

**MULTIMODAL LEARNING: GENERATING PRECISE CHEST X-RAY REPORT ON
THORAX ABNORMALITY**

By

Gaurab Subedi

BE, Tribhuvan University, Nepal, 2019

A Thesis Submitted in Partial Fulfillment of
the Requirements for the Degree of Master of Science

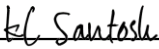
Department of Computer Science

Master Of Science Program

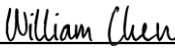
In the Graduate School
The University of South Dakota
December 2023

Copyright By:
GAURAB SUBEDI
2023
All Rights Reserved

The members of the Committee appointed to examine
the Thesis of Gaurab Subedi
find it satisfactory and recommend that it be accepted.

DocuSigned by:

99944614C924467...

Chairperson

DocuSigned by:

9671140C01924FB...

DocuSigned by:

056A76355F814ED...

ABSTRACT

Chronic respiratory diseases, ranking as the third leading cause of death worldwide according to the 2017 World Health Organization (WHO) report, affect a staggering 544.9 million individuals. Compounding this public health challenge is the fact that over 80% of health systems grapple with shortages in their radiology departments, highlighting an urgent need for accessible and efficient diagnostic solutions. While various image classification models for analyzing thorax abnormalities have been developed, relying solely on one type of dataset (image data, for example) for thorax abnormality analysis is insufficient. Integrating texts with image data could provide more accuracy as well as analysis. In response to this challenge, we propose a multimodal approach to generate detailed radiology reports from chest X-ray images and their corresponding radiological reports (Impression and Findings). Our framework integrates a pre-trained Convolutional Neural Network (CNN) for robust image feature extraction, a Recurrent Neural Network (RNN), and a visual attention mechanism to ensure coherent sentence generation. The image encoder employs the ResNet152 architecture to extract nuanced visual features from chest X-ray images. Simultaneously, the sentence generation model utilizes a Long Short-Term Memory (LSTM) layer to process textual data and generate contextually relevant reports. On an IU dataset of 7470 pairs of X-ray images and 3995 reports, our model exhibited superior performance based on language generation metrics (BLEU1= 0.4424, BLEU2= 0.2923, BLEU3= 0.207, BLEU4= 0.1464, ROUGE= 0.3396, and CIDEr= 0.2268), providing accurate and coherent impressions and findings compared to other benchmark models. For a reproducibility purpose, the implementation code is available: <https://github.com/2ai-lab/Report-Generation>

DocuSigned by:

1D1EB20650034B9...

Thesis Advisor: _____

Dr. KC Santosh

ACKNOWLEDGEMENTS

I extend my deepest gratitude to my esteemed thesis advisor, Dr. KC Santosh. His expert guidance, unwavering support, and insightful feedback were indispensable throughout the journey of this thesis. Without his mentorship, this work would not have come to fruition.

I also wish to express my sincere appreciation to the distinguished members of my thesis committee—Dr. William Chen and Dr. Rodrigue Rizk. Their active involvement, scholarly input, and constructive critiques significantly enriched the quality of this research.

To the entire 2AI lab 2022 - 2023 team, I am profoundly thankful for your valuable assistance and unwavering support during the thesis writing process. Your collective contributions have been indispensable, and I am truly grateful for the collaborative spirit that defined this academic endeavor.

This work stands as a testament to the collaborative effort and support of these exceptional individuals and the 2AI lab team. Thank you for being instrumental in the realization of this academic milestone.

DEDICATION

To all who believe in me,

To my unwavering pillars of support – my parents, whose boundless love and encouragement have been my guiding light throughout this academic journey. Your sacrifices and belief in my potential have fueled my determination.

To my friends, whose camaraderie and shared moments of both joy and struggle have made this academic endeavor not only bearable but truly memorable.

And to my esteemed professor, whose mentorship and guidance have shaped my intellectual growth. Your dedication to fostering knowledge has left an indelible mark on my academic pursuits.

This thesis stands as a testament to the collective belief and support of those who have stood by me. Thank you for being the foundation upon which this achievement rests.

Table of Contents

Committee Signature Page.....	i
ABSTRACT.....	ii
ACKNOWLEDGEMENTS.....	iii
DEDICATION.....	iv
LIST OF TABLES.....	vii
LIST OF ILLUSTRATIONS.....	viii
1. Introduction.....	1
1.1. Context and problem.....	1
1.2. Goal.....	2
1.3. Research hypothesis and contributions.....	2
1.4. Thesis outline.....	3
1.5. Summary.....	4
2. Related works.....	5
2.1. State-of-the-art literature.....	6
2.2. Evaluation metrics.....	6
2.2.1. BLEU.....	7
2.2.2. ROUGE.....	8
2.2.3. CIDEr.....	8
2.3. Summary.....	10
3. Dataset.....	11
3.1. Dataset collection.....	11
3.2. Dataset description.....	11
3.3. Summary.....	12
4. Deep learning, encoder-decoder architecture, visual attention.....	13
4.1. Deep learning.....	13
4.1.1. Perceptron.....	13
4.1.2. Activation function.....	14
4.1.3. Convolutional neural network.....	20
4.1.4. Recurrent neural network.....	22
4.2. Encoder-decoder architecture.....	24
4.3. Visual attention.....	25

4.3.1.	Visual feature	27
4.4.	Summary	28
5.	Multimodal learning and implementation	29
5.1.	Multimodal learning	29
5.1.1.	Basics	29
5.1.2.	Data types.....	30
5.1.3.	Application.....	30
5.1.4.	Challenges.....	31
5.2.	Implementation.....	31
5.2.1.	Image encoder.....	32
5.2.2.	Sentence and paragraph generation model	32
5.2.3.	Sentence encoder	34
5.2.4.	Sentence decoder	34
5.3.	Summary	35
6.	Experiments	37
6.1.	Setup.....	37
6.2.	Results	38
6.3.	Comparison	38
6.4.	Discussion	40
6.5.	Summary	40
7.	Conclusion	41
	References.....	42

LIST OF TABLES

Table 1 Evaluation metrics for Impression, Finding and Impression and finding.....	38
Table 2 Evaluation of generated reports on our testing set using BLEU, ROUGE, and CIDEr NLG metrics. We compare our models with five baseline models including a baseline implementation of the hierarchical generation reinforcement model.	39

LIST OF ILLUSTRATIONS

Figure 1 NLG Evaluation System.....	7
Figure 2 Sample chest radiograph dataset with its respective findings and impressions	12
Figure 3 Perceptron.....	14
Figure 4 Linear function plot	15
Figure 5 Nonlinear function plot.....	16
Figure 6 Sigmoid function plot.....	16
Figure 7 Tanh function plot	17
Figure 8 ReLU function plot.....	18
Figure 9 ELU function plot.....	19
Figure 10 Convolutional neural network	20
Figure 11 ResNet152 Architecture	22
Figure 12 Standard RNN network	23
Figure 13 LSTM architecture.....	23
Figure 14 Encoder decoder architecture	24
Figure 15 Example showing attention. (Source: https://theaisummer.com/attention/).....	26
Figure 16 Visual features	28
Figure 17 Multimodal theoretical architecture	29
Figure 18 Example of different modalities	30
Figure 19 Schema of proposed model	31
Figure 20 Sample output with ground truth and predicted output	39
Figure 21 Sample output generated in txt file.....	40

CHAPTER 1

1. Introduction

1.1. Context and problem

Pulmonary abnormalities [1] [2] encompass a range of disruptions in the natural lung function, often stemming from various lung diseases like Pneumonia [3], Chronic Obstructive Pulmonary Disease (COPD), Atelectasis, Effusion, Pneumothorax, Cardiomegaly, and others. Notably, the year 2017 saw a staggering 544.9 million individuals worldwide grappling with chronic respiratory diseases, marking them as the third leading cause of death for that year, as meticulously documented by the World Health Organization (WHO) [4]. Pneumonia alone accounted for the loss of around 808,000 lives in 2017, with children under five years old tragically contributing to 15% of these fatalities. While the mortality rate among the elderly has exhibited consistency since 1990, tuberculosis (TB) casts a shadow over 10 million individuals (about half the population of New York) and resulted in 1.4 million fatalities in 2019. These compelling statistics poignantly underscore the imperative need to focus on understanding pulmonary disorders and abnormalities [5], a sentiment echoed in.

To diagnose such conditions, it's universally acknowledged that thorough analysis of medical images such as ultrasound, X-ray, Magnetic Resonance Imaging (MRI), Computed Tomography (CT), or pathological imaging is quintessential. Many deep learning models used only images data to analyze pulmonary diseases [6] including COVID-19 [7] [8] [9]. Although it is analyzed with medical images it needs to be generalized and presented in the form of a report for better application. This demanding task calls for the expertise of skilled physicians, and radiologists who meticulously compose diagnostic reports for single patients. An example of one of these reports is provided in Figure 2. Even while a single medical report may appear simple, a considerable proportion of individuals come up with unexpectedly complex and aberrant medical images [10]. Consequently, analyzing and articulating textual reports, tasks that demand seasoned expertise, can exact substantial time and stress from professionals. Within this context, the emergence of automated diagnostic report generation with the help of medical images stands as an imperative trend, positioned to alleviate this burdensome challenge [11].

The trend toward automated diagnostic medical report generation from medical images is becoming increasingly indispensable to alleviate this workload. To achieve this, researchers have harnessed a diverse range of deep learning (DL) algorithms. These advanced approaches have effectively facilitated the generation of reports for lung diseases, thereby significantly reducing the

time required for prognosis. Additionally, the widespread implementation of various DL techniques using an encoder and decoder architecture which was used for machine translation before has further extended the capacity to generate reports from medical images, demonstrating the capabilities of cutting-edge techniques in this domain [12]. While impression can be generated by existing image captioning models that describe an image with a sentence [13].

These existing models' RNNs can't handle extended paragraphs or sequences because of their exploding or vanishing gradients [14]. With a gating mechanism to learn long-term dependencies, Long Short-Term Memory (LSTM) helps to some extent to ease this problem, but it is still unable to fully prevent gradient vanishing and presents challenges when modeling long-term sequences. In the field of natural image captioning, hierarchical recurrent networks have been the subject of some groundbreaking work to provide a paragraph description or extended sequence [15]. For paragraph generation, they often employ two layers of RNNs: a paragraph-level RNN creates certain topics first, and a sentence-level RNN uses the topics as input to produce related sentences.

1.2. Goal

Overall, the main objective of the thesis is to generate multi-sentence textual reports describing abnormalities in detail comparable to human-authored reports, as evaluated by clinical radiologists through quantitative metrics and qualitative assessments. We designed an end-to-end neural network model architecture that effectively represents both visual image data from chest x-rays and sequentially generated textual findings integrating convolutional image encoding with recurrent text decoding components where the first sentence originates, and each subsequent sentence is generated using the input image's encoding as well as the previous sentences.

1.3. Research hypothesis and contributions

The integration of deep neural network models, featuring a Convolutional Neural Network (CNN) for image encoding, a specialized Recurrent Neural Network (RNN) for initial sentence generation, a convolutional encoder for sentence representation, and a recurrent decoder with attention for subsequent sentence generation, will enable the effective generation of radiology reports from chest X-ray images. Leveraging transfer learning with a pre-trained ResNet-152 model and training end-to-end on a chest X-ray dataset, the proposed model will demonstrate a balance between visual evidence representation and textual description. The hypothesis posits that the developed deep learning techniques will yield clinically accurate and utility-enhanced radiology reports, as evidenced by quantitative metrics (BLEU, ROUGE, CIDEr) and validated through human evaluations by clinical radiologists.

In this thesis, we developed deep neural network models to generate radiology findings from chest x-ray images. The methods consist of four key components image encoder, initial sentence generator, sentence encoder and decoder. Image encoder is a Convolutional Neural Network (CNN) which encodes chest x-rays to extract visual features where transfer learning will utilize a pre-trained ResNet-152 model initialized with weights learned on ImageNet. This CNN will encode both local and global image features to represent visual evidence. Initial sentence generator is a specialized Recurrent Neural Network (RNN) module which generates the first impression

sentence based solely on the global image features. Thus, this provides an introductory summary of the image. Sentence encoder is a convolutional encoder that will encode the previous sentence into a condensed semantic vector representation. This captures the meaning to provide context. A recurrent decoder RNN with attention will generate each subsequent sentence conditioned on the image features and previous sentence representation. The attention mechanism will focus on pertinent local visual evidence.

The model will be trained end-to-end on the chest x-ray dataset including the radiologist's report to learn parameters recursively until a stop condition. Natural Language Generation (NLG) metrics such as BLEU, ROUGE and CIDEr will be used in Quantitative evaluation. Human evaluations by clinical radiologists will assess clinical accuracy and utility. The results identify strengths, limitations, and areas of improvement. This approach will demonstrate the ability of deep learning techniques developed to generate radiology reports directly from medical images. The method strikes a balance between the representation of visual evidence and the textual description.

1.4. Thesis outline

Chapter 2: Related works

This chapter reviews existing literature and related works, with a particular focus on metrics such as BLEU, ROUGE, and CIDEr. This section serves to establish a foundation of knowledge by examining prior research in the domain of generating textual outputs from visual inputs, providing insights into the metrics used for evaluation.

Chapter 3: Dataset

The dataset section provides an in-depth exploration of the dataset used in the study. A detailed description of the dataset is presented, highlighting its relevance to the research. The section concludes with a summary, synthesizing the key aspects of the dataset that contribute to the study.

Chapter 4: Deep learning, encoder-decoder architecture, visual attention

This chapter is a comprehensive exploration of the proposed deep learning model. It begins with an overview, followed by detailed discussions on various components such as neural networks, activation functions, ResNet, recurrent neural networks, attention networks, and visual features [16]. The section aims to provide a thorough understanding of the model's architecture and its constituent elements.

Chapter 5: Multimodal learning and implementation

This chapter introduces the concept of multimodal learning, exploring its various modalities such as text, image, audio, and video. Applications of multimodal learning are discussed, including image and text captioning, speech-to-text, video analysis, healthcare informatics, human-computer interaction, and social media analysis. Challenges in multimodal learning, such as heterogeneity, data fusion, and ethical considerations, are also addressed. This section also explains

implementation employed in the study. Each step of the process is explained, including the image encoder, sentence and paragraph generation model, sentence encoder, and sentence decoder. The section concludes with a summary that ties together the key methodological choices made in the study.

Chapter 6: Experiments

This chapter begins with an exploration of the experimental setup, detailing how the model was trained and evaluated. The results of the experiments are presented, offering insights into the model's performance. Sample outputs are showcased to provide tangible examples. The section concludes with a summary, summarizing the findings and outcomes.

Chapter 7: Conclusion

This chapter of the thesis wraps up the study by presenting conclusions drawn from the research. It also outlines potential avenues for future research, suggesting areas where further exploration and refinement of the proposed model could be undertaken.

1.5. Summary

This chapter addresses the critical issue of diagnosing pulmonary abnormalities using medical images, considering the prevalence of chronic respiratory diseases. The introduction emphasizes the global impact of such conditions, underscoring the need for efficient diagnostic tools. The main goal is to generate detailed multi-sentence textual reports for abnormalities, comparable to human-authored reports. The proposed end-to-end neural network model combines convolutional image encoding and recurrent text decoding, integrating deep learning techniques. The research hypothesis posits that this model, leveraging transfer learning and trained on a chest X-ray dataset, will yield clinically accurate radiology reports. The thesis outline includes chapters on related works, dataset exploration, model architecture, multimodal learning, methodology, and experimental results. The concluding chapter summarizes the findings, presents conclusions, and suggests future research directions.

CHAPTER 2

2. Related works

In the last few years, numerous datasets of chest radiographs, comprising nearly one million X-ray images, have been released to the public. The development of efficient computational models, harnessing information from both medical images and textual reports, is an evolving area. Integrating image and text data proves beneficial in enhancing model performance for tasks like image annotation and the automated generation of reports [17].

One of the most popular related works to ours is image captioning [18] which describes images with sentences. Different from the image captioning model, radiology report generation requires much longer generated outputs consisting of different patterns as features, so this task has its own characteristics that require solutions. For example, Xue, et al. [19] proposed an attention mechanism and leveraged a hierarchical Long Short-Term Memory (LSTM) to generate automated reports introducing the concept of a multimodal network. Xuewei, et al. [20] proposed contrastive attention for automatic chest x-ray report generation.

Schlegl et al. [21] introduced a weakly supervised learning approach, using semantic descriptions in reports as labels to enhance the classification of tissue patterns in optical coherence tomography (OCT) imaging. In radiology, Shin et al. (2016) proposed a framework involving convolutional and recurrent networks jointly trained on image and text data to annotate disease, anatomy, and severity in chest X-ray images. Similarly, Moradi et al. [22] processed image and text signals together to generate regions of interest in chest X-ray images. Shin et al. [23], Wang et al. [24] utilized radiological reports to create disease and symptom concepts as labels. They employed techniques like Latent Dirichlet Allocation (LDA) for topic identification and disease detection tools such as DNorm, MetaMap, and various Natural Language Processing (NLP) tools for downstream chest X-ray classification with a convolutional neural network. Additionally, they provided the label set alongside the image data.

Subsequently, Wang et al. [25] utilized the same Chest X-ray dataset to enhance the performance of disease classification and the generation of reports from images. In the realm of report generation, Jing et al. [26] developed a multi-task learning framework featuring a co-attention mechanism module and a hierarchical long short-term memory (LSTM) module. This framework was designed for radiological image annotation and the generation of report paragraphs. Li et al. [27] introduced a reinforcement learning-based Hybrid Retrieval-Generation Reinforced Agent

(HRGR-Agent) to train a report generator capable of deciding whether to retrieve a template or generate a new sentence.

2.1. State-of-the-art literature

Johnson et al. [28] a pre-trained dense-captioning model to identify image semantic regions. Nevertheless, there are no pre-trained models accessible for medical images. Shin et al. [23] developed a DL system to automatically annotate chest X-rays with Medical Subject Headings (MeSH) annotations for the first time, to achieve the aims of medical image annotation. To categorize the X-ray pictures with various disease diagnoses, they employ CNN. Next, more detailed descriptions of the settings around identified diseases are provided to RNNs who are trained for it. Moreover, a cascade approach is utilized to integrate textual and visual contexts to enhance annotation performance. In Zhang et al. [29], a direct multimodal mapping is established between diagnostic results and medical images. Their most effective method of learning image-language alignments is the Auxiliary Attention Sharpening (AAS) module. However, compared to conventional radiology report generation, their problem is less complex because the diagnostic reports they generate are limited to defining five categories of cell appearance attributes.

A hierarchical encoder-decoder design is suggested by Yuan et al. [30] to provide textual reports. The authors argue that rather than being processed independently, frontal, and lateral X-ray pictures should be complementary to one another, therefore pairs of these images are fed into the network as input. Three outputs are employed later: anticipated observations, medical concepts, and global and local aspects of the images. The encoder is the ResNet-152 model, which was pre-trained on the CheXpert [31] dataset. The sentence decoder and word decoder are the two components of the hierarchical LSTM decoder. The sentence decoder uses visual features to produce a hidden state for every sentence. The word decoder then uses these hidden states and the anticipated medical ideas to produce the report.

2.2. Evaluation metrics

The survey by Messina et al. [32] categorizes evaluation metrics in the report generation task into two main categories: text quality measures, which are traditional Natural Language Processing (NLP) or Natural Language Generation (NLG) metrics, and clinical correctness measures, which aim to assess the clinical facts stated in the reports. Here we are focusing on the first metric which is NLG metrics. The evaluation metrics are BLEU [33], ROUGE [34], and CIDEr [35] which measure n-gram matching between the ground truth and a generated text. These metrics are very popular in machine translation, image captioning, and other NLP tasks. The next subsections describe in further detail the NLP metrics.

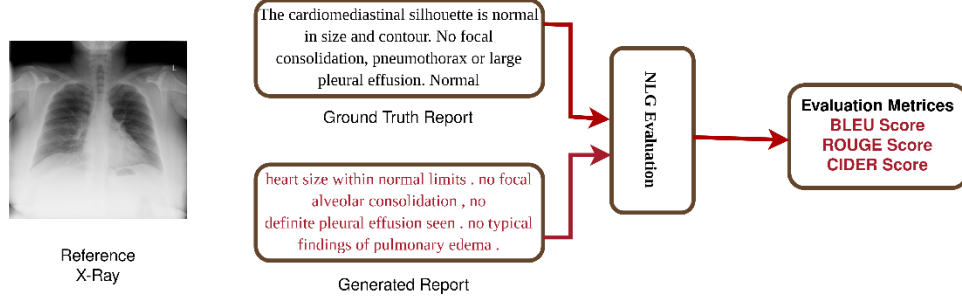


Figure 1 NLG Evaluation System

NLG metrics were developed for tasks involving natural language in the broader domain, such as machine translation, text summarization, or image captioning. These metrics are designed to provide a score indicating the similarity between a ground truth (or reference) text and a generated (or candidate) text. Additionally, in the general domain, these metrics are crafted to accept one or more references per sample to accommodate various ways of rephrasing a sentence while preserving the same meaning. All metrics have a scale from 0 (indicating the worst performance) to 1 (indicating the best performance), except for CIDER-D, the robust version of CIDEr, which has a range from 0 to 10. The widely adopted Python library for computing these metrics is based on the Microsoft COCO Captions Challenge [36].

2.2.1. BLEU

Papineni et al. [33] the Bilingual Evaluation Understudy (BLEU) metric for assessing machine translation. BLEU is a precision-based measure that assesses n-gram overlaps between a target text and one or more reference texts. BLEU-n, where n represents the size of the n-grams, can be computed (e.g., BLEU-1 for unigrams, BLEU-2 for bigrams). In tasks like report generation, BLEU-n is commonly calculated with n values ranging from 1 to 4 (Messina et al. [32]). BLEU focuses on precision rather than recall, indicating how well the generated report aligns with the ground truth but not how much information is accurately captured or omitted. To address this limitation, BLEU includes a penalty for concise candidate sentences in its calculation.

The authors propose calculating a modified n-gram precision q_n for each value of n, shown in equation 2.1, The counters k and l sum over all the samples in the corpus, C_k and C_l are candidate sentences, $Count_{C_l}$ (m-gram) is the number of times that m-gram appears in the candidate C_l , $Count_{C_l \text{ clip } GT_k}$ (n-gram) is the number of times n-gram appears in the candidate C_k and in the ground truth GT_l , clipped to disallow matching the same n-gram multiple times:

$$q_n = \frac{\sum_{k \in \text{Samples}} \sum_{n\text{-gram} \in C_k} Count_{C_k \text{ clip } GT_k}(n\text{-gram})}{\sum_{l \in \text{Samples}} \sum_{m\text{-gram} \in C_l} Count_{C_l}(m\text{-gram})} \quad (2.1)$$

To compensate for the precision-only orientation, the calculation includes a penalization for short sentences, namely the brevity penalty (BP), shown in the equation below, where r is the length of the reference and c is the length of the candidate text:

$$BP = \begin{cases} 1 & c > r \\ e^{(1-\frac{r}{c})} & c \leq r \end{cases} \quad (2.2)$$

Hence, BLEU-N is calculated as the geometric average of the modified precision values up to N, weighted by custom w_n factors. Typically, w_n are uniform (e.g., $w_n = 0.25$ for $N = 4$).

$$BLEU - N = BP \cdot \exp \left(\sum_{n=1}^N w_n \log p_n \right) . \quad (2.3)$$

2.2.2. ROUGE

Lin [34] introduced the Recall-Oriented Understudy for Gisting Evaluation (ROUGE), a suite of metrics designed for assessing text similarity in text summarization. These metrics include ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. In the context of medical report generation, many studies prefer using the ROUGE-L metric (Messina et al. [32]). ROUGE-L is based on evaluating the longest common sub-sequence between the generated and ground truth texts. It incorporates a hyperparameter that allows the metric to be biased towards precision, recall, or an average of both (F-score). In practical terms, the score tends to slightly favor recall in the coco-caption package.

Let us consider a generated text to be a sequence of words $Gen = w_1 w_2 \dots w_n$ and a ground truth text to be a sequence $GT = r_1 r_2 \dots r_m$. As a reminder, by a definition sequence is a subsequence of $Y = y_1 \dots y_M$ if all its elements x_i appear in Y in the same order, though there may be other elements y_j in between. Then, let $LCS(Gen, GT)$ be the length of the longest common subsequence between Gen and GT . Intuitively, if Gen is more like GT , the longer the longest common subsequence found will be. Hence, a notion of recall (R_{lcs}) and precision (P_{lcs}) can be compared:

$$R_{lcs} = \frac{LCS(Gen, GT)}{length(GT)} \text{ and} \quad (2.4)$$

$$P_{lcs} = \frac{LCS(Gen, GT)}{length(Gen)} . \quad (2.5)$$

Thus, ROUGE-L is calculated as a harmonic average between the two measures (F-score),

Using a hyper-parameter β .

$$ROUGE - L = \frac{(1 + \beta^2) \cdot R_{lcs} \cdot P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} . \quad (2.6)$$

If $\beta = 1$, the F_{lcs} is exactly F-1 score; if $\beta = 0$ is the precision, and if $\beta \rightarrow \infty$ it approximates the recall. In practice, β is set to 1.2 in the coco-caption package.

2.2.3. CIDEr

Vedantam et al [35] introduced Consensus-based Image Description Evaluation (CIDEr) as a metric tailored for image captioning. CIDEr characterizes each sentence through a Term Frequency-Inverse Document Frequency (TF-IDF) score across its n-grams. The TF component emphasizes the presence of each n-gram in the sentence, while the IDF component assigns greater

importance to rarer n-grams in the dataset, assuming they convey more valuable information. The similarity between two sentences is determined by the similarity of their TF-IDF representations, with the authors asserting that this approach captures both precision and recall, preserving grammatical and semantic aspects across multiple n values. The authors also introduced CIDEr-D; a variant less susceptible to gaming effects. The original CIDEr ranges from 0 (indicating the worst performance) to 1 (indicating the best), while CIDEr-D ranges from 0 to 10. In the context of the report generation task, many authors do not explicitly specify the variant used (Messina et al. [32]), but the implementation in coco-caption defaults to CIDEr-D.

To compute the TF-IDF score for a given sentence s and an n-gram k , the process is as follows. In simple terms, the TF term quantifies how frequently the n-gram k occurs in s relative to all the n-grams in s . Conversely, the IDF term gauges the inverse of the frequency of the n-gram k across the entire dataset. Therefore, the TF-IDF score, denoted as $g_k(s)$, is approximately determined as follows:

$$g_k(s) = TF \cdot IDF \quad , \quad (2.7)$$

$$g_k(s) = \frac{\# \text{ appearances } k \text{ in } s}{\# \text{ appearances any } n - \text{ gram in } s} \cdot \log \left(\frac{\text{dataset size}}{\# \text{ appearances } k \text{ in dataset}} \right) \quad . \quad (2.8)$$

$$g_k(s) = \frac{h_{k(s)}}{\sum_{l \in n\text{-grams}} h_l(s)} \cdot \log \left(\frac{\# \text{ Images}}{\sum_{i \in \text{Images}} \sum_{q=1}^m h_k(GT_{iq})} \right) \quad , \quad (2.9)$$

where $h_y(x)$ represents the frequency of the n-gram y in the sentence x , and GT_{iq} for $q \in \{1, \dots, m\}$ denotes the m ground truth sentences for the image i ³. Subsequently, for all existing n-grams k_1, k_2, \dots, k_M , a vector $g^{-n}(s)$ is created for each sentence s , with each position containing the TF-IDF score for the respective n-gram k_1, k_2, \dots, k_M . Finally, the similarity between two sentences is computed as the cosine similarity between their vectors, using the equation 2.10 for a specific n and equation 2.11 for averaging up to N-grams.

$$CIDEr_n(Gen_i, GT_i) = \frac{1}{m} \sum_{j=1}^m \frac{g^{-n}(Gen_i) \cdot g^{-n}(GT_{ij})}{\|g^{-n}(Gen_i)\| \|g^{-n}(GT_{ij})\|} \quad . \quad (2.10)$$

$$CIDEr(Gen_i, GT_i) = \sum_{n=1}^N w_n CIDEr_n(Gen_i, GT_i) \quad . \quad (2.11)$$

Typically, N is set to 4, with uniform weights $w_i = 0.25$

Finally, the authors introduced the CIDEr-D variant, which is designed to be more resilient against gaming effects. This is achieved by incorporating a penalty for differences in sentence lengths and employing a more robust counting mechanism that restricts the matching of the same n-gram multiple times.

2.3. Summary

This chapter addresses the related works from image captioning due to the need for longer and pattern-rich outputs. Various approaches are explored in related works, such as attention mechanisms, multimodal networks, and hierarchical encoder-decoder designs. Noteworthy examples include contrastive attention for chest X-ray reports and DL systems for annotating medical images. Evaluation metrics, primarily from the Natural Language Processing domain, include BLEU, ROUGE, and CIDEr, emphasizing text quality measures. BLEU assesses n-gram overlaps with penalties for conciseness, while ROUGE considers the longest common subsequence. CIDEr introduces TF-IDF scores for capturing precision and recall aspects. The paper provides a comprehensive overview of methodologies and evaluation metrics, shedding light on the intricacies of radiology report generation.

CHAPTER 3

3. Dataset

3.1. Dataset collection

Chest X-ray dataset collection for detecting abnormalities was not trivial but the collection of X-ray images with their respective textual form of report was important for us. There were limited sources of dataset that were available publicly. The Indiana University Chest X-ray dataset has emerged as a cornerstone in the realm of medical report generation, as highlighted in the work by Demner-Fushman et al. [37]. This openly accessible data set comprises pairs of chest X-rays coupled with their corresponding semi-structured textual radiology reports. Importantly, it is freely accessible on the web, with no additional prerequisites for downloading. Users have the flexibility to choose between obtaining the reports alone or opting for the images, available in either PNG or DICOM format. This flexibility enhances the utility of the dataset, catering to diverse research needs in the medical imaging domain.

3.2. Dataset description

- **Image Dataset:** It consists of 7470 pairs of image dataset having both frontal and lateral images respectively. The image dataset consists of both normal and abnormal chest X-rays. Basically, there are fourteen types of thoracic pathologies present in the dataset namely Atelectasis, Consolidation, Infiltration, Pneumothorax, Edema, Emphysema, Fibrosis, Effusion, Pneumonia, Pleural Thickening, Cardiomegaly, Nodule, Mass and Hernia.
- **Textual Dataset:** A Textual Dataset basically is a radiological report that consists of distinct sections - Indication, Findings, and Impression. There were about 3995 reports out of which 1525 (38%) of the reports were normal and the remaining 2470 reports represent abnormal findings and diagnoses.



Figure 2 Sample chest radiograph dataset with its respective findings and impressions

The dataset diversity in terms of normal and pathological findings, paired imagery, and reports, as well as textual sections provide rich annotated data for developing and evaluating models for abnormality detection and report generation. However, larger datasets from multiple institutions could further improve model robustness. Overall, the current dataset provides a good foundation for research despite limited public availability of such medical data pairs.

3.3. Summary

This section discusses datasets used for abnormality detection and report generation in chest X-ray images is primarily sourced from the publicly available Indiana University Chest X-ray dataset, widely acknowledged in the field. Comprising 7470 pairs of frontal and lateral images, the dataset includes both normal and abnormal X-rays featuring fourteen thoracic pathologies. The textual component consists of 3995 radiology reports, with 38% being normal and the remainder detailing abnormal findings and diagnoses. Each report encompasses distinct sections—Indication, Findings, and Impression. Despite its richness in annotated data, the study notes the potential for enhanced model robustness with larger dataset from diverse institutions. Nonetheless, the existing dataset forms a solid foundation for research in abnormality detection and report generation, addressing the challenges posed by limited public availability of comprehensive medical data pairs.

CHAPTER 4

4. Deep learning, encoder-decoder architecture, visual attention

4.1. Deep learning

4.1.1. Perceptron

In 1957, Frank Rosenblatt introduced Perceptron [38]. Depending on the original MCP neuron, perceptron learning rule was created. It is a supervised learning technique for binary classification. It allows neurons to learn and process individual components in training sets. It is the simplest form of a single layer fully connected neural network.

Perceptron is an algorithm for learning binary classifier which is also a mathematical model of a biological neuron. While in actual neurons the dendrite receives electrical signals from the axons of other neurons, in perceptron these electrical signals are represented as numerical values. At the synapses, the electrical signals are modulated in various amounts which is also modeled in the perceptron using weights. Actual neuron fires only when total input crosses a certain threshold which is modeled using a threshold function in our case activation function. The model maps multiple values of input into one output either belonging to some class or not. Perception is the basic building block of a neural network. The mathematical formula for the perceptron expanded as:

$$y = activation(\sum_i(w_i \cdot x_i) + b) , \quad (4.7)$$

where,

- x_i represents input features.
- w_i represents corresponding weights,
- b is the bias term,
- $\sum_i(w_i \cdot x_i)$ represents the weighted sum of inputs plus the bias.
- $activation(.)$ is the activation function.

Now, in a fully connected layer of a neural network, each neuron in the layer is connected to every neuron in the previous layer. The output of each neuron in the fully connected layer is calculated in a similar way to perceptron, considering all the inputs from previous layer, their respective

weights, and a bias term. There are multiple neurons in fully connected layer, each with its own set of weights and biases. In summary, a perceptron can be viewed as a single-neuron fully connected layer, and a fully connected layer in a neural network is an extension of perceptron concept to multiple neurons.

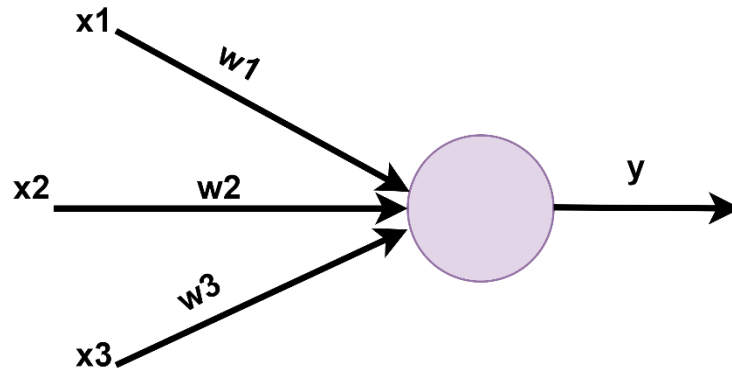


Figure 3 Perceptron

The activation function is a crucial component of the perceptron. It introduces non-linearity to the model, allowing the perceptron to learn complex relationships in the data. Common activation functions include the step function, sigmoid function, or the more widely used rectified linear unit (ReLU).

Perceptron is often arranged into layers to form more complex neural networks. A single-layer perceptron is limited in its ability to solve complex problems because it can only learn linear decision boundaries. However, stacking multiple perceptron in layers and introducing non-linear activation functions allows neural networks to learn and approximate more intricate functions.

The perceptron model and its subsequent extensions laid the foundation for the development of more advanced neural network architectures, such as multilayer perceptron (MLPs) and deep neural networks (DNNs). The ability of these networks to learn hierarchical representations makes them powerful tools in various machine learning tasks, including image and speech recognition, natural language processing, and more.

4.1.2. Activation function

The activation function [39] serves as a crucial element in neural networks, determining the output of a node based on its inputs and associated weights. When tackling complex problems, the utilization of a nonlinear activation function becomes essential. In essence, this function evaluates the weighted sum of inputs, incorporating bias, and decides whether a neuron should be activated. The primary role of the activation function is to introduce non-linearity into the neuron's output. Broadly, activation functions can be categorized into two main types: Linear Activation Functions and Nonlinear Activation Functions. This distinction plays a pivotal role in enabling neural networks to learn and model intricate patterns and relationships within data, extending their capabilities beyond linear mappings.

Linear activation function

The linear activation function calculates the output as a simple linear combination of the inputs and weights. Mathematically, it can be represented as:

$$f(x) = ax + b, \quad (4.1)$$

where a is the weight, x is the input, and b is the bias term.

The key attribute of a linear activation function is that it produces a linear relationship between inputs and outputs. This means that the network can only learn linear mappings from input to output, and stacking multiple layers of linear activations would not provide any additional modeling capability.

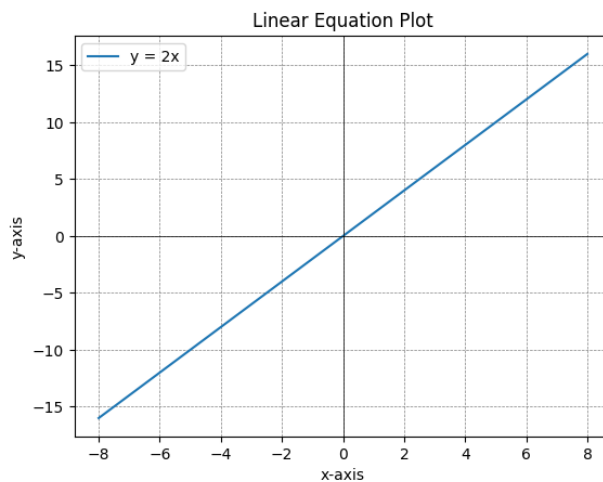


Figure 4 Linear function plot

Nonlinear activation function

Nonlinear activation functions introduce nonlinearity into the network, enabling it to learn and approximate more intricate relationships in the data. It makes it easy for the model to generalize or adapt with a variety of data and to differentiate between the output. The Nonlinear Activation Functions are divided based on their range or curves. The basic plot for nonlinear function is shown below:

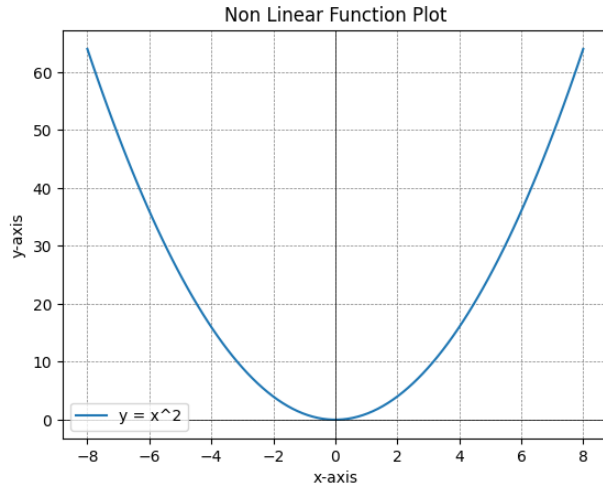


Figure 5 Nonlinear function plot

Popular nonlinear activation functions include:

Sigmoid or logistic activation function

This function squashes input values between 0 and 1, making it useful for binary classification problems. It is mathematically expressed as:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (4.2)$$

The sigmoid function [40] curve looks like an S-shape. This function is differentiable which means we can find the slope of the sigmoid curve of any two points where z is the input function. The logistic sigmoid function can cause a neural network to get stuck at the training time.

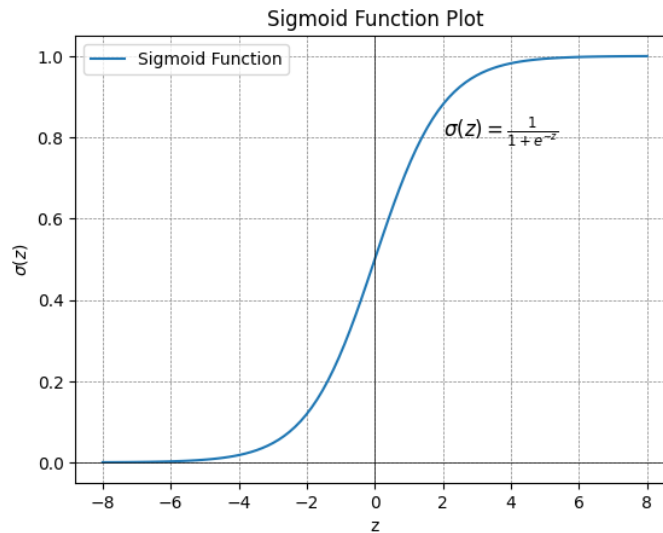


Figure 6 Sigmoid function plot

Hyperbolic tangent activation function or tanh function

The hyperbolic function [41] shares similarities with the sigmoid function. However, unlike the sigmoid output range restricted between 0 and 1, the hyperbolic tangent function spans from -1 to 1. While this may not precisely mirror the behavior of neurons in the brain, the hyperbolic tangent function tends to offer advantages over the sigmoid, particularly in the training of neural networks.

The equation for the hyperbolic tangent (tanh) function is:

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} = \frac{1 - e^{-2z}}{1 + e^{-2z}} . \quad (4.3)$$

This equation 4.3 represents how the hyperbolic tangent function transforms the input z to an output within the range of -1 to 1. We can use this function in neural networks as an activation function to introduce non-linearity and handle a broader range of input values.

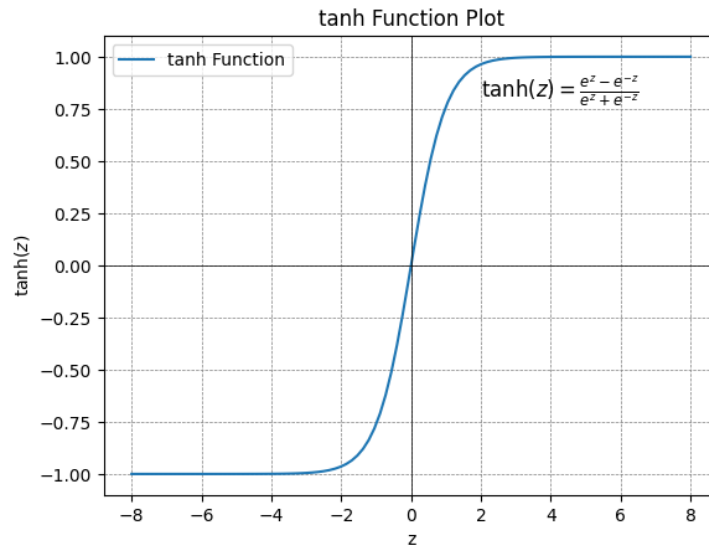


Figure 7 Tanh function plot

In comparison to the sigmoid function, which can sometimes lead to training difficulties when inputs are strongly negative and keep the output near zero, the hyperbolic tangent's range from -1 to 1 allows it to handle negative inputs more effectively. This broader output range helps prevent the issue of neural networks getting "stuck" during training, providing a smoother learning process. The hyperbolic tangent function is a popular choice as an activation function in neural networks due to its improved ability to capture and learn from a wider range of input values, contributing to more robust and effective training outcomes.

Rectified Linear Unit (ReLU)

The Rectified Linear Unit (ReLU) [42] is indeed one of the most widely used activation functions in deep learning, known for its efficiency and simplicity. ReLU introduces non-linearity to the model, allowing it to learn and approximate complex relationships in the data. It is considered biologically plausible, mimicking the firing behavior of neurons in the human brain.

The ReLU activation function is defined as:

$$\text{Relu}(z) = \max(0, z) \quad . \quad (4.4)$$

This function outputs the input value z if it is positive and zero otherwise. The simplicity of ReLU makes it computationally efficient, and its ability to produce sparse activations (zero for negative inputs) contributes to the model's capacity to learn robust representations.

One of the significant advantages of ReLU is that it avoids issues like the vanishing gradient problem encountered by traditional activation functions, such as sigmoid or hyperbolic tangent, which tend to saturate for extreme values. ReLU allows for faster convergence during training, as it does not saturate for positive inputs.

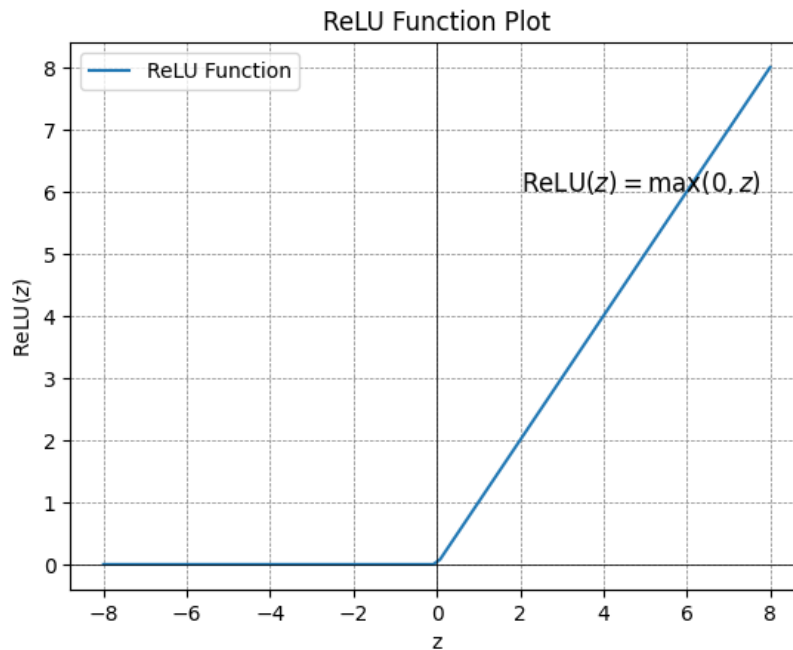


Figure 8 ReLU function plot

The graphical representation of ReLU typically shows a piecewise linear function, where the output is zero for negative inputs and follows a linear slope for positive inputs. This simplicity facilitates gradient-based optimization and makes ReLU well-suited for deep neural networks. However, it's worth noting that ReLU is not without its challenges. The "dying ReLU" problem may occur when neurons become inactive (outputting zero) for all inputs during training, leading to dead pathways in the network. Variations like Leaky ReLU or Parametric ReLU aim to address this issue by allowing a small, non-zero output for negative inputs.

Exponential Linear Unit (ELU)

The Exponential Linear Unit (ELU) [43] is an activation function that shares similarities with the Rectified Linear Unit (ReLU) but introduces a smooth transition for negative input values. The primary motivation behind ELU is to mitigate some of the limitations associated with ReLU, such

as the "dying ReLU" problem where neurons can become inactive for certain inputs during training.

The ELU activation function is defined as:

$$f(z) = \begin{cases} z, & \text{if } z \geq 0 \\ \alpha(e^z - 1), & \text{if } z < 0 \end{cases} \quad (4.5)$$

where α is a small positive constant. When z is non-negative, ELU behaves like the identity function, allowing positive value to pass through unchanged. For negative values of z , ELU smoothly transitions, avoiding the abrupt cutoff at zero seen in ReLU.

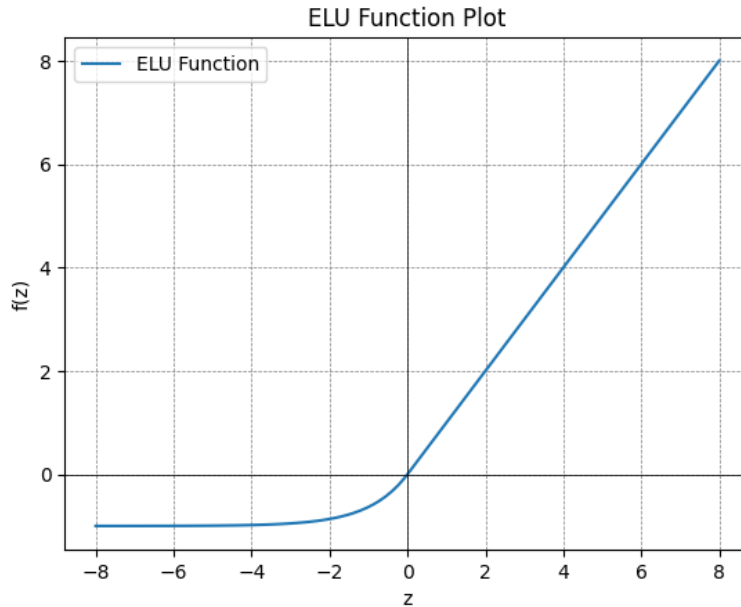


Figure 9 ELU function plot

SoftMax function

The SoftMax function [44] is especially used in the output layer of neural network models dealing with multi-class classification problems. It transforms a vector of real numbers into a probability distribution, where the probability of each class is proportional to the exponentiated value of that class's score relative to the sum of exponentiated scores across all classes. This ensures that the output values lie in the range (0, 1) and sum up to 1, making them interpretable as probabilities. The mathematical expression for the SoftMax function for a vector $z = (z_1, z_2, \dots, z_k)$ of k real numbers is given by:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \quad \text{for } i = 1, 2, \dots, k \quad (4.6)$$

where:

$\sigma(z_i)$ is the i -th component of the Softmax output.

e^{z_i} is the exponential of the i -th component of the input vector.

The denominator $\sum_{j=1}^k e^{z_j}$ is the sum of exponentiated values across all components of the input vector.

This function essentially normalizes the input vector into a probability distribution. The larger the exponentiated value of a component, the higher its probability in the resulting distribution. The SoftMax function is crucial in multi-class classification scenarios where the goal is to assign an input to one of several possible classes. The SoftMax function is often used in conjunction with the categorical cross-entropy loss function during the training of neural networks for multi-class classification tasks. The combination of SoftMax and categorical cross-entropy allows the model to learn meaningful representations and make probabilistic predictions across multiple classes.

4.1.3. Convolutional neural network

Convolutional Neural Networks (CNNs) [45] represent a specialized class of neural networks designed to process and analyze visual data, making them particularly effective in computer vision applications. Unlike traditional neural networks, CNNs leverage a unique architecture inspired by the visual processing in the human brain, allowing them to automatically learn hierarchical representations of features from input images.

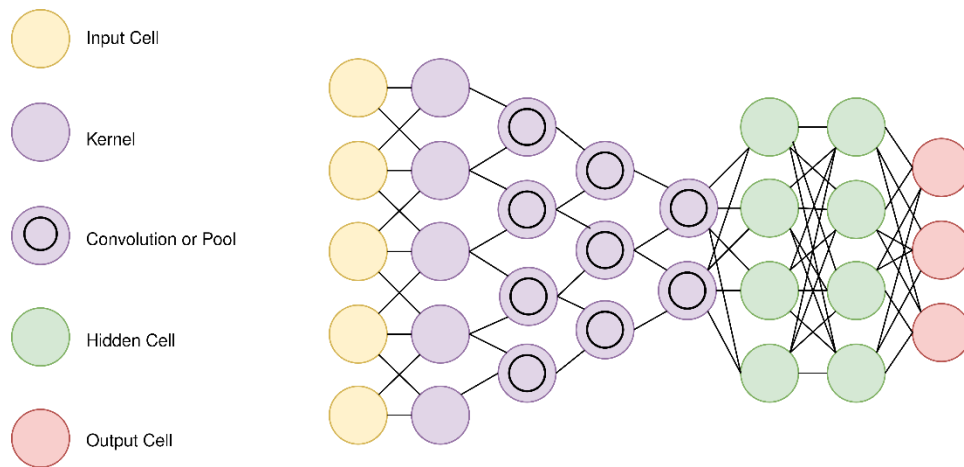


Figure 10 Convolutional neural network

Convolution layer is the core building block of CNNs is the convolutional layer. This layer applies convolutional operations to the input data, using filters (also called kernels) to extract local patterns and features. This process enables the network to detect simple features like edges and textures in the early layers and progressively more complex features in deeper layers. Pooling layers follow convolutional layers and serve to reduce the spatial dimensions of the data, making computations more efficient and lowering the risk of overfitting. Common pooling operations include max pooling, which retains the maximum value in each region, and average pooling, which calculates the average. Toward the end of the network, fully connected layers are employed to make predictions based on the high-level features learned in the previous layers. These layers connect

every neuron to every neuron in the adjacent layers, enabling the network to make complex decisions.

Filters are the small windows applied to input data during convolutional operations. Kernels represent the weights associated with these filters. Learning the values of these kernels allows the network to automatically extract relevant features from the input. Activation functions, such as ReLU (Rectified Linear Unit), introduce non-linearity to the network, enabling it to learn complex patterns. ReLU, for example, replaces negative values with zero, facilitating efficient training. Striding refers to the step size with which the filter moves across the input during convolution. Adjusting the stride affects the spatial dimensions of the output, influencing the receptive field and the amount of information retained.

Convolutional Neural Networks (CNNs) have evolved with various architectures to address different tasks and challenges in computer vision. Some of them are AlexNet [46], LeNet-5 [47], VGGNet [48], Inception [49], Xception [50], ResNet [51], MobileNet [52], DenseNet [53] and so on. Here in our experiment, we have worked on ResNet.

ResNet

The ResNet (Residual Network) [51] architecture introduced by Kaiming He et al. has several variants, each denoted by the depth of the network. The key idea behind ResNet is the use of residual blocks, which contain shortcut connections (skip connections) that bypass one or more layers, enabling the model to learn residual functions. Here are some notable types of ResNet architectures:

ResNet-18 is a relatively shallow variant of the architecture. It consists of 18 weight layers, including convolutional layers, pooling layers, fully connected layers, and skip connections. ResNet-18 is often used for tasks where computational resources are limited.

ResNet-50 is a deeper variant that gained attention for winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [54] in 2015. It includes 50 weight layers and introduces bottleneck blocks, which use 1x1 convolutions to reduce the number of parameters and computational complexity.

Resnet-101 is an extension of ResNet-50, featuring 101 weight layers. It provides a deeper architecture, allowing for more complex feature learning. ResNet-101 is often preferred for tasks that demand a higher level of representation.

Wide ResNet introduces width as a factor in addition to depth. Wider networks, such as WRN-50-2, have more filters in each layer, promoting feature diversity. The "2" in WRN-50-2 indicates that the width factor is 2. Wide ResNets aim to achieve better performance by increasing model width.

ResNet-PreAct modifies the original ResNet architecture by incorporating batch normalization and ReLU before each convolution operation rather than after. This adjustment helps with the training of very deep networks by mitigating issues like vanishing or exploding gradients.

ResNeXt is an extension of ResNet that introduces a cardinality parameter, representing the number of independent paths within a group. This modification enhances the expressive power of

the network. ResNeXt architectures, such as ResNeXt-50, offer competitive accuracy with improved efficiency.

ResNet-152 is one of the deepest variants within the original ResNet architecture family. Introduced by He, et al. [51], ResNet-152 is characterized by its exceptional depth, consisting of 152 weight layers. This depth allows the network to capture intricate and hierarchical features, making it well-suited for demanding computer vision tasks where a high level of representation is crucial.

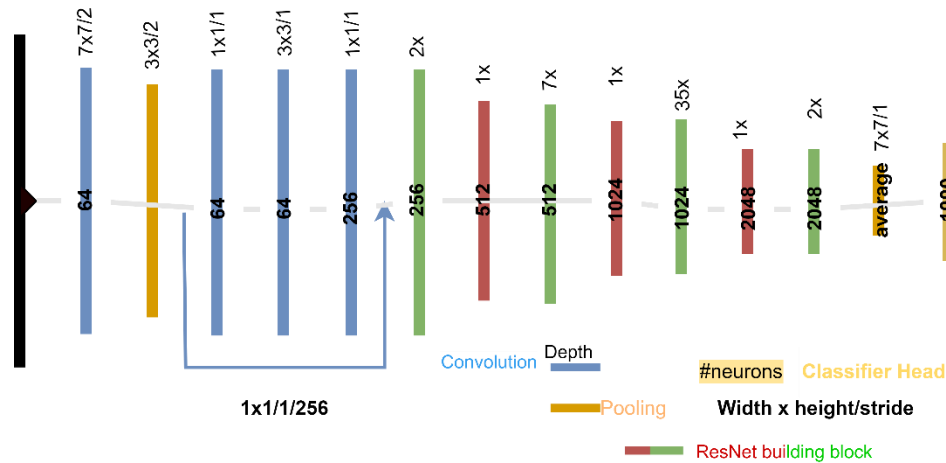


Figure 11 ResNet152 Architecture

The primary distinction of ResNet-152 lies in its depth. With 152 weight layers, including convolutional layers, residual blocks, pooling layers, and fully connected layers, it can automatically learn and represent highly complex patterns and features in input data. ResNet-152 utilizes residual blocks as its fundamental building units. Each residual block contains skip connections (shortcut connections) that allow the gradient to flow more efficiently during backpropagation. This mitigates the vanishing gradient problem, enabling the successful training of very deep networks. Like other deep ResNet variants, ResNet-152 adopts a bottleneck architecture in its residual blocks. This involves the use of 1x1 convolutions to reduce the number of parameters and computational load, allowing the network to maintain efficiency despite its depth. ResNet-152 was trained on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset, demonstrating its ability to achieve state-of-the-art performance in image classification tasks. The network excels in recognizing and categorizing objects within images across a wide range of classes. Due to its pre-trained weights on ImageNet, ResNet-152 is often used as a base model for transfer learning. Transfer learning involves fine-tuning the model on a specific task with a smaller data set. This is particularly useful when working with limited labeled data for a specialized application.

4.1.4. Recurrent neural network

Recurrent Neural Networks (RNNs) [55] are a class of neural networks designed to handle sequential data by capturing dependencies and patterns over time. Unlike traditional feedforward

neural networks, RNNs possess internal memory, allowing them to maintain a state representation that evolves as new inputs are processed. This unique architecture makes RNNs particularly effective for tasks involving sequences, such as time series prediction, natural language processing, and speech recognition. Since we are trying to generate reports from sequence of tokens this neural network model should perform better than previous dense feed forward model. Unlike the feedforward model, these neural networks can use their internal state to process the sequences of inputs because of which they have been used in Language Modeling, Machine Translation, Speech Recognition, and various other tasks. But because of two major problems in RNN model called vanishing and exploding gradients problem we use LSTM [56] model for training the classifier as LSTM model uses forget gates to decide when to forget and remember information for long periods of time of time.

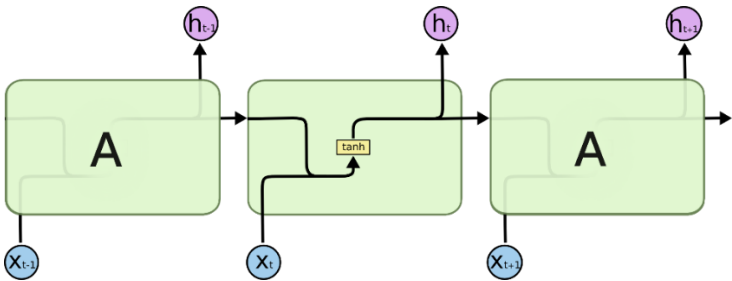


Figure 12 Standard RNN network

Long Short-Term Memory (LSTM)

LSTM [56] are a special kind of RNN, capable of learning long term dependencies. They were introduced by Hochreiter and Schmidhuber in 1997 and were refined and popularized by many people. They work tremendously well on a large variety of problems and are now widely used. All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

The LSTM's also have a chain like structure, but repeating modules have different structures as shown in Figure 13:

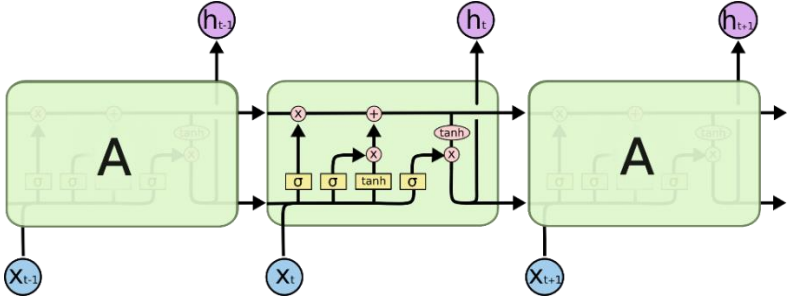


Figure 13 LSTM architecture

The above shown LSTM cells are good at remembering the data is because they can choose to either remember or forget the data based on the weights of the network. This feature of this network helps to tackle the major problem of general RNN's.

4.2. Encoder-decoder architecture

Deep Learning (DL) [57] model as an encoder-decoder architecture with an attention mechanism is a sophisticated approach within artificial intelligence. This architecture is widely used in tasks involving sequence-to-sequence transformations, including machine translation and image captioning. The encoder component handles the processing of input data, often in the form of sequential data like sentences or images and generates a concise representation that captures the essential features. In contrast, the decoder uses this representation to gradually produce the desired output sequence. The integration of an attention mechanism further strengthens the model's capacity to selectively concentrate on specific parts of the input sequence during the decoding process.

Encoder plays a vital role in deep learning applications, as it is responsible for transforming the data from source into a format that can be easily understood by machines. Utilizing neural network structures, this crucial component extracts significant features and representations from source data. A commonly used and highly effective method is the Convolutional Neural Network (CNN) [45] for images and Long Short-Term Memory (LSTM) [56] for text data sources.

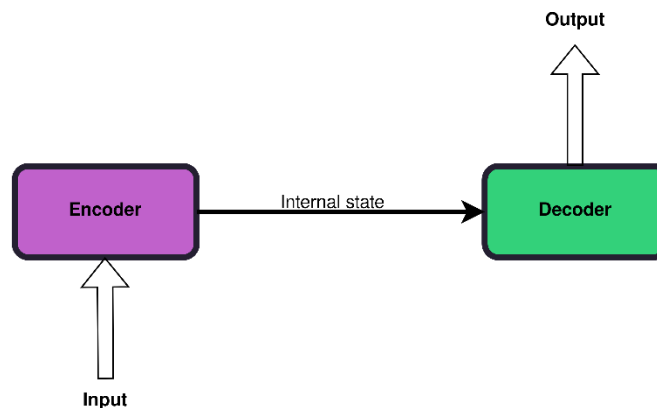


Figure 14 Encoder decoder architecture

CNN processes images through multiple layers of convolution and pooling, enabling the capture of hierarchical features. These learned representations, presented in the form of vectors or tensors, offer a compact and semantic encoding of the original image, and greatly enhance the performance of downstream tasks such as image classification, object detection, and image generation. In essence, image encoders serve as a vital bridge between the visual world and machine understanding. Here in our experiment, the image encoder comprises the CNN model which is used to extract global and local visual features.

Whereas, LSTMs, designed to capture sequential dependencies, process input sentences token by token, updating hidden states at each step. The final hidden state encapsulates the comprehensive information from the entire sentence, presenting a distilled and contextual encoding. In our experiment, the sentence encoder utilizes LSTM, acting as a powerful tool to understand and represent the sequential nuances within the text. This encoding is invaluable for various natural language processing tasks, including sentiment analysis, text classification, and language generation, effectively bridging the semantic understanding of sentences for downstream applications in our model.

The sentence decoder processes the encoded information, often in the form of a fixed-dimensional vector or tensor, to produce a sequential output. A common choice for the decoding component is recurrent neural networks (RNNs) [55] or, more recently, attention-based mechanisms and transformer architectures. For example: in a model with Long Short-Term Memory (LSTM) decoders, the decoder processes the encoded sentence representation and generates the output sequence step by step. At each decoding step, the LSTM considers the previously generated tokens and their own internal state, producing the next token in the sequence. The process continues iteratively until the entire output sequence is generated.

Also, incorporating attention mechanisms in the decoder enhances its ability to selectively focus on different parts of the encoded input, allowing the model to align its attention with relevant information during the generation process. This is particularly useful for handling long sentences or sequences where certain parts require more emphasis.

4.3. Visual attention

Attention mechanisms have become a core component of many state-of-the-art deep learning models, especially in natural language processing and computer vision tasks. Attention allows models to selectively focus on parts of a large input that are most relevant to the task being performed. This contrasts with encoding the entire input into a single fixed-length vector, which can lose important details.

For example, in neural machine translation, attention allows the model to dynamically attend to certain words in the source sentence when generating each word in the target sentence. This provides proper context for the model to translate words appropriately based on the full source sentence, rather than relying solely on a fixed encoding.

Here in our case, the core idea behind attention is to compute relevance scores between elements of one modality (e.g., words in a sentence) and elements of another modality (e.g., regions in an image). This allows the model to dynamically attend to inputs that are pertinent to generating the output, as opposed to relying solely on fixed encodings.

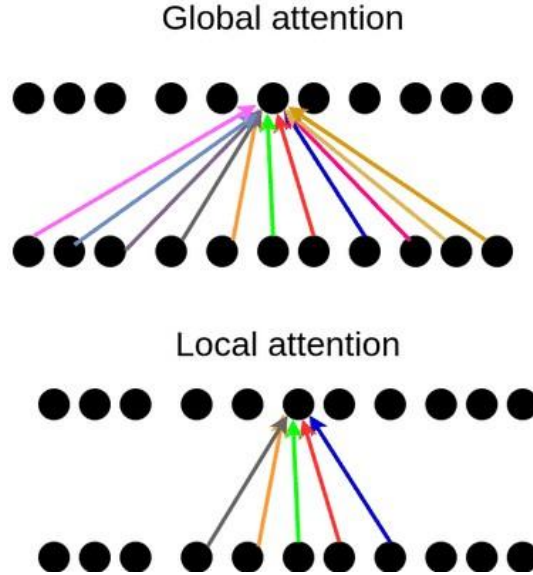


Figure 15 Example showing attention. (Source: <https://theaisummer.com/attention/>)

Formally, consider input vectors $\{x_1, x_2, \dots, x_n\}$ that represent elements of one modality, like words in a source sentence. We also have context vectors $\{h_1, h_2, \dots, h_m\}$ that encode contextual information from the second modality, like the partially generated target sentence.

First, the input and context elements are projected into a joint embedding space using learned projection matrices W_x and W_h :

$$u_i = W_x x_i \text{ and} \quad (4.8)$$

$$v_j = W_h h_j . \quad (4.9)$$

Then, relevance scores are computed between each input x_i and context h_j using a scoring function f like dot product:

$$e_{ij} = f(u_i, v_j) . \quad (4.10)$$

Common choices for f are dot product or a small multilayer perceptron. The scores are normalized using a SoftMax to get attention probabilities:

$$a_{ij} = \exp(e_{ij}) / \sum_k \exp(e_{ik}) . \quad (4.11)$$

These probabilities represent the relevance of each input element to the current context. The attended input is a weighted sum of the inputs, with the weights given by the attention probabilities:

$$\text{attended input} = \sum_i a_{ij} \cdot x_i . \quad (4.12)$$

So, the most relevant inputs are dynamically amplified while irrelevant ones are suppressed. The projections and scoring function are learned via backpropagation during training.

Attention mechanisms excel at selectively focusing on pertinent information needed for the task based on the context, rather than relying solely on fixed-length encodings. They have led to state-of-the-art results in machine translation, text summarization, image captioning, and other tasks involving sequential data like speech recognition. As models scale to even larger dataset, selective attention provides an efficient and flexible approach to handling long, high-dimensional inputs.

4.3.1. Visual feature

A core part of image-to-text generation systems involves encoding meaningful visual representations from the input images [58]. Typically, CNNs pre-trained on large dataset are used to extract visual features. CNNs apply a series of convolutional and pooling layers to transform the raw image pixels into higher-level feature representations. Lower layers detect basic visual concepts like edges and textures. Higher layers encode more complex semantics related to full objects and scenes. Global and Local are two types of visual features which can be extracted from CNNs.

Global features provide a holistic representation of the entire image. Features from the final CNN layer summarize the full semantic content in a single vector. However, spatial information is discarded.

Local features retain spatial localization by extracting patches or region vectors from intermediate CNN layers. This provides a grid of local feature vectors, each encoding information about a part of the image.

Combining global and local features allows models to leverage both holistic scene understanding and fine-grained spatial details. The global representation captures high-level concepts, while local features focus on spatial areas relevant to generating coherent output like captions. Attention mechanisms are often applied over the local feature grid to emphasize image regions dynamically based on context from the textual modality. For instance, generating the word "dog" in a caption may highlight local features corresponding to the dog's location. This allows contextual, spatially aware focusing during multimodal output generation. Hence, global features summarize whole-image semantics while local features retain spatial information. CNNs provide an effective means to extract both representations, which can be fused in context-aware ways to generate coherent multimodal outputs. The global-local duality augments visual understanding for downstream generation tasks.

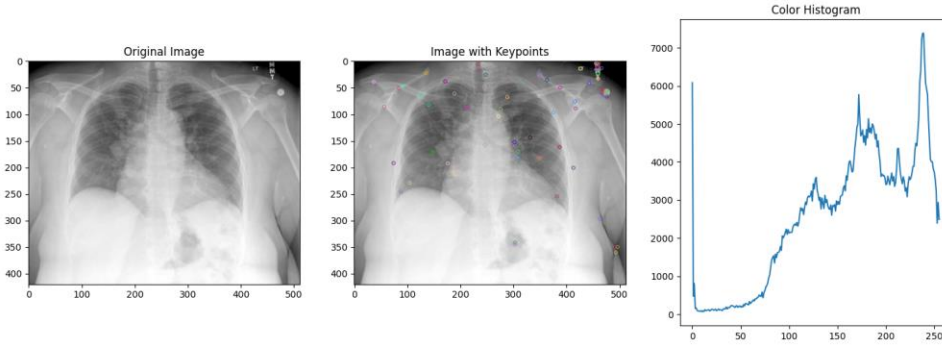


Figure 16 Visual features

Here, Local features capture information from specific regions or points in an image. In this case, SIFT detects key points (interest points) and computes descriptors that describe the local appearance and texture around each key point. These features are invariant to scale changes and rotations, making them suitable for various computer vision tasks and the color histogram represents the distribution of colors across the entire image. It is a global feature because it summarizes information about the entire image without focusing on specific regions or points. Color histograms are often used for color-based image retrieval and analysis.

4.4. Summary

The section discusses various deep learning techniques for sequence-to-sequence tasks like machine translation and image captioning. An effective approach for sequence-to-sequence tasks like machine translation and image captioning is an encoder-decoder architecture with attention mechanisms. The encoder processes the input and generates a representation while the decoder produces the output sequence, using the attention mechanism to focus on relevant input parts. For images, CNNs provide global and local hierarchical visual features and RNNs like LSTMs model textual sequential dependencies and long-term context. Components like convolutional layers, fully connected layers, activation functions and optimization algorithms enable the models to learn complex multimodal patterns. When combined in an end-to-end framework, these specialized neural modules allow generating coherent outputs reflecting understanding of both modalities, such as image captions attending to visual elements based on textual context. The goal is to leverage the strengths of diverse deep learning architectures to build integrated systems capable of relating and translating between multiple data types, through components like encoders extracting representations, decoders generating relevant outputs, and attention focusing on pertinent inputs.

CHAPTER 5

5. Multimodal learning and implementation

5.1. Multimodal learning

5.1.1. Basics

Within the vast domain of artificial intelligence and machine learning, scientists are investigating the merging of data from various sources to build increasingly complex and human-like systems. This problem is addressed by multimodal learning, an interdisciplinary field at the nexus of computer vision, natural language processing, and audio analysis. Multimodal learning [59] enables machines to extract meaningful information from a variety of modalities, including text, images, audio, and video. Multimodal learning makes use of the diversity of information streams, in contrast to traditional approaches that concentrate on a single modality, to provide a more comprehensive comprehension of complicated data.

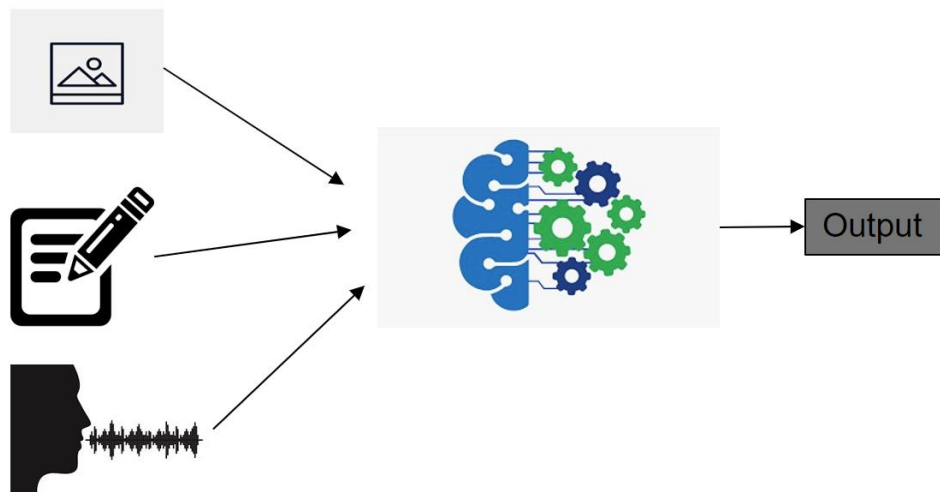


Figure 17 Multimodal theoretical architecture

In the quest to develop intelligent systems that can mimic human thought processes, multimodal learning has become a viable approach. Humans effortlessly integrate information from various senses, blending textual information with visual cues, sounds, and other sensory inputs to form a nuanced understanding of their environment. Recognizing the importance of this integrated

perception, researchers are turning to multimodal learning to imbue machines with the ability to process and interpret information across different modalities.

The motivation behind embracing multimodal learning is multifaceted. In many practical scenarios, information naturally manifests in multiple modalities, necessitating a more comprehensive approach to analysis. For instance, understanding an image may be greatly enhanced by accompanying textual descriptions, and deciphering spoken language may benefit from the inclusion of visual context. Multimodal learning, therefore, addresses the inherent complexity of real-world data and presents a means to overcome the limitations of unimodal models, offering improvements in performance, robustness, and interpretability.

5.1.2. Data types

Multimodal learning involves different types of information, or modalities, working together for a more complete understanding. One crucial modality is text, which includes written words and linguistic expressions. Natural Language Processing (NLP) helps extract meaning from text, making it useful for tasks like sentiment analysis and language translation. Another key modality is images, where computer vision algorithms analyze visual data for tasks such as image recognition and captioning. Audio is a modality focused on sound and speech, used in tasks like speech recognition and emotion detection. Videos, with both spatial and temporal features, form another modality, aiding in tasks like action recognition and video captioning. Combining these modalities enhances the overall interpretation of information, allowing models to grasp diverse aspects and gain a more comprehensive understanding of content.



Figure 18 Example of different modalities

5.1.3. Application

The key applications include image and text captioning, enhancing content accessibility and assistive technologies. Speech processing benefits from multimodal learning in tasks like accurate speech-to-text transcription and natural text-to-speech synthesis. Video analysis, incorporating spatial and temporal features, supports tasks such as action recognition, event detection, and summarization, applicable in surveillance and sports analysis. In healthcare, multimodal learning aids comprehensive diagnostics by integrating medical images, patient records, and clinical notes. Human-computer interaction is shaped by multimodal interfaces, offering natural user experiences in virtual assistants, augmented reality, and education. Social media analysis benefits from extracting insights across text, images, and videos, enabling sentiment analysis, content recommendation, and trend detection. These diverse applications highlight the transformative impact of multimodal learning across multiple domains.

5.1.4. Challenges

One primary challenge lies in the heterogeneity of modalities, as text, images, audio, and video inherently differ in data structures and representations. Harmonizing these modalities for effective information fusion poses a non-trivial task. Integrating information from multiple modalities requires addressing challenges in data fusion and alignment, particularly when dealing with unstructured data such as images and text. The scarcity of labeled multimodal data, especially in specialized domains like medical imaging, hinders robust model training. Intermodal variability, where different modalities may describe the same concept variably, and the inherent complexity of multimodal learning models, especially in terms of scalability and ethical considerations, further underscore the challenges faced by practitioners in unlocking the full potential of multimodal approaches to AI and machine learning. Developing standardized benchmarks for fair comparison and defining appropriate evaluation metrics are ongoing efforts within the research community to address the diversity of tasks and modalities in multimodal learning.

5.2. Implementation

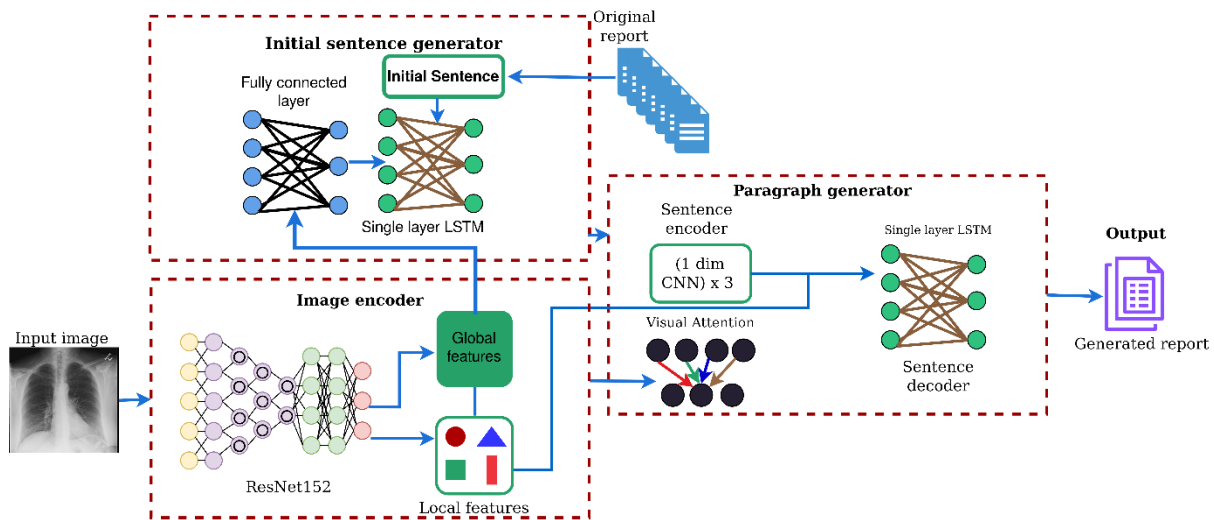


Figure 19 Schema of proposed model

Figure 19 illustrates the entire architecture of our system, which receives medical images from multiple perspectives as input and produces a radiology report with impressions and findings. We use an encoder-decoder model, which accepts an image pair as input and outputs the first phrase, to create the findings paragraph. The semantic representation of the first sentence is then generated by processing it through a sentence encoding network. Subsequently, a multimodal recurrent generation network is employed to generate the next sentence by combining the visual features of the image with the semantic features of the previous sentence. Until the model produces the last sentence in the paragraph, this iterative procedure is carried out.

5.2.1. Image encoder

We begin a procedure for image processing using a pre-trained ResNet-152 model as an image encoder. The process starts by resizing all input frontal images to a consistent size of 256×256 pixels, followed by cropping them to 224×224 pixels to meet the input requirements of the ResNet-152 architecture. The key objective is to extract both local and global visual details from the images. For local feature extraction, the script focuses on the "res4b35" layer of the ResNet-152 model, resulting in a local feature matrix denoted as f in $\mathbb{R}_{1024 \times 196}$. This matrix is derived from 196 sub-regions, each corresponding to a regional feature vector, effectively dividing the image into smaller components. Simultaneously, a global feature vector (f in \mathbb{R}_{2048}) is obtained from the last average pooling layer of ResNet-152, providing a comprehensive representation of the entire image.

During the training phase, a notable aspect is that all parameters in the layers generated from the ResNet-152 architecture remain constant. This approach, known as feature extraction, involves using the pre-trained model as a fixed feature extractor. By keeping these parameters unchanged, the script maximizes computing efficiency, as training a large model like ResNet-152 from scratch can be computationally intensive. The decision to use pre-trained weights ensures that the model benefits from the knowledge acquired during its initial training on a diverse dataset, enhancing its ability to extract meaningful features from the input frontal images. Overall, the script underscores the practical considerations of leveraging a pre-trained deep learning model for efficient image processing while balancing the extraction of both local and global features.

5.2.2. Sentence and paragraph generation model

In the findings section's introductory sentence, we encapsulate crucial information pertaining to the image, and to enhance this process, we've developed a dedicated sentence generation model. This model is designed to consider the global features, which are acquired through the image encoder. These specialized models are meticulously trained with a specific emphasis on generating expressive impressions. For the initiation of sentence generation, findings are employed as the initialization for a recurrent model, as detailed in Equation 2. In this context, a single-layer Long Short-Term Memory (LSTM) network is utilized for sentence decoding. To kickstart the sentence-creation process, the LSTM's initial hidden states and cell states are set to zero. An intriguing aspect of our approach lies in our ability to anticipate the first word of the sentence by utilizing the visual feature vector as the initial input for the LSTM. Subsequently, the entire sentence is constructed word by word.

Before inputting data into the LSTM, we employ a fully connected layer to transform the visual feature vector, ensuring its dimensionality aligns with that of the word embeddings. Throughout our research, the sizes of the word embeddings and hidden states are fixed at 512 and 1024, respectively, for every LSTM module employed. As highlighted earlier, our paragraph generation algorithm operates by generating results sentence by phrase, utilizing both sentence and image attributes as input. This comprehensive approach involves two primary components, emphasizing the integration of visual information with the semantic richness derived from the sentence generation model. The synergy between the image encoder and the LSTM-based sentence

generation model forms a robust foundation for generating detailed and contextually rich descriptions based on the visual content.

Let us assume that we are generating paragraphs of findings that contain P sentences. The probability of generating k -th sentence with length L satisfies and the probability of the sequence of words $W_i = (s_1, s_2, \dots, s_L)$ given the context Q and parameters α is calculated as:

$$P(W_i = (s_1, s_2, \dots, s_L) | Q; \alpha) = P(W_1 | Q) \prod_{i=2}^{k-1} P(W_i | Q, W_1, \dots, W_{i-1})$$

$$P(s_1 | Q, W_{k-1}) \prod_{t=2}^L P(s_t | Q, W_{k-1}, s_1, \dots, s_{t-1}), \quad (6.1)$$

where Q represents the provided medical image, while α denotes the model parameter. W_i represents the i^{th} sentence, and s_t represents the t^{th} token within that sentence. We employ a Markov assumption for sentence-level generation, akin to the n -gram assumption in language models. Specifically, we adopt a "2-gram" model, where the generation of the current sentence depends solely on its immediately preceding sentence and the accompanying image. This simplification allows us to estimate the probability in a more straightforward manner given as:

$$\hat{P}(W_i = (s_1, s_2, \dots, s_L) | Q; \alpha) = (P(W_1 | Q)) \prod_{j=2}^{k-1} P(W_j | Q, W_{j-1})$$

$$P(s_1 | Q, W_{k-1}) \prod_{t=2}^L P(s_t | Q, W_{k-1}, s_1, \dots, s_{t-1}). \quad (6.2)$$

Our objective is to determine the best parameter values for the Maximum Log-Likelihood Estimation (MLE).

$$\alpha^* = \operatorname{argmax}_{\alpha} \sum_{k=1}^P \log \hat{P}(W_k = D_k | Q; \alpha), \quad (6.3)$$

where D_k represents the ground truth for the k^{th} sentence in the findings paragraph. As illustrated in Equation 6.2, we break down this equation into three distinct parts, where the first probability

function is the first part, the second probability function is the second part and rest of all is third part and introduce our model step by step.

5.2.3. Sentence encoder

The methodology employed in our approach plays a pivotal role in extracting semantic vectors from textual descriptions, enhancing our understanding of the underlying meaning within the content. The primary focus lies in distilling meaningful information from textual descriptions to enable a more nuanced representation of the content's semantics. For the task of sentence encoding, we utilize a 1D Convolutional Neural Network (CNN), specifically tailored to process 512-dimensional word embeddings. Word embeddings serve as continuous vector representations of words, facilitating the neural network's ability to comprehend and analyze sequential data, such as sentences. The structure of our sentence encoding CNN comprises three convolutional layers, each strategically designed to capture hierarchical features present in the input word embeddings. These convolutional layers use filters with a kernel size of three and a stride of one. The kernel size defines the filter's dimensions, while the stride determines the step size of the filter as it scans the input data. Each convolutional layer produces 1024 feature channels, representing distinct patterns or aspects within the input data. To distill meaningful information from these channels, we apply a max-pooling operation to the feature maps generated by each layer. This operation results in concise 1024-dimensional feature vectors for each layer, encapsulating the most salient features from the hierarchical representations. The final sentence feature is derived by concatenating the feature vectors obtained from different layers. This approach enables the model to capture and leverage a comprehensive set of semantic features extracted from the input text descriptions. In essence, our methodology with the 1D CNN, convolutional layers, max-pooling, and concatenation forms a robust framework for extracting meaningful semantic vectors from textual data.

5.2.4. Sentence decoder

It is essential to our model since it uses the previously created sentence as well as local visual data as a multimodal input. Its primary function is to generate the next sentence, which encompasses the second and third part of equation 6.2. Comprising just a single layer of LSTM, this decoder efficiently processes the information. Image V undergoes conversion to become input for this LSTM layer. Simultaneously, the encoded representation of the preceding sentence serves as a guide for our model to construct the subsequent sentence. This process is iteratively repeated until an empty sentence is generated, signaling the conclusion of the paragraph. This meticulous approach ensures the coherence and context consistency within the paragraph, aligning with our overarching objectives.

To focus different sentences to different regions of images and learn the relation between sentences, sentence-based visual attention model [60] is proposed. To capture the semantic features from the previous sentence and the regional visual representations, we employ a two-step process. Initially, these inputs are passed through a fully connected layer, followed by a SoftMax layer. This operation enables us to obtain an attention [61] distribution across a total of $k = 196$

distinct image regions. In our model, we employ an attention mechanism, denoted as 'a', which is formulated as follows:

$$b = S_{\text{att}} \tanh(S_q q + S_w w_1^i) \quad , \quad (6.4)$$

where $q \in \mathbb{R}^{a_v \times i}$ represents the regional visual features learned by the image encoder, while $q \in \mathbb{R}^{a_w}$ stands for the encoding of the preceding sentence. The parameters $S_{\text{att}} \in \mathbb{R}^{1 \times i}$, $S_v \in \mathbb{R}^{i \times a_v}$, and $S_w \in \mathbb{R}^{i \times d_w}$ are essential components of the attention network. Additionally, $1_i \in \mathbb{R}^{1 \times i}$ is a vector composed entirely of ones. Here, a_v is designated as 1024, representing the dimensionality of the regional visual feature, while a_w denotes the dimension of the sentence feature. Specifically, a_w takes on a value of 2048 for the Bi-LSTM configuration and 3072 for the CNN sentence encoder. Subsequently, we proceed to normalize this attention mechanism across all regions to obtain the attention distribution.

$$\theta_i = \frac{\exp(b_i)}{\sum_i \exp(b_i)} \quad , \quad (6.5)$$

where θ_i represents the attention weight for the i^{th} region, computed using the exponential function and normalized by the sum of all exponentials. Finally, the weighted visual representation ' v_{att} ' is calculated by aggregating the regional features:

$$v_{\text{att}} = \sum_{k=1}^i \theta_k v_k. \quad (6.6)$$

In the sentence decoder, this v_{att} serves as the input. The attention model dynamically focuses on different areas of the image as we construct different sentences, considering the context that the sentence before it provides. Our method uses a mechanism to remove parts of the image or features that don't relate to the present sentences. This selective process ensures that the model focuses on the most pertinent information, reducing the risk of overfitting to the semantic input.

Adam optimizer trains our model. The learning rate decays by a factor of 0.1 every 10 epochs, with the initial learning rate set at 1e-4. For training, the batch size is 32. We implement a teacher-forcing policy during the training, meaning that we constantly feed our decoder words or sentences that are ground truth for the generation in the next time. Greedy search is utilized during testing to produce words and sentences in each timestamp. The decoder will receive previously created words or sentences as input for the subsequent word or sentence. Until it produces an empty sentence, the recurrent generative model will continue to produce sentences. Through collaborative end-to-end training, all modules are trained by reducing cross-entropy loss.

5.3. Summary

In this chapter we discussed about the basics of multimodal learning, datatypes, application and challenges faced while making a multimodal model also this section discusses about the methodology for a multimodal learning system is detailed, focusing on the creation of radiology reports from medical images. The proposed model employs an encoder-decoder architecture,

utilizing a pre-trained ResNet-152 model for image processing and extracting both local and global visual details. The paragraph generation model involves a dedicated sentence generation model and a recurrent generation network that iteratively combines visual and semantic features to generate detailed findings paragraphs. The methodology includes a sentence encoder using a 1D Convolutional Neural Network for semantic vector extraction from textual descriptions and a sentence decoder, employing a single-layer LSTM and an attention mechanism to focus on different regions of images for each sentence. The overall approach ensures coherence and context consistency within the generated paragraphs. Training involves teacher-forcing, and the model is optimized using the Adam optimizer with a decaying learning rate. The end-to-end training process minimizes cross-entropy loss, ultimately producing detailed and contextually rich radiology reports from medical images.

CHAPTER 6

6. Experiments

6.1. Setup

Here we used the Indiana University Chest X-ray dataset [37] along with their reports. We only took the frontal images of the chest x-ray and took impression and finding parts of the textual report. Since the textual form of the report was in XML format, we did some data preprocessing in it. At first, we extracted the impression and findings of the chest x-ray report and split it into train, validation set in JSON format. After that, we create a vocabulary for impression and findings. Here, we pad each sentence with start and end tokens which are represented as <start> and <end>. After all this preprocessing we save a pickle file for vocabulary which is used during the training period.

Our training is based on the Pytorch framework where our model consists of an Image encoder known as EncoderCNN, an Impression decoder, a Sentence Encoder, and Attention Decoder. Image Encoder compromises ResNet152 pre-trained weights from ImageNet where we extract feature vectors from input images. In the impression decoder, we set the hyperparameter and build the layer for it. Here visual embedding, word embedding is done with the presence of embedding size, vocabulary size, and number of layers including global and local features. Also, here visual feature is combined with a semantic sentence vector to get hidden and cell state. This is the main part used for encoding where we get the initial sentence. This part is the Initial sentence generator block of our system architecture. After that, we have got a sentence encoder which consists of a CNN layer. Here we also set the hyper-parameter and build the layers for the sentence encoder. Our Attention decoder model consists of LSTM and a 1D convolution layer. Adam optimizer [62] trains our model for 80 epochs. The learning rate decays by a factor of 0.1 every 10 epochs, with the initial learning rate set at $1e-4$. For training, the batch size is 32. We implement a teacher-forcing policy during the training, meaning that we constantly feed our decoder words or sentences that are ground truth for the generation the next time. Greedy search [63] is utilized during testing to produce words and sentences in each timestamp. The decoder will receive previously created words or sentences as input for the subsequent word or sentence. Until it produces an empty sentence, the recurrent generative model will continue to produce sentences. Through collaborative end-to-end training, all modules are trained by reducing cross-entropy loss. The GPU used for the training was Volta. It took about three hours to complete the training process. Results were based on Natural Language Generation (NLG) Metrics.

6.2. Results

We measured the performance based on BLEU1, BLEU2, BLEU3, BLEU4, ROUGE, and CIDEr on three observations: Impression (I), Finding (F), and both Impression and Finding (I+F). The best score for BLEU1 (0.4424), BLEU2 (0.2923), BLEU3 (0.207), and BLEU4 (0.1464) was achieved when both Impression and Finding was accounted whereas the best ROUGE and CIDEr value of 0.5014 and 1.3678, respectively, was observed with Impression only.

Table 1 Evaluation metrics for Impression, Finding and Impression and finding.

	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE	CIDEr
Impression (I)	0.266	0.1878	0.1417	0.1066	0.5014	1.3678
Finding (F)	0.4328	0.2831	0.1962	0.1395	0.317	0.2312
Both (I + F)	0.4424	0.2923	0.207	0.1464	0.3396	0.2268

6.3. Comparison

We conducted a comprehensive comparison of our proposed approaches with several contemporary models dedicated to medical report creation and image captioning. The models encompass a spectrum ranging from advanced state-of-the-art methodologies to simpler baseline models. The 1-NN (1-Nearest Neighbor) model involves determining the nearest neighbor in the image embedding space of the training set when presented with a test image, and the corresponding report of this nearest neighbor is assigned as the result for the test image. Show and Tell (S&T) [18], a benchmark method for image captioning, and Show, Attend, and Tell (SA&T) [61], a model integrating attention mechanisms to enhance caption generation, were included in the comparison. TieNet [25], presumed to offer unique features or methods in the realm of medical report generation or image captioning, was also part of the evaluation. Additionally, a reinforcement learning-based CNN-RNN-RNN model proposed by Liu, Guanxiong, et al. [64] for automated medical report generation was considered in our comparative analysis. The evaluation was conducted on the Open-I dataset, employing the same set of evaluation metrics. The results, represented by the obtained scores for respective metrics, are presented in the tabulated form below for comprehensive assessment.

Table 2 Evaluation of generated reports on our testing set using BLEU, ROUGE, and CIDEr NLG metrics. We compare our models with five baseline models including a baseline implementation of the hierarchical generation reinforcement model.

Models	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE	CIDEr
1-NN	0.232	0.116	0.051	0.018	0.201	0.728
S&T[18]	0.265	0.157	0.105	0.073	0.306	0.926
SA&T [25]	0.328	0.195	0.123	0.080	0.313	1.276
TieNet [25]	0.330	0.194	0.124	0.081	0.311	1.334
CNN-RNN (NLG) [25]	0.369	0.246	0.171	0.115	0.244	0.036
Ours (I+F)	0.4424	0.2923	0.207	0.1464	0.3396	0.2268

Our model generates output of findings and impression with image id respectively. Sample output is shown in Figure 20.



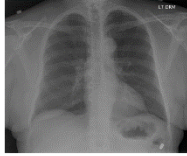

Chest X-Ray Images	Ground Truth	Predicted Output
	The cardiomeastinal silhouette and pulmonary vasculature are within normal limits. There is no pneumothorax or pleural effusion. There are no focal areas of consolidation. Cholecystectomy clips are present. Small T-spine osteophytes. There is biapical pleural thickening, unchanged from prior. Mildly hyperexpanded lungs.	the heart is normal in size . the mediastinum is unremarkable . the lungs are clear . there is no pleural effusion or pneumothorax . there is no focal air space opacity to suggest a pneumonia . there are mild degenerative changes of the spine .
	The cardiomeastinal silhouette is normal in size and contour. No focal consolidation, pneumothorax or large pleural effusion. Normal XXXX. XXXX cholecystectomy.	heart size within normal limits . no focal alveolar consolidation , no definite pleural effusion seen . no typical findings of pulmonary edema .
	No focal consolidation, pneumothorax, or pleural effusion. Cardiomeastinal silhouette unremarkable. Stable bilateral calcified granulomas/lymph XXXX. A bullet is present in the posterior soft tissues of the left chest wall, stable compared to prior examination.	the heart size and mediastinal contours appear within normal limits . no focal airspace consolidation , pleural effusion or pneumothorax . no acute bony abnormalities .
	Lungs are clear. There is no pleural effusion or pneumothorax. The heart and mediastinum are normal. The skeletal structures and soft tissues are normal.	the heart is normal in size . the mediastinum is unremarkable . the lungs are clear . there is no pleural effusion or pneumothorax . there is no focal air space opacity to suggest a pneumonia . there is a calcified granuloma in the right upper lobe . there are a calcified granuloma is present in the right upper

Figure 20 Sample output with ground truth and predicted output

```

CR359_2M-1765-0001-0001
Impression: left basilar patchy opacities, which may represent atelectasis or infection.
Findings: normal cardiomeastinal contours. no pneumothorax or large pleural effusions. left basilar patchy opacities. small hiatal hernia.
CR476_2M-2101-1001
Impression: no acute disease.
Findings: the lungs are clear. no pleural effusion is identified. the heart is normal. there are calcifications of the aortic aorta. the skeletal structures are normal.
CR688_2M-2256-1001
Impression: nub 1, there are numerous air-filled dilated loops of small bowel over the mid abdomen, these findings are consistent with small bowel obstruction, chest 1, left basilar airspace disease, aortic atelectasis.
Findings: nub 1 centered over the mid abdomen there are multiple air-filled dilated loops of small bowel measuring the aorta which measure up to about 3.7 cm in diameter. there is also an extremely dilated aorta in the same region which measures 5.9 cm in diameter. there is extensive soft tissue pannus prior abdominal surgery. chest. there is aortic left basilar opacity. no visualized pneumothorax. the heart size is normal. there is mild elevation of the left hemidiaphragm. there are no large pleural effusions. there is thickening of the fissure.
CR637_2M-2361-1001
Impression: no acute cardiopulmonary process.
Findings: cardiomeastinal silhouette is within normal limits. lungs are clear without areas of focal consolidation. right hilar calcifications aortic sequela of prior granulomatous disease. no pneumothorax or large pleural effusion. no acute bone abnormality.
CR2679_2M-1631-1001
Impression: normal chest exam.
Findings: normal heart. clear lungs. no pneumothorax. no pleural effusion.
CR3587_2M-1972-1001
Impression: 1, no acute cardiopulmonary abnormality, 2, abnormal configuration of the heart and mediastinum suggestive of right aortic aorta versus dextrocardia.
Findings: the lungs are clear bilaterally. specifically, no evidence of focal consolidation, pneumothorax, or pleural effusion. cardio mediastinal suggests possible right aorta versus dextrocardia. visualized osseous structures of the thorax are without acute abnormality.
CR302_2M-2405-1001
Impression: stable changes of ccpp.
Findings: there is interstitial thickening bilaterally, more prominent in the bases. the cardiomeastinal silhouette is normal in size and appearance. there is hyperexpansion. no aortic infiltrates. two bullae are seen in the right upper lung. small calcified granuloma stable from prior exam.
CR2972_2M-0706-1001
Impression: stable. nonenlarged cardiomeastinal silhouette. left upper lobe calcified granuloma noted, epigastric and right upper quadrant postsurgical changes. interval increased bilateral interstitial opacities, with probable left lower lobe infiltrate.
Findings: stable, nonenlarged cardiomeastinal silhouette. left upper lobe calcified granuloma noted. epigastric and right upper quadrant postsurgical changes. interval increased bilateral interstitial opacities, with probable left lower lobe infiltrate.
CR2821_2M-1244-1001
Impression: no acute cardiopulmonary findings.
Findings: cardiomeastinal silhouette and pulmonary vasculature are within normal limits. aortic calcifications and tortuosity. lungs are clear. no pneumothorax or pleural effusion. no acute osseous findings. degenerative changes of the thoracic spine.
CR790_2M-2329-1001
Impression: no acute disease.
Findings: both lungs are clear and expanded. heart and mediastinum normal. surgical clips are in the epigastrium of the abdomen.

```

Figure 21 Sample output generated in txt file.

6.4. Discussion

Here we developed a robust deep learning model for generating medical reports from chest X-ray images, utilizing the Indiana University Chest X-ray dataset. Through meticulous preprocessing, we extracted impressions and findings from XML-formatted reports, and our model, composed of an Image Encoder, Impression Decoder, Sentence Encoder, and Attention Decoder, demonstrated efficacy in encoding and decoding relevant information. Training on a Volta GPU using pre-trained ResNet152 weights from ImageNet, the model optimized cross-entropy loss over 80 epochs with an Adam optimizer and employed a teacher-forcing policy during training and a greedy search strategy during testing. Comprehensive performance evaluation, utilizing BLEU1-4, ROUGE, and CIDEr metrics, revealed superior results when considering both impressions and findings. In a comparative analysis against contemporary models, including 1-NN, Show and Tell, Show, Attend, and Tell, TieNet, and a reinforcement learning-based CNN-RNN-RNN model, our model consistently outperformed across all metrics, affirming its efficacy in medical report generation. Sample outputs further illustrated the model's proficiency, emphasizing its potential for accurate and contextually relevant medical reporting in clinical applications.

6.5. Summary

This section discusses a deep learning model which is developed for medical report generation from chest X-ray images, using the Indiana University Chest X-ray dataset. Extensive data preprocessing involved extracting impressions and findings from XML-formatted reports and creating a vocabulary for the model. The architecture included an Image Encoder (ResNet152), Impression Decoder, Sentence Encoder, and Attention Decoder, trained collaboratively over 80 epochs using PyTorch and an Adam optimizer. The model demonstrated superior performance, particularly when considering both impressions and findings, as evaluated by BLEU1-4, ROUGE, and CIDEr metrics. Comparative analyses against contemporary models, ranging from simpler baselines to advanced methodologies, consistently showcased the model's excellence. Sample outputs further demonstrated the model's proficiency in generating accurate and contextually relevant medical reports, indicating its potential impact in clinical applications.

CHAPTER 7

7. Conclusion

In this study, our primary objective has been the development of a comprehensive medical reporting system, particularly focusing on radiology impressions. The attention-based model we've proposed, incorporating multimodal inputs such as text and images, has proven effective in generating detailed reports. However, challenges arise in accurately distinguishing abnormal cases, potentially due to constraints in our training dataset, particularly the limited samples for abnormal cases, and inherent inconsistencies within the original ground truth reports. Furthermore, the model encounters difficulty in generating entirely new sentences, indicating the need for a more extensive and meticulously annotated dataset, along with innovative training strategies. Addressing these limitations requires a focus on keyword accuracy, syntactic correctness, and grammatical accuracy, prompting the exploration of new evaluation metrics. Despite these challenges, our attention-based model stands as a valuable component in computer-aided reporting systems, providing radiologists with tools to make informed decisions based on detailed and contextually rich reports, especially within the context of chest radiography.

To enhance the model's efficiency, one key avenue involves augmenting the training dataset with a more extensive collection of cases, with a particular emphasis on abnormal cases. Additionally, exploring the applicability of different chest X-ray datasets could further refine the model's performance and generalization. Also, the concept of Active learning can be introduced with this framework because active learning is a subfield of machine learning that allows model to perform good on limited available limited dataset if model have some role in selecting the data it wants to learn from[65]. Moreover, extending our experiments to include multilingual datasets and enabling the generation of radiology reports in various languages presents an exciting prospect for future advancements in this field. Addressing the challenges of generating entirely new sentences demands a meticulous approach, necessitating a combination of innovative training strategies and the acquisition of a more diverse and expansive dataset.

References

- [1] KC Santosh and S. Antani, "Automated chest x-ray screening: Can lung region symmetry help detect pulmonary abnormalities?," *IEEE Trans Med Imaging*, vol. 37, no. 5, 2018, doi: 10.1109/TMI.2017.2775636.
- [2] KC Santosh, S. Vajda, S. Antani, and G. R. Thoma, "Edge map analysis in chest X-rays for automatic pulmonary abnormality screening," *Int J Comput Assist Radiol Surg*, vol. 11, no. 9, 2016, doi: 10.1007/s11548-016-1359-6.
- [3] Md. K. Mahbub, M. Biswas, L. Gaur, F. Alenezi, and KC Santosh, "Deep features to detect pulmonary abnormalities in chest X-rays due to infectious diseaseX: Covid-19, pneumonia, and tuberculosis," *Inf Sci (N Y)*, vol. 592, pp. 389–401, May 2022, doi: 10.1016/j.ins.2022.01.062.
- [4] J. B. Soriano *et al.*, "Prevalence and attributable health burden of chronic respiratory diseases, 1990–2017: a systematic analysis for the Global Burden of Disease Study 2017," *Lancet Respir Med*, vol. 8, no. 6, 2020, doi: 10.1016/S2213-2600(20)30105-3.
- [5] D. Das, KC Santosh, and U. Pal, "Cross-population train/test deep learning model: Abnormality screening in chest x-rays," in *Proceedings - IEEE Symposium on Computer-Based Medical Systems*, 2020. doi: 10.1109/CBMS49503.2020.00103.
- [6] KC Santosh, S. Allu, S. Rajaraman, and S. Antani, "Advances in Deep Learning for Tuberculosis Screening using Chest X-rays: The Last 5 Years Review," *J Med Syst*, vol. 46, no. 11, 2022, doi: 10.1007/s10916-022-01870-8.
- [7] KC Santosh, S. Ghosh, and D. Ghoshroy, "Deep Learning for Covid-19 Screening Using Chest X-Rays in 2020: A Systematic Review," *Intern J Pattern Recognit Artif Intell*, vol. 36, no. 5, 2022, doi: 10.1142/S0218001422520103.
- [8] KC Santosh and S. Ghosh, "Covid-19 Imaging Tools: How Big Data is Big?," *J Med Syst*, vol. 45, no. 7, 2021, doi: 10.1007/s10916-021-01747-2.
- [9] KC Santosh, "AI-Driven Tools for Coronavirus Outbreak: Need of Active Learning and Cross-Population Train/Test Models on Multitudinal/Multimodal Data," *J Med Syst*, vol. 44, no. 5, p. 93, May 2020, doi: 10.1007/s10916-020-01562-1.

- [10] A. Makkar and KC Santosh, "SecureFed: federated learning empowered medical imaging technique to analyze lung abnormalities in chest X-rays," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 8, 2023, doi: 10.1007/s13042-023-01789-7.
- [11] T. Pang, P. Li, and L. Zhao, "A survey on automatic generation of medical imaging reports based on deep learning," *BioMedical Engineering Online*, vol. 22, no. 1. 2023. doi: 10.1186/s12938-023-01113-y.
- [12] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, "A convolutional encoder model for neural machine translation," in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2017. doi: 10.18653/v1/P17-1012.
- [13] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. doi: 10.1109/CVPR.2015.7298935.
- [14] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *30th International Conference on Machine Learning, ICML 2013*, 2013.
- [15] J. Krause, J. Johnson, R. Krishna, and L. Fei-Fei, "A hierarchical approach for generating descriptive image paragraphs," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. doi: 10.1109/CVPR.2017.356.
- [16] S. Vajda *et al.*, "Feature Selection for Automatic Tuberculosis Screening in Frontal Chest Radiographs," *J Med Syst*, vol. 42, no. 8, 2018, doi: 10.1007/s10916-018-0991-9.
- [17] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42. 2017. doi: 10.1016/j.media.2017.07.005.
- [18] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge," *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 4, 2017, doi: 10.1109/TPAMI.2016.2587640.
- [19] Y. Xue *et al.*, "Multimodal recurrent model with attention for automated radiology report generation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. doi: 10.1007/978-3-030-00928-1_52.
- [20] F. Liu, C. Yin, X. Wu, S. Ge, P. Zhang, and X. Sun, "Contrastive Attention for Automatic Chest X-ray Report Generation," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021. doi: 10.18653/v1/2021.findings-acl.23.
- [21] T. Schlegl, S. M. Waldstein, W. D. Vogl, U. Schmidt-Erfurth, and G. Langs, "Predicting semantic descriptions from medical images with convolutional neural networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015. doi: 10.1007/978-3-319-19992-4_34.

- [22] M. Moradi, A. Madani, Y. Gur, Y. Guo, and T. Syeda-Mahmood, “Bimodal network architectures for automatic generation of image annotation from text,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. doi: 10.1007/978-3-030-00928-1_51.
- [23] H. C. Shin, K. Roberts, L. Lu, D. Demner-Fushman, J. Yao, and R. M. Summers, “Learning to Read Chest X-Rays: Recurrent Neural Cascade Model for Automated Image Annotation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. doi: 10.1109/CVPR.2016.274.
- [24] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. doi: 10.1109/CVPR.2017.369.
- [25] X. Wang, Y. Peng, L. Lu, Z. Lu, and R. M. Summers, “TieNet: Text-Image Embedding Network for Common Thorax Disease Classification and Reporting in Chest X-Rays,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018. doi: 10.1109/CVPR.2018.00943.
- [26] B. Jing, P. Xie, and E. P. Xing, “On the automatic generation of medical imaging reports,” in *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2018. doi: 10.18653/v1/p18-1240.
- [27] C. Y. Li, Z. Hu, X. Liang, and E. P. Xing, “Hybrid retrieval-generation reinforced agent for medical image report generation,” in *Advances in Neural Information Processing Systems*, 2018.
- [28] J. Johnson, A. Karpathy, and L. Fei-Fei, “DenseCap: Fully convolutional localization networks for dense captioning,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. doi: 10.1109/CVPR.2016.494.
- [29] Z. Zhang, Y. Xie, F. Xing, M. McGough, and L. Yang, “MDNet: A semantically and visually interpretable medical image diagnosis network,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. doi: 10.1109/CVPR.2017.378.
- [30] J. Yuan, H. Liao, R. Luo, and J. Luo, “Automatic Radiology Report Generation Based on Multi-view Image Fusion and Medical Concept Enrichment,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019. doi: 10.1007/978-3-030-32226-7_80.
- [31] J. Irvin *et al.*, “CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison,” in *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI*

- Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, 2019. doi: 10.1609/aaai.v33i01.3301590.
- [32] P. Messina *et al.*, “A Survey on Deep Learning and Explainability for Automatic Image-based Medical Report Generation,” *ArXiv Preprint*, 2020.
- [33] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2002.
- [34] C. Y. Lin, “Rouge: A package for automatic evaluation of summaries,” *Proceedings of the workshop on text summarization branches out (WAS 2004)*, no. 1, 2004.
- [35] R. Vedantam, C. L. Zitnick, and D. Parikh, “CIDEr: Consensus-based image description evaluation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. doi: 10.1109/CVPR.2015.7299087.
- [36] “CodaLab - Competition.” Accessed: Nov. 27, 2023. [Online]. Available: <https://competitions.codalab.org/competitions/3221>
- [37] D. Demner-Fushman *et al.*, “Preparing a collection of radiology examinations for distribution and retrieval,” *Journal of the American Medical Informatics Association*, vol. 23, no. 2, 2016, doi: 10.1093/jamia/ocv080.
- [38] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychol Rev*, vol. 65, no. 6, 1958, doi: 10.1037/h0042519.
- [39] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete, “A survey on modern trainable activation functions,” *Neural Networks*, vol. 138. 2021. doi: 10.1016/j.neunet.2021.01.026.
- [40] D. B. Mulindwa and S. Du, “An n-Sigmoid Activation Function to Improve the Squeeze-and-Excitation for 2D and 3D Deep Networks,” *Electronics (Switzerland)*, vol. 12, no. 4, 2023, doi: 10.3390/electronics12040911.
- [41] Wafaa M. Taha, Abbas H. Kadhim, Raad A. Hameed, and M. S. M. Noorani, “New modified tanh-function method for nonlinear evolution equations,” *Tikrit Journal of Pure Science*, vol. 21, no. 6, 2023, doi: 10.25130/tjps.v21i6.1100.
- [42] Y. Bai, “RELU-Function and Derived Function Review,” *SHS Web of Conferences*, vol. 144, 2022, doi: 10.1051/shsconf/202214402006.
- [43] S. Singh, “ELU as an Activation Function in Neural Networks .,” *Deep Learning University*. 2020.
- [44] J. Brownlee, “Softmax Activation Function with Python,” *Mahine Learning Mastery*.
- [45] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017*, 2017. doi: 10.1109/ICEngTechnol.2017.8308186.

- [46] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *37th International Conference on Machine Learning, ICML 2020*, 2020.
- [47] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, 1998, doi: 10.1109/5.726791.
- [48] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [49] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. doi: 10.1109/CVPR.2015.7298594.
- [50] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. doi: 10.1109/CVPR.2017.195.
- [51] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. doi: 10.1109/CVPR.2016.90.
- [52] A. G. Howard *et al.*, “MobileNets,” *arXiv preprint arXiv:1704.04861*, 2017.
- [53] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. doi: 10.1109/CVPR.2017.243.
- [54] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int J Comput Vis*, vol. 115, no. 3, 2015, doi: 10.1007/s11263-015-0816-y.
- [55] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network,” *Physica D*, vol. 404, 2020, doi: 10.1016/j.physd.2019.132306.
- [56] G. Van Houdt, C. Mosquera, and G. Nápoles, “A review on the long short-term memory model,” *Artif Intell Rev*, vol. 53, no. 8, 2020, doi: 10.1007/s10462-020-09838-1.
- [57] C. Janiesch, P. Zschech, and K. Heinrich, “Machine learning and deep learning,” *Electronic Markets*, vol. 31, no. 3, 2021, doi: 10.1007/s12525-021-00475-2.
- [58] M. Weinmann, “Visual Features—From Early Concepts to Modern Computer Vision,” 2013. doi: 10.1007/978-1-4471-5520-1_1.
- [59] A. Zadeh, P. P. Liang, and L. P. Morency, “Foundations of Multimodal Co-learning: Multimodal Co-learning,” *Information Fusion*, vol. 64, 2020, doi: 10.1016/j.inffus.2020.06.001.

- [60] J. Lu, C. Xiong, D. Parikh, and R. Socher, “Knowing when to look: Adaptive attention via a visual sentinel for image captioning,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. doi: 10.1109/CVPR.2017.345.
- [61] K. Xu *et al.*, “Show, attend and tell: Neural image caption generation with visual attention,” in *32nd International Conference on Machine Learning, ICML 2015*, 2015.
- [62] Z. Zhang, “Improved Adam Optimizer for Deep Neural Networks,” in *2018 IEEE/ACM 26th International Symposium on Quality of Service, IWQoS 2018*, 2019. doi: 10.1109/IWQoS.2018.8624183.
- [63] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, “Sequence level training with recurrent neural networks,” in *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2016.
- [64] G. Liu *et al.*, “Clinically Accurate Chest X-Ray Report Generation,” in *Proceedings of Machine Learning Research*, 2019.
- [65] KC Santosh and S. Nakarmi, *Active Learning to Minimize the Possible Risk of Future Epidemics*. Singapore: Springer Nature Singapore, 2023. doi: 10.1007/978-981-99-7442-9.