Cairo University
Faculty of Engineering
Department of Computer Engineering

X-Reporto

A Graduation Project Report Submitted

to

Faculty of Engineering, Cairo University

in Partial Fulfillment of the requirements of the degree

of

Bachelor of Science in Computer Engineering.

Presented by

Basma Elhoseny

Supervised by

Dr. Yahia Zakaria

July 2024

Abstract

This project addresses the increasing burden on radiologists due to the rising prevalence of chest diseases, which results in long queues of X-ray reports needing diagnosis. The objective of the project is to develop a semi-automated reporting system that supports radiologists by generating preliminary reports for each anatomical region and identifying diseases in those regions. Our approach involves enhancing chest X-ray images to correct defects caused by the X-ray devices, ensuring that critical diagnostic information is preserved and easily identifiable. The tool generates comprehensive, template-based reports tailored to the specific diseases identified, thereby streamlining the reporting process and reducing the time required for diagnosis.

The primary outputs of the project include the enhanced X-ray images and the detailed, disease-specific reports generated by the tool. Development and testing of the tool were conducted to ensure accuracy and reliability. Testing results demonstrate significant improvements in image clarity and diagnostic accuracy, as well as a reduction in the time required for report generation.

This project successfully developed and implemented the tool. The outcomes of this project not only alleviate the workload of radiologists but also contribute to better patient care by enabling faster and more precise medical interventions, ultimately increasing patient survival rates.

This project is sponsored by Voyance Health [1], a software development company in the medical field, which has provided invaluable support by offering server resources for training models on our large dataset. They also contributed the basic idea of the project and have been closely following our progress, offering guidance and feedback throughout the development process.

[1] https://voyance.health/

الملخص

هذا المشروع يعالج العبء المتزايد على أطباء الأشعة بسبب الانتشار المتزايد لأمراض الصدر، مما يؤدي إلى طوابير طويلة من تقارير الأشعة السينية التي تحتاج إلى تشخيص. هدف المشروع هو تطوير نظام تقارير شبه آلي يدعم أطباء الأشعة من خلال إنشاء تقارير أولية لكل منطقة تشريحية وتحديد الأمراض في تلك المناطق. نهجنا يتضمن تحسين صور الأشعة السينية لتصحيح العيوب التي تسببها أجهزة الأشعة السينية، وضمان الحفاظ على المعلومات التشخيصية الحيوية وجعلها سهلة التعرف. يقوم الأداة بإنشاء تقارير شاملة تعتمد على قوالب مخصصة للأمراض المحددة، مما يسهل عملية إعداد التقارير ويقلل من الوقت المطلوب للتشخيص.

تشمل النتائج الرئيسية للمشروع الصور المحسنة للأشعة السينية والتقارير التفصيلية الخاصة بالأمراض التي تنتجها الأداة. تم إجراء تطوير واختبار الأداة لضمان الدقة والموثوقية. تُظهر نتائج الاختبار تحسينات كبيرة في وضوح الصور ودقة التشخيص، وكذلك تقليل الوقت المطلوب لإعداد التقارير.

نجح هذا المشروع في تطوير وتنفيذ الأداة. نتائج هذا المشروع لا تخفف فقط من عبء العمل على أطباء الأشعة، بل تساهم أيضًا في تحسين رعاية المرضى من خلال تمكين التدخلات الطبية السريعة والدقيقة، مما يزيد في نهاية المطاف من معدلات بقاء المرضى على قيد الحياة.

هذا المشروع مدعوم من شركة Voyance Health، والتي وهي شركة تطوير برامج مرتبطة بالمجال الطبي، قدمت دعمًا لا يقدر بثمن من خلال توفير موارد الخادم لتدريب النماذج على مجموعة بياناتنا الكبيرة. كما ساهموا بالفكرة الأساسية للمشروع وتابعوا تقدمنا عن كثب، مقدمين التوجيه والملاحظات طوال عملية التطوير

ACKNOWLEDGMENT

We extend our sincerest gratitude and appreciation to our esteemed supervisor, Dr. Yehia Zakaria, for his dedicated supervision, invaluable guidance, and unwavering support throughout our project. His mentorship has been instrumental in our journey.

We also wish to thank Voyance Health for their sponsorship and provision of servers, which were essential for training our models.

Special thanks go to Eng. Mohamed Shawky and Eng. Omar Samir, technical assistants at the Faculty of Engineering, Cairo University, for their continuous support and assistance.

Additionally, we express our heartfelt appreciation to our department for providing us with the necessary courses and resources to undertake this project. Their support and encouragement have been pivotal.

Thank you all for making this journey possible and for your unwavering belief in our capabilities.

Table of Contents

Contents

List of Figures

List of Tables

Team Members

Name	Email	Phone Number
Basma Elhoseny	basma.elhoseny01@eng-st.cu.edu.eg	+2 01152668680

Supervisor

Name	Email	Number
Dr. Yahia Zakria	Yahiazakaria13@gmail.com	+2 0111 729 7303

Chapter 1: Introduction

X-Reporto is a specialized tool designed to alleviate the challenges faced by radiologists in managing an increasing number of patients and optimizing the medical reporting process for chest X-ray images. It enhances chest X-ray images affected by device defects, ensuring clearer and more accurate diagnostic images. Additionally, X-Reporto automates the generation of detailed reports based on chest X-ray findings, using templates that streamline the reporting workflow. By analyzing the images, X-Reporto provides insights into potential diseases and pinpoints their locations within the chest. This functionality saves valuable time for radiologists, enhances diagnostic accuracy, and ultimately improves patient care and treatment outcomes in medical settings

1.1. Project Outcomes

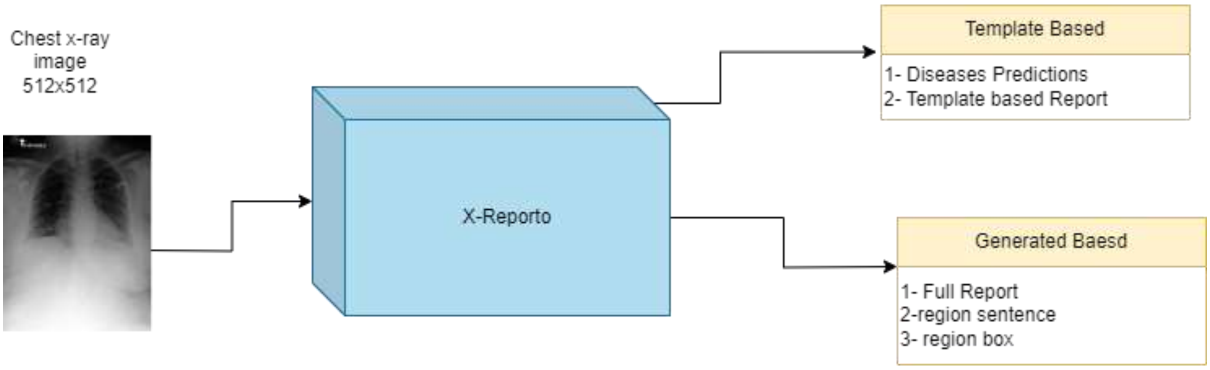


Figure1 X-Reporto Outcomes

Our project outcomes focus on developing a tool that enhances the efficiency and accuracy of chest X-ray diagnostics. This tool generates detailed reports for each anatomical region and identifies specific diseases, improving X-ray image quality by correcting device defects. It provides radiologists with template-based reports tailored to the identified diseases, streamlining the reporting process and reducing diagnosis time. These advancements help radiologists manage their workload more effectively and contribute to better patient care through faster and more precise medical interventions.

Chapter 3: Literature Survey

3.1 Class Activation Mapping (CAM)

Zhou B et al [1] proposed a method to explain decisions made by neural networks (NN) for classification tasks. CNNs have been shown to function as effective object detectors, allowing for region localization without requiring prior network training on localization tasks. However, this capability diminishes upon adding fully connected (FC) layers. To address this, Zhou B et al [1] introduced Global Average Pooling (GAP), which performs average pooling across the output of the backbone layers. By using weights in the FC layer, we can then pinpoint the regions of activation that underpin the classification decisions made by the network.

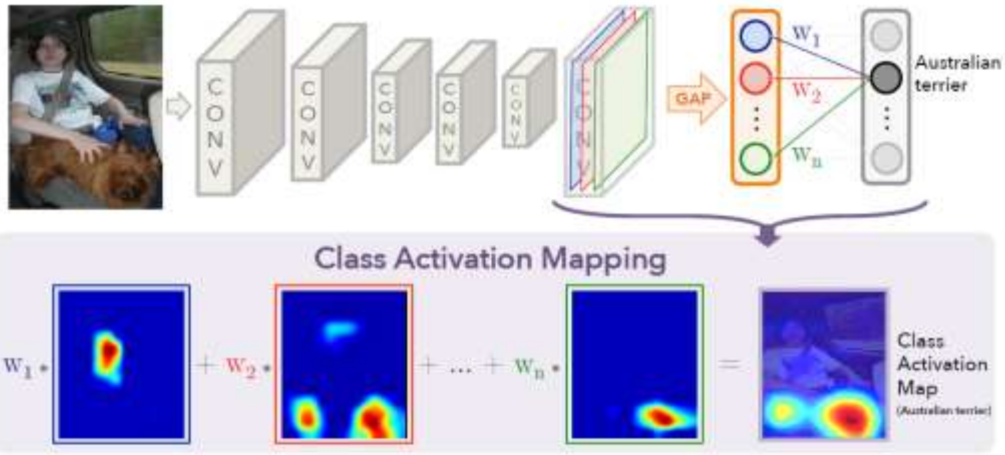


Figure 2 Class Activation Mapping describing activation regions for label prediction

The activation Maps are computed using the weights of the FC layer and the Final Transition Layer in the CNN.

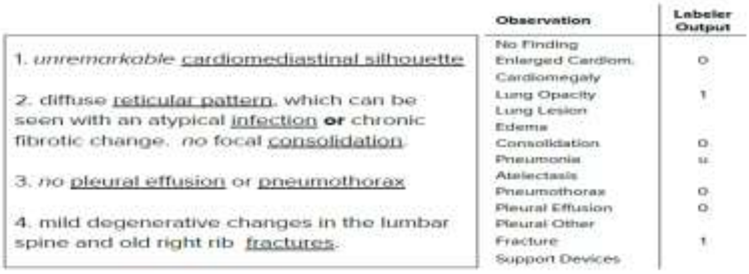
$$
\begin{aligned}
S_c &= \sum_k w_k^c \sum_{x,y} f_k(x,y) \\
&= \sum_{x,y} \sum_k w_k^c f_k(x,y).
\end{aligned}
$$

Chapter 4: Datasets Literature Survey

4.1. Labels for MIMIC-CXR

As mentioned in section 4.2, we gained access to the MIMIC-CXR dataset, which contains DICOM format images accompanied by free-text radiology reports. However, for our template-based report generation detailed in section x.x, we require labeled findings from these X-rays. These labels were sourced from external resources, specifically extracted using CheXpert, an open-source rule-based tool built on NegBio. The labels correspond to the following 14 most prevalent observations in chest diseases:No Finding - Enlarged Cardiom - Cardiomegaly - Lung Lesion - Lung Opacity - Edema - Consolidation - Pneumonia - Atelectasis - Pneumothorax - Pleural Effusion - Pleural Other - Fracture - Support Devices

Each label may have one of the following values indicating the presence and certainty of the finding in the report: 0 for negative presence, 1 for positive presence, -1 for uncertain polarity, and NaN for not mentioned or failed to parse.



1. *unremarkable* cardiomediastinal silhouette

2. diffuse reticular pattern, which can be seen with an atypical infection **or** chronic fibrotic change. *no* focal consolidation.

3. *no* pleural effusion or pneumothorax

4. mild degenerative changes in the lumbar spine and old right rib fractures.

Observation	Labeler Output
No Finding	
Enlarged Cardiom.	0
Cardiomegaly	
Lung Opacity	1
Lung Lesion	
Edema	
Consolidation	0
Pneumonia	u
Atelectasis	
Pneumothorax	0
Pleural Effusion	0
Pleural Other	
Fracture	1
Support Devices	

Figure 3 Output example for CheXpert labeler on a free-text Report

Chapter 5: System Design and Architecture

5.1. System Architecture

The architecture of the system is structured around a modular design that enhances scalability and maintainability. The overall system is represented as a block diagram that shows interactions between the different modules and their respective functions.

As we have 6 separated modules, each module has specific functionality and expects certain modules output and generates correct output for next modules. Follow of our modules is that

1. We first should remove any noise in the x-ray chest.
2. We detect anatomical regions in chest and generate visual features for them
3. We Select some regions that have findings in it.
4. From these visual features we generate corresponding findings in it including abnormality and diagnosis.
5. Finally, we generate predictions for possible diseases and its location in x-ray.

5.1.2. Block Diagram

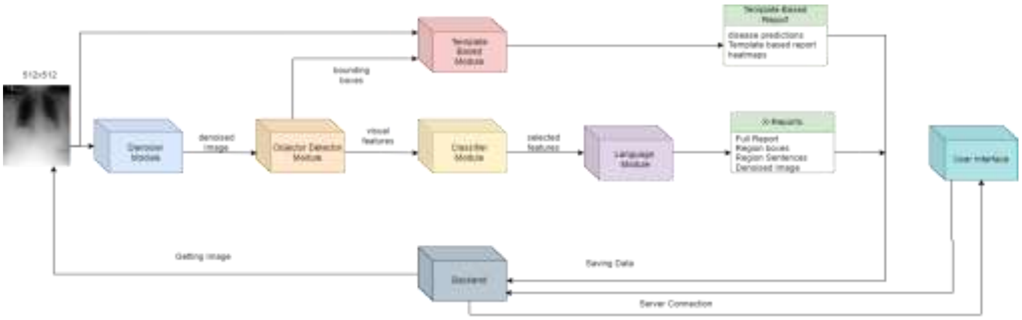


Figure 4 X-Reporto System Block Diagram

5.2. Classifiers

5.2.1. Functional Description

We have two types of classifiers, each serving a specific function:

- **Region Selection Classifier:** This classifier's primary role is to determine which of the 29 anatomical regions should be included in the report. It identifies and selects relevant regions based on the input data.
- **Abnormal Classifier:** Serving as an intermediate module, the Abnormal Classifier enhances visual features extracted by the object detector. Although

lacking definitive ground truth for abnormalities, this model is instrumental in training and fine-tuning the object detector to enhance its ability to detect abnormal visual features.

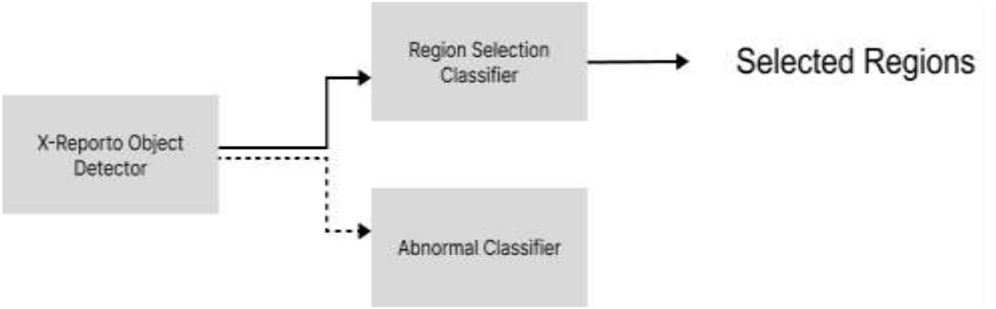
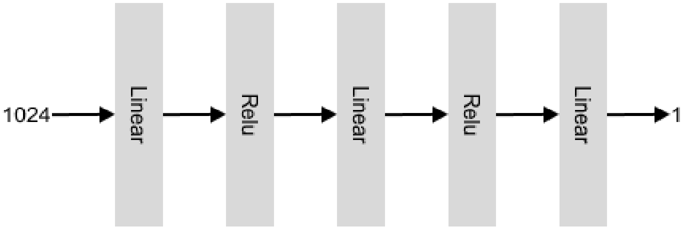


Figure 5 Classifier in X-Reporto System

5.2.2. Modular Decomposition

Both classifiers share a similar architecture tailored for their respective tasks. They process visual features extracted for the 29 anatomical regions by the object detector. The architecture includes three linear layers, each followed by a ReLU activation function.



Binary Classifier

Figure 6 Classifier Architecture

5.2.3. Design Constraints

Due to dataset imbalance in both region selection and abnormal labels, a weighted loss function (Binary Cross Entropy) was employed. The weights for the loss function were determined through statistical analysis of the training dataset.

5.2.4. Methodologies

5.2.4.1. Training

Training of both classifiers, Region Selection and Abnormal, occurred simultaneously with separate sets of weights. Initially, the object detector was frozen for several epochs while updating only the classifier weights. Later, the object detector was unfrozen with a small learning rate for additional epochs. This approach allowed the Abnormal Classifier to update the object detector's weights through backpropagation, while the Region Selection Classifier updated its weights independently.

5.3. Template Based Report Generator

5.3.1. Functional Description

The template-based report generator Module main functionality is to take the chest x-ray as input and generate a detailed, template-based report that summarizes findings in the x-ray. Advanced features are incorporated into these reports to provide more insights to radiologists, making the reports not only comprehensive but also informative. Additionally, the module outputs heatmaps that assist in reasoning and explainable classification by localizing areas of interest for each finding, thereby enhancing the radiologist's ability to interpret the results accurately.

5.3.2. Modular Decomposition

The Template-based report generator is mainly divided into 3 sub-modules discussed in detail in the following subsections.

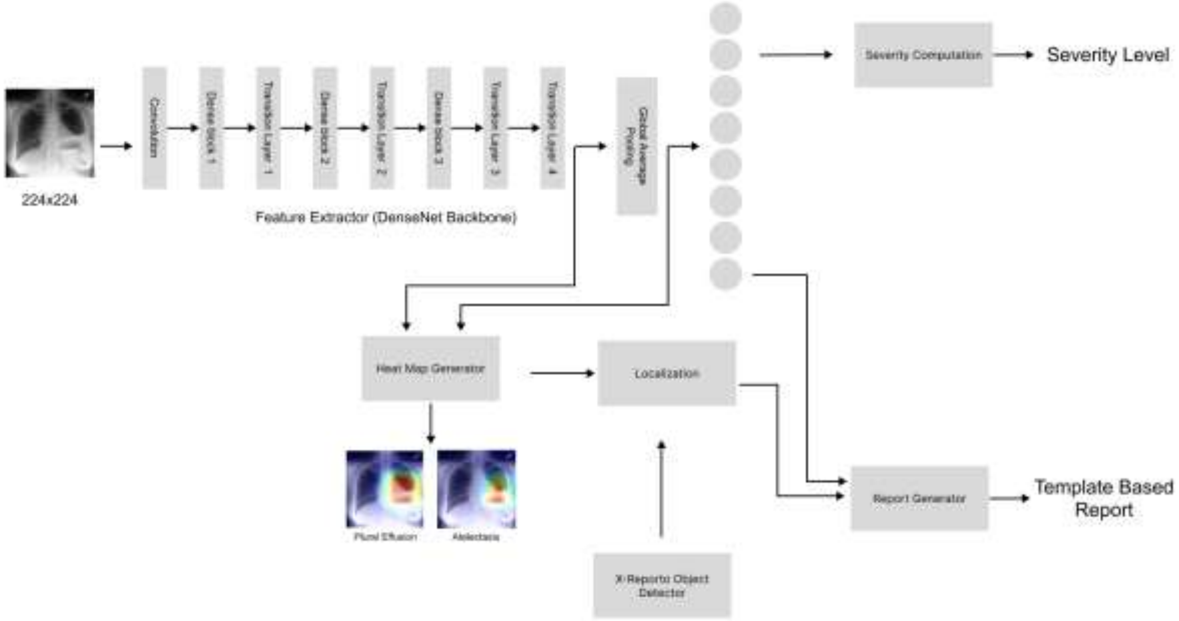


Figure 7 Template-based Report Generator Block Diagram

5.3.2.1. Findings Classifier

The submodule's primary function is to perform multilabel classification on chest X-ray images, targeting eight specific findings derived from the dataset. It employs DenseNet-121, chosen for its effective feature extraction capabilities over other popular architectures like AlexNet, GoogleNet, VGGNet, and ResNet. DenseNet-121 utilizes both forward and backward skip connections to mitigate the vanishing gradient problem, facilitating enhanced information propagation across all layers.

The output from DenseNet-121's backbone is a 7x7x1024 feature map. To accurately localize activations within this feature map, Global Average Pooling (GAP) is adopted. Unlike Global Max Pooling (GMP), which limits activation localization to specific points within object boundaries, GAP provides a comprehensive spatial understanding of identified findings' extents.

The GAP layer outputs a flattened vector of size 1024, serving as input to a final fully connected (FC) layer comprising eight neurons—one for each finding. Sigmoid activation is applied to these neuron outputs to derive probabilities for each finding's presence in the chest X-ray image.

This submodule leverages state-of-the-art CNN architecture and pooling techniques to achieve accurate multilabel classification of chest X-ray images, enhancing diagnostic capabilities by providing probabilities for multiple findings simultaneously.

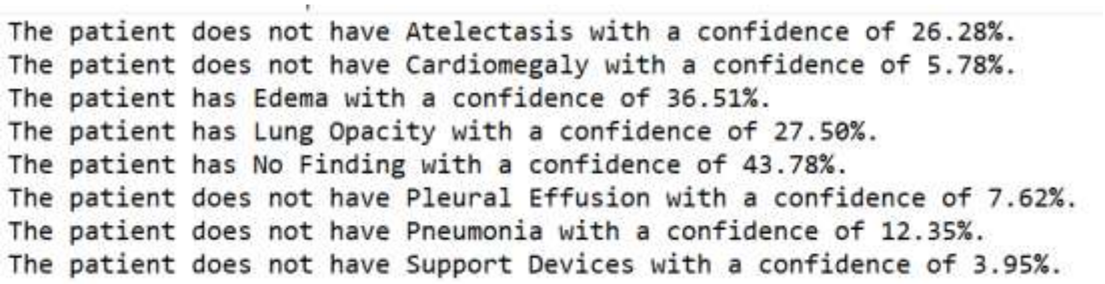
5.3.2.2. Heatmap Generator

For each of the 8 classification labels, feature maps are generated independently using DenseNet-121. Weights from the final FC layer are multiplied with activations from the last dense block, following a methodology like [3], albeit without their transition layer. Applying the Relu function ensures only positive values contribute to the heatmap. The final FC layer, with sigmoid activation, normalizes feature map values to 0-1 range for consistent scaling and enhanced contrast. Feature maps are initially 7x7 and resized during inference to 224x224 pixels, preserving aspect ratio by padding the smaller dimension. Using OpenCV's Colormap function, higher values correspond to warm colors (like red, yellow), while lower values to cool colors (like blue, green) in the heatmap. The heatmap is then overlaid onto the original image via a weighted sum formula.

$$blended\_image = image\_resized \times 1 + heatmap\_resized \times 0.35 + 0$$

5.3.2.3. Report Generator

Our Main Focus was to not just classify the image according to the class we previously defined, we needed to produce a template-based report. Our first version of the report generator, as shown in figure 5.9.2.3.1, was just taking the output of the sigmoid functions. According to predefined thresholds obtained during model training we state whether the patient has this finding or not. But we need to give insight to the radiologist and he has the final word so it was essential to add the model confidence to the report so that we don't mislead the radiologist.This info was very essential because we already has imbalanced data so choosing the optimal threshold based on the validation data set will make the model unable to generalize.



```
The patient does not have Atelectasis with a confidence of 26.28%.
The patient does not have Cardiomegaly with a confidence of 5.78%.
The patient has Edema with a confidence of 36.51%.
The patient has Lung Opacity with a confidence of 27.50%.
The patient has No Finding with a confidence of 43.78%.
The patient does not have Pleural Effusion with a confidence of 7.62%.
The patient does not have Pneumonia with a confidence of 12.35%.
The patient does not have Support Devices with a confidence of 3.95%.
```

Figure 8 First version of the template-based report with findings confidence

Moreover, we need to make our report informative, we need not to just give the doctor the probability of a finding, instead we need to help him with the location of the findings in case the heatmap isn't available. So, we used the bounding boxes generated by the X-Report object-detector. Our product is as follows we have only the resized 224x224 feature map without being blended which is in RGB so we compute the mean over the 3 channel with equal weights.

Then we take as input the Bounding boxes from the object detector, but these measurements are relative to the 512x512 image output from the denoiser explained in before, so we need to rescale them to be relative to the 224x224 heat map we have.

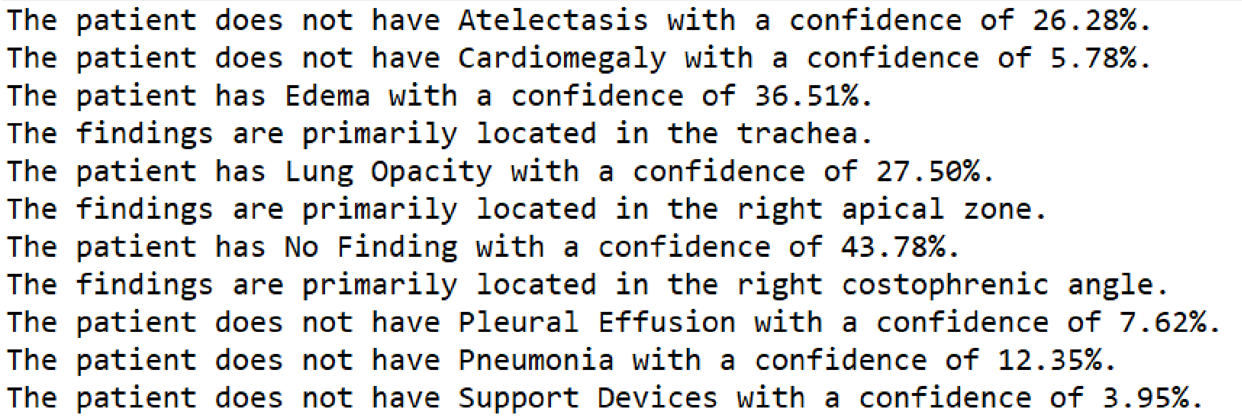
Using the coordinates of the fixed bounding boxes we get mean of activations for each region and choose the one with the max activation, but following this procedure we overcome the problem of overlapped regions in case we just check where pixels are inside region boundary

To enhance the informativeness of our reports, we go beyond providing the probability of findings to assist doctors in pinpointing their locations, especially when heatmaps are unavailable. We achieve this using bounding boxes generated by the X-Report object detector, and we got our final template-based report as illustrated in the following Our procedure is as follows:

We start with the resized 224x224 RGB feature map, without blending, and compute the mean activation over the three channels equally weighted. The bounding boxes provided by the object detector are relative to the 512x512 image output from the denoiser (explained in section 5.1). Thus, we rescale these coordinates to be relative to our 224x224 heatmap, expressed by the equation:

$$box_{relative\ to\ 224} = box_{relative\ to\ 512} * \frac{224}{512}$$

From the fixed bounding box coordinates, we compute the mean activations for each region and identify the region with the maximum activation. This method initially involved checking the most active pixel in the heatmap and determining its corresponding boundary. However, this approach proved inadequate due to potential overlaps between regions.



```
The patient does not have Atelectasis with a confidence of 26.28%.
The patient does not have Cardiomegaly with a confidence of 5.78%.
The patient has Edema with a confidence of 36.51%.
The findings are primarily located in the trachea.
The patient has Lung Opacity with a confidence of 27.50%.
The findings are primarily located in the right apical zone.
The patient has No Finding with a confidence of 43.78%.
The findings are primarily located in the right costophrenic angle.
The patient does not have Pleural Effusion with a confidence of 7.62%.
The patient does not have Pneumonia with a confidence of 12.35%.
The patient does not have Support Devices with a confidence of 3.95%.
```

Figure 9 Second version of the template-based report with findings confidence

5.3.3. Design Constraints

5.3.3.1. Network Design

We opted to use DenseNet-121 as our backbone feature extractor instead of ResNet-50, which we found less effective due to the vanishing gradient problem. With our training relying heavily on gradient descent optimization, the inability of gradients to propagate effectively was hindering our progress, resulting in stock losses. DenseNet-121's bidirectional skip connections were pivotal for addressing our needs, particularly in handling images with multiple findings, requiring decisions based on comprehensive feature complexities extracted through convolutional layers. This architecture proved effective during training, demonstrated by a gradual decrease in losses, indicating the network's improved learning capability compared to our unsuccessful trial with ResNet.

In contrast to the approach proposed by [3], which utilized a transition layer before Global Average Pooling (GAP), we determined that such a layer was unnecessary for our setup. This decision was influenced by the unique characteristics of DenseNet-121, which incorporates bidirectional transition layers designed to enhance the discriminative power of features before applying GAP. Thus, our choice of DenseNet-121 as the backbone aligns with its intended design for feature extraction.

We used FC Layer with 8 outputs followed by a sigmoid since our problem is multi-label classification, so we need probability for each finding independently.

5.3.3.2. Accumulative Learning

Due to the limitation of the training resources, training this network with large batch sizes was a limitation for us, so we followed the technique of accumulating learning that increases the training time but reduces the resources usage making it double to train such a network on our limited resources. Following this technique, we were able to train this classifier in Colab [2] free trial plan with 15 GPU VRAM

5.3.4. Methodologies

5.3.4.1. Data Preprocessing

We used the same tarin-validation-test split used in X-Reporto trainer to prevent data leakage, especially when we decided to integrate the object detector with the report generator submodule. The first step involved running Exploratory Data Analysis (EDA) or on the data labels we obtained, as explained in the dataset section. During the analysis, we discovered the problem of null and -1 labels in the dataset. As shown below the histogram for the training subset in figure 5.9.4.1. 1. There are a lot of null values compared to the -1 or 0 labels.

We proposed several approaches with dealing with -1 labels:

1. **Dropping Examples with -1 Labels:** Initially we considered dropping examples with uncertain labels (-1). However, since this is a multi-label problem, an example might have a -1 for some labels while having 0 or 1 for others. Dropping these examples resulted in a significant loss of data, leaving us with too little to learn from.
2.
3. **Converting -1 to 1:** Our final approach and the most logical one was to convert the -1 labels to 1 label marking them as they need attention from the model. This reframed our problem from deciding whether a label is present to determining whether the model needs to focus on that label or not. We got the histogram after aping the U-Ones-Approach as illustrated in figure 4.3.3.6.

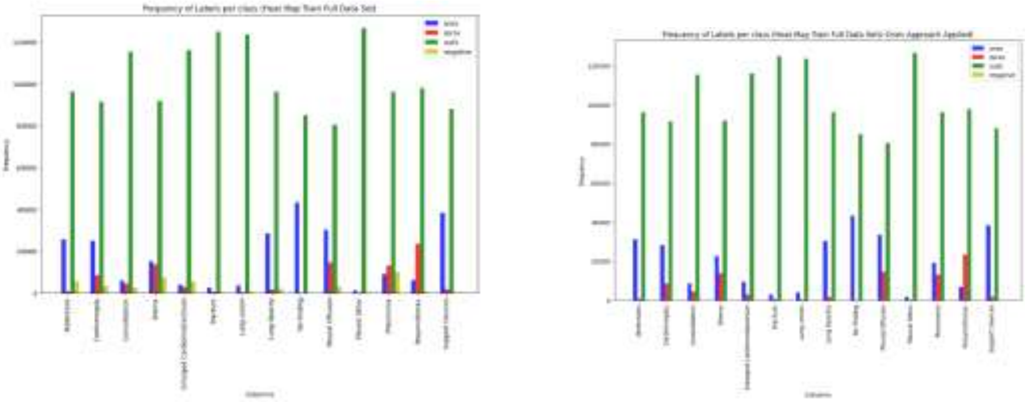


Figure 10 Histogram of the training split (4 Labels)

[2] https://colab.research.google.com/

After fixing the problem of the -1 labels, it was necessary to think about how we will deal with the null labels. In addition to the problem of data imbalance, we have a huge number of nulls compared to the 1 and even the 0 labels. We need to balance our data,

We proposed several approaches with dealing with Nan labels:

1. **Nan Balancing Approach:** We tried to propose a new approach to deal with this imbalance, so we decided to convert some null labels to 0 till we reach an equal number of zeros and ones for each label. So, we got balanced 0,1 labels but still a lot of nulls as shown in figure. During the training we applied a mask on the loss computation for the unconverted null labels, demonstrating our ignorance to the decision taken by the model on such labels. Unfortunately, we found out that this approach causes oscillations, and they couldn't achieve model convergence
2. **Converting Nan to 0:** Following the same procedure we used in dealing with -1, we decided to convert the null labels to zero. This decision is taken based on the evidence that not mentioning means not occurring because since the radiologist hasn't mentioned that this finding is neither positive nor negative, it is sufficient to label it as negative. This is due to the nature of such a problem, while diagnosing the X-Ray the radiologist checks every finding not just focus on some findings and ignoring the presence of others it is his responsibility.

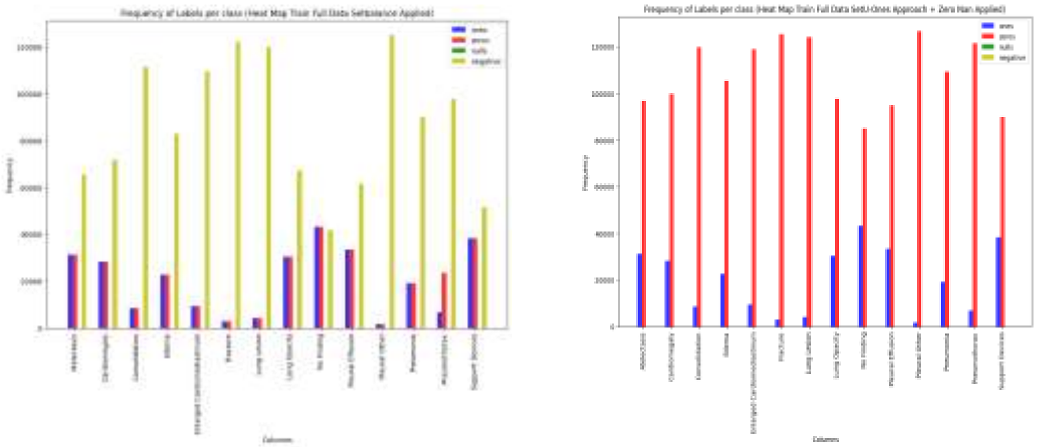


Figure 11 Histogram of the training subset with balanced 0 and 1 labels

The final preprocessed version of the dataset is shown in figure 5.9.4.1 4. We converted the -1 to be 1 and Nan to be 0 resulting in just 0,1 labels. This data has a severe class imbalance. The dataset lacks positive labels using such data will result in a biased classifier.

We proposed several approaches with dealing with Nan labels:

1. **Dropping examples:** We tried to drop the training examples with the number of 1 label below a certain threshold, but this resulted in the issue when we tried to drop the examples with -1 labels. No examples left to learn from due to the multi-label nature of the problem.
2. **Data Augmentation:** We applied several Data Augmentation techniques such as applying random Gaussian noise on the training examples. In addition, we applied random rotation for the image from -2 degrees and 2 degrees. Since data augmentation is a commonly used technique to address the problem of data imbalance. Although the data approach of augmenting data to address the problem of class imbalance is a common technique for dealing with class imbalance since here, we have multi-labels augmenting increase examples of 1 in certain label results in increasing more examples in Nans in other labels since no of Nans are a lot no patient doesn't have nan in its gold labels.
3. **Drop Labels:** Some Labels occur rarely such as:
   i. Consolidation
   ii. Enlarged Cardiomediastinum
   iii. Fracture
   iv. Lung Lesion
   v. Pleural Other
   vi. Pneumothorax

So, we decided to drop out these examples since there is actually nothing to learn from and using them in the training will only increase the complexity of the problem making it harder for the model to learn.

4. **Weighted Loss:** We decided to use weight loss to solve this issue. We computed weight per each label from the training data set computed as shown in the following equation. This was the most effective solution that worked with our trails

$$weight_{c_i} = \frac{no\ of\ zeros\ in\ c_i}{no\ of\ ones\ in\ c_i}$$

5.3.4.2. Training

In our training setup we used Adam optimizer to update the model's parameters. The optimizer is initialized with a learning rate of 5.12e-05, and beta values of 0.9 and 0.999 for the first and second moment estimates, Additionally, we employ a learning rate scheduler to further refine the training process. Specifically, we use the StepLR scheduler, which reduces the learning rate by a factor of 0.9 every epoch.

Regarding the loss function for the optimizer, we used Binary cross entropy with mean as the reduction function. These losses are used to update every 2 successive batches since batch is 32 and the effective batch size is 64. Training using the whole dataset with batch size 32 we get the total no of batches for one training epoch 4011. After finishing a complete epoch, we run another epoch on the validation dataset to keep track of overfitting. We first trained the network for 5 epochs, but it didn't saturate the losses in a continuous decrease so we realized we needed to add more epochs, but we were afraid of overfitting, so we kept track of the validation losses after each complete epoch.

So continued training for 5 more epochs but the best model was saved after the 8th epoch. After that we saw an increase in the validation losses. We suggest that the model overfit after completing 8 epochs due to the usage of GAP which is robust to overfitting which was proposed by [] a technique for regularizing training.

Chapter 6: System Testing and Verification

6.1. Classifier Testing

To test our classifiers, we conducted validation testing on a test split of our dataset and computed Precision, Recall, and F1-score for each classifier. The results for the Abnormal Classifier are illustrated in table

Regions	Precision	Recall	F1 Score
All	0.5915	0.8991	0.7133
Normal	0.607	0.4594	0.8973
Abnormal	1	0.8991	0.9469

Table 1 Abnormal Classifier Evaluation Metric Results

- **All**: Classification of regions as either normal or abnormal.
- **Normal**: Metrics computed only on regions classified as normal to assess detection accuracy.
- **Abnormal**: Metrics computed only on regions classified as abnormal to assess detection accuracy."

This structure provides a clear overview of how the results are organized in the table, distinguishing between overall classification and specific performance metrics for normal and abnormal classifications. Adjust the table headers and content based on your actual results for precision, recall, and F1-score.

The testing result for the region selection classifier is shown below in table

Regions	Precision	Recall	F1 Score
All	0.372	0.9076	0.47

Table 2 Region Selection Classifier Evaluation Metric Results

6.2. Report Generator Classifier Submodule Testing

Since the data is highly imbalanced then using f1-score as an evaluation metric for such a problem won't be effective so we used the AUC as an evaluation metric We compute the optimal thresholds from the true positive curve but since computing the ROC takes a lot of time so we decided to finish training and just validate each epoch

based on the loss to save the best models and later on we loaded the best trained model saved based on the average loss on the validation dataset and get the optimal thresholds per each class as shown in Table 3.It is clear that our thresholds are very small, near 0.2. This is due to the class imbalance problem we discussed before. This forces us to display the confidence level in correlation with the report, not just the polarity of the findings. The high sensitivity to the chosen threshold results from the significant data imbalance.
Investigating results in Table 4 computed the ROC Curve of our classification network on the validation subset we found that the results are perfect. We are having AUC for most of the labels above

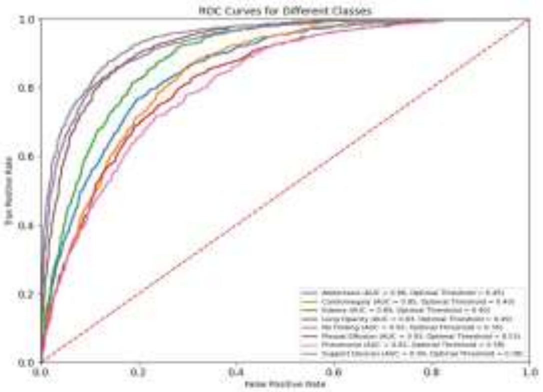


Figure 12 ROC Curve for Classifier

Label	Optimal Threshold
Atelectasis	0.279
Cardiomegaly	0.279
Edema	0.214
Lung Opacity	0.21
No Finding	0.35
Pleural Effusion	0.228
Pneumonia	0.135

Table 3 Optimal thresholds based on ROC Curve

Due to the severe imbalance in the data set using f1-score as a metric will be very sensitive to such problem so the main metric we used for the evaluation of our model is the AUC, but we have also run the evaluation for f1-sore and classification retics to just have intuition about the behavior of the classifier.

Class	Precision	Recall	F1 Score	FP	FN	TP	TN
Atelectasis	0.3767	0.6864	0.4865	134	37	81	448
Cardiomegaly	0.3065	0.6477	0.4161	129	31	57	483
Edema	0.3873	0.6875	0.4955	87	25	55	533
Lung Opacity	0.3849	0.6644	0.4874	155	49	97	399
No Finding	0.6824	0.8202	0.7450	121	57	260	262
Pleural Effusion	0.5692	0.8102	0.6687	84	26	111	479
Pneumonia	0.3415	0.5932	0.4334	135	48	70	447
Support Devices	0.5621	0.7167	0.6300	67	34	86	513

Table 4 Template Based Classifier Metric Results

6.3 Saliency Map Results:

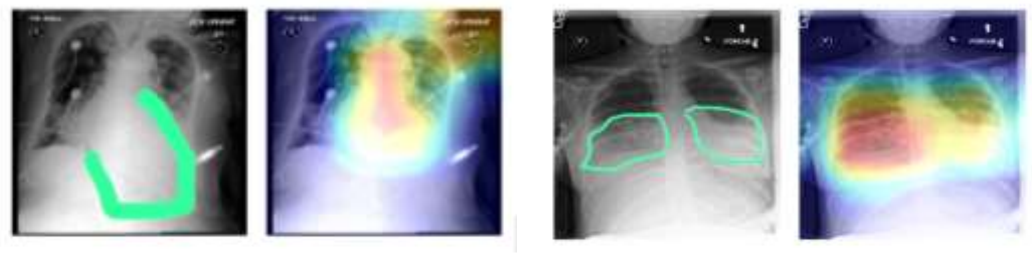


Figure 13 Cardiomology and Penenminia localization saliency maps with doctor's annotation

Chapter 7: Conclusions and Future Work

My biggest challenge was entering the AI field, needing to study PyTorch to build my first module, the classifier. Later, I faced the challenge of template-based report generation. This problem was significant as it involved solving issues like dataset imbalance, training neural networks, and validating results, including heatmaps. Since ground truths for heatmaps were unavailable, I manually annotated them with the help of doctors. Another challenge was designing our tool. I consulted numerous Egyptian radiologists about their tools for writing reports, finding that most used Microsoft Word processing. There was no existing system like ours for handling X-rays in hospitals, even without AI. I researched medical companies to see their tools and incorporated ideas from Egyptian doctors into my UX study. Subsequently, I began designing the UI for our tool, aiming to make it user-friendly for radiologists. My future work is to use the report history as input with image to the report generation to get better results. I will another approaches like grid-CAM for better localization

References

[1] Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 2921-2929).

[2] Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 2097-2106).

[3] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 4700-4708).

[4] M. Lin, Q. Chen, and S. Yan. Network in network. International Conference on Learning Representations, 2014.

Appendix A: Other Workload

I have built the Docker files for the AI service for further deployment of our tool on the cloud. I was responsible for the template-based and bounding boxes pages in the tool as part of the frontend team./I conducted a UX study for the users of this application before we began the development process.
I built the screens for the UI of the application to make it easy to use and suitable for the medical field. All the project screens are available on Figma.[3]

[3] https://www.figma.com/design/5oAl4ysefyB13EsmY8ReZV/X-Reporto-UI?node-id=0-1&t=jQhbua1ZbnVaVAJK-1