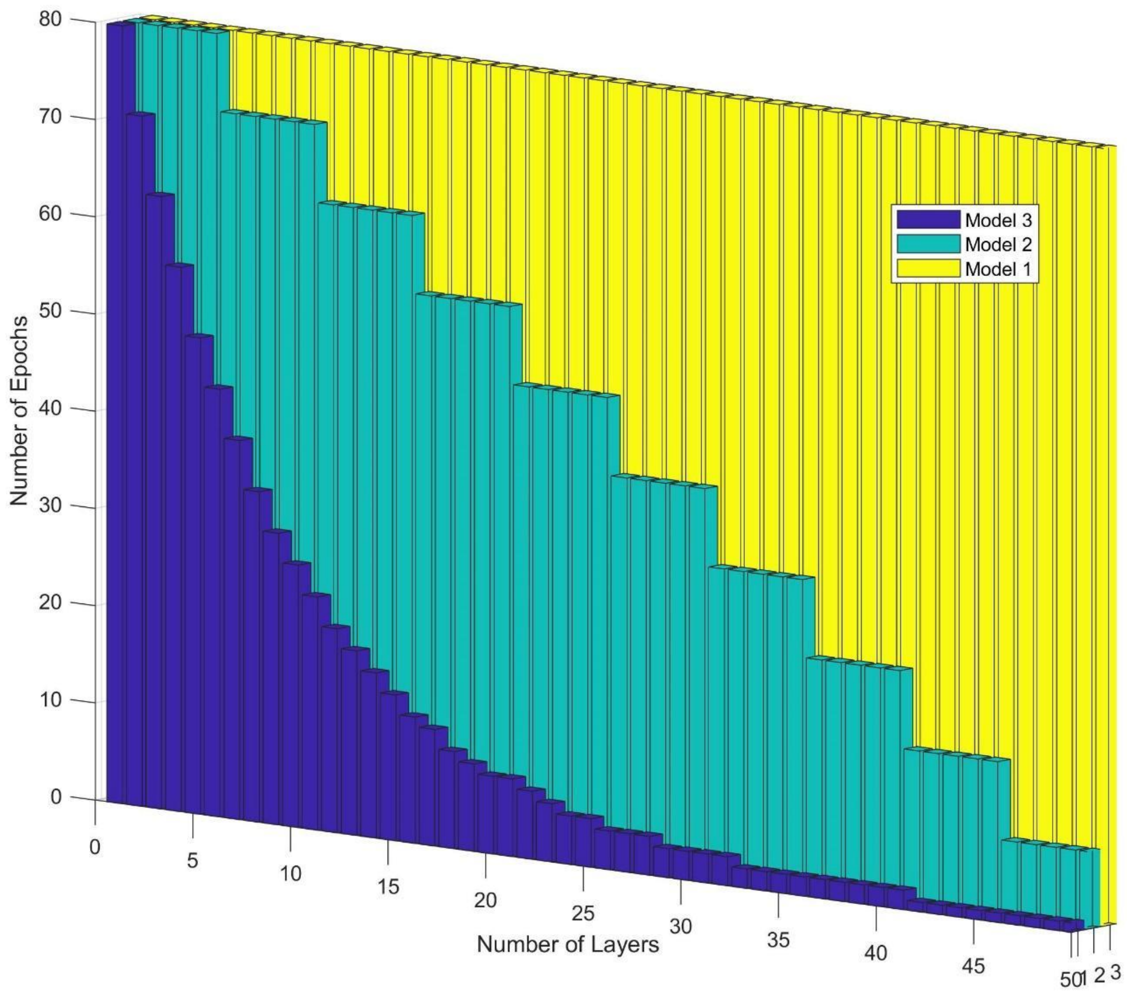

Figure 3.8 Fine Tuning Number of Epochs and Layers

3.9. Evalutaion Metrics

We used the following evaluation metrics: accuracy, recall, precision, and f1-score as in Equations 2-5.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$f1 - Score = \frac{2(Precision * Recall)}{Precision + Recall} \quad (5)$$

Table 3.3 Model Parameters

Parameter	Value
Learning Rate	0.0001
Optimizer	Adam
Epochs	50+80
Batch Size	32

Table 3.4 Models Accuracy and Loss Values

Model	Pre-Fine-tune Acc.	Loss	After Fine Tune Acc.	Loss
Model 1	0.9409	0.1223	0.9588	0.0969
Model 2	0.9429	0.1335	0.9739	0.0675
Model 3	0.9425	0.1268	0.9781	0.0417

where True Positive TP is the case where the output of the algorithm outputs YES when the actual state is YES, False Positive FP is the case where the output of the algorithm outputs NO when the actual state is YES, True Negative TN is the case where the output of the algorithm outputs NO when the actual state is NO, False Negative FN is the case where the output of the algorithm YES when the actual state is NO.

4. RESULTS

In this paper, we implemented 3 different fine-tuning approaches for classifications of 3 different classes, namely, normal, COVID-19, and viral pneumonia, on X-ray images. We combined two datasets containing 9457 images. We evaluated our model using two standard evaluation methods, separating the data 30-70% randomly test-train splits and 5-fold cross-validation to validate our models. Experimental tests were conducted on a PC with Intel I7 6700hq 2.60GHz CPU, NVIDIA GTX970M having 3GB GPU, and 16GB RAM. The proposed approach was implemented using the Keras-Tensorflow library. We evaluated our model using performance metrics given in Eqs. 2-5. Table 4.1 shows pre-fine-tuning and after fine-tuning accuracy results of three models. Models 1, 2, and 3 demonstrate proposed classical, step, and exponential fine-tuning models, respectively. We used 50 epochs training to calculate pre-fine-tuning metrics and 80 epochs training to calculate after fine-tuning metrics. The number of epochs and learning rate were determined empirically. Previous studies showed that using Adam as an optimizer achieved promising results (Bera & Shrivastava, 2020). Thus, Adam was chosen as the optimizer. In addition, we defined the batch size as 32 and the learning rate as 0.0001. All parameters of the model are summarized in Table 3.3.
As can be seen from the Table 3.4 that before fine-tuning models gave about 94% accuracy rates. Although these models are the same in pre-tuning stage, the results are slightly different because of the random separation of datasets and the small number of training epochs. Results after the fine-tuning show that proposed fine-tuning mechanisms achieved better accuracies than classical fine-tuning mechanism. Furthermore, these models fine-tuned at 80, 44, and 14.46 epochs on average for models 1, 2, and 3, respectively. Model 3 decreased 81.92% of total fine-tuning epochs and still achieved better results.

The detailed classification results achieved from the three models are given in Table 4.1 in terms of evaluation metrics using 30-70% split of data. The results show that proposed Models 2-3 give similar classification rates. Model 3 attained slightly better results and achieved accuracy, recall, precision, and f1-scores of 97.78%, 97.77%, 97.77%, and 98.64%, respectively. Model 1 attained the lowest scores.

Table 4.1 Numerical results for proposed models for 30-70 split

Model	Accuracy	Recall	Precision	F1 Score
Model 1	0.9588	0.9587	0.9585	0.9588
Model 2	0.9739	0.9739	0.9737	0.9737
Model 3	0.9778	0.9777	0.9777	0.9777

Table 4.2 Numerical results for Model 1 using five-fold cross-validation

K-Fold	Accuracy	Recall	Precision	F1 Score
Fold 1	0.9571	0.9571	0.9568	0.9569
Fold 2	0.9582	0.9582	0.9582	0.9582
Fold 3	0.9587	0.9587	0.9582	0.9583
Fold 4	0.9557	0.9577	0.9590	0.9753
Fold 5	0.9513	0.9513	0.9511	0.9511
Mean	0.9562	0.9566	0.9558	0.9559

Table 4.3 Numerical results for Model 2 using five-fold cross-validation

K-Fold	Accuracy	Recall	Precision	F1 Score
Fold 1	0.9635	0.9635	0.9634	0.9634
Fold 2	0.9582	0.9582	0.9579	0.9580
Fold 3	0.9603	0.9603	0.9599	0.9599
Fold 4	0.9640	0.9640	0.9636	0.9638
Fold 5	0.9592	0.9592	0.9592	0.9592
Mean	0.9610	0.9610	0.9608	0.9608

Table 4.4 Numerical results for Model 3 using five-fold cross-validation

K-Fold	Accuracy	Recall	Precision	F1 Score
Fold 1	0.9756	0.9656	0.9656	0.9656
Fold 2	0.9725	0.9725	0.9725	0.9725
Fold 3	0.9793	0.9793	0.9792	0.9792
Fold 4	0.9777	0.9777	0.9776	0.9776
Fold 5	0.9751	0.9751	0.9750	0.9751
Mean	0.9761	0.9761	0.9759	0.9760

We used five-fold cross-validation technique to evaluate the proposed model and to compare the results to the results of other studies. The results of cross-validation of each model are shown in Table 4.2, 4.3 and, 4.4. As can be seen from the tables that, Model 3 performed better results and achieved five-fold mean accuracy, recall, precision, and f1-scores of 97.61%, 97.60%, 97.59%, and 97.60%, respectively.

Figure 4.1 presents the confusion matrices for the Models 1-3 for 30-70% split of data. From Figure 4.1-a, we can note that while 1094 COVID-19, 432 normal, and 1194 pneumonia samples were accurately classified, 3 COVID-19, 61 normal, and 53 pneumonia samples were misclassified using Model 1. Thus, the correct classification rates of COVID-19 samples were 99.72%, while 87.62% for normal and 94.23% pneumonia samples. As can be seen in Figure 10-b, 1095 COVID-19, 447 normal, and 1221 pneumonia samples were classified correctly, and 2 COVID-19, 46 normal, and 26 pneumonia cases were misclassified using Model 2. As a result, 99.81%, 90.66%, and 97.91% correct classification rates of COVID-19, normal, and pneumonia were achieved, respectively. It can also be noted in Figure 4.1-c that 1095 COVID-19, 461 normal, and 1218 pneumonia samples were classified correctly, and 2 COVID-19, 32 normal, and 29 pneumonia cases were misclassified using Model 3. Therefore, the correct classification rates of COVID-19, normal, and pneumonia were 99.81%, 93.50%, and 97.67%, respectively.

Figure 4.2 presents the confusion matrices of five-fold mean values for the models 1-3. From Figure 4.2-a, we can note that while 717.4 COVID-19, 275 normal, and 816.2 pneumonia samples were accurately classified, 5.8 COVID-19, 40.2 normal, and 36.8 pneumonia samples were misclassified using Model 1. Thus, the correct classification rates of COVID-19 samples were 99.72%, while 87.62% for normal and 94.23% pneumonia samples. As can be seen in Figure 4.2-b, 718 COVID-19, 278.6 normal, and 821.2 pneumonia samples were classified correctly, and 5.2 COVID-19, 37.6 normal, and 31.8 pneumonia cases were misclassified using Model 2. As a result, 99.28%, 88.10%, and 96.27% correct classification rates of COVID-19, normal, and pneumonia were achieved, respectively. It can also be noted in Figure 4.2-c that 832.8 COVID-19, 292.2 normal, and 832.8 pneumonia samples were classified correctly, and 5.8 COVID-19, 23 normal, and 20.2 pneumonia cases were misclassified using Model 3. Therefore, the correct classification rates of COVID-19, normal, and pneumonia were 99.20%, 92.76%, and 97.63%, respectively.

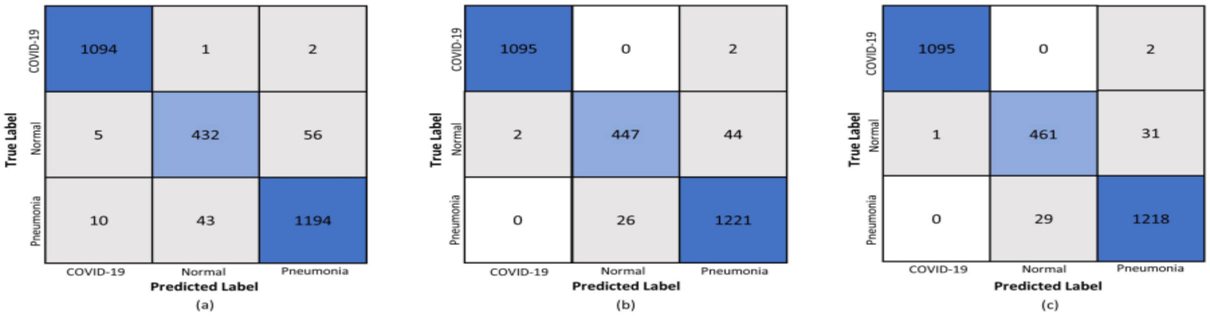

Figure 4.1 Confusion Matrices of Models 1-3

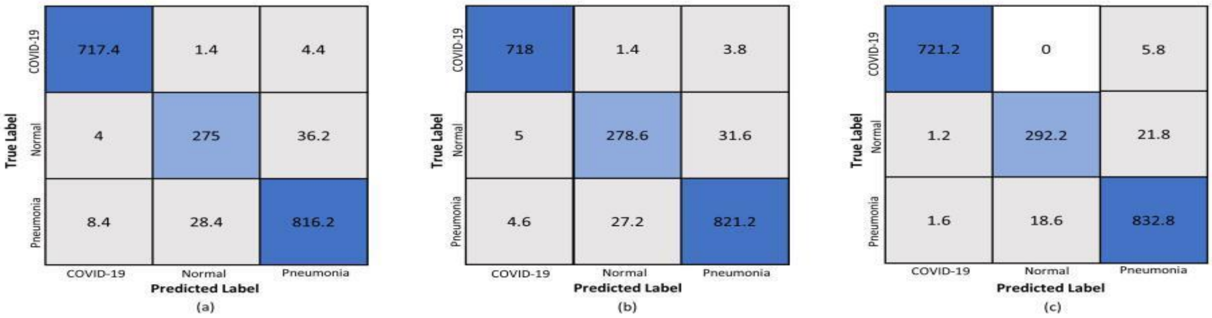

Figure 4.2 Five-Fold Mean Confusion Matrices of Models 1-3

In addition to these evaluation metrics, the Friedman test, a popular non-parametric statistical test used to rank the algorithms based on their results without specifying any statistical difference, was performed for 5-fold cross-validation results. We achieved chi-square test statistics of 8.0 and a p-value of 0.0183. Friedman test results show that the p-value is less than 0.05. Thus, there are significant differences in the methods. We also employed Nemenyi post-hoc test, used to determine the statistical difference with method pairs, to evaluate the performance of proposed methods. Table 4.5 shows the analysis of the Nemenyi tests on the methods using a significance level of 5%. It can be seen from the table that the results of Model 1 and Model 3 have significant differences, and Model 3 achieves better results.

Table 4.5 Statistical Test Results

Model	**1**	**2**	**3**
1	1.0000	0.3341	0.0129
2	0.3341	1.0000	0.3341
3	0.0129	0.3341	1.0000

Figures 4.3, 4.4, and 4.5 demonstrate the training process of fine-tuned models 1-3 using 30-70% data split, respectively. The green vertical line in the graphs shows the 50th epoch where fine-tuning started. We previously stated that we trained the classification layer added to the base model for 50 epochs and fine-tuned the whole model for additional 80 epochs in total 130 epochs. As shown from Figure 4.4, validation accuracy flattened about 94% before fine-tuning, and after fine-tuning, it increased to about 95%.

Colored vertical lines in Figure 4.4 show stages of step fine-tuning. It is clearly seen that fine-tuning using model 2 increased validation accuracy from 94% to about 97%. Beginning of the fine-tuning, validation accuracy fluctuated strongly. In later epochs, it settled down to better rates. It was also observed in Figure 4.5 that after fine-tuning, validation accuracy was increased sharply in early epochs.

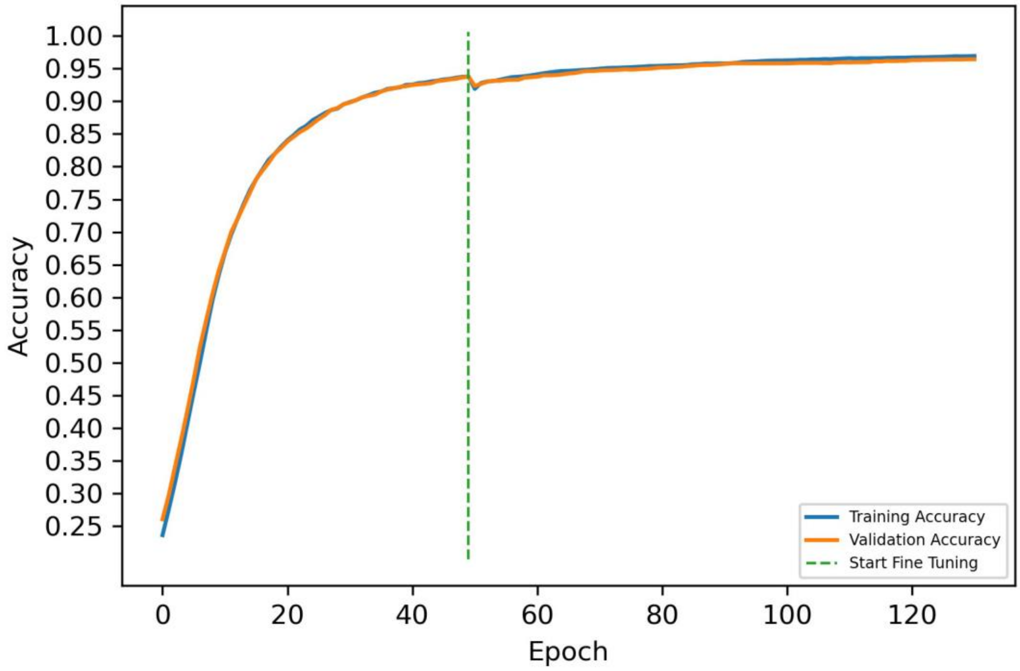

Figure 4.3 The Training Process of Model 1

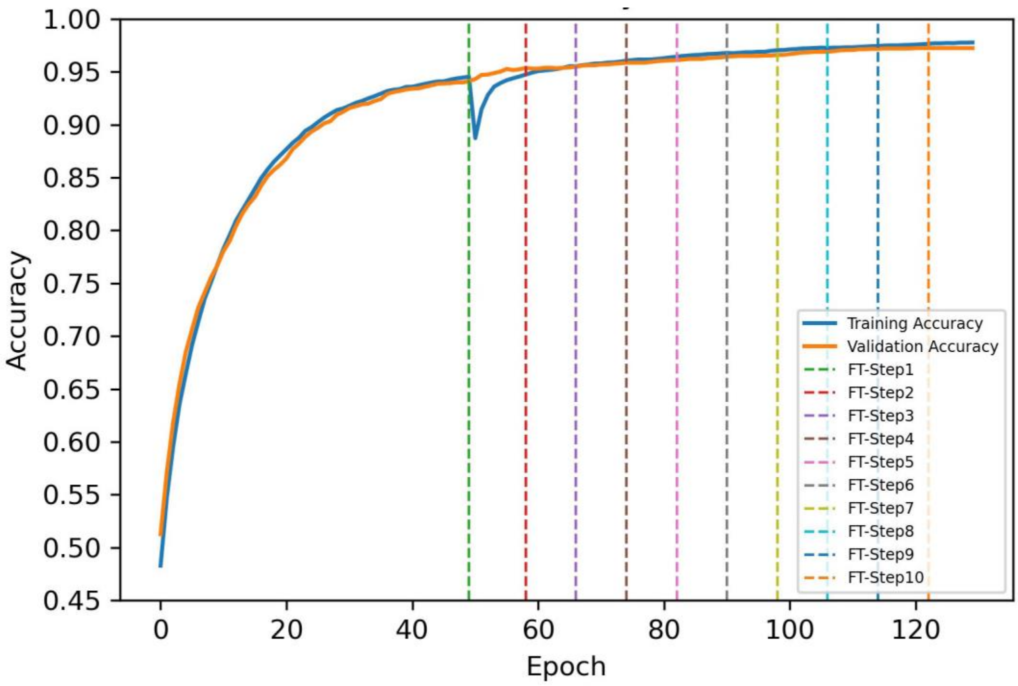

Figure 4.4 The Training Process of Model 2

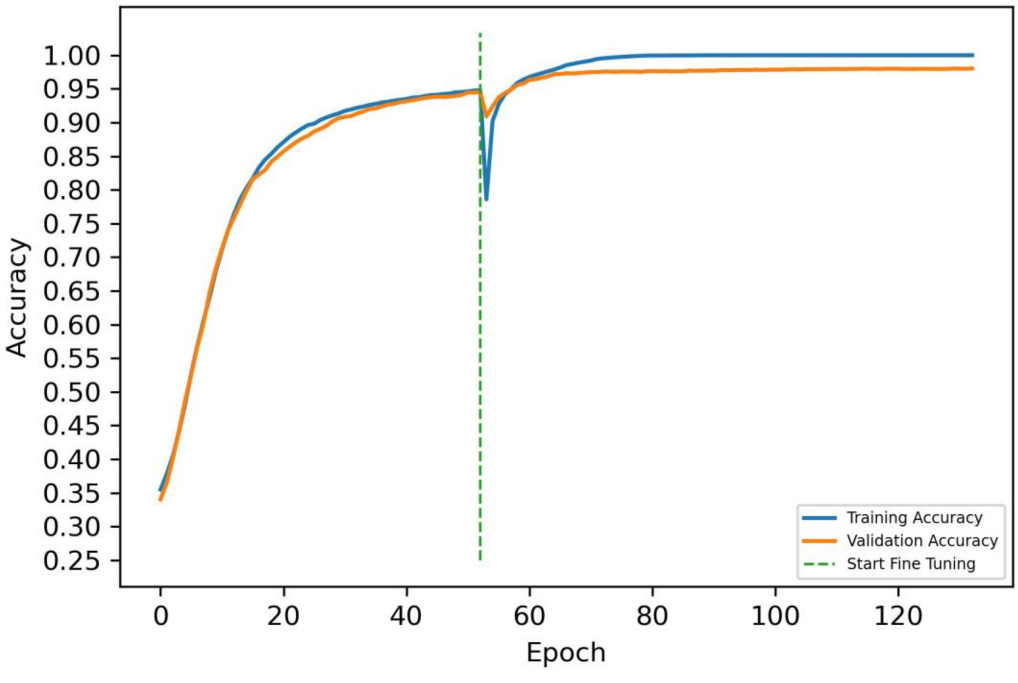

Figure 4.5 The Training Process of Model 3

5. DISCUSSIONS

This study aims to develop a CNN-based COVID-19 diagnostic model using new fine-tuning mechanisms. It also aims to bring new approaches to the literature by incorporating different proposals into the methodological design. Most recent studies on this topic using deep learning methods are summarized in Table 5.1 in terms of accuracy metrics.

As can be seen in Table 5.1, many studies have been conducted so far that also incorporate the CNN architecture. The major advantage of the CNN architecture is that it enables end-to-end learning. Three types of models are distinguished in the literature. These are deep learning models that are trained from scratch, transfer learning approaches that use pre-trained models, fine tuning, and hybrid approaches that combine deep learning models with classical machine learning methods.

Table 5.1 Comparison of the Models

Study	**Methodology**	**Accuracy (%)**
Rehman et al.	ResNet101, MobileNet	98.75
Shorfuzzaman and Masud	VGG16, ResNet50, Xception, MobileNet and DenseNet121	99.26
Degadwala et al.	Fine-tuned CNN	90.70
Minaee et al.	ResNet, SqueezeNet, and DenseNet121	98.00
Iskenderani et al.	DenseNet	96.25
Pathak et al.	ResNet32	96.22
Apostopoulos and Mpesiana	MobileNetV2	96.78
Kaur et al.	AlexNet	99.52
Wang et al.	ResNet101 and ResNet152	96.10
Misra et al	ResNet18	93.90
Punn and Agarwal	ResNet, InceptionV3, and NASNetLarge	97.00
Ucar and Korkmaz	Bayes-SqueezeNet	98.30
Narin et al.	ResNet and Inception	96.10

Lee et al.	VGG16	95.00
Ismael and Şengül	ResNet50	92.60
Wang et al.	COVID-Net	94.00
Nishio et al.	VGG16	86.30
Monshi et al.	COVIDX-rayNet	95.82
Das et al.	VGG16	97.67
Rahaman et al.	VGG19	89.30
Moujahid et al.	VGG19	96.97
Khan et al.	CoroNetV2	95.00
Ohata et al.	DenseNet201	95.60
Manokaran et al.	DenseNet201	92.19
Garg et al.	DenseNet121	94.00
Naronglerdrit et al.	MobileNet	96.76
Xu et al.	ResNet	86.70
Ozturk et al.	DarkCovidNet	87.02
Asif and Wenhui	InceptionV3	96.00
Luz et al.	EfficientNet	93.90
Bargshady et al.	CycleGAN	94.20
Rahimzadeh et al.	Xception+ResNet50V2	91.40
Model1	MobileNetV2+Classical FT	95.49
Model2	MobileNetV2+Step FT	97.56
Model3	MobileNetV2+Exponential FT	97.66

In table 5.1, some studies achieved high success rates in classification using a pre-trained CNN model (Iskanderani et al., 2021), (Pathak et al., 2020), (Kaur et al., 2021), (Misra et al., 2020), (Apostolopoulos & Mpesiana, 2020b), (Luz et al., 2021), (Bukhari et al., 2020), (Lee et al., 2020), (Ismael & Şengür, 2021), (Nishio et al., 2020), (Das et al., 2021), (Rahaman et al., 2020), (Moujahid et al., 2020), (Ohata et al., 2020), (Manokaran et al., 2021), (Garg et al., 2020), (Naronglerdrit et al., 2021), (Asif & Wenhui, 2020).

It is seen that different pre-trained CNN model designs attained varied results on the same dataset (Rehman et al., 2020), (Shorfuzzaman & Masud, 2020), (Minaee et al., 2020), (N. Wang et al., 2020), (Narin et al., 2021), (Punn & Agarwal, 2021), (Dansana et al., 2020), (Rahimzadeh & Attar, 2020) the authors used binary classification (COVID-19, normal) and achieved more successful results.

To summarize these results, Table 9 includes different studies using CNN models that give promising results. However, these studies use different CNN architectures, training samples, number of classes, and design parameters. Thus, it is unfair to compare these studies directly. It can be seen from the table that some studies with good results either have fewer training samples (Shorfuzzaman & Masud, 2020), (Kaur et al., 2021), (Rehman et al., 2020), (Mansour et al., 2021), (Bukhari et al., 2020) or have a binary classification (Minaee et al., 2020), (Rehman et al., 2020).

It is common knowledge that the model trained using a small amount of training data results in a poor approximation and the model trained using fewer number of classification classes achieve better results. In other words, if the model contains a small amount of data or fewer classification classes, which will result in an uncertain and high variance estimate compared to its actual performance (Barbedo, 2018).

6. CONCLUSION

This study proposes a new deep CNN model based on transfer learning that incorporates novel fine-tuning approaches for the detection of COVID-19. Our recommended fine-tuning methods are different from the classical fine-tuning approaches and enable the detection of COVID-19 using chest radiographs. The first proposed approach is the classical fine-tuning approach. In this approach, the last 50 slices of the CNN model are trained with 80 epochs while the remaining slices are frozen. In contrast to the classical fine-tuning approach, the second proposed approach trains the last 50 layers of the CNN model in descending order using a step function instead of training them all at once. In this way, the model achieved higher success than the first approach by minimizing data loss and increasing the number of transmitted features in the fine-tuning process. In the final fine-tuning approach, we trained the proposed model by determining the number of epochs and the layers to be opened using a given exponential function. The second and third proposed models achieved promising results, achieving a classification rate of 99.28% and 99.20%, respectively, for COVID-19 with five-fold cross-validation.

Moreover, Model 3 achieved this result with only 18.07% of 4000 total fine-tuning training procedures. Our study provides a robust and reliable model that can be used as part of a decision support system for the detection of COVID-19 disease. In this way, computer-aided diagnostic systems can be created, reducing the time required for radiographic evaluation by experts. The numerical results show that the training of the model's lower layers can generalize to the new problem and improve the classification results.

7. RECOMMENDATIONS

In future work, metaheuristics-based hyper-parameter tuning can be implemented to optimize the model and achieve better results. A lightweight deep learning model can be trained from scratch to classify COVID-19 cases and can be implemented on mobile devices for intelligent health applications.