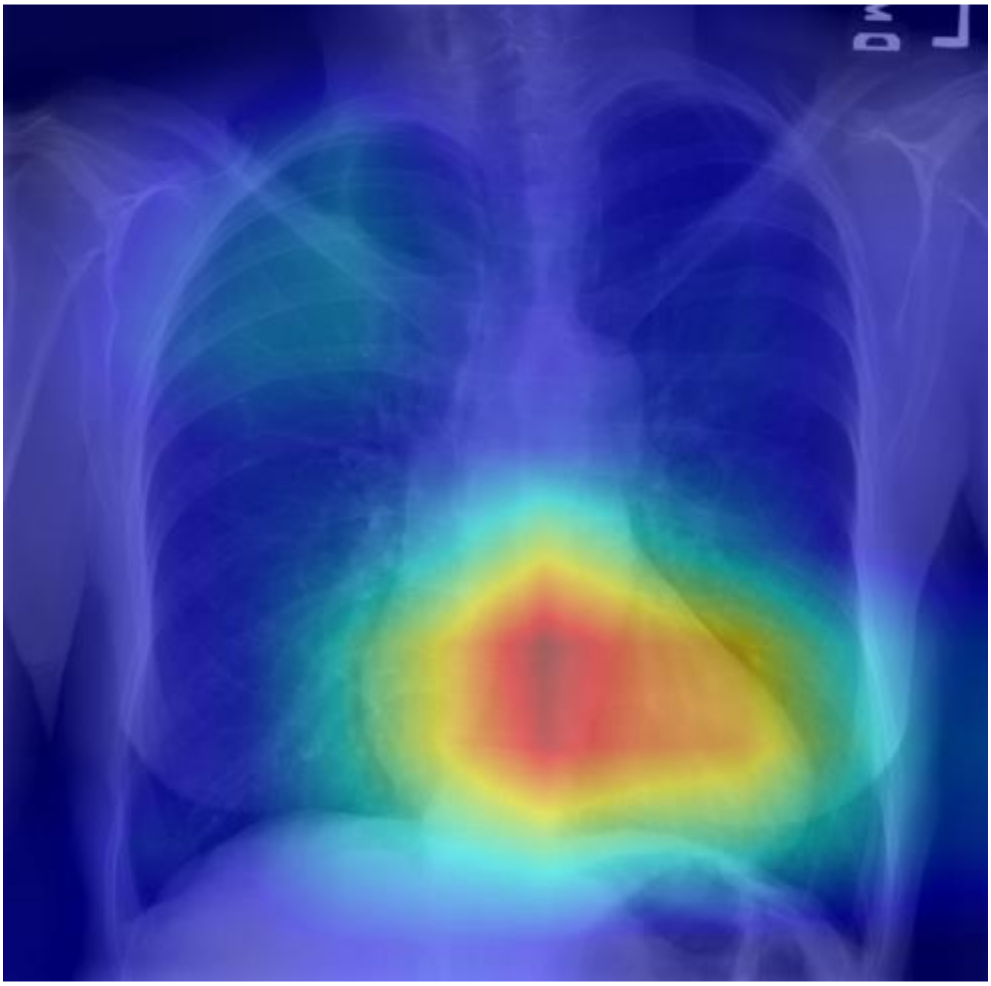


CHEST X RAY ANALYSIS

Project Report

Submitted by:
Amritha Pradeep

INDEX

ACKNOWLEDGEMENT

In completing this project, I am grateful for the support and guidance of various individuals and organisations who have contributed significantly to its successful outcome.

Firstly, I would like to express my deepest gratitude to my mentor Mr. Adarsh P S and project supervisor Ms. Arya Krishna, for their invaluable insights, feedback, and mentorship throughout the project journey. Their expertise and guidance have been instrumental in shaping the direction and scope of this project.

I am also thankful to my teammates and colleagues for their collaborative efforts, dedication, and shared vision, which have brought this project to fruition. Their contributions have been essential to the project's success, and I am grateful for their willingness to share their knowledge and skills.

Additionally, I would like to acknowledge the contributions of my friends and family who have offered unwavering support and encouragement, helping me overcome challenges and stay motivated. Their enthusiasm and support have been a constant source of motivation.

Lastly, I would like to acknowledge the support received from Technovally Software solution for providing the necessary resources and encouragement that enabled this collaborative effort.

In conclusion, I am deeply grateful for the contributions of all those who have supported me throughout this project journey. Their support, guidance, and encouragement have played a significant role in making this project a reality.

ABSTRACT

Chest X-Ray classification is a critical task in medical imaging, aimed at identifying various thoracic diseases from radiographic images. This project introduces a deep learning model based on the DenseNet-121 architecture, tailored for the classification of 14 distinct chest conditions. The model leverages the ChestX-Ray14 dataset, comprising over 112,000 labelled images, to train and validate its performance.

The model employs advanced techniques in deep learning and transfer learning to enhance its diagnostic accuracy. The model's architecture is designed to capture intricate patterns in chest X-Rays, facilitating the detection of conditions such as pneumonia, fibrosis, and cardiomegaly.

This project addresses the issue of class imbalance in the dataset by utilizing a customized weighted cross-entropy loss function, ensuring more accurate training across all classes. The model was trained using TensorFlow, with the training process significantly accelerated by the use of pre-trained weights and optimized with the Adam optimizer.

Development Roadmap:

Phase 1: Planning and Research

- Define Objectives: Clearly outline the goals, such as improving diagnostic accuracy and reducing radiologist workload.
- Dataset Acquisition: Obtain the ChestX-Ray14 dataset from NIH, ensuring you have the necessary permissions and understand the dataset's structure.
- Exploratory Data Analysis: This helps better understand the data and the patterns within the data to further develop a model based on the data.

Phase 2: Data Preparation

- Data Splitting: Divide the dataset into training, validation, and test sets

Phase 3: Model Development

- Model Selection: Choose the DenseNet-121 architecture as the base model.
- Transfer Learning: Utilize pre-trained weights to initialize the model

Phase 4: Training

- Hyperparameter Tuning: Experiment with different learning rates, batch sizes, and optimizers to find the best configuration.
- Training the Model: Train the model on the training set, using the validation set to monitor performance and prevent overfitting.
- Loss Function: Implement a customized weighted cross-entropy loss to address class imbalance.

Phase 5: Evaluation

- Performance Metrics: Evaluate the model using metrics like AUC.
- Visualization: Use Grad-CAM to create localized heatmaps for each class to interpret the model's predictions.

Expected Outcome:

The expected outcome of the project is to develop an automated, accurate, and efficient model capable of classifying 14 distinct chest conditions from frontal chest X-ray images.

The model aims to assist in early and accurate diagnosis of chest conditions, particularly in regions with limited access to trained radiologists. By providing reliable and rapid diagnostic support, the model could improve healthcare delivery, especially in rural areas where access to radiological expertise is limited. Additionally, the project seeks to demonstrate the feasibility of using deep learning models like DenseNet-121 in medical imaging tasks, potentially paving the way for further applications in other medical diagnostics.

INTRODUCTION

This project presents a customized DenseNet-121 model designed to classify 14 different chest conditions from frontal-view chest X-rays. The model leverages deep learning and transfer learning techniques to improve classification accuracy while significantly reducing training time.

The goal of this project is to develop a robust and efficient model that can assist radiologists in diagnosing chest conditions, particularly in areas with a shortage of medical professionals. By automating the classification process, the trained model has the potential to enhance diagnostic accuracy and speed, ultimately contributing to better healthcare outcomes.

BACKGROUND OF STUDY

Chest X-rays are one of the most commonly used imaging techniques worldwide, playing a critical role in the early detection, diagnosis, and treatment of various chest-related conditions, such as pneumonia, lung cancer, and heart disease. Despite their widespread use, accurately interpreting chest X-rays remains a challenging task due to the complexity of the human chest's anatomical structures and the subtle differences in X-ray images that various medical conditions present.

To address these challenges, there has been a growing interest in developing automated systems that can assist in the interpretation of chest X-rays. Recent advancements in deep learning, particularly in Convolutional Neural Networks (CNNs), have shown great promise in medical image classification tasks. DenseNet, a state-of-the-art CNN architecture, has been particularly effective in image classification due to its ability to improve feature propagation and reduce the number of parameters needed, making it well-suited for deep networks.

The study’s ultimate goal is to provide a tool that can assist radiologists in making faster and more accurate diagnoses, particularly in resource-limited settings, thereby improving access to quality healthcare and reducing the burden on overstretched medical professionals.

STATEMENT OF STUDY

The study aims to develop a deep learning-based model, specifically a customized DenseNet-121, to automate the classification of chest X-ray images for detecting 14 distinct chest conditions. This approach addresses the significant challenge of limited access to trained radiologists, especially in remote areas like Nepal, where there is a severe shortage of medical professionals capable of interpreting diagnostic images. By leveraging transfer learning and deep neural network architectures, the study seeks to enhance the accuracy and speed of chest condition detection, providing a tool that can potentially match or exceed the diagnostic performance of existing models and professional radiologists. The goal is to improve early detection and treatment outcomes for chest-related illnesses through efficient and accessible technology-driven solutions.

THE SOLUTION

To address the challenges of chest X-ray classification and improve diagnostic accuracy, especially in regions with limited access to radiologists, this study proposes a deep learning-based solution using a customized DenseNet-121 model. The solution is designed to automatically classify chest X-rays into 14 distinct chest conditions, thereby aiding in early detection and treatment of various chest-related diseases.

DenseNet architectures are known for their efficiency in feature reuse and gradient flow, which helps in training deep networks more effectively. The standard DenseNet-121 model is adapted by replacing its final fully connected layer with a global average pooling layer, followed by a new fully connected layer that outputs a 14-dimensional vector, each representing the probability of one of the chest conditions.

The ChestX-Ray14 dataset exhibits significant class imbalance, with some conditions being far less common than others. To address this issue, the solution employs a customized weighted cross-entropy loss function. This function assigns different weights to each class based on their frequency, ensuring that the model does not become biased towards the more prevalent classes.

To make the predictions interpretable, we use Gradient-weighted Class Activation Mapping (Grad-CAM) to generate heatmaps highlighting the regions of the X-ray images that contributed most to the model's decision. This feature not only adds transparency to the model's operations but also assists healthcare professionals in understanding and trusting the AI-driven diagnostic process.

Aims and Objectives:

The objective of the project is to develop a deep learning model to accurately diagnose 14 chest conditions from X-Ray images. This aims to improve diagnostic accuracy, reduce radiologist workload, and enhance healthcare access, especially in underserved areas. It also makes use of Gradient-weighted Class Activation Mapping (Grad-CAM) to generate heatmaps highlighting the regions of the X-ray images that contributed most to the model's decision.

Scope and Limitations:

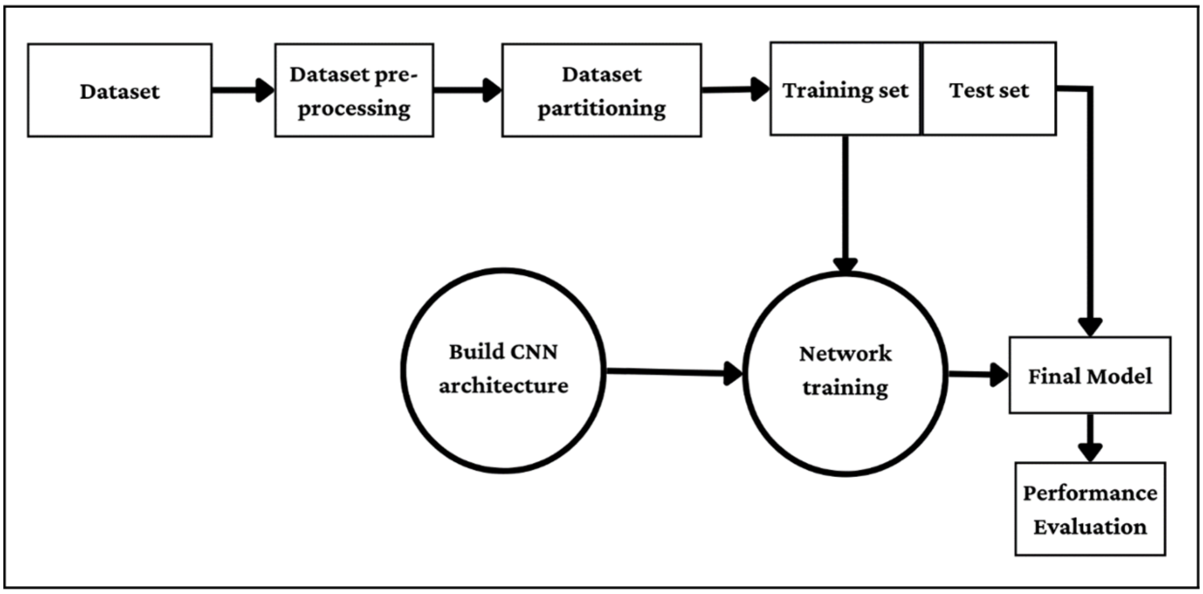
Scope

- **Automated Diagnosis:** Develop a model to accurately diagnose 14 chest conditions from X-Ray images.
- **Efficiency:** Ensure the model is efficient in terms of training and inference times.
- **Healthcare Access:** Provide a scalable solution to address the shortage of radiologists, especially in underserved areas.
- **Interpretability:** Use Grad-CAM to generate heatmaps for better understanding of model predictions.

Limitations

- **Data Quality:** The model's performance is dependent on the quality and diversity of the ChestX-Ray14 dataset.
- **Class Imbalance:** Handling imbalanced classes in the dataset can be challenging and may affect model accuracy.
- **Generalization:** The model may not generalize well to X-Ray images from different sources or with different imaging conditions.
- **Clinical Validation:** Extensive clinical validation is required to ensure the model's reliability in real-world settings.
- **Computational Resources:** Training deep learning models requires significant computational power, which may not be accessible to all users.

RESEARCH METHODOLOGY



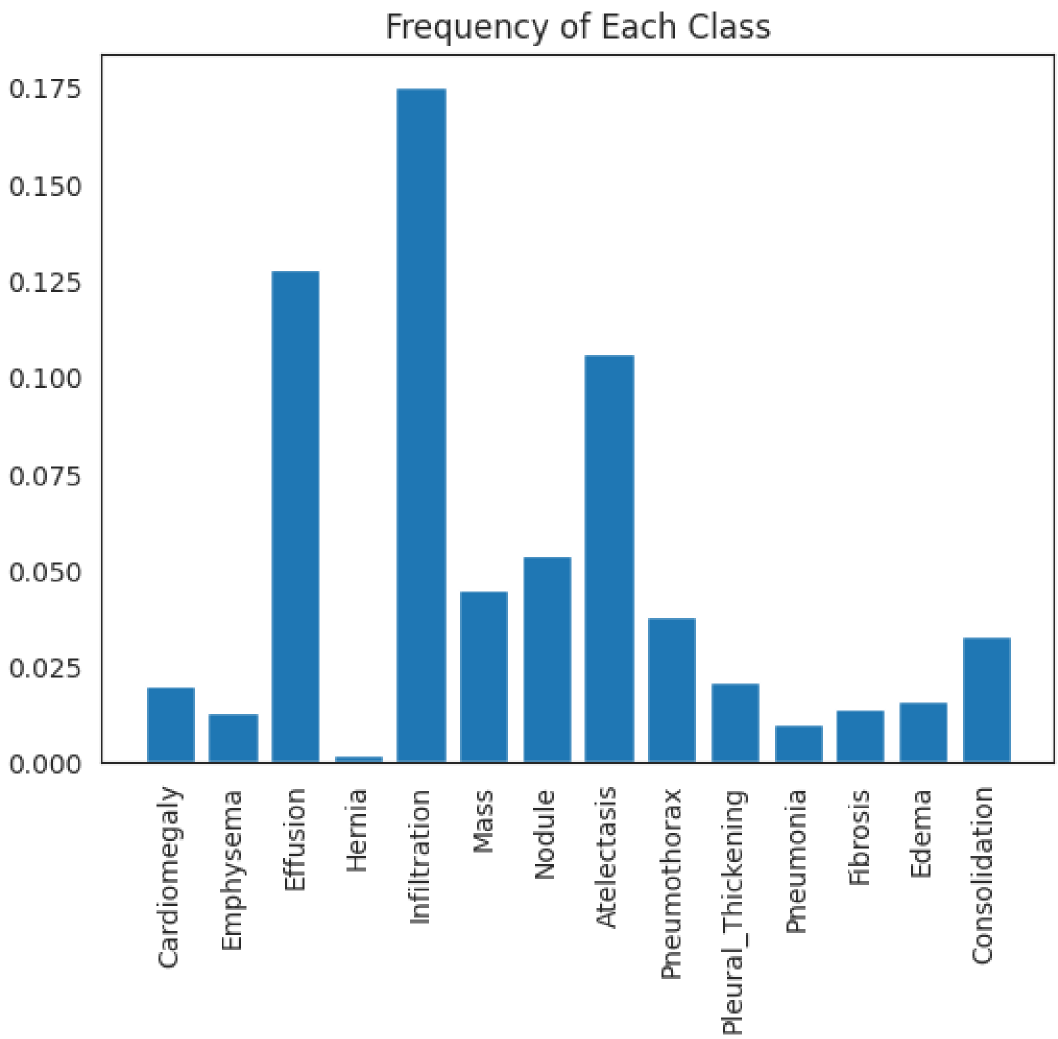
Data Preprocessing:

The labels for all images were stored in a CSV file. The file was then loaded into dataframe using pandas. The path to each image was obtained and added to dataframe. All of labels were identified and one-hot encoding was performed. The images were normalized based on the mean and standard deviation of images in the dataset. The images were then resized to target size of 320x320.

Loss function and Class Imbalance:

The dataset that was used to train the model is prone to class imbalance problem. EDA on the training dataset led to the following results:

- The most unbalanced pathology is Hernia, with 0.1% of patients testing positive for training.
- However, only 17.5% of the training instances for the Infiltration pathology, which has the least degree of imbalance, have been classified as positive.



For a balanced data set the loss function is:

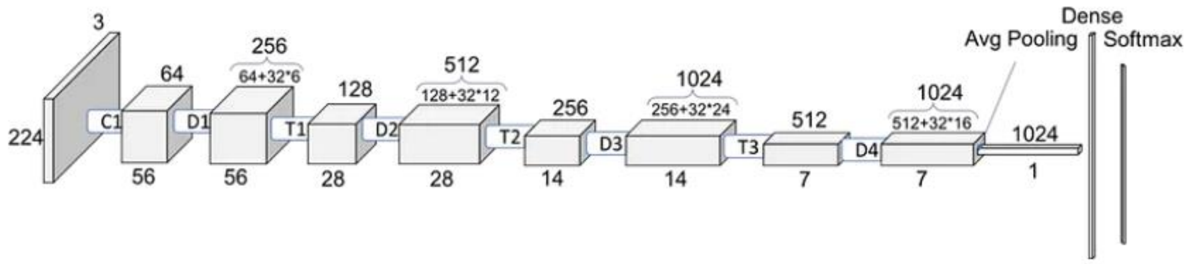
$$L(xi) = -(yilog(f(xi)) + (1 - yi)log(1 - f(xi)))$$

where xi and yi are the input features and their corresponding labels and f(xi) is the output of the model which indicates the probability that it is positive. With the use of this formulation, we can observe that the loss will be dominated by the negative class in situations when there is a significant imbalance and there are few positive training events. One way of balancing such datasets require multiplying each class by a class-specific weight factors, w_p and w_n where w_p is the frequency of negative samples and

w_n is the frequency of positive samples for each class. Then the previous unweighted loss function was modified as:

$$Lw(x) = -(wpylog(f(x)) + wn(1-y)log(1- f(x)))$$

DenseNet-121 Architecture:

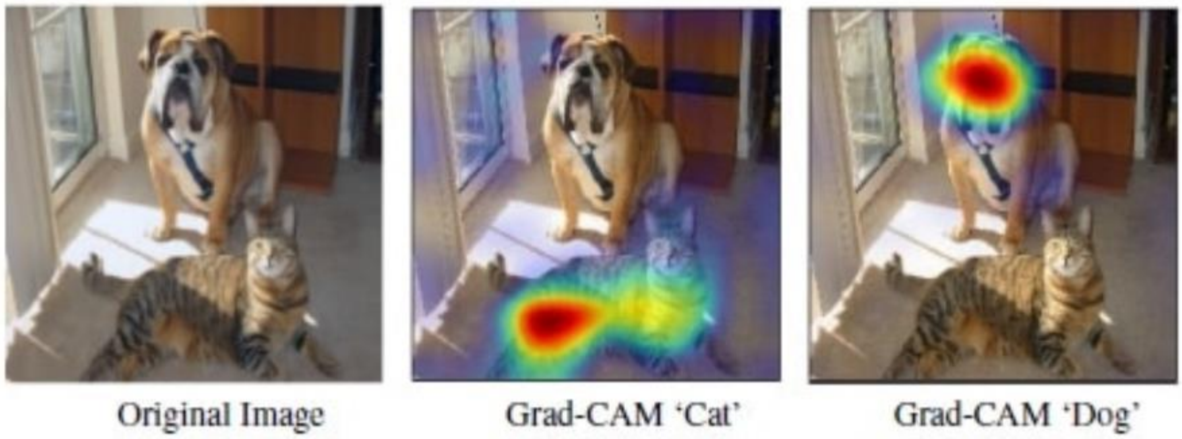


DenseNet-121 has 121 layers, including convolutional layers, dense blocks, and transition layers. Compared to other deep networks, DenseNet-121 has fewer parameters due to its efficient use of feature maps and the dense connectivity pattern.

Unlike traditional CNNs that use sequential layers, DenseNet introduces dense connections between layers. In a DenseNet, each layer receives input from all previous layers and passes its own feature maps to all subsequent layers. This means that each layer has direct access to the gradients and feature maps from all preceding layers, which helps in mitigating the vanishing gradient problem and encourages feature reuse.

The presented model is a customized Densent-121 a 121-layer Convolutional Neural Network trained on the ChestX-ray14 dataset. DenseNets enhance information flow and gradients inside the network, making very deep network optimization feasible.

Grad-CAM:



The gradient-weighted class activation map (Grad CAM) produces a heat map that highlights important regions of an image using the target gradients (dog, cat) of the final convolutional layer.

Grad CAM consists in finding out which parts of the image have led a convolutional neural network to its final decision. This method consists of producing heat maps representing the activation classes on the images received as input. Each activation class is associated with a specific output class.

SIGNIFICANCE OF STUDY

- **Addressing Radiologist Shortage:** The study highlights the critical shortage of radiologists in Nepal, especially in rural areas, making it difficult for patients to access diagnostic imaging services2.
- **Early Detection:** The model aims to improve early detection of chest illnesses, which are a major cause of morbidity and mortality.
- **Healthcare Improvement:** By automating chest X-ray classification, the model can enhance healthcare delivery and provide access to medical imaging expertise in underserved regions.

LITERATURE REVIEW

The literature review is an expressive study based on a detailed review of earlier prominent studies related to the various concepts of Chest X Ray analysis to discover the concept of using deep learning to automate this task. It highlights the importance and challenges, factors affecting usage of deep learning for this task. The architecture and usage of DenseNet for image classification task was studied as well the using Grad-CAM for further interpreting the results.

Here are a few of the papers studied:

- Karna, Ankit & Jha, Aadarsh & Dahal, Alish & Pandey, Anup & Jha, Tantra. (2023). Chest X-Ray Classification using DenseNet.
- R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 618-626, doi: 10.1109/ICCV.2017.74. keywords: {Visualization;Cats;Dogs;Computer architecture;Knowledge discovery},
- Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

PROJECT DETAILS

This project presents a model for detecting 14 different chest conditions from X-ray images that include **Effusion, Cardiomegaly, Emphysema, Nodule, Pneumonia, Pleural Thickening, Hernia, Fibrosis, Infiltration, Pneumothorax, Edema, Consolidation, Mass and Atelectasis.**

This model is a customized version of **DenseNet-121**, a deep convolutional neural network, and it leverages transfer learning techniques to achieve high accuracy and faster training times.

The model was trained on the **ChestX-Ray14 dataset**, which includes over 112,000 frontal view X-ray images from 30,805 patients. Various tools like **NumPy, Pandas, Matplotlib, and Seaborn** were used for data analysis and visualization, and implemented the model in TensorFlow.

The model achieved a mean **AUC of 0.82**. It also leverages the usage for **Grad-CAM** to further understand the results by visualizing them using heatmaps. The project highlights the potential of using this trained model to aid in the early detection and diagnosis of chest conditions, especially in regions with a shortage of radiologists.

TOOLS USED

Python Libraries:

- **Pandas (pd):** Useful for data manipulation and analysis, particularly for handling datasets, like loading CSV files containing labels and image paths, performing data preprocessing, and managing data frames.
- **NumPy (np):** Provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. It's essential for numerical operations needed in image processing.
- **OS (os):** Useful for interacting with the operating system, such as reading files, navigating directories, and handling paths which is important for loading images and managing file structures.
- **Random (random):** Used for generating random numbers, which can be helpful in tasks like shuffling datasets, splitting data into training, validation, and test sets, or augmenting data.
- **Matplotlib (plt):** A plotting library that is great for visualizing data. It will help in plotting graphs, such as training loss curves, histograms of data distribution, or displaying images.
- **Seaborn (sns):** A Python visualization library based on Matplotlib that provides a high-level interface for drawing attractive statistical graphics, useful for creating more advanced plots like heatmaps, box plots, or violin plots.
- **OpenCV (cv2):** An open-source computer vision library that provides tools for image processing, such as reading, resizing, and augmenting images, which will be crucial for preprocessing the X-ray images.
- **TensorFlow Keras Image Preprocessing (ImageDataGenerator):** This tool is used for data augmentation, which helps in artificially expanding the size of a training dataset by creating modified versions of images. Common augmentations include rotation, zooming, shifting, and flipping, which can help improve the robustness and generalization of your model.
- **DenseNet121**: A pre-trained convolutional neural network that is 121 layers deep. DenseNet architectures are known for their dense connectivity between layers, which improves gradient flow and allows for deeper networks. DenseNet121 can be used as a feature extractor in transfer learning, where you fine-tune the model on your specific dataset.

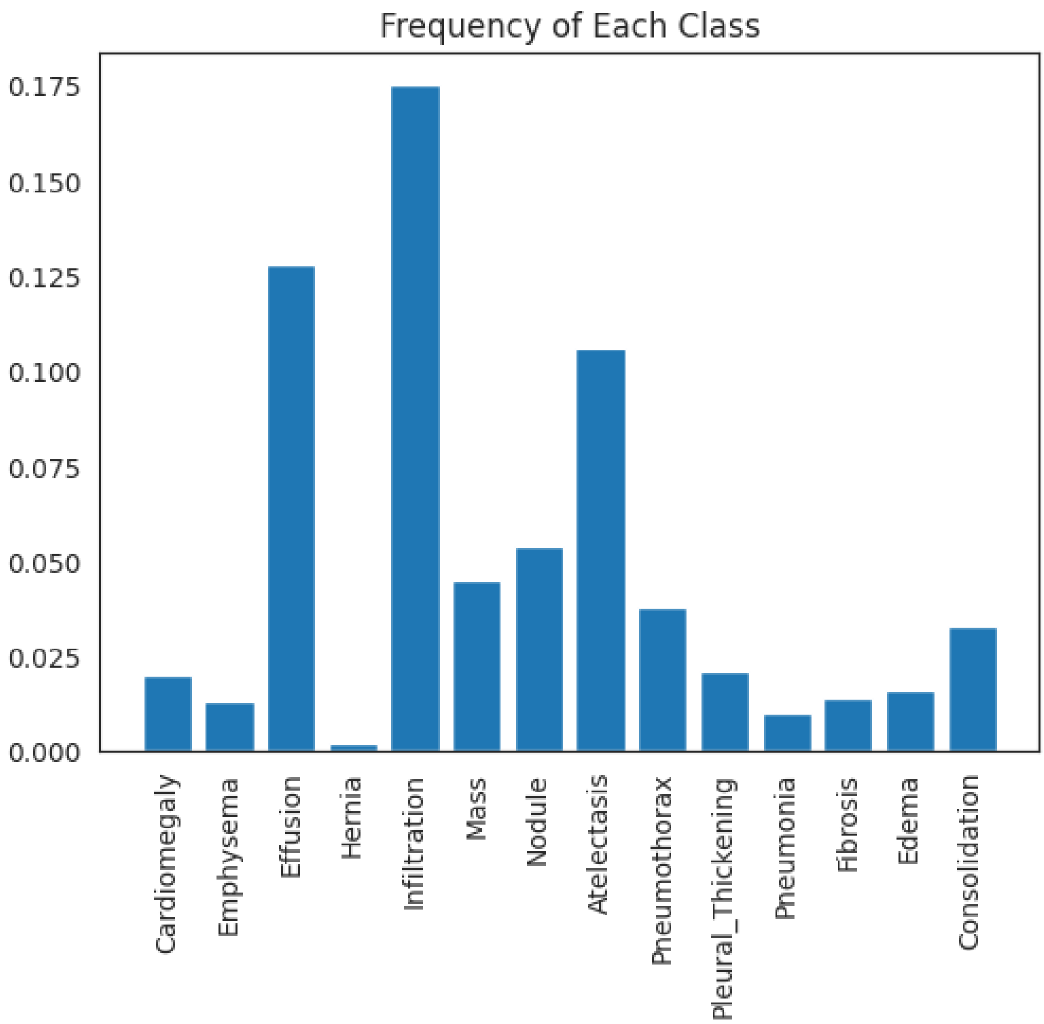
- **Keras Layers (Dense, GlobalAveragePooling2D):**
  - **Dense Layer**: Fully connected layer used for making predictions from features extracted by the convolutional layers. It often serves as the output layer in classification tasks.
  - **GlobalAveragePooling2D**: Reduces the spatial dimensions of the feature maps from the CNN to a single value per feature map, effectively summarizing each feature map and reducing the model's parameters.
- **Keras Model (Model):** This is used to define the architecture of the neural network. You can build custom models by chaining different layers together, setting input and output configurations.
- **TensorFlow (tf):** This is the core framework for defining, training, and deploying machine learning models. TensorFlow provides the backend support for running Keras models on various hardware accelerators like GPUs.
- **ROC Curve (roc_curve):** The ROC (Receiver Operating Characteristic) curve is a graphical representation of a classifier's performance across different threshold values. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. This curve helps visualize the trade-off between sensitivity (recall) and specificity.
- **ROC AUC Score (roc_auc_score):** The ROC AUC (Area Under the Curve) score is a single scalar value that summarizes the overall performance of the classifier. A higher AUC indicates a better performing model, with 1.0 being a perfect classifier and 0.5 representing random guessing.
- **Keras load_img:** This function is used to load an image from a file path into a PIL (Python Imaging Library) format, which is easy to manipulate. You can specify the target size of the image to resize it while loading.
- **Keras img_to_array:** This function converts the PIL image into a NumPy array format, which is necessary for feeding the image into a Keras model. The array format is required as models operate on numerical data.

Grad-Cam:

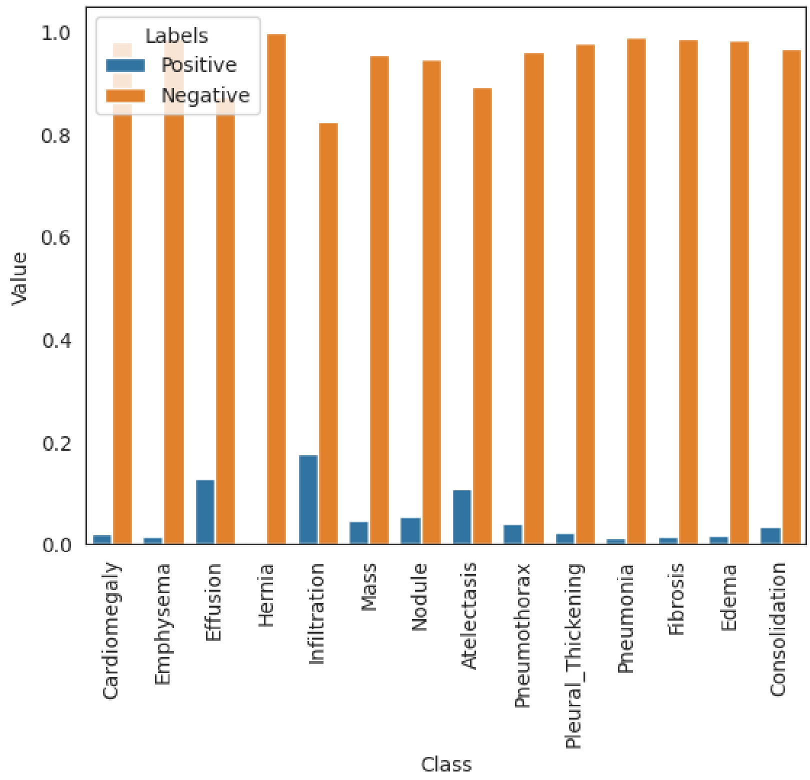
Grad-CAM uses the gradient of the classification score with respect to the convolutional features determined by the network in order to understand which parts of the image are most important for classification.

OUTPUT

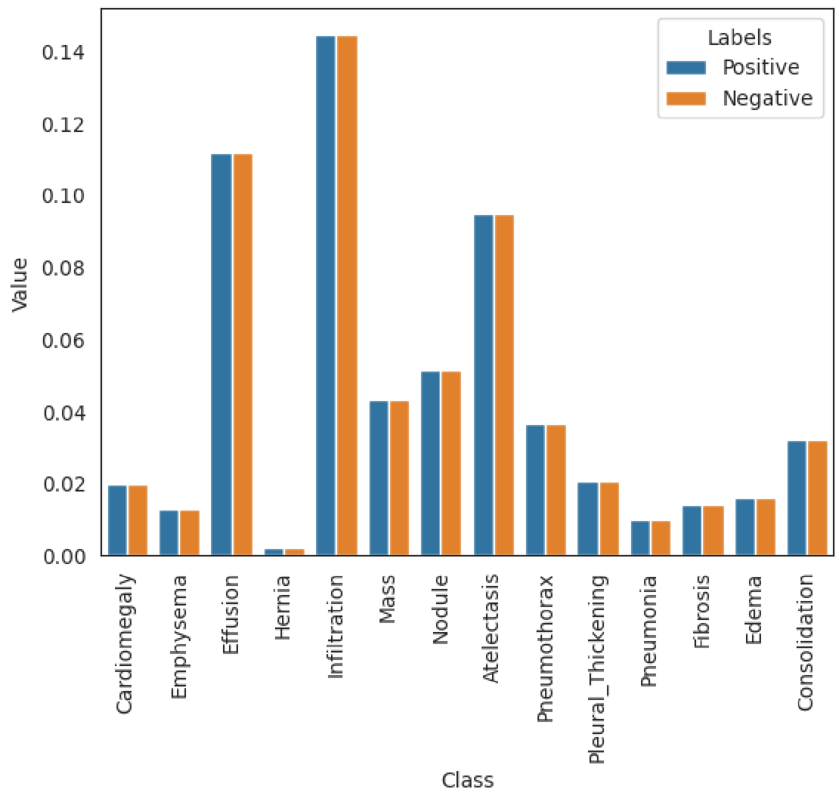
Frequency Distribution of Each Class:



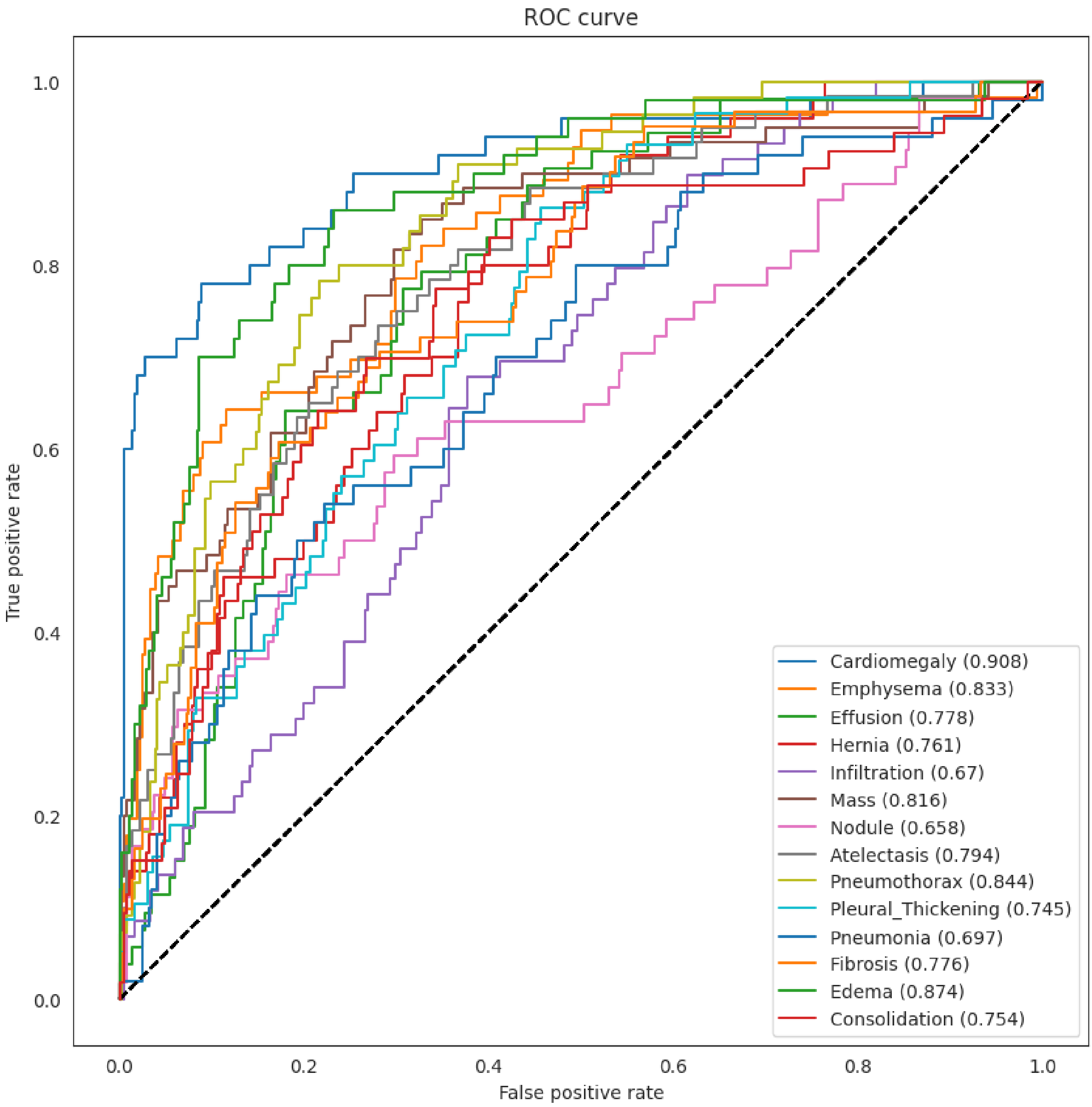
Frequency Distribution Positive and Negative Cases for Each Class:



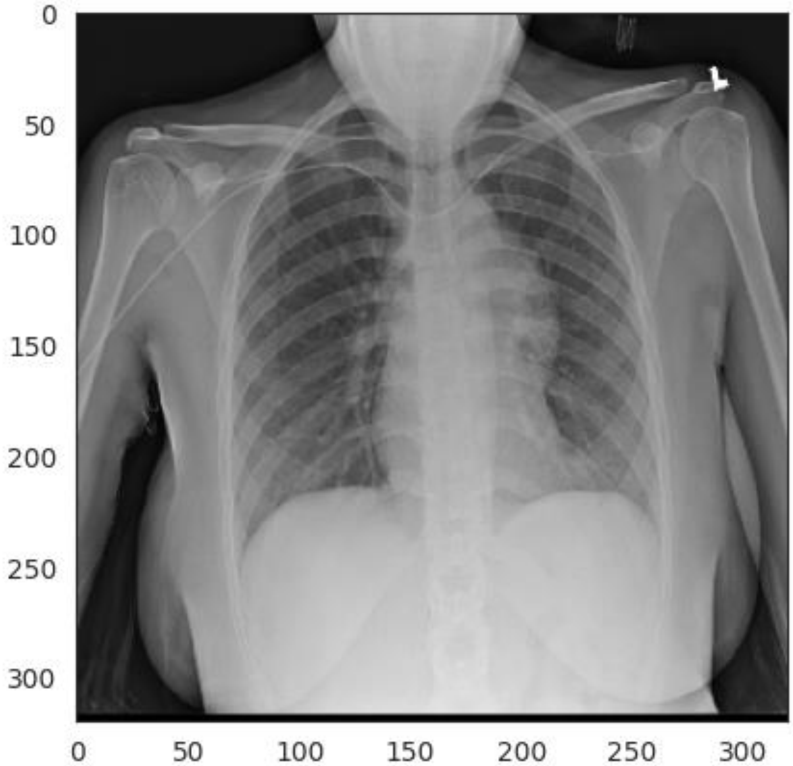
Frequency Distribution after balancing the contribution of positive and negative labels:



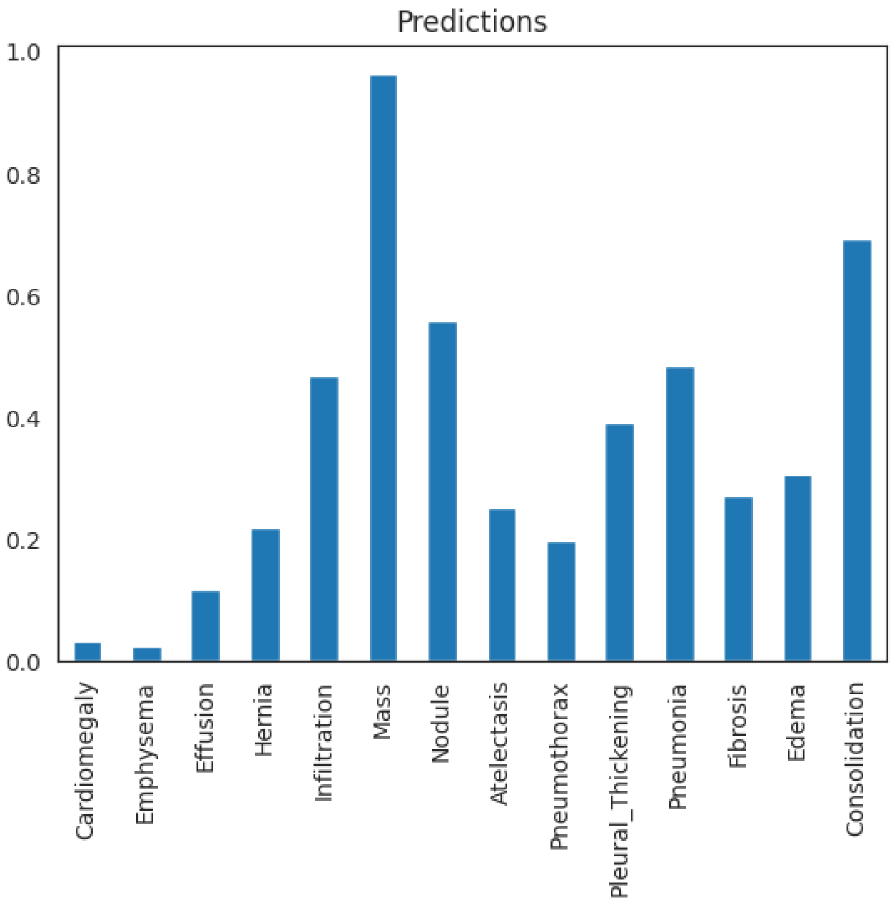
<u>ROC Curve:</u>


ROC curve
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
Cardiomegaly (0.908)
Emphysema (0.833)
Effusion (0.778)
Hernia (0.761)
Infiltration (0.67)
Mass (0.816)
Nodule (0.658)
Atelectasis (0.794)
Pneumothorax (0.844)
Pleural_Thickening (0.745)
Pneumonia (0.697)
Fibrosis (0.776)
Edema (0.874)
Consolidation (0.754)

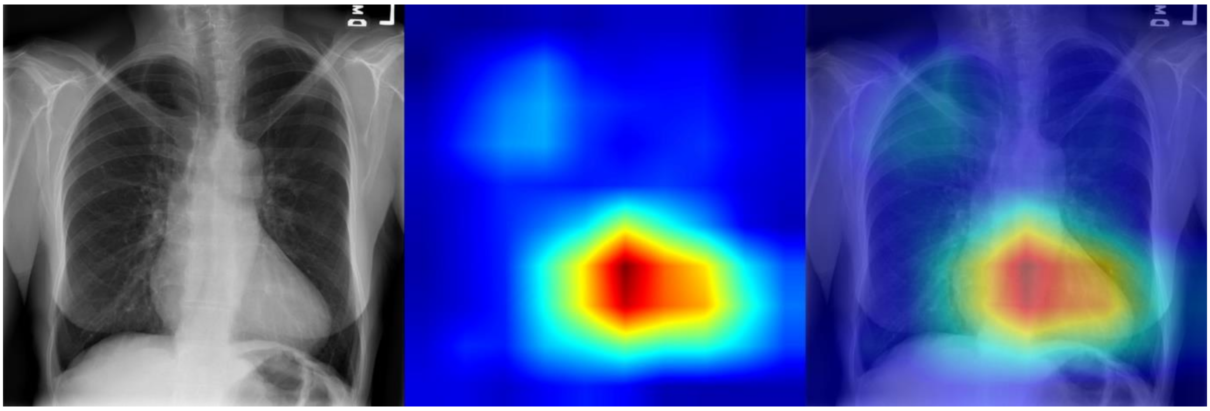
Input for prediction:



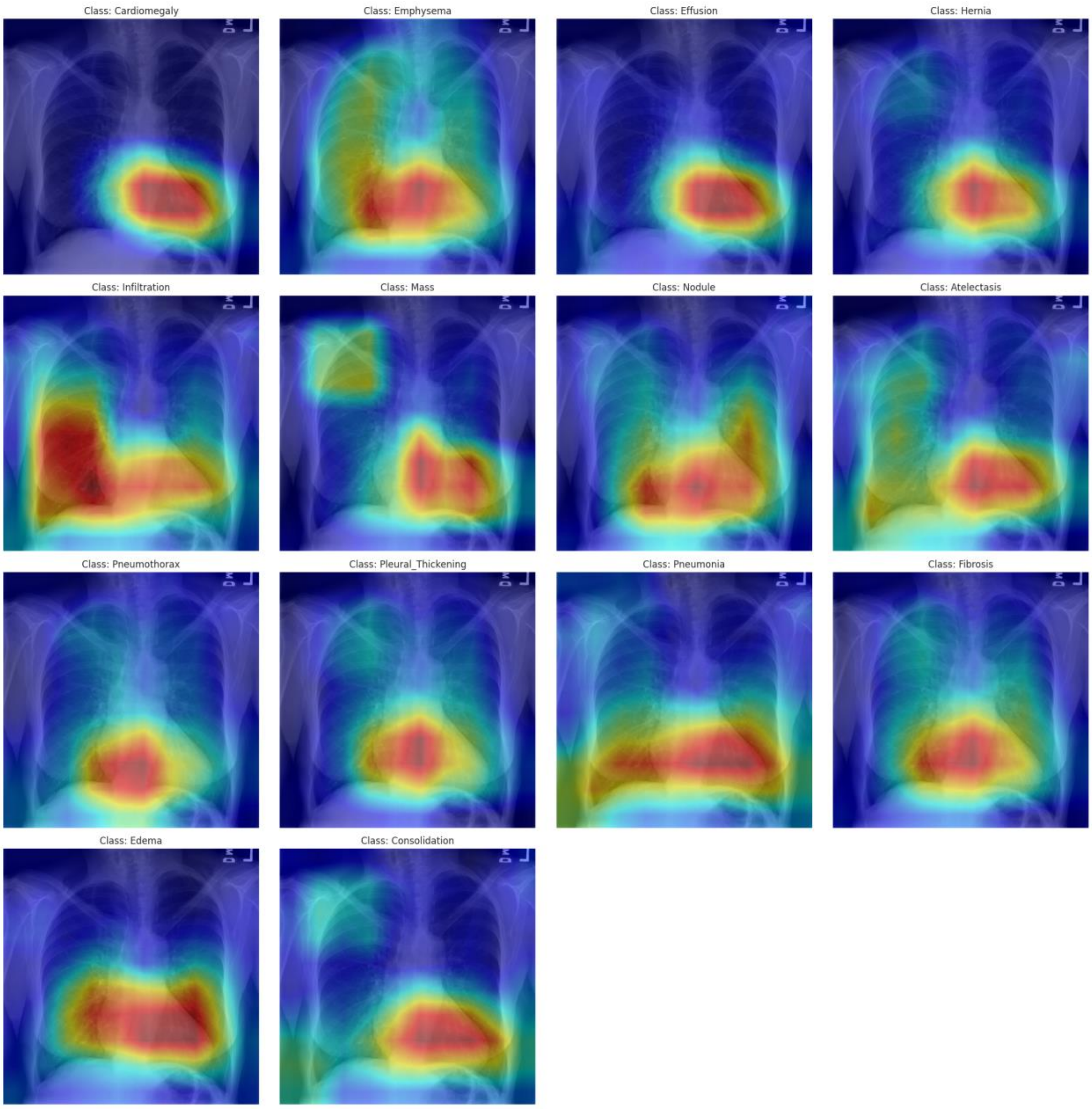
Obtained Prediction:



Grad-CAM for the class with Highest Prediction Probability:



Grad-CAM for all classes:



THE CONCLUSION

The completion of this Deep Learning project marks a significant milestone in my journey to develop a robust deep learning model. Throughout the development process, I encountered various challenges and opportunities for learning, ultimately contributing to the enhancement of our skills and understanding of deep learning.

The utilization of the DenseNet framework proved to be a pivotal choice, offering a structured and efficient way to build the model. Grad-CAM technique provided a clear visualization of how the model predicts the classes, facilitating better understanding of how the model works.

The Python Keras Library was a significant tool in the development process from preprocessing the input to building the model.

The development of this project has been a rewarding experience that not only showcases my technical capabilities but also highlights the importance of collaboration, adaptability, and continuous learning in the dynamic field of Data Science. I am proud of the final product and confident that it lays a solid foundation for future development endeavours.

In conclusion, the Chest X Ray Analysis project represents a successful fusion of technological innovation and medical expertise, demonstrating the potential of DenseNet in creating robust, scalable, and user-friendly deep learning model in the ever-evolving medical landscape. I believe that this project will prove to be helpful in the medical field for easy, early and accurate detection of lung diseases in patients especially in remote locations, where there are few or no radiologists accessible, the lack of radiologists is extremely severe, making it challenging for patients to get diagnostic imaging services.