MAKERERE UNIVERSITY

College of Computing & Information Science

Department of Computer Science

Course	Course Unit	Assignment 2
MSc in Computer Science	Deep Learning	Chest Opacities

Name: Sudi Murindanyi

Reg. No.: 2022/HD05/5583X, Student No.: 2200705583

Short Report on Chest Opacities: Assignment 2

Small description of the instructions

Required to create a deep learning-based chest opacity diagnosis based on the dataset provided. This project involves training, validating, and testing a Python-based deep learning model using the provided "known_images" dataset. The trained model has to be shared with the lecturer, along with the source code. Additionally, use the model to classify the images in the "unknown_images" directory and create a CSV file indicating the predicted labels. Finally, evaluate the model's performance on "dataset2" and present the test results to the lecturer.

Summary of the approach used

I primarily utilized Google Colab for my coding tasks due to its provision of free GPUs, which greatly expedited model training. Relying solely on my personal machine would have been time-consuming or even infeasible, considering the various experiments I wanted to conduct within a limited timeframe. Additionally, I employed the "os" and "shutil" libraries to efficiently read and organize image folders, ensuring smooth workflow in my working environment. For generating image datasets for training, validation, and testing, I opted to utilize the "tensorflow.keras.applications.resnet50" module. Although I experimented with multiple models, I ultimately chose to focus on ResNet50 due to its excellent performance. Furthermore, to assess my model's effectiveness and provide interpretability, I employed GradCAM, which offered valuable insights into both the model's predictions and its underlying reasoning.

A total of 715 images from the known_images dataset were used for training, validation, and testing purposes. The data was split into three sets: 70% (499 images) for training, 15% (106 images) for validation, and the remaining 15% (110 images) for testing. This partitioning ensured a balanced distribution of data across the different subsets, allowing for robust model training, evaluation, and performance assessment.

Description of the model trained

In this assignment, a ResNet50-based model was developed to address the task of image classification. The ResNet50 model, pretrained on the "imagenet" dataset, served as the foundation for my approach. By excluding the top classification layer and freezing all existing layers, I aimed to leverage the powerful feature extraction capabilities of ResNet50 while minimizing the computational complexity during training. To adapt the model to our specific classification problem, additional layers were added, including a fully connected layer with 512 units and ReLU activation. Dropout regularization with a rate of 0.5 was incorporated to mitigate overfitting. The final binary classification output was obtained using a dense layer with a sigmoid activation function. Overall, this model architecture aimed to strike a balance between leveraging the strengths of the pretrained ResNet50 and fine-tuning the network for optimal performance on my image classification task.

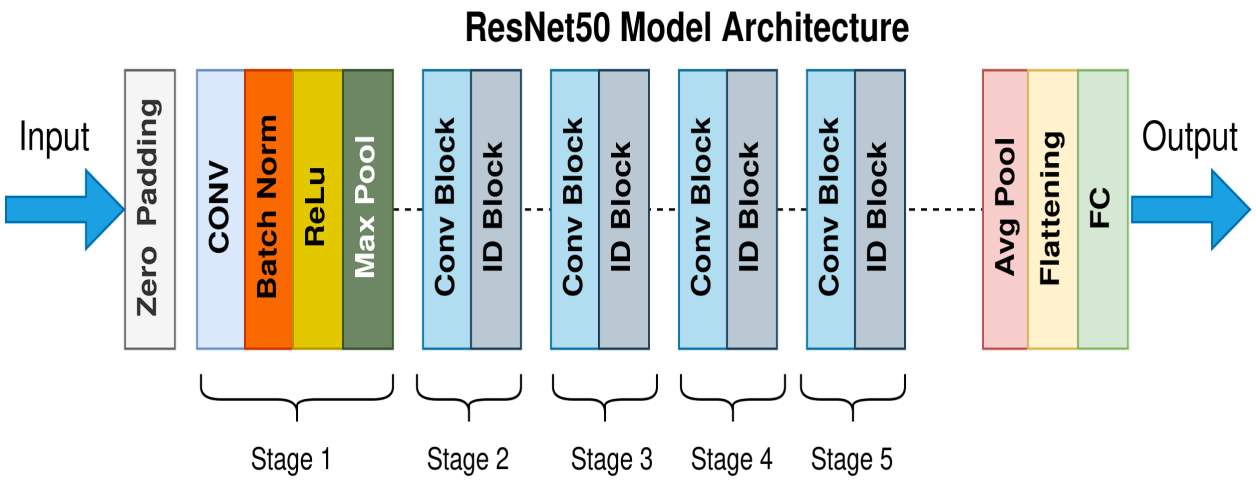


Figure 1: Image of ResNet50(From Online resource)

Model Results

Upon completion of the training process, the models successfully yielded the outcomes presented in Table 1, which is displayed below.

Table 1: Model Results After Training

Training Accuracy	Validation Accuracy
98%	97%
Loss: 0.0613	Loss: 0.1458

Figure 2 vividly depicts the gradual decline of the loss function throughout the training process, showcasing both the training and validation loss curves. Notably, the final losses were computed as 0.061 for the training set and 0.158 for the validation set, as highlighted in Table 1. Furthermore, Figure 3 provides insight into the training accuracy curve, revealing that the model achieved an impressive accuracy of 98% on the training set and 97% on the validation set. In summary, the model exhibited strong performance during the training phase, prompting the subsequent step of evaluating its effectiveness on the independent test set.

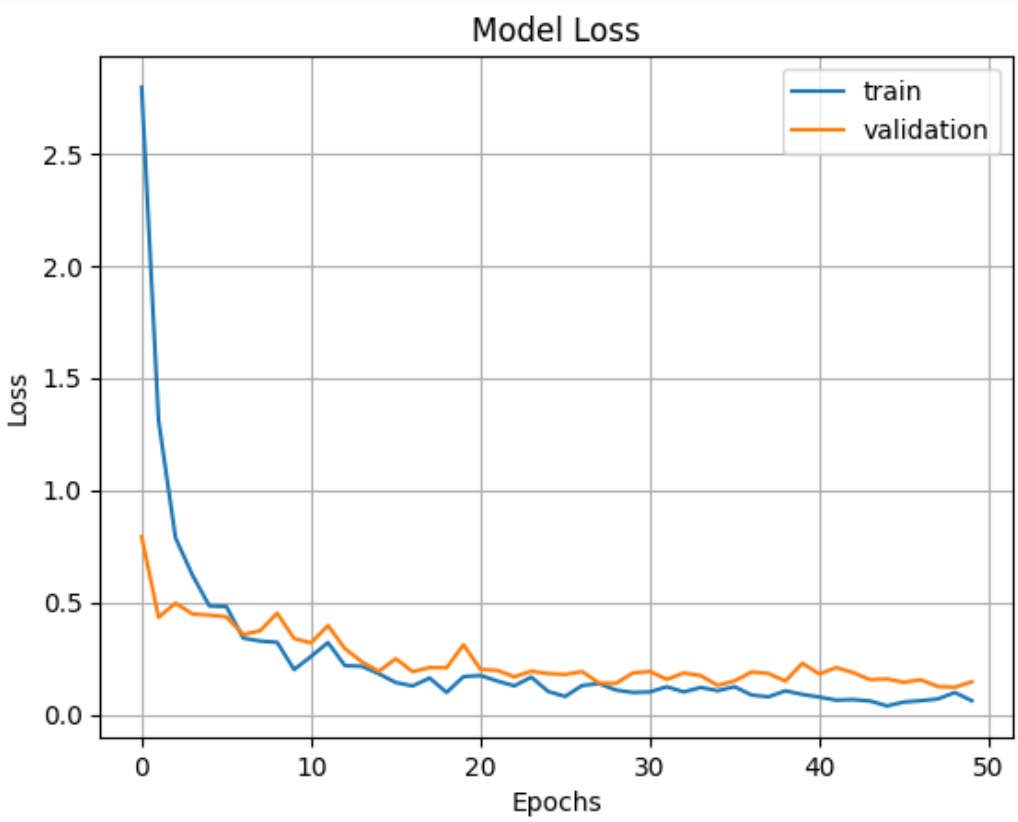


Figure 2: Model Loss curve in Training Process

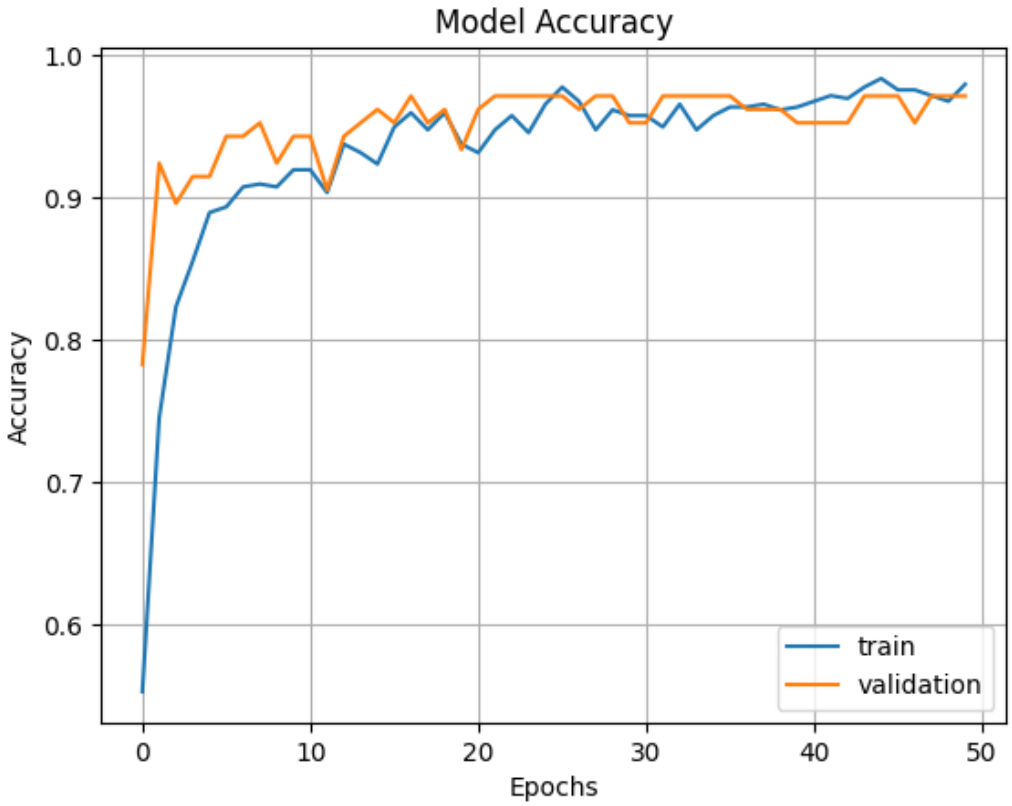


Figure 3: Model Accuracy Curve in Training Process

Model Evaluation

On the test set

Table 2: Model Results on test set

	precision	recall	f1-score	support
INFECTED	0.96	0.98	0.97	54
NORMAL	0.98	0.96	0.97	56
accuracy			0.97	110
macro avg	0.97	0.97	0.97	110
weighted avg	0.97	0.97	0.97	110

The model's exceptional performance extended to the test data, achieving an impressive accuracy of 97%. Additionally, the classification report presented in the above table highlights remarkable results, with weighted average precision, recall, and F1-score all measuring 97%. These findings underscore the model's robustness and effectiveness in accurately classifying the test samples.

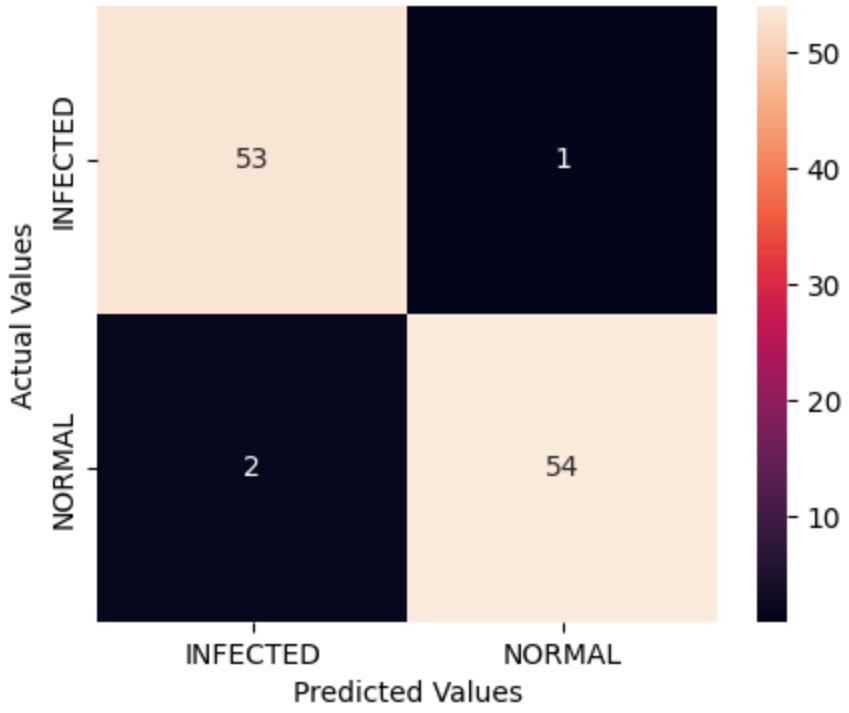


Figure 4: Confusion Matrix on Test Data

Figure 4 illustrates the confusion matrix generated when the model was applied to the test data. Among the 110 test images, a mere three images were misclassified, exemplifying the exceptional performance of the model. This outcome highlights the model's remarkable accuracy and reinforces its proficiency in accurately categorizing the majority of the test samples.

You can access the notebook for this experiment by clicking here.

Model Testing on Unknown data and Dataset 2

1. Unknown data

In compliance with the assignment instructions, our model was deployed and tested on the images located in the "unknown_images" folder within the dataset1 directory. Subsequently, a CSV file was generated, containing the model's predictions for each image. We present the results in the provided CSV file, demonstrating the outcome of this evaluation. Find the CSV here.

2. Dataset 2

Table 3: Classification report on Dataset 2

	precision	recall	f1-score	support
0	0.52	0.97	0.67	112
1	0.77	0.09	0.16	112
accuracy			0.53	224
macro avg	0.64	0.53	0.42	224
weighted avg	0.64	0.53	0.42	224

The classification report in Table 3 provides an overview of the model's performance on Dataset 2. Unfortunately, the model's performance was below expectations, with an accuracy of only 53%, a prediction rate of 64%, a recall rate of 53%, and an f1-score of 42%.

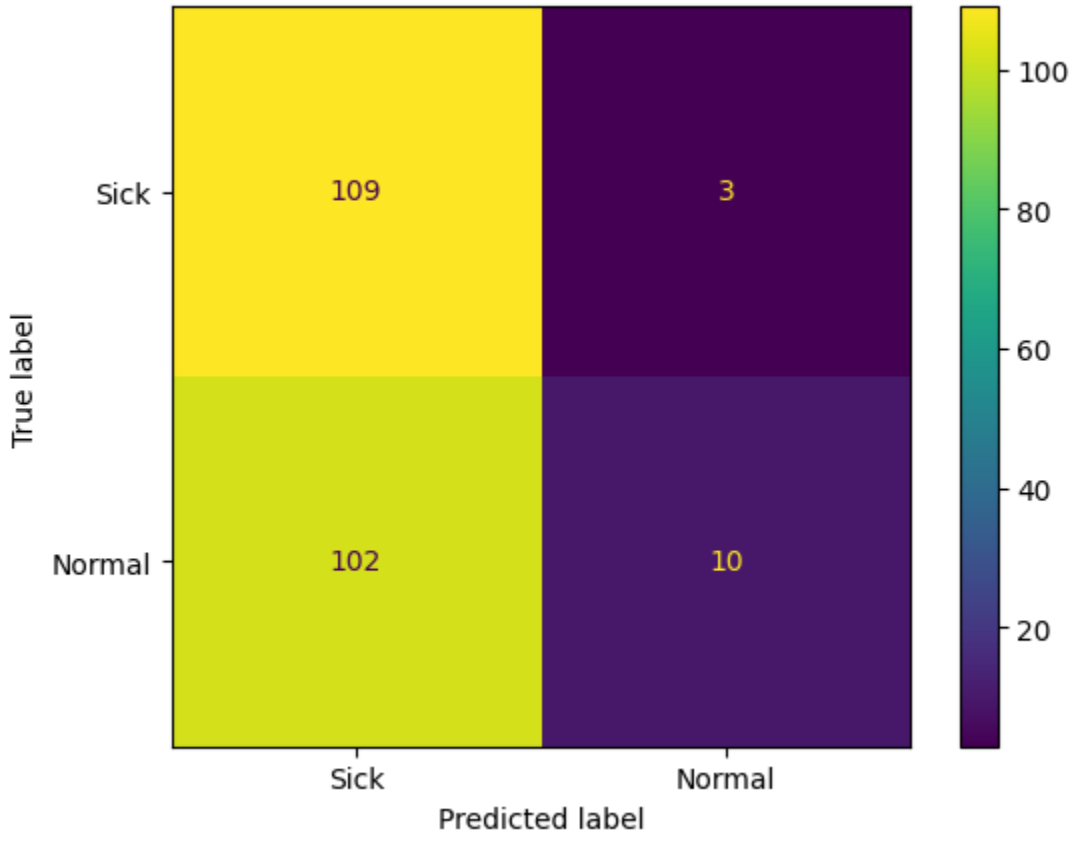


Figure 6: Confusion Matrix on Dataset 2

Since dataset 2 contained labeled data, we were able to utilize the model to make predictions on the images and generate a confusion matrix that compares these predictions with the provided labels. Some might argue that the model's performance is equivalent to random guessing due to its accuracy falling below 60%. However, this assessment is incorrect. In reality, the model demonstrated good performance in correctly classifying diseased images, as evident from Table 3, which displays a recall rate of 97% for sick images and 0.9% for normal images. This indicates that the model is effectively identifying most of the images as sick. Additionally, Figure 6, the confusion matrix, reveals that all but 13 images were classified as sick. You can find the notebook used for evaluating the model performance here.

In order to understand the reasoning behind these predictions, an Explainable Artificial Intelligence(XAI) technique was employed. To achieve this, the GRADCAM method was utilized as part of the explainable AI process. This technique provided insights into the underlying factors contributing to the model's predictions.

Explainability

You can find the notebook of the explainability here

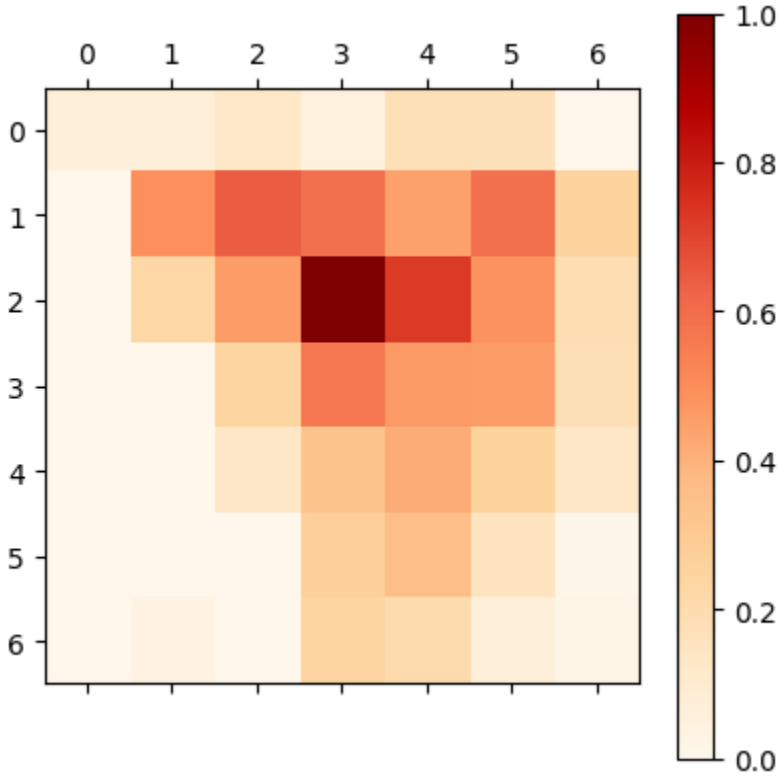


Figure 7: Heatmap

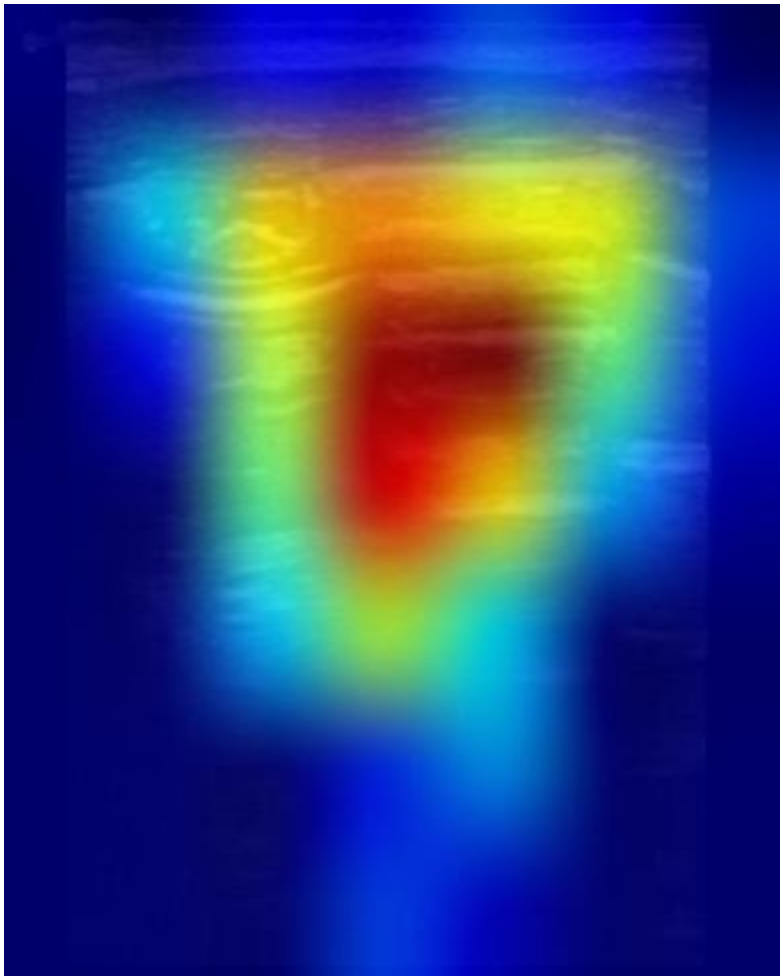
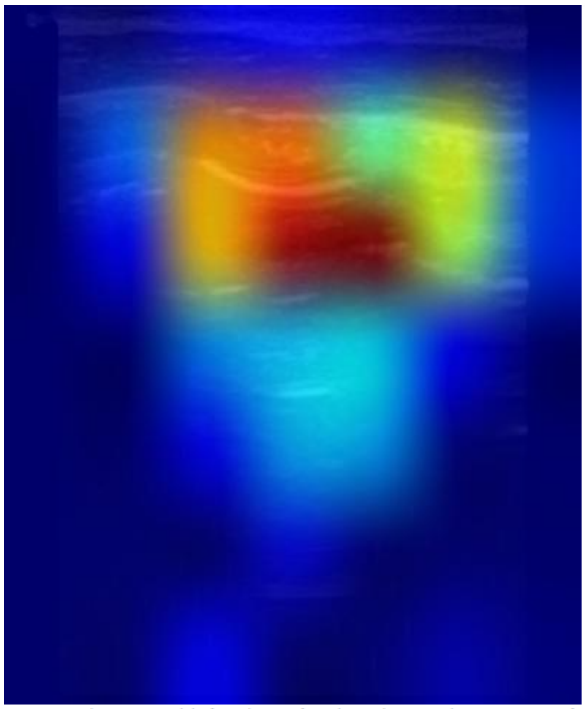


Figure 8: Grad-CAM image

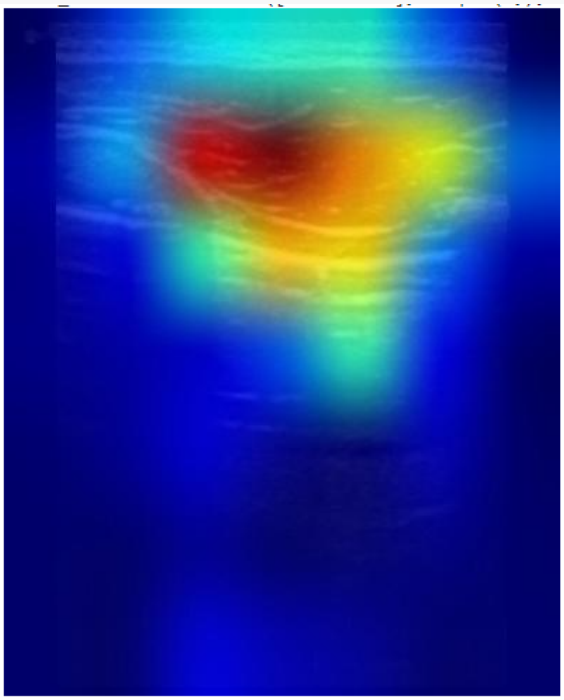
In my approach, I initially generated a heatmap, which was then linked to the corresponding image to produce the Grad-CAM heatmap. This Grad-CAM heatmap highlights the regions within the image that have the greatest impact on the network's prediction. By analyzing the gradients flowing into the final convolutional layer of the network, it indicates the importance of different regions. Specifically, in Figure 7 and Figure 8, the Grad-CAM results for image 35 in the unknown folder are displayed.

However, it was insufficient as I had not yet demonstrated what was happening with dataset 2. To address this, I applied the Grad-CAM technique to all the images in both the unknown folder and dataset 2. Presented below are a few sample results obtained from this analysis.

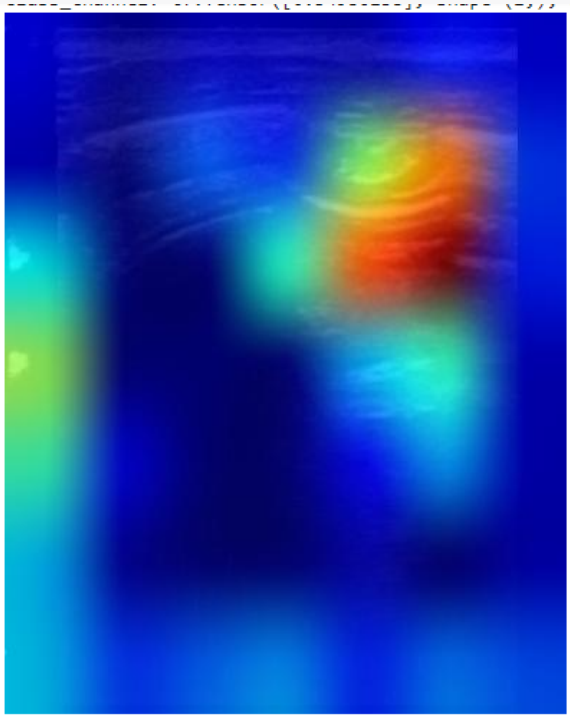
Unknown folder



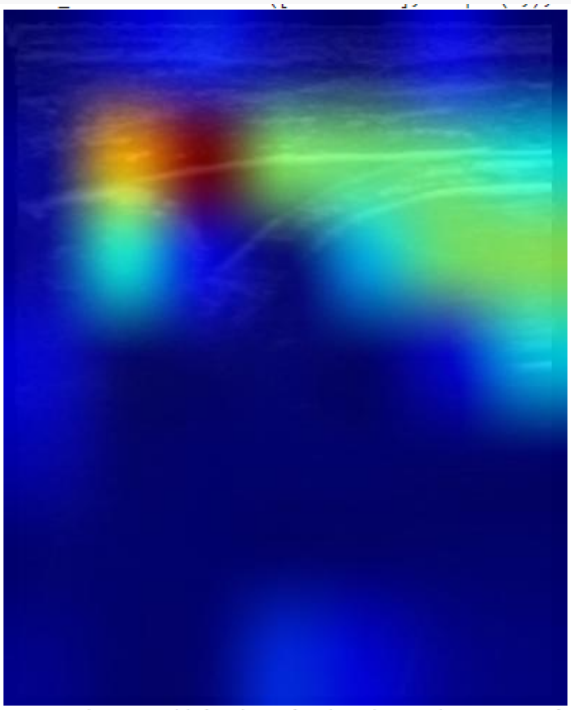
```
Image: /content/drive/MyDrive/MCS/Sem 2/Deep Learning/Deep_learning_assigmnent_2/dataset1/unknown_images/1.png
Prediction: Normal/ Not Sick
```



```
Image: /content/drive/MyDrive/MCS/Sem 2/Deep Learning/Deep_learning_assigmnent_2/dataset1/unknown_images/11.png
Prediction: Normal/ Not Sick
```



```
Image: /content/drive/MyDrive/MCS/Sem 2/Deep Learning/Deep_learning_assigmnent_2/dataset1/unknown_images/27.png
Prediction: Infected/ Sick
```



```
Image: /content/drive/MyDrive/MCS/Sem 2/Deep Learning/Deep_learning_assigmnent_2/dataset1/unknown_images/38.png
Prediction: Infected/ Sick
```

Dataset 2

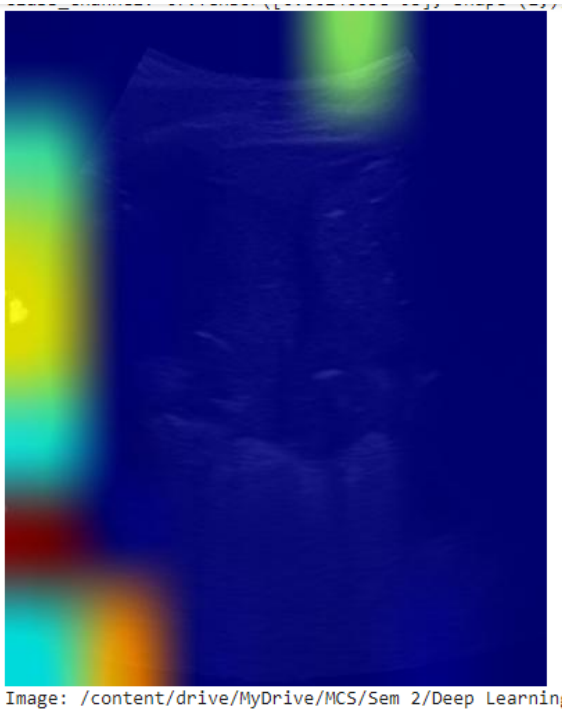


Image: /content/drive/MyDrive/MCS/Sem 2/Deep Learning/Deep_learning_assigmnent_2/dataset2/images/sick_eff11.png
Prediction: Infected/ Sick

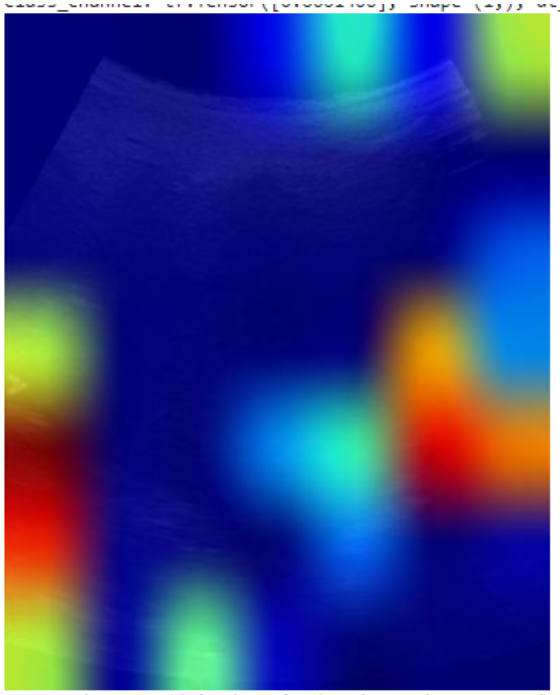


Image: /content/drive/MyDrive/MCS/Sem 2/Deep Learning/Deep_learning_assigmnent_2/dataset2/images/sick_eif98.png
Prediction: Infected/ Sick

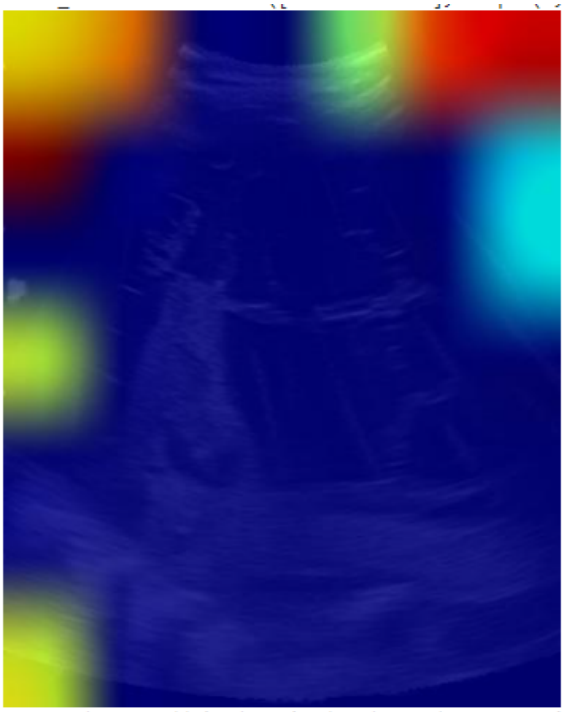


Image: /content/drive/MyDrive/MCS/Sem 2/Deep Learning/Deep_learning_assigmnent_2/dataset2/images/sick_eff85.png
Prediction: Infected/ Sick

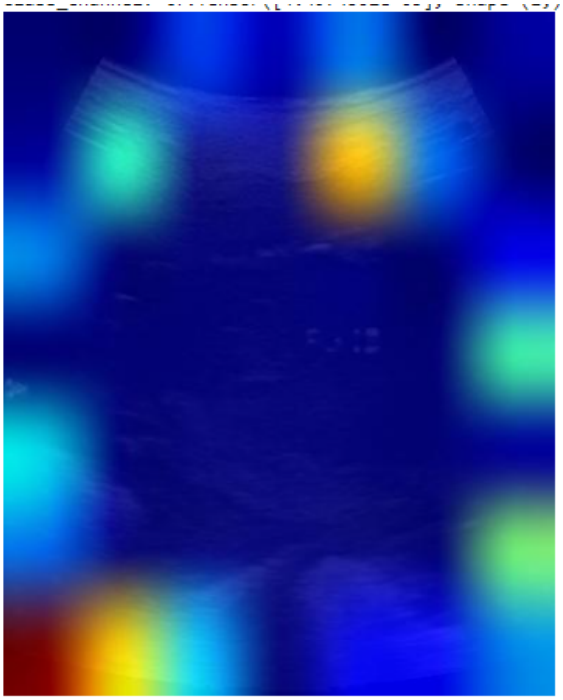


Image: /content/drive/MyDrive/MCS/Sem 2/Deep Learning/Deep_learning_assigmnent_2/dataset2/images/sick_eif102.png
Prediction: Infected/ Sick

Upon examining the results obtained from Grad-CAM, as a non-radiologist, it is challenging for me to interpret the specific features highlighted by Grad-CAM. However, I can extract two key insights from these findings. Firstly, the model's performance seems to be better than initially perceived based on the accuracy of dataset 2. The Grad-CAM analysis demonstrates that the model is capable of accurately capturing and highlighting relevant features in the unknown data, indicating its effectiveness in making predictions.

Secondly, the poor performance on dataset 2 can be attributed to the substantial differences between the images in dataset 2 and those used for training. This suggests the need for special preprocessing techniques to ensure compatibility between the testing images and the model trained on images from the known folder. These insights emphasize the importance of adapting the training and testing data to enhance the model's performance and generalize well to unseen data.

The Grad-CAM results reveal that the model's focus seems to be on random aspects that are not relevant to the intended detection and classification task. This observation is consistent across all the images in dataset 2 as well as the images in my notebook. It suggests that the model may be emphasizing irrelevant features instead of the important ones required for accurate classification. The main reason is different kinds of images, not specifically the model.

Next steps

I am genuinely enthusiastic about the project and these valuable insights provided by the Grad-CAM analysis have motivated me to strive for better results. My next step is to apply a special preprocessing technique to align dataset 2 with dataset 1, with the expectation of achieving improved performance. Additionally, I plan to implement LIME as another XAI technique to explore its explanatory outcomes.

I have a sense of confidence that dedicating more time and effort to this project will yield better results. Even after my exams, I intend to continue training the model, hoping for significant progress. I eagerly anticipate sharing the outcomes as soon as they become available.

Here are all my notebooks:

Model training: Here

Model Evaluations: Here

Model Explainability: Here

Results for unknown data:

Thank you