

**USMAN INSTITUTE OF TECHNOLOGY**

Affiliated with NED University of Engineering & Technology, Karachi

**Department of Computer Science**

B.S. Computer Science / Software Engineering

FINAL YEAR PROJECT REPORT

**Batch-2020**

**Optimizing Deep Learning Models for Pathology Classification in**

**Chest X-ray and CT Scan Images**

**CS-025**

**By**

| Syed Hashir Ali Hussaini | 20B-051-CS |
| --- | --- |
| Sayed M. Abdullah | 20B-063-CS |
| M. Huzaifa Khan | 20B-039-CS |

**Supervised by**

Miss Farah Adil

*ST-13, Block 7, Gulshan-e-Iqbal, Abul Hasan Isphahani Road, Opposite Safari Park, P.O. Box 75300,*

*Karachi, Pakistan. Phone: 34978274-5; 34994305; 34982476;* [*http://www.uit.edu*](http://www.uit.edu/)

**Submission Performa**

Name (1)………………. (Syed Hashir Ali Hussaini)

(2)………………. (Sayed Mohammad Abdullah)

(3)………………. (Muhammad Huzaifa Khan)

Address (1)…………. … (B-18 Duplex Naveed bangalows Gulistan-e-Johar block 17 Karachi)

(2)…………. … (3-F 9/16 Nazimabad Number 3 Karachi)

(3)…………. … (House A-298 Sector 14-B Shadman town 1 Karachi)

Optimizing Deep Learning Models for Pathology Classification in

Chest X-ray and CT Scan Images

Miss Farah Adil

This report is submitted as required for the Project in accordance with the rules laid down by the Usman Institute of technology as part of the requirements for the award of the degree of Bachelor **Computer Science/Software Engineering**. I declare that the work presented in this report is my own except where due reference or acknowledgement is given to the work of others.

Signatures of students Date

(1)…………………………….. ……………………..

(2)……………………………. …………………….

(3)…………………………….. …………………….

**Acknowledgments**

This page should consist of the acknowledgements to the people, companies and institutions that have been helpful to the author in compiling the reports. It is normal practice to thank the Head of Institute for the use of facilities with which the project was carried out, the head of department, the supervisor for his/her suggestions and guidance and any other member of the academic and technical support staff who have made a significant contribution to the success of the project, and finally your family member (that is optional).

**Abstract**

This report delves into the intersection of radiology and deep learning, highlighting the utilization of advanced computational models to analyze chest X-ray and CT scan images. It emphasizes the significance of deep learning in addressing the limitations of conventional radiological methods and enhancing the accuracy and efficiency of disease detection and classification. The project aims to develop an integrated medical imaging system that automates disease diagnosis, severity assessment, and standardized reporting, thus improving patient care and healthcare efficiency.

The work was undertaken through the integration of high-performance GPUs, robust CPUs, and sufficient RAM for data processing. The project extensively utilized Python, TensorFlow, React, and OpenCV for deep learning model development, user interface construction, and image manipulation. Secure storage mechanisms like MySQL and MongoDB were employed for efficient data management, while Git and Trello facilitated collaborative development and project management.

From the comprehensive analysis of various deep learning models, it was concluded that the integration of techniques such as transfer learning, data augmentation, and advanced classification methods significantly improved the accuracy and robustness of disease detection from medical images. The project's primary aim to enhance disease diagnosis accuracy was successfully achieved, along with the additional goals of automated severity assessment, patient prioritization, and standardized reporting, ensuring improved healthcare management and patient outcomes.

Keywords: Radiology, deep learning, medical imaging, disease diagnosis, chest X-ray, CT scan, TensorFlow, transfer learning, data augmentation, standardized reporting.

**Table of contents**

Contents

[**Declaration 2**](#_heading=h.3tbugp1)

[**Acknowledgments 3**](#_heading=h.gjdgxs)

[**Abstract 4**](#_heading=h.30j0zll)

[**Table of contents 5**](#_heading=h.1fob9te)

[**List of tables 7**](#_heading=h.3znysh7)

[**List of figures 9**](#_heading=h.2et92p0)

[**List of symbols and Units 10**](#_heading=h.tyjcwt)

[**1.**](#_heading=h.3dy6vkm) **Introduction 11**

[**2.**](#_heading=h.1t3h5sf) **Background and Literature Review 12**

[**3.**](#_heading=h.4d34og8) **Aim and Statement of Problem 13**

[**4.**](#_heading=h.2s8eyo1) **Hardware, Software analysis and requirements 14**

[**5.**](#_heading=h.17dp8vu) **Software design and modeling 15**

[**6.**](#_heading=h.3rdcrjn) **Algorithm analysis and complexity 16**

[**7.**](#_heading=h.26in1rg) **Implementation 17**

[**8.**](#_heading=h.lnxbz9) **Testing 18**

[**9.**](#_heading=h.35nkun2) **Discussion (optional) 19**

[**10.**](#_heading=h.1ksv4uv) **Conclusions 20**

[**11.**](#_heading=h.44sinio) **Future work 21**

[**12.**](#_heading=h.2jxsxqh) **Achievements 22**

[**13.**](#_heading=h.z337ya) **Appendices 23**

[**14.**](#_heading=h.3j2qqm3) **General Guidelines 24**

[14.1](#_heading=h.1y810tw) Typing and size of paper 24

[14.2](#_heading=h.4i7ojhp) Page number and Chapter number 24

[14.3](#_heading=h.2xcytpi) Margin boundaries 25

[14.4](#_heading=h.1ci93xb) Diagrams and figures 25

[14.5](#_heading=h.3whwml4) Photocopying 25

[14.6](#_heading=h.2bn6wsx) Fixing of photograph 26

[14.7](#_heading=h.qsh70q) Tables 26

[14.8](#_heading=h.3as4poj) Equations 26

[14.9](#_heading=h.1pxezwc) Units 26

[14.10](#_heading=h.49x2ik5) References 26

[*14.10.1*](#_heading=h.2p2csry) *Citations/references with multiple authors 27*

[*14.10.2*](#_heading=h.147n2zr) *Reference examples 27*

[*14.10.3*](#_heading=h.3o7alnk) *Book 27*

[*14.10.4*](#_heading=h.23ckvvd) *Book chapter 27*

[*14.10.5*](#_heading=h.ihv636) *Electronic Book 28*

[*14.10.6*](#_heading=h.32hioqz) *Journal article 28*

[*14.10.7*](#_heading=h.1hmsyys) *E-Journal article 28*

[*14.10.8*](#_heading=h.41mghml) *Conference papers 29*

[*14.10.9*](#_heading=h.2grqrue) *Reports 29*

[*14.10.10*](#_heading=h.vx1227) *Patents 29*

[*14.10.11*](#_heading=h.3fwokq0) *Standards 29*

[*14.10.12*](#_heading=h.1v1yuxt) *Theses/Dissertations 30*

[*14.10.13*](#_heading=h.4f1mdlm) *Datasheets 30*

[*14.10.14*](#_heading=h.2u6wntf) *Online Documents 30*

[*14.10.15*](#_heading=h.19c6y18) *Websites 30*

The table of the contents is on a separate page and is numbered iii. The table of contents lists the sections of the report, a list of figures and a list of tables along with the page on which they begin. It is to be generated by MS word -> references -> table of contents, similarly list of tables and list of figures.

**List of tables**

**List of figures**

**List of symbols and Units**

A list of symbols and units should be included to assist the reader. This should include any Greek letters or other mathematical symbols together with the quantities to which they refer, and their appropriate units. Preferred 2-3 pages of UML notations used in the project.

# Introduction

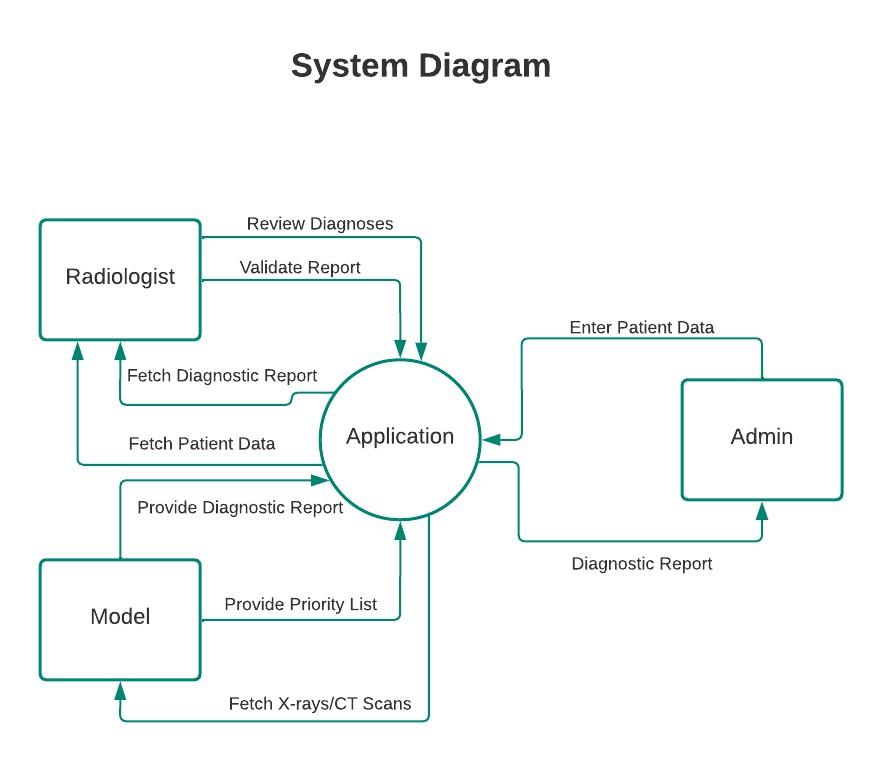
Radiology has long been a cornerstone of modern medical diagnostics, providing crucial insights into a diverse array of physiological conditions and pathologies. Conventional radiological methods, such as chest X-rays and CT scans, have played a pivotal role in the detection and diagnosis of thoracic diseases, including pneumonia, lung cancer, and tuberculosis. These techniques have traditionally relied on the expertise of skilled radiologists who visually analyze complex images to identify abnormalities and make informed diagnostic decisions. However, this process is not without its limitations, as it is often susceptible to subjective variability and human error, particularly in cases where diseases present with similar imaging patterns, leading to potential misdiagnosis and delayed treatment.

Considering these challenges, recent advancements in deep learning methodologies have sparked a paradigm shift in the field of medical image analysis, particularly in the domain of chest radiography. Deep learning techniques, such as convolutional neural networks (CNNs), have demonstrated remarkable capabilities in automatically extracting intricate patterns and features from radiographic images, enabling more precise and efficient disease detection and classification. The integration of deep learning into radiological practice has the potential to significantly augment the diagnostic process, providing radiologists with powerful tools for more accurate and timely assessments of thoracic pathologies.

This report delves into the intersection of radiology and deep learning, focusing on the utilization of advanced computational models to analyze chest X-ray and CT scan images. It aims to provide a comprehensive overview of the application of deep learning in addressing the limitations of conventional radiological methods, thereby enhancing the accuracy and efficiency of disease detection and classification. By leveraging the power of deep learning algorithms, researchers and clinicians can augment their diagnostic capabilities, leading to improved patient outcomes and more effective healthcare management.

Throughout this report, we will delve into the conventional methods of radiological diagnostics, emphasizing their significance in the detection and diagnosis of various thoracic diseases. We will then highlight the limitations and challenges associated with these conventional approaches, underscoring the need for more advanced and automated techniques to mitigate potential diagnostic errors and improve overall patient care. Additionally, we will explore the underlying reasons for the transition from conventional radiological methods to deep learning, emphasizing the advantages of leveraging computational intelligence to enhance the efficiency and accuracy of disease detection in chest radiography.

The subsequent chapters will delve into the role of deep learning in revolutionizing the field of radiology, showcasing its potential in automating disease detection and classification. We will provide a comprehensive analysis of various deep learning models and methodologies employed in the interpretation of chest radiographs, highlighting their performance specifications and their impact on enhancing diagnostic accuracy. Furthermore, we will discuss the future directions and potential implications of these advancements, underlining the transformative potential of deep learning in the domain of medical image analysis and diagnostics.



# Background and Literature Review

The emergence of deep learning (DL) techniques in the field of medical image analysis has revolutionized the diagnostic process, particularly in the detection of pneumonia and COVID-19 conditions using chest X-ray images. Numerous research efforts have been directed towards developing accurate and efficient DL models to aid in the early and accurate diagnosis of these critical respiratory conditions. DL models, such as Convolutional Neural Networks (CNNs), have gained significant attention due to their ability to automatically extract intricate features from medical images, enabling precise classification and detection.

The current landscape of research reveals a growing body of literature focused on the application of DL in medical image classification. Various studies have employed a range of DL architectures, including VGG19, DenseNet121, InceptionV3, ResNetV2, Inception-ResNet-V2, Xception, and MobileNetV2, among others, to achieve high accuracy and performance in the classification of pneumonia and COVID-19 conditions from chest X-ray and CT images.

**Literature Review:**

In their systematic review, researchers provide a comprehensive analysis of the existing survey papers, highlighting the limitations and gaps in the research related to DL-based classification of chest X-ray images. The review underscores the need for addressing critical aspects such as loss calculation, optimization, evaluation metrics, and model selection guidelines, which have been overlooked in previous surveys.

Several studies have demonstrated the effectiveness of DL models in accurately diagnosing COVID-19 from chest X-ray images. For instance, one study proposed a framework based on Capsule Networks, achieving a high accuracy of 95.7% in COVID-19 detection. Similarly, another study developed a hybrid system using machine learning and deep learning algorithms, specifically employing a CNN with a softmax classifier for COVID-19 detection based on chest X-ray images.

Furthermore, the use of transfer learning techniques has been prominent in overcoming data scarcity challenges. Studies have applied transfer learning approaches, utilizing various pre-trained CNN models such as VGG16, ResNet, and DenseNet for pneumonia classification, demonstrating promising results in the classification of chest X-ray images.

Moreover, the utilization of CT images for the early detection of COVID-19 has been explored. One study proposed a weakly-supervised deep learning model, DeCoVNet, which incorporates a 3D deep neural network for the identification of COVID-19 infection from CT scans. This approach achieved an impressive accuracy of 90.1%.

Despite the progress in the field, several challenges persist, including data scarcity, overfitting, and the need for robust detection methods. Studies have addressed these challenges by employing data augmentation, generative adversarial networks, and advanced classification techniques, resulting in improved accuracy and robustness in the detection of pneumonia and COVID-19 from chest X-ray images.

The current literature presents a comprehensive understanding of the state-of-the-art DL techniques, their implementations, and the challenges encountered in the classification of pneumonia and COVID-19 from medical imaging data. However, gaps still exist in terms of standardization, optimization, and the integration of these models into clinical practice. This review aims to bridge these gaps and provide valuable insights for the development of effective DL-based solutions in medical imaging applications.

**Algorithms**:

* **ImageNet Classification with Deep Convolutional Neural Networks**

This paper proposes a deep convolutional neural network architecture for image classification on the ImageNet dataset. The network, called AlexNet, achieves state-of-the-art results on the ImageNet classification task, with an error rate of 37.5%.

**Key contributions:**

The authors introduce several techniques that improve the performance of deep convolutional neural networks, including:

ReLU activation functions, which are more efficient than sigmoid or tanh activation functions.

Dropout regularization, which helps to prevent overfitting.

Data augmentation techniques, which artificially increase the size of the training dataset.

**Limitations:**

The AlexNet architecture is computationally expensive to train and deploy.

* **Xception: Deep Learning with Depth wise Separable Convolutions**

This paper proposes a new convolutional neural network architecture called Xception. Xception is based on the Inception architecture, but it uses depth wise separable convolutions instead of regular convolutions. Depth wise separable convolutions are more efficient than regular convolutions, and they can achieve comparable performance on image classification tasks.

**Key contributions:**

The authors introduce depth wise separable convolutions, which are a more efficient way to perform convolutions.

They show that Xception can achieve state-of-the-art results on image classification tasks, while being more efficient than previous architectures.

**Limitations:**

Xception is still computationally expensive to train and deploy.

* **Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning**

This paper proposes a new convolutional neural network architecture called Inception-v4. Inception-v4 is based on the Inception architecture, but it introduces a few new features, including residual connections and batch normalization. Residual connections help to make the network deeper and more powerful, while batch normalization helps to stabilize the training process.

**Key contributions:**

The authors introduce residual connections and batch normalization, which are two techniques that can improve the performance of deep convolutional neural networks.

They show that Inception-v4 can achieve state-of-the-art results on image classification tasks, while being more efficient than previous architectures.

**Limitations:**

Inception-v4 is still computationally expensive to train and deploy.

* **EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks**

This paper proposes a new approach to scaling up convolutional neural networks called EfficientNet. EfficientNets use a compound coefficient to scale the depth, width, and resolution of the network, which allows them to achieve state-of-the-art results on image classification tasks while being more efficient than previous architectures.

**Key contributions:**

The authors propose a new approach to scaling up convolutional neural networks called EfficientNet.

They show that EfficientNets can achieve state-of-the-art results on image classification tasks, while being more efficient than previous architectures.

**Limitations:**

EfficientNets are still computationally expensive to train and deploy.

* **Densely Connected Convolutional Networks**

DenseNet is a convolutional neural network (CNN) architecture that is based on the idea of feature reuse. DenseNets connect each layer to every other layer in the network, which allows them to reuse features that have been learned by previous layers. This makes DenseNets more efficient and accurate than traditional CNN architectures.

DenseNets were first introduced in the paper "Densely Connected Convolutional Networks" by Huang et al.. The authors of the paper showed that DenseNets could achieve state-of-the-art results on a number of image classification and object detection benchmarks.

DenseNets have been used for a variety of tasks, including image classification, object detection, and semantic segmentation. They have been shown to be particularly effective for tasks where the input data is large and complex.

**Key contributions:**

1. Feature reuse: DenseNets connect each layer to every other layer in the network, which allows them to reuse features that have been learned by previous layers. This makes DenseNets more efficient and accurate than traditional CNN architectures.
2. Dense connectivity: DenseNets use dense connectivity to connect the layers in the network. This means that each layer is connected to every other layer in the network, both before and after it. This dense connectivity helps DenseNets to learn more complex features.
3. Bottleneck layers: DenseNets use bottleneck layers to reduce the number of parameters in the network. Bottleneck layers are small layers that are followed by a batch normalization layer and a ReLU activation function. This helps to reduce the number of parameters in the network without sacrificing accuracy.

DenseNets have been shown to achieve state-of-the-art results on a number of image classification and object detection benchmarks. They are a powerful and versatile CNN architecture that can be used for a variety of tasks.

* **CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning**

CheXNet is a convolutional neural network (CNN) architecture that is specifically designed for the task of chest X-ray classification. It is trained on a large dataset of chest X-rays that are labeled with different diseases, such as pneumonia, tuberculosis, and cardiomegaly. CheXNet has been shown to achieve state-of-the-art results on a number of chest X-ray classification benchmarks.

CheXNet is a DenseNet architecture, which means that it connects each layer to every other layer in the network. This allows CheXNet to learn more complex features from the chest X-rays. CheXNet also uses a technique called transfer learning, which means that it is pre-trained on a large dataset of natural images. This helps CheXNet to learn general features that are useful for chest X-ray classification.

CheXNet has been shown to be effective for a variety of chest X-ray classification tasks, including detecting pneumonia, tuberculosis, and cardiomegaly. CheXNet has also been shown to be effective for detecting other chest X-ray abnormalities, such as lung nodules and masses.

CheXNet is a powerful tool that can be used to improve the accuracy of chest X-ray classification. It is particularly useful for detecting diseases that are difficult to diagnose on chest X-rays, such as pneumonia and tuberculosis. CheXNet can also be used to identify patients who are at high risk of developing certain diseases, such as heart disease and cancer.

**Key contributions:**

1. DenseNet architecture: CheXNet is a DenseNet architecture, which means that it connects each layer to every other layer in the network. This allows CheXNet to learn more complex features from the chest X-rays.
2. Transfer learning: CheXNet uses transfer learning, which means that it is pre-trained on a large dataset of natural images. This helps CheXNet to learn general features that are useful for chest X-ray classification.
3. State-of-the-art performance: CheXNet has been shown to achieve state-of-the-art results on a number of chest X-ray classification benchmarks.
4. Effective for a variety of tasks: CheXNet has been shown to be effective for a variety of chest X-ray classification tasks, including detecting pneumonia, tuberculosis, cardiomegaly, lung nodules, and masses.
5. Improves accuracy of chest X-ray classification: CheXNet can be used to improve the accuracy of chest X-ray classification, particularly for diseases that are difficult to diagnose on chest X-rays.
6. Identifies patients at high risk of developing certain diseases: CheXNet can be used to identify patients who are at high risk of developing certain diseases, such as heart disease and cancer.

* **Deep Residual Learning for Image Recognition**

VGGNet is a convolutional neural network (CNN) architecture that was proposed by Simonyan and Zisserman in 2014. It is a simple but effective architecture that has been used to achieve state-of-the-art results on a number of image classification benchmarks.

VGGNet is characterized by its use of small 3x3 convolutional filters and its use of multiple convolutional layers with the same filter size and stride. This allows VGGNet to learn more complex features from the input images.

VGGNet has been used for a variety of tasks, including image classification, object detection, and semantic segmentation. It has been shown to be particularly effective for tasks where the input images are small and low-resolution.

**Key contributions:**

1. Small 3x3 convolutional filters: VGGNet uses small 3x3 convolutional filters. This allows VGGNet to learn more complex features from the input images.
2. Multiple convolutional layers with the same filter size and stride: VGGNet uses multiple convolutional layers with the same filter size and stride. This allows VGGNet to learn more complex features from the input images.
3. State-of-the-art performance: VGGNet has been shown to achieve state-of-the-art results on a number of image classification benchmarks.
4. Effective for a variety of tasks: VGGNet has been used for a variety of tasks, including image classification, object detection, and semantic segmentation.
5. Effective for tasks with small and low-resolution input images: VGGNet has been shown to be particularly effective for tasks where the input images are small and low-resolution.

VGGNet is a powerful and versatile CNN architecture that can be used for a variety of tasks. It is a good choice for tasks where the input images are small and low-resolution, or where state-of-the-art performance is required.

* **Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning**

Inception-ResNet is a convolutional neural network (CNN) architecture that combines the Inception module from the Inception family of networks with the residual connections from the ResNet family of networks. It was proposed by Szegedy et al. in 2016.

Inception-ResNet is a very deep network, with up to 152 layers. It uses a combination of Inception modules and residual connections to learn complex features from the input images.

Inception modules are a type of CNN module that combines multiple convolutional layers with different filter sizes and strides. This allows Inception modules to learn a variety of features from the input images.

Residual connections are a type of CNN connection that allows the network to learn residual functions. This means that the network can learn how to add a residual function to the input image, rather than learning the entire function from scratch.

Inception-ResNet has been shown to achieve state-of-the-art results on a number of image classification benchmarks, including the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It has also been shown to be effective for other tasks, such as object detection and semantic segmentation.

**Key contributions:**

1. Very deep network: Inception-ResNet is a very deep network, with up to 152 layers. This allows it to learn complex features from the input images.
2. Combination of Inception modules and residual connections: Inception-ResNet combines the Inception module from the Inception family of networks with the residual connections from the ResNet family of networks. This allows it to learn a variety of features from the input images and to learn residual functions.
3. State-of-the-art performance: Inception-ResNet has been shown to achieve state-of-the-art results on a number of image classification benchmarks.
4. Effective for other tasks: Inception-ResNet has also been shown to be effective for other tasks, such as object detection and semantic segmentation.

Inception-ResNet is a powerful and versatile CNN architecture that can be used for a variety of tasks. It is a good choice for tasks where state-of-the-art performance is required, or where the input images are very complex.

* **MobileNetV2: Inverted Residuals and Linear Bottlenecks**

MobileNet is a convolutional neural network (CNN) architecture that is designed to be lightweight and efficient. It was proposed by Howard et al. in 2017.

MobileNet uses a number of techniques to reduce the complexity of the network, while still maintaining good performance. These techniques include:

1. Depthwise separable convolutions: MobileNet uses depthwise separable convolutions instead of traditional convolutions. Depthwise separable convolutions split the convolution operation into two steps: a depthwise convolution and a pointwise convolution. This reduces the number of parameters and computations required.
2. Inception modules: MobileNet uses a modified version of the Inception module from the Inception family of networks. This modified Inception module is smaller and more efficient than the original Inception module.
3. Squeeze-and-excitation blocks: MobileNet uses Squeeze-and-Excitation blocks to improve the performance of the network. Squeeze-and-Excitation blocks help the network to learn how to focus on the most important features in the input image.

MobileNet has been shown to achieve good performance on a number of image classification benchmarks, while still being significantly more lightweight than other CNN architectures. For example, MobileNet v2 achieves 70.6% top-1 accuracy on the ImageNet dataset, while only using 3.5M parameters.

MobileNet is a good choice for applications where performance is important, but where the network needs to be lightweight and efficient. For example, MobileNet can be used for image classification on mobile devices or embedded systems.

**Key contributions:**

1. Lightweight and efficient: MobileNet is designed to be lightweight and efficient, making it suitable for mobile devices and embedded systems.
2. Good performance: MobileNet has been shown to achieve good performance on a number of image classification benchmarks.
3. Uses depthwise separable convolutions, Inception modules, and Squeeze-and-Excitation blocks: MobileNet uses a number of techniques to reduce the complexity of the network, while still maintaining good performance.

Overall, MobileNet is a powerful and versatile CNN architecture that can be used for a variety of applications where performance is important, but where the network needs to be lightweight and efficient.

**References:**

| **[1]** | [Pneumonia detection in chest X-ray images using an ensemble of deep learning models](https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0256630) |
| --- | --- |
| **[2**] | [A deep learning architecture for multi-class lung diseases classification using chest X-ray (CXR) images](https://www.sciencedirect.com/science/article/pii/S1110016822007104#s0010)  [Mufarah M. Alshmrani a, Qiang Ni a, Richard Jiang a, Haris Pervaiz a, Nada M. Elshennawy b](https://www.sciencedirect.com/science/article/pii/S1110016822007104#s0010) |
| **[3]** | [Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases](https://www.sciencedirect.com/science/article/pii/S0010482521001426#sec2) |
| **[4]** | [Deep-learning framework to detect lung abnormality – A study with chest X-Ray and lung CT scan images](https://www.sciencedirect.com/science/article/abs/pii/S0167865519303277) |
| **[5]** | [A Review of Recent Advances in Deep Learning Models for Chest Disease Detection Using Radiography](https://www.mdpi.com/2075-4418/13/1/159#B13-diagnostics-13-00159) |
| **[6]** | [Deep learning for chest X-ray analysis: A survey](https://www.sciencedirect.com/science/article/pii/S1361841521001717#bib0050) |
| **[7]** | [Chest X-ray analysis empowered with deep learning: A systematic review](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9393235/) |
| **[8]** | [CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning](https://arxiv.org/abs/1711.05225) |
| **[9]** | [COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images](https://arxiv.org/abs/2003.11055) |
| **[10]** | [Densely Connected Convolutional Networks](https://arxiv.org/abs/1608.06993) |
| **[11]** | [EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks](https://arxiv.org/abs/1905.11946) |
| [12] | [ImageNet Classification with Deep Convolutional Neural Networks](https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf) |
| [13] | [Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning](https://arxiv.org/abs/1602.07261) |
| **[14]** | [Rethinking the Inception Architecture for Computer Vision](https://arxiv.org/abs/1512.00567v3) |
| **[15]** | [MobileNetV2: Inverted Residuals and Linear Bottlenecks](https://arxiv.org/abs/1801.04381) |
| **[16]** | <https://stanfordmlgroup.github.io/projects/chexnext/> |
| **[17]** | [A deep learning architecture for multi-class lung diseases classification using chest X-ray (CXR) images](https://www.sciencedirect.com/science/article/pii/S1110016822007104) |
| **[18]** | [Deep learning based detection and analysis of COVID-19 on chest X-ray images](https://link.springer.com/article/10.1007/s10489-020-01902-1) |
| **[19]** | [CheXNet-with-localization](https://github.com/thtang/CheXNet-with-localization) |
| **[20]** | [CXNet-m2: A Deep Model with Visual and Clinical Contexts for Image-Based Detection of Multiple Lesions](https://link.springer.com/chapter/10.1007/978-3-030-23597-0_33#:~:text=of%20Multiple%20Lesions-,CXNet-m2%3A%20A%20Deep%20Model%20with%20Visual%20and%20Clinical%20Contexts,Based%20Detection%20of%20Multiple%20Lesions) |
| **[21]** | [Computer aided detection of tuberculosis on chest radiographs: An evaluation of the CAD4TB v6 system](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7099074/) |
| **[22]** | [AI-RADS: Successes and challenges of a novel artificial intelligence curriculum for radiologists across different delivery formats](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9845918/) |
| **[23]** | [DeepPrior++: Improving Fast and Accurate 3D Hand Pose Estimation](https://arxiv.org/pdf/1708.08325v1.pdf) |

# Aim and Statement of Problem

The aim of this project is to develop an integrated medical imaging system that utilizes advanced deep learning models to provide precise and efficient disease diagnosis from chest X-ray and CT scan images. This system will not only enhance the speed and accuracy of diagnosis but also automate disease prioritization based on severity and generate comprehensive diagnostic reports. By achieving this aim, the project strives to address critical challenges in medical imaging, particularly concerning chest and lung-related diseases, and significantly improve patient care and healthcare efficiency.

**Statement of the Problem:**

**Problem 1: Accuracy and Efficiency in Disease Diagnosis**

**Scope:** The foremost issue in medical imaging is the accuracy and efficiency of disease diagnosis, especially for a wide range of chest and lung-related conditions. Current diagnostic methods, often reliant on manual interpretation, are time-consuming and subject to human error.

**Solution Approach:** To address this problem, we employed deep learning models capable of automating disease diagnosis from medical images. These models not only expedite the diagnostic process but also significantly enhance accuracy.

**Problem 2:** **Patient Severity Assessment and Prioritization**

**Scope:** Prioritizing patients based on the severity of their conditions is a critical challenge in healthcare. Manual assessment can be subjective and inefficient.

**Solution Approach:** To tackle this problem, the project incorporated advanced machine learning techniques to automatically assess disease severity and prioritize patients. This ensures that healthcare providers can allocate resources to those in most urgent need of care.

**Problem 3:** **Comprehensive and Standardized Reporting**

**Scope:** Varied reporting practices and formats in medical imaging lead to inefficiencies and inconsistencies in patient care and data management.

**Solution Approach:** The project addressed this problem by developing a system that generates standardized diagnostic reports. These reports ensure that healthcare professionals receive clear and consistent information for making treatment decisions.

**Problem 4: Managing Increasing Volume of Medical Imaging Data**

**Scope:** The exponential growth of medical imaging data poses challenges in data management, retrieval, and analysis.

**Solution Approach:** The project incorporates efficient data management and retrieval techniques to handle the growing volume of medical images, ensuring timely access for diagnosis.

**Problem 5: Adapting to Evolving Healthcare Technology**

**Scope:** The healthcare technology landscape is rapidly evolving, and the project needed to stay adaptable to emerging technologies.

**Solution Approach:** The project adopted a flexible approach, allowing for the integration of the latest advancements in deep learning and medical imaging technology to remain at the forefront of the field.

**Changes in Aims and Reasons:**

Initially, the project aimed primarily at improving disease diagnosis accuracy. However, recognizing the need for comprehensive patient care, we expanded the project's scope to include automated severity assessment, patient prioritization, and standardized reporting. These changes were made to ensure that the project addresses the broader challenges faced in healthcare.

# Hardware, Software analysis and requirements

* **Graphics Processing Unit (GPU):**

The project demands a high-performance GPU, preferably from NVIDIA, to accelerate the training of deep learning models for disease diagnosis and image analysis. An NVIDIA RTX 30 series GPU with a significant amount of VRAM is recommended for handling large medical image datasets and complex deep learning models effectively.

* **Central Processing Unit (CPU):**

A robust CPU with multiple cores is essential to support various software processes. Consider using an Intel Core i9 or AMD Ryzen 9 series CPU, which can efficiently handle data preprocessing, model management, and user interface interactions.

Random Access Memory (RAM):

To cope with memory-intensive tasks associated with deep learning, image manipulation, and data management, a minimum of 32GB RAM is essential. This ensures smooth operation during model training and image processing.

* **Storage Space and Execution:**

Adequate storage space is necessary to store the project's model files, medical image datasets, and user-generated images. Consider a fast SSD with at least 1TB of storage capacity for quick access to data. In cases where local hardware resources are insufficient, cloud services like AWS or Google Colab can be utilized for model training and execution.

**Software Analysis and Requirements:**

1. **Data Gathering and Storage:**

* **NumPy:** For fundamental numerical operations and data handling.
* **Pandas:** For data manipulation and structured data handling.
* **OpenCV:** Essential for image preprocessing, augmentation, and handling.

1. **Medical Imaging Data Handling:**

* **PyDicom:** Specifically, for working with DICOM format medical images.

1. **Data Preprocessing and Image Augmentation:**

* **imgaug:** To perform data augmentation and generate new samples from existing images.
* **scikit-image:** Provides additional image processing functions.
* **Albumentations:** Versatile, deep learning-focused image augmentation.
* **Augmentor:** Simplifies image augmentation for machine learning.

1. **Imbalanced Dataset Handling:**

* **imbalanced-learn:** Useful for addressing imbalanced datasets.

1. **Model Development and Comparative Analysis:**

**Deep Learning Frameworks:**

* **PyTorch:** A popular framework known for flexibility and dynamic computation graphs.
* **TensorFlow:** A widely used framework known for scalability and production readiness.
* **Keras:** A high-level API that can run on top of both PyTorch and TensorFlow.

1. **Model Architecture and Training Tools:**

* **PyTorch Lightning (for PyTorch):** Simplifies PyTorch model training and experiment management.
* **TensorFlow Addons (for TensorFlow):** Offers additional tools for TensorFlow models.

1. **GPU Support (Optional but Recommended):**

* **NVIDIA CUDA Toolkit:** Required if using NVIDIA GPUs for acceleration.

1. **Hyperparameter Tuning:**

**Optimization Libraries (Optional):**

* **Optuna:** A Python library for optimizing machine learning model parameters.
* **Keras Tuner:** Specifically designed for tuning hyperparameters in Keras models.

1. **Model Evaluation:**

* **scikit-learn:** For model evaluation, including metrics, cross-validation, and data splitting.

1. **Deployment and Monitoring (In Production, After Model Selection):\*\***

* **Docker:** For containerization of your deep learning environment and model.

1. **Web Application Frameworks (Optional):**

* **Flask, Django, or FastAPI:** For building web-based applications for model access and interaction.

1. **Model Deployment Tools (In Production):**

* **TensorFlow Serving, TensorFlow Lite, ONNX:** For deploying deep learning models to production.
* **AWS SageMaker, Azure ML:** Cloud-based deployment platforms.

1. **Monitoring and Logging (In Production):**

* **Prometheus, Grafana:** cloud-specific monitoring services.

1. **Security and Compliance (In Production):**

* Tools and practices to ensure data security, encryption, access control, and compliance with healthcare regulations **(e.g., HIPAA)**.

1. **Testing and CI/CD (In Production):**

* **Jenkins, Travis CI, GitLab CI/C:** testing and continuous integration/continuous deployment tools.

1. **Medical Imaging Libraries (Throughout):**

* **SimpleITK:** A high-level image processing library for medical images.
* **ANTsPy:** A library for image registration, normalization, and analysis of medical images.
* **Nibabel:** For reading and writing neuroimaging data, which can be adapted for medical image formats.
* **Medical Image Analysis Libraries (As Needed):** Libraries like MIALab that offer tools for feature extraction, image registration, and segmentation of medical images.

**Project Management:**

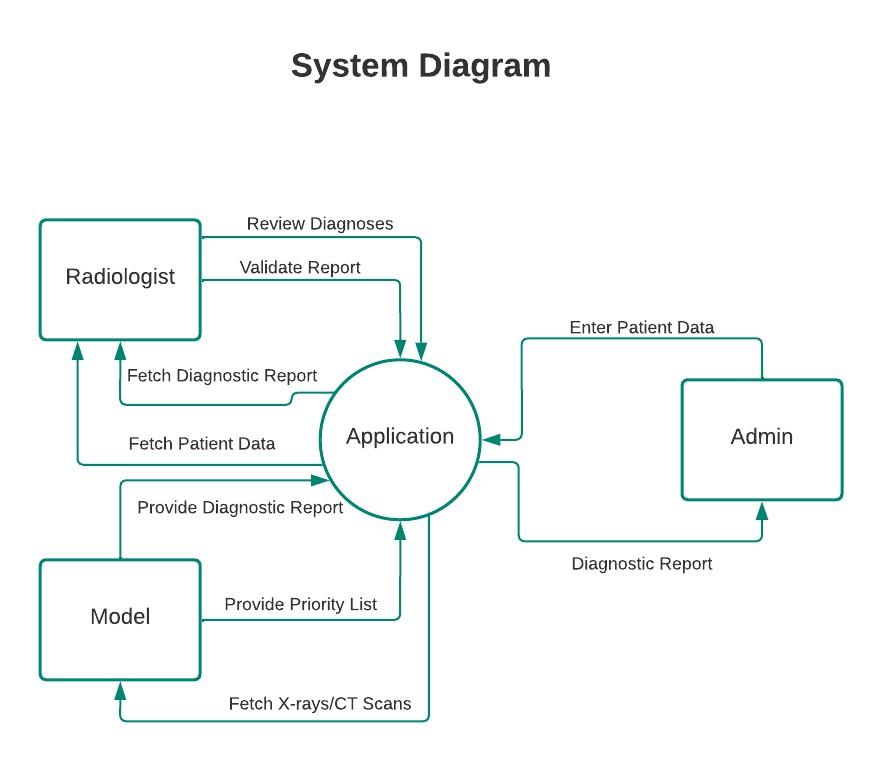
Consider using project management tools like Trello to organize tasks, track progress, and collaborate effectively. Trello can help streamline project workflows, manage deadlines, and ensure efficient project management throughout the development process.

**From User’s POV:**

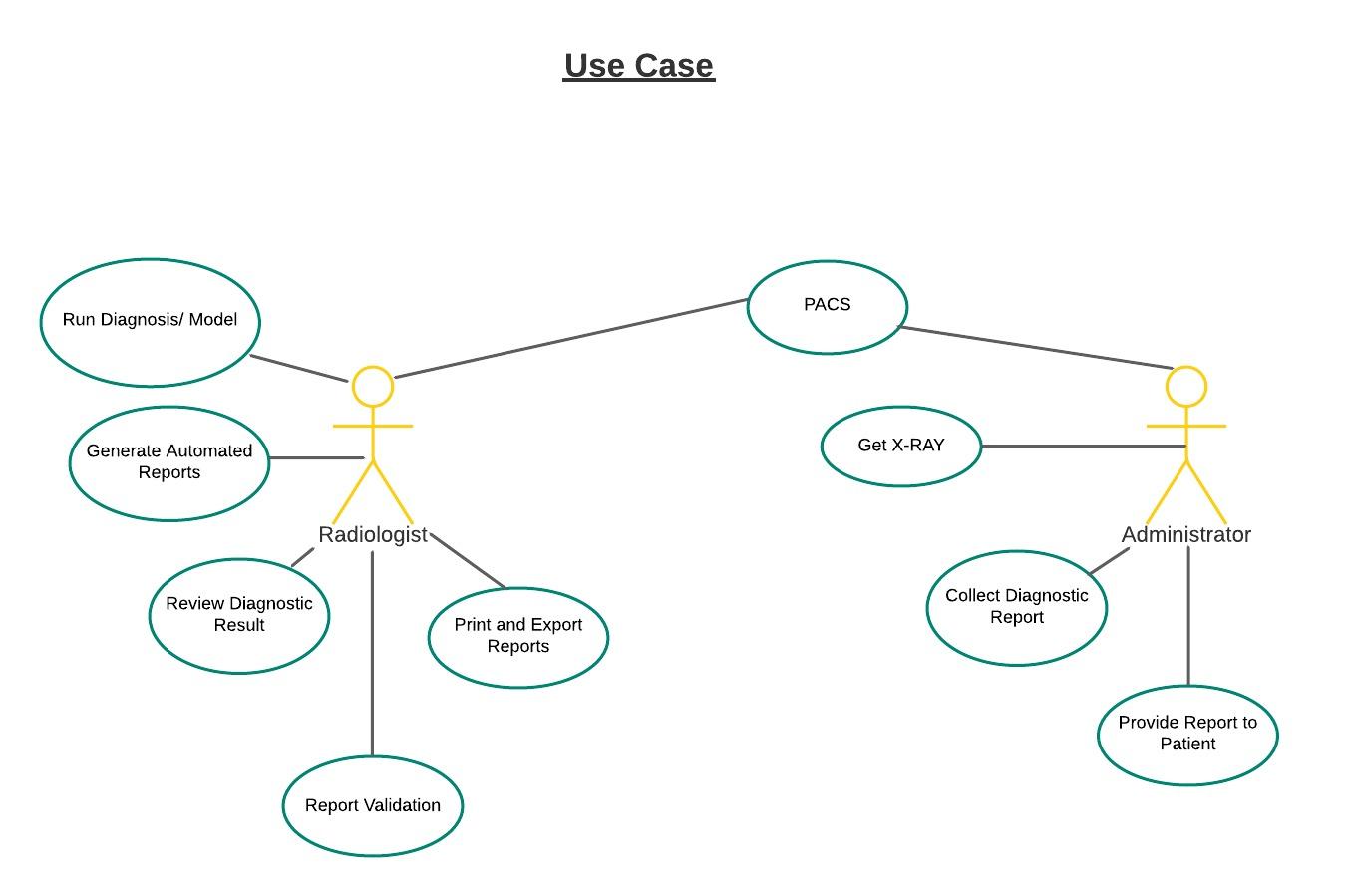
1. As a healthcare provider, I want to efficiently access and interpret medical images through an intuitive user interface, so that I can make accurate diagnoses and treatment decisions.
2. As a radiologist, I want the system to automatically prioritize patients based on the severity of their conditions, so that I can provide timely care to those in critical need.
3. As a medical professional, I want the system to generate standardized diagnostic reports, so that I can improve communication and decision-making among the healthcare team and provide patients with clear, consistent information.
4. As a system administrator, I want secure data storage and retrieval mechanisms to protect patient information and ensure compliance with data privacy regulations, so that patient data remains confidential and secure.
5. As a researcher, I want the system to support the integration of the latest deep learning models and algorithms, so that I can stay at the forefront of medical imaging technology and contribute to advancements in the field.

These user stories represent the core requirements of the project and the needs of different stakeholders involved in medical image analysis and diagnosis.

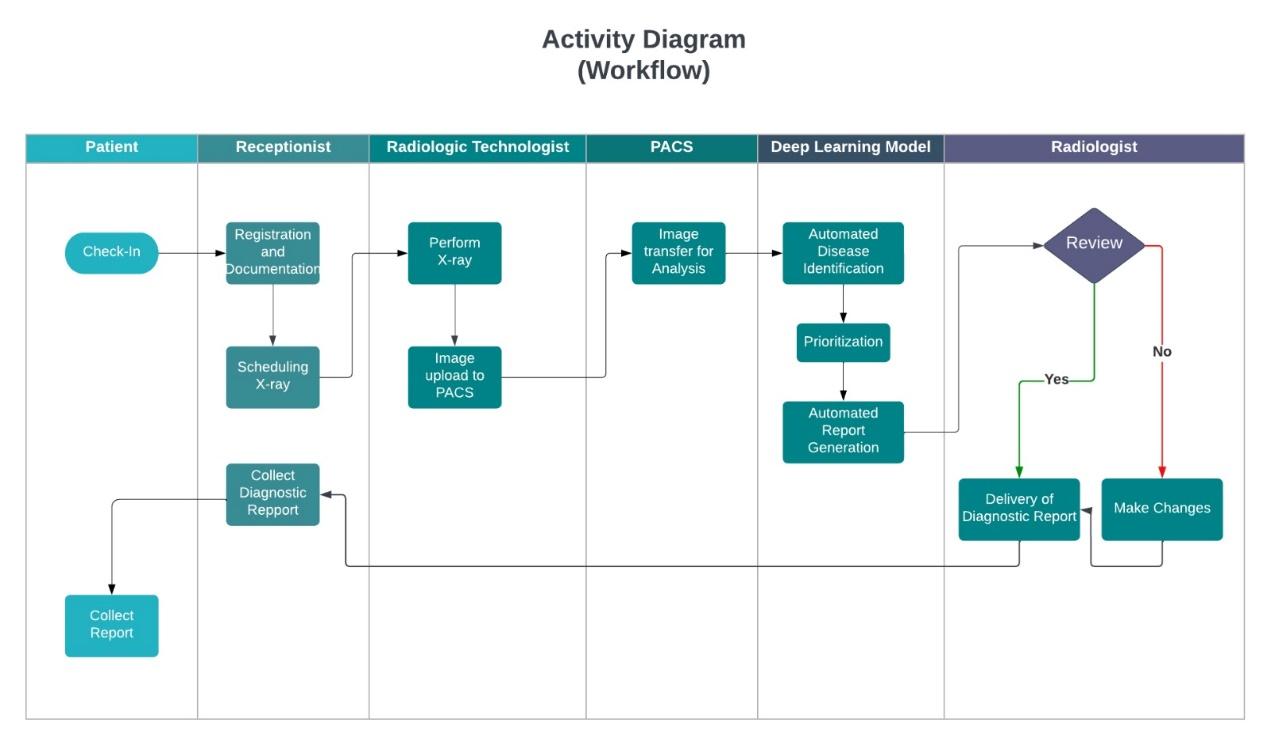
**System Diagram:**



**Actor-Use Case Diagram:**



**Activity Diagram:**



# Software design and modeling

In this chapter project architecture is presented (explain which architecture is used), overall design diagrams (complete object diagram, complete class diagram, database diagram, etc.) to be shown. (as such diagrams are usually huge in nature, it is appropriate to print in on A3 or bigger sheets, and then fold it to A4 size).use few behavioral diagrams (sequence diagram, timing diagram, activity diagram, state transition diagram, or composite diagram) only for core technical functionality of the project against use cases. Show application’s landing/main/home interface, also show high fidelity prototypes (input, output, inquiry, debugging, configuration, and wrapper) on (Image, Audio, video, text, tags, AR, VR) against specific use case.

Resources:

[**https://www.uml-diagrams.org/uml-25-diagrams.html**](https://www.uml-diagrams.org/uml-25-diagrams.html)

[**https://sysml.org/**](https://sysml.org/)

# Algorithm analysis and complexity

In this chapter you have to mention algorithms that are used in project. Its purpose and significance, along with its pseudocode. Compare your selected algorithm with such other algorithms. For each algorithm show its best, average, and worst values in context of time and space complexity. Show primary references of all mentioned algorithms.

# Implementation

Give code details (not a complete listing, but description of key parts). Discuss the most important/interesting aspects. It probably won’t be possible to discuss everything- give a rationale for what you do. Code shall not be more than 3-5 pages. Use appropriate code writing standards, draw operational diagram, component diagram and deployment diagram. Show only two to three technical interfaces that represents the core project functionality with explanation (You may use POC interfaces). Draw state transition diagram of project interface (input, output, and processes).

# Testing

This chapter contains White box of most logical code, and black box testing of that interfaces that represents core functionality of the project1 (onlyshow test cases and their results). Some of the system level structural and functional testing must be shown with the help of tools2. You have to show test plan – how the program/system was verified. Put the actual test results in the annexure.

This chapter also covers results of different types of experiments/simulations that were carried out with the code written. Why were certain experiments in the simulation used and how did they affect the results? If there are very many graphs and tables associated with this Chapter they may be placed in annexure.

***Resources:***

1. Pressman, Roger S. "Software Engineering: A practitioner’s approach." *9th Edition* (2020), Chapter 19-21
2. Perry, William E. *Effective Methods for Software Testing, CafeScribe: Includes Complete Guidelines, Checklists, and Templates*. John Wiley & Sons, 2007, Chapter 8

# Discussion (optional)

This Chapter should fully and logically discuss the progression of the project including the methods used and the results of experimentation, or the design; in such a way that examiner can evaluate the worth of the project. The discussion should be backed by detailed reference to material in the testing chapter of the report.

# Conclusions

This Chapter should be a concise statement of the conclusions which may be drawn from the work attempted. The reader needs to be convinced that the design will work. If Uncertainties remain, they should be pointed out, and alternatives, such as modifying performance specifications, should be spelled out to deal with foreseeable outcome.

# Future work

This Chapter may be used to propose further work which may be carried out on the project in subsequent study projects. Suggestions of this type should be limited to proposals which involve significant amounts of work such as major modifications of equipment or development of student practical experiments/enhancement. If any component is developed, how it can be utilized with proper documentation. Any suggestion is to be given in sufficient detail to provide adequate information for a future student to be able to fully appraise the proposal. which other similar project can be developed by using same concept with different domain/technology.

# Achievements

In this chapter you have to summarize your participations in different competitions, conferences, incubation activities, and exhibitions. It is desired to express your experience about such activities. Also mention what you achieved in such activities e.g experience, acknowledgement, certificates, souvenirs , and rewards. If it is in process show correspondence evidence.

# Appendices

These shall be used to give detailed results that shall be summarized in main text. The normal practice is “Annex A, B, C…” and, when required, “Appendix (to annex) 1, 2, 3…..” They should identify on every page by running header. Following items should be included in appendices

In **acknowledgement chapter**, you may include official letters from organizations.

In **introduction chapter** 2-3 pages about organization for which you are developing the project

In **background and literature review chapter**, research paper that is basis of your project, details of similar projects, any UML diagrams from other sources that has strong relationship with your project

In **hardware, software analysis and requirement,** you may add hardware pacification, use case narrations, or detailed requirement specification document.

In **software design and modeling chapter**, you may add detailed design documents other than most significant.

In **software algorithm and complexity**, you may attach actual algorithm or its research paper.

In **achievement chapter**, you have to mention correspondence (letters, emails etc.), copy of certificate, pictures of participation specially at time of award ceremony.

You may add any detail that is summarized in any chapters but need more focus and clarity for reader.

# General Guidelines

- Begin each chapter on new page.

- Each chapter should have small introduction at beginning of chapter. Introduction must link to previous chapter. It is a one or more then one paragraph but not more than one page that introduces the reader to the subject. The introduction presents basic background material, the history of the problem and contains the key sentence outlining the subjects to be discussed.

The total report length should be **100- 120 pages**; most projects somewhat shorter. There is no value in trying to artificially lengthen your project by ‘padding’ it. Each project is unique and has its own natural length, and you will probably know when you have said everything that you need to be said.

## Typing and size of paper

I. The report is to be typewritten on one side of the paper on international size A4 paper (297mmX210mm). This paper must be good quality bond (70-90 gsm).

ii. Reports length should be 100-120 pages.

iii. Use Times New Roman, size 12 font throughout the reports.

Use 1.5 or double spacing.

## Page number and Chapter number

- Use lower case Roman numerals for preliminary pages

I. Title page (not numbered on page)

ii. Abstract

Table of Contents

The text of the report begins with Arabic number 1. Number all pages. Page numbers can be inserted either at the bottom/top right or the bottom/top center.

All appendices should number as A-1, A-2, etc. for pages under appendix A, and B-1, B-2, etc. for pages under appendix B (See Table of Contents.).

A hierarchical numbering scheme for chapter numbering shall be used. For instance, use 1 for chapter one, 2 for chapter 2, 1.1 for the subsection 1 of chapter 1, etc. (See the Table of Contents).

## Margin boundaries

I. 1 -inch left margin.

ii. 0.5-inch margin on the other three sides.

## Diagrams and figures

Figures and table should be inserted in the text in one of the three places

A full page figure or illustration must be inserted on the left hand side facing the typescript which described it.

Small figure should be incorporated in the text with the legend appearing below (not recommended).

Each graph, figure, etc., should have a figure number and title typed below it. The type style should be same as the text. Figures should be numbered by chapters (Fig. 1.1, Fig 1.2, Fig. 2.1, etc). explain each figure by referring its number (e.g. in Fig 1.1), don’t assume any figure is self-explanatory. Whichever numbering system you use, make sure that you follow the same system for tables and equations, also explain then as figures.

Line drawings, graphs, and monograms should be in bold clear lines. Where graphs, diagrams and figures cannot be mounted vertically on the page these are to be mounted and labeled in such a way that they can be read from the right hand side(900 on the page) of the page .

All the axes of graphs are to be labeled with the parameter and its units. Information on illustrations and graphs such as labels, scales etc. must be typewritten.

## Photocopying

All the figures, etc. must be reproduced by an electronic or electrostatic or photographic method which is known not to fade.

## Fixing of photograph

Full page photographs should be bound into the report. Small photographs must be firmly fixed to the paper. An alternative is to use color photocopying or digital processing.

## Tables

Each table should be numbered consecutively (Table 1, Table 2) or by chapter (Table 1.1, Table 1.2, Table 2.1).Table number should be centre above the top of the table and be followed on the next line by a brief descriptive caption, preferably in cap. The type should be the same as the text. Refer to each table in text by number “In Table 1, one can clearly see………”The same rules for location of figures apply to tables.

TABLE 1. MEASURED RESISTOR VALUES AND THE METER ERROR

| Nominal Value Marked | Measured Value | Error (%) |
| --- | --- | --- |
|  |  |  |
|  |  |  |

## Equations

Centre each equation on separate line. Number equations consecutively in parentheses at the right margin. Equation may be referenced by number in the text, using parentheses around the number.

Y (t) = ∫sin (x) dx (1)

## Units

The S.I. system of units is to be used throughout. Where difficulties are introduced by quotation of imperial units from reference source, these should be accompanied by the appropriate conversion to S.I. units in parentheses.

## References

At the end of your work, list full details of all of the sources which you have cited in your text in a section headed *References*, in numeric order. References listed must follow IEEE formatting guidelines (see reference examples overleaf). Your reference list should allow anyone reading your work to identify and find the material to which you have referred.

In IEEE style your reference list should be formatted in the following way:

* Align references left
* Single-space each entry, double-space between every new entry
* Place number of entry at left margin, enclose in square brackets [ ] Indent text of entries

### Citations/references with multiple authors

If you choose to mention the author(s) of a source whilst citing it in the text of your work, if there are three or more you can abbreviate them using ‘et al.’ e.g. During their research, Fan, et al. [4] discuss lasers in detail. However, in general you do not need to mention the authors by name, just use the numeric citation in square brackets. In your full reference list at the end however, you always give the authors’ names. In the reference list you can only abbreviate these using ‘et al.’ if there are six or more authors.

### Reference examples

There are standard reference formats for most types of document. Below are examples of the most common types of document you might want to reference. Each of the following gives a suggested standard format for the reference followed by examples for the different document types.

### Book

[Ref number] Author’s initials. Author’s Surname, *Book Title*, edition (if not first). Place of publication: Publisher, Year.

[1] I.A. Glover and P.M. Grant, *Digital Communications*, 3rd ed. Harlow: Prentice Hall, 2009.

### Book chapter

[Ref number] Author’s initials. Author’s Surname, “Title of chapter in book,” in *Book Title,* edition (if not first), Editor’s initials. Editor’s Surname, Ed. Place of publication: Publisher, Year, page numbers.

[2] C. W. Li and G. J. Wang, "MEMS manufacturing techniques for tissue scaffolding devices," in *Mems for Biomedical Applications*, S. Bhansali and A. Vasudev, Eds. Cambridge: Woodhead, 2012, pp. 192-217.

### Electronic Book

[Ref number] Author’s initials. Author’s Surname. (Year, Month Day). *Book Title* (edition) [Type of medium]. Available: URL

[3] W. Zeng, H. Yu, C. Lin. (2013, Dec 19). *Multimedia Security Technologies for Digital Rights Management* [Online]. Available: http://goo.gl/xQ6doi

Note: If the e-book is a direct equivalent of a print book e.g. in PDF format, you can reference it as a normal print book.

### Journal article

[Ref number] Author’s initials. Author’s Surname, “Title of article,” *Title of journal abbreviated in Italics,* vol. number, issue number*,* page numbers, Abbreviated Month Year.

[4] F. Yan, Y. Gu, Y. Wang, C. M. Wang, X. Y. Hu, H. X. Peng, et al., "Study on the interaction mechanism between laser and rock during perforation," *Optics and Laser Technology,* vol. 54, pp. 303-308, Dec 2013.

Note: the above example article is from a journal which does not use issue numbers, so they are not included in the reference.

### E-Journal article

PDF versions of journal articles are direct copies of the print edition, so you can cite them as print journals.

[Ref number] Author’s initials. Author’s Surname. (Year, Month). “Title of article.” *Journal Title* [type of medium]. volume number, issue number, page numbers if given. Available: URL

[5] M. Semilof. (1996, July). “Driving commerce to the web-corporate intranets and the internet: lines blur”. *Communication Week* [Online]. vol. 6, issue 19. Available: http://www.techweb.com/se/directlinkcgi?CWK19960715S0005

**When you are compiling your reference list you may abbreviate journal titles:**

For a list of IEEE abbreviations go to:

<https://www.ieee.org/documents/trans_journal_names.pdf>

For non IEEE journal abbreviations go to:<http://www.bath.ac.uk/library/help/infoguides/abbreviations.html>

For further information on the common abbreviations of words used in references for the IEEE style go to:

<http://www.ieee.org/documents/style_manual.pdf>

### Conference papers

[Ref number] Author’s initials. Author’s Surname, “Title of paper,” in *Name of Conference,* Location, Year, pp. xxx.

[6] S. Adachi, T. Horio, T. Suzuki. "Intense vacuum-ultraviolet single-order harmonic pulse by a deep-ultraviolet driving laser," in *Conf.* *Lasers and Electro-Optics*, San Jose, CA, 2012, pp.2118-2120.

Standard abbreviations may be applied to the title of the conference. For a table of abbreviations go to: <http://www.ieee.org/documents/ieeecitationref.pdf>

### Reports

The general form for citing technical reports is to place the name and location of the company or institution after the author and title and to give the report number and date at the end of the reference. If the report has a volume number add it after the year.

[Ref number] Author’s initials. Author’s Surname, “Title of report,” Abbreviated Name of Company., City of Company., State, Report number, year.

[7] P. Diament and W. L. Luptakin, “V-line surface-wave radiation and scanning,” Dept. Elect. Eng., Colombia Univ., New York, Sci Rep. 85, 1991.

### Patents

[Ref number] Author’s initials. Author’s Surname, “Title of patent,” Country where patent is registered. Patent number, Abbrev of Month Day Year.

[8] J. P. Wilkinson, “Nonlinear resonant circuit devices,” U.S. Patent 3 624 125, July 16 1990.

Note: Use “issued date” if several dates are given.

### Standards

[Reference number] *Title of Standard*, Standard number, date.

[9] *Shunt power capacitors*, IEEE standard18-2012, 2013.

### Theses/Dissertations

[Ref number] Author’s initials. Author’s Surname, “Title of thesis,” Designation type, Abbrev. Dept., Abbrev. Univ., City of Univ., State, Year.

[10] J. O. Williams, “Narrow-band analyser,” Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

### Datasheets

[Ref number] Author’s initials. Authors Surname, “Title of Datasheet,” Part datasheet, Publication date [Latest revision date].

[11] Texas Instruments, “High speed CMOS logic analog multiplexers/demultiplexers,” 74HC4051 datasheet, Nov. 1997 [Revised Sept. 2002].

### Online Documents

If you are using documents such as a report, conference paper, standard, patent or thesis online and it also exists as an identical print equivalent i.e. with the same format and pagination, it can be usually be referenced as the print version.

If it is e-only, you can make the standard reference template an electronic version by adding the material type in square brackets

e.g. [Online] after the document title. If there is no specific document title you can place this after the document number (e.g. patent number).

At the end of the reference add: Available: URL. See below for an example of an online patent:

[12] M.R. Brooks, “Musical toothbrush with adjustable neck and mirror,” U.S Patent *326189* [Online], May 19 1992. Available: http://goo.gl/VU1WEk

### Websites

Note: Include as much of the key information as you can find for a given website. If a web page has no personal author, you can use a corporate author. Failing that, you can use either Anon. (for anonymous) or it is permissible to use the title of the site.

[Ref number] Author’s initials. Authors Surname. (Year, Month. Day). *Title of web page* [Online]. Available: URL

1. BBC News. (2013, Nov. 11). *Microwave signals turned into electrical power* [Online]. Available: http://www.bbc.co.uk/news/technology-24897584

1. M. Holland. (2002). *Guide to citing internet sources* [Online]. Available: http://www.bournemouth.ac.uk/library/using/guide\_to\_citing\_internet\_sourc.html