**Second Review Document**

**MEDICAL REPORT GENERATION ON CHEST X-RAY IMAGES**

R.S. Rahul Sai

17BCE0136

9131038763

rs.rahulsai2017@vitstudent.ac.in

Prof. Sharmila Banu K (11989)

Associate Professor Grade 1

9942469321

sharmilabanu.k@vit.ac.in

**B.Tech.**

in

**Computer Science and Engineering**

**School of Computer Science & Engineering**



1. **Introduction**
   1. **Theoretical Background**

Computer Aided Diagnosis (CAD) and medical imaging systems have evolved in the past decade to a point where they can partially mimic radiologists and doctors. These systems can learn and differentiate the features and abnormalities in medical images, and provide objective evidence with a higher diagnostic confidence and faster inference. In this project, we focus on generating detailed medical reports on chest X-ray images, which can be adapted later to work with other diagnostic tools such as ultrasounds and mammograms. The Indiana University dataset provides us with CXR images corresponding to various lung and heart ailments, along with well-defined reports and findings. The generation of medical reports mainly consists of two broad tasks. The first task is to treat the problem as a multi-label classification task to obtain the accurate tags for a particular image from the visual features. This is performed using Convolutional Neural Networks and a transfer learning framework called ChexNet, which is specialized for chest X-ray images. The second task is to generate the reports using these aforementioned tags, which requires the use of recurrent neural networks such as hierarchical LSTMs. To improve the quality of the sentences produced, a co-attention mechanism is also required, which makes use of the spatial information from the convolutional layers and the generated words in order to find localized regions from which the abnormalities are found.

* 1. **Motivation**

The use of deep learning in image captioning has been a popular use case over the past few years. It is a challenging problem as it requires the machine to generate textual description from the contents of an image, similar to how a human brain would describe an image. But consider the same scenario for medical images, with the example of Chest X-ray images. For a normal human eye, chest X-ray images are just images with the skeletal and muscular features of the lungs defined in black and white. But highly trained radiologists who have studied and diagnosed various respiratory and cardiovascular abnormalities, can mark multiple areas of the images and can write down clear reports for potential abnormalities. However, to read a chest X-ray image properly, it is important that the radiologist has a thorough knowledge of the human thorax, and how various respiratory diseases might affect them. This comes with multiple years of experience by analyzing chest x-ray images with a fixed pattern of evaluation, and evolves over time depending on the history of cases a particular radiologist may handle. But, even for highly trained and experienced radiologists, writing reports is highly time-consuming especially in regions with higher population density. Looking at the other side of the spectrum, radiologists or pathologists in rural areas, with inferior imaging equipment face a similar issue. They either cannot get objective evidence to diagnose anything properly or lack the knowledge of the respiratory or cardiovascular abnormalities. Misdiagnosis of symptoms and medical errors are the third-leading cause of deaths across the world, leading to more than 3 million deaths. So, this project focuses on generating detailed medical reports using Chest X-ray images, which can facilitate the diagnosis of various respiratory and cardiovascular diseases.

* 1. **Aim of the proposed Work**

To generate detailed medical reports with three heterogenous sections i.e., Findings, Impressions and Indications from IU Chest X-Ray images, highlighting the context of the particular disease i.e., location, severity and affected organs, using a combination of CNN, hierarchical LSTMs and co-attention mechanism.

* 1. **Objective(s) of the proposed work**

1. Preparing an NLP Pipeline for the original findings, impressions and indication
2. Removing Contractions, Punctuations and Numbers
3. Tokenization
4. Representing the reports in word embeddings
5. Obtain the image features using a Convolutional Neural Network and the Transfer Learning framework i.e., ChexNet (acts as the encoder here).
6. Generate the text using the labels/tags obtained from the encoder using Hierarchical LSTMs which act as the decoder in our Sequence-to-Sequence Model.
7. Substituting Attention mechanisms to improve the encoder-decoder approach and compare the results.
8. Improving the results by utilizing newer NLP techniques such as BERT Embeddings and Transformers and generate better reports.
9. **Literature Survey**
   1. **Survey of the Existing Models/Work**

Survey

* 1. **Summary/Gaps identified in the Survey**

Summary

1. **Overview of the Proposed System**
   1. **Introduction and Related Concepts**

A medical report can usually consist of various parts, which are usually heterogenous in nature. There may be images, abbreviations and complicated terminology in these reports. To avoid this issue, we will focus on three sections from a medical report i.e., findings – a large paragraph which contains keywords and parameters, indication – the doctor’s advice for the patient and impression – a sentence finalizing the results for a report. To achieve this, we propose a multi-task architecture which works by predicting the keywords/tags (findings) by treating it as a multi-label image classification task and generate longer descriptions by using a text generating mechanism such as hierarchical LSTMs. Hierarchical LSTMs **[1]** are specialized recurrent neural networks which are often used for text generation from images and video frames. It is built to consider both high-level language features from the training text and low-level visual features obtained from the processed image.

The dataset to be used in this project is the Indiana University Chest X-Ray (CXR) Image dataset. It is a high resolution CXR dataset with multiple views i.e., frontal, side and posterior views. There are 7,470 images accompanying 3,955 well written reports encoded in XML. These XML reports have references to the CXR images, the findings, impressions and the indication from the CXR images. These CXR images are obtained from patients diagnosed with tuberculosis, pneumonia and various heart ailments.

* 1. **Architecture for the Proposed System**

The architecture of the proposed medical report generation system can be divided into three distinct parts:

1. **NLP Pipeline** – The findings, impressions and the indications obtained from the reports have to be properly cleaned to be used in the model. This involves the following steps:
   1. Converting all characters into lowercase
   2. Removing contractions from the text e.g., won’t – will not, can’t – cannot.
   3. Removing punctuation from text with the exception of full stop, as the findings from the reports may contain more than one sentences.
   4. Removing all numbers and redacted data from the text.
   5. Removing smaller words and adverbs with the exception of “no” as it adds significant value. e.g., adverbs such as “there”, “then”.
   6. Tokenization and addition of identifier tokens such as “\_start” and “\_end” tokens which are necessary in the text generation process.

**Examples –**

Findings before the cleaning process:

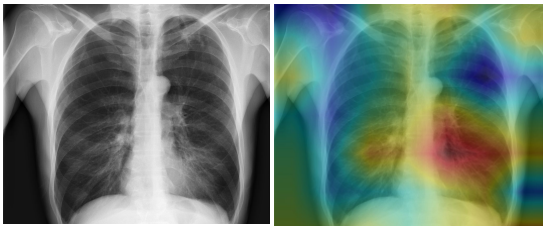
No pleural effusion or pneumothorax. No acute bone abnormality.

Findings after the cleaning process:

***\_start*** no pleural effusion no pneumothorax. no acute bone abnormality. \_***end***

1. **Convolutional Neural Networks –** In order to extract the visual features of the image, we make use of convolutional neural networks. The CNN acts as the encoder in our model, and provides us a feature vector with the visual features of the image, which can be used to predict the tags for that particular X-ray. This is done by considering the problem as a multi-label classification task where the output layer provides you with the probability of a set number of tags. These predicted tags help us massively in text generation which is discussed in the next section. However, the size of our dataset (7,470 images) is not enough to train a CNN properly. To remove this bottleneck, we have to use a transfer learning framework. Most transfer learning frameworks such as VGG16 or InceptionV3 are trained over generic image datasets which doesn’t serve our purpose.

Fortunately, ChexNet is a convolutional neural network especially trained on Chest X-ray images. It was trained over 1,12,120 images and contains 121 layers where the input is a chest X-ray image, and the output is the probability of 14 different diseases along with a localized heatmap which highlights the visual features of the chest x-ray image. However, we do not need to classify the image into one of those 14 categories, so we can remove the final classification layer. From a image of dimensions (224,224,3), we get a feature vector with a length of 1,024. We have two images associated with a report, so we concatenate the two feature vectors to get a feature vector with length 2,048. This final feature vector will be passed along with the report to the decoder which is discussed in the next section.

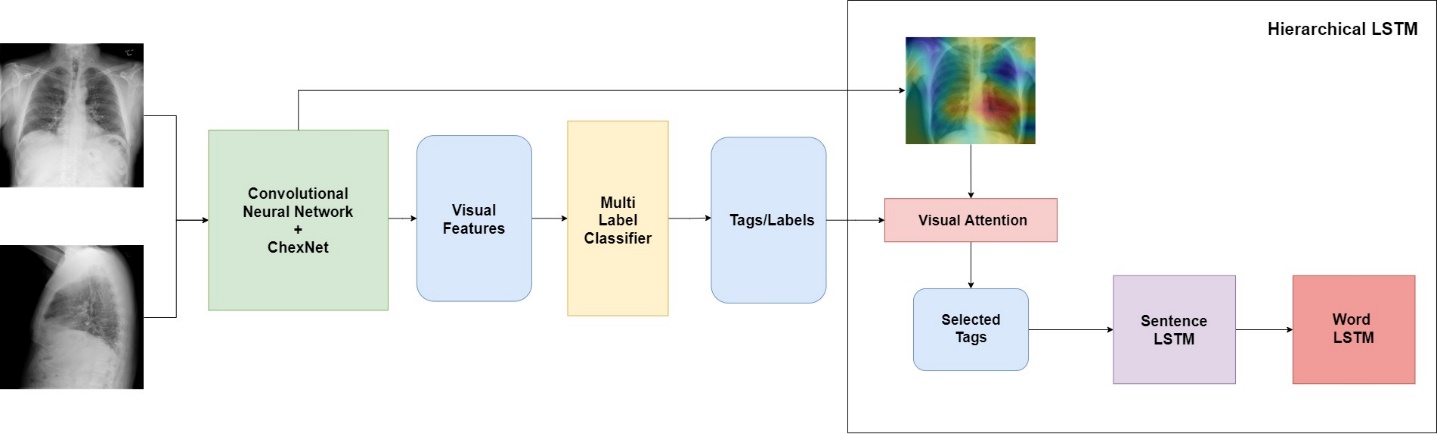
****

**Figure 2: Chest X-ray image from the dataset before and after passing through the CNN**

1. **Hierarchical LSTMs and Attention Mechanism –** Hierarchical LSTMs [1] are specialized recurrent neural networks which are often used for text generation from images and video frames. It is built to consider both high-level language features from the training text and low-level visual features obtained from the processed image. Note that the keywords/tags obtained from the image are generated by the fully connected layer, which results in a loss of spatial information. To improve these results, an additional mechanism known as Co-Attention is added. Co-Attention mechanism uses the spatial information from the visual features of the convolutional layers and the semantic features obtained from the tags of the specific image (which are generated by the fully connected layer).

When the visual features and the tags arrive at the decoder, the high-level spatial information provided by the localized heatmap help us to focus on the tags which are visually highlighted more, and yield better results. This new context vector with the embeddings of the selected tags is passed on to a sentence LSTM. A Sentence LSTM will generate multiple sentences as suggestions to the provided words, using a technique known as beam search which predicts the probability distribution across the given vocabulary and returns the words which have the maximum probability to be subsequent words in the sentence. Beam search selects multiple alternatives for an input sequence, based on conditional probability and a parameter known as beam width. When the sentence vector is successfully produced, it is passed on as a context vector to the Word LSTM. Word LSTM employs a greedy search mechanism, which selects a single candidate which is suitable for the input sequence in a time step. This improves the quality of the final sentence generated.

* 1. **Proposed System Model (ER Diagram/UML Diagram/Mathematical Modeling)**



1. **Proposed System Analysis and Design**
   1. **Introduction**
   2. **Requirement Analysis** 
      1. **Functional Requirements**
         1. **Product Perspective**

A medical report can usually consist of various parts, which are usually heterogenous in nature. There may be images, abbreviations and complicated terminology in these reports. To avoid this issue, we will focus on three sections from a medical report i.e., findings – a large paragraph which contains keywords and parameters, indication – the doctor’s advice for the patient and impression – a sentence finalizing the results for a report. To achieve this, this application uses a multi-task architecture which works by predicting the keywords/tags (findings) by treating it as a multi-label image classification task and generate longer descriptions by using a text generating mechanism such as hierarchical LSTMs.

* + - 1. **Product features**

The application provides us a comprehensive medical report with three heterogenous sections i.e., Findings, Impressions and Indications from Chest X-Ray images, highlighting the context of the particular disease i.e., location, severity and affected organs, using a combination of CNN, hierarchical LSTMs and co-attention mechanism.

* + - 1. **User characteristics**

The intended audience for this application consists mostly of medical professionals and radiologists. The users of the application should be aware of the all the different functions of the application and the proper procedure to use the application.

* + - 1. **Assumption & Dependencies**

It is assumed that the dataset and the associated reports being used to implement the application is publicly available with all the required legal permissions. The images and reports have been thoroughly checked in order to redact any personal patient data in order to maintain their privacy. The application’s accuracy partially depends on the provided data (both training and testing) as well as the data provided by the intended users.

* + - 1. **Domain Requirements**

Domain Requirements

* + - 1. **User Requirements**

User Requirements

* + 1. **Non-Functional Requirements**
       1. **Product Requirements**
          1. **Efficiency**

The application should be computationally efficient and the cost of working on larger datasets should be proportional. The efficiency can be greatly improved by parallelizing the training and testing of the deep learning models as they are the ones which require the most amount of computational time and resources.

* + - * 1. **Reliability**

Due to the sensitive and critical nature of medical expert systems (especially involving diagnostics), it is necessary that the application is highly reliable and accurate. The complete application can be deployed as a web application on highly available servers (like AWS or GCP) to be accessed from anywhere. An alternative could also include packaging this as a standalone application for offline support which might be beneficial in rural areas where connectivity might be an issue.

* + - * 1. **Portability**

Machine Learning models can be easily deployed online using micro-web frameworks such as Flask and FastAPI which are intended to be used on Python. TensorFlow along with Keras allows us to save the model weights in order to quickly model predictions on the given data. TensorFlow also allows us to convert these models in order to achieve compatibility with Android/iOS mobile applications and JavaScript. So, it’s entirely possible to deploy this application as a web application making it extremely portable, with the ability to run on multiple platforms.

* + - * 1. **Usability**

With a well-prepared documentation, FAQs and tutorials, the application will be usable and convenient for every intended user. Future revisions can include multilingual support in order to improve usability among a wide variety of users. However, the multilingual support for the medical reports would require a highly accurate language translator or an accurate dataset with the reports in the language to be supported. This is due to the fact that medical reports are highly critical and sensitive in nature.

* + - * 1. **Correctness**

The correctness of a medical expert system is highly relevant as the results can be essential.

* + - 1. **Organizational Requirements**
         1. **Implementation Requirements (in terms of deployment)**
         2. **Engineering Standard Requirements**
      2. **Operational Requirements**
* **Economic –** Theapplication should be economically viable in terms of modelling predictions and should be cheap and efficient when the application needs to be scaled.
* **Social –** The application should be socially responsible in terms of the data and the predictions generated from it. The application and the necessary data should be publicly accessible.
* **Political –** The application should be unbiased and free of any political propaganda.
* **Ethical –** The application should respect the personal privacy of the users and provide unbiased results irrespective of race or gender.
* **Health and Safety –** Due to the application being an important part in the medical system, it is important that health and safety regulations are strictly followed, which may require validation from regulatory authorities. The regulatory checks will be to check the viability and the accuracy of the predicted results due to the sensitive and critical nature of the task at hand.
* **Sustainability –** The application should be easy to update to the latest versions in order to provide the intended features without any interruption to the users of the applications.
* **Legality –** The datasets used to train the model should be open-source or publicly available along with the guarantee that the provided information in the images and reports are correct and validated by proper authorities, given that medical reports are highly critical. Any personal data in the datasets as well as the application should be redacted or encrypted as per requirements.
* **Inspectability –** Proper measures should be taken in order to facilitate any debugging in case of any errors, along with a flexible and modular design architecture which can be inspected and validated by independent as well as government authorities.
  + 1. **System Requirements**
       1. **H/W Requirements**

The model was trained over a Colab Pro instance, which has the following specifications. The processor is a server-grade Intel Xeon processor with support for multithreading which helps with the initial preprocessing and streaming of the data during the training phase. The GPU is one of the most essential components when training deep learning models as it allows us to distribute the massive amount of mathematical computation amongst many cores. The Nvidia Tesla P100 is the world’s most advanced GPU for HPC and Deep Learning workloads. It has a 16GB of HBM2 memory along with 3584 CUDA cores. The Colab instance also has 28 GB of usable RAM and disk storage of 150GB which provides us plenty of memory in order to retain the model data, weights and various other information.

|  |  |
| --- | --- |
| CPU | Intel Xeon (Skylake) CPU 2/4 @ 2.20 Ghz |
| GPU | Nvidia Tesla P100 16GB |
| RAM | 28 GB |
| Disk Storage | 150 GB |

* + - 1. **S/W Requirements (details about Application Specific Software)**

The software requirements including the programming languages, libraries/frameworks and dependencies are listed below.

1. **Results and Discussion**
2. **References**