

PNEUMONIA DETECTION USING ENSEMBLE LEARNING

## A PROJECT REPORT

***Submitted by***

# AKSHAYA.K [REGISTER NO: 211417104010]

# ANUSHA.R [REGISTERNO: 211417104020]

# HARRSHEETHA.S [REGISTER NO: 211417104082]

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

# BACHELOR OF ENGINEERING

**IN**

## COMPUTER SCIENCE AND ENGINEERING

**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.**

# ANNA UNIVERSITY: CHENNAI 600025

**AUGUST 2021**

# BONAFIDE CERTIFICATE

Certified that this project report **“PNEUMONIA DETECTION SUING ENSEMBLE LEARNING** “is the bonafide work of **“AKSHAYA K (211417104010), ANUSHA R (211417104020), HARRSHEETHA S (211417104082)”** who carried out the project work under my supervision.

## SIGNATURE SIGNATURE

**Dr.S.MURUGAVALLI,M.E.,Ph.D., ANITHA MOSES.V**

**HEAD OF THE DEPARTMENT SUPERVISOR**

**ASSOCIATE PROFESSOR**

DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE NAZARATHPETTAI, NAZARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600123.

Certified that the above candidate(s) was/were examined in the Anna University Project Viva-Voce Examination held on....05/08/2021......

## INTERNAL EXAMINER EXTERNAL EXAMINE

**ACKNOWLEDGEMENT**

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like express our heartfelt and sincere thanks to our Directors

## Tmt.C.VIJAYARAJESWARI, Dr.C.SAKTHIKUMAR,M.E.,Ph.D.,and

**Tmt. SARANYASREE SAKTHIKUMAR B.E., M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Guide:** **ANITHA MOSES** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

**AKSHAYA .K**

**ANUSHA. R**

**HARRSHEETHA S**

i

**ABSTRACT**

Pneumonia is a life-threatening infectious disease affecting one or both lungs in humans commonly caused by bacteria called Streptococcus pneumonia. One in three deaths in India is caused due to pneumonia as reported by World Health Organization (WHO). Chest X-Rays which are used to diagnose pneumonia need expert radiotherapists for evaluation. Thus, developing an automatic system for detecting pneumonia would be beneficial for treating the disease without any delay particularly in remote areas. Due to the success of deep learning algorithms in analyzing medical images, Convolutional Neural Networks (CNNs) have gained much attention for disease classification. In addition, features learned by pre-trained CNN models on large-scale datasets are much useful in image classification tasks. In this work, we appraise the functionality of pre-trained CNN models utilized as feature-extractors followed by different classifiers for the classification of abnormal and normal chest X-Rays. We analytically determine the optimal CNN model for the purpose. Statistical results obtained demonstrates that pretrained CNN models employed along with supervised classifier algorithms can be very beneficial in analyzing chest X-ray images, specifically to detect Pneumonia.

ii

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGENO.** |
|  | **ABSTRACT** | ii |
|  |
|  | **TABLE OF CONTENTS**  **LISTOF FIGURES** | iii  v |
|  |  |  |
| **1.** | **INTRODUCTION** | 1 |
|  | 1.1 Overview | 1 |
|  | 1.2 Problem Definition | 4 |
| **2.** | **LITERATURE SURVEY** | 5 |
| **3.** | **SYSTEM ANALYSIS** | 7 |
|  | 3.1 Existing System | 7 |
|  | 3.2 Proposed system | 7 |
|  | 3.3 Requirement Analysis and Specification | 8 |
|  | 3.3.1 Input Requirements | 8 |
|  | 3.3.2 Output Requirements | 8 |
|  | 3.3.3 Functional Requirements | 8 |
|  | 3.4 Technology Stack | 9 |

iii

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGENO.** |
|  |  |  |
| **4.** | **SYSTEM ARCHITECTURE** | 11 |
|  |  | 14 |
| 4.1 Module Design Specification | | |
| 4.2 Program Design Language 16  **5. SYSTEM IMPLEMENTATION** 17 | | |
| **6.** | **SYSTEM TESTING** | 22  23 |
|  | 6.1 Unit Testing | 24 |
|  | 6.2 Integration Testing | 25 |
|  | 6.3 Test Cases & Reports/ Performance Analysis |  |
| **7.** | **CONCLUSION** | 40 |
|  | 7.1 Conclusion and Future Enhancements | 40 |
|  | **APPENDICES** | 41 |
|  | A.1 Sample Screens | 42 |
|  | A.2 Publications | 49 |

**REFERENCES** 49

iv

## 

## LIST OF FIGURES

1.1.1 Structure of Data Mining 1

3.4.1 Python Working Architecture 10

3.4.2 Jupyter Notebook Working 10

4.1.2 Convolutional Neural Network 11

4.1.3Algorithms of CNN classifier model 1 (left) and model 2 (right) 12

4.1.4 VGG16 Architecture 13

4.1.5 VGG16 Working 13

4.1.6 Resnet Architecture 14

4.1.7(A)Normal patient 15

4.1.7(B) Pneumonia affected patient 15

4.1.8 Workflow for development of the model 16

6.2.1 Integration Testing 22

6.1.1 Unit Testing 23

A.1.3 Model VGG16 41

A.1.4 Model VGG16 Summary 41

A.1.5 Model VGG16 Model accuracy and loss 44

A.1.6 Resnet50 46

A.1.7 Resnest50 model summary 46

A.1.8 Resnest50 model accuracy and loss 48

v

1. **INTRODUCTION**
   1. **Overview**

**What is Data Mining?**

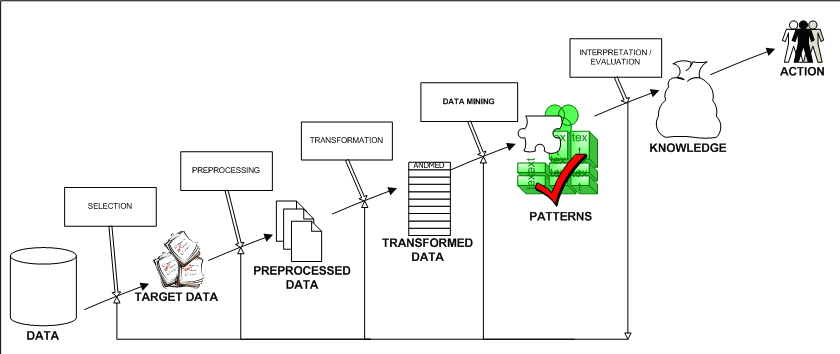


Fig 1.1.1

Structure of Data Mining

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

**How Data Mining Works?**

While large-scale information technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks. **Generally, any of four types of relationships are sought:**

* **Classes**: Stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.
* **Clusters**: Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities.
* **Associations**: Data can be mined to identify associations. The beer-diaper example is an example of associative mining.

1

* **Sequential patterns**: Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

**Data mining consists of five major elements:**

1. Extract, transform, and load transaction data onto the data warehouse system.
2. Store and manage the data in a multidimensional database system.
3. Provide data access to business analysts and information technology professionals.
4. Analyze the data by application software.
5. Present the data in a useful format, such as a graph or table.

**Different levels of analysis are available:**

* **Artificial neural networks**: Non-linear predictive models that learn through training and resemble biological neural networks in structure.
* **Genetic algorithms**: Optimization techniques that use process such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution.
* **Decision trees**: Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID). CART and CHAID are decision tree techniques used for classification of a dataset. They provide a set of rules that you can apply to a new (unclassified) dataset to predict which records will have a given outcome. CART segments a dataset by creating 2-way splits while CHAID segments using chi square tests to create multi-way splits. CART typically requires less data preparation than CHAID.
* **Nearest neighbor method**: A technique that classifies each record in a dataset based on a combination of the classes of the *k* record(s) most similar to it in a historical dataset (where *k*=1). Sometimes called the *k*-nearest neighbor technique.
* **Rule induction**: The extraction of useful if-then rules from data based on statistical significance.
* **Data visualization**: The visual interpretation of complex relationships in multidimensional data. Graphics tools are used to illustrate data relationships.

**Characteristics of Data Mining:**

* **Large quantities of data**: The volume of data so great it has to be analyzed by automated techniques e.g. satellite information, credit card transactions etc.
* **Noisy, incomplete data**: Imprecise data is the characteristic of all data collection.
* **Complex data structure**: conventional statistical analysis not possible
* **Heterogeneous data stored in legacy systems**

**Benefits of Data Mining:**

1. It’s one of the most effective services that are available today. With the help of data mining, one can discover precious information about the customers and their behavior for a specific set of products and evaluate and analyze, store, mine and load data related to them

2

1. An analytical CRM model and strategic business related decisions can be made with the help of data mining as it helps in providing a complete synopsis of customers
2. An endless number of organizations have installed data mining projects and it has helped them see their own companies make an unprecedented improvement in their marketing strategies (Campaigns)
3. Data mining is generally used by organizations with a solid customer focus. For its flexible nature as far as applicability is concerned is being used vehemently in applications to foresee crucial data including industry analysis and consumer buying behaviors
4. Fast paced and prompt access to data along with economic processing techniques have made data mining one of the most suitable services that a company seek

**Advantages of Data Mining:**

### Marketing / Retail:

Data mining helps marketing companies build models based on historical data to predict who will respond to the new marketing campaigns such as direct mail, online marketing campaign…etc. Through the results, marketers will have appropriate approach to sell profitable products to targeted customers.

Data mining brings a lot of benefits to retail companies in the same way as marketing. Through market basket analysis, a store can have an appropriate production arrangement in a way that customers can buy frequent buying products together with pleasant. In addition, it also helps the retail companies offer certain discounts for particular products that will attract more customers.

### Finance / Banking

Data mining gives financial institutions information about loan information and credit reporting. By building a model from historical customer’s data, the bank and financial institution can determine good and bad loans. In addition, data mining helps banks detect fraudulent credit card transactions to protect credit card’s owner.

### Manufacturing

By applying data mining in operational engineering data, manufacturers can detect faulty equipments and determine optimal control parameters. For example semi-conductor manufacturers has a challenge that even the conditions of manufacturing environments at different wafer production plants are similar, the quality of wafer are lot the same and some for unknown reasons even has defects. Data mining has been applying to determine the ranges of control parameters that lead to the production of golden wafer. Then those optimal control parameters are used to manufacture wafers with desired quality.

### Governments

Data mining helps government agency by digging and analyzing records of financial transaction to build patterns that can detect money laundering or criminal activities.

1. **Law enforcement:**

Data mining can aid law enforcers in identifying criminal suspects as well as apprehending these criminals by examining trends in location, crime type, habit, and other patterns of behaviors.

3

1. **Researchers:**

Data mining can assist researchers by speeding up their data analyzing process; thus, allowing those more time to work on other projects.

**1.2 Problem Definition**

In present day, one in three deaths in India is caused due to pneumonia as reported by World Health Organization (WHO).Thus, developing an automatic system for detecting pneumonia would be beneficial for treating the disease without any delay particularly in remote areas. Due to the success of deep learning algorithms in analyzing medical images, Convolutional Neural Networks (CNNs) have gained much attention for disease classification. In this work, we appraise the functionality of pre-trained CNN models utilized as feature-extractors followed by different classifiers for the classification of abnormal and normal chest X-Rays. We analytically determine the optimal CNN model for the purpose.

4

**2. LITERATURE SURVEY**

The problem of classifying chest x-ray images into different classes has been significantly explored in the field of medical diagnosis. Many research papers have been published, tackling this problem.

1. Rajpurkar et al. trained a deep learning model to detect pneumonia in chest x-ray images on the dataset ChestX-ray14. Using ChXNet, which is a 121 layer CNN they classified chest x-ray images at a level exceeding practicing radiologists. Apart from detecting pneumonia, their model also detected 14 other diseases. They compared the performance of their model with practicing academic radiologists. Their model provides a state of the art performance and hopes to improve the delivery of healthcare. Guan Q. et al. developed an AG-CNN model approach to detect thorax disease from chest x-ray images. This research has been conducted on Chest X-ray 14 dataset. The classification was done using two branch attention guided CNN. The two branches being global and local pick up global and local cues to predict thorax disease. Heat maps are also used to train the CNN model. They compared their model’s performance with other models. Their approach outperformed various other models, having an average AUC of 0.871. [2]

2. Xu Y. et al. trained a CNN model for classification and segmentation of brain tumor images of large dimensions. This model uses data augmentation, feature selection, and feature pooling techniques. The accuracy of segmentation and classification of this model are 84% and 97.5% respectively. They presented their approach in MICCAI 2014 Brain Tumor Digital Pathology Challenge. Rubin et al. presented a dual CNN which performs large scale automatic recognition of front and lateral images of chest x-ray on MIMIC-CXR dataset, which is the largest available dataset of chest X-rays till date. This neural network is used to detect common thorax disease. The dataset was divided into training data, testing data, and validation data. 70% of the data was used for training, 20% was used for testing, and 10% for validation. Their model has an average AUC of 0.721 and 0.668 for PA and AP, respectively. They aim to improve their model’s performance by using data augmentation and pixel normalization techniques to provide aid to the workflow of the process to identify common thorax disease. [61]

3. Lakhani P. et al. trained a deep CNN for automated classification of pulmonary tuberculosis from chest radiographs. AlexNet and GoogLeNet, which are dual CNNs, were used for classification purposes. The dataset was pre-processed before evaluation. Their model had an astounding AUC of 0.99. Their model had a specificity 100% and sensitivity of 97.3%. CNN is used to detect and classify abnormalities in frontal chest radiographs using deep convolutional neural networks was trained by Cicero M. et al. The input images were of the size 256X256 pixels. The AUC of the model is 0.964 with an average specificity and sensitivity of 91% showing that deep convolutional neural networks can be developed with high classification accuracy and can help in the diagnosis procedure.[27]

4. Anthimopoulos M. et al. presented a CNN model to identify interstitial lung disease patterns. Their model consists of 5 convolutional layers, employing leaky ReLU activation function, average pooling layers, and three dense layers. The dataset on which it was trained contains seven classes, and the dataset has 14696 images. Their model had an accuracy of 85.5%. They hope to extend their model to classify 3D images to be a supportive tool for diagnostic purposes.[3]

5. Glozman T. et al. presented a transfer model, which is an extension to AlexNet to classify Alzheimer’s disease on the ADNI database. Data augmentation technique was employed to enhance the performance of the deep neural network. Cho Y. et al. presented an ISC method which is based on incremental learning. They used a dataset which comprised of cortical thickness data. Their model achieved a specificity of 93% on the classification of AD patients from HC subjects. [14]

5

6. Hemanth D. J. et al. dealt with the problem of the high convergence time period for ANNs. They presented two models, which are MCPN and MKNN, which classified MR images iteration free with high accuracy. They used sensitivity and specificity as performance measures for their models. Three new deep

CNN models were presented by Szegedy C. et al. which are variants of the combination of Inception and ResNet models. Their model showed promising results. They achieved 3.08% top 5 error on the testing dataset of ImageNet classification challenge. The ability of deep CNN models to achieve groundbreaking results on complex datasets was shown by Krizhevsky A. et al. achieving a top 5 error percent of 17%. The dataset used was the ImageNet dataset. Dropout increased the efficiency of the model considerably. Their network contains 60 million total number of parameters and has five convolutional layers and max-pooling layers. Three fully connected layers were used to provide optimum results. State of the art deep CNN model, which was submitted to ILSVRC 2014 developed by Simonyan K. et al. which was also used in this paper achieved a 92.7% top-5 test accuracy on the ImageNet dataset. Their model has multiple variants, being widely used for classification purposes in medical research. This model was the first model to introduce small kernel sized filters one after the other instead of using one large kernel sized filter. He K. et al. presented the approach of residual learning for classification purposes. This model introduces shortcut connections to improve performance. The dataset used for training and testing was the ImageNet dataset. This model was submitted to ILSVRC 2015. [17]

7. Jaiswal, A. et al. presented a Mask-RCNN based identification model for pixel-wise segmentation incorporating global and local features. They introduced critical alterations in the training process merging bounding boxes from multiple models. The performances evaluated on chest radiograph dataset which depict potential pneumonia causes. The quality of images is an imperative factor in diagnosis of disease. Elhoseny, M. et al proposed an optimal bilateral filter to remove noise from the medical images. A detailed review is presented by Chandra, T et al. analyzing the filters to reduce quantum noise in chest x-ray images.[20]

6

**3. SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

Antibiotics are used to recover the bacterial infected pneumonia while viral infected patient need the different medication and supportive care for recovery of the diseases. Chest X-ray (CXR) analysis is the chosen as one of the most preferably for the medical practitioner for diagnosing the pneumonia disease. Accurately identifying and categorizing the pneumonia subtypes is an important and challenging clinical task, and automated methods can be used to save time and reduce error.  Deep learning techniques can be used to recognize and identify the type of pneumonia infection in the patient and proper clinical treatment can be provided towards making the patient cure from this kind of respiratory infection. Deep Siamese based CNN method can be associated with the radiographic examination towards detecting the particular type of pneumonia.

DISADVANTAGES OF EXISTING SYSTEM:

Despite of several advantages of X-ray imaging, still in some cases it is not possible to identify the correct region of interest in the radiographic image for detecting the diseases.  It is also observed that as the diagnostic accuracy of automated method reach the human level. This makes researchers to develop automated methods for detection of diseases through radiography imagery. Automated detection method provides an add on support to the clinical experts towards a ease of diagnosing the disease

**3.2 PROPOSED SYSTEM**

An optimum solution for the detection of pneumonia from chest X-rays is proposed in this work. Data augmentation was used to address the problem of the limited dataset, and then, state-of-the-art deep learning models, as discussed and was fine-tuned for pneumonia classification. Then, predictions from these models were combined, using a weighted classifier to compute the final prediction.

ADVANTAGES OF PROPOSED SYSTEM:

To sum up, our work has three major contributions as follows:

* Have trained our model for both the right and left part of chest X-ray that makes the process robust towards better classification. Overall our approach has provided the add on benefits to inexperienced practitioners in towards diagnosing the specific type of pneumonia in the early stage.

7

**3.3 REQUIREMENT ANALYSIS AND SPECIFICATION**

**3.3.1 INPUT REQUIREMENTS**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**3.3.2 OUTPUT REQUIREMENTS**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

8

**3.4 TECHNOLOGY STACK**

**HARDWARE REQUIREMENTS:**

* System : Intel Core 2 Duo.
* Hard Disk        : 40 GB.
* Monitor : 15 VGA Color.
* Mouse : Logitech.
* Ram : 2GB.

**SOFTWARE CONFIGURATION:-**

* Operating System : Windows 8/Windows 10
* Programming Language : Python
* Front End : Jupyter Notebook.

**PYTHON (PROGRAMMING LANGUAGE)**

Python is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with its notable use of [significant indentation](https://en.wikipedia.org/wiki/Off-side_rule). Its [language constructs](https://en.wikipedia.org/wiki/Language_construct) as well as its [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help [programmers](https://en.wikipedia.org/wiki/Programmers) write clear, logical code for small and large-scale projects.

Python is [dynamically-typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigms), including [structured](https://en.wikipedia.org/wiki/Structured_programming) (particularly, [procedural](https://en.wikipedia.org/wiki/Procedural_programming)), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library)

CHARACTERATICS OF PYTHON

* Easy to code
* Free and Open Source
* Object-Oriented Language
* GUI Programming Support
* High-Level Language
* Python is Portable language

9

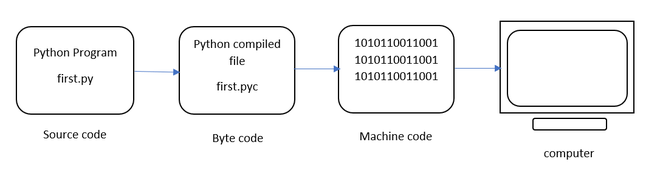


Fig 3.4.1 Python Working Architecture

**THE JUPYTER NOTEBOOK**

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at [Project Jupyter](http://jupyter.org/).

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

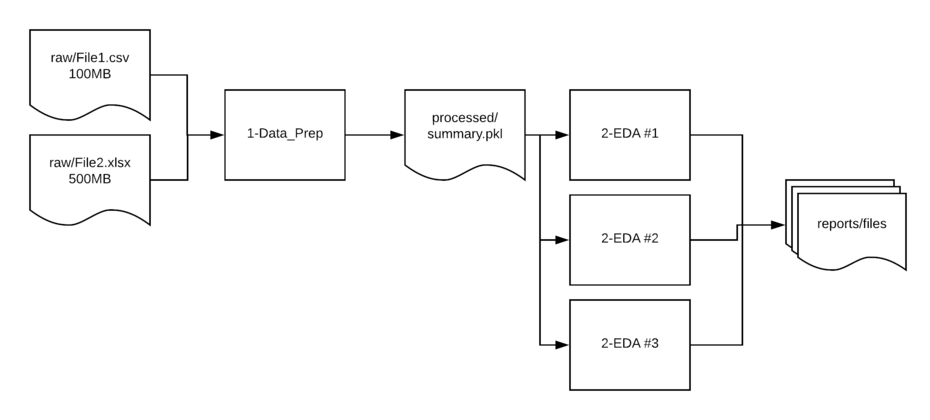
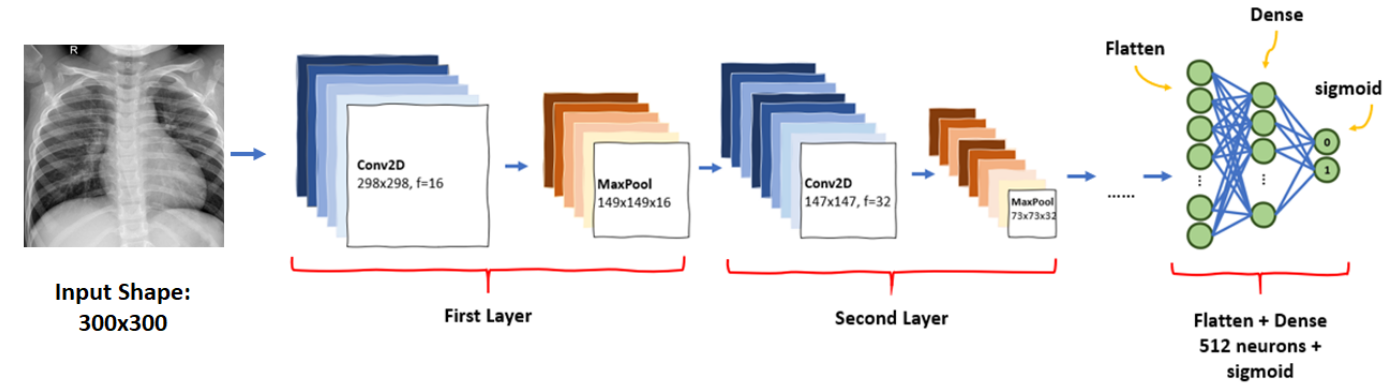


Fig 3.4.2 Jupyter Notebook Working

10

**4. SYSTEM ARCHITECTURE**

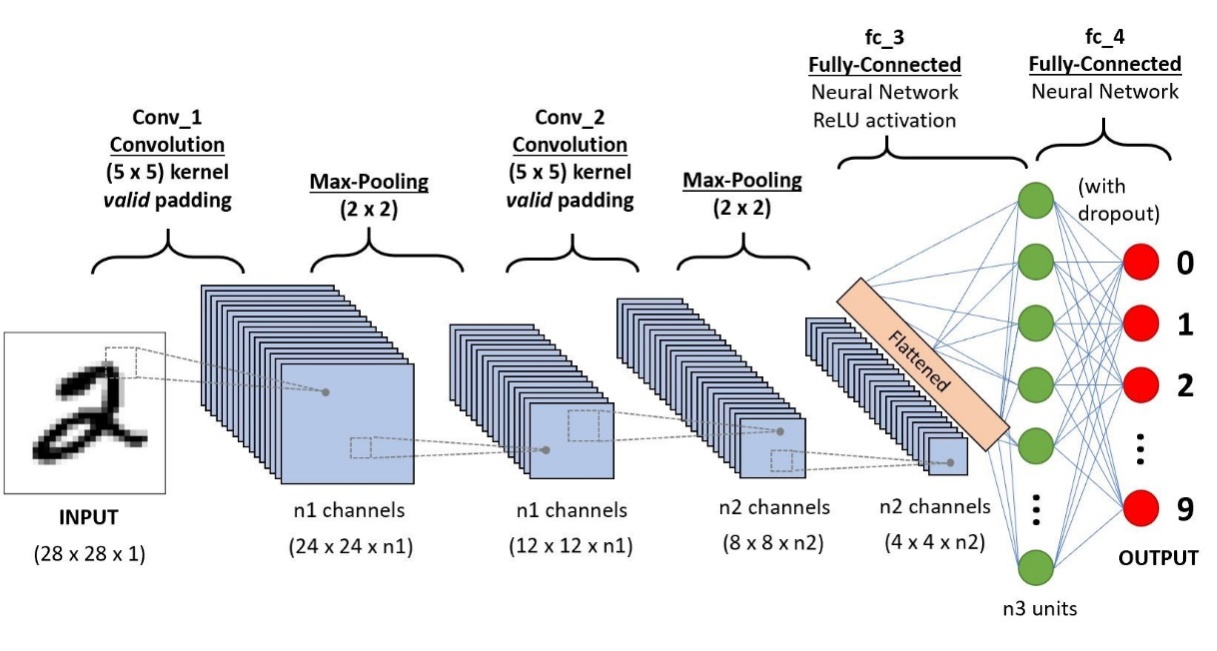
**4.1 ARCHITECTURE OVERVIEW**

****

**Fig 4.1.1 Architecture Diagram**

**CONVOLUTIONAL NEURAL NETWORK**

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

****

**Fig 4.1.2 Convolutional Neural Network**

11

Algorithm of CNN classifiers. The algorithms used in the convolutional neural network classifiers have been explained in Figs. 1 and 2. Figure 3 shows the flowchart of the overall schema of research. The number of epochs for all the classifier models presented in this paper was fixed at 20 after training and testing several CNN models over the course of research. Classifier models trained for more number of epochs have showed overfitting. Several optimizer functions were also trained and studied. Adam optimizer function was finalized to be used for all classifiers after it gave the best results. Initially, a simple classifier model with convolutional layer of image size set to 64 \* 64, 32 feature maps and employing ReLU activation function was trained. Fully connected dense layer with 128 perceptrons was utilized. To improve the result, the second classifier model was trained with one more convolutional layer of 64 feature maps for better feature extraction. The number of perceptrons in dense layer was also doubled to 256, so that better learning could be achieved. The third model was trained for three convolutional layers with 128 feature maps in third convolutional layer for more detailed feature extraction. Dense layer was kept unchanged. Dropout layer was introduced at 0.3, and learning rate of optimizer was

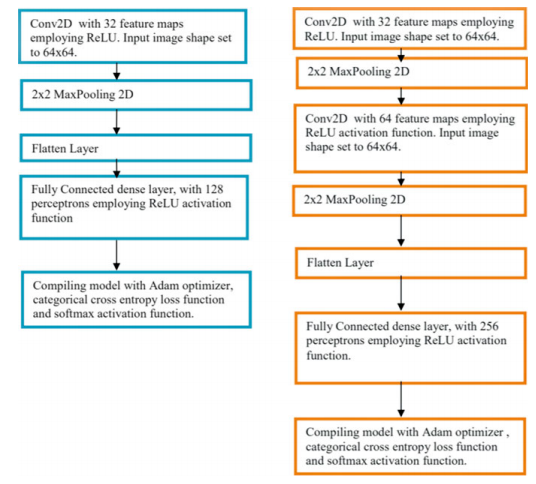


Fig. 4.1.3 Algorithms of CNN classifier model 1 (left) and model 2 (right)

12

**VGG16 ARCHITECTURE**

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous model submitted to ILSVRC-2014. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another. VGG16 was trained for weeks and was using NVIDIA Titan Black GPU’s.



Fig 4.1.4 VGG16 Architecture



Fig 4.1.5 VGG16 Working

13

**RESNET**

A residual neural network (ResNet) is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections, or shortcuts to jump over some layers. Typical ResNet models are implemented with double- or triple- layer skips that contain nonlinearities (ReLU) and batch normalization in between. An additional weight matrix may be used to learn the skip weights; these models are known as HighwayNets. Models with several parallel skips are referred to as DenseNets.In the context of residual neural networks, a non-residual network may be described as a plain network.

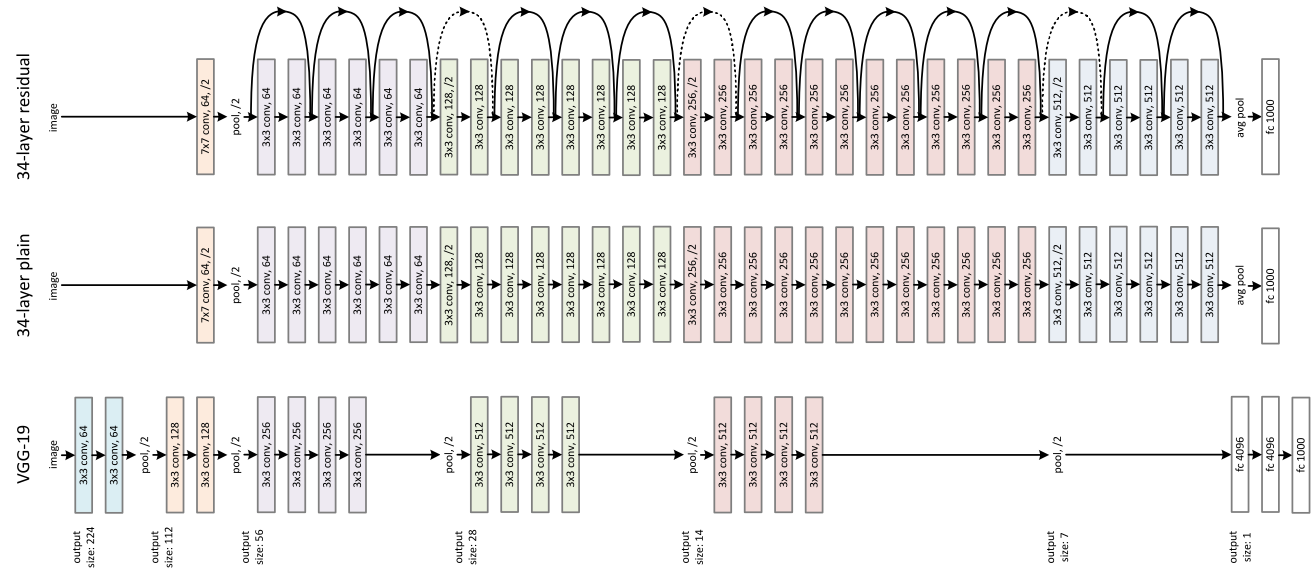


Fig 4.1.6 Resnet Architecture

**4.1 MODULE DESIGN SPECIFICATION**

**MODULES:**

* Dataset Collection
* Data Preprocessing
* Splitting training and testing data
* Build CNN VGG16 AND RESNET50 model
* Testing and evaluate the model

**DATASET:**

The chest x-ray image dataset for pneumonia detection consists of two categories of images – Pneumonia and normal. A total of 5,216 images are assigned to the training set and 624 images are assigned to the test set.

14

        Fig 4.1.7(A)                                   Fig 4.1.7(B)

                                       Normal patient                        Pneumonia affected patient

**DATA PREPROCESSING:**

In this work, several preprocessing methods were employed to increase the quality of the image data. The rescale operation was used for image size reduction, since the images were of various dimensions. The input images are generated using Image Data Generator class and augmentation techniques are used to rotate, shift and zoom the images. The rotation range of 10 denotes the range in which the images were randomly rotated during training. Width shift is the horizontal translation of the images by 0.1 percent and height shift is the vertical translation of the images by 0.1 percent. The zoom range randomly zooms the images to the ratio of 0.2 percent.

**Workflow for development of the model:**

As illustrated above, the first step is data collection. It is followed by data preprocessing to prepare the data for application of algorithm. Then, VGG16 AND RESNET50 CNN architecture is built to classify images as pneumonia and normal. The testing phase involves predicting the presence of pneumonia in X-ray image using the trained model. Finally, the performance of the model is evaluated using various metrics.

15

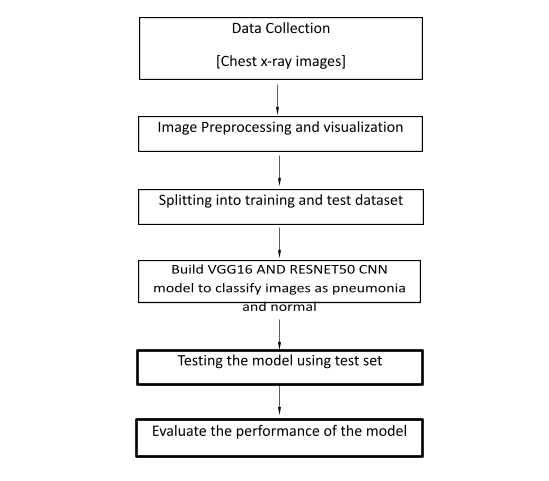


Fig 4.1.8 Workflow for development of the model

**4.2 PROGRAM DESIGN LANGUAGE**

**Algorithm:**

Convolutional Neural Network (CNN) is a deep learning algorithm that can recognize and classify features in images for computer vision. It is a multi-layer neural network designed to analyze visual inputs and perform image classification. It assigns importance to various aspects in the image and differentiate one from the other.This work implemented the VGG16 AND RESNET50 (Visual Geometry Group) model, which is a CNN transfer learning model.

The architecture consists of 16 layers. There are 2 contiguous blocks of 2 convolution layers, each block followed by max-pooling. Then, there are 3 contiguous blocks of 3 convolution layers, each block followed by max-pooling. At last, there are 3 dense layers. After developing the model, it is trained in batches of 32 images. The evaluation metric used for model is accuracy. The loss function used is sparse categorical cross entropy because the classes are mutually exclusive (i.e) each image belongs exactly to one class. The optimizer function used in the model is Adam optimize.

16

**5. SYSTEM IMPLEMENTATION**

**MODEL VGG16**

import os

import zipfile

import numpy as np

import tensorflow as tf

import matplotlib.pyplot as plt

from glob import glob

from tqdm import tqdm\_notebook

from keras.preprocessing.image import ImageDataGenerator

from keras.layers import Input, Lambda, Dense, Flatten

from keras.models import load\_model, Model

from keras.applications.vgg16 import VGG16

from keras.applications.vgg16 import preprocess\_input

%matplotlib inline

tf.\_\_version\_\_

from google.colab import drive

drive.mount('/content/drive')

# setup file structure

base\_dir = "/content/drive/MyDrive/chest\_xray/chest\_xray"

train\_dir = "/content/drive/MyDrive/chest\_xray/chest\_xray/train/"

test\_dir = "/content/drive/MyDrive/chest\_xray/chest\_xray/test/"

val\_dir = "/content/drive/MyDrive/chest\_xray/chest\_xray/val/"

print("Number of images in Train is {}".format(len(glob(train\_dir + "\*/\*"))))

print("Number of images in Test is {}".format(len(glob(test\_dir + "\*/\*"))))

print("Number of images in Validation is {}".format(len(glob(val\_dir + "\*/\*"))))

Number of images in Train is 5216

Number of images in Test is 624

Number of images in Validation is 16

IMG\_SHAPE = (224, 224,3)

base\_model = VGG16(input\_shape=IMG\_SHAPE, include\_top=False, weights='imagenet')

17

Downloading data from <https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5>

58892288/58889256 [==============================] - 1s 0us/step

model.compile(  loss='categorical\_crossentropy',

  optimizer='adam',

  metrics=['accuracy'])

train\_datagen = ImageDataGenerator(rescale = 1./255,

                                   shear\_range = 0.2,

                                   zoom\_range = 0.2,

                                   horizontal\_flip = True)

test\_datagen = ImageDataGenerator(rescale = 1./255)

training\_set = train\_datagen.flow\_from\_directory(train\_dir,

                                                 target\_size = (224, 224),

                                                 batch\_size = 32,

                                                 class\_mode = 'categorical')

test\_set = test\_datagen.flow\_from\_directory(test\_dir,

                                            target\_size = (224, 224),

                                            batch\_size = 32,

                                            class\_mode = 'categorical')

valid\_loss, valid\_accuracy = model.evaluate\_generator(test\_set)

plt.plot(history.history['loss'])

plt.plot(history.history['val\_loss'])

plt.title('model loss')

plt.ylabel('loss')

plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left')

plt.show()

plt.plot(history.history['accuracy'])

plt.plot(history.history['val\_accuracy'])

plt.title('model accuracy')

plt.ylabel('accuracy')

plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left')

plt.show()

18

# summarize history for loss

plt.plot(history.history['loss'])

plt.plot(history.history['val\_loss'])

plt.title('model loss')

plt.ylabel('loss')

plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left')

plt.show()

**RESNET50**

import os

import zipfile

import numpy as np

#import tensorflow as tf

import matplotlib.pyplot as plt

from glob import glob

from tqdm import tqdm\_notebook

from keras.preprocessing.image import ImageDataGenerator

from keras.layers import Input, Lambda, Dense, Flatten

from keras.models import load\_model, Model

from keras.applications.resnet50 import ResNet50

from keras.applications.resnet50 import preprocess\_input

%matplotlib inline

tf.\_\_version\_\_

base\_dir = "/content/drive/MyDrive/Kaggel/pneumonia\_data/chest\_xray"

train\_dir = os.path.join(base\_dir, "train/")

test\_dir = os.path.join(base\_dir, "test/")

val\_dir = os.path.join(base\_dir, "val/")

print("Number of images in Train is {}".format(len(glob(train\_dir + "\*/\*"))))

print("Number of images in Test is {}".format(len(glob(test\_dir + "\*/\*"))))

print("Number of images in Validation is {}".format(len(glob(val\_dir + "\*/\*"))))

Number of images in Train is 5216

Number of images in Test is 624

Number of images in Validation is 16

IMG\_SHAPE = (224, 224,3)

19

base\_model = ResNet50(input\_shape=IMG\_SHAPE, include\_top=False, weights='imagenet')

Downloading data from <https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5>

94773248/94765736 [==============================] - 1s 0us/step

base\_model.trainable = False

base\_model.output

# useful for getting number of classes

folders = glob(train\_dir+'/\*')

# our layers - you can add more if you want

x = Flatten()(base\_model.output)

# x = Dense(1000, activation='relu')(x)

prediction = Dense(len(folders), activation='softmax')(x)

model = Model(inputs=base\_model.input, outputs=prediction)

model.summary()

model.compile(  loss='categorical\_crossentropy',

  optimizer='adam',

  metrics=['accuracy'])

train\_datagen = ImageDataGenerator(rescale = 1./255,

                                   shear\_range = 0.2,

                                   zoom\_range = 0.2,

                                   horizontal\_flip = True)

test\_datagen = ImageDataGenerator(rescale = 1./255)

training\_set = train\_datagen.flow\_from\_directory(train\_dir,

                                                 target\_size = (224, 224),

                                                 batch\_size = 32,

                                                 class\_mode = 'categorical')

test\_set = test\_datagen.flow\_from\_directory(test\_dir,

                                            target\_size = (224, 224),

                                            batch\_size = 32,

                                            class\_mode = 'categorical')

20

valid\_loss, valid\_accuracy = model.evaluate\_generator(test\_set)

print("Accuracy after transfer learning: {}".format(valid\_accuracy))

plt.plot(history.history['loss'])

plt.plot(history.history['val\_loss'])

plt.title('model loss')

plt.ylabel('loss')

plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left')

plt.show()

plt.plot(history.history['accuracy'])

plt.plot(history.history['val\_accuracy'])

plt.title('model accuracy')

plt.ylabel('accuracy')

plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left')

plt.show()

# summarize history for loss

plt.plot(history.history['loss'])

plt.plot(history.history['val\_loss'])

plt.title('model loss')

plt.ylabel('loss')

plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left')

plt.show()

21

**6. SYSTEM TESTING**

            The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

**6.1 UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

***Test strategy and approach***

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

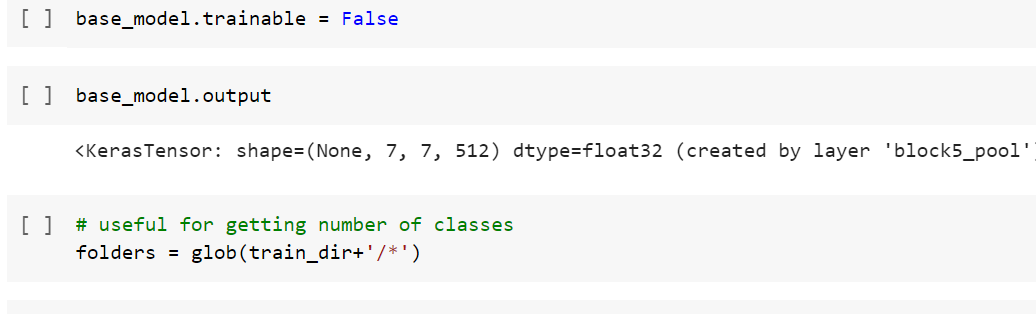


Fig 6.1.1 Unit Testing

22

**6.2 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually run as one program.  Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at   exposing the problems that arise from the combination of components.

****

**Fig 6.2.1 Integration Testing**

23

**6.3 TEST CASES & REPORTS / PERFORMANCE ANALYSIS**

**6.3.1 Model VGG16**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

input\_1 (InputLayer) [(None, 224, 224, 3)] 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block1\_conv1 (Conv2D) (None, 224, 224, 64) 1792

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block1\_conv2 (Conv2D) (None, 224, 224, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block1\_pool (MaxPooling2D) (None, 112, 112, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block2\_conv1 (Conv2D) (None, 112, 112, 128) 73856

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block2\_conv2 (Conv2D) (None, 112, 112, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block2\_pool (MaxPooling2D) (None, 56, 56, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block3\_conv1 (Conv2D) (None, 56, 56, 256) 295168

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block3\_conv2 (Conv2D) (None, 56, 56, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block3\_conv3 (Conv2D) (None, 56, 56, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block3\_pool (MaxPooling2D) (None, 28, 28, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block4\_conv1 (Conv2D) (None, 28, 28, 512) 1180160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block4\_conv2 (Conv2D) (None, 28, 28, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block4\_conv3 (Conv2D) (None, 28, 28, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block4\_pool (MaxPooling2D) (None, 14, 14, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block5\_conv1 (Conv2D) (None, 14, 14, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block5\_conv2 (Conv2D) (None, 14, 14, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block5\_conv3 (Conv2D) (None, 14, 14, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

block5\_pool (MaxPooling2D) (None, 7, 7, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten (Flatten) (None, 25088) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 2) 50178

=================================================================

Total params: 14,764,866

Trainable params: 50,178

Non-trainable params: 14,714,688

24

**6.3.2Resnet50**

Model: "model\_3"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param # Connected to

==================================================================================================

input\_6 (InputLayer) [(None, 224, 224, 3) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1\_pad (ZeroPadding2D) (None, 230, 230, 3) 0 input\_6[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1\_conv (Conv2D) (None, 112, 112, 64) 9472 conv1\_pad[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1\_bn (BatchNormalization) (None, 112, 112, 64) 256 conv1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1\_relu (Activation) (None, 112, 112, 64) 0 conv1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pool1\_pad (ZeroPadding2D) (None, 114, 114, 64) 0 conv1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pool1\_pool (MaxPooling2D) (None, 56, 56, 64) 0 pool1\_pad[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block1\_1\_conv (Conv2D) (None, 56, 56, 64) 4160 pool1\_pool[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block1\_1\_bn (BatchNormali (None, 56, 56, 64) 256 conv2\_block1\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block1\_1\_relu (Activation (None, 56, 56, 64) 0 conv2\_block1\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block1\_2\_conv (Conv2D) (None, 56, 56, 64) 36928 conv2\_block1\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

25

conv2\_block1\_2\_bn (BatchNormali (None, 56, 56, 64) 256 conv2\_block1\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block1\_2\_relu (Activation (None, 56, 56, 64) 0 conv2\_block1\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block1\_0\_conv (Conv2D) (None, 56, 56, 256) 16640 pool1\_pool[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block1\_3\_conv (Conv2D) (None, 56, 56, 256) 16640 conv2\_block1\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block1\_0\_bn (BatchNormali (None, 56, 56, 256) 1024 conv2\_block1\_0\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block1\_3\_bn (BatchNormali (None, 56, 56, 256) 1024 conv2\_block1\_3\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block1\_add (Add) (None, 56, 56, 256) 0 conv2\_block1\_0\_bn[0][0]

conv2\_block1\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block1\_out (Activation) (None, 56, 56, 256) 0 conv2\_block1\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block2\_1\_conv (Conv2D) (None, 56, 56, 64) 16448 conv2\_block1\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block2\_1\_bn (BatchNormali (None, 56, 56, 64) 256 conv2\_block2\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block2\_1\_relu (Activation (None, 56, 56, 64) 0 conv2\_block2\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block2\_2\_conv (Conv2D) (None, 56, 56, 64) 36928 conv2\_block2\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block2\_2\_bn (BatchNormali (None, 56, 56, 64) 256 conv2\_block2\_2\_conv[0][0]

26

conv2\_block2\_2\_relu (Activation (None, 56, 56, 64) 0 conv2\_block2\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block2\_3\_conv (Conv2D) (None, 56, 56, 256) 16640 conv2\_block2\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block2\_3\_bn (BatchNormali (None, 56, 56, 256) 1024 conv2\_block2\_3\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block2\_add (Add) (None, 56, 56, 256) 0 conv2\_block1\_out[0][0]

conv2\_block2\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block2\_out (Activation) (None, 56, 56, 256) 0 conv2\_block2\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block3\_1\_conv (Conv2D) (None, 56, 56, 64) 16448 conv2\_block2\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block3\_1\_bn (BatchNormali (None, 56, 56, 64) 256 conv2\_block3\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block3\_1\_relu (Activation (None, 56, 56, 64) 0 conv2\_block3\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block3\_2\_conv (Conv2D) (None, 56, 56, 64) 36928 conv2\_block3\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block3\_2\_bn (BatchNormali (None, 56, 56, 64) 256 conv2\_block3\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block3\_2\_relu (Activation (None, 56, 56, 64) 0 conv2\_block3\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block3\_3\_conv (Conv2D) (None, 56, 56, 256) 16640 conv2\_block3\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block3\_3\_bn (BatchNormali (None, 56, 56, 256) 1024 conv2\_block3\_3\_conv[0][0]

27

conv2\_block3\_add (Add) (None, 56, 56, 256) 0 conv2\_block2\_out[0][0]

conv2\_block3\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2\_block3\_out (Activation) (None, 56, 56, 256) 0 conv2\_block3\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block1\_1\_conv (Conv2D) (None, 28, 28, 128) 32896 conv2\_block3\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block1\_1\_bn (BatchNormali (None, 28, 28, 128) 512 conv3\_block1\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block1\_1\_relu (Activation (None, 28, 28, 128) 0 conv3\_block1\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block1\_2\_conv (Conv2D) (None, 28, 28, 128) 147584 conv3\_block1\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block1\_2\_bn (BatchNormali (None, 28, 28, 128) 512 conv3\_block1\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block1\_2\_relu (Activation (None, 28, 28, 128) 0 conv3\_block1\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block1\_0\_conv (Conv2D) (None, 28, 28, 512) 131584 conv2\_block3\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block1\_3\_conv (Conv2D) (None, 28, 28, 512) 66048 conv3\_block1\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block1\_0\_bn (BatchNormali (None, 28, 28, 512) 2048 conv3\_block1\_0\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block1\_3\_bn (BatchNormali (None, 28, 28, 512) 2048 conv3\_block1\_3\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block1\_add (Add) (None, 28, 28, 512) 0 conv3\_block1\_0\_bn[0][0]

28

conv3\_block1\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block1\_out (Activation) (None, 28, 28, 512) 0 conv3\_block1\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block2\_1\_conv (Conv2D) (None, 28, 28, 128) 65664 conv3\_block1\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block2\_1\_bn (BatchNormali (None, 28, 28, 128) 512 conv3\_block2\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block2\_1\_relu (Activation (None, 28, 28, 128) 0 conv3\_block2\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block2\_2\_conv (Conv2D) (None, 28, 28, 128) 147584 conv3\_block2\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block2\_2\_bn (BatchNormali (None, 28, 28, 128) 512 conv3\_block2\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block2\_2\_relu (Activation (None, 28, 28, 128) 0 conv3\_block2\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block2\_3\_conv (Conv2D) (None, 28, 28, 512) 66048 conv3\_block2\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block2\_3\_bn (BatchNormali (None, 28, 28, 512) 2048 conv3\_block2\_3\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block2\_add (Add) (None, 28, 28, 512) 0 conv3\_block1\_out[0][0]

conv3\_block2\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block2\_out (Activation) (None, 28, 28, 512) 0 conv3\_block2\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block3\_1\_conv (Conv2D) (None, 28, 28, 128) 65664 conv3\_block2\_out[0][0]

29

conv3\_block3\_1\_bn (BatchNormali (None, 28, 28, 128) 512 conv3\_block3\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block3\_1\_relu (Activation (None, 28, 28, 128) 0 conv3\_block3\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block3\_2\_conv (Conv2D) (None, 28, 28, 128) 147584 conv3\_block3\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block3\_2\_bn (BatchNormali (None, 28, 28, 128) 512 conv3\_block3\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block3\_2\_relu (Activation (None, 28, 28, 128) 0 conv3\_block3\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block3\_3\_conv (Conv2D) (None, 28, 28, 512) 66048 conv3\_block3\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block3\_3\_bn (BatchNormali (None, 28, 28, 512) 2048 conv3\_block3\_3\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block3\_add (Add) (None, 28, 28, 512) 0 conv3\_block2\_out[0][0]

conv3\_block3\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block3\_out (Activation) (None, 28, 28, 512) 0 conv3\_block3\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block4\_1\_conv (Conv2D) (None, 28, 28, 128) 65664 conv3\_block3\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block4\_1\_bn (BatchNormali (None, 28, 28, 128) 512 conv3\_block4\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block4\_1\_relu (Activation (None, 28, 28, 128) 0 conv3\_block4\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block4\_2\_conv (Conv2D) (None, 28, 28, 128) 147584 conv3\_block4\_1\_relu[0][0]

30

conv3\_block4\_2\_bn (BatchNormali (None, 28, 28, 128) 512 conv3\_block4\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block4\_2\_relu (Activation (None, 28, 28, 128) 0 conv3\_block4\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block4\_3\_conv (Conv2D) (None, 28, 28, 512) 66048 conv3\_block4\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block4\_3\_bn (BatchNormali (None, 28, 28, 512) 2048 conv3\_block4\_3\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block4\_add (Add) (None, 28, 28, 512) 0 conv3\_block3\_out[0][0]

conv3\_block4\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3\_block4\_out (Activation) (None, 28, 28, 512) 0 conv3\_block4\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block1\_1\_conv (Conv2D) (None, 14, 14, 256) 131328 conv3\_block4\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block1\_1\_bn (BatchNormali (None, 14, 14, 256) 1024 conv4\_block1\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block1\_1\_relu (Activation (None, 14, 14, 256) 0 conv4\_block1\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block1\_2\_conv (Conv2D) (None, 14, 14, 256) 590080 conv4\_block1\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block1\_2\_bn (BatchNormali (None, 14, 14, 256) 1024 conv4\_block1\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block1\_2\_relu (Activation (None, 14, 14, 256) 0 conv4\_block1\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block1\_0\_conv (Conv2D) (None, 14, 14, 1024) 525312 conv3\_block4\_out[0][0]

31

conv4\_block1\_3\_conv (Conv2D) (None, 14, 14, 1024) 263168 conv4\_block1\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block1\_0\_bn (BatchNormali (None, 14, 14, 1024) 4096 conv4\_block1\_0\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block1\_3\_bn (BatchNormali (None, 14, 14, 1024) 4096 conv4\_block1\_3\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block1\_add (Add) (None, 14, 14, 1024) 0 conv4\_block1\_0\_bn[0][0]

conv4\_block1\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block1\_out (Activation) (None, 14, 14, 1024) 0 conv4\_block1\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block2\_1\_conv (Conv2D) (None, 14, 14, 256) 262400 conv4\_block1\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block2\_1\_bn (BatchNormali (None, 14, 14, 256) 1024 conv4\_block2\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block2\_1\_relu (Activation (None, 14, 14, 256) 0 conv4\_block2\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block2\_2\_conv (Conv2D) (None, 14, 14, 256) 590080 conv4\_block2\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block2\_2\_bn (BatchNormali (None, 14, 14, 256) 1024 conv4\_block2\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block2\_2\_relu (Activation (None, 14, 14, 256) 0 conv4\_block2\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block2\_3\_conv (Conv2D) (None, 14, 14, 1024) 263168 conv4\_block2\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block2\_3\_bn (BatchNormali (None, 14, 14, 1024) 4096 conv4\_block2\_3\_conv[0][0]

32

conv4\_block2\_add (Add) (None, 14, 14, 1024) 0 conv4\_block1\_out[0][0]

conv4\_block2\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block2\_out (Activation) (None, 14, 14, 1024) 0 conv4\_block2\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block3\_1\_conv (Conv2D) (None, 14, 14, 256) 262400 conv4\_block2\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block3\_1\_bn (BatchNormali (None, 14, 14, 256) 1024 conv4\_block3\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block3\_1\_relu (Activation (None, 14, 14, 256) 0 conv4\_block3\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block3\_2\_conv (Conv2D) (None, 14, 14, 256) 590080 conv4\_block3\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block3\_2\_bn (BatchNormali (None, 14, 14, 256) 1024 conv4\_block3\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block3\_2\_relu (Activation (None, 14, 14, 256) 0 conv4\_block3\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block3\_3\_conv (Conv2D) (None, 14, 14, 1024) 263168 conv4\_block3\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block3\_3\_bn (BatchNormali (None, 14, 14, 1024) 4096 conv4\_block3\_3\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block3\_add (Add) (None, 14, 14, 1024) 0 conv4\_block2\_out[0][0]

conv4\_block3\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block3\_out (Activation) (None, 14, 14, 1024) 0 conv4\_block3\_add[0][0]

33

conv4\_block4\_1\_conv (Conv2D) (None, 14, 14, 256) 262400 conv4\_block3\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block4\_1\_bn (BatchNormali (None, 14, 14, 256) 1024 conv4\_block4\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block4\_1\_relu (Activation (None, 14, 14, 256) 0 conv4\_block4\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block4\_2\_conv (Conv2D) (None, 14, 14, 256) 590080 conv4\_block4\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block4\_2\_bn (BatchNormali (None, 14, 14, 256) 1024 conv4\_block4\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block4\_2\_relu (Activation (None, 14, 14, 256) 0 conv4\_block4\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block4\_3\_conv (Conv2D) (None, 14, 14, 1024) 263168 conv4\_block4\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block4\_3\_bn (BatchNormali (None, 14, 14, 1024) 4096 conv4\_block4\_3\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block4\_add (Add) (None, 14, 14, 1024) 0 conv4\_block3\_out[0][0]

conv4\_block4\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block4\_out (Activation) (None, 14, 14, 1024) 0 conv4\_block4\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block5\_1\_conv (Conv2D) (None, 14, 14, 256) 262400 conv4\_block4\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block5\_1\_bn (BatchNormali (None, 14, 14, 256) 1024 conv4\_block5\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block5\_1\_relu (Activation (None, 14, 14, 256) 0 conv4\_block5\_1\_bn[0][0]

34

conv4\_block5\_2\_conv (Conv2D) (None, 14, 14, 256) 590080 conv4\_block5\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block5\_2\_bn (BatchNormali (None, 14, 14, 256) 1024 conv4\_block5\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block5\_2\_relu (Activation (None, 14, 14, 256) 0 conv4\_block5\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block5\_3\_conv (Conv2D) (None, 14, 14, 1024) 263168 conv4\_block5\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block5\_3\_bn (BatchNormali (None, 14, 14, 1024) 4096 conv4\_block5\_3\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block5\_add (Add) (None, 14, 14, 1024) 0 conv4\_block4\_out[0][0]

conv4\_block5\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block5\_out (Activation) (None, 14, 14, 1024) 0 conv4\_block5\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block6\_1\_conv (Conv2D) (None, 14, 14, 256) 262400 conv4\_block5\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block6\_1\_bn (BatchNormali (None, 14, 14, 256) 1024 conv4\_block6\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block6\_1\_relu (Activation (None, 14, 14, 256) 0 conv4\_block6\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block6\_2\_conv (Conv2D) (None, 14, 14, 256) 590080 conv4\_block6\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block6\_2\_bn (BatchNormali (None, 14, 14, 256) 1024 conv4\_block6\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block6\_2\_relu (Activation (None, 14, 14, 256) 0 conv4\_block6\_2\_bn[0][0]

35

conv4\_block6\_3\_conv (Conv2D) (None, 14, 14, 1024) 263168 conv4\_block6\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block6\_3\_bn (BatchNormali (None, 14, 14, 1024) 4096 conv4\_block6\_3\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block6\_add (Add) (None, 14, 14, 1024) 0 conv4\_block5\_out[0][0]

conv4\_block6\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv4\_block6\_out (Activation) (None, 14, 14, 1024) 0 conv4\_block6\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block1\_1\_conv (Conv2D) (None, 7, 7, 512) 524800 conv4\_block6\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block1\_1\_bn (BatchNormali (None, 7, 7, 512) 2048 conv5\_block1\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block1\_1\_relu (Activation (None, 7, 7, 512) 0 conv5\_block1\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block1\_2\_conv (Conv2D) (None, 7, 7, 512) 2359808 conv5\_block1\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block1\_2\_bn (BatchNormali (None, 7, 7, 512) 2048 conv5\_block1\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block1\_2\_relu (Activation (None, 7, 7, 512) 0 conv5\_block1\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block1\_0\_conv (Conv2D) (None, 7, 7, 2048) 2099200 conv4\_block6\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block1\_3\_conv (Conv2D) (None, 7, 7, 2048) 1050624 conv5\_block1\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block1\_0\_bn (BatchNormali (None, 7, 7, 2048) 8192 conv5\_block1\_0\_conv[0][0]

36

conv5\_block1\_3\_bn (BatchNormali (None, 7, 7, 2048) 8192 conv5\_block1\_3\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block1\_add (Add) (None, 7, 7, 2048) 0 conv5\_block1\_0\_bn[0][0]

conv5\_block1\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block1\_out (Activation) (None, 7, 7, 2048) 0 conv5\_block1\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block2\_1\_conv (Conv2D) (None, 7, 7, 512) 1049088 conv5\_block1\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block2\_1\_bn (BatchNormali (None, 7, 7, 512) 2048 conv5\_block2\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block2\_1\_relu (Activation (None, 7, 7, 512) 0 conv5\_block2\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block2\_2\_conv (Conv2D) (None, 7, 7, 512) 2359808 conv5\_block2\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block2\_2\_bn (BatchNormali (None, 7, 7, 512) 2048 conv5\_block2\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block2\_2\_relu (Activation (None, 7, 7, 512) 0 conv5\_block2\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block2\_3\_conv (Conv2D) (None, 7, 7, 2048) 1050624 conv5\_block2\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block2\_3\_bn (BatchNormali (None, 7, 7, 2048) 8192 conv5\_block2\_3\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block2\_add (Add) (None, 7, 7, 2048) 0 conv5\_block1\_out[0][0]

conv5\_block2\_3\_bn[0][0]

37

conv5\_block2\_out (Activation) (None, 7, 7, 2048) 0 conv5\_block2\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block3\_1\_conv (Conv2D) (None, 7, 7, 512) 1049088 conv5\_block2\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block3\_1\_bn (BatchNormali (None, 7, 7, 512) 2048 conv5\_block3\_1\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block3\_1\_relu (Activation (None, 7, 7, 512) 0 conv5\_block3\_1\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block3\_2\_conv (Conv2D) (None, 7, 7, 512) 2359808 conv5\_block3\_1\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block3\_2\_bn (BatchNormali (None, 7, 7, 512) 2048 conv5\_block3\_2\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block3\_2\_relu (Activation (None, 7, 7, 512) 0 conv5\_block3\_2\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block3\_3\_conv (Conv2D) (None, 7, 7, 2048) 1050624 conv5\_block3\_2\_relu[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block3\_3\_bn (BatchNormali (None, 7, 7, 2048) 8192 conv5\_block3\_3\_conv[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block3\_add (Add) (None, 7, 7, 2048) 0 conv5\_block2\_out[0][0]

conv5\_block3\_3\_bn[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv5\_block3\_out (Activation) (None, 7, 7, 2048) 0 conv5\_block3\_add[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_3 (Flatten) (None, 100352) 0 conv5\_block3\_out[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_3 (Dense) (None, 2) 200706 flatten\_3[0][0]

38

Total params: 23,788,418

Trainable params: 200,706

Non-trainable params: 23,587,712

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Performance Analysis:**

****

39

**7. CONCLUSION**

**7.1 CONCLUSION AND FUTURE ENHANCEMENT**

In this article, our goal is to propose a deep learning-based approach to classify pneumonia from chest X-ray images using transfer learning. In this framework, we adopted the transfer learning approach and used the pretrained architectures, VGG16 and ResNet trained on the ImageNet dataset, to extract features. These features were passed to the classifiers of respective models, and the output was collected from individual architectures. Finally, we employed an ensemble model that used pretrained models and outperformed all other models.

We observed that performance could be improved further, by increasing dataset size, using a data augmentation approach, and by using hand-crafted features, in future. Our findings support the notion that deep learning methods can be used to simplify the diagnostic process and improve disease management. While pneumonia diagnoses are commonly confirmed by a single doctor, allowing for the possibility of error, deep learning methods can be regarded as a two-way confirmation system. In this case, the decision support system provides a diagnosis based on chest X-ray images, which can then be confirmed by the attending physician, drastically minimizing both human and computer error. Our results suggest that deep learning methods can be used to improve diagnosis relative to traditional methods, which may improve the quality of treatment. When compared with the previous state-of-the-art methods, our approach can effectively detect the inflammatory region in chest X-ray images. Furthermore, the obtained results show that the Resnet50, gave highly performance (accuracy is more than 96%) against other architectures cited in this work (accuracy is around 85%).

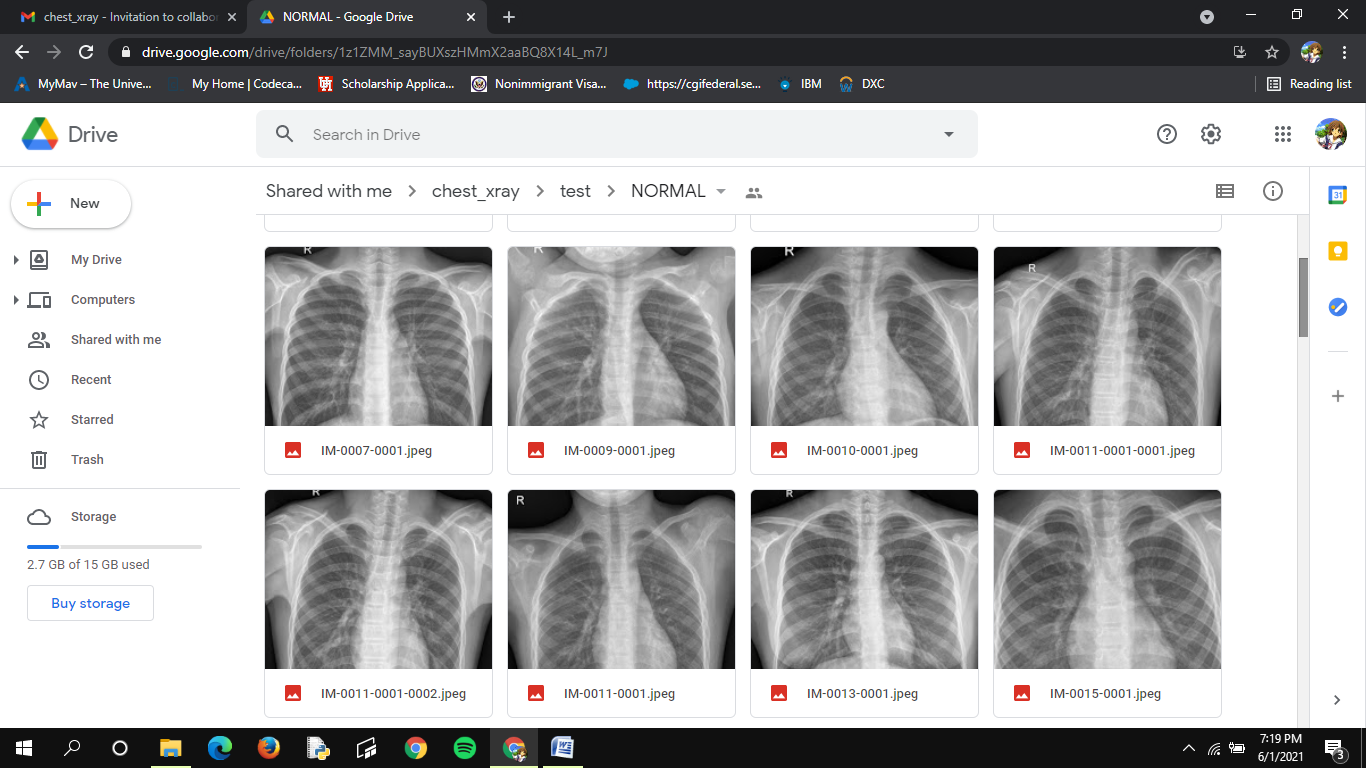
**Future Work**

 In the future this work could be extended to detect and classify X-ray images consisting of lung cancer and pneumonia. Distinguishing X-ray images that contain lung cancer and pneumonia has been a big issue in recent times, and our next approach should be to tackle this problem.

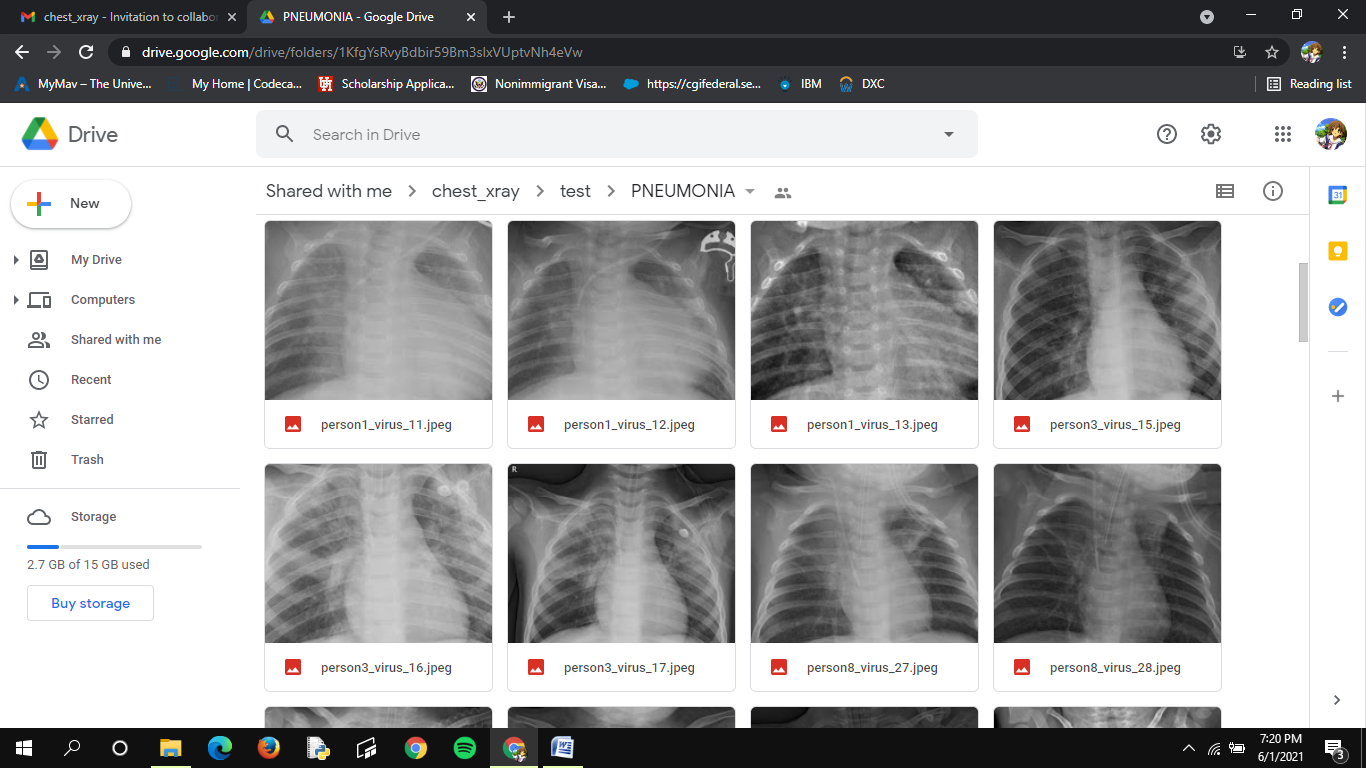
40

**APPENDICES**

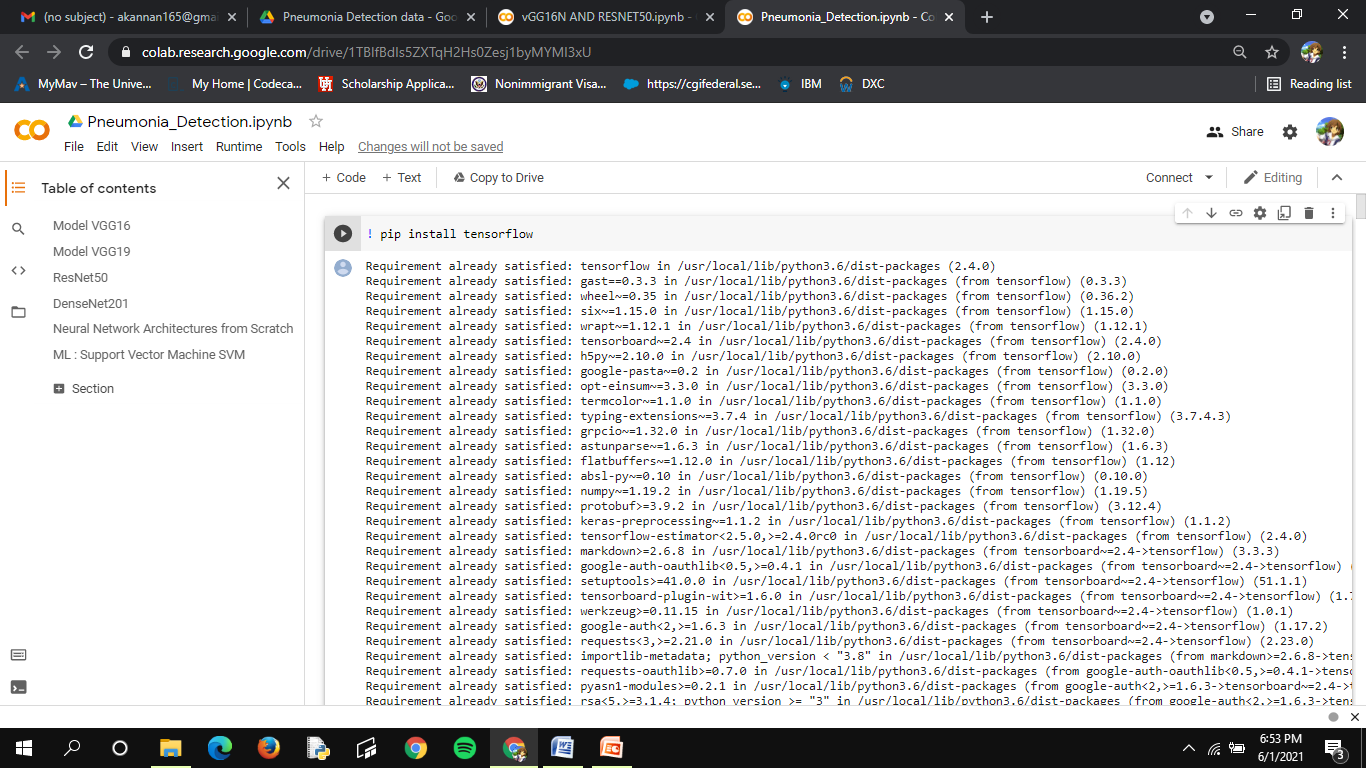
**A.1 SAMPLE SCREENS**

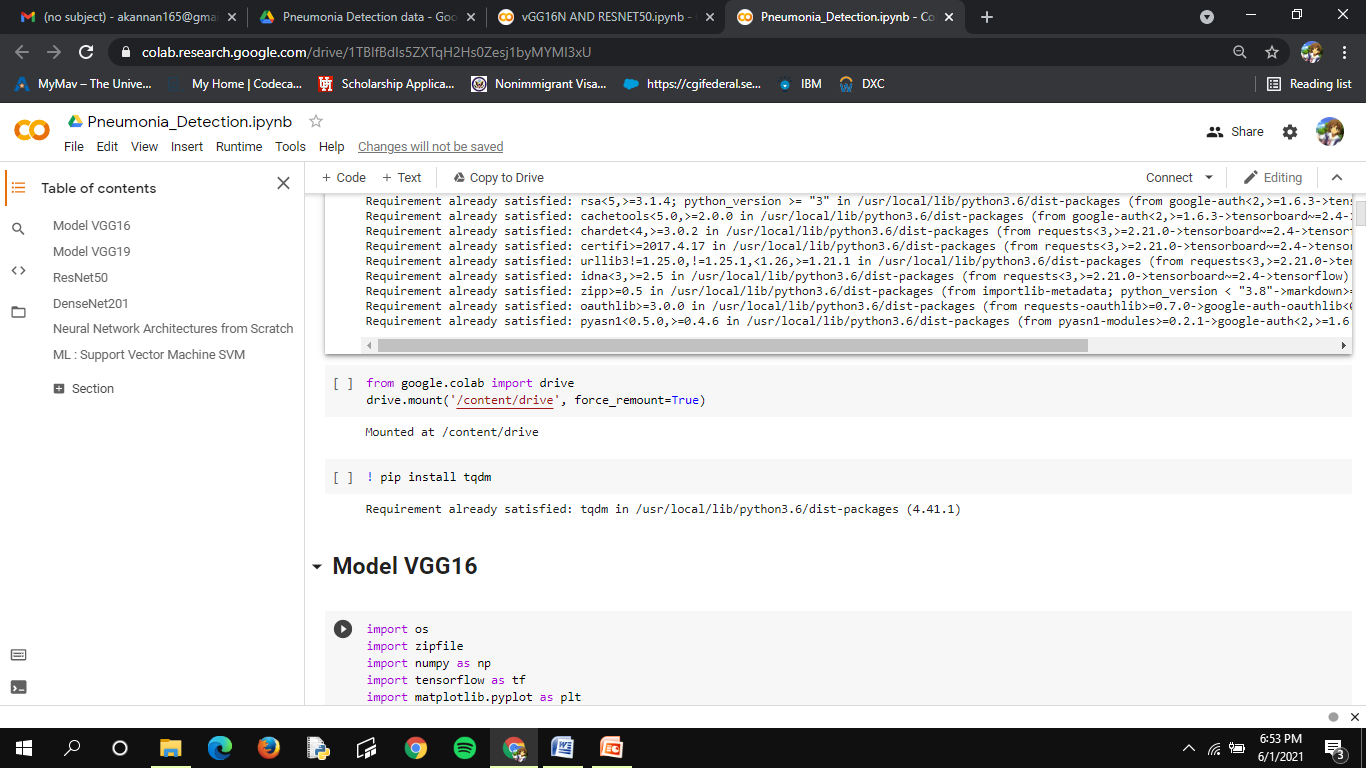
****

**Fig A.1.1 Pictures of normal lungs**

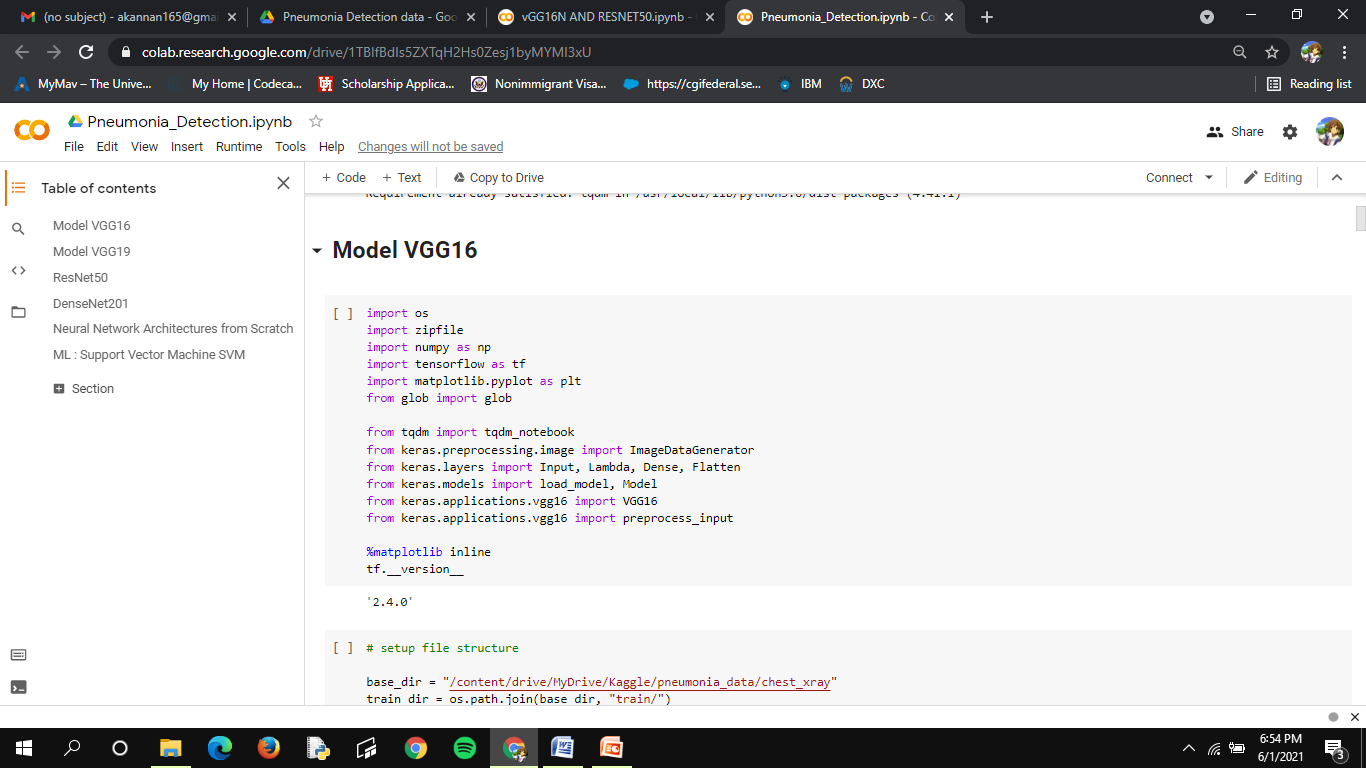
**Fig A.1.1 Pictures of Pneumonia affected lungs**

41

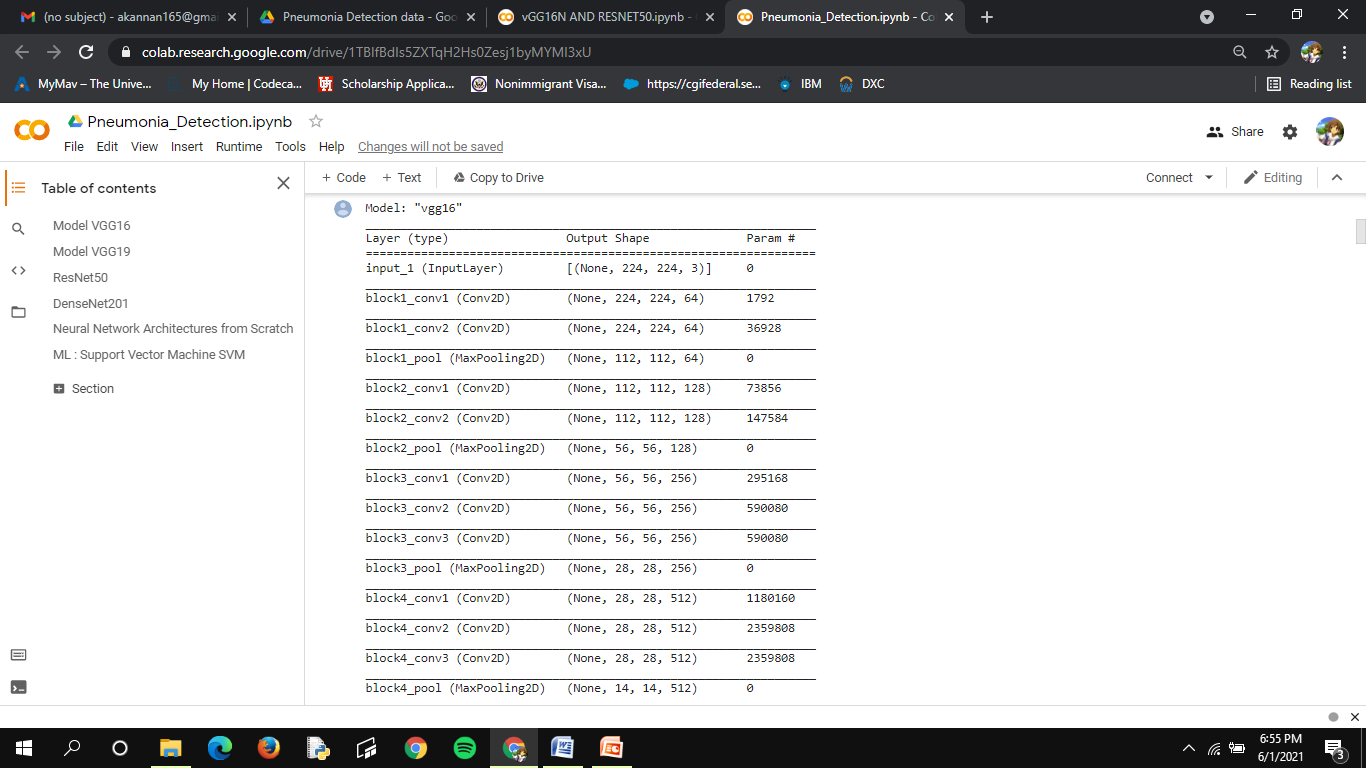
****

****

42

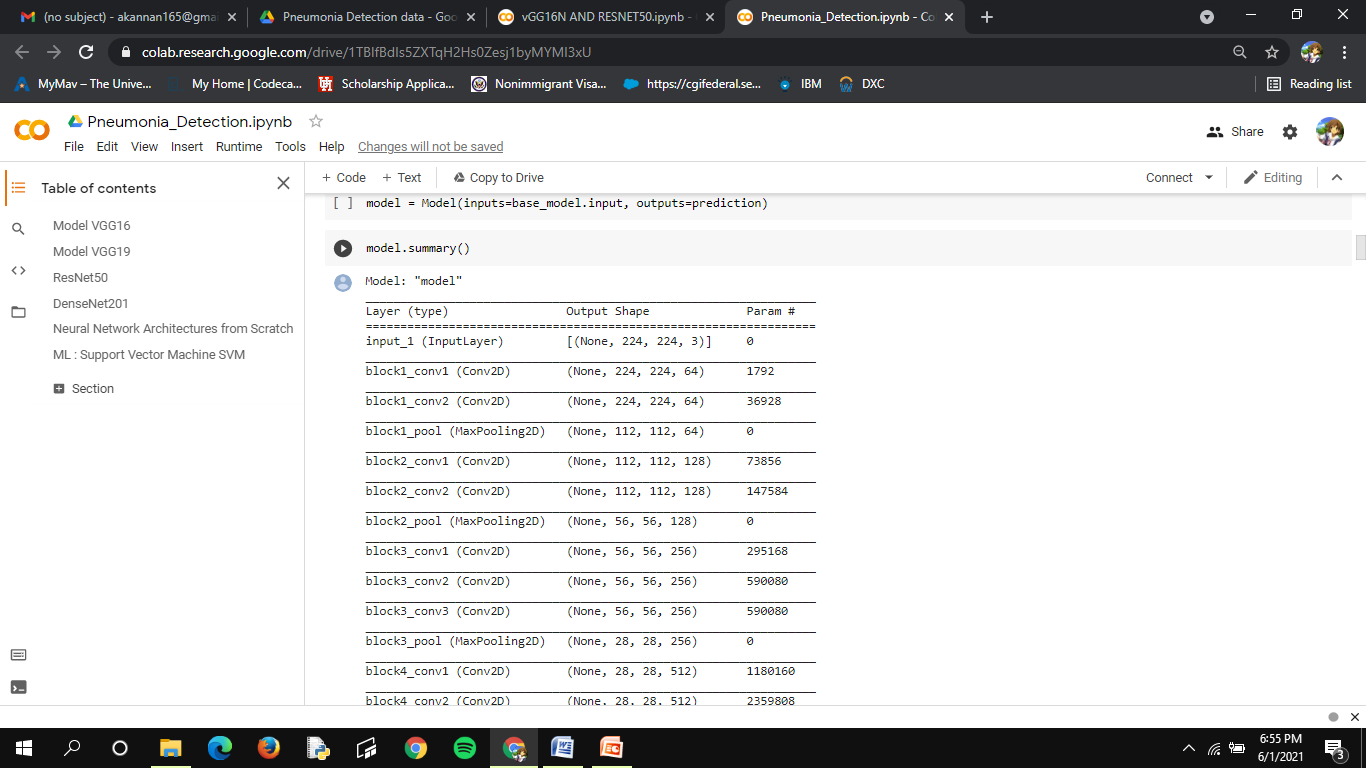
****

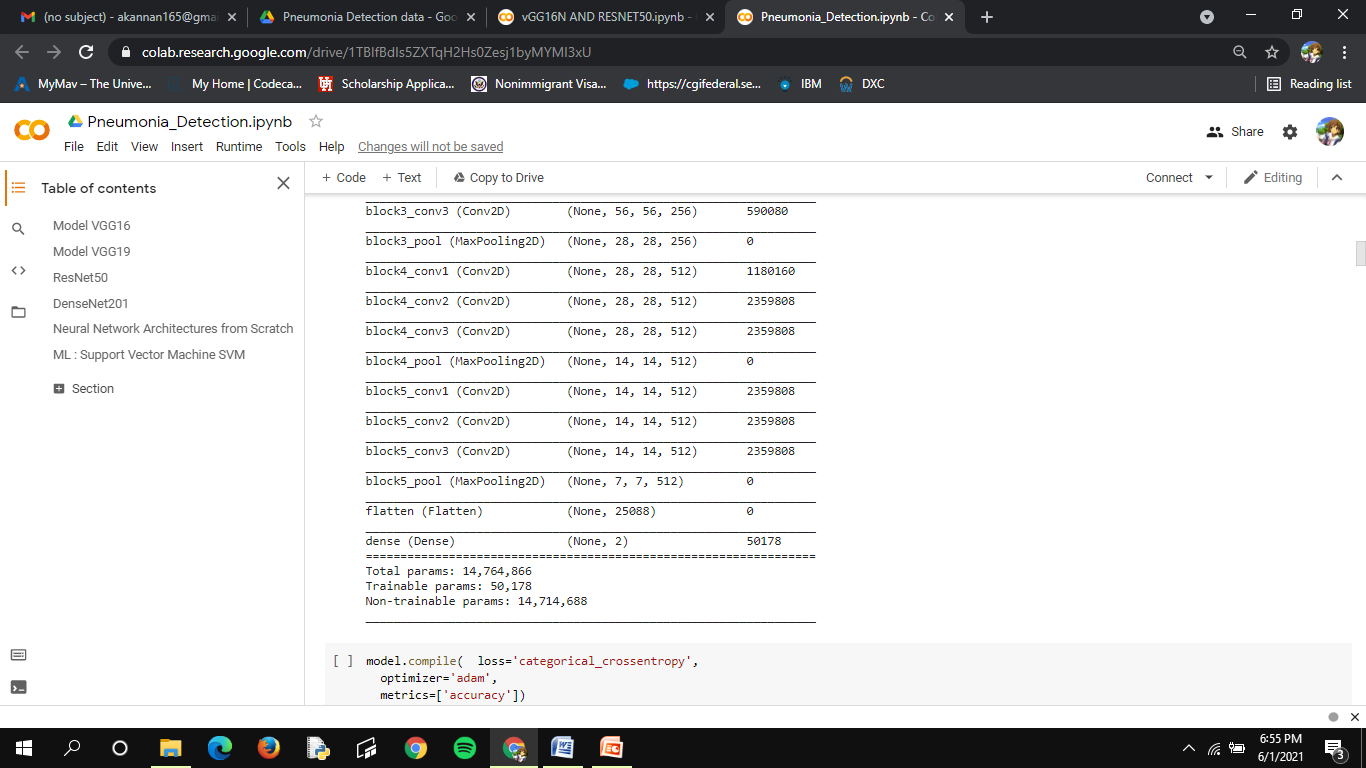
**Fig A.1.3 Model VGG16**

****

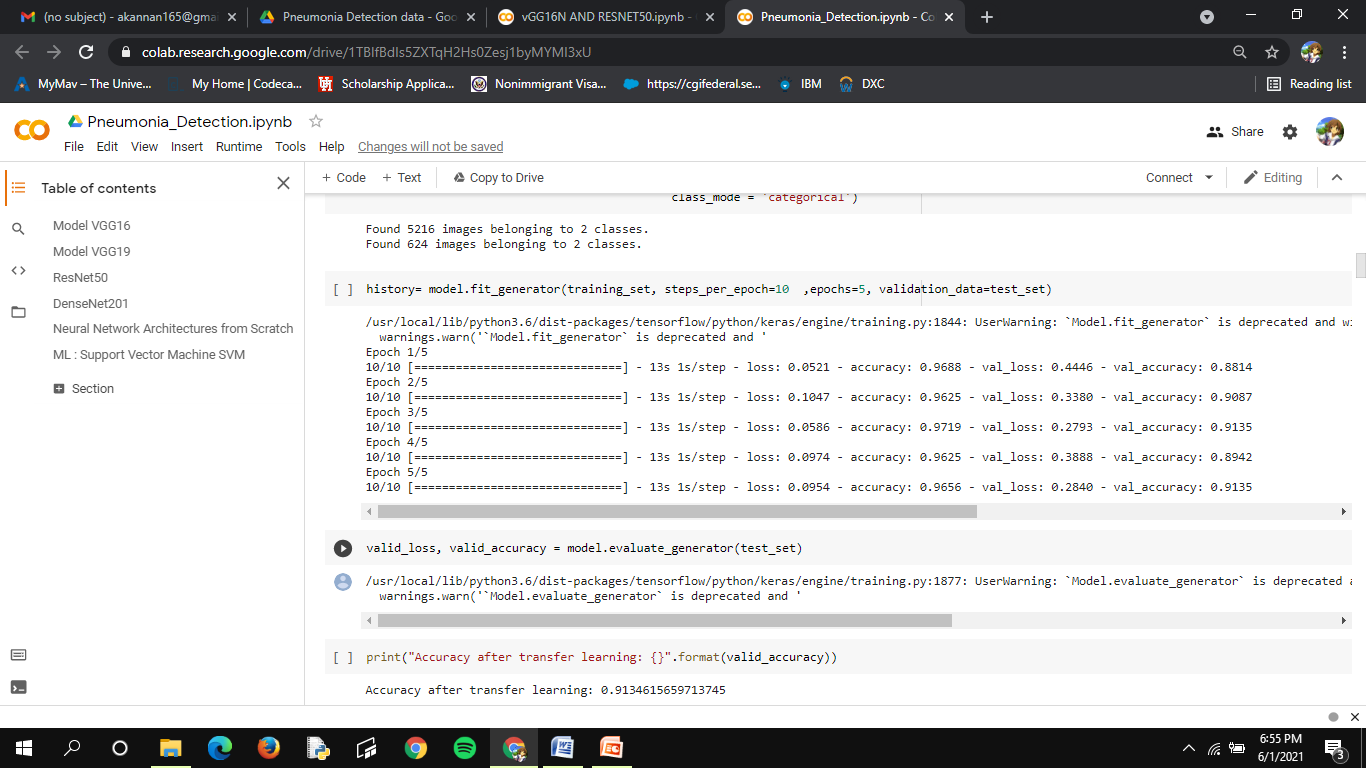
**Fig A.1.4 Model VGG16 Summary**

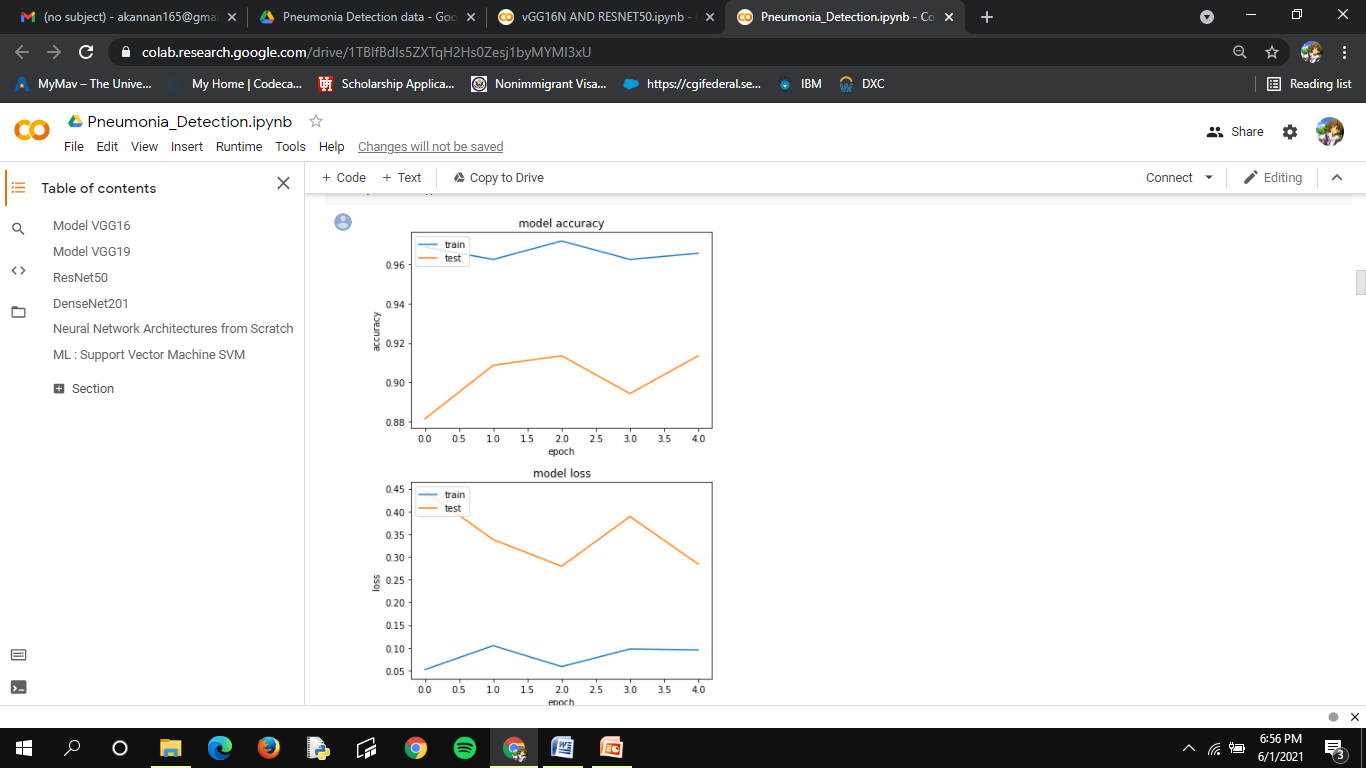
43

****

****

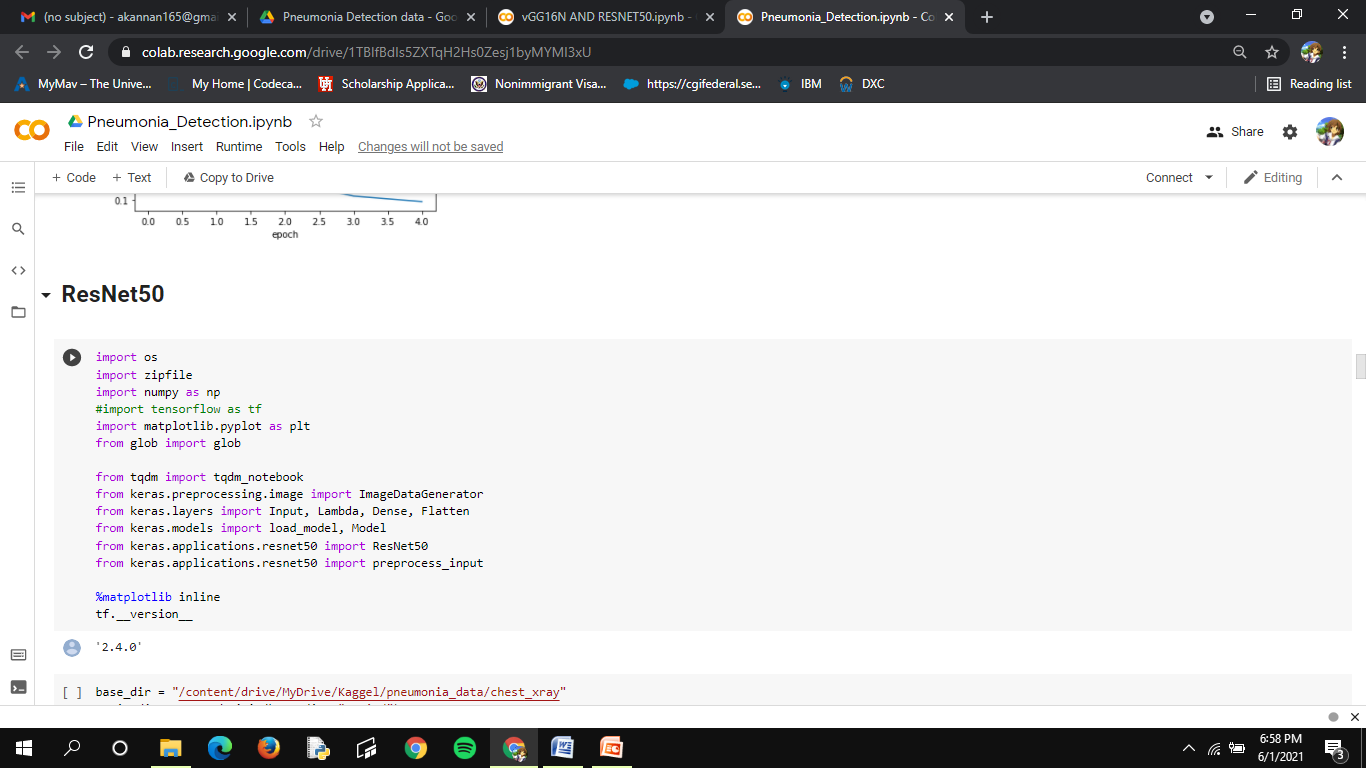
44

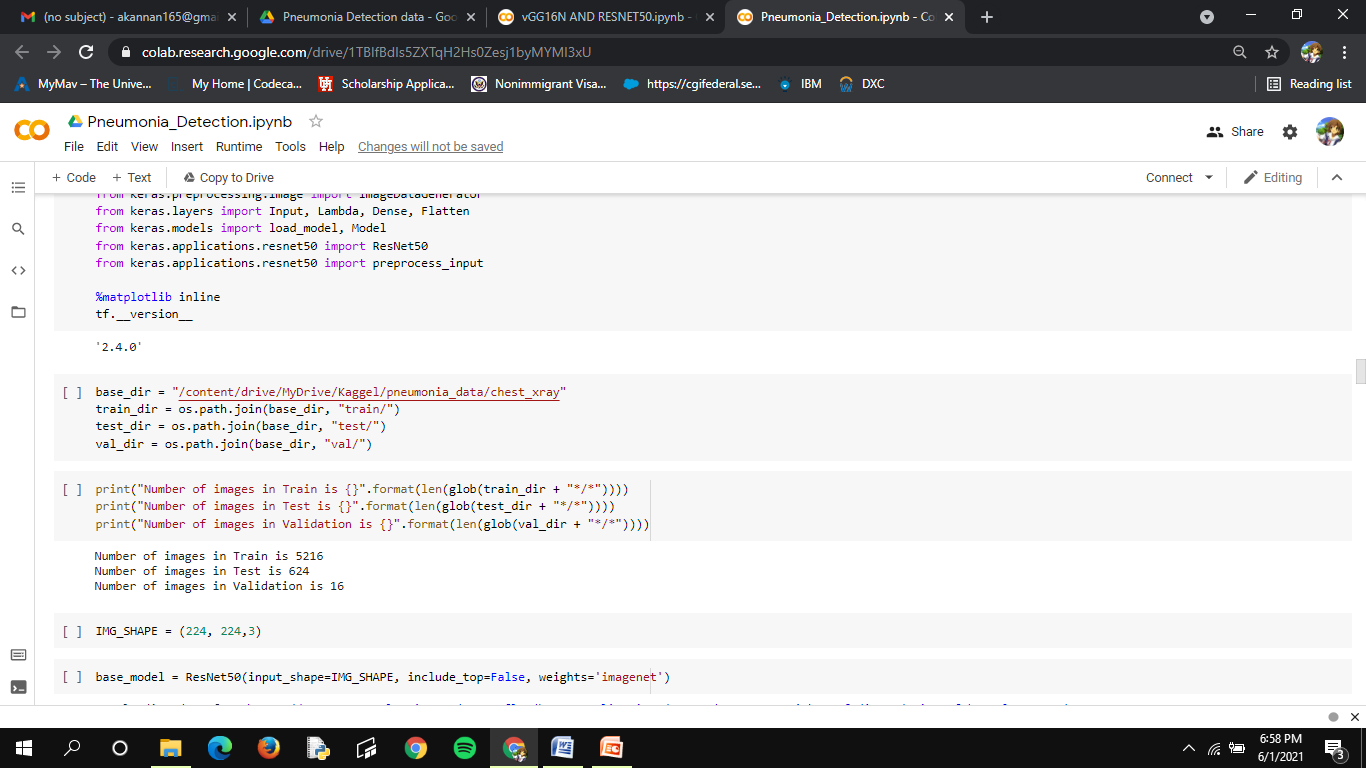
****

****

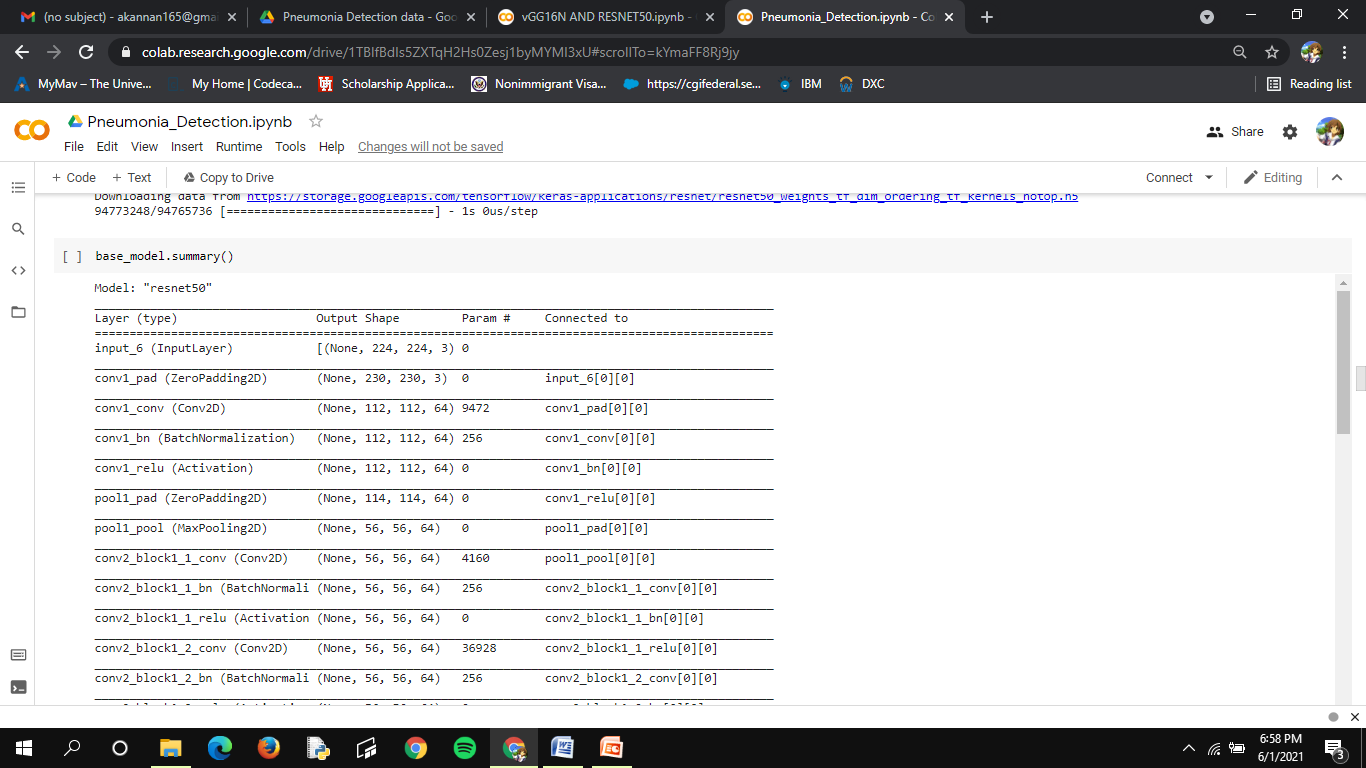
**Fig A.1.5 Model VGG16 Model accuracy and loss**

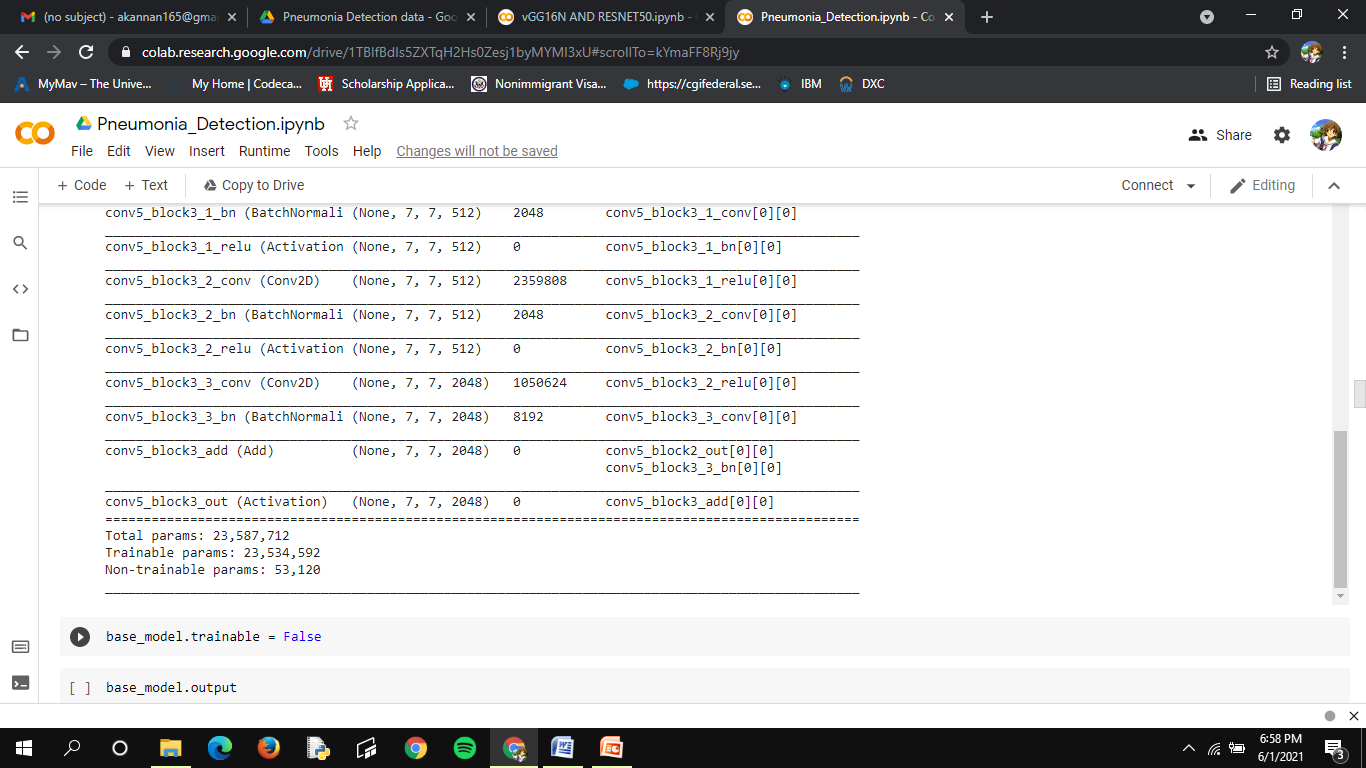
45

****

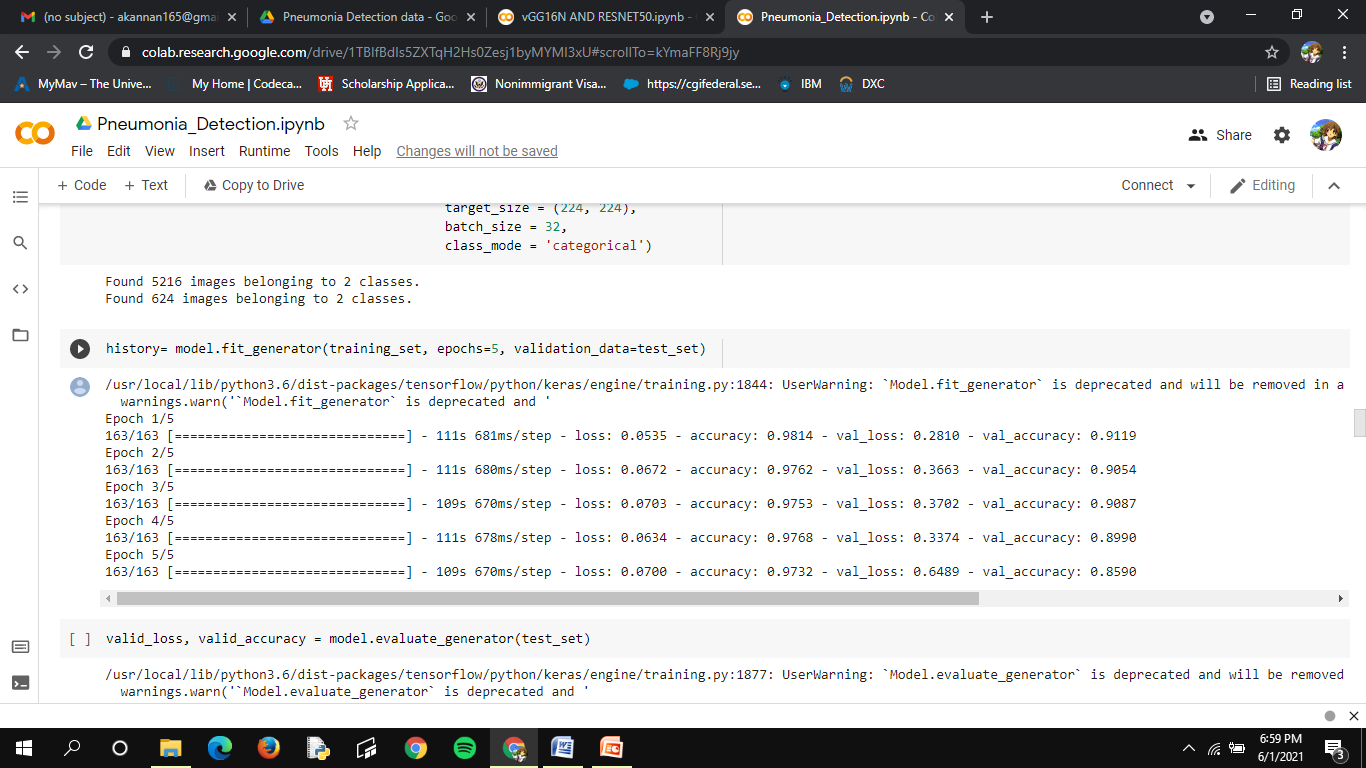
**Fig A.1.6 Resnet50 **

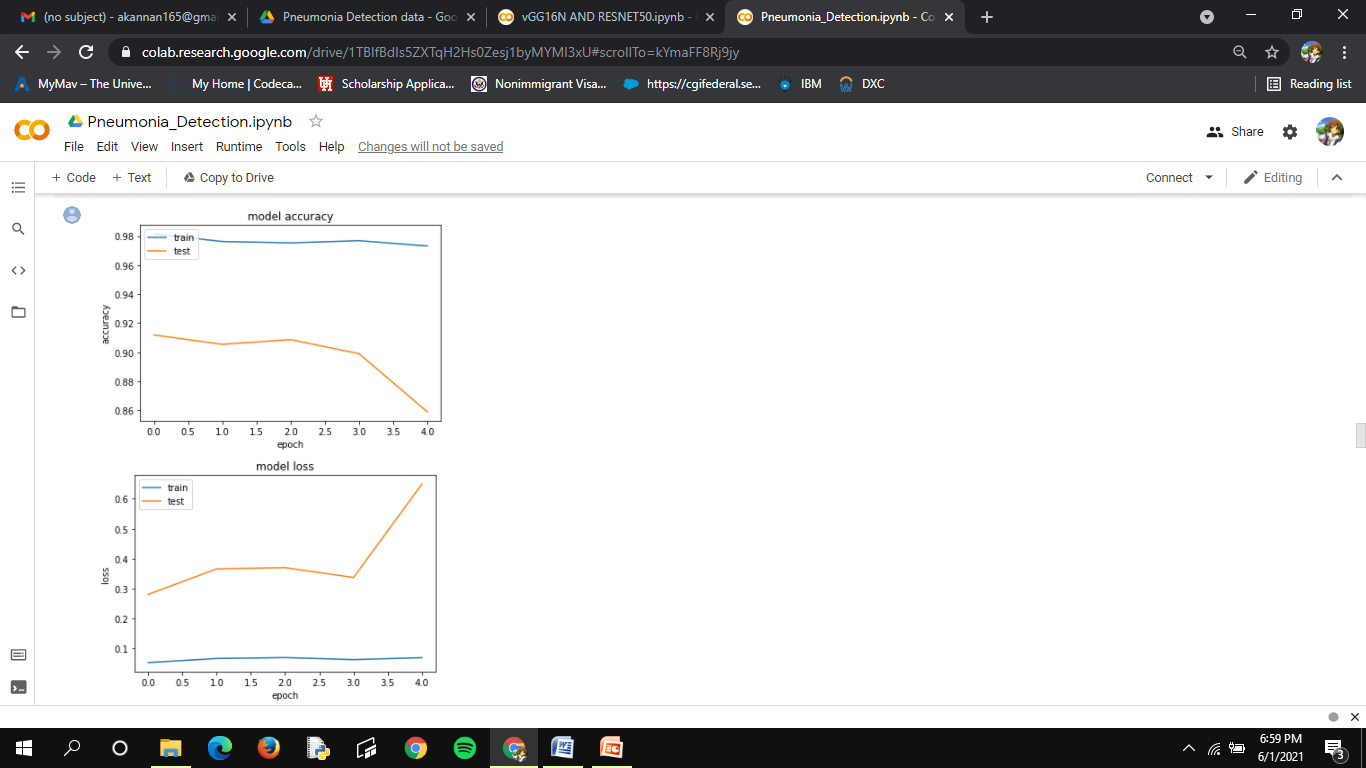
46

****

**Fig A.1.7 Resnest50 model summary **

47

****

****

**Fig A.1.8 Resnest50 model accuracy and loss**

48

**A.2 PUBLICATIONS**

This project was published as a paper in International Journal Of Creative Research thoughts on 24/04/2021 by Anitha Moses, Akshaya K, Anusha R, Harrsheetha S.

**Publishes issue**: Vol 9 issue 4, April 2021

**URL :** [http://ijcrt.org/viewfull.php?&p\_id=IJCRT2104532](http://ijcrt.org/viewfull.php?&p_id=IJCRT2104532.pdf)

**REFERENCES**

1. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 Augus 2017; pp. 1–6.
2. Asiri, N.; Hussain, M.; Al Adel, F.; Alzaidi, N. Deep learning based computer-aided diagnosis systems fordiabetic retinopathy: A survey. Artif. Intell. Med. 2019, 99.
3. Bailer, C.; Habtegebrial, T.; Varanasi, K.; Stricker, D. Fast Feature Extraction with CNNs with Pooling Layers. arXiv 2018, arXiv:1805.03096.
4. Bakator, M.; Radosav, D. Deep Learning and Medical Diagnosis: A Review of Literature. Multimodal Technol. Interact. 2018, 2, 47.
5. Behzadi-khormouji, H.; Rostami, H.; Salehi, S.; Derakhshande-Rishehri, T.; Masoumi, M.; Salemi, S.; Keshavarz, A.; Gholamrezanezhad, A.; Assadi, M.; Batouli, A. Deep learning, reusable and problem-based architectures for detection of consolidation on chest X-ray images. Comput. Methods Programs Biomed. 2020, 185, 105162.  [PubMed]
6. Bouch, C.; Williams, G. Recently published papers: Pneumonia, hypothermia and the elderly. Crit. Care 2006, 10, 167.  [PubMed]
7. Brunetti, A.; Carnimeo, L.; Trotta, G.F.; Bevilacqua, V. Computer-assisted frameworks for classification of liver, breast and blood neoplasias via neural networks: A survey based on medical images. Neurocomputing 2019, 335, 274–298.
8. Cohen, J.P.; Bertin, P.; Frappier, V. Chester: A Web Delivered Locally Computed Chest X-Ray Disease Prediction System. arXiv 2019, arXiv:1901.11210.
9. Da Nóbrega, R.V.M.; Rebouças Filho, P.P.; Rodrigues, M.B.; da Silva, S.P.P.; Dourado Júnior, C.M.J.M.; de Albuquerque, V.H.C. Lung nodule malignancy classification in chest computed tomography images using transfer learning and convolutional neural networks. Neural Comput. Appl. 2018, 1–18.
10. Dai, W.; Chen, Y.; Xue, G.-r.; Yang, Q.; Yu, Y.; Koller, D.; Schuurmans, D.; Bengio, Y.; Bottou, L. (Eds.) Translated Learning: Transfer Learning across Different Feature Spaces. In Advances in Neural Information Processing Systems 21, Proceedings of the Neural Information Processing Systems 2008, Vancouver, BC, Canada, 8–10 December 2008; Neural Information Processing Systems Foundation, Inc. (NIPS): Vancouver, BC, Canada, 2008; pp. 353–360.
11. Fan, J.; Cao, X.; Yap, P.T.; Shen, D. BIRNet: Brain image registration using dual-supervised fully convolutional networks. Med. Image Anal. 2019, 54, 193–206.  [PubMed]
12. Gilani, Z.; Kwong, Y.D.; Levine, O.S.; Deloria-Knoll, M.; Scott, J.A.G.; O’Brien, K.L.; Feikin, D.R. A literature review and survey of childhood pneumonia etiology studies: 2000–2010. Clin. Infect. Dis. 2012, 54 (Suppl. 2), S102–S108.  [PubMed]
13. Goyal, M.; Goyal, R.; Lall, B. Learning Activation Functions: A new paradigm of understanding Neural Networks. arXiv 2019, arXiv:1906.09529. Appl. Sci. 2020, 10, 559 17 of 17

49

1. Gu, Y.; Lu, X.; Yang, L.; Zhang, B.; Yu, D.; Zhao, Y.; Gao, L.; Wu, L.; Zhou, T. Automatic lung nodule detection using a 3D deep convolutional neural network combined with a multi-scale prediction strategy in chest CTs. Comput. Biol. Med. 2018, 103, 220–231.
2. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. arXiv 2015, arXiv:1512.03385.
3. Heron, M. Deaths: Leading causes for 2010. Natl. Vital. Stat. Rep. 2013, 62, 1–96.
4. Ho, T.K.K.; Gwak, J. Multiple feature integration for classification of thoracic disease in chest radiography. Appl. Sci. 2019, 9, 4130.
5. Holzinger, A.; Langs, G.; Denk, H.; Zatloukal, K.; Müller, H. Causability and explainability of artificial intelligence in medicine. WIREs Data Min. Knowl. Discov. 2019, 9, e1312.  66. Shorten, C.; Khoshg
6. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. arXiv2016, arXiv:1608.06993.
7. Jaiswal, A.K.; Tiwari, P.; Kumar, S.; Gupta, D.; Khanna, A.; Rodrigues, J.J.P.C. Identifying pneumonia in chest X-rays: A deep learning approach. Meas. J. Int. Meas. Confed. 2019, 145, 511–518.  Appl. Sci. 2020, 10, 559 16 of 17
8. Jung, H.; Kim, B.; Lee, I.; Lee, J.; Kang, J. Classification of lung nodules in CT scans using three-dimensional deep convolutional neural networks with a checkpoint ensemble method. BMC Med. Imaging 2018, 18, 48.  [PubMed]
9. Kallianos, K.; Mongan, J.; Antani, S.; Henry, T.; Taylor, A.; Abuya, J.; Kohli, M. How far have we come? Artificial intelligence for chest radiograph interpretation. Clin. Radiol. 2019, 74, 338–345.
10. Ke, Q.; Zhang, J.; Wei, W.; Połap, D.; Wo ´zniak, M.; Ko´smider, L.; Damaševiˇcius, R. A neuro-heuristic approach for recognition of lung diseases from X-ray images. Expert Syst. Appl. 2019, 126, 218–232.
11. Kermany, D.S.; Goldbaum, M.; Cai, W.; Valentim, C.C.S.; Liang, H.; Baxter, S.L.; McKeown, A.; Yang, G.; Wu, X.; Yan, F.; et al. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. Cell 2018, 172, 1122–1131.  [PubMed]
12. Kingma, D.P.; Ba, J.L. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference for Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. Commun. ACM 2017, 60, 84–90.
14. Lakhani, P.; Sundaram, B. Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks. Radiology 2017, 284, 574–582.  [PubMed]
15. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. Proc. IEEE 1998, 86, 2278–2324.
16. Li, X.; Shen, L.; Xie, X.; Huang, S.; Xie, Z.; Hong, X.; Yu, J. Multi-resolution convolutional networks for chest X-ray radiograph based lung nodule detection. Artif. Intell. Med. 2019, 101744.  [PubMed]
17. Liang, G.; Zheng, L. A transfer learning method with deep residual network for pediatric pneumonia diagnosis. Comput. Methods Programs Biomed. 2019, 104964.  [PubMed]
18. Litjens, G.; Kooi, T.; Bejnordi, B.E.; Setio, A.A.A.; Ciompi, F.; Ghafoorian, M.; van der Laak, J.A.W.M.; Ginneken, B.; Sánchez, C.I. A survey on deep learning in medical image analysis. Med. Image Anal. 2017, 42, 60–88.  [PubMed]
19. Liu, N.; Wan, L.; Zhang, Y.; Zhou, T.; Huo, H.; Fang, T. Exploiting Convolutional Neural Networks With Deeply Local Description for Remote Sensing Image Classification. IEEE Access 2018, 6, 11215–11228.

50

1. Malukas, U.; Maskeli ¯ unas, R.; Damaševiˇcius, R.; Wo ´zniak, M. Real time path finding for assisted living using deep learning. J. Univers. Comput. Sci. 2018, 24, 475–487.
2. Meyer, P.; Noblet, V.; Mazzara, C.; Lallement, A. Survey on deep learning for radiotherapy. Comput. Biol. Med. 2018, 98, 126–146.  Appl. Sci. 2020, 10, 559 15 of 17
3. Nam, J.G.; Park, S.; Hwang, E.J.; Lee, J.H.; Jin, K.; Lim, K.Y.; Park, C.M. Development and validation of deep learning-based automatic detection algorithm for malignant pulmonary nodules on chest radiographs. Radiology 2019, 290, 218–228.
4. Nasrullah, N.; Sang, J.; Alam, M.S.; Mateen, M.; Cai, B.; Hu, H. Automated lung nodule detection and classification using deep learning combined with multiple strategies. Sensors 2019, 19, 3722.
5. National Center for Health Statistics (NCHS); Centers for Disease Control and Prevention (CDC) FastStats: Pneumonia. Last Updated February 2017. Available online: <http://www.cdc.gov/nchs/fastats/pneumonia.htm> (accessed on 21 November 2019).
6. Pasa, F.; Golkov, V.; Pfeiffer, F.; Cremers, D.; Pfeiffer, D. Efficient deep network architectures for fast chest X-ray tuberculosis screening and visualization. Sci. Rep. 2019, 9, 6268.
7. Pezeshk, A.; Hamidian, S.; Petrick, N.; Sahiner, B. 3-D convolutional neural networks for automatic detection of pulmonary nodules in chest CT. IEEE J. Biomed. Health Inform. 2019, 23, 2080–2090.
8. Raghu, M.; Zhang, C.; Kleinberg, J.M.; Bengio, S. Transfusion: Understanding Transfer Learning with Applications to Medical Imaging. arXiv 2019, arXiv:1902.07208.
9. Rajpurkar, P.; Irvin, J.; Ball, R.L.; Zhu, K.; Yang, B.; Mehta, H.; Duan, T.; Ding, D.; Bagul, A.; Langlotz, C.P.; et al. Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists. PLoS Med. 2018, 15, e1002686.
10. Ravishankar, H.; Sudhakar, P.; Venkataramani, R.; Thiruvenkadam, S.; Annangi, P.; Babu, N.; Vaidya, V. Understanding the Mechanisms of Deep Transfer Learning for Medical Images. In Deep Learning and Data Labeling for Medical Applications; DLMIA 2016, LABELS 2016; Carneiro, G., Ed.; Springer: Cham, Switzerland, 2016; Volume 10008.
11. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer International Publishing: New York, NY, USA; pp. 234–241.
12. Roth, H.R.; Lu, L.; Seff, A.; Cherry, K.M.; Hoffman, J.; Wang, S.; Liu, J.; Turkbey, E.; Summers, R.M. A new 2.5 d representation for lymph node detection using random sets of deep convolutional neural network observations. In Proceedings of the 17th International Conference on Medical Image Computing and Computer-Assisted Intervention, Boston, MA, USA, 14–18 September 2014; Springer International Publishing: New York, NY, USA; pp. 520–527.
13. Scott, J.A.; Brooks, W.A.; Peiris, J.S.; Holtzman, D.; Mulholland, E.K. Pneumonia research to reduce childhood mortality in the developing world. J. Clin. Investig. 2008, 118, 1291–1300.  [PubMed]
14. Shickel, B.; Tighe, P.J.; Bihorac, A.; Rashidi, P. Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. IEEE J. Biomed. Health Inform. 2018, 22, 1589–1604.
15. Shin, H.-C.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. IEEE Trans. Med. Imaging 2016, 35, 1285–1298.
16. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.

51

1. Sirazitdinov, I.; Kholiavchenko, M.; Mustafaev, T.; Yixuan, Y.; Kuleev, R.; Ibragimov, B. Deep neural network ensemble for pneumonia localization from a large-scale chest x-ray database. Comput. Electr. Eng. 2019, 78, 388–399.
2. Souza, J.C.; Bandeira Diniz, J.O.; Ferreira, J.L.; França da Silva, G.L.; Corrêa Silva, A.; de Paiva, A.C. An automatic method for lung segmentation and reconstruction in chest X-ray using deep neural networks. Comput. Methods Programs Biomed. 2019, 177, 285–296.  [PubMed]
3. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. arXiv 2014, arXiv:1409.4842.
4. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. arXiv 2015, arXiv:1512.00567.
5. Taylor, A.G.; Mielke, C.; Mongan, J. Automated detection of moderate and large pneumothorax on frontal chest X-rays using deep convolutional neural networks: A retrospective study. PLoS Med. 2018, 15, e1002697.  [PubMed]
6. Wang, Q.; Shen, F.; Shen, L.; Huang, J.; Sheng, W. Lung nodule detection in CT images using a raw patch-based convolutional neural network. J. Digit. Imaging 2019, 32, 971–979.  [PubMed]
7. Wang, X.; Peng, Y.; Lu, L.; Lu, Z.; Bagheri, M.; Summers, R.M. Chestx-ray8: Hospital scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA; pp. 2097–2106.
8. Wo ´zniak, M.; Połap, D.; Capizzi, G.; Sciuto, G.L.; Ko´smider, L.; Frankiewicz, K. Small lung nodules detection based on local variance analysis and probabilistic neural network. Comput. Methods Programs Biomed. 2018, 161, 173–180.
9. World Health Organization. The Top 10 Causes of Death; World Health Organization: Geneva, Switzerland, 2017 Available online: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death (accessed on 10 November 2019).
10. Wunderink, R.G.; Waterer, G. Advances in the causes and management of community acquired pneumonia in adults. BMJ 2017, 358, j2471.  [PubMed]
11. Xiao, Z.; Du, N.; Geng, L.; Zhang, F.; Wu, J.; Liu, Y. Multi-scale heterogeneous 3D CNN for false-positive reduction in pulmonary nodule detection, based on chest CT images. Appl. Sci. 2019, 9, 3261.
12. Xie, Y.; Zhang, J.; Xia, Y. Semi-supervised adversarial model for benign–malignant lung nodule classification on chest CT. Med. Image Anal. 2019, 57, 237–248.
13. Xu, S.; Wu, H.; Bie, R. CXNet-m1: Anomaly detection on chest X-rays with image-based deep learning. IEEE Access 2019, 7, 4466–4477.
14. Yates, E.J.; Yates, L.C.; Harvey, H. Machine learning “red dot”: Open-source, cloud, deep convolutional neural networks in chest radiograph binary normality classification. Clin. Radiol. 2018, 73, 827–831.
15. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H.; Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N.D.; Weinberger, K.Q. (Eds.) How transferable are features in deep neural networks? In Advances in Neural Information Processing Systems 27, Proceedings of the Neural Information Processing Systems 2014, Montreal, QC, Canada, 8–13 December 2014; Neural Information Processing Systems Foundation, Inc. (NIPS): Montreal, QC, Canada, 2014; pp. 3320–3328.
16. Zhang, X.; Yao, L.; Wang, X.; Monaghan, J.; McAlpine, D. A Survey on Deep Learning based Brain Computer Interface: Recent Advances and New Frontiers. arXiv 2019, arXiv:1905.04149.
17. Zhou, T.; Thung, K.; Zhu, X.; Shen, D. Effective feature learning and fusion of multimodality data using stage-wise deep neural network for dementia diagnosis. Hum. Brain Mapp. 2018, 40, 1001–1016.

52