**COVID-19 DETECTION USING**

**CONVOLUTIONAL NEURAL NETWORK**

**Review Paper**

### A Project Work

*Submitted in the partial fulfillment for the award of the degree of*

# BACHELOR OF ENGINEERING

### IN

### COMPUTER SCIENCE and ENGINEERING

### With Specialization in

### Artificial Intelligence and Machine Learning

### Submitted by:

### MANASIJ HALDAR

### 20BCS6838

### Under the Supervision of:

### PRABHJOT SINGH MANOCHA



# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING APEX INSTITUE OF TECHNOLOGY

### CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,

**PUNJAB**

#### MAY,2022

**DECLARATION**

I, **Manasij Haldar**, student of **Bachelor of Engineering in Computer Science and Engineering with Specialization in Artificial Intelligence and Machine Learning**, **session:2020-2024**, Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled ‘**Covid-19 Detection using Convolutional Neural Network’** is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

**Manasij Haldar**

**Candidate UID:20BCS6838**

**Date: 4/20/2022**

**Place: Chandigarh University, Mohali, Punjab, India**

## Abstract

*In early December 2019, A novel severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) was responsible for the Coronavirus Disease 2019 (COVID-19) outbreak in Wuhan, Hubei Province, China. On January 30, 2020, the World Health Organization designated the outbreak as a Public Health Emergency of International Concern. The emergence of COVID-19 in 2020 has been a historical moment. This virus has spread to many countries and rendered many people housebound. Many studies have attempted to examine the impact of this pandemic from various angles; however, this study will focus on how it has affected and may affect children aged 0 to 12 years in the future after schools have been closed for months. As of February 14, 2020, 49,053 laboratory-confirmed cases and 1,381 deaths had been reported worldwide. In response to the perceived risk of contracting disease, many countries have implemented a variety of control measures.*

*We conducted a literature review of publicly available information to summarize what we know about the virus and the current epidemic. This literature review covers the causal agent, pathogenesis, and immunological responses, epidemiology, diagnosis, treatment, and management of the disease, as well as control and prevention efforts.*

*The Coronavirus Disease 2019 (COVID-19) pandemic is still wreaking havoc on the global population's health and well-being. Effective screening of infected patients is a critical step in the fight against COVID-19, with radiology examination using chest radiography being one of the key screening approaches. Early studies discovered that patients with COVID-19 infection have abnormalities in chest radiography images. In this study, we introduce a deep convolutional neural network design tailored for the detection of COVID-19 cases from chest X-ray (CXR) images, motivated by this and inspired by the research community's open-source efforts. For the training, we used various open source CXR datasets available on the internet to create a larger and more efficient dataset that is now uploaded and available on Kaggle. We present an open access benchmark dataset that we created with around 10,000 CXR images and, to the best of the authors' knowledge, the largest number of publicly available COVID-19 positive cases.*

*In addition, we presented a comparison study of how different transfer learning models make predictions using an explainable method in an attempt to gain deeper insights into critical factors associated with COVID cases, which can aid clinicians in better screening, as well as to validate that it is making decisions based on relevant information from the CXR images. The goal of the study is to provide a better understanding for future COVID-19 or other imagery-based predictive decision-making tasks in terms of selecting the best transfer learning model to achieve the highest accuracy in the operations.*

**Keywords**

* 2019-nCoV
* COVID-19 Outbreak
* SARS-CoV-2
* Novel coronavirus
* Convolutional Neural Network
* Deep Learning
* Transfer Learning
* Python
* Kaggle
* Model Training
* Model Testing

# Table of Contents

|  |  |  |
| --- | --- | --- |
|  | Title Page  Declaration of the Student  Abstract  Keywords | i |
| ii |
| iii |
| iii |
|  |
|  |
|  |
| **1.** | **INTRODUCTION** |  |
|  | * 1. A Brief Background   2. Problem Definition   3. Project Overview   4. Software Specification   5. Data Specification |  |
| 1. **LITERATURE REVIEW** 2. **PROBLEM FORMULATION**  * Background * Method | |  |
|  |
|  |
| 1. **MODEL ARCHITCTURE** 2. **MODEL COMPARISON REVIEW**  * Inception Model * ResNet * DenseNet * VGG16  1. **SOURCE-CODE** 2. **RESULT** 3. **REFERENCES** | |

# INTRODUCTION

# 1.1 A Brief Background

Throughout the first months of 2020, the new SARS-CoV-2 coronavirus, which causes COVID-19, had the entire world on edge. It caused many nations' borders to close and millions of inhabitants to be confined to their homes due to sick persons, with 868,000 confirmed cases worldwide at this time (April 2020). In December of this year, the virus was discovered in China. Since March 2020, Europe has been the epicenter of the virus's spread.

China has managed to contain the virus over three months after the outbreak began in December 2019, with a total of 3,312 deaths and more than 81,000 afflicted persons. According to a report released in April 2020, Italy, which overtook the Asian countries in death toll in March 2020, has become the most impacted country, with more than 10,000 people killed. This number was steadily increasing. Various studies projected the expansion of infection curves based on various parameters such as the number of exposed, infected, and recovered humans. These investigations helped researchers to gain a better understanding of the transmission dynamics that may occur in each country.

The outbreak's cause has yet to be determined. The first cases were discovered in December of this year. Respiratory symptoms, fever, cough, dyspnea, and viral pneumonia are all clinical features of COVID-19. The fundamental issue with these symptoms is that there are virus-infected persons who are asymptomatic.

COVID-19 is detected through the collection of samples from the respiratory tract. When the case study is asymptomatic or symptoms are light, it is performed at home by a health care expert, or in a health facility or hospital if the patient is admitted for a serious ailment. In nations like Germany and South Korea, performing as many tests as possible has proven to be the most effective way to combat the virus. Because Spain was unable to conduct so many tests, it is critical to explore and create alternate techniques for doing these tests in a timely and efficient manner.

In the detection and follow-up of disease, AI and radiomics applied to X-Ray and Computed Tomography (CT) are useful tools. According to CT scans, significant ground grass opacity lesions in the peripheral and posterior lungs indicate COVID-19 pneumonia. As a result, once abnormalities in chest radiographs are suggestive of coronavirus, CT can play a crucial role in the diagnosis of COVID-19 as an advanced imaging evidence. AI algorithms and radiomics features generated from chest X-rays would be extremely useful in implementing large-scale screening programmes in any country with X-ray capability and would aid in the diagnosis of COVID-19.

The current circumstance necessitates the implementation of an automatic detection system as an alternative diagnosis option to stop COVID-19 from spreading. Several researchers have used machine learning to accomplish this goal, such as the Size Aware Random Forest approach (iSARF) proposed by, in which participants were divided into groups with varying sizes of infected lesions. Then, with each group, a random forest-based classifier was trained. Under five-fold cross-validation, their proposed technique generated an accuracy of 0.879, a sensitivity of 0.907, and a specificity of 0.833, according to the results.

In order to produce better outcomes than more typical machine learning approaches, deep learning techniques are also applied. Convolutional neural networks are one of the most widely utilized approaches in picture categorization (CNNs). This type of model has been used in several studies to detect COVID-19 in medical images, such as in, where the authors propose a CNN model trained with a randomly selected set of image regions of interest (ROIs), achieving an accuracy of 85.2 percent, a specificity of 0.83, and a sensitivity of 0.67. Another example of what may be accomplished with CNNs is offered by. They propose the COVID-Net CNN network, which achieves 92.4 percent accuracy, 80% sensitivity, and 88.9% specificity. COVIDX-Net is a method that combines seven distinct deep convolutional neural network designs, including a modified version of the Visual Geometry Group Network (VGG19) and the second edition of Google MobileNet. Each deep neural network model can categorize the patient's status as a negative or positive COVID-19 case by analyzing the normalized intensities of the X-ray image. For healthy and COVID-19 detection, their trials reach f1-scores of 0.89 and 0.91, respectively.

The findings in the cited papers suggest that deep learning techniques are beneficial for virus identification and that they improve the metrics acquired using more typical machine learning methods

Our paper's main contribution is to increase COVID-19 detection accuracy by offering a new dataset that combines COVID-19 and pneumonia images to create more consistent predictions and by using image processing to allow image normalization and improve model learning.

# Problem Definition

# The reference diagnostic test for COVID-19 pneumonia is real-time reverse transcription-polymerase chain reaction (RT-PCR). The specificity of RT-PCR is approximately 95%, but the sensitivity of RT-PCR at the initial presentation is 60% to 71% because of kit performance, sampling, and transportation limitations. Because of these low sensitivity rates and the need for rapid diagnosis, X-Ray has been frequently used in the current pandemic condition. Also, several cases with initial negative RT-PCR results are reported to have positive chest CT findings or X-rays. So, in the direction of finding an accurate way of testing Covid19, X-rays tend to be more trustworthy than any RT-PCR reports, blood reports or various other symptoms. Which led us here for the project of Covid19 Detection analyzing X-Ray imagery using Neural Networks.

# Project Overview

# In this project, we aim to predict two classes outcome from the input fed Image data of a patient’s Chest X-Ray, to determine whether the patient is suffering from COVID-19 or not. We plan to use Transfer Learning, under the domain of Deep Learning using Convolutional Neural Network to proceed with the project. We are to create a model with a very large dataset, split into Train Test and Validation segments. Four different Transfer Learning models are proposed to be used on the dataset and the outcome is to be on the basis of the most efficient model out of them.

* 1. **Software Specification**

The Project is carried on Python 3.7 kernel with environments of Jupyter Notebook on Kaggle, Google Colab, and PyCharm IDE.

Important Python Libraries were used such as:

**Pandas** (for Data Processing),

**Tensorflow-Keras**(for transfer learning and deep learning models and tools), **OS, Shutil** (for Dataset manipulations),

**Matplotlib, Seaborn**(For Data and Outcome Visualization)

**IPython**(for Image Processing)

* 1. **Data Specification**

Various Datasets were collected from open-source platforms comprising of Positive and Negative Covid CXR images. The entire data is cleaned and processed to form a workable dataset containing around 10000 images of both classes.

Graphical user interface, application, Word

Description automatically generated

The Dataset is publicly uploaded and available on Kaggle Datasets.

1. **LITERATURE REVIEW**

CXRs are a good monitor of COVID-19 chest manifestations, and its scoring system provides an accurate method to predict the disease severity. The study also revealed a positive correlation between the patients’ age and total severity score to the final disease outcome providing a good indicator for clinician to identify at an early stage the patients with the highest risk and plan specific treatment strategies for them.

COVID-19 is a highly infectious disease that has been spread widely throughout the world. The disease management strategies primarily depend upon the early disease diagnosis. However, the dramatic dissemination of the disease created a great challenge due to the insufficient laboratory kits. That is why radiology has become a forefront method during the outbreak of COVID-19.

Current literature is mostly assessing COVID-19 CT findings, as it offers more sensitive results than chest X-ray (CXR) especially in the initial assessment of the patients. The increased number of hospitalized patients and the consequent increase in radiological examinations would make the constant use of chest CT scan (from diagnosis to discharge) difficult to sustain over time. The dependence on CT creates a huge burden on radiology departments and this makes the CXRs greatly substitute the CT examinations. Although chest X-ray (CXR) is considered less sensitive for the detection of pulmonary involvement in early-stage disease, it is useful for monitoring the rapid progression of lung abnormalities in COVID-19[8] [9], especially in critical patients admitted to intensive care units. To provide valuable help for the clinicians and improve the stratification of the disease risk, chest X-ray (CXR) scoring system was tailored providing a semi-quantitative tool for assessment of lung abnormalities.

In this study, we analyzed the CXRs findings and severity scores of patients proven to have COVID-19 in different stages of disease. CXRs abnormalities were detected in 268 of 350 patients (77%) at certain points of the disease course. In our study, each lung was given a score of 0–4 depending on the extent of lung involvement (score 0 = no involvement; 1 ≤ 25%; 2 = 25–50%; 3 = 50–75%; 4 ≥ 75% lung affection). A total severity score was calculated by summing both lung scores (total severity scores ranged from 0 to 8). Borghesi et al. made another CXR scoring system for COVID-19 pneumonia (Brixia score) by dividing the lungs to six zones on frontal projection (upper, middle, and lower zones); then, a score (from 0 to 3) is assigned to each zone based on the lung abnormalities detected on frontal chest projection as follows: score 0, no lung abnormalities; score 1, interstitial infiltrates; score 2, interstitial and alveolar infiltrates (interstitial predominance); and score 3, interstitial and alveolar infiltrates (alveolar predominance). The scores of the six lung zones are then added to obtain an overall “CXR SCORE” ranging from 0 to 18 . In our study, most of the patients showed bilateral lung affection (181 patients, 67.5%) with lower zonal predominance (196, 73.1%) and peripheral distribution (156 patients, 58.2%). The most common CXRs features detected in COVID-19 cases were consolidation seen in 218 patients (81.3%), followed by reticular interstitial thickening seen in 107 patients (39.9%) and GGO seen in 87 patients (32.5%). Few cases showed pulmonary nodules seen in 25 patients (9.3%) and pleural effusion seen in 20 patients (7.5%). This agreed with Wong et al. who did a study on 64 COVID-19 patients, they found that Consolidation was the most common finding (47%), followed by GGO (33%). Also, peripheral predominance was seen in 41% of CXR abnormalities with lower zone distribution (50%) with bilateral lung involvement (50%). Pleural effusion was uncommon, only seen in 3%. Also, Lomoro et al. performed a study on thirty-two patients of COVID-19 disease; they found that consolidation is the most common finding (46.9%) with bilateral lung infection in (78.1%) and lower zone involvement (52%). No pleural effusion was identified. Jacobi et al. stated that standard CXR can easily identify reticular opacities accompanying regions of ground glass attenuation. They state that air space consolidation opacities with peripheral and lower zone distribution are unique for COVID-19 disease. Chen et al. reported bilateral pneumonia as the most common finding on chest radiographs. While Ng et al. reported that CXR lacks sensitivity in the early stages of lung disease. In most studies, pleural effusions, pneumothorax, and lung cavitation are rare in COVID-19 infected patients. Pneumothorax was detected in 2 cases in our study, and it was iatrogenic due to mechanical ventilation in intubated patients. We classified patients according to the stage of illness into four stages (1–4 days, 5–9 days, 10–15 days, and > 15 days). The degree of disease severity was assessed using a semi-quantitative CXRs severity score that reflects the severity of different stages of this disease. The total severity score was lowest at stage 1 compared to other stages, with significant difference among other stages, indicating that the disease changed rapidly within 10–15 days after the onset of the initial symptoms. We estimated the total severity score in the baseline and follow-up CXR, and it ranged from 0 to 8. In most cases (230 patients, 65.7%), TSS was mild, ranging between 0 and 2. While, in 82 patients (23.4%), there was a moderate severity score ranging between 3 and 5. Severe cases with a severity score of between 6 and 8 was found in 38 patients (10.9%) with more disseminated lung involvement. Wong et al. found in their study that 41% had mild findings with a total severity score of 1–2, while moderate and severe cases with more extensive lung involvement were seen in 20% and 8% patients, who had severity scores of 3–4 and 5–6, respectively. There was no patient who had a severity score of > 6 on their baseline CXR with the severity of CXR findings peaking at 10–15 days from the date of symptom onset. In our study, the maximum total severity score was reached in 113 patients (42.2%) in the initial baseline CXR with mean total severity score 1.49 ± 1.53 followed by 92 patients (34.3%) who reached the maximum TSS at 1st follow-up CXR done (done 1–4 days) with mean total severity score 2.08 ± 1.83. The highest total severity score of the CXR findings was found in the 4th follow-up CXR 15 days after the onset of the symptoms with its mean 4.51 ± 1.61. Our study correlated the disease outcome to the patients’ age with a significant difference between the age of the patients and COVID-19 disease outcome (P value = 0.008). The mean age for the recovered patients was 41.09 ± 14.14 while the mean age for the dead patients was 51.04 ± 10.17. Lowest mortality rate was observed in 20–40 years, while patients aging 40–59 and ≥ 60 years showed significantly higher mortality rate. In our study, there were 261 males (74.6%) and 89 females (25.4%) with male patients showing significantly higher mortality rate compared to the female patients (P value 0.025). This agreed with Borghesi et al., who did a study on 783 Italian patients. They found that most patients (67.9%) were males and only 15.2% were younger than 50 years. They stated that in older age groups between 50 and 79 years, there was more significant pulmonary infection with highest severity score seen in males ≥ 50 years or female ≥ 80 especially that underlying comorbidities (such as hypertension, diabetes, cardiovascular disease, and oncologic history) are risk factors of fatal outcome in adult patients with confirmed SARS-CoV-2 infection. In our study, the disease outcome showed a positive correlation with the maximum severity score (6.87 ± 0.71 for the dead patients and 2.06 ± 1.84 for the surviving patients) with high statistical significance (P value < 0.001). In patients with TSS 2, there was a statistical significance between the TSS and the outcome of COVID disease for the survived patients (P value 0.032), while, in patients with TSS 7 and 8, there was a highly statistical significance for the outcome for the dead patients (P value < 0.001). This agreed with Toussie et al. that showed that the severity of lung involvement on the initial chest radiograph was associated with more need for patients’ hospitalization as well as the increased risk of intubation and have proposed the use of initial CXR severity scores as a prognostic indicator of COVID-19 patients’ outcome. The major strength of this study is the large sample size, which consisted of 350 COVID-19 patients. Our study had some limitations. First, it is a retrospective analysis. Second, the lack of correlation between CXR severity score and patient comorbidities (such as hypertension, diabetes, cardiovascular disease, and oncologic history). Third, not all the patients could be followed till the final outcome as the course of the disease was truncated in these patients. Fourth, CXR serial follow-up studies were not performed in a uniform pattern as it was dedicated by the clinician as regards the clinical condition. Fifth, for severe cases in the intensive care unit, the portable AP CXR was suboptimal with only few cases performed CT, so we could not judge the sensitivity of CXR.

# PROBLEMFORMULATION

## Background

During a pandemic, such as COVID-19, a timely and precise diagnosis is critical. It improves patient outcomes and relieves burden on health-care systems that are dealing with a growing rate of infection.

The polymerase chain reaction (PCR) is the current preferred approach for diagnosing COVID-19 (PCR). However, some of the worst-affected locations are unable to obtain enough kits to fulfill demand, and many countries are unable to conduct tests due to a lack of lab facilities.

Deep learning models, which are a type of artificial intelligence (AI), are being studied and used to detect and diagnose a wide range of diseases.

Deep learning algorithms could be employed in this case to identify infected individuals using chest X-ray scans, which are readily available around the world. This approach could be employed in situations where PCR diagnostics are currently unavailable.

Deep learning for X-ray analysis might drastically cut the time it takes to diagnose patients, with an AI model processing up to 200 images in the time it takes a radiologist to analyze one.

## Method

We propose using a deep convolutional neural network trained on COVID-19 and pneumonia images, as well as a fresh dataset comprising COVID-19 and pneumonia images. Both are open to the public via GitHub and Kaggle, respectively. COVID-19 cases are represented by the chest X-ray or CT images available on GitHub. It was made by putting together medical photos from publicly accessible sources and publications. There are 204 COVID-19 X-ray pictures in this dataset. The Kaggle dataset, on the other hand, was produced for a pneumonia detection challenge. Bounding boxes surround sick lung regions in the photos. Without the boundary boxes, the samples are negative and show no signs of pneumonia. The presence of bounding boxes in samples indicates the presence of pneumonia.

By combining COVID-19 and pneumonia images, we propose a new dataset that is both larger and more diversified. Because normal pneumonia and COVID-19 have comparable appearances in chest X-ray images, including pneumonia images in the training dataset implies an extra advantage. This dataset merger enables the creation of a more robust model capable of distinguishing between those diseases. Another benefit of this merge is that it expands the train dataset, which is important because COVID-19 photos are scarce at the time of authoring this research. Because of the similarities between pneumonia and COVID-19, this merge does not increase the size of the COVID-19 picture collection, but it does improve detection quality.

To avoid biased findings, we split the photos into train and test sets, dividing all the data in a balanced way, meaning that all samples of each class in the training sets are well-balanced. Despite the fact that we have a significant number of pneumonia and normal photos, we created a dataset of 1500 photographs for this reason.

For training, we chose 104 COVID-19 images, 205 health lung images, and 204 pneumonia images, and for testing, we chose 100 COVID-19 images, 444 health lung images, and 443 pneumonia images. There are more samples of pneumonia and normal photos in the Kaggle dataset, but we chose only 205 for training to ensure a balanced dataset. We add more pneumonia and normal photos to the test stage to demonstrate the model's robustness in detecting COVID-19 with no false positives. To summarize, we employ a total of 513 photos in the train set and 887 images in the test set.

X-ray of a person's head

Description automatically generated with low confidence

1. **MODEL ARCHITECTURE**

The employed architecture was chosen based on the good results gained with CNNs in state-of-the-art works for COVID-19 image classification, as well as the good results obtained with this kind of architecture in other related tasks. Based on the Single Shot Multibox Detector, we adopted the same network design provided in (SSD). Using a single deep neural network, this design is optimized for recognizing objects in photos. This method divides the output space of bounding boxes into a series of default boxes for each feature map point, using varying aspect ratios and sizes. The network generates scores for the presence of each item type in each default box at prediction time, and then adjusts the box to better match the object shape.

SSD offers comparable accuracy to approaches that use more than one architecture for detecting objects, while being faster and providing a single framework for both training and inference, according to experimental results on various notable datasets. SSD offers substantially better accuracy than other single-stage algorithms, even with a smaller input image size.

A picture containing chart

Description automatically generated

In this design, we employ VGG-16 as the basis network for feature extraction. Fast R-CNN is also used in this model. We have many boxes with varied sizes and aspect ratios over the entire image during training. When compared to the ground truth, SSD determines the box with the highest Intersection-Over-Union (IoU)

Our main goal is to develop a more robust model that can handle a wide range of input item sizes and forms. As a result, a data augmentation step is conducted during the SSD training. The following operations are applied to each image in the dataset as part of this process:

• Keep the original input image in its entirety.

• Pick a patch with a minimum overlap of 0.1, 0.3, 0.5, 0.7, or 0.9 with the objects. Each sampled patch is a proportion of the original image size between [0.1, 1].

• Pick a patch at random.

1. **MODEL COMPARISON REVIEW**

Before finalizing the Model with VGG-16, a few other models were also trained and tested with the same dataset for a comparative study on the accuracy on each of them.

For the study, the following pre-built models were used from the keras.applications library:

● DenseNet169

● VGG16

● InceptionV3

● ResNet152V2

Details on the models are provided for a better idea.

**Inception Model:**

**Introduction:**

In 2014, Google (together with other academic institutions) published a paper describing a novel deep learning convolutional neural network architecture, which at the time was the largest and most efficient deep neural network architecture available.

An Inception Network was used as the unique design, and a GoogLeNet version won the ImageNet Large-scale Visual Recognition Challenge 2014 classification computer vision competition with the best performance (ILVRC14).

Since 2014, both we and the deep learning field have come a long way. Several deep learning architectures will approach or exceed human-level classification and object detection performance by 2020.

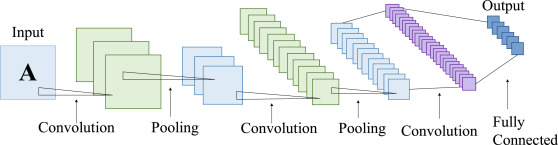
On the other hand, convolutional neural networks owe their innovations and advancements to their forefathers.

**Architecture of Inception Model:**

An inception network is a deep neural network with Inception modules, which are repeating architectural designs.

The following are the essential principles to remember when developing an Inception model:

1. High performance necessitates the use of large deep neural networks. To be considered huge, a neural network must include multiple more layers and units inside these levels.
2. Convolutional neural networks benefit from extracting features at various scales. The biological human visual brain recognizes patterns at different scales, which are then combined to create larger object perceptions. As a result, multi-scale networks are able to learn more.
3. Consider the Hebbian Principle, which claims that neurons that fire at the same time will connect.



Table

Description automatically generated

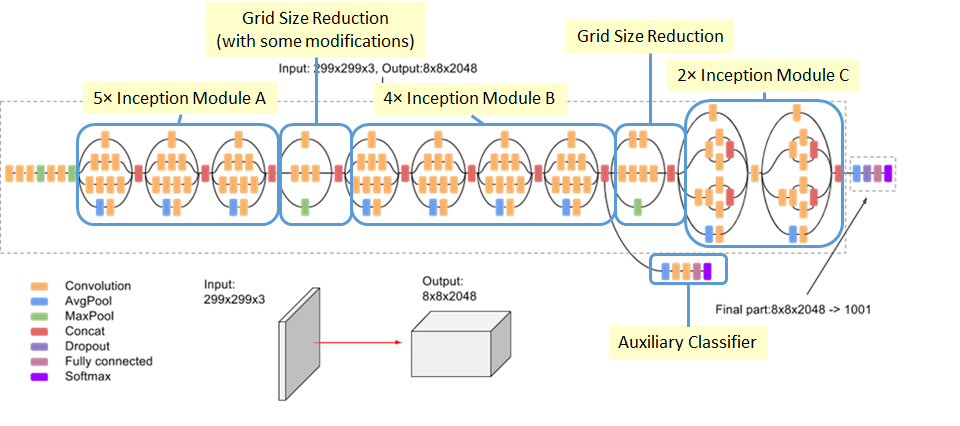
In practical cases, the above-mentioned guidelines have several technological flaws. In large networks, overfitting is prevalent, and chaining numerous convolutional methods together increases the network's computing cost. To make the Inception network and module a reality, the researchers designed intuitive convnet topologies.

In this 2013 report, Lin et al. proposed 1x1 convolution, popularly known as "Network in Network." A 1x1 convolution uses the element-wise product of all pixel values in an image. The image (input data) is convoluted with the conv 1x1 filter, resulting in an output with the dimensions 1 x 1 x n (where 'n' is the number of filters). A 1x1 filter does not learn spatial patterns inside the image, but it does learn patterns across the depth of the image (cross channel). As a result, 1x1 convolution filters not only decrease method dimension but also allow the network to learn more.

The 1x1 convolution reduces the number of input channels, which results in fewer channels in the output. This section is the Inception network's bottleneck layer (shown in a diagram further down below). Pooling layers down-sample images as they go through the network (lower the height and width). 1x1 convolution has the added benefit of reducing the image's height, width, and number of channels.

To improve the performance of a neural network, increase the number of layers (depth) and units/neuron within the layers (width), which ineffectively develops a more comprehensive network.

Structure of a basic Inception model is given below:



The Inception network has the luxury of using different filter sizes within its convolutional layers. The 1x1 convolution, one of the convolution filter sizes used by Inception, has already been addressed. The others are 3x3 and 5x5.

Within a convnet, different conv filter sizes learn spatial patterns and recognize properties at different scales.

Prior to the advent of the Inception network, researchers had to choose filter sizes to use in deep convolutional neural networks to obtain optimal performance.

By utilizing different filter sizes of 1x1, 3x3, and 5x5, Inception eliminates the need for such decisions. 1x1 learns patterns across the depth of the input, whereas 3x3 and 5x5 learn spatial patterns across the three dimensions of the input (height, width, and depth). The representational capacity rises when all of the patterns acquired from the various filter sizes are combined.

To give a single Inception module output, the Inception module contains a concatenation layer that merges all of the conv filters' outputs and feature maps into a single object.

**Benefits of the Inception Module:**

1. Convolutional neural networks provide a high level of performance gain.
2. Efficient computing resource utilization for an Inception network's high-performance output with minimum increase in compute load
3. Extraction of features from input data at various scales using different convolutional filter sizes
4. Cross-channel patterns are learned by 1x1 conv filters, allowing the network to extract more information.

**ResNet**

**Introduction:**

ResNet, or Residual Networks, is a well-known neural network that is used to perform a variety of computer vision tasks. This model was the ImageNet challenge winner in 2015. ResNet was a game-changer because it allowed us to train 150-layer deep neural networks. Prior to ResNet, training very deep neural networks was challenging due to the problem of vanishing gradients.

AlexNet, the ImageNet 2012 winner and the model that seems to have spurred interest in deep learning, has just eight convolutional layers, compared to 19 for the VGG network, 22 for Inception or GoogleNet, and 152 for ResNet 152.

**Architecture of ResNet:**

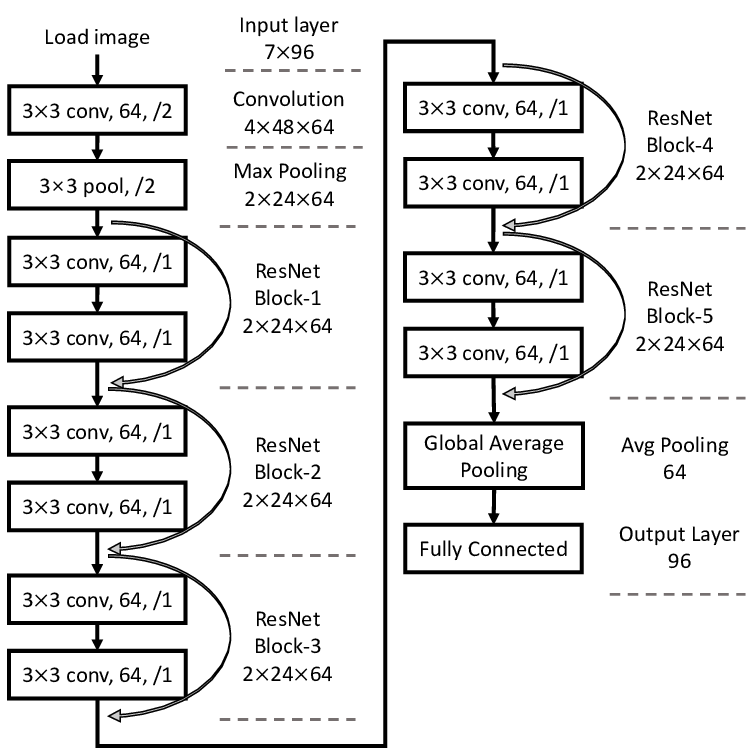
ResNet is a Convolutional Neural Network (CNN) architecture that tackles the "vanishing gradient" problem, enabling deeper networks to outperform shallower networks. A decreasing gradient happens during backpropagation. When there are too many layers in the neural network training process, the gradient becomes very small until it disappears, and optimization is halted. The problem is resolved by ResNet adopting "identity shortcut connections."

It has two stages of implementation:

ResNet generates and bypasses a large number of levels that aren't used initially, recycling activation functions from preceding layer.

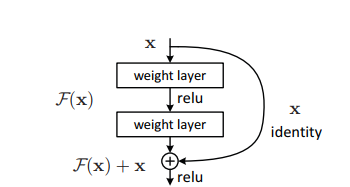
A second stage involves retraining the network and extending the "residual" convolutional layers. This enables the investigation of more parts of the feature space that a shallow convolutional network architecture would have skipped.

Structure of a basic ResNet model is given below:



ResNet was the first to introduce the concept of skip connection. The graphic below depicts the skip connection. Convolution layers are piled one on top of the other in the figure on the left. Convolution layers are still stacked on the right, but the original input is now added to the convolution block's output. This is known as "skipping connection."

These Residual blocks were introduced to help with the issue of training very deep networks, and the ResNet model is made up of them.



The introduction of these Residual blocks alleviated the challenge of training very deep networks, and the ResNet model is built up of these blocks. The first thing we note in the above diagram is that there is a direct connection that skips several of the model's levels. The heart of residual blocks is a connection known as the ‘skip connection.' Because of the skip connection, the output is not same. Without the skip connection, input 'X is multiplied by the layer's weights, then a bias term is added. Then comes the activation function, f() and we get the output as H(x).

**H(x)=f(wx + b) or H(x)=f(x)**

Thanks to the installation of a new skip connection method, the output is now H(x), where

**H(x)=f(x)+x**

However, when using a convolutional layer or pooling layers, the input dimension may differ from the output dimension. As a result, these two ways can be used to solve the problem:

* To enhance its dimensions, zero is padded with the skip connection.
* To match the dimensions, 11 convolutional layers are added to the input.

In this situation, the result is:

**H(x)=f(x)+w1.x**

In this case, an additional parameter w1 is added, but in the first technique, no additional parameter is added.

ResNet's skip connections strategy tackles the problem of disappearing gradient in deep CNNs by enabling the gradient to flow along an additional shortcut channel. In addition, if any layer degrades architecture performance, regularization will skip it.

The architecture is inspired on VGG-19 and has a 34-layer plain network to which shortcut and skip connections are added. These residual blocks or skip connections change the design into a residual network.

**Advantages of ResNet:**

1. Networks with a large number of layers (even thousands) can be easily taught without increasing the percentage of training errors.
2. Using identity mapping, ResNets can help solve the vanishing gradient problem.

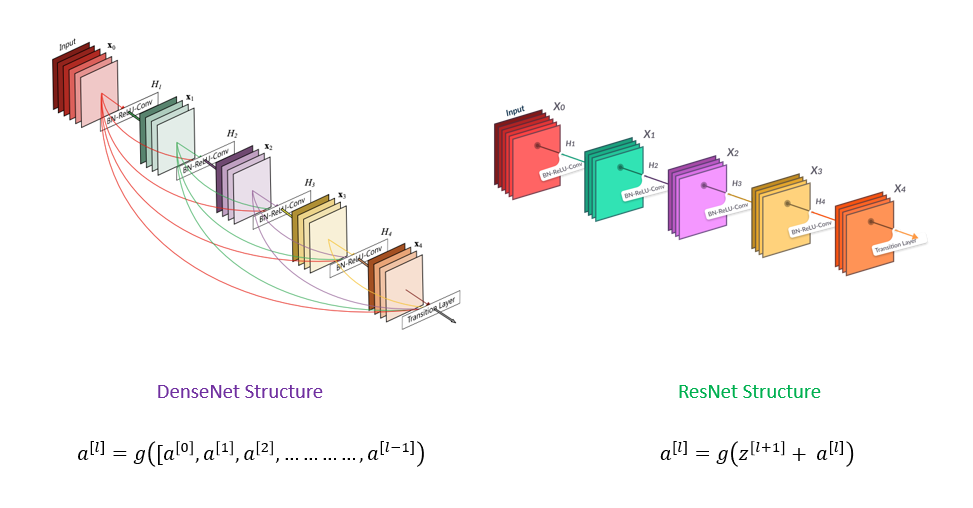
**DenseNet:**

**Introduction:**

The Dense Convolutional Network, or simply DenseNet, is an innovative neural network discovery for visual object detection. DenseNet is quite similar to ResNet, although there are several key distinctions. DenseNet concatenates (.) the output of the previous layer with the output of the future layer, whereas ResNet utilizes an additive approach (+) that combines the previous layer (identity) with the future layer. DenseNet was created primarily to address the vanishing gradient's effect on high-level neural networks' accuracy. Simply said, the information evaporates before reaching its target due to the longer travel between the input and output layers. It’s a convolutional neural network that use Dense Blocks to create dense connections between layers, with all layers (with matching feature-map sizes) connected directly to one another. To retain the feed-forward nature, each layer gets extra inputs from all preceding levels and passes on its own feature-maps to all subsequent layers.

**Architecture:**

Dense Convolutional Network (DenseNet) is a feed-forward network that connects each layer to every other layer. Our network has L(L+1)/2 direct connections, whereas standard convolutional networks with L layers have L connections - one between each layer and its succeeding layer. All previous layers' feature maps are utilized as inputs into each layer, and its own feature maps are used as inputs into all subsequent layers. DenseNets have a number of compelling advantages, including the elimination of the vanishing-gradient problem, improved feature propagation, feature reuse, and a significant reduction in the number of parameters.

Table

Description automatically generated

The standard ResNet structure and a 5-layer dense block with a growth rate of k = 4are shown in this figure. Using the composite function operation, the previous layer's output becomes the second layer's input. Convolution, pooling, batch normalization, and non-linear activation layers are all part of this composite operation.

The network has L(L+1)/2 direct connections because of these links. The architecture has L layers.

DenseNet comes in a variety of flavors, including DenseNet-121, DenseNet-160, and DenseNet-201. The numbers represent the neural network's layer count.

DenseNet (Dense Convolutional Network) is a network design that focuses on making deep learning networks grow deeper while also making them more efficient to train by employing shorter connections between layers. DenseNet is a convolutional neural network in which each layer is connected to all other layers deeper in the network, so the first layer is connected to the 2nd, 3rd, 4th, and so on. This is done to allow maximal information flow between network tiers. Each layer takes inputs from all previous levels and passes on its own feature maps to all subsequent layers to maintain the feed-forward character. Unlike ResNets, it concatenates features instead of summarizing them. The 'ith' layer, then, contains I inputs and is made up of feature maps from all the convolutional blocks before it. All of the following 'I-i' layers receive its own feature maps. In contrast to standard deep learning designs, this introduces '(I(I+1))/2' connections to the network. As a result, it has fewer parameters than standard convolutional neural networks because no meaningless feature maps need to be learned.

Apart from the basic convolutional and pooling layers, DenseNet has two key components. Dense Blocks and Transition layers are what they're called.

DenseNet begins with a basic layer of convolution and pooling. Then there's a dense block followed by a transition layer, followed by another dense block followed by a transition layer, followed by another dense block followed by a transition layer, and lastly a dense block followed by a classification layer.

The first convolution block contains 64 7x7 filters with a stride of 2. A MaxPooling layer with 3x3 max pooling and a stride of 2 follows.

Each convolutional block passes through the following phases after receiving the input: BatchNormalization, ReLU activation, and lastly the actual Conv2D layer. Every dense block has two convolutions, with kernel sizes of 1x1 and 3x3. This is repeated six times in dense block 1, twelve times in dense block 2, twenty-four times in dense block 3, and sixteen times in dense block four.

Each of the 1x1 convolutions in dense block has four times the number of filters. As a result, we use four filters, but three of them are only present once. In addition, the input and output tensors must be concatenated.

**Advantages of DenseNet:**

They solve the vanishing-gradient problem, improve feature propagation, promote feature reuse, and cut the number of parameters in half. In comparison to alternative systems, the DenseNet architecture has numerous notable advantages. The authors first claim that their architecture outperforms the other competing architectures in ImageNet. My research in Near-Identical Images confirmed that the DenseNet design indeed provide the optimum image representation. Second, the authors claim that their increased parameter efficiency makes it easier to train the network. This is true when compared to other network topologies of equal size. I believe that the training time is competitive with that of certain lower-layer networks. The improved performance is well worth the additional training time.

**VGG16:**

**Introduction:**

**In their publication "Very Deep Convolutional Networks for Large-Scale Image Recognition," K. Simonyan and A. Zisserman from the University of Oxford proposed the VGG16 convolutional neural network model. In ImageNet, a dataset of over 14 million images belonging to 1000 classes, the model obtains 92.7 percent top-5 test accuracy. It was a well-known model that was submitted to the ILSVRC-2014. It outperforms AlexNet by sequentially replacing big kernel-size filters (11 and 5 in the first and second convolutional layers, respectively) with numerous 33 kernel-size filters. VGG16 had been training for weeks on NVIDIA Titan Black GPUs.**

VGG is a multilayer deep Convolutional Neural Network (CNN) architecture that stands for Visual Geometry Group. VGG-16 and VGG-19 feature 16 and 19 convolutional layers, respectively, and the term "deep" refers to the number of layers.

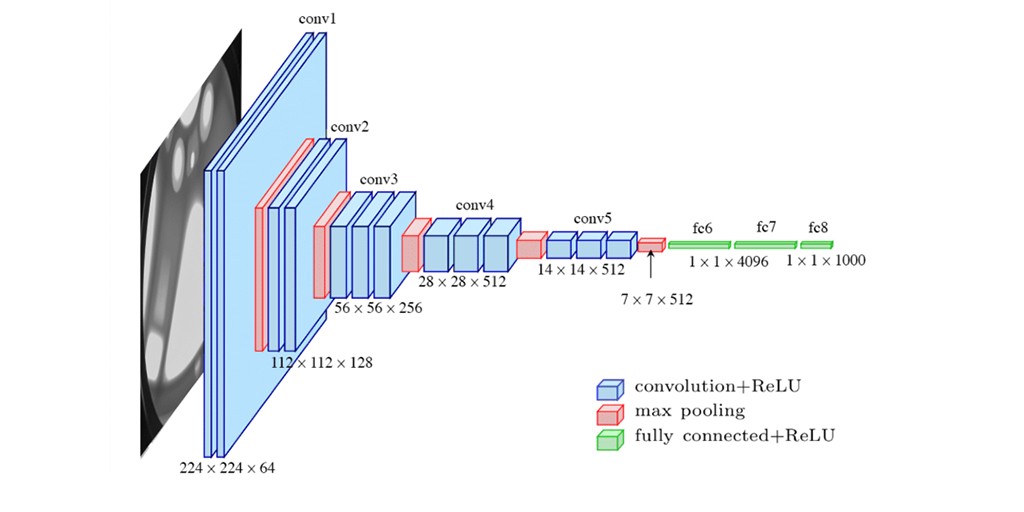
Cutting-edge object recognition models are built on top of the VGG architecture. On a range of applications and datasets other than ImageNet, the VGGNet, which was constructed as a deep neural network, beats baselines. It is also one of the most commonly utilized image recognition architectures today. The VGGNet-16 contains 16 layers and can categorize pictures into 1000 different object categories, including keyboards, animals, pens, and mice. The model also accepts images with a resolution of 224 by 224 pixels.

**Architecture:**

Small convolutional filters are used to build the VGG network. There are 13 convolutional layers and three fully linked layers in the VGG-16.

Let's look at VGG's architecture in more detail:

* **Input:** The VGGNet accepts images with a size of 224x224 pixels. To keep the image input size consistent for the ImageNet competition, the model's authors chopped out the center 224x224 patch in each image.
* **Convolutional Layers**: VGG's convolutional layers use a small receptive field (3x3), the smallest size that still captures up/down and left/right movement. There are also 1x1 convolution filters that perform a linear transformation of the input. Then there's a ReLU unit, which is a significant AlexNet invention that cuts training time in half. The rectified linear unit activation function (ReLU) is a piecewise linear function that outputs the input if it is positive and zero otherwise. To maintain spatial resolution after convolution, the convolution stride is set to 1 pixel (stride is the number of pixel shifts over the input matrix).
* **Hidden Layers:** The VGG network's hidden layers all use ReLU. Local Response Normalization (LRN) is rarely used in VGG since it increases memory usage and training time. Furthermore, it has no effect on total accuracy.
* **Fully Connected Layers:** There are three fully connected layers in the VGGNet. The first two layers each have 4096 channels, whereas the third layer has 1000 channels, one for each class.

Table

Description automatically generated

The number 16 in the term VGG alludes to the deep neural network's 16 layers (VGGNet). This indicates that VGG16 is a large network with over 138 million parameters. Even by modern standards, it is a massive network. The simplicity of the VGGNet16 architecture, on the other hand, is what makes the network appealing. It may be argued that its architecture is quite uniform just by glancing at it.

A few convolution layers are followed by a pooling layer that decreases the height and width of the image. When it comes to the number of filters we can employ, we have roughly 64 options, which we can expand to around 128 and then to 256. We can utilize 512 filters in the final levels.

Every step or stack of the convolution layer doubles the number of filters that can be used. This is a key design principle for the VGG16 network's architecture. One of the major disadvantages of the VGG16 network is that it is a large network, which means that training its parameters takes longer.

The VGG16 model is over 533MB in size because to its depth and number of fully connected layers. Implementing a VGG network is hence time-consuming.

Although the VGG16 model is employed in a variety of deep learning image classification challenges, smaller network topologies like GoogleNet and SqueezeNet are frequently preferred. In any case, the VGGNet is an excellent learning building block because it is simple to implement.

In the ILSVRC-2012 and ILSVRC-2013 contests, VGG16 outperformed prior versions of models. Furthermore, the VGG16 result is fighting for the classification task winner (GoogLeNet with 6.7 percent error) and exceeds the ILSVRC-2013 winning submission Clarifai by a significant margin. It scored roughly 11.7 percent using external training data and 11.2 percent without. In terms of single-net performance, the VGGNet-16 model outperforms a single GoogLeNet by roughly 0.9 percent, with about 7.0 percent test error.

**Advantages of VGG16:**

As the number of layers in CNN grows, the model's ability to fit more complex functions grows as well. As a result, more layers imply better performance. This is not the same as an Artificial Neural Network (ANN), where increasing the number of layers does not always result in higher performance.

In both the ILSVRC-2012 and ILSVRC-2013 competitions, VGG16 outperformed the previous generation of models. In terms of single-net performance, the VGG16 architecture came out on top (7.0 percent test error). The mistake rates are listed in the table below.

When working with a VGG network, there are two major downsides to be mindful of. To begin with, training requires time. Second, the weights associated with network architecture are substantial. The trained VGG16 model is almost 500MB in size because to its depth and amount of completely linked nodes. Smaller network topologies are often preferred over VGG16 in many deep learning image categorization challenges (such as SqueezeNet, GoogleNet, etc.

1. **SOURCE-CODE**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedBackground pattern

Description automatically generated with low confidenceGraphical user interface, text, application

Description automatically generatedTable

Description automatically generated with medium confidence

Application

Description automatically generated with low confidence

A picture containing calendar

Description automatically generated

|  |  |
| --- | --- |
| Table  Description automatically generatedTable  Description automatically generatedTable  Description automatically generated with medium confidenceA picture containing graphical user interface  Description automatically generatedTable  Description automatically generatedTable  Description automatically generated | Table  Description automatically generatedTable  Description automatically generatedTable  Description automatically generatedTable  Description automatically generatedTable  Description automatically generated |

|  |  |
| --- | --- |
| A screenshot of a computer  Description automatically generated with medium confidenceTable  Description automatically generatedTable  Description automatically generatedTable  Description automatically generated | Table  Description automatically generatedTable  Description automatically generatedTable  Description automatically generatedA screenshot of a computer  Description automatically generated with low confidence |

|  |  |
| --- | --- |
| Table  Description automatically generatedTable  Description automatically generatedTable  Description automatically generatedTable  Description automatically generated | A picture containing text, document  Description automatically generatedA screenshot of a computer  Description automatically generated with low confidenceTable  Description automatically generatedTable  Description automatically generated |

|  |  |
| --- | --- |
| Text  Description automatically generated with medium confidence  Table  Description automatically generatedTable  Description automatically generated with medium confidenceCalendar  Description automatically generatedA picture containing chart  Description automatically generatedA picture containing table  Description automatically generated | Table  Description automatically generatedTable  Description automatically generatedTable  Description automatically generated |

|  |  |
| --- | --- |
| Table  Description automatically generatedTable  Description automatically generatedTable  Description automatically generated | Table  Description automatically generatedCalendar  Description automatically generatedTable  Description automatically generatedTable  Description automatically generated |

|  |  |
| --- | --- |
| Table  Description automatically generatedTable  Description automatically generatedTable  Description automatically generatedCalendar  Description automatically generated with medium confidence | Table  Description automatically generatedTable  Description automatically generatedCalendar  Description automatically generatedTable  Description automatically generated |

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

|  |  |
| --- | --- |
| Table  Description automatically generatedTable  Description automatically generatedTable  Description automatically generated | Table  Description automatically generatedTable  Description automatically generated |

|  |  |
| --- | --- |
| Table  Description automatically generated | Table  Description automatically generatedTable  Description automatically generatedTable  Description automatically generated |

|  |  |
| --- | --- |
| Table  Description automatically generatedTable  Description automatically generatedTable  Description automatically generated |  |

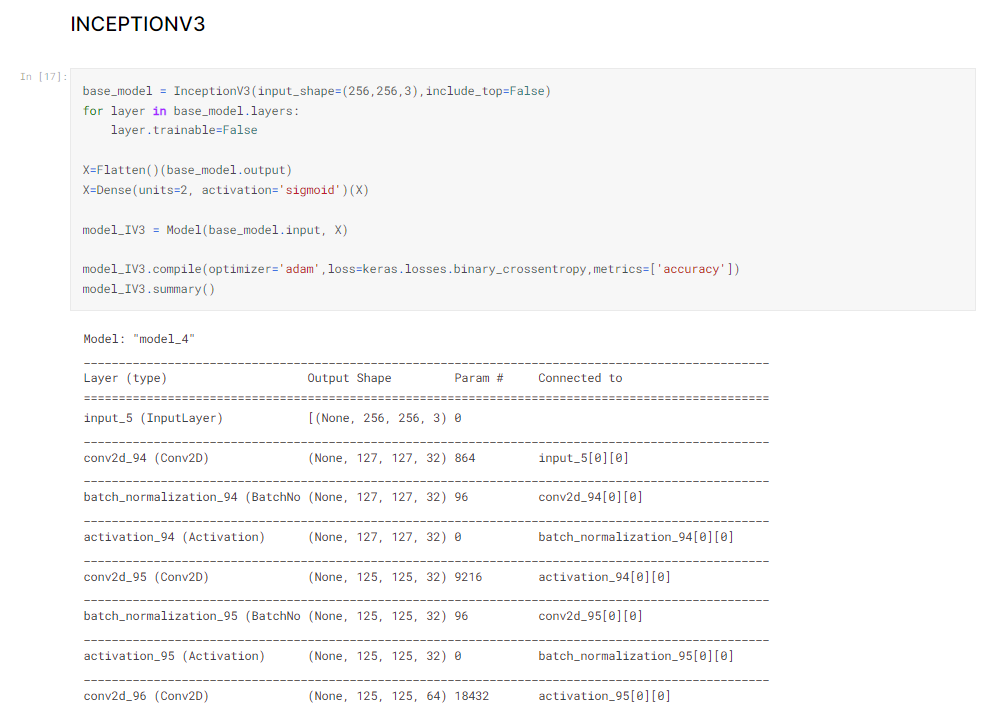
|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

Table

Description automatically generatedTable

Description automatically generated

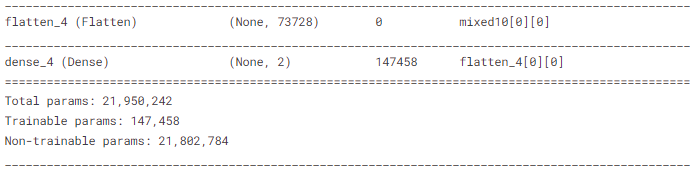




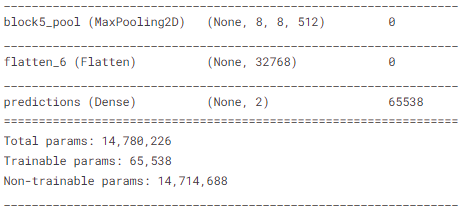
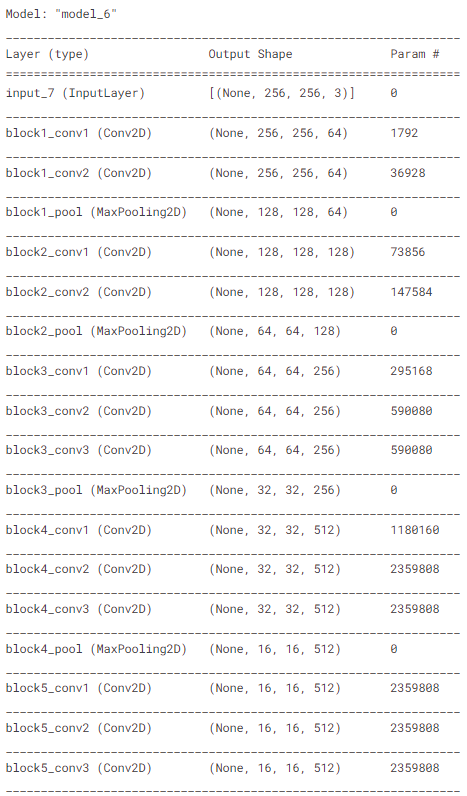
|  |  |
| --- | --- |
|  |  |

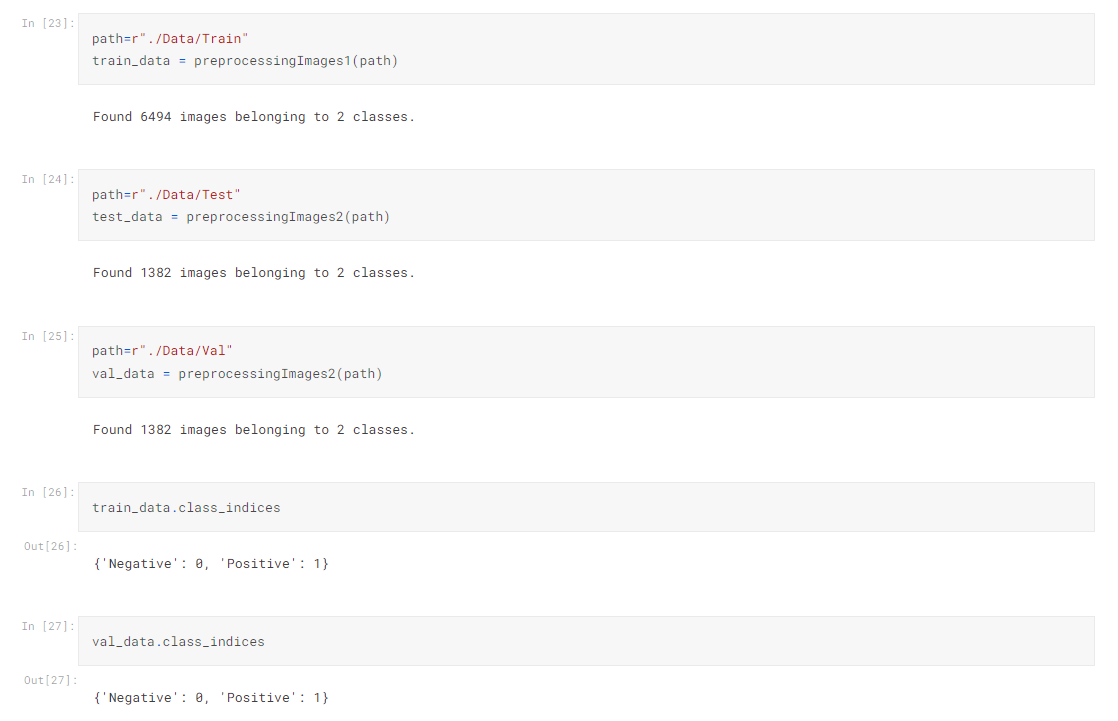
|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |





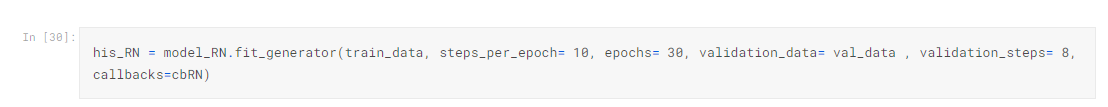




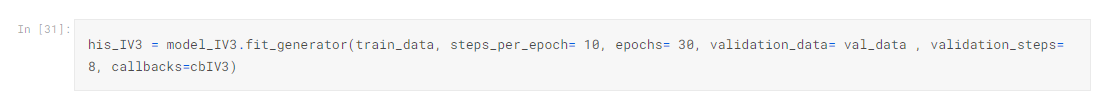




|  |  |
| --- | --- |
|  |  |



|  |  |
| --- | --- |
|  |  |



|  |  |
| --- | --- |
|  |  |



|  |  |
| --- | --- |
|  |  |

Graphical user interface, text, application, Teams

Description automatically generated

Graphical user interface, text, application

Description automatically generatedText

Description automatically generated with medium confidenceText

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generated

# RESULTS

# A picture containing chart Description automatically generatedA picture containing graphical user interface Description automatically generated

# X-ray of a person's chest Description automatically generated with low confidenceA picture containing text, x-ray film Description automatically generated

# REFERENCES

1. M. Dur-e-Ahmad and M. Imran, “Transmission Dynamics Model of Coronavirus COVID-19 for the Outbreak in Most Affected Countries of the World,” International Journal of Interactive Multimedia and Artificial Intelligence, vol. In Press, no. In Press, pp. 1-4, 2020.
2. S. J. Fong, N. D. G. Li, R. Gonzalez-Crespo and E. Herrera-Viedma, “Finding an Accurate Early Forecasting Model from Small Dataset: A Case of 2019-nCoV Novel Coronavirus Outbreak,” International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, no. 1, pp. 132- 140, 2020.
3. Q. Li, X. Guan, P. Wu, X. Wang, L. Zhou, Y. Tong, R. Ren, K. S. M. Leung, E.H.Y. Lau, J. Y. Wong and others, “Early transmission dynamics in Wuhan, China, of novel coronavirus–infected pneumonia,” New England Journal of Medicine, 2020.
4. D. Wang, B. Hu, C. Hu, F. Zhu, X. Liu, J. Zhang, B. Wang, H. Xiang, Z. Cheng, Y. Xiong, Y. Zhao, Y. Li, X. Wang and Z. Peng, “Clinical Characteristics of 138 Hospitalized Patients With 2019 Novel Coronavirus–Infected Pneumonia in Wuhan, China,” JAMA, vol. 323, pp. 1061-1069, 3 2020.
5. S. Chauvie, A. De Maggi, I. Baralis, F. Dalmasso, P. Berchialla, R. Priotto, P. Violino, F. Mazza, G. Melloni and M. Grosso, “Artificial intelligence and radiomics enhance the positive predictive value of digital chest tomosynthesis for lung cancer detection within SOS clinical trials,” European Radiology, p. 1–7, 2020.
6. G. Chassagnon, M. Vakalopoulou, N. Paragios and M.-P. Revel, “Artificial intelligence applications for thoracic imaging,” European Journal of Radiology, vol. 123, p. 108774, 2020.
7. F. Song, N. Shi, F. Shan, Z. Zhang, J. Shen, H. Lu, Y. Ling, Y. Jiang and Y. Shi, “Emerging 2019 Novel Coronavirus (2019-nCoV) Pneumonia,” Radiology, vol. 295, pp. 210-217, 2020.
8. J. C. L. Rodrigues, S. S. Hare, A. Edey, A. Devaraj, J. Jacob, A. Johnstone, R. McStay, A. Nair and G. Robinson, “An update on COVID-19 for the radiologist-A British society of Thoracic Imaging statement,” Clinical Radiology, 2020.
9. J. Wu, J. Liu, X. Zhao, C. Liu, W. Wang, D. Wang, W. Xu, C. Zhang, J. Yu, B. Jiang and others, “Clinical characteristics of imported cases of COVID-19 in Jiangsu province a multicenter descriptive study,” Clinical Infectious Diseases, 2020.
10. F. Shi, L. Xia, F. Shan, D. Wu, Y. Wei, H. Yuan, H. Jiang, Y. Gao, H. Sui and D. Shen, Large-Scale Screening of COVID-19 from Community
    1. Acquired Pneumonia using Infection Size-Aware Classification, arXiv preprint arXiv:2003.09860, 2020.
11. S. Wang, J. M. Bo Kang, X. Zeng and M. Xiao, “A deep learning algorithm using CT images to screen for Corona Virus Disease COVID-19),” medRxiv, 2020.
12. L. Wang and A. Wong COVID-Net A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest Radiography Images, arXiv preprint arXiv:2003.09871 ,2020.
13. E. E.-D. Hemdan, M. A. Shouman and M. E. Karar, COVIDX-Net A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images, arXiv preprint arXiv: arXiv:2003.11055, 2020.
14. J. P. Cohen, P. Morrison and L. Dao, COVID-19 Image Data Collection, arXiv preprint arXiv: arXiv, arXiv:2003.11597, 2020.
15. RSNA Pneumonia Detection Challenge. Kaggle. [online] Available at: https://www.kaggle.com/c/rsna-pneumonia-detection-challenge, Accessed 29 April 2020.
16. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman, “The pascal visual object classes (voc) challenge,” International journal of computer vision, vol. 88, p. 303–338, 2010.
17. W.Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, “SSD Single Shot MultiBox Detector,” Lecture Notes in Computer Science, p. 21–37, 2016.
18. K.Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” CoRR, vol. abs1409.1556, 2014.
19. H. Jansen, Radiología dental. Principios y técnicas., Mc Graw Hill, 2002.
20. A. M. Reza, “Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement,” Journal of VLSI signal processing systems for signal, image and video technology, vol. 38, p. 35–44, 2004.
21. H.-W. Ng, V. D. Nguyen, V. Vonikakis and S. Winkler, “Deep learning for emotion recognition on small datasets using transfer learning,” in Proceedings of the 2015 ACM on international conference on multimodal interaction, 2015.
22. Q. Sun, Y. Liu, T.-S. Chua and B. Schiele, “Meta-transfer learning for few-shot learning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019.