University of Calcutta

Department of Computer Science and Engineering

Medical Image Report Generator From X-Ray Images

A project report submitted to the Department of Computer Science and Engineering, University of Calcutta, in the fulfillment of the requirements for the degree of Bachelors in Technology in Computer Science and Engineering.

Submitted By

Rupam Kumar Roy

Roll No.: T91/PST/184132

Reg No.: D01-1111-0094-18

University Of Calcutta

Amartya Bhattacharya

Roll No.:T91/CSE/184040

Reg No.:D01-1111-0063-18

University Of Calcutta

Hrithik Anand

Roll No.:T91/CSE/184037

Reg No.:D01-1111-0060-18

University Of Calcutta

Under the Supervision of

Dr Rajib Kumar Das

Associate Professor

Department of Computer Science and Engineering

University of Calcutta

June,2022

Department of Computer Science and Engineering
University of Calcutta

CERTIFICATION

This is to certify that the project entitled “Medical **Image Report Generator From X-Ray Images**” submitted by Rupam Kumar Roy, Amartya Bhattacharya and Hrithik Anand developed under the supervision of Dr Rajib Kumar Das as been prepared according to the rules and regulations for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** under **University of Calcutta** for the year **2022** and the candidates have fulfilled the requirements for submission of the project.

……………………………..
Chairman, Board of Studies,
B.Tech (Computer Science and Engineering)
Department of Computer Science and Engineering
University of Calcutta.

………………………………………….
Dr. Rajib Kumar Das
Project Supervisor
Department of Computer Science and Engineering
University of Calcutta.

…………………………………………
External Examiner

ACKNOWLEDGEMENT

We wish to take this opportunity to express our deep sense of respect and sincere gratitude to our project supervisor Dr. Rajib Kumar Das, Associate Professor, Department of Computer Science and Engineering for giving us the golden opportunity to do this interesting project under his supervision. His expertise and valuable suggestions have been a constant source of encouragement for working towards the development of the project work.

We would also like to convoy our regards and thanks to the Department of Computer Science and Engineering , University of Calcutta for providing us the wonderful environment for the study and perform our research work in a world class facility in the laboratory. We would also like to pay our sincere thanks to our seniors, classmates and also to all the faculty members for their cooperation during the preparation of this project.

…………………………….
Rupam Kumar Roy
Roll No.: T91/PST/184132
Reg No.: D01-1111-0094-18
University Of Calcutta

……….……………………...
Amartya Bhattacharya
Roll No.:T91/CSE/184040
Reg No.:D01-1111-0063-18
University Of Calcutta

………………………………
Hrithik Anand
Roll No.:T91/CSE/184037
Reg No.:D01-1111-0060-18
B.Tech Semester:VIII
University Of Calcutta

CONTENTS

1. Introduction

Generation of reports from chest X-Rays is a task that is done manually by the doctors. This involves a time consuming process where the patients have to wait for days in order to get an appointment with the doctors after receiving their X-Ray reports. This also incurs a certain charge for the patients which plays a huge role when patients are not able to deal with the expensive appointment costs. Medical Imaging has been under the focus of the researchers especially after the happening of the COVID-19 virus when thousands of patients were getting admitted every hour and the doctors had to take a lot of stress in order to create such reports timely and accurately. And in these cases of emergency, the time required to get such reports plays a vital role for the survival of a patient thus there is a need for automatic generation of reports from chest X-ray images.

In this project we tried to build a neural network based model that is capable of generating reports from chest X-ray images. For this work we took help of the open source data of both images and their corresponding reports provided by Indiana University. Where the images are in the format of ".png" and the reports are in the XML format. We try to build an accurate model and also to deploy the model into a web application so that the general people can use them in order to get their reports within a few seconds.

Below we have dealt with the sections of related works, data analysis part, the methods applied, the training process done, the results obtained as a part of the experiment, the conclusions made, the acknowledgement and the references taken.

2. Related Works

The problem of captioning images has been a topic of research for quite some years. It started with the work of Antol et.al.2015 [6]. The model was created in order to find the object and name it using a convolutional neural network based model. The objective was to identify the object present in the image and name it. The applications of visual question answering have been very few in the medical domain. The notable works that have been by Allaouzi et.al.2019[7], where the researchers suggested an encoder-decoder based model for image captioning. This model's objective was to extract the features from images and map it with the suitable reports. Although the work provided a pathway for the development of such encoder decoder based models for image captioning the model failed to achieve a good accuracy score as discussed in the Results section of this project. Moreover it was used to predict a single word which was the name of the disease present if there were any rather than a detailed report that we wanted.

Recent works include the work done by Sharma et.al 2021[8]. In his research paper he suggested a multi-modal model capable of captioning the image with the help of already existing reports and the images. This work fails to generate reports from the images only. In our paper we wish to present a model capable of generating reports from Chest X-Ray images only. The model although requires the reports for training the images, it doesn't involve the report for testing the data making our model semi supervised. Our model aims to achieve a high accuracy even on the outliers that are present in our data. Also we wish to build a web application capable of showing the reports to the user after accepting one or two chest X-ray images.

3. Data Analysis

The data as discussed in the introduction section was taken from an open source forum and was hand curated by the researchers in Indiana University. The data consisted of chest X-Rays and their corresponding reports in XML format.

There were multiple images associated with a single report. For most of the cases it was observed that there were 2 images associated with a single report while the range of the number of reports was from 0 to 5, the distribution of which has been plotted below in **Fig 1.**

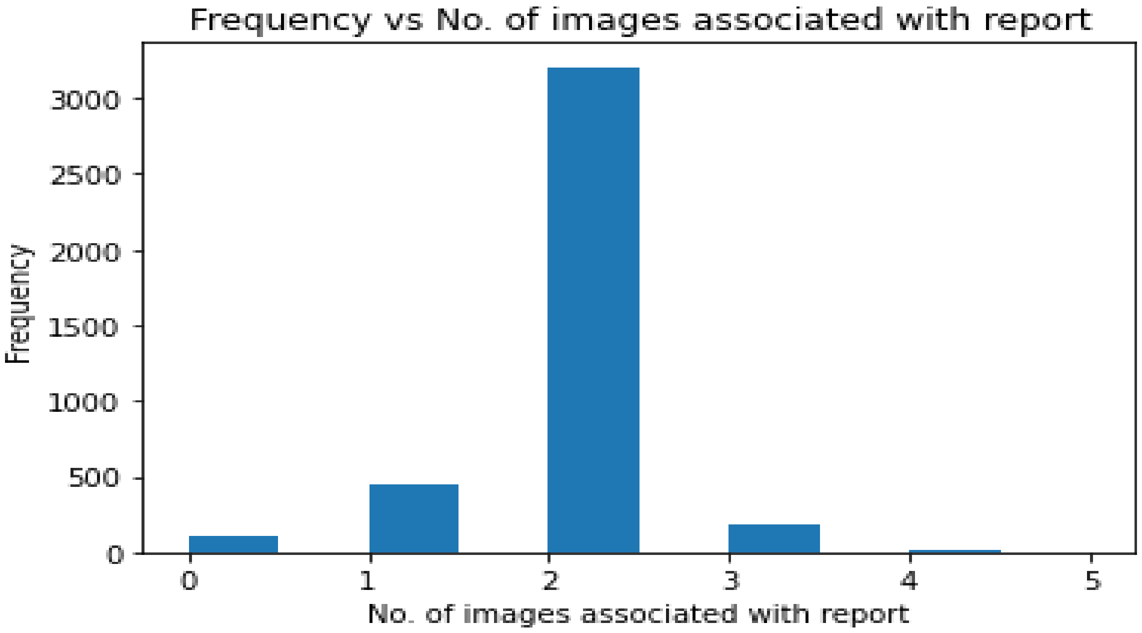


Fig 1. Frequency vs Number of images associates with the report

Since in maximum of the cases the number of images associated was found to be two, we took 2 images i.e the front face and the side face image of the chest X-Rays. **Fig 2.** And **Fig 3.** represent the front as well as the side faced images.

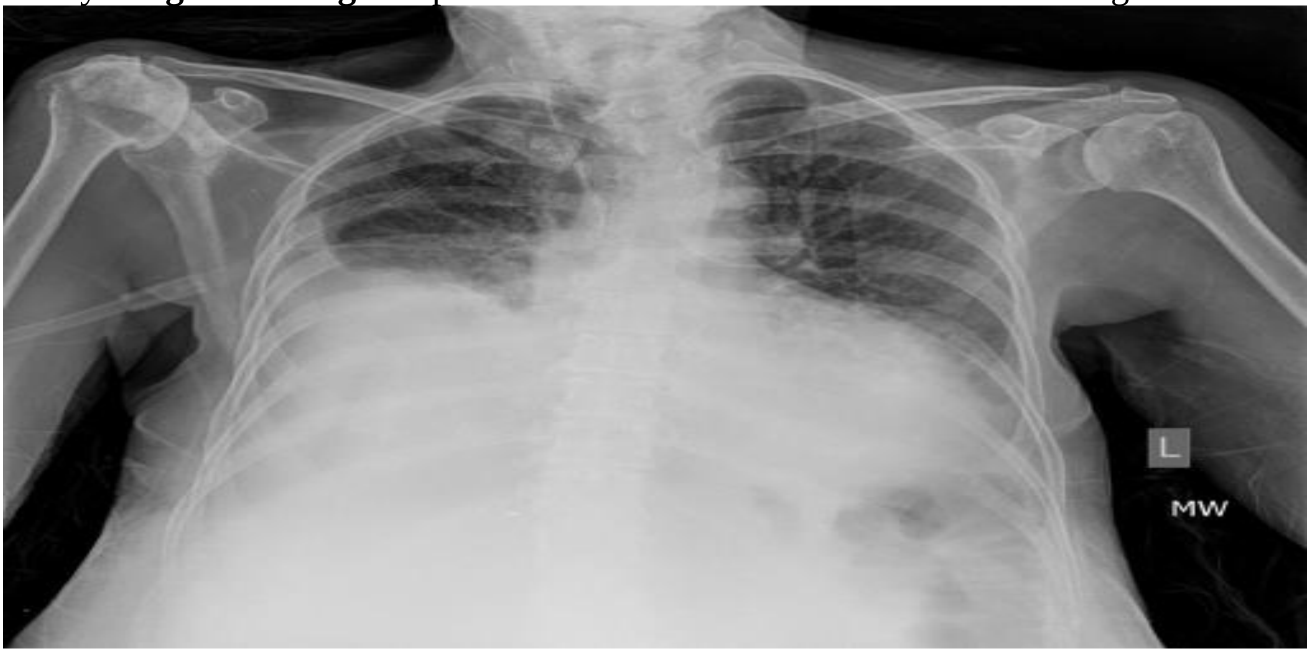


Fig 2. Front face of the chest X-Ray

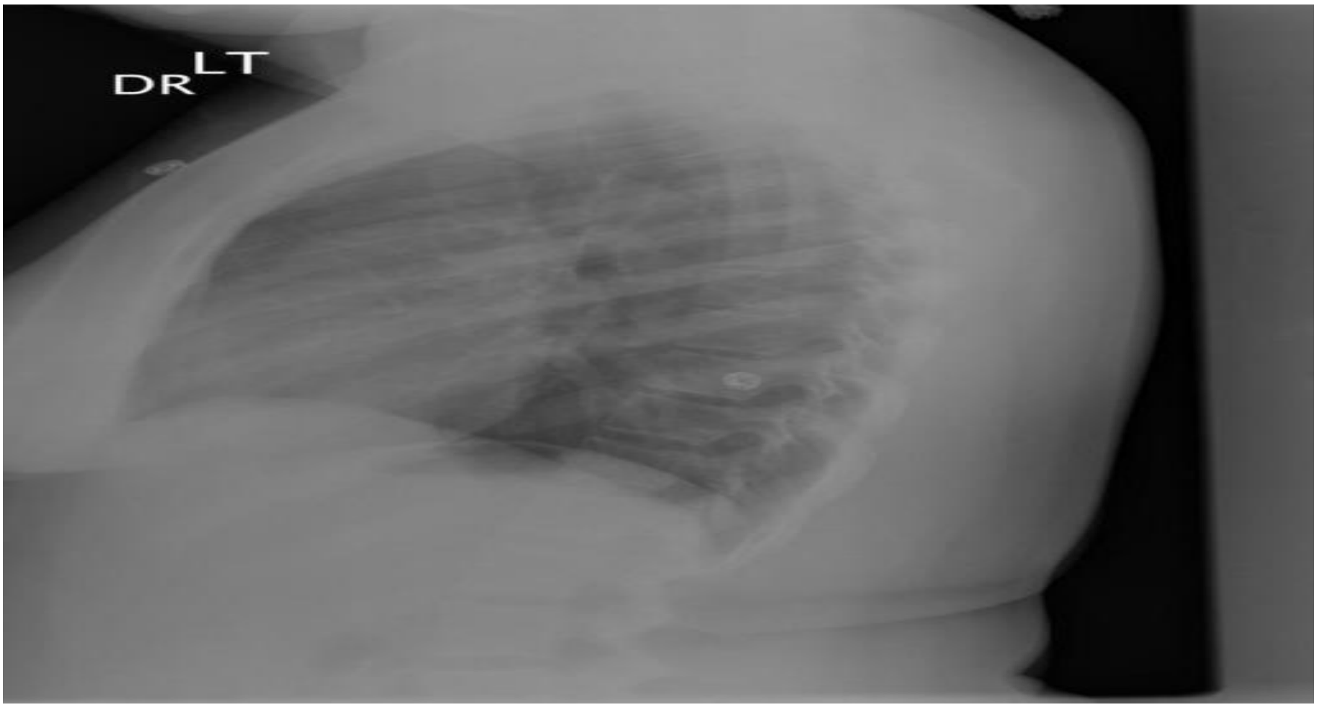
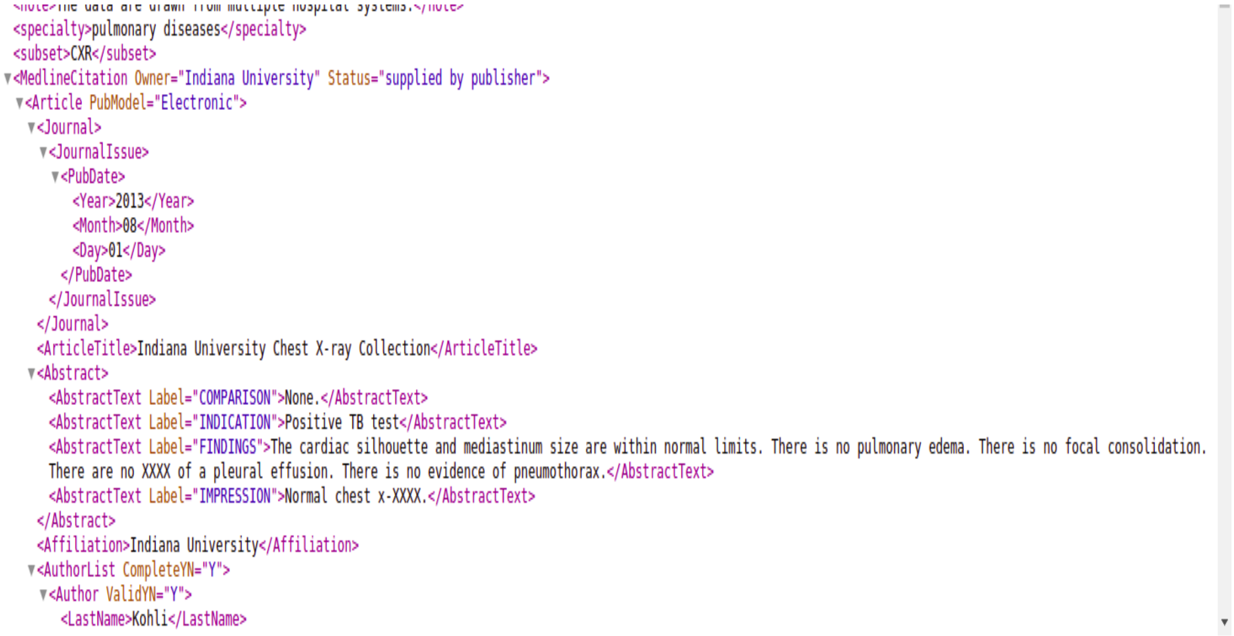


Fig 3. Side face of the chest X-Ray

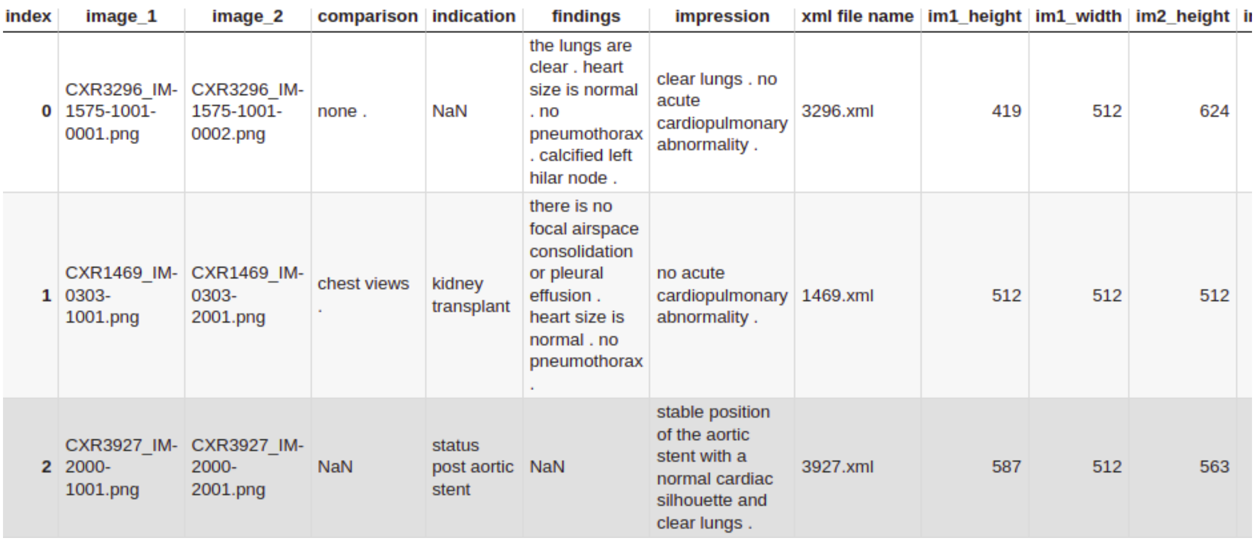
Extraction of the report was done using a regex as the format was in XML. The part of the sample XML file has been shown in **Fig 4.**



```
<note>The data are drawn from multiple hospital systems.</note>
<specialty>pulmonary diseases</specialty>
<subset>CXR</subset>
▼<MedlineCitation Owner="Indiana University" Status="supplied by publisher">
  ▼<Article PubModel="Electronic">
    ▼<Journal>
      ▼<JournalIssue>
        ▼<PubDate>
            <Year>2013</Year>
            <Month>08</Month>
            <Day>01</Day>
          </PubDate>
        </JournalIssue>
      </Journal>
      <ArticleTitle>Indiana University Chest X-ray Collection</ArticleTitle>
    ▼<Abstract>
        <AbstractText Label="COMPARISON">None.</AbstractText>
        <AbstractText Label="INDICATION">Positive TB test</AbstractText>
        <AbstractText Label="FINDINGS">The cardiac silhouette and mediastinum size are within normal limits. There is no pulmonary edema. There is no focal consolidation.
        There are no XXXX of a pleural effusion. There is no evidence of pneumothorax.</AbstractText>
        <AbstractText Label="IMPRESSION">Normal chest x-XXXX.</AbstractText>
      </Abstract>
      <Affiliation>Indiana University</Affiliation>
    ▼<AuthorList CompleteYN="Y">
      ▼<Author ValidYN="Y">
          <LastName>Kohli</LastName>
```

Fig 4. Sample Report in XML format

Using Regular Expressions(RegEx) we extracted the labels "Comparison", "Indication" and "Impression" as these 3 features were important for report generation.
The image names as well as their corresponding report was stored into a dataframe format so that it can be easier to access the images as well as the reports. **Fig 5.** depicts the table clearly the features which were taken.



index	image_1	image_2	comparison	indication	findings	impression	xml file name	im1_height	im1_width	im2_height	i
0	CXR3296_IM-1575-1001-0001.png	CXR3296_IM-1575-1001-0002.png	none .	NaN	the lungs are clear . heart size is normal . no pneumothorax . calcified left hilar node .	clear lungs . no acute cardiopulmonary abnormality .	3296.xml	419	512	624	
1	CXR1469_IM-0303-1001.png	CXR1469_IM-0303-2001.png	chest views .	kidney transplant	there is no focal airspace consolidation or pleural effusion . heart size is normal . no pneumothorax .	no acute cardiopulmonary abnormality .	1469.xml	512	512	512	
2	CXR3927_IM-2000-1001.png	CXR3927_IM-2000-2001.png	NaN	status post aortic stent	NaN	stable position of the aortic stent with a normal cardiac silhouette and clear lungs .	3927.xml	587	512	563	

Fig 5. Sample Dataframe containing information

As null values can be noticed in the data frames, we chose to drop the rows containing the null values in our way forward to creating the clean data. Now, it is important to note that all the images present in our data have different sizes(refer to **Fig 6.**) which is difficult to deal with, while creating a model so we decided to make all the images of equal shape with the help of the resize function present in OpenCV.

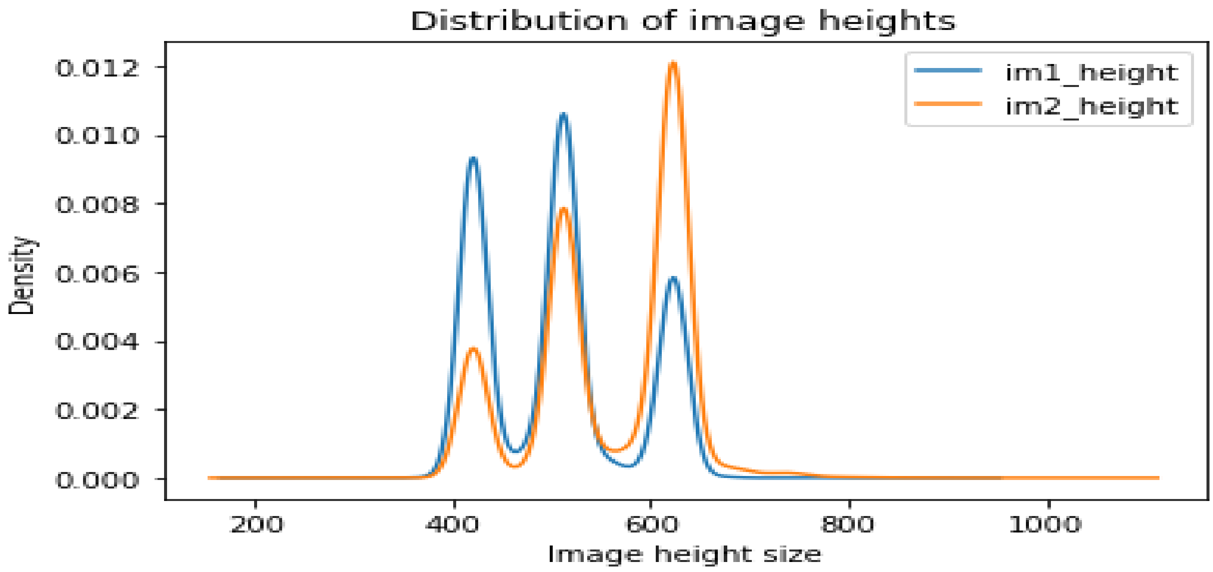
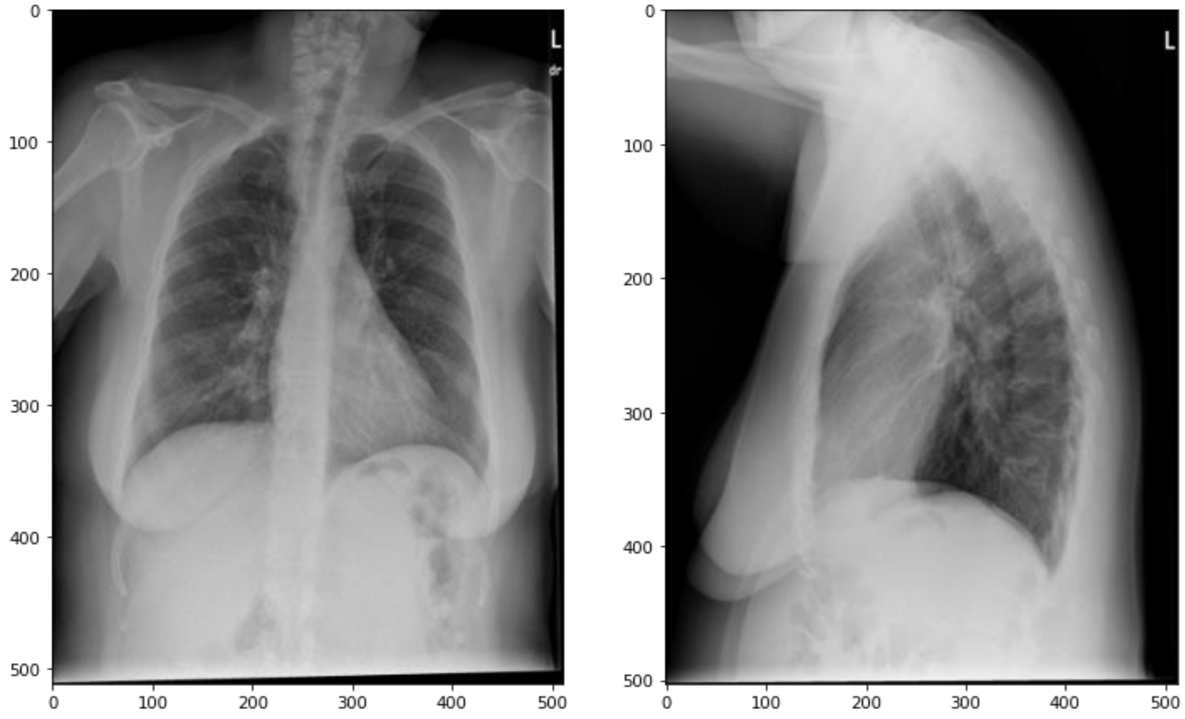
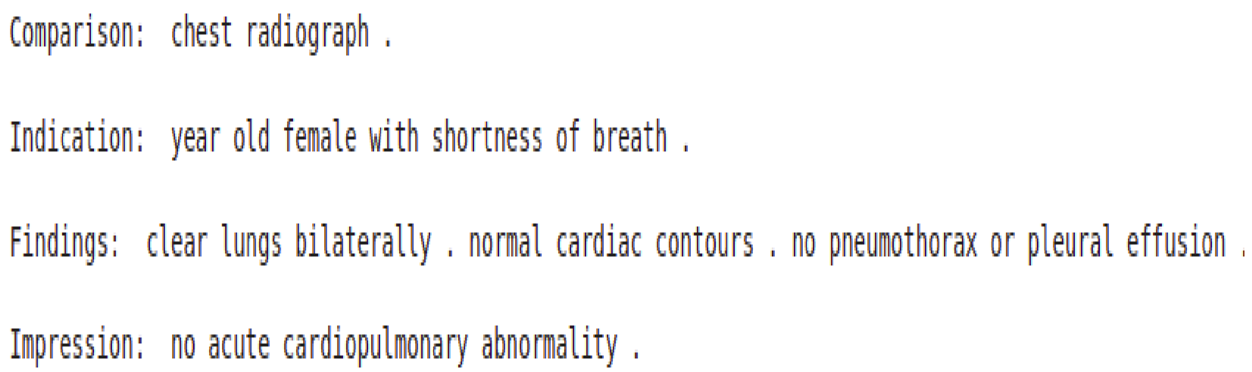


Fig 6. Distribution of image heights

Now the images along with their reports were plotted as samples to show how the whole data would look like for a single instance.
This has been shown in **Fig 7.**





Comparison: chest radiograph .

Indication: year old female with shortness of breath .

Findings: clear lungs bilaterally . normal cardiac contours . no pneumothorax or pleural effusion .

Impression: no acute cardiopulmonary abnormality .

Fig 7. Sample image and its corresponding report

4. Methods And Models

We went on ahead with two approaches one method was taking the classification approach. We tried to create the 112 target variables which represented each of the unique words present in the impressions, findings and indications. For each of the target variables we tried to train a model which would predict the presence of a certain word. If the word was present it would be marked as 1 else 0. But one challenge was to extract the given words and also there are cases where adjectives and the nouns go together and need to be extracted as a single word. This was done with the help of word cloud. The sample word cloud image has been given in **Fig 8.**

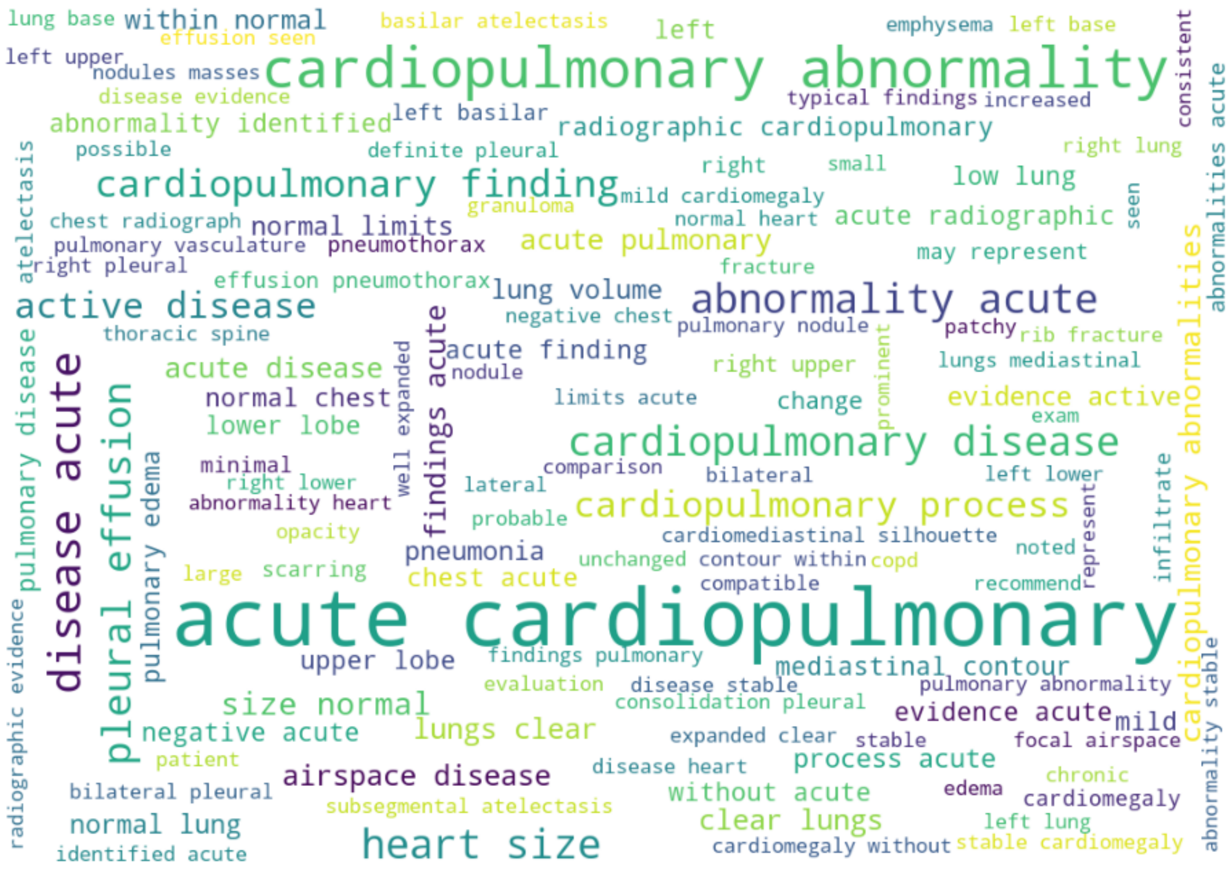


Fig 8. Word Cloud

5. Simple Classification Model

For the encoder portion we chose a CNN based architecture that has the best score on the CheXpert site[1]. The model can extract features and classify the chest X-ray images based on the features extracted into 1000 different classes. So we chose the only extraction of features, part of the model in order to classify the words present and the part was attached to a classifier. The model architecture presented of the CheXpert model has been given below in **Fig 9.**

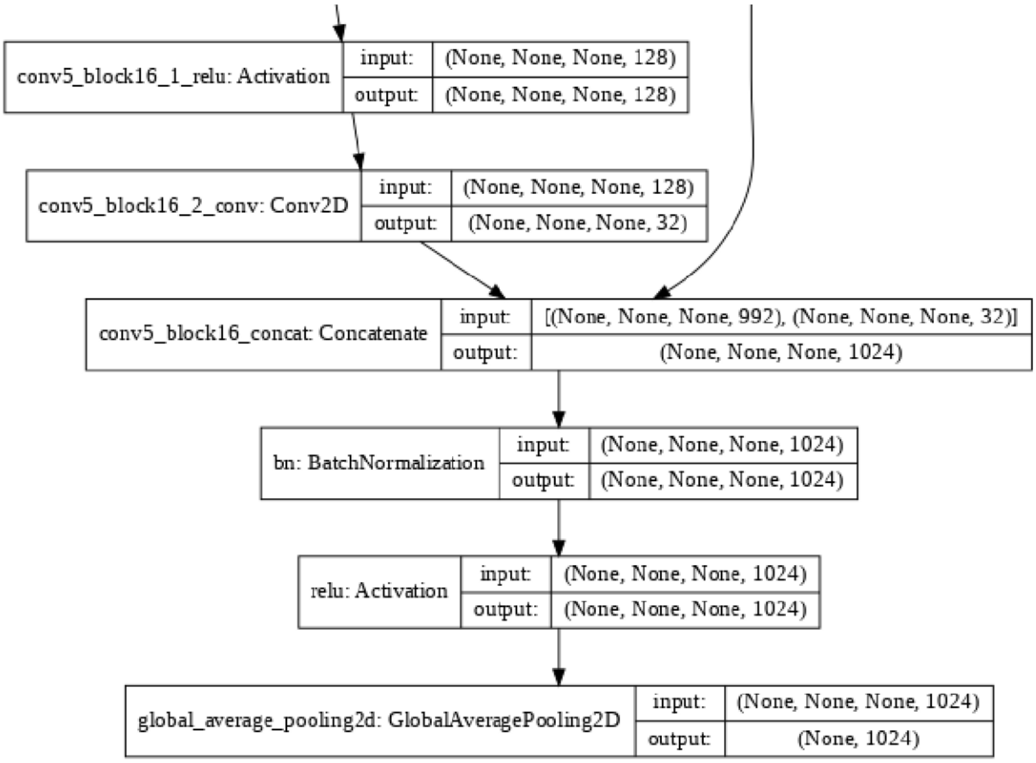

conv5_block16_1_relu: Activation
input: (None, None, None, 128)
output: (None, None, None, 128)
conv5_block16_2_conv: Conv2D
input: (None, None, None, 128)
output: (None, None, None, 32)
conv5_block16_concat: Concatenate
input: [(None, None, None, 992), (None, None, None, 32)]
output: (None, None, None, 1024)
bn: BatchNormalization
input: (None, None, None, 1024)
output: (None, None, None, 1024)
relu: Activation
input: (None, None, None, 1024)
output: (None, None, None, 1024)
global_average_pooling2d: GlobalAveragePooling2D
input: (None, None, None, 1024)
output: (None, 1024)

Fig 9. CNN part of the classification

Here there are various layers and each of the layers has been explained below

i) **Conv2D** - The Conv2D represents a layer for the operation of convolution on a 2D matrix. A 2D kernel is selected and it is slid over through all of the pixels. While sliding through the window of pixels each of them, every time the pixels get multiplied by the values present in the kernel or the filters and are added. While representing as a vector operation each of the convolution operations involve the calculation of a dot product. **Fig 18.** shows how the convolutional operations work.

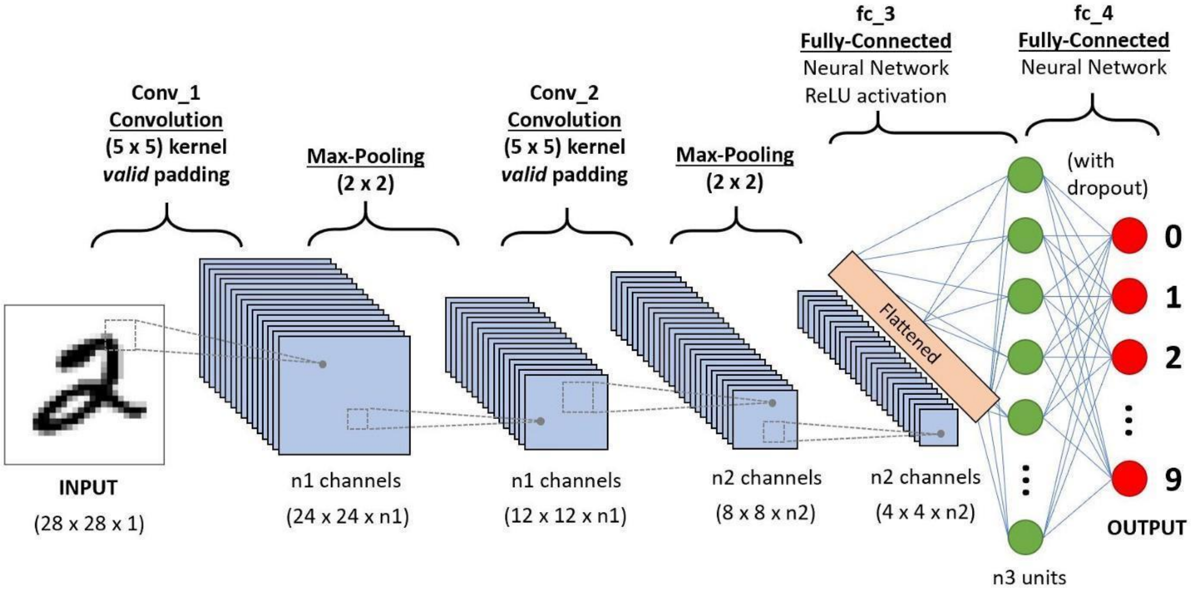


Fig 9.Operation of Convolutional 2D layer on an Image

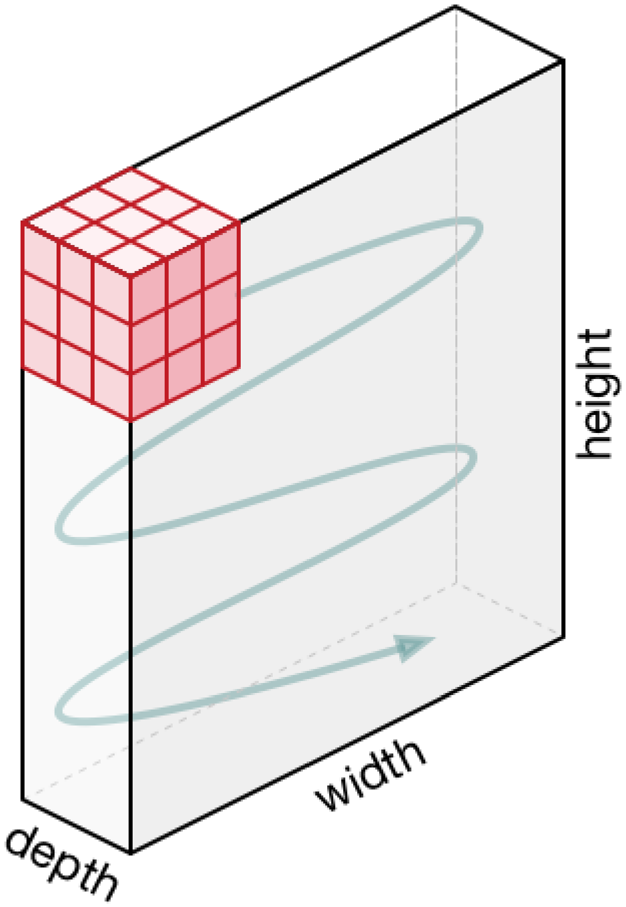


Fig 10. The sliding window operation of a kernel on a 2D matrix

Fig 10. depicts how a chosen filter slides through the 2D matrix which may or may not represent an image.

ii) **Activation Layer-** Activation layers are one of the most important components of a neural network model. It has majorly 2 functions i.e a) Bring the nonlinearity into the equation so that the model can fit the cases having non linear nature. and b) Reduce the range of the values which are obtained by multiplying the pixel values with the weights of the filter values. Some of the activation layers used have been given below

- Rectified Linear Unit (ReLU) : ReLu is a non-linear activation function that is used in multi-layer neural networks or deep neural networks. This function can be represented as:

$$\boldsymbol{f(x) = \max(0, x)} \qquad (1)$$

where x = an input value

According to equation 1, the output of ReLu is the maximum value between zero and the input value. An output is equal to zero when the input value is negative and the input value when the input is positive. Thus, we can rewrite equation 1 as follows:

$$\boldsymbol{f(x) = \begin{cases} 0, if\ x < 0 \\ x, if\ x \geq 0 \end{cases}} \qquad (2)$$

where x = an input value

- Sigmoid Activation Layer: The sigmoid activation layer is a layer that uses the sigmoid function in order to bring down the value of a certain outcome to a value between 0 and 1. The sigmoid is generally used in the last layer of the neural network of a classification model. The sigmoid function has been shown below

$$\sigma(x) = \frac{1}{1+e^{-x}} \qquad (3)$$

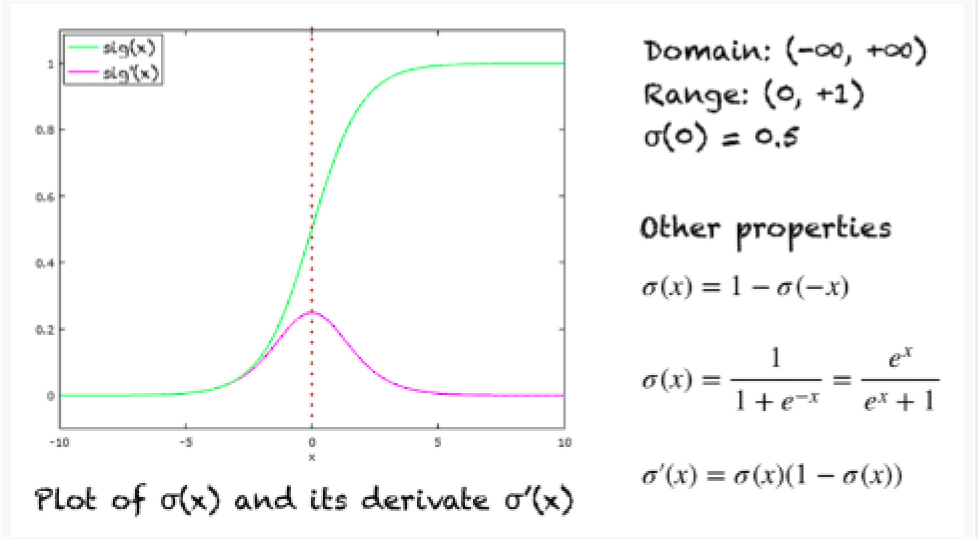


Fig 11.The domain and working principle of sigmoid function

The domain and the behavior of the sigmoid function has been shown in **Fig 11.**

iii) BatchNormalization Layer: Normalization is a data pre-processing tool used to bring the numerical data to a common scale without distorting its shape. Generally, when we input the data to a machine or deep learning algorithm we tend to change the values to a balanced scale. The reason we normalize is partly to ensure that our model can generalize appropriately. Now coming back to Batch normalization, it is a process to make neural networks faster and more stable through adding extra layers in a deep neural network. The new layer performs the standardizing and normalizing operations on the input of a layer coming from a previous layer. But what is the reason behind the term “Batch” in batch normalization? A typical neural network is trained using a collected set of input data called batch. Similarly, the normalizing process in batch normalization takes place in batches, not as a single input.

6. Encoder Decoder Model

In this part, we built an encoder-decoder based model, where the “CheXpert” model acts as the encoder part and is responsible for downsampling the images into features and then we pass into the model a notation of the start of the sentence i.e **<SRC>** . Both the features and the word <src> are passed into a

LSTM based decoder that works by predicting the corresponding words, correspondingly. The end of the line is denoted by **<EOL>** word. So the model keeps on predicting words until it predicts the <EOL> word. The decoder architecture has been given in the **Fig 12** below.

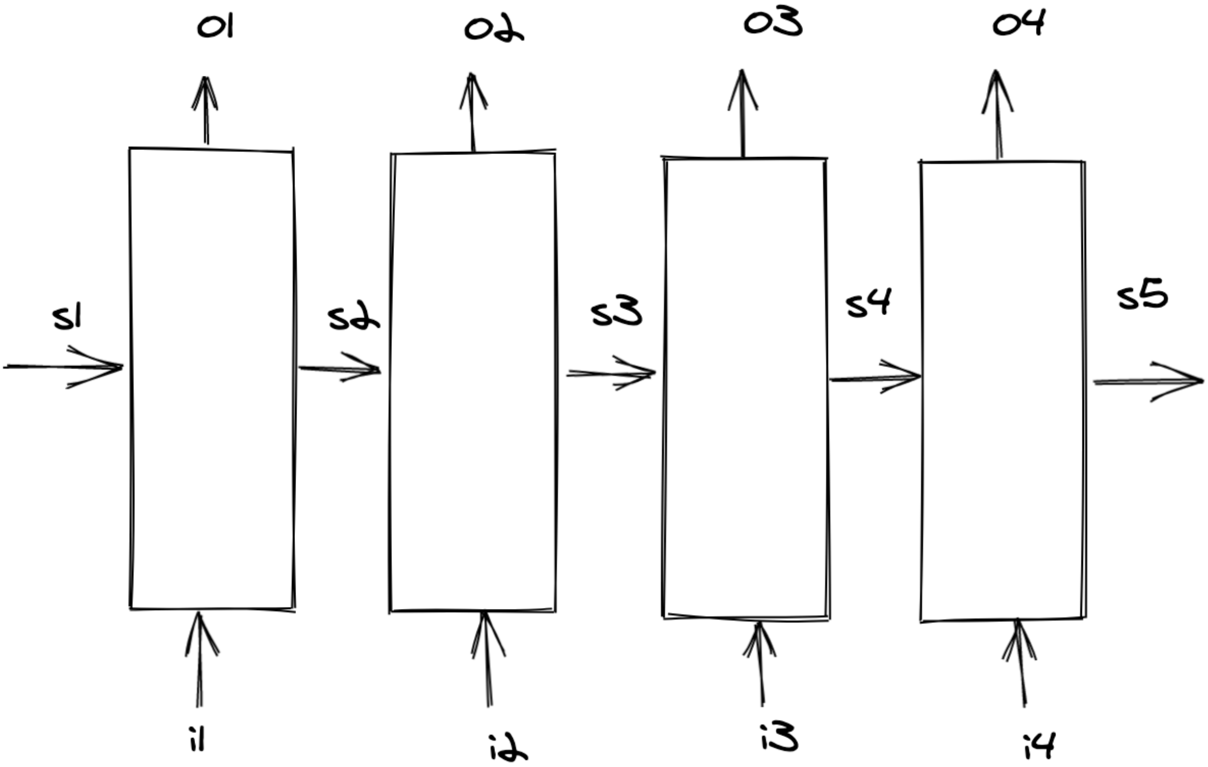


Fig 12. Decoder part of the model

Here the S1 is the input features extracted from the image and the word <SRC>.
The output consists of 29 LSTM blocks each designed to output one word.
The whole model has been shown in **Fig 13**.

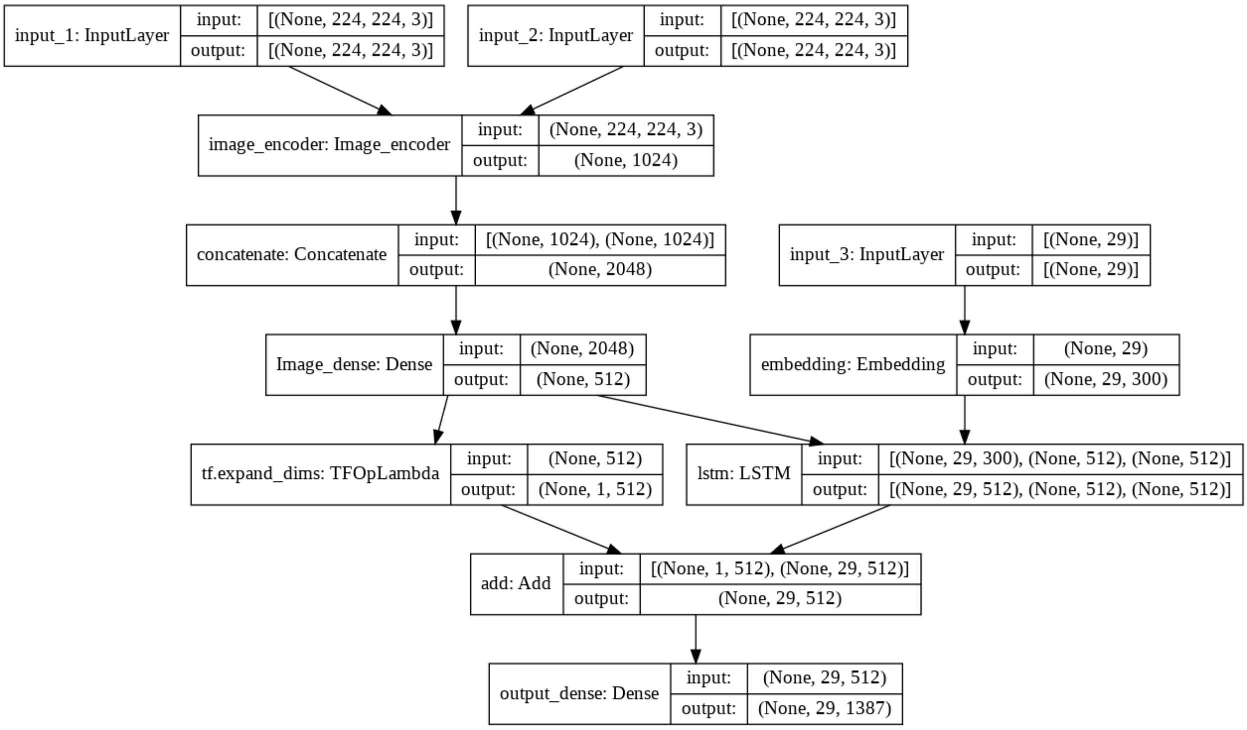


Fig 13. Encoder-Decoder Model

The LSTM blocks act as the decoder model part. Each of the LSTM blocks are responsible for generating a single word. The working principle of the Long Short Term Memory or the LSTM blocks have been explained below.

LSTM[9] - The key to LSTMs is the cell state, the horizontal line running through the top of the diagram. The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It’s very easy for information to just flow along it unchanged.

The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.

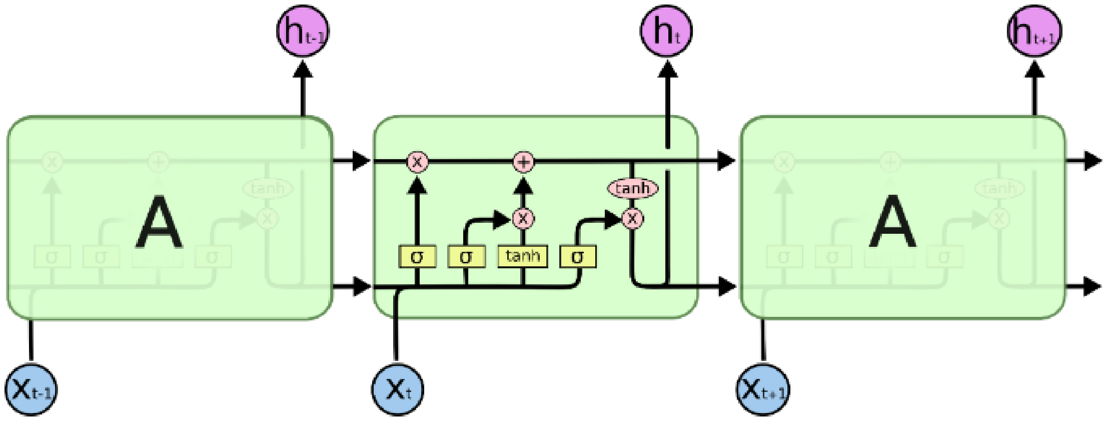


Fig 14. The working principle of LSTM networks

Fig 14. explains the working principle of the LSTM networks

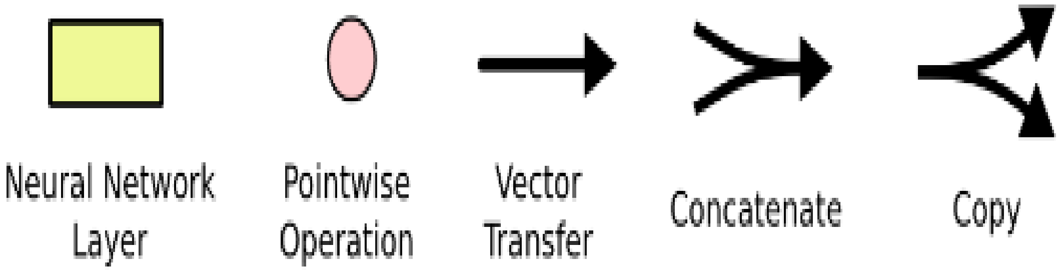


Fig 15. The functions used in a LSTM block

These operations are mentioned in **Fig 15.** are used while doing the operations in a single LSTM block. The point wise AND operation is responsible for “forgetting” the irrelevant information. The concatenate operation is used to add the important information or the context for a given LSTM block. The Xi are the inputs given in a sequential manner to the network.

7. Attention Model

Here we used the concept of global attention[2]. Attention works in a way so as to give more weight to the words that are important for our context. For example the words that belong to the Noun and the adjective class are given more preference in the sentence as it acts as the subject. Here we used the same pre-trained model as used in the encoder-decoder model, i.e CheXpert. It was used for extracting the features from the chest X-Ray images. These features were then formed into vectors which were then passed to the decoder which consisted of the attention layers. Before passing the text data into the model, embeddings were taken. These embeddings generally gave weights to the words that are similar to each other. For example the objects are given weights from a distribution of very low variance. But the difference between the weights between objects and adjectives is large. These embeddings helped us to capture the context. For each of the words, each word while getting predicted is passed through a global attention layer and then into a LSTM layer which uses the importance of the input words in order to predict the next word. This was continued until the <EOL> was predicted.

Attention is a technique where we give more weight to the relevant information and less weight to the irrelevant information. The importance of a certain word in a sentence is judged by a series of vector operations. The operations have been given below in **Fig 16.**

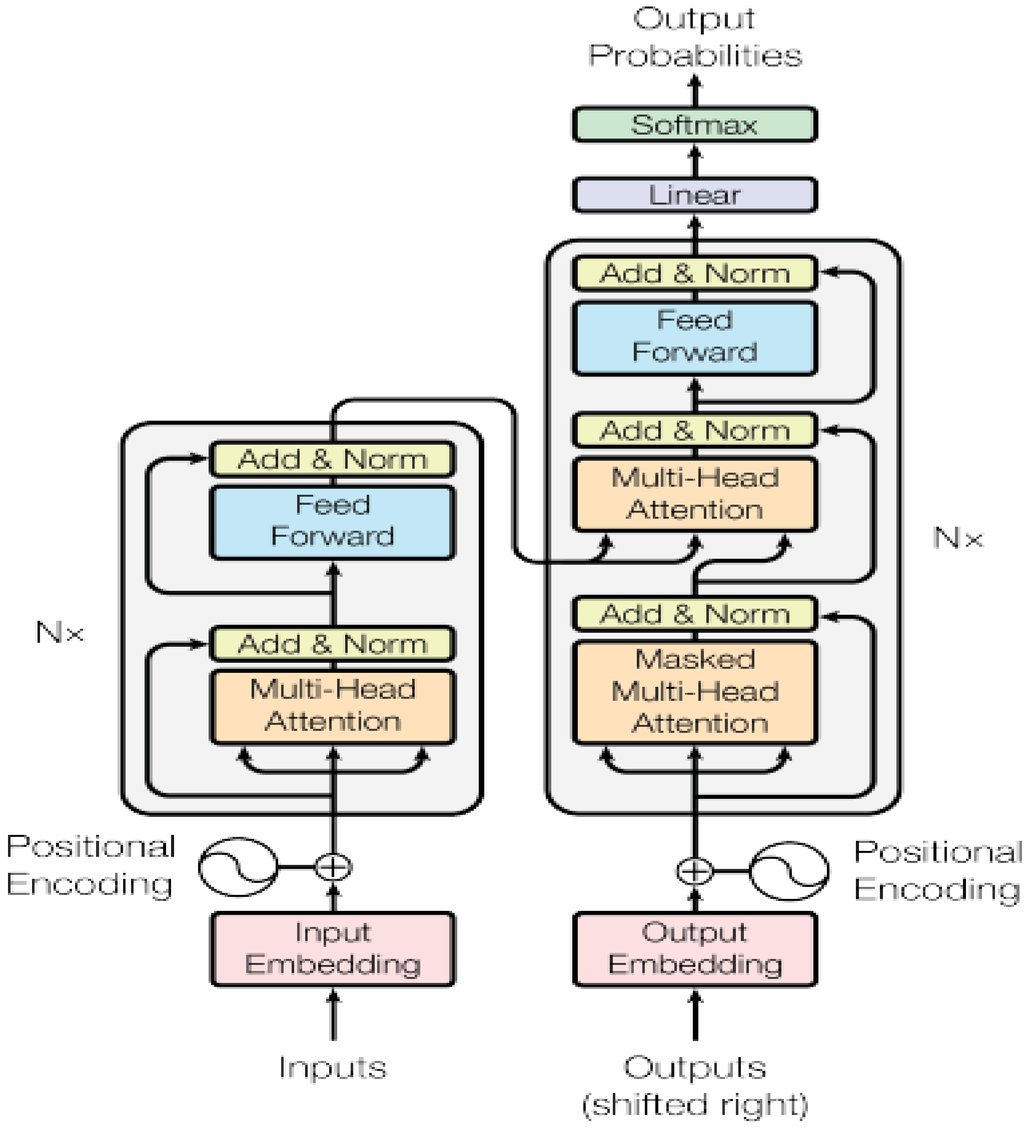


Fig 16. Attention mechanism architecture

The whole architecture of the transformer with self attention is given above. This model is capable of accepting words in parallel instead of in a sequential manner and thus reducing the overall computation time than the LSTM networks. This whole model contains a multi-head attention layer along with a feedforward network which forwards the attention values for further processing. The process of the Multi-Head attention has been given below.

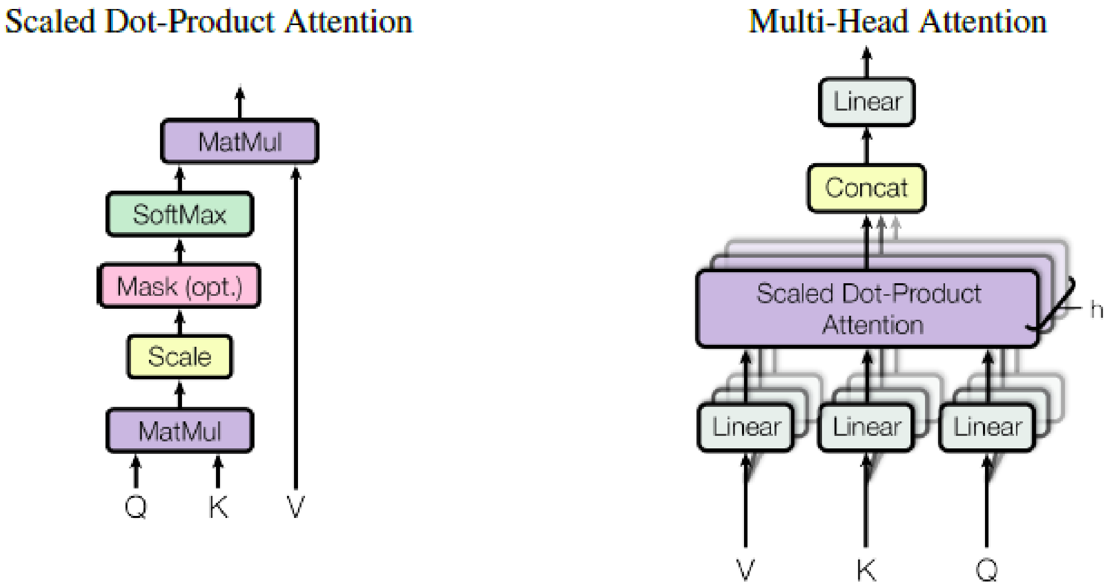


Fig 17. The multi-head attention mechanism

The multihead attention involves taking multiple inputs and performing scaled dot product attention on them and then going to the feed forward portion.

7.1. Training

Before the training process was started the feature named “impression” was padded with <SRC> and <EOL> string at the start and the end of the sentence to mark the beginning and the end.
It was noted that the one particular report was in majority of the samples and some of the reports were sparse in number. Thus minority upsampling and majority downsampling[3] was used in order to create an even distribution of the reports. All the images were then resized into the shape of (224,224,3) as the inputs of the CheXNet demanded the same size.
As for the hyperparameters, the loss function was the custom loss function that would determine the loss by comparing the number of common words present both in the true variable and the predicted variable with the help of Sparse Categorical Crossentropy[4]. The optimizer was chosen as the Adam optimizer with a learning rate of 0.001. The whole training process was monitored by the reduction of the validation loss. If the validation loss was found to be near constant continuously for 5 epochs, the process was stopped.

The metrics chosen was the accuracy score. Though the method of calculating the accuracy score was modified. It was made to compare the true and the predicted sentences and determine the number of words that match with the prediction and true labels.

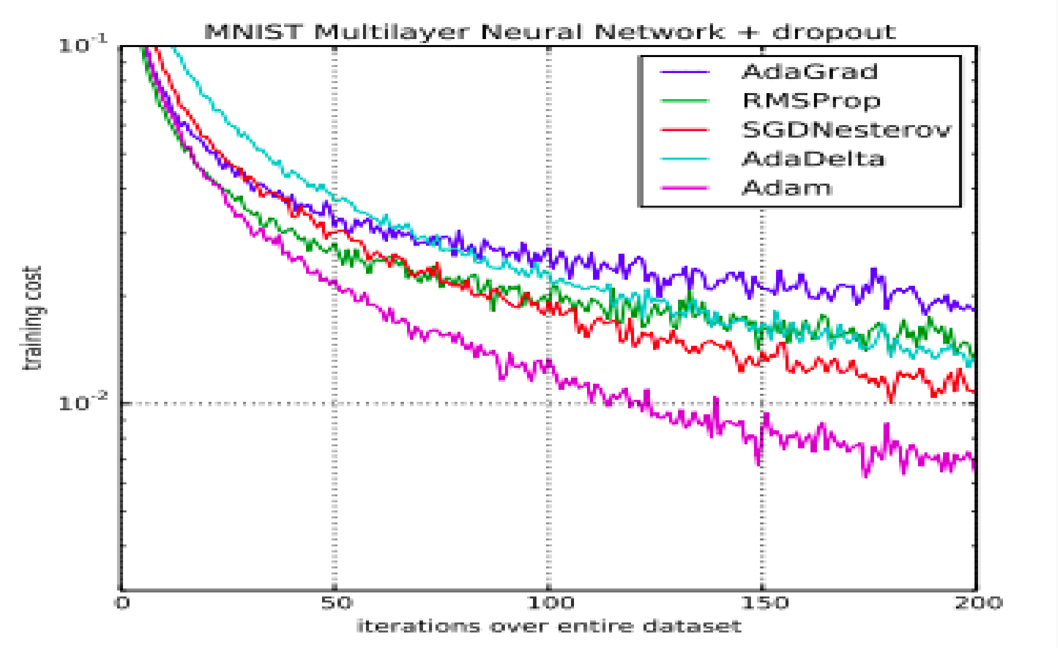
The hyperparameters have been explained below:

i) Sparse Categorical Cross Entropy Loss: It is the sum of the products of each of the classes and logarithmic value of the class as given in the below equation

$$\text{Loss} = -\frac{1}{\substack{\text{output}\\\text{size}}} \sum_{i=1}^{\substack{\text{output}\\\text{size}}} y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log\,(1 - \hat{y}_i) \tag{4}$$

ii) Adam Optimizer:

Adam optimizer is one of the most used and most effective optimizers to be used in a problem of neural networks. The comparison of the performance of the Adam optimizer with the other existing optimizers has been given below in **Fig 18.**



Comparison of Adam to Other Optimization Algorithms Training a Multilayer Perceptron
Taken from Adam: A Method for Stochastic Optimization, 2015.

Fig 18.The comparison of the training cost of all the existing optimizers

As can be inferred from the image the Adam optimizer performs the best and thus it was chosen as the primary optimizer.

7.2. Training Results

SL No.	Model	Training Accuracy	Test Accuracy
1.	Classifier	71.34%	64.22%
2.	Encoder Decoder	84.51%	81.23%
3.	Attention Model	93.75%	88.91%

TABLE I. Training Result

From **TABLE I** it can be observed that the attention model reaches the highest accuracy among the 3 models implemented. Thus we did our predictions using the Attention model and deployed it on our website.

The loss for the attention model has been given below in **Fig 19.**

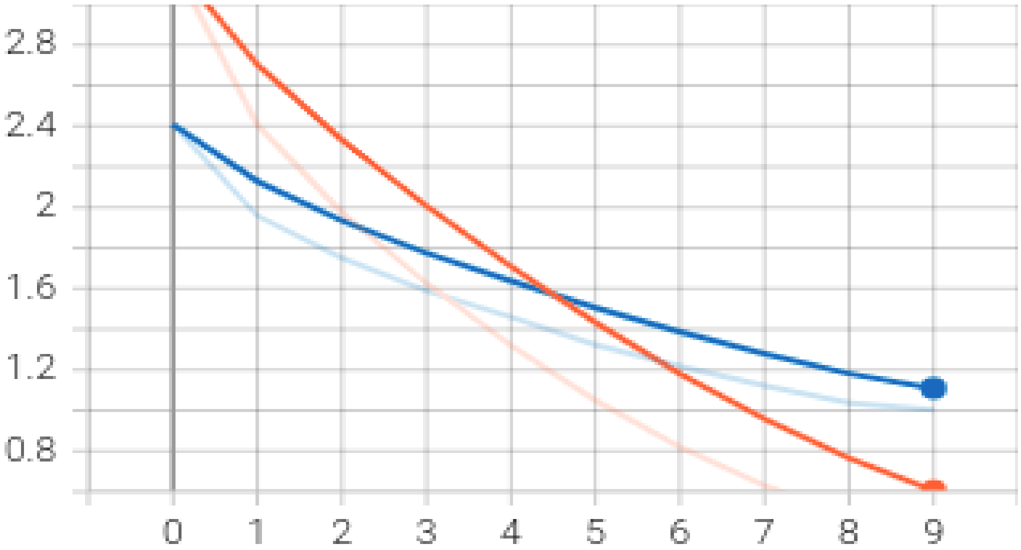


Fig 19. Loss Value with respect to number of epoch

The orange line denotes the loss value on the training data with every epoch while the blue line denotes the loss value on the validation data with every epoch.

Moreover we also judged the performances of the models using the BiLingual Evaluation Underscoring(BLEU)[5] method on the test data.

The BLEU score comparison has been given below in **TABLE II**.

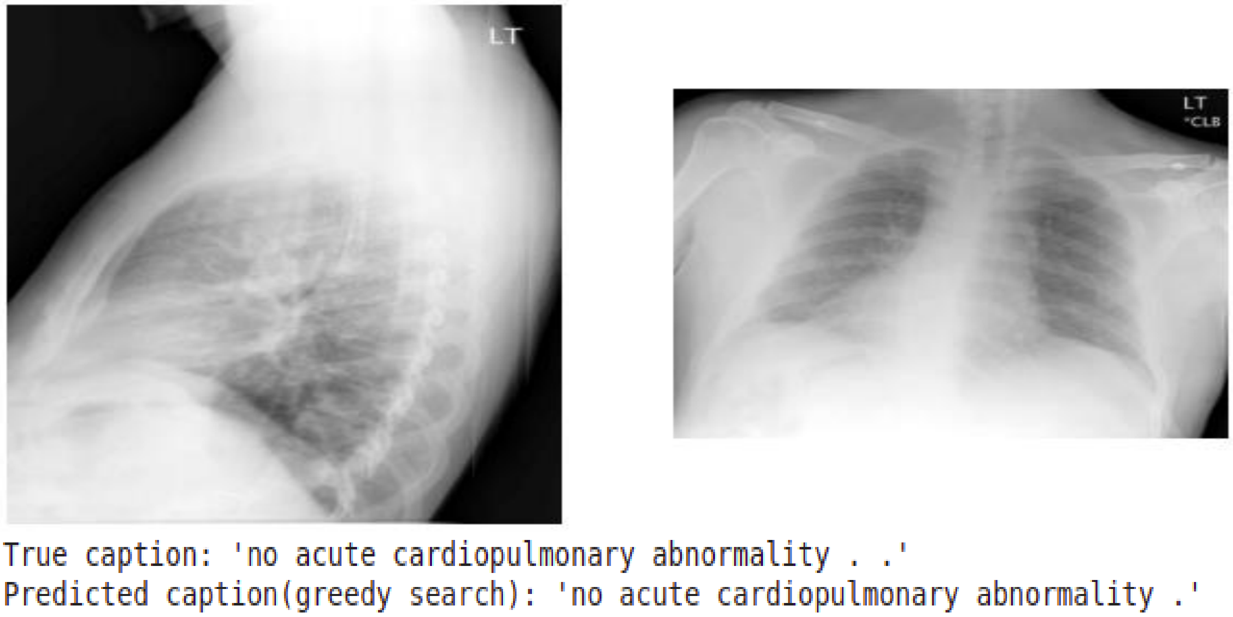
SL No.	Model	Test Accuracy	BLEU-4
1.	Classifier	71.34%	0.3667
2.	Encoder Decoder	84.51%	0.3652
3.	Attention Model	93.75%	0.3871

TABLE II BLEU Score Comparison

From **TABLE II ,** also it can be seen that the attention model performs the best among the three.

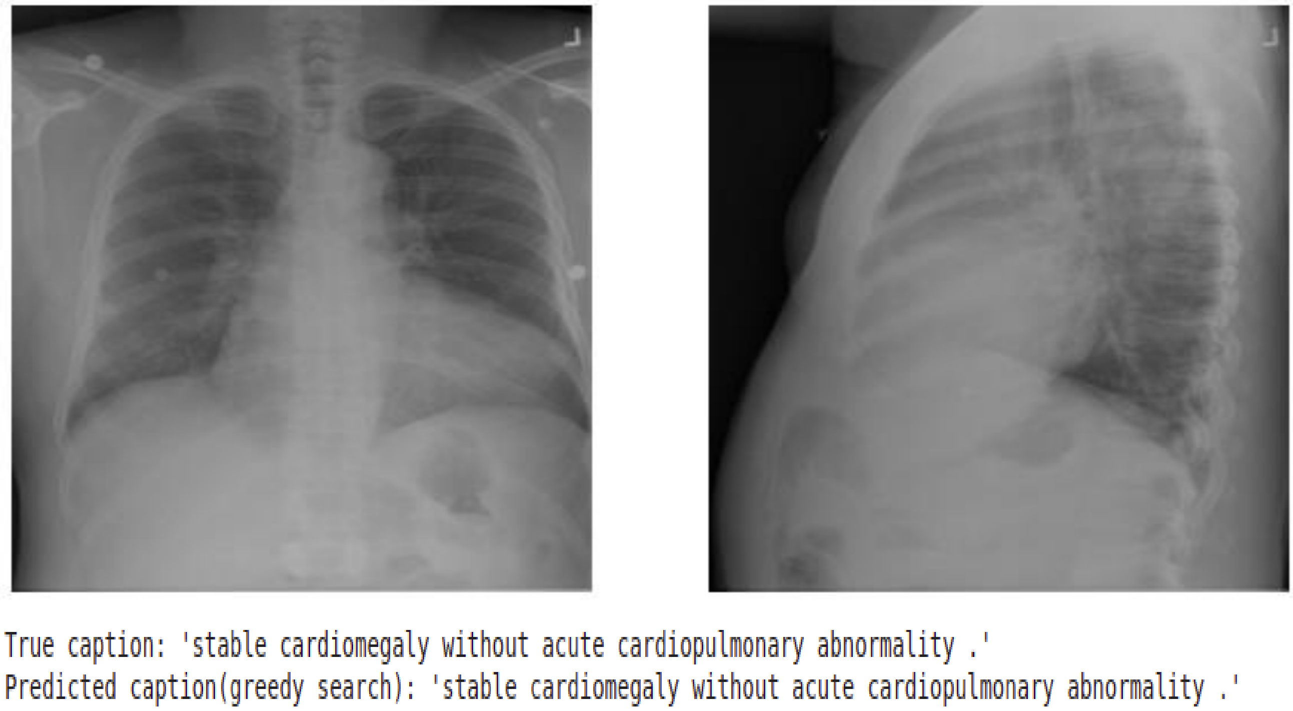
8. Results

As from the previous section we got to know that the best results were achieved by the attention model, we did our predictions and showed them below using it. For the predictions the greedy search method was used, i.e for every word there were 112 predictions with different probabilities of each of the words. From that word having the maximum likelihood estimate(MLE) was chosen. Beam Search was also applied but both of them gave almost the same BLEU score so greedy search was preferred as it was less complex to implement.
The results on some of the test images are given below and also the true and predicted reports are shown.



```
True caption: 'no acute cardiopulmonary abnormality . .'
Predicted caption(greedy search): 'no acute cardiopulmonary abnormality .'
```

Fig 20. Sample Result I

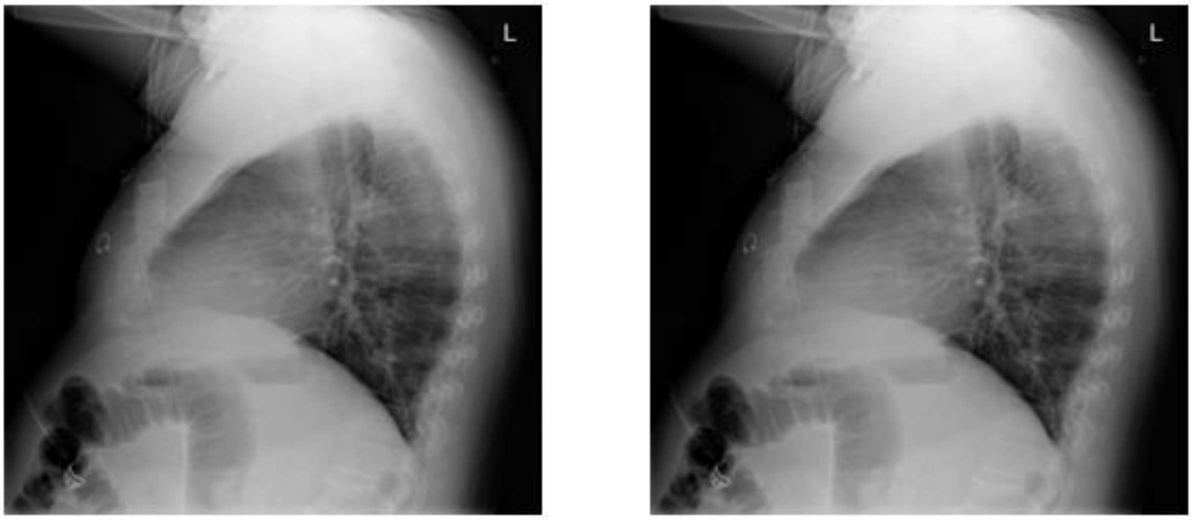


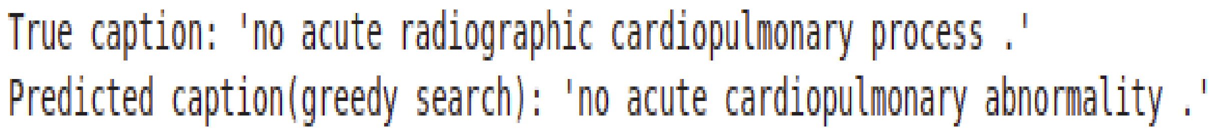
```
True caption: 'stable cardiomegaly without acute cardiopulmonary abnormality .'
Predicted caption(greedy search): 'stable cardiomegaly without acute cardiopulmonary abnormality .'
```

Fig 21. Sample Result II

9. Handling a case of Outlier

The below results(**Fig 22.**) show the prediction on an image that is an outlier since both the images associated with the report are the same. Even though this case was an outlier the model performed fairly well and predicted almost exact findings.



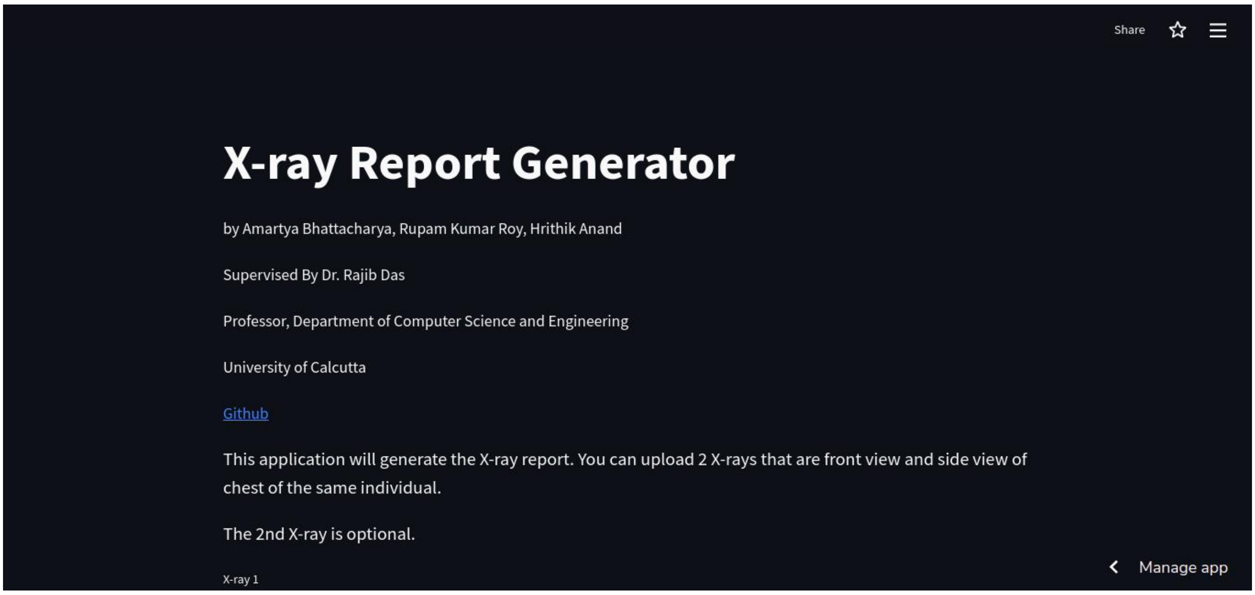


```
True caption: 'no acute radiographic cardiopulmonary process .'
Predicted caption(greedy search): 'no acute cardiopulmonary abnormality .'
```

Fig 22. Sample Result III

10. Deployment

The whole model was then deployed to a server using Streamlit. The model was capable of generating results using one image only. As we can see from the case of the outlier both of the images are same and then also it was predicting similar reports. We used this feature, i.e when only one image is given we replicate the image and pass it as the second image and give our predictions.
The whole model along with the website link is given in the github link: Github.The requirements needed for the project to run has been given in the requirements.txt file and the operation has been explained in the form of a '.gif' file. The screenshots of which have been attached and shown below in **Fig 23.** and **Fig 24.**



X-ray Report Generator

by Amartya Bhattacharya, Rupam Kumar Roy, Hrithik Anand

Supervised By Dr. Rajib Das

Professor, Department of Computer Science and Engineering

University of Calcutta

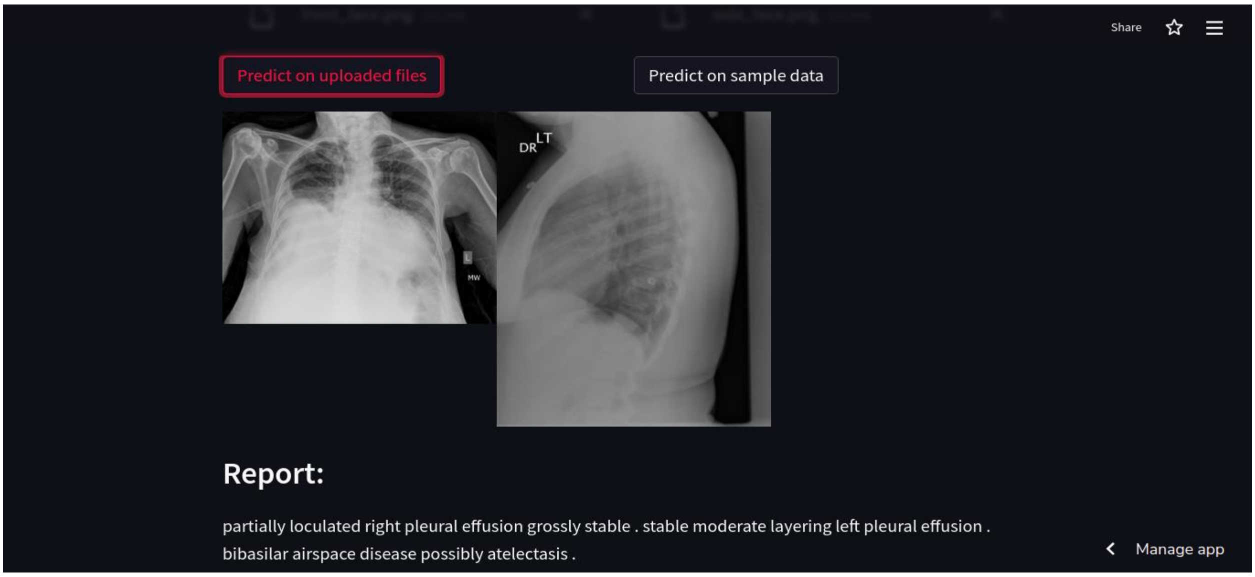
Github

This application will generate the X-ray report. You can upload 2 X-rays that are front view and side view of chest of the same individual.

The 2nd X-ray is optional.

X-ray 1

Fig 23. Website Frontend



Report:

partially loculated right pleural effusion grossly stable . stable moderate layering left pleural effusion . bibasilar airspace disease possibly atelectasis .

Fig 24. Result obtained from website

From the above figures we can see how the application operates using two images.

11.Conclusion and Future Works

In this project we tried to build an automatic Medical Report Generator with the help of a neural-network based model. We created an attention based model that would take features from images and generate the reports for the given input images. This whole work aims to decrease the time for report generation in case of emergency situations like the recent COVID-19. Our model achieved an accuracy of almost 89% with the help of a convolutional neural network based architecture for feature extraction and LSTM based network for report generation along with the inbuilt attention layers. Although the accuracy achieved is fairly good, better accuracies can be obtained by using the word feature embeddings using latest state-of-the-art transformer based models which was not used in our case due to hardware constraints.

12. Hardware Specifications

The whole project was done on a system with Intel I3 2.6 Ghz processor, 4GB RAM and 64GB hard disk.

13. Reference

1. Irvin, Jeremy, et al. "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison." Proceedings of the AAAI conference on artificial intelligence. Vol. 33. No. 01. 2019.
2. Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
3. https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data
4. https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy
5. Papineni, Kishore & Roukos, Salim & Ward, Todd & Zhu, Wei Jing. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. 10.3115/1073083.1073135.
6. Antol, Stanislaw, et al. "Vqa: Visual question answering." Proceedings of the IEEE international conference on computer vision. 2015.
7. Allaouzi, Imane, Mohamed Ben Ahmed, and Badr Benamrou. "An Encoder-Decoder Model for Visual Question Answering in the Medical Domain." CLEF (Working Notes). 2019.
8. Sharma, Dhruv, Sanjay Purushotham, and Chandan K. Reddy. "MedFuseNet: An attention-based multimodal deep learning model for visual question answering in the medical domain." Scientific Reports 11.1 (2021): 1-18.
9. https://colah.github.io/posts/2015-08-Understanding-LSTMs/