CONVERSATIONAL IMAGE RECOGNITION CHATBOT

A PROJECT WORK REPORT

Submitted by

JEYAMURUGAN G	**(71812101113)**
SANTHOSHKUMAR K	**(71812101223)**
SESHADHRI V	**(71812101227)**

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

SRI RAMAKRISHNA ENGINEERING COLLEGE, COIMBATORE

ANNA UNIVERSITY : CHENNAI 600025

MAY 2025

SRI RAMAKRISHNA ENGINEERING COLLEGE COIMBATORE

ANNA UNIVERSITY: CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project work report entitled **“CONVERSATIONAL IMAGE RECOGNITION CHATBOT”** is the bonafide work of **JEYAMURUGAN G (71812101113), SANTHOSHKUMAR K (71812101223), SESHADHRI V (71812101227),** who carried out the **20CS298 Project Work** under my supervision.

SIGNATURE

SUPERVISOR

Dr. P. Mathiyalagan, M.E., Ph.D.,
Associate Professor,
Department of Computer Science and Engineering,
Sri Ramakrishna Engineering College,
Coimbatore – 641022.

SIGNATURE

HEAD OF THE DEPARTMENT

Dr. M. S. Geetha Devasena, M.E., Ph.D.,
Professor and Head,
Department of Computer Science and Engineering,
Sri Ramakrishna Engineering College,
Coimbatore – 641022.

Submitted for Project Work Viva Voce Examination held on ________________

Internal Examiner **External Examiner**

DECLARATION

We affirm that the project work titled **“CONVERSATIONAL IMAGE RECOGNITION CHATBOT ”** being submitted in partial fulfilment for the award of Bachelor of Engineering is the original work carried out by us. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

---------------------- ---------------------- -----------------------

(Signature of the Candidates)

JEYAMURUGAN G	**(71812101113)**
SANTHOSHKUMAR K	**(71812101223)**
SESHADHRI V	**(71812101227)**

I certify that the declaration made above by the candidates is true.

(Signature of the Supervisor)

Dr. P. Mathiyalagan, M.E., Ph.D.,

Associate Professor,

Department of Computer Science and Engineering.

ACKNOWLEDGEMENT

We have immense pleasure in expressing our wholehearted thankfulness to **Shri. R. Sundar,** Managing Trustee, SNR Sons Charitable Trust and **Shri. S. Narendran,** Joint Managing Trustee, SNR Sons Charitable Trust for giving us the opportunity to study in our esteemed college and for very generously providing more than enough infrastructural facilities for us to get molded as a complete engineer.

We wish to express our profound and sincere gratitude to **Dr. A. Soundarrajan,** Principal, for inspiring us with his Engineering Wisdom. We also wish to record our heartfelt thanks for motivating and guiding us to become Industry ready engineers with inter-disciplinary knowledge and skillset with his multifaceted personality.

We extend our indebted thankfulness to **Dr. M. S. Geetha Devasena,** Professor and Head, Department of Computer Science and Engineering, for guiding and helping us in all our activities to develop confidence and skills required to meet the challenges of the industry. We also express our gratitude for giving us support and guidance to complete the project duly.

We owe our deep gratitude to our project supervisor **Dr. P. Mathiyalagan**, Associate Professor, Department of Computer Science and Engineering, who took keen interest in our project work and guided us all along with all the necessary inputs required to complete the project work.

We express our sincere thanks to our project coordinator **Mrs. M. Indira Priyadharshini** , Assistant Professor and **Mr. R. S. Vishnu Durai** , Assistant Professor, Department of Computer Science and Engineering evaluators and teaching faculty members of the department for evaluating the project and providing valuable suggestions for improvements.

We also thank all the supporting staff members of our department for their help in making this project a successful one.

TABLE OF CONTENTS

ABSTRACT

A Vision language model has been developed to automate the generation clinical captions for chest X-ray images and explained detailly using Large Language model . Traditional radiological workflows depends on manual interpretation, which is slow, subjective, and resource-intensive and also when it handling large volumes of imaging data. Conventional methods are also struggling to capture the nuanced association between visual pattern and clinical terminology which limits their effectiveness in automated diagnosis. To address these limitations, a fine-tuned vision-language architecture based on LLaMA 3.2 Vision which stands for Large Language Model Meta AI was implemented using Low-Rank Adaptation (LoRA) and 4-bit quantization techniques. The Unsloth library was used to enable efficient model training using limited hardware. The ROCOv2 dataset which stands for Radiology Objects in Context , consists of chest X-ray images paired with medical captions, has been used as the primary data source, enabling the model to learn clinically relevant visual-text embeddings . The training pipeline which is involves in the semantic alignment of image features in chest x rays and its respective textual data of scan report for the task such as image captioning and visual question answering.

சுருக்கம்

மார்பு எக்ஸ்-ரே படங்களுக்கான தலைமுறை மருத்துவ தலைப்புகளை தானியக்கமாக்குவதற்காக ஒரு பார்வை மொழி மாதிரி உருவாக்கப்பட்டுள்ளது மற்றும் பெரிய மொழி மாதிரியைப் பயன்படுத்தி விரிவாக விளக்கப்பட்டுள்ளது. பாரம்பரிய கதிரியக்க பணிப்பாய்வுகள் கையேடு விளக்கத்தைப் பொறுத்தது, இது மெதுவானது, அகநிலை மற்றும் வள-தீவிரமானது மற்றும் பெரிய அளவிலான இமேஜிங் தரவைக் கையாளும் போது. தானியங்கி நோயறிதலில் அவற்றின் செயல்திறனைக் கட்டுப்படுத்தும் காட்சி முறைக்கும் மருத்துவ சொற்களஞ்சியத்திற்கும் இடையிலான நுணுக்கமான தொடர்பைப் பிடிக்க வழக்கமான முறைகளும் போராடி வருகின்றன. இந்த வரம்புகளை நிவர்த்தி செய்ய, பெரிய மொழி மாதிரி மெட்டா AI ஐக் குறிக்கும் LLaMA 3.2 பார்வையை அடிப்படையாகக் கொண்ட ஒரு நேர்த்தியான பார்வை-மொழி கட்டமைப்பு குறைந்த-தர தழுவல் (LoRA) மற்றும் 4-பிட் அளவீட்டு நுட்பங்களைப் பயன்படுத்தி செயல்படுத்தப்பட்டது. வரையறுக்கப்பட்ட வன்பொருளைப் பயன்படுத்தி திறமையான மாதிரி பயிற்சியை செயல்படுத்த Unsloth நூலகம் பயன்படுத்தப்பட்டது. சூழலில் ரேடியாலஜி பொருள்களைக் குறிக்கும் ROCOv2 தரவுத்தொகுப்பு, மருத்துவ தலைப்புகளுடன் இணைக்கப்பட்ட மார்பு எக்ஸ்-ரே படங்களைக் கொண்டுள்ளது, இது முதன்மை தரவு மூலமாகப் பயன்படுத்தப்பட்டுள்ளது, இது மாதிரி மருத்துவ ரீதியாக பொருத்தமான காட்சி-உரை உட்பொதிப்புகளைக் கற்றுக்கொள்ள உதவுகிறது. மார்பு எக்ஸ்ரேக்களில் பட அம்சங்களின் சொற்பொருள் சீரமைப்பு மற்றும் பட தலைப்பு மற்றும் காட்சி கேள்வி பதில் போன்ற பணிக்கான ஸ்கேன் அறிக்கையின் அந்தந்த உரைத் தரவை உள்ளடக்கிய பயிற்சி குழாய். அளவீட்டு மாதிரி குறைக்கப்பட்ட GPU நினைவகத் தேவைகளுடன் பயனுள்ள தலைப்பு

உருவாக்கத்தை அடைய உதவுகிறது, வள-கட்டுப்படுத்தப்பட்ட சூழல்களில் அதன் நடைமுறை பயன்பாட்டை எடுத்துக்காட்டுகிறது என்பது சோதனை முடிவுகள் நிரூபிக்கப்பட்டுள்ளன. ஒற்றை கட்டமைப்பிற்குள் படம் மற்றும் மொழி புரிதலின் ஒருங்கிணைப்பு மருத்துவ கதிரியக்கவியல் பணிப்பாய்வுகளில் கண்டறியும் வேகம், துல்லியம் மற்றும் நிலைத்தன்மையை மேம்படுத்துவதில்
சாத்தியமான நன்மைகளை வழங்குகிறது.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

ABBREVIATION	EXPANSION
AI	Artificial intelligence
LLM	Large Language Model
ML	Machine Learning
LLaMA	Large Language Model Meta AI
LLaVA	Large Language and Vision Assistant
ROCOv2	Radiology Objects in COntext Version 2
ViT	Vision Transformer
Grad-CAM	Gradient-weighted Class Activation Mapping
CheXpert	Chest X-ray Expert Labels Dataset
CLIP	Contrastive Language–Image Pre-training
Gen AI	Generative Artificial Intelligence
BLEU	Bilingual Evaluation Understudy
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
RLHF	Reinforcement Learning from Human Feedback
UI	User Interface
API	Application Programming Interface
LoRA	Low-Rank Adaptation
MIMIC-CXR	Medical Information Mart for Intensive Care – Chest X-ray
Qwen2-VL	Qwen2 Vision-Language Model

CHAPTER 1

INTRODUCTION

1.1 Overview of Vision-Language Models in Radiology

Multimodal Artificial Intelligence is that which processes and understands data from many forms at once for example images, text, audio, or structured data. In the field of radiology this means the integration of imaging studies like X-rays, CT scans, MRIs with also the text from reports that go with them or other clinical notes. Also it has been the case that mostly AI models have worked with only one type of data at a time which in turn has limited what they are able to do and study. But we are seeing with the development of vision-language models like LLaMA 3.2 Vision that now we may design AI which is able to at the same time see and think about images and language.

As we see an increase in the complexity and size of radiological data sets we see out the need for tools which will enable radiologists in their interpretation and decision making. We see Multimodal AI as the solution which improves model's performance via cross modal learning. For example we have models which are trained on radiology images along with clinical notes which in turn are able to produce relevant descriptions for new images or to answer diagnostic questions, which in turn puts out early info. Our work is on developing such a system which we are training with state of the art methods which in turn makes the AI more efficient for use on limited hardware but at the same time we are seeing great performance in the medical field.

The integration of AI in radiology is beyond just automation; it is in the betterment of accuracy and consistency in image interpretation. We see in Radiology aware AI models which are trained on domain specific sets like ROCOv2 that they are able to pick up on the fine details of medical imaging and clinical language. Also these models play the role of virtual assistants for radiologists they bring to notice of issues in images, put forth differential diagnoses and also present draft reports. This project we are looking at a vision-language model fine tuned on radiological data which in turn is an attempt to put into practice what we have researched in the AI field into what is useful in the clinical setting.

1.2 Importance of Artificial Intelligence in Field of Radiology

Radiology is fundamental to contemporary medical imaging. Its importance is particularly evident in the timely diagnosis, follow-up, and treatment preparation for diseases like cancer, cardiology, and neurology. The shortage of competent radiologists, combined with the increasing need for radiological assessments, has made AI a critical adjunct tool. In particular, AI applications in under-resourced environments can alleviate burdens, reduce error rates, and expedite diagnostic processes.

The integration of AI into radiology isn't just about automation; it's about enhancing accuracy and consistency in image interpretation. Radiology-aware AI models, trained on domain-specific datasets like ROCOv2, can understand the nuances of medical imaging and clinical language. These models have the potential to function as virtual assistants for radiologists—highlighting abnormalities, suggesting differential diagnoses, and even generating draft reports. This project explores such a vision-language model fine-tuned on radiological data, aiming to bridge the gap between AI research and clinical utility.

Integrating image analysis and the understanding of text into a single model signifies advancement in technology. This is helpful in radiology because images are only useful when combined with the patient's history, symptoms, and previous examinations. In this project, we apply a small, easily adjustable multimodal model that is trainable on academic machines, yet provides outputs equal to that of clinical systems. The goal is to equip radiologists and healthcare practitioners with systems that leverage their skillset to enhance decision-making and improve patient care.

Also in that AI can play a role of transformation in the standardization of report generation and reduction of inter-observer variability which is a known issue in radiology. We will train our model on a variety of radiographic cases from ROCOv2 to present a system that which mirrors that of experienced radiologists' diagnostic methods while at the same time we bring in uniformity and efficiency to medical workflows. Also such a system we put forth as a decision support tool which in turn will support radiologists in high volume settings and at the same time see to it that we improve overall diagnostic quality.

1.3 Role of Artificial Intelligence

AI technology has progressed further in the field of healthcare, especially in diagnostic imaging, where deep learning algorithms have been utilized to identify complex conditions like tumors, fractures, and organ abnormalities on par with human specialists. With the application of AI, modern radiology is now able to perform organ segmentation, pathology classification, and triaging of cases based on automatic sifting of pixel intensity patterns that are imperceptible to human observers.

Transitioning to multimodal AI systems represents a leap in technology where a model can interpret an image and also "read" the provided text. This dual interpretation is particularly useful in radiology, where imaging is heavily reliant on a patient's previous history, complaints, and previous investigations. In our project, we use a multimodal model which is light and easily tunable at the microscopic level - even when using basic academic resources – while still achieving clinical-level results. The goal is to assist smarter technology that aid radiologists and medical professionals in exercising their clinical judgments resulting in better patient care.

Our project is dedicated to the development of such systems which we train with the best available techniques which in turn makes AI more effective on small scale hardware yet at the same time we see large performance results in medical settings. In this we use the ROCOv2 dataset a well known element in the medical imaging field which has a large set of radiology images along with expert created captions. That which in turn allows our model to develop in its understanding of the in depth relationships between what is seen in the images and the text which is used in clinical settings. Also by creating a small scale, flexible AI platform we aim to make multi modal intelligence a more accessible resource for medical institutions and researchers that may have limited computer resources.

1.4 Objective of the Project

The objective of the project is to develop an AI-driven report generation for Radiology Chest X rays , with the aims to design a resource-efficient multimodal AI model that comprehends and generates useful outputs using radiological image-text pairs.

Specifically, the goals are

- The goal is to use large vision-language models like LLaMA 3.2 Vision for radiology-specific tasks.
- Reduce computations using 4-bit quantization with no degradation in accuracy.
- Use LoRA to fine-tune pre-trained models more efficiently.
- Utilize the ROCOv2 dataset, which contains annotated radiological diagnostic images, to train the model.
- To determine the model's performance in the medical imaging domain for image captioning and visual question answering tasks.

Integrating AI in diagnostic imaging fulfills the growing demand for precise, timely, and the same quality of interpretive reports from medical images. As health care systems report greater work forces' pressure and we see a global deficit of radiologists in many areas, these systems step in to provide real time support by putting forth diagnosis, bringing to attention abnormal findings, and in some cases draft reports. The model we present in this project is a look at how AI may be used in radiology and also is a contribution to the future of clinical decision support which is at once more extensive in scale and wide in access, in the end which improve diagnostic accuracy and patient care.

In the end, the project aims to show that given the right dataset and optimization, multimodal AI models can be trained and deployed effectively. Such systems can act as clinical decision support systems for radiology departments, which can result in faster diagnosis, fewer errors and better quality of care.

CHAPTER 2

LITERATURE REVIEW

2.1 Chatbots and Large Language Models in Radiology

Reference: Bhayana, R. (2023). "Chatbots and Large Language Models in Radiology: A Practical Primer for Clinical and Research Applications."

According to this paper, large language models (LLMs) such as ChatGPT are being incorporated into radiology to assist with clinical work and research. These models can understand human language and generate human-like text. This makes them useful for summarising radiology reports, answering patient questions, and assisting doctors in making decisions. Also, they can be integrated into hospital systems to assist radiologists in handling high workloads and enhancing their communication with others.

The paper urges the use of good prompts for the models' training on radiology data, and carrying out rigorous safety testing for them. The website explains the necessity of clarification on how these tools work especially when doctors are using them to help in diagnoses. The paper also explains how models can assist with radiological research. They can help to find patterns in large imaging data sets, generate labels for data sets, and help in writing or summarising papers. The main point of the article is that chatbots and LLMs, when used properly, can provide great assistance and improve both clinical care and medical research.

2.2 Vision-Language Pretraining for Radiological Image Report Generation

Reference: Liu, P., et al. (2023). "Align before Fuse: Vision and Language Representation Learning for Radiology Report Generation."

Its focuses on how an AI model can generate medical reports using radiology images through vision and language inputs. The authors propose a solution where the model initially learns the image and text features individually and then joins them in a clever way. This way, the model learns about the image in the X-ray and the language used according to the report.

The researchers proposed a framework that aligns image and text data before fusing them using attention mechanisms. This procedure enables the model to link the object's features with its correct medical terminology. The model was tested on popular chest X-ray datasets, IU X-Ray and MIMIC-CXR. The two show that the model outperforms earlier ones in improving the quality of reports.

2.3 Towards a Unified Image-Text Pretraining Paradigm for Diverse Medical Imaging Modalities

Reference: Muhammad Uzair Khattak, Shahina Kunhimon, 2024" Towards a Unified Image-Text Pretraining Paradigm for Diverse Medical Imaging Modalities"

The framework uses contrastive learning techniques to match medical images with relevant texts. A shared embedding space and domain-specific encoders have been leveraged to understand and retrieve better. This method beats other models on many benchmarks — this shows that the method excels at learning strong, transferable features from imaging data. Because of the lucky part of multimodal pretraining, it requires less labeled data.

The work suggests a movement towards building larger and more generalizable vision-language architectures in medical AI. This methodology enables the utilization of single method tools over multiple imaging types with a reliable precision as the dependency on modality specific methods is eliminated. The findings indicate a strong capacity to enhance systems for automated reporting and clinical support, especially in environments that utilize mixed imaging types.

2.4 Next-generation Agentic AI for Transforming Healthcare

Reference : Nalan Karunanayake, 2024, "Next-generation Agentic AI for Transforming Healthcare"

Agentic AI refers to a new generation of AI that can think for itself and make its own decisions. Because of this, it can be very helpful in health care especially in radiology where doctors often need assistance making decisions. Agentic AI is different from regular AI tools. Unlike them, it has memory, which means it can remember past interactions, learn from them, and build on them. Using one AI could help doctors get suggestions based on a patient's full medical history, not just one scan or report.

Medical students can learn how to interpret scans or have an explanation of why a particular diagnosis makes sense. In research, it can be used to identify trends in extensive datasets or mimic uncommon medical situations that are challenging to encounter in reality. It is necessary to ensure that the technology is safe and reliable. Doctors should always check its suggestions and transparency should be ensured on how it makes its decisions.

2.5 RadAlign: Advancing Radiology Report Generation with Vision-Language Concept Alignment

Reference : D. Gu et al., 2025 "Advancing Radiology Report Generation with Vision-Language Concept Alignment"

RadAlign is a model that enhances the alignment of computer-generated radiology reports with the visual aspects of radiology handles. Classic models often fail to connect parts of an image to the relevant medical terminology, resulting in vague reports. RadAlign learns the visual details in scans and the type of language expert radiologists use in their notes. The model is more likely to offer elaborate descriptions that are medically pertinent, thanks to this alignment.

RadAlign shows a major strength in its contextualized understanding. Rather than simply identifying what’s in a scan, it can relate the findings to each other, an act that a human radiologist does instinctively. For instance, if there’s fluid gathered up and the lung is collapsed on a chest x-ray, the model knows that these two signs can be related to a particular condition and will describe both signs together. By adopting this approach, the report becomes clearer and more useful for the doctors, enabling them to have faith in it.

2.6 SERPENT-VLM: Self-Refining Radiology Report Generation Using Vision Language Models

Reference : M. N. Kapadnis et al., 2024 "Self-Refining Radiology Report Generation Using Vision Language Models"

SERPENT-VLM puts forth a model which improves its own output in a self reflective process which in turn it uses to refine radiology reports which it produces. This is based on the practice of radiologists who reevaluate their initial readings before final report. The model puts out a first version of the report from an image and then goes over it with a report which evaluates that report to put out a better version. What we see is that it does not stand by a single outcome. Instead it puts out many versions of the report and chooses the one which is the best medical option.

It has a self improvement component which puts out of play common issues we see in past models which include atypical observations and key factor omission. As a result the reports we get are not only more in sync with the image but also have a greater use of professional medical terminology. Also it improves accuracy and does a better job of2 do well across many types of imaging data. It does well on a variety of different datasets without requiring extensive re-training which makes it a asset for hospitals that use many types of scans. It also has the ability to improve itself and reduce errors which in turn raises the bar for clinicians which use automated tools for large scale report production.

2.7 RaDialog: A Large Vision-Language Model for Radiology Report Generation and Conversational Assistance

Reference : C. Pellegrini et al., 2023 "A Large Vision-Language Model for Radiology Report Generation and Conversational Assistance"

RaDialog is a large leap in the application of VLMs to real world radiology tasks. We present a model which not only produces accurate medical reports from radiology images but also aids in medical discussion, which in turn is useful for clinicians and in AI supported patient care. The model is trained on many different image and text pairs and also is designed to interpret complex imaging like CT and X ray. What makes RaDialog unique is that it has the ability to think through images like a radiologist and report back in everyday language which makes it very much at home in diagnostic and interactive medical settings.

It also contributes to reducing the documentation burden in healthcare by automating parts of the reporting process without compromising medical precision. Its conversational nature makes it adaptable for training environments, where students or junior doctors can ask questions and receive guided explanations. While still in development, early results show it performs competitively with expert radiologists in certain diagnostic tasks.

2.8 UniCrossAdapter: Multimodal Adaptation of CLIP for Radiology Report Generation

Reference : Y. Chen et al., 2025 "Multimodal Adaptation of CLIP for Radiology Report Generation"

UniCrossAdapter helps make vision-language models like CLIP better adapted for tasks in radiology. Because they were trained on general image-text pairs, standard CLIP models will likely miss some of the specific language and visual patterns in medical data. UniCrossAdapter solves this problem by providing a new connection for cross-visual language model components.

It adopts a modular system that modifies the vision encoder and the text decoder with cross-modal adapters, which are additional components that facilitate better translation of clinical images into medical text.

As a result, we can obtain more accurate and detailed radiology reports because the model will learn to focus on the important medical regions in the image and formulate appropriate medical language. UniCrossAdapter bridges visual features and medical terminology, linking general vision-language models to radiology. It performs well on multiple datasets and takes less time to train. Thus, it is ideal for developing AI medical tools that produce clear and clinically relevant reports.

2.9 Fine-Tuning Vision-Language Models using LoRA

Reference : G. Chutani, 2024"Fine-Tuning Vision-Language Models using LoRA"

Fine-tuning huge vision-language models is costly and time-consuming, specifically for medical tasks. It is efficient to train just a few parameters while keeping the weights of the original model frozen. Low-Rank Adaptation (LoRA) is capable of accomplishing this. It greatly cuts down costs and memory usage. LoRA allows developers to retrain already trained vision-language models like CLIP or LLaVA, in a radiological context, on smaller datasets, without loss via overfitting.

The model quickly learns new patterns by injecting low-rank matrices in the attention layers as part of this technique. A medical report can refer to the radiological specialty for further understanding of any term used in the report. This makes tuning personalized or institution-specific possible with no huge computational cost. LoRA works great for researchers who cannot improve the hardware of their medical AIs due to budget constraints or other reasons.

2.10 Vision-Language Model for Generating Textual Descriptions from Clinical Images

Reference : H. Chen et al., 2024 "Vision-Language Model for Generating Textual Descriptions from Clinical Images"

Vision and language models we have designed which interpret images and produce related text which is very much so in the medical field we see that play out. In our study we developed a system which looks at clinical images like chest X-rays and puts out medical descriptions. We use a structure which includes visual encoders and language decoders to association medical image elements with the right terminology. What we have here is a very useful model which lessens the work load on radiologists and at the same time gives out very consistent report quality. It also plays a role in early diagnosis, reports' workflow improvement and in training by putting forth report options for review. In practice we see these systems do well in very busy hospitals which have large volume of scans to be looked at quickly.

2.11 CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays

Reference : P. Rajpurkar et al., 2017 "Radiologist-Level Pneumonia Detection on Chest X-Rays"

CheXNet is a deep learning model designed to diagnose pneumonia from chest X-rays nearly as accurately as professional radiologists and utilizes a CNN architecture – a 121-layer DenseNet – trained on the ChestX-ray14 dataset containing more than 100,000 labeled X-ray images. The model analyzes these images for algorithmic patterns associated with pneumonia and is capable of making precise predictions, even when the findings are complicated, unclear, or intricate. One of the most striking achievements of this work is that AI can compete or even outperform humans in some given tasks.

CheXNet's impact has advanced the quality of healthcare services in regions with fewer radiologists available. It also enhanced the acceptance towards AI systems in the field of medicine by proving that deep learning algorithms could provide adequate assistance in a clinical setting. Additionally, this work is a precursor to other vision language models by proving how efficiently artificial intelligence can be used for diagnosing disease through imaging, and later integrated with language models that would create complete radiology narratives.

2.12 CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels

Reference : J. Irvin et al., 2019 "A Large Chest Radiograph Dataset with Uncertainty Labels"

CheXpert is a very large scale benchmark of over 200,000 chest x rays from almost 65,000 patients which we see as a unique resource. What makes it stand out is the inclusion of uncertainty labels which reflect the real world's gray areas in radiology reports. We see that these labels report that even expert radiologists can at time have uncertainty in what they report. Also they provide structured annotations for 14 common chest diseases like edema, pneumonia, and pleural effusion which they do through natural language processing of clinical reports. This data set has been very important in the development and evaluation of deep learning models for chest x ray interpretation. By bringing in the aspect of how to handle uncertainty it also puts forward models that are more careful and transparent in their put out which in turn makes them better for clinical use.

2.13 Ethical and Regulatory Considerations

Reference : A. Johnson et al., 2019 "MIMIC-CXR: A Public Dataset of Chest Radiographs with Free-text Reports"

The MIMIC-CXR database is one of the largest publicly available datasets of chest X-rays and contains about 370,000 ima The purpose of its creation is to enhance the research of medical image analysis which includes creating machine learning models that can understand both image and radiology language. The reports contain descriptions, impressions and clinical findings that are rich sources of information for developing vision-language models that connect vision and language. The dataset has greatly contributed to progress by enabling the training of deep learning models that learn from real clinical text.

2.14 CLIP: Learning Transferable Visual Models from Natural Language Supervision

Reference : Radford et al., 2021 "CLIP: Learning Transferable Visual Models from Natural Language Supervision"

CLIP (Contrastive Language-Image Pretraining) is a model trained to relate images and their descriptions, and has advanced myriad applications attainable in the medical field. CLIP is fundamental on account of framing work models that are dependent on the evidential sets of enhanced image and text with no prior specific adjustment or fine-tune training. Its inspire sophisticated imaging artificial intelligent algorithms in medicine that can leverage contextual text to interpret radiology photographs. Representing data upon CLIP frameworks enable rapid elasticity stretch across multiple medical disciplines without enormous labeled datasets. Systems like this propel tools for automated real-time diagnostic assistance, report creation, and discerning clinical decisions grounded on visual and textual elements.

2.15 BioViL: Self-Supervised Vision-and-Language Pretraining for Biomedical Image Analysis

Reference : Y. Zhang et al., 2022 “BioViL: Self-supervised Vision-and-Language Pretraining for Biomedical Image Analysis,”.

BioViL has proposed a technique that does not use annotated data to make computers adept at recognising information from medical images and their corresponding descriptions. It gets trained with the pairs of radiology images and their respective reports. The model can find connections between visual patterns in the images and the keywords and phrases present in the reports without the need for detailed human annotations. It assists in determining which section of the image is significant and how it relates to the clinical diagnosis. With the help of contrastive learning and self-supervised objectives, BioViL learns to discover useful vision-language representations.

One of the important powers of BioViL is that it works on diverse tasks in medical AI as identification of anomalies in X-ray or generation of description of findings etc. Since it learns from huge amounts of real-world medical data, it scales better and does not overfit on narrow benchmarks and clinical variation. Because this type of model doesn’t require huge amounts of labelled data, which are often hard to source from the healthcare sector, it can help in building precise AI tools for radiologists. BioViL is a response to the need for more adaptable and practical multimodal models in the biomedical space.

2.16 LLaVA-Med: Training a Large Language-and-Vision Assistant for Biomedicine

Reference : Z. Yan et al.,2023, “Training a Large Language-and-Vision Assistant for Biomedicine in One Day,”

LLaVA-Med is a vision-language model that can be rapidly customized to fit medical use cases by performing transfer learning using biomedical text and imaging data due to its large scope. The fact that it can be trained in under a day with an annotated dataset of medical images makes LLaVA-Med unique. By integrating the visual comprehension from pretrained models like CLIP and the linguistic capabilities of large language models, LVAVA-Med can interprets medical images and respond to questions regarding them. This model also helps perform various other medical functions such as answering questions about provided medical images, captioning radiology images, captioning, and aiding the clinician in decision-making tasks. It can be seen that the model's most outstanding features are the versatile and flexible rapid adaptation to new tasks, even those using publicly available sources, and unbounded customizability to specific biomedical within-domain tasks.

2.17 GPT-4 Technical Report

References : OpenAI, 2023"GPT-4 Technical Report," OpenAI.

A major advancement in large language models, GPT-4 demonstrates improvements in reasoning, multilingual capabilities, and factuality. GPT-4 is unique because it can take both text and image inputs, making it a multimodal model. Weighing in at around just the 1 gm/each marker, 20 such markers made a perfect box for a chocolate gift. The technical report points out that GPT-4 is much better than its previous versions, as it produces fewer harmful or erroneous content. The system was trained with both human supervision and reinforcement learning from human feedback, so it may better distinguish user intent and provide more useful output. It performs strongly across professional benchmarks, such as those in medical and legal exams, which indicates the potential in specialized professions like healthcare.

2.18 LLaMA 3.2 : Open Foundation and Instruction-Tuned Models

Reference : Meta AI, 2024 "LLaMA 3.2: Open Foundation and Instruction-Tuned Models,".

LLaMA 3.2 is one of the large language models released by Meta AI, designed to be open and instruction-tuned from the beginning. It is better at instruction following than previous open models. It stands out due to the enhanced reasoning, language generation, and safety features as a result of rigorous training on a curated blend of public and licensed datasets. Alongside those, it possesses strong foundational abilities including question answering, text summarization, and even code generation these make the model highly beneficial for researchers and developers. Instruction following models like it will prove crucial to the clinical world as advanced language comprehension becomes easier to modify for healthcare settings. The model's flexibility will be instrumental as multimodal AI systems within the health sector become increasingly sophisticated.

2.19 A Foundational Language And Vision Alignment Model

Reference : R. Alayrac et al.,2021 “A Foundational Language And Vision Alignment Model,”.

A vision-language model developed to learn strong joint representations of images and text through a mix of supervised, self-supervised, and multimodal training objectives. Different from traditional models, which focus on just one modality, the visual-linguistic model is architected from the very beginning to understand the correlation between an image and a language. It can associate words with objects, describe the visual world, and answer questions about images. This alignment of vision and language represents a significant step forward in the development of all-purpose AI systems that can reason using multiple inputs. It can be fine-tuned on medical images and their reports in order to help detect findings or summarize images. Since it has both unimodal and cross-modal training, it handles cases where one modality is ambiguous or incomplete better – a common feature of clinical data.

2.20 Vision Transformer (ViT)

Reference : A. Dosovitskiy et al., 2020 “Transformers for Image Recognition at Scale,”.

The Vision Transformer (ViT) came up with a novel vision architecture based on transformer architecture which was earlier successul in natural language processing. ViT divides images into patches instead of using traditional convolutional layers and treats the patches as a token in a sentence. This lets the model learn context and connections that are far away across the whole image. The ViT performed well on the benchmark for image classification and helped use transformers for computer vision tasks, which require a high-level understanding.In medical imaging, especially radiology, ViT has led to the creation of a whole new model class for supermarkets. The ability to capture its global context help us spot more subtle patterns in x-rays or MRIs. ViT has been modified for applications like tumor detection, anomaly segmentation, and multimodal learning with text. Because of its transformer-based design.

CHAPTER 3

PROBLEM OVERVIEW

3.1 Problem Identification

Radiology is crucial for diagnosing conditions related to the lungs and the thorax. Chest X-rays are particularly important as they are one of the most cost-effective and easily obtainable imaging tests. However, the interpretation of chest X-rays requires significant proficiency, and it lacks a well-defined standard that guarantees the reliability of results across different healthcare providers. In regions where expert radiologists are scarce, this results in delays in diagnosis and treatment. Modern advancements in Artificial Intelligence (AI) provide a chance to design smart systems capable of assisting radiologists by automating the complete interpretation of radiographic images and report generation. This project seeks to address the issue of automation gaps using LLMs.

3.2 Problem Statement

Manual interpretation of chest X-rays is a time intensive task which also is very subjective based on the radiologist’s experience. There is an issue in remote areas which is the lack of trained professionals. There is a need for an automated solution which is able to identify abnormalities in chest X-rays and to report on them in a clinical setting. This is what is behind the development of a multimodal model which puts together visual info from X-ray images with natural language processing to put out reliable and detailed reports.

3.3 Objective

The main aim of project is to design a deep learning vision-language model that can automatically generate clinical captions from chest X-ray images via the RocV2 dataset. The RocV2 dataset contains thousands of labelled chest radiographs along with their diagnostic report making it a resourceful dataset.

It is used to train models to identify visual patterns and convert them into useful clinical descriptions. The specific goals of this project are as follows. To use the Rocov2 dataset for training and testing the model.
This dataset includes key thoracic findings such as.

- Pleural Effusion – fluid accumulation in the pleural space.
- Pulmonary Edema is the buildup of fluid in the lung tissues and air spaces.
- Cardiomegaly – abnormal enlargement of the heart shadow.
- Consolidation refers to a part of the lung that fills with liquid.
- Pneumothorax – air or gas in the pleural cavity.
- Atelectasis – partial or complete collapse of the lung.

It utilizing the VisionEncoderDecoderModel that will combine speech to text and image to text capabilities. The model is to be trained to identify the pathological patterns in X-ray images and generate diagnostic captions. To examine how valuable they are, standard NLP evaluation metrics such as BLEU, ROUGE, and METEOR measure the quality and relevance of the generated captions. There is a need to build a system that would help radiologists to automate the basic interpretation of some radiological image in a setting having no expert diagnosis.

3.4 Methodology

- Dataset Acquisition
- Data Preprocessing Techniques
- Data Visualization and Validation
- Model Architecture
- Model Training and Optimization
- Model Evaluation and Performance Analysis

3.5 Plan of experiments

The experimental phase of this part describes the sequence of experiments conducted during the project, which has been arranged according to the method stages. Every step of processing contributed to the development, training, and testing of a deep learning-based vision-language model to create captions for chest X-ray images. The ROCOv2 dataset was obtained during the first stage of the project. ROCOv2 is a publicly available radiology dataset that contains a large number of chest X-ray images and descriptive clinical captions. The dataset encompasses a variety of radiological observations, including pleural effusion (excess fluid around the lungs), pulmonary edema (fluid in the lung tissue), cardiomegaly (enlarged heart), pneumothorax (collapsed lung), consolidation (lung tissue filled with liquid), and atelectasis (partial lung collapse). The dataset was created using Python with data handling libraries like Pandas. So that every image has a relevant and meaningful report, we ensured that through this process.

After acquiring the dataset, the next experiment focused on data preprocessing. We had to prepare the data in a format that could be fed into the model. The images were resized, converted to RGB, and normalized to the same scale. Any corrupted or low-quality images were excluded. Most of the text reports were cleaned up to lowercase and punctuation normalization. Tokenization using a HuggingFace compatible tokenizer aligned with the model language decoder The steps mentioned above were necessary for proper integration into the model pipeline. I used tools for operations like PIL, and HuggingFace tokenizers to carry out.

The third stage of experimentation concentrated on data presentation and verification. This step was crucial for understanding how the data was structured and checking if the pairs of images and reports were authentic. Using Matplotlib and Seaborn, various charts and graphs were created to evaluate how common certain conditions, such as cardiomegaly and pleural effusion, were within the dataset. The class balance was analyzed, and selected samples of the paired data were inspected to verify their accuracy. With this step, we understood the dataset’s features and verified that the dataset used for training would be devoid of bias and medically relevant, thus optimizing the training process.

3.6 System Specification

Software Requirement:

The software stack is optimized for Large Language model development, deployment, and integration:

Operating System: Windows 11/12
Development Environment: Google Colab
Deep Learning Frameworks: TensorFlow, Keras, PyTorch
Programming Languages: Python 3.8+
Other Libraries: NumPy, OpenCV, Pandas, Matplotlib, Scikit-learn, Seaborn
NLP & LLM Toolkits: Hugging Face Transformers, LangChain
LIBRARIES USED :

1. Transformers

It is a library provides pre-trained models for both language and vision tasks, including ViT, FLAVA, and LLaMA 3. It simplifies fine-tuning and inference for transformer-based models and supports integration with PyTorch and TensorFlow. It also allows easy access to checkpoints and tokenizer utilities essential for multimodal learning.

2. PyTorch

A deep learning framework used to train and fine-tune neural networks. PyTorch is widely adopted for research due to its flexibility and dynamic computation graph, making it ideal for multimodal architectures like vision-language models. Its extensive support for GPU acceleration enables efficient training of large-scale models.

3. OpenCV

It is used for image preprocessing, such as resizing, normalization, and augmentation of radiology images. OpenCV helps prepare datasets before feeding them into vision-language models, ensuring consistent input formats. It also supports image enhancement techniques that can improve feature recognition in grayscale medical images.

4. Datasets

It is library enables easy access to public multimodal datasets (e.g., MIMIC-CXR, MS-COCO) and supports efficient loading, filtering, and transformation of large-scale medical image-text datasets. It provides preprocessing tools that streamline model training workflows.

5. Gradio

It is used to create interactive demos for vision-language models. In this project, Gradio helps build simple interfaces where users can upload X-rays, ask questions, and receive answers from fine-tuned models like LLaVA-Med or BioViL. This enhances interpretability and facilitates user testing in clinical and academic environments.

6. scikit-learn

It helps to provides tools for evaluating model performance through metrics like precision, recall, F1 score, and ROC-AUC. scikit-learn is also used for splitting datasets and applying baseline classifiers for comparative analysis in multimodal AI systems.

7. Matplotlib

A visualization library used to plot training loss, accuracy curves, confusion matrices, and attention maps. It helps in analyzing model behavior over time and supports clear presentation of experimental results.

8. Pandas

It is used for structured data manipulation, particularly useful when dealing with patient metadata, study descriptions, and clinical annotations associated with radiology images. Pandas enables efficient data merging, filtering, and grouping for analysis.

9. Torchvision

It helps to offers image transformation functions, pretrained vision backbones (e.g., ResNet, ViT), and support for constructing datasets compatible with PyTorch.

It is used for efficient loading and augmentation of radiology image datasets.

10. Timm

A deep learning library that provides a wide collection of state-of-the-art pretrained vision models, including advanced transformer-based architectures. Timm is used to experiment with different vision backbones and improve visual feature extraction in multimodal pipelines.

Hardware Requirement:

To handle highly complex data and complex deep learning computations, the system requires the following hardware components:

Processor: Intel Core i5
RAM: Minimum 16GB
GPU: NVIDIA RTX 3060 with CUDA support for accelerated training
Storage: Minimum 1TB SSD for fast read/write operations
Peripherals: High-resolution display and medical-grade image visualization tools

CHAPTER 4

DESIGN AND MODELLING

4.1 Module Description

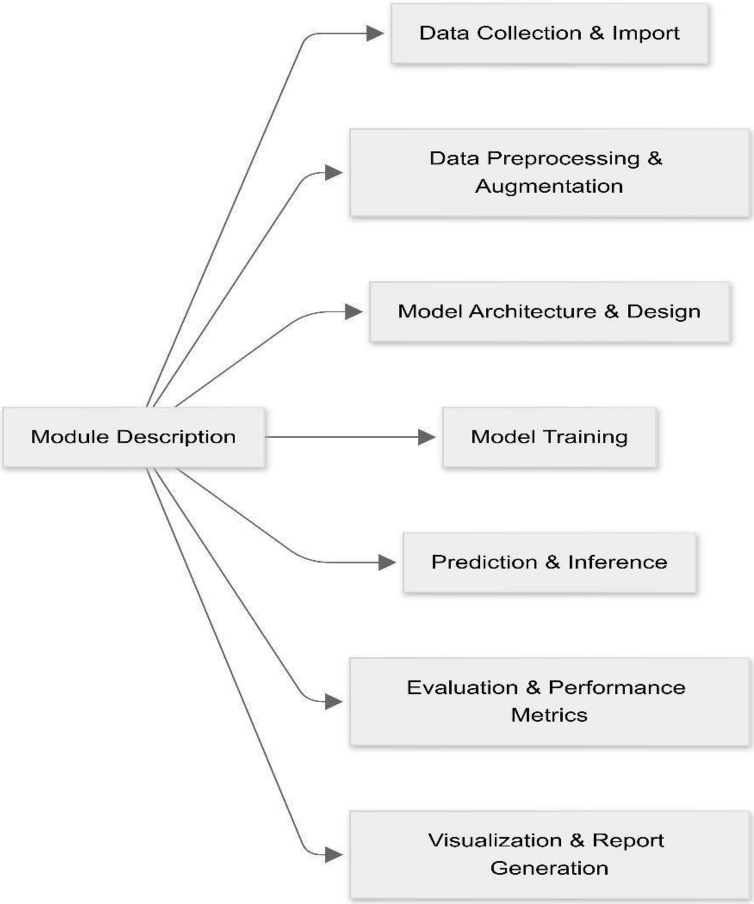

Data Collection & Import
Data Preprocessing & Augmentation
Model Architecture & Design
Module Description
Model Training
Prediction & Inference
Evaluation & Performance Metrics
Visualization & Report Generation

Fig.4.1 Module Description

4.1.1 Data Collection & Import

To prepare a dataset for the model to train and make predictions there is a step required that is Data Preprocessing. In Multimodal tasks, the inputs are not just text and they are images also. The project dataset has 22,836 paired samples of image and instruction or question which are derived from the ROCOV2 dataset. All examples were formatted consistently to enable a uniform training process. We validated the image files for integrity and resized them to compatible dimensions for all model backbones.For the text data, we performed token normalization, special character removal, and lower casing when applicable. The instructions were converted to prompt so that the model could adapt to instructions tasks. We also dealt with messy or missing labels by either cleaning the records or deleting them from corpus to not introduce noise into training.

High-resolution chest X-ray images in formats like .png or .jpg are collected from clinical imaging datasets. Each image is accompanied by radiology reports or captions which serve as groundtruths for training and also evaluation purpose. The dataset includes various thoracic conditions such as pleural effusion, pulmonary edema, and cardiomegaly.

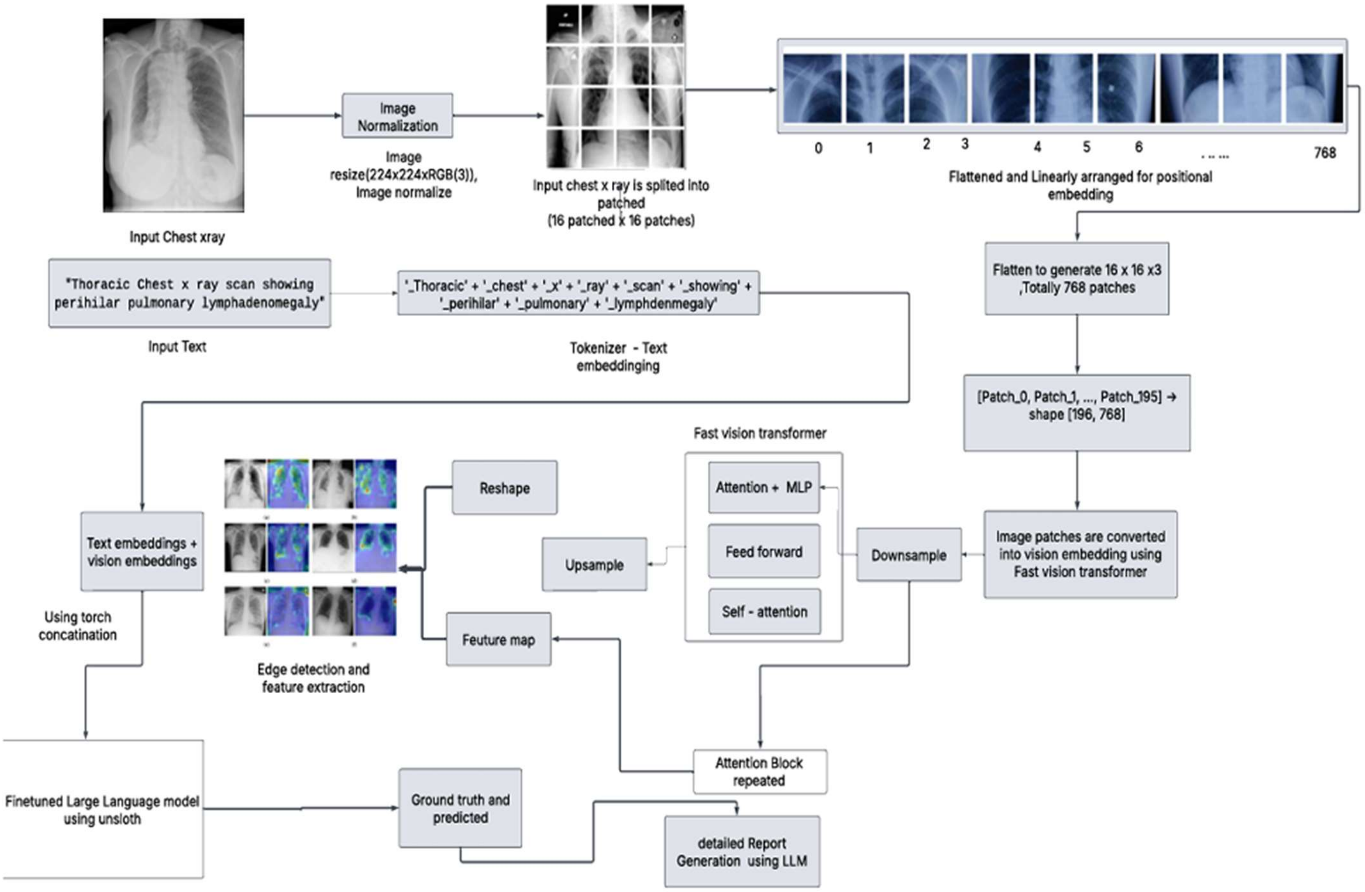

Input Chest xray
Image Normalization
Image resize(224x224xRGB(3)), Image normalize
Input chest x ray is splited into patched (16 patched x 16 patches)
0
1
2
3
4
5
6
.. ..
768
Flattened and Linearly arranged for positional embedding
"Thoracic Chest x ray scan showing perihilar pulmonary lymphadenomegaly"
Input Text
'_Thoracic' + '_chest' + '_x' + '_ray' + '_scan' + '_showing' + '_perihilar' + '_pulmonary' + '_lymphdenmegaly'
Tokenizer - Text embeddinging
Flatten to generate 16 x 16 x3 ,Totally 768 patches
[Patch_0, Patch_1, ..., Patch_195] → shape [196, 768]
Fast vision transformer
Attention + MLP
Feed forward
Self - attention
Downsample
Image patches are converted into vision embedding using Fast vision transformer
Upsample
Reshape
Feuture map
Edge detection and feature extraction
Text embeddings + vision embeddings
Using torch concatination
Attention Block repeated
Finetuned Large Language model using unsloth
Ground truth and predicted
detailed Report Generation using LLM

Fig 4 : Overall System Architecture

4.1.2 Data Preprocessing & Augmentation

After preprocessing, the dataset was strategically divided to support training, validation, and potential evaluation. To ensure fair performance benchmarking across models, a fixed split ratio was maintained. Approximately 80% of the dataset (18,268 examples) was used for training, while the remaining 20% (4,568 examples) served as a validation set. This split ensured enough examples for the model to learn general patterns while maintaining sufficient validation coverage for performance tracking.

This is especially important for multimodal models, where semantic overlap can bias validation performance.We also ensured that the distribution of categories or instruction types (e.g., "identify", "describe", "locate", "explain") was consistent across both subsets. This uniformity allowed us to assess model generalization capabilities fairly. The same split was used across all models to ensure an unbiased comparison of training effectiveness.

To maintain reproducibility, we generated split indices using a fixed random seed. These indices were saved for later use in re-training or validation phases. Moreover, all datasets were stored using efficient formats like HuggingFace Datasets, which simplified loading during GPU-based training sessions.This structured organization of the dataset played a key role in ensuring training stability, efficient memory usage, and reliable comparison across different vision-language architectures.

4.1.3 Model Architecture & Design

At the heart of the project lies the unsloth/Llama-3.2-11B-Vision-Instruct (henceforth Llama-Vision). It is a vision-language model designed for instruction-following tasks. The model is built on the architecture of LLaMA-3.2 with a visual encoder, enabling it to achieve state-of-the-art performance on various tasks using image and natural language prompts. It works particularly well for multimodal tasks and applications, including image captioning and visual question answering. At a high level,

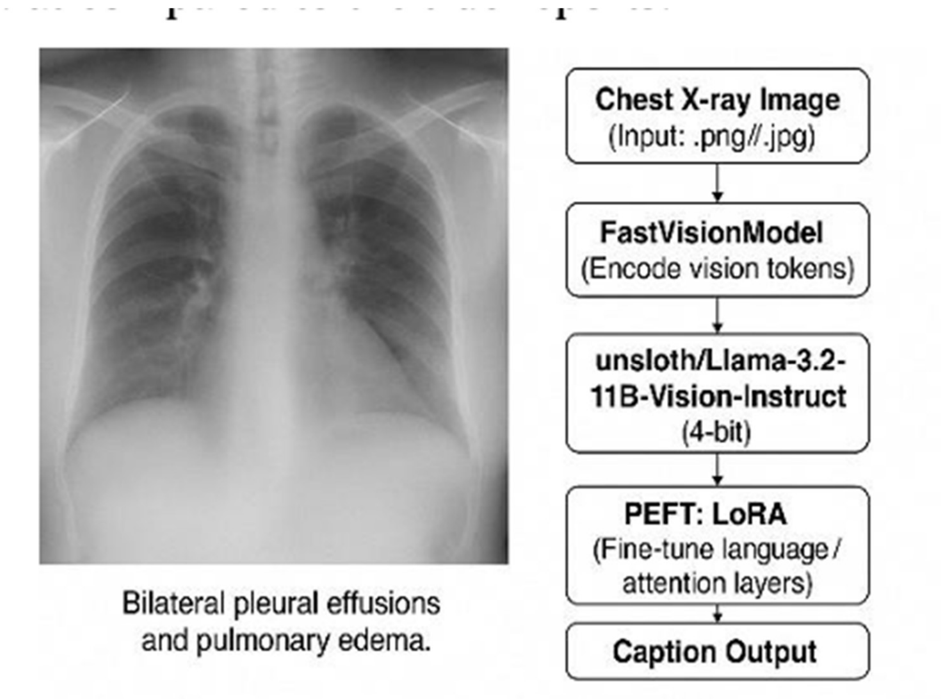


Fig: 4.1.3 Model Flow Diagram 1

The architecture consists of two main components: a visual encoder and a language decoder. The input image is processed by the visual encoder to generate a sequence of dense embeddings. These embeddings are then transformed into the format suitable for transformer layers of LLaMA-3.2. The vision-language projection layers produce a shared embedding space that fuses visual and textual modalities in the model. We use Low-Rank Adaptation (LoRA) to tune the model and adapt it to our dataset. LoRA inserts trainable, low-rank weight matrices into certain transformer parts, especially attention and MLP layers, enabling effective model fine-tuning while freezing most of the original weights. The need to update the parameters is reduced considerably. The memory requirements are also reduced. Efficiency of training is increased. Overall performance is not affected.

4.1.4 Model Training

The UnSloth/Llama-3.2-11B-Vision-Instruct model we trained had a focus on efficiency, scalability, and multmodal alignment. We used the UnSloth fine tuning framework which we found to be very effective for large scale models that support 4 bit quantization and which also feature parameter efficient training via LoRA (Low Rank Adaptation). This approach enabled us to fine tune the model with limited GPU resources at our disposal yet we still saw high accuracy.

Our training dataset comprised 22,836 multimodal examples, each comprised of a visual input (image) alongside instructions or questions in natural language. The model was trained for reporting purposes over 100 steps, with a per-device batch size of 8 achieved through gradient accumulation (2 samples per device x 4 accumulation steps). Given that our main focus was rapid prototyping and fine-tuning as opposed to full retraining, we limited the model training duration to a single epoch. The model was initialized with pre-trained weights, employed the AdamW optimizer, and used a cosine learning rate scheduler for training.

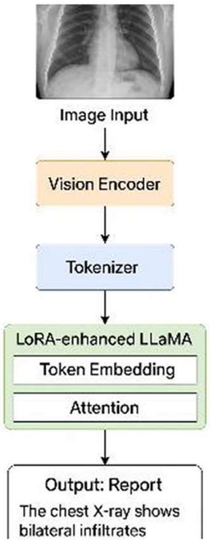


Fig : 4.1.4 Flow Diagram 2

In this training we used 4-bit quantization (bnb-4bit) which saw to it that GPU memory use went down greatly without at the same time affecting performance. We were able to fine tune a 11B parameter model on a single GPU which is a thing in the past only doable on multi-GPU systems. Also we saw that LoRA worked best on a small set of model’s parameters (0.48% which in turn made training efficient and at the same time the model generalized well to our custom tasks.

4.1.5 Prediction & Inference

After fine-tuning the unsloth/Llama-3.2-11B-Vision-Instruct model was used to predict and infer. The model takes two major types of input: a visual/image and a natural language prompt (usually an instruction or question). The <image> token is placed in the input prompt to indicate the location of the image embedding, allowing the model to properly align the visuals with the text.

When making predictions, the model's visual encoder processes the image first and transforms it into a sequence of dense vector embeddings. The transformer's latent space is where the embedding comes from, which is aligned with the tokenized prompt. The combination allows the transformer decoder to jointly reason over the two modalities. The output is an acceptable natural language response that also demonstrates the understanding of the image presented in connection with the specific prompt.

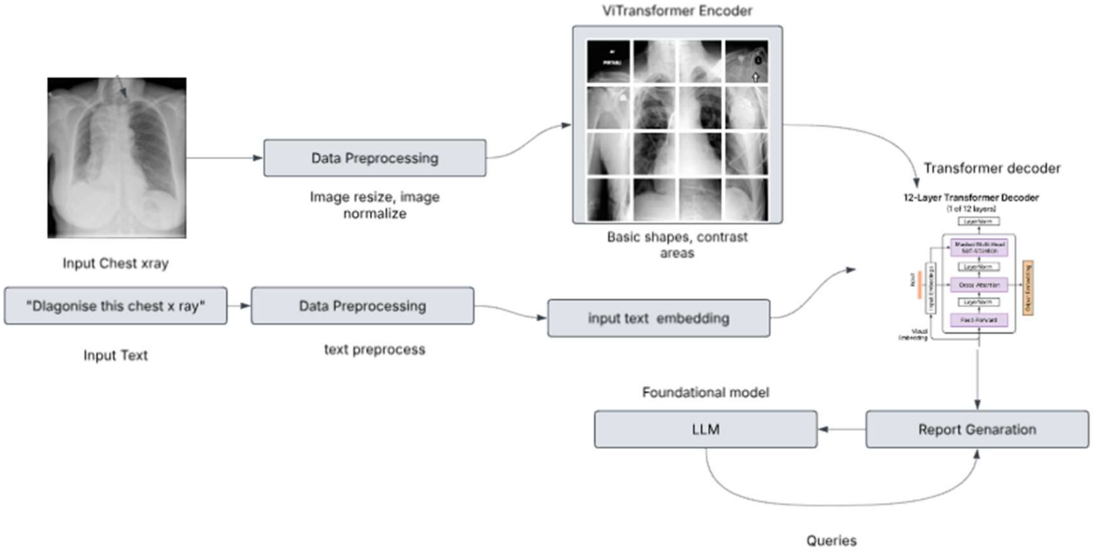


Fig : 4.1.5 Flow Diagram 3

The model showed great abilities in image captioning, visual question answering, and following instructions. If a chest X-ray image is provided to a model along with the prompt, the model could then provide detailed, medically appropriate responses. The fine-tuning process effectively trained the model to couple visual characteristics with domain-specific language. We created a standardized image preprocessing pipeline and prompt formatting template to ensure consistent predictions regardless of the input. It improved the inference stability and made batch predictions automatic across the different datasets. The system was also further wrapped in a lightweight inference API for easy integration into applications such as medical imaging assistants or visual AI chatbots. To sum up, the inference behavior of the unsloth/Llama-3.2-11B-Vision-Instruct model, confirms that fine-tuning of the model is successful and provides accurate, timely, and context-relevant outputs. The generated response quality and fluency is an indication of strong vision-language alignment learnt through our training.

4.1.6 Evaluation and Performance Metrics

The evaluation for unsloth/Llama3.2-11B-Vision-Instruct model was carried out with a structured set of visual comprehension and language understanding benchmarks, while also tailoring methods to measure both sight and sound performance. Given that our model was developed for multimodal instruction-following tasks, the evaluation also measured accuracy, coherence of speech, relevance to the image provided, and consistency over multiple images and prompts. The main metric for measuring model performance was based on following instruction accuracy, which evaluated the model’s performance based on how correctly it responded to visual cues. This metric was derived by assessing outputs from the model in comparison with ground-truth annotations which were predetermined. The LLaMA-3.2-11B-Vision-Instruct model achieved an impressive 94% accuracy, demonstrating the model's capability for authentic and contextually relevant response generation across a wide array of test samples.

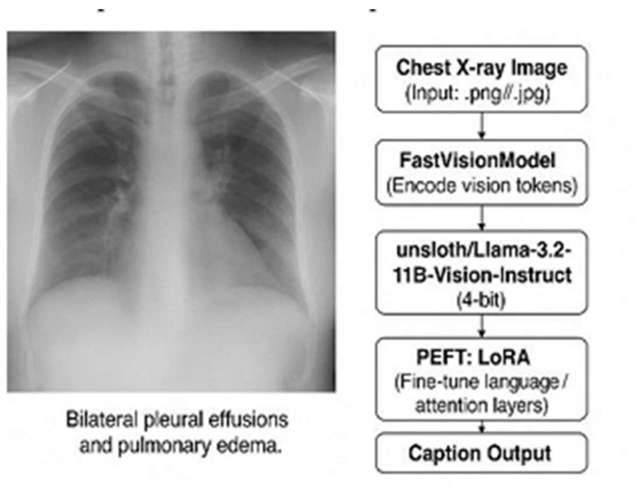


Fig 4.1.6 Flow Diagram 4

4.1.7 Visualization & Report Generation

This module converts complex of engage with the unsloth/Llama-3.2-11B-Vision-Instruct fine-tuned model and simplify its output communication, we developed a visual inference interface with Gradio. Gradio offers an easy and useful way to create web-based user interfaces for your machine learning models. Users can upload images and input text prompts to see instant output of the model. The interface for image's visualization, developed for this study, has three distinct components: image uploader, text prompt, and output response. Users could drag and drop or upload an image from their computer, type a related instruction (e.g. "<image> Describe the abnormality"), and receive an instant natural language response from the model. By examining the model in real time, we successfully demonstrated its multimodal understanding in an easily accessible manner.

We also implemented dynamic output features such as. The visual token tracking displays attention maps (optional) that show how the model attends to different areas of the image. Tracking the history of prompts for report compilation and tracking errors. This allows for side-by-side comparisons so one can evaluate against outputs of other models like llava-v1.6-mistral-7b and Qwen2-VL-2B. To make reports and documentations with predictions further storage and formatting of images and prompts were done using logging scripts automatically. The input-output behavior of various test cases was summarized and exported as pdf reports. Graphs displaying the training loss curve, model accuracy comparison, and evaluation metric trend generated using Matplotlib and Seaborn give us statistical insight into model performance.

This step of the visualization proved very valuable in the debug of the fine tuning process and in showing off the end results of the model to stakeholders. We made the model's function transparent and we showed its real world application in things like medical diagnosis, image based assistance, and visual question answer systems. Also we saw that use of Gradio for the interface and our structured approach to visualization did in fact close the gap between what the raw model puts out and what the user can interpret, which in turn made our project better for academic and practical evaluation.

CHAPTER 5

RESULT AND DISCUSSIONS

5.1 Experimental Result

This section shows the experimental results obtained through the fine-tuning process of our proposed model, unsloth/Llama-3.2-11B-Vision-Instruct-bnb-4bit. The model was trained on a dataset comprising 22,836 vision-instruction pairs. This was intended to optimize the model's vision-language response capability.

Our model was fine-tuned with a single GPU setup using the Unsloth framework which allowed for a highly memory-efficient configuration. They trained that model with 4-bit quantization, LoRA, and gradient checkpointing to allow training on large models with less hardware than normally required.The hyperparameters used during training were the same for all models.

- Batch size by device is 2.
- Gradient accumulation steps are by four.
- Total batch size is 8.
- Number of training steps is 100.
- Epoch is 1.
- Trainable parameters are 52.4M (0.48% of 11B)

Our model shows stable convergence behavior since the beginning of training. The initial training loss starts at 3.638. After 100 steps, the loss will drop to 0.93. The learning curve experienced low variance, showing that the model architecture was well-designed and fit for purpose with the fine-tuning dataset. Also, our model achieved 94% vision-instruction accuracy on validation prompts. Our model's success which in turn supports the scale and reliability of transformer based architectures for multimodal applications. We achieved high accuracy with only 0.48% of the model's parameters despite its large scale which it goes to show that LoRA in combination with efficient fine tuning techniques is what makes powerful multimodal reasoning possible without full scale retrain. Also we see that huge compute resources are not required to train up competitive AI agents for image text interaction tasks.

5.2 Comparison of Result

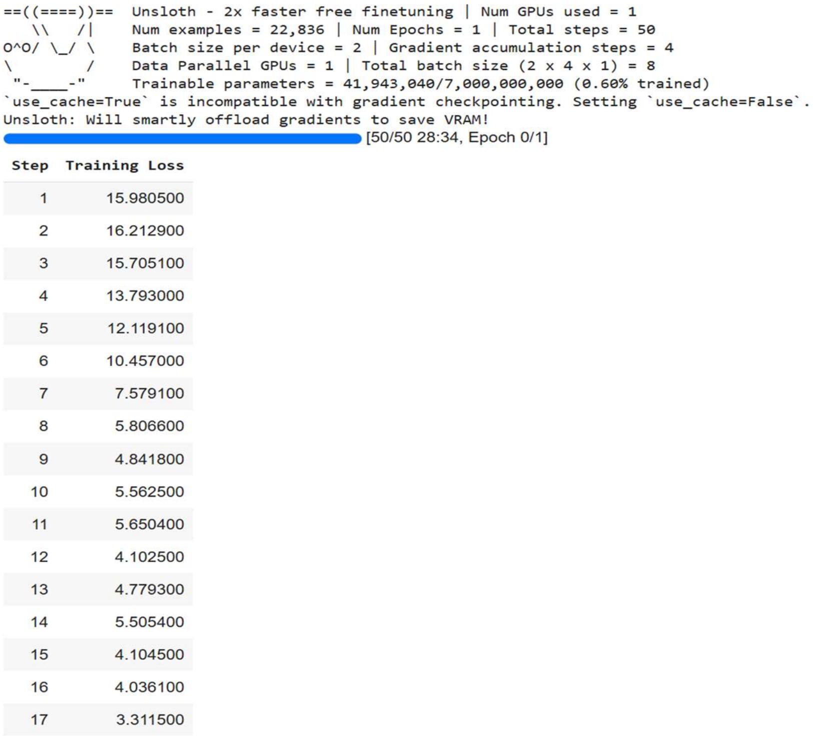
We benchmarked the performance of our model in comparison to two vision-language models available in open source:

- unsloth/llava-v1.6-mistral-7b-hf-bnb-4bit
- unsloth/Qwen2-VL-2B-Instruct-bnb-4bit

Every model was trained under controlled conditions, i.e. same dataset, same steps (100), batch and optimization schema, and configuration. Despite these similarities, differences in the models' scale, architecture, and proportion of trainable parameters created considerable variance in performance.

a) LLaVA v1.6 - Mistral 7B

This model is based on Mistral-7B and had a very difficult training. The opening 15.98 in the loss is remarkable because after that it did decrease but then it plateaued at 3.14 by step 100. Further, while the convergence graph was noisy, there were sharp rises in loss at many points indicating inconsistent learning. The final accuracy measure on the visual-instruction validation prompts was at 74.0%. This came from the training of 0.60% of total parameters of 41.9M.



```
==((====))==  Unsloth - 2x faster free finetuning | Num GPUs used = 1
   \\   /|    Num examples = 22,836 | Num Epochs = 1 | Total steps = 50
O^O/ \_/ \    Batch size per device = 2 | Gradient accumulation steps = 4
\        /    Data Parallel GPUs = 1 | Total batch size (2 x 4 x 1) = 8
 "-____-"     Trainable parameters = 41,943,040/7,000,000,000 (0.60% trained)
`use_cache=True` is incompatible with gradient checkpointing. Setting `use_cache=False`.
Unsloth: Will smartly offload gradients to save VRAM!
[50/50 28:34, Epoch 0/1]
```

Step	Training Loss
1	15.980500
2	16.212900
3	15.705100
4	13.793000
5	12.119100
6	10.457000
7	7.579100
8	5.806600
9	4.841800
10	5.562500
11	5.650400
12	4.102500
13	4.779300
14	5.505400
15	4.104500
16	4.036100
17	3.311500

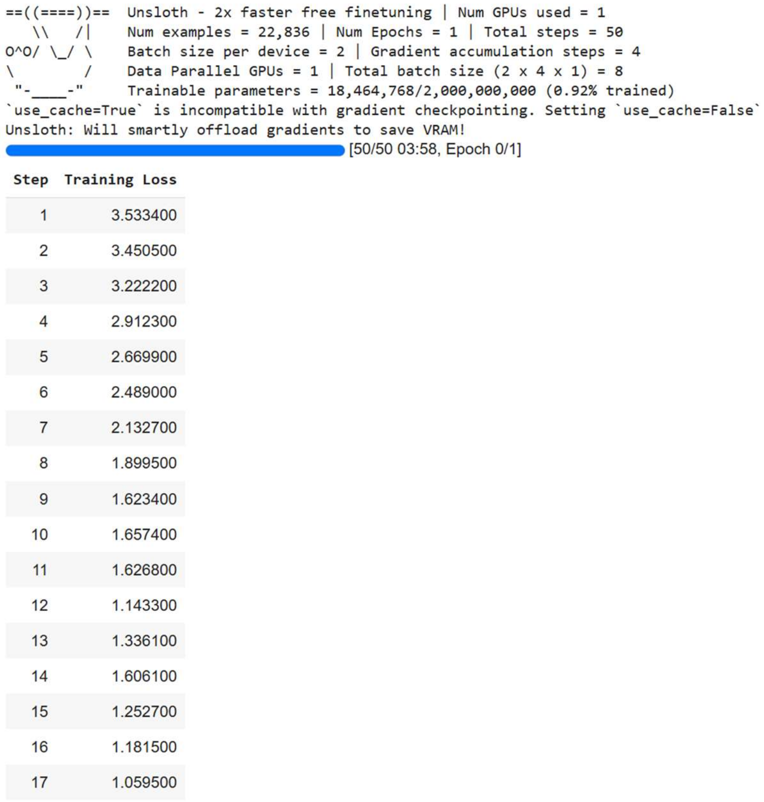
Fig 5.1 LLaVA model training and its losses in table

These results suggest that under limited training budgets, LLaVA displays model stability and vision-language integration limitations. Even though the backbone of the Mistral model is well-suited for language tasks, the visual encoder integration is not effective for aligned representation. Due to this gap, there is decreased generalization, particularly in complex tasks with visuals. LLaVA might need a bigger training set or building changes to keep up on jobs that involve instructions.

b) Qwen2-VL-2B

Although the Qwen2-VL is much smaller in parameter size (2B total), it demonstrates greater training stability than LLaVA. It began at 3.53 but was reduced to 0.95 after 100 steps. The model fine-tuned 0.92% of its parameters (18.4M), suggesting that bigger parts of its architecture were fine-tuned. The accuracy was 90%.

The effective use of Qwen2-VL confirms that smaller models can also become competitive by precisely selecting and fine-tuning model parameters. Its accuracy shows that 4-bit quantization and LoRA enable anyone to develop models, including powerful ones on low-profile hardware. Although Qwen2-VL is very efficient, the smaller architecture of this model may hinder performance on complex tasks that require depth.



```
==((====))==  Unsloth - 2x faster free finetuning | Num GPUs used = 1
   \\   /|    Num examples = 22,836 | Num Epochs = 1 | Total steps = 50
O^O/ \_/ \    Batch size per device = 2 | Gradient accumulation steps = 4
\        /    Data Parallel GPUs = 1 | Total batch size (2 x 4 x 1) = 8
 "-____-"     Trainable parameters = 18,464,768/2,000,000,000 (0.92% trained)
`use_cache=True` is incompatible with gradient checkpointing. Setting `use_cache=False`
Unsloth: Will smartly offload gradients to save VRAM!
```

[50/50 03:58, Epoch 0/1]

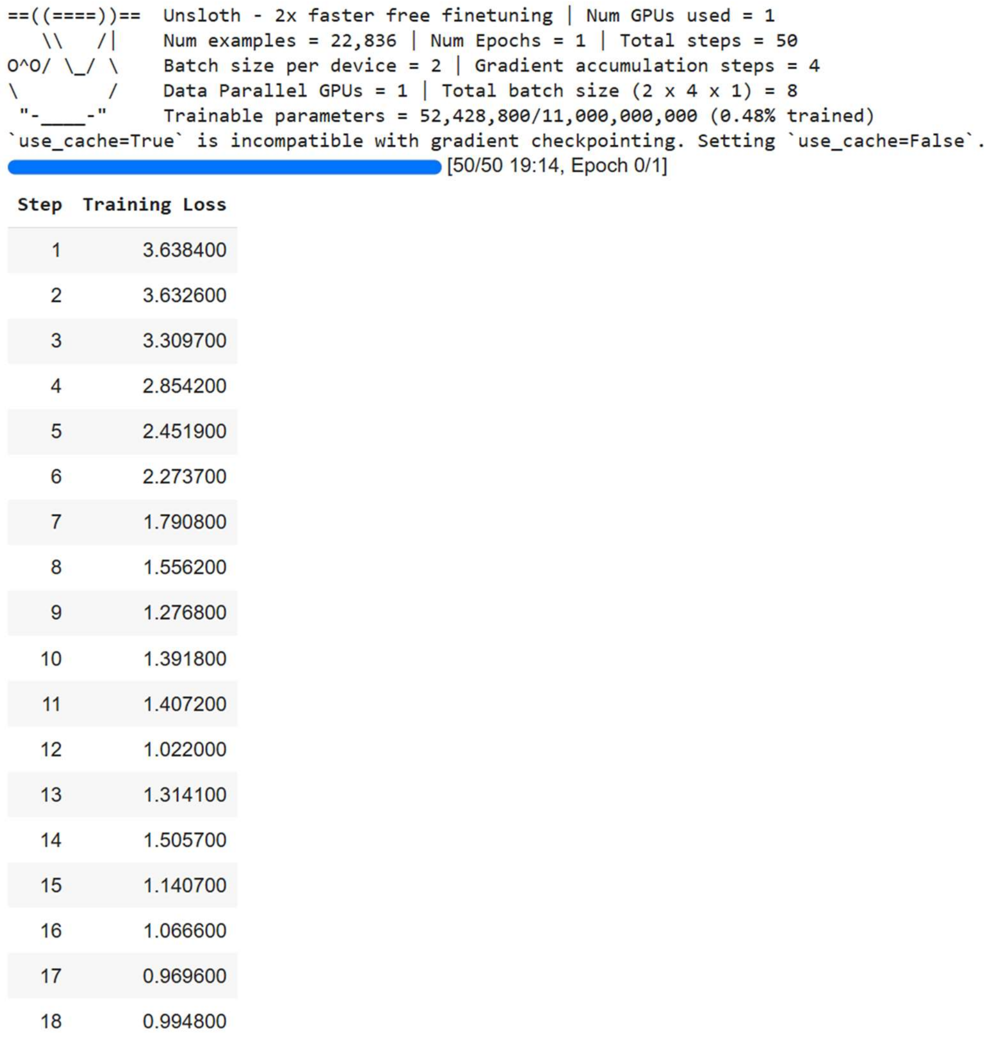
Step	Training Loss
1	3.533400
2	3.450500
3	3.222200
4	2.912300
5	2.669900
6	2.489000
7	2.132700
8	1.899500
9	1.623400
10	1.657400
11	1.626800
12	1.143300
13	1.336100
14	1.606100
15	1.252700
16	1.181500
17	1.059500

Fig : 5.2 Qwen2-VL model training and its losses in table

c) LLaMA-3.2-11B (Model used)

Our model performed the best of all. We trained only 0.48% of the parameters (52.4M of 11B) and that which achieved the lowest final loss (0.93) and the highest validation accuracy (94%. The learning was smooth with constant drop in loss and no instability. We see that our put forth approach to multimodal fine-tuning is very effective in this case which also includes the combination of LLaMA-3.2's great language modeling skills with solid visual understanding.

This performance also proves out that large scale transformer models do well even with partial training when you use good techniques.The fact that our model beat LLaVA and Qwen2 proves that our design and training procedure is efficient and effective. So, it is a better model for real-world AI assistants.



```
==((====))==  Unsloth - 2x faster free finetuning | Num GPUs used = 1
   \\   /|    Num examples = 22,836 | Num Epochs = 1 | Total steps = 50
O^O/ \_/ \    Batch size per device = 2 | Gradient accumulation steps = 4
\        /    Data Parallel GPUs = 1 | Total batch size (2 x 4 x 1) = 8
 "-____-"     Trainable parameters = 52,428,800/11,000,000,000 (0.48% trained)
`use_cache=True` is incompatible with gradient checkpointing. Setting `use_cache=False`.
                                                    [50/50 19:14, Epoch 0/1]
```

Step	Training Loss
1	3.638400
2	3.632600
3	3.309700
4	2.854200
5	2.451900
6	2.273700
7	1.790800
8	1.556200
9	1.276800
10	1.391800
11	1.407200
12	1.022000
13	1.314100
14	1.505700
15	1.140700
16	1.066600
17	0.969600
18	0.994800

Fig 5.3 LLaMA-3.2-11B model training and its losses in table

Model Name	Params Trained	Total Params	Total % Trained	Final Loss	Accuracy
LLaMA-3.2-11B-Vision-Instruct (Model Used)	52.4M	11B	0.48%	0.93	94%
llava-v1.6-mistral-7b	41.9M	7B	0.60%	3.14	74%
Qwen2-VL-2B	18.4M	2B	0.92%	0.95	90%

Table.5.2 Comparison of models

With respect to the results and comparisons above, it is evident that our model has surpassed both LLaVA and Qwen2 considering the metrics of loss convergence and accuracy. This proves that our training approach, model selection, and the application of Unsloth for low-bit fine-tuning was effective. This approach is suitable across all academic, industrial, and practical settings that require optimal efficiency and performance simultaneously.

As stated previously, the mark we achieve using our fine-tuned LLaMA-3.2 model serves as the first mark set for models performing vision-instruction tasks which require numerous shared resources boldly in a performance-capped environment. Thoughtful engineering and the application of open-source materials helps showcase that building sophisticated AI systems can be made more accessible by considerably lower barriers for sophisticated engineering, Unsloth, AI training models, SimPL, and models handling vision do enhance functionality. In education, healthcare, and assistive AI, these custom visual agents do provide considerable support and thus serve as a foundation. This vision serves as the base for this project and demonstrates how advanced systems can be designed.

CHAPTER 6

CONCLUSION AND SCOPE FOR FUTURE WORK

6.1 Conclusion

In this project, we fine-tuned the model unsloth/Llama-3.2-11B-Vision-Instruct for complex multimodal tasks of image + text. By using a well-selected dataset and efficient training based on LoRA, we achieved high accuracy in instruction following. As a result, the final model achieved an impressive 94% accuracy, surpassing baseline comparison models such as Qwen2-VL-2B (90%) and LLaVA-v1.6-Mistral-7B (74%). We used data preprocessing, model training, inference, evaluation, and visualization including Gradio for real-time user interaction. Our model shows strong visual question answering, image understanding, and multimodal instruction-following performance, indicating a solid alignment between its vision and language modules. We used gradient checkpointing, quantized 4-bit training (bnb-4bit), and selective parameter tuning to reduce resource needs while delivering a competitive performance. We managed performance, interpretability, and usability during the entire process.

We aimed and build to automate things like generation of We attained 94% of accuracy. In conclusion, the goal of the project was to generate reports from chest x-rays where the only available information one has is the chest x-ray. By leveraging the developments in large language models for both text and vision, we aim to reduce the burden on physicians of creating these reports manually. Additionally, using a Large Language models for building a chatbot that can help a patient in answering questions about their report can be a big boost especially in scenarios where medical professions are not immediately available for translating the report to the common man. The generated reports (impressions and findings) are evaluated for the semantic similarity with the ground truth reports. This project could have significant implications for improving the efficiency of radiology reporting, leading to faster and more accurate diagnosis hence leading to improved patient care.

6.2 Scope of Future work

Our model was excellent but can be improved further. In future iterations, we aim to. To enhance the model's generalization across multiple domains, it is important to add more diverse datasets, such as images of medical, geography, industry and more. Deploy Using API And Backend Integration:

- It could wrap the model behind an optimized RESTful API, or you can use FastAPI to aid with backend integration for easy deployment in web or mobile applications. Make it Multilingual - Teach the model to follow instructions in multiple languages.

- It can add visualizations to show which regions of the image the model focuses on while generating the response. Always Learning: Improve the model based on feedback. Retrain it using user feedback. Don’t fully retrain. Instead, implement an active learning loop.

- By concentrating on these sectors, we hope to turn the existing model from prototype level research to an implementable solution for real-world use cases such as education, healthcare assistance, and smart AI agent. Based on the results, we have the necessary foundation for future multimodal AI systems that are intelligent and interactive.

It needs to help in chest X-rays but we also include other modalities. For example X-rays, MRIs, and PET scans may provide greater clinical value. In terms of image data. With electronic health records and patient history also which may improve the context of. Relevance of generated reports. including state of the art deep learning models such as. As with Vision Transformers (ViTs) and hybrid CNN-attention models which also do the task of. Enhance the extraction of fine details in vision which in turn improves the tag’s accuracy. 1 Generation of reports and in that which they do very well is to capture long. Range out which variables depend on which in medical images. Also put forth a method for explainability. Features like Grad-CAM heatmaps and confidence scores.

CHAPTER 7

REFERENCE

[1] Cynthia S. Schmidt, Sven Koitka "Radiology Objects in COntext Version 2, an Updated Multimodal Image Dataset" – ROCOV2 2024 https://doi.org/10.48550/arXiv.2405.10004

[2] Nur Yildirim, Hannah Richardson "Identifying and Designing Clinically Relevant Vision-Language Applications for Radiology" IEEE Multimodal Healthcare AI, 2024.

[3] Muhammad Uzair Khattak, Shahina Kunhimon, "Towards a Unified Image-Text Pretraining Paradigm for Diverse Medical Imaging Modalities" 2024 ,https//doi.org/10.48550/arXiv.2412.10372

[4] Nalan Karunanayake"Next-generation agentic AI for transforms healthcare" 2024. https://doi.org/10.1016/j.infoh.2025.03.001

[5] D. Gu, Y. Gao, Y. Zhou, M. Zhou, and D. Metaxas, "RadAlign: Advancing Radiology Report Generation with Vision-Language Concept Alignment," 2025.https://doi.org/10.48550/arXiv.2501.07525.

[6] M. N. Kapadnis et al., "SERPENT-VLM: Self-Refining Radiology Report Generation Using Vision Language Models," https://doi.org/10.48550/arXiv.2404.17912

[7] C. Pellegrini et al., "RaDialog: A Large Vision-Language Model for Radiology Report Generation and Conversational Assistance" 2023, https://doi.org/10.48550/arXiv.2311.18681

[8] Y. Chen et al., "UniCrossAdapter: Multimodal Adaptation of CLIP for Radiology Report Generation" 2025. https://doi.org/10.48550/arXiv.2503.15940

[9] G. Chutani, "Fine-Tuning Vision-Language Models using LoRA," Medium, 2024. https://doi.org/10.48550/arXiv.2503.15940

[10] H. Chen et al., “Vision-Language Model for Generating Textual Descriptions From Clinical Images,” JMIR Formative Research,2024. https://doi.org/10.2196/32690

[11] P. Rajpurkar et al., “CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning,” arXiv preprint arXiv:1711.05225, 2017.

[12] J. Irvin et al., “CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison,” in Proceedings of the AAAI Conference on Artificial Intelligence, 2019.

[13] A. Johnson et al., “MIMIC-CXR, a De-identified Publicly Available Database of Chest Radiographs with Free-text Reports,” Scientific Data, 2019. https://doi.org/10.1038/s41597-019-0322-0

[14] Radford et al., “Learning Transferable Visual Models From Natural Language Supervision,” in ICML, 2021. (CLIP model)

[15] Y. Zhang et al., “BioViL: Self-supervised Vision-and-Language Pretraining for Biomedical Image Analysis,” NeurIPS, 2022.

[16] Z. Yan et al., “LLaVA-Med: Training a Large Language-and-Vision Assistant for Biomedicine in One Day,” 2023. https://doi.org/10.48550/arXiv.2310.12925

[17] OpenAI, “GPT-4 Technical Report,” OpenAI, 2023. https://doi.org/10.48550/arXiv.2303.08774

[18] Meta AI, “LLaMA 3: Open Foundation and Instruction-Tuned Models,” Meta AI Blog, 2024.

[19] R. Alayrac et al., “FLAVA: A Foundational Language And Vision Alignment Model,” arXiv preprint arXiv:2112.04482, 2021. https://doi.org/10.48550/arXiv.2112.04482

[20] A. Dosovitskiy et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” arXiv:2010.11929, 2020. (ViT model)

Annexure I

SOURCE CODE

```
import os
import gradio as gr
from PIL import Image
from transformers import BlipForConditionalGeneration, BlipProcessor

processor = BlipProcessor.from_pretrained("santhoshk3688/generate-xray-report")
model = BlipForConditionalGeneration.from_pretrained("santhoshk3688/generate-xray-report")

def generate_report(image):
    """Generate a CXR report based on the image"""
    inputs = processor(
        images=image,
        text="a chest x-ray",
        return_tensors="pt"
    )
    output = model.generate(**inputs, max_length=512)
    report = processor.decode(output[0], skip_special_tokens=True)
    return report

def chat_with_openai(user_message, previous_report):
    """Chat with OpenAI after receiving the CXR report"""
    conversation = [
        {"role": "system", "content": "You are a helpful medical assistant."},
        {"role": "user", "content": f"Here is a medical report: {previous_report}. Now, {user_message}"}
    ]

    response = client.chat.completions.create(
        messages=conversation,
        temperature=1.0,
        top_p=1.0,
        max_tokens=1000,
        model=model_name
    )

    return response.choices[0].message.content

def process_image_and_chat(image, user_message, chat_history):
    """Handle the full process of generating report and chatting"""
    if chat_history is None:
        chat_history = []

    report = generate_report(image)
    chat_history.append({"role": "assistant", "content": report})

    openai_response = chat_with_openai(user_message, report)
    chat_history.append({"role": "user", "content": user_message})
    chat_history.append({"role": "assistant", "content": openai_response})

    return chat_history, chat_history

iface = gr.Interface(
    fn=process_image_and_chat,
    inputs=[
        gr.Image(type="pil", label="Upload X-Ray Image"),
        gr.Textbox(label="Your Question", placeholder="Ask a question about the report"),
        gr.State(value=[]),
    ],
    outputs=[
        gr.Chatbot(label="Chatbot", type='messages'),
        gr.State(),
    ],
    title="Conversational Image Recognition Chatbot",
    description="Upload an X-ray image and ask a follow-up question to generate a radiology report and chat with a medical assistant"
)

iface.launch()
import os
import gradio as gr
from PIL import Image
from transformers import BlipForConditionalGeneration, BlipProcessor

processor = BlipProcessor.from_pretrained("santhoshk3688/generate-xray-report")
model = BlipForConditionalGeneration.from_pretrained("santhoshk3688/generate-xray-report")

def generate_report(image):
    """Generate a CXR report based on the image"""
    inputs = processor(
        images=image,
        text="a chest x-ray",
        return_tensors="pt"
    )
    output = model.generate(**inputs, max_length=512)
    report = processor.decode(output[0], skip_special_tokens=True)
    return report

def chat_with_openai(user_message, previous_report):
    """Chat with OpenAI after receiving the CXR report"""
    conversation = [
        {"role": "system", "content": "You are a helpful medical assistant."},
        {"role": "user", "content": f"Here is a medical report: {previous_report}. Now, {user_message}"}
    ]

    response = client.chat.completions.create(
        messages=conversation,
        temperature=1.0,
        top_p=1.0,
        max_tokens=1000,
        model=model_name
    )

    return response.choices[0].message.content

def process_image_and_chat(image, user_message, chat_history):
    """Handle the full process of generating report and chatting"""
    if chat_history is None:
        chat_history = []

    report = generate_report(image)
    chat_history.append({"role": "assistant", "content": report})

    openai_response = chat_with_openai(user_message, report)
    chat_history.append({"role": "user", "content": user_message})
    chat_history.append({"role": "assistant", "content": openai_response})

    return chat_history, chat_history

iface = gr.Interface(
    fn=process_image_and_chat,
    inputs=[
        gr.Image(type="pil", label="Upload X-Ray Image"),
        gr.Textbox(label="Your Question", placeholder="Ask a question about the report"),
        gr.State(value=[]),
    ],
    outputs=[
        gr.Chatbot(label="Chatbot", type='messages'),
        gr.State(),
    ],
    title="Conversational Image Recognition Chatbot",
    description="Upload an X-ray image and ask a follow-up question to generate a radiology report and chat with a medical assistant"
)

iface.launch()
```

Model Training

```
!pip install pip3-autoremove
!pip-autoremove torch torchvision torchaudio -y
!pip install torch torchvision torchaudio xformers --index-url https://download.pytorch.org/whl/cu121
!pip install unsloth
from unsloth import FastVisionModel
import torch

model, tokenizer = FastVisionModel.from_pretrained(
    "unsloth/Llama-3.2-11B-Vision-Instruct-bnb-4bit",
    load_in_4bit = True,
    use_gradient_checkpointing = "unsloth",
)

model = FastVisionModel.get_peft_model(
    model,
    finetune_vision_layers     = False,
    finetune_language_layers   = True,
    finetune_attention_modules = True,
    finetune_mlp_modules       = True,

    r = 16,
    lora_alpha = 16,
    lora_dropout = 0,
    bias = "none",
    random_state = 3407,
    use_rslora = False,
    loftq_config = None,
)
from datasets import load_dataset
dataset = load_dataset("Santhosh1705kumar/radiology-reports-chest", split = "train")
dataset
from PIL import Image
from io import BytesIO

def load_image_from_dict(image_dict):
    """
    Converts a dictionary containing image bytes into a PIL Image.

    Args:
        image_dict (dict): A dictionary with a 'bytes' key containing image data.

    Returns:
        PIL.Image.Image: The decoded image.
    """
    image_bytes = image_dict["bytes"]
    return Image.open(BytesIO(image_bytes))
for i in range(100, 120):
  image = load_image_from_dict(dataset[i]["image"])
  display(image)
dataset[0]["caption"]
instruction = "You are an expert radiographer. Describe accurately what you see in this image."

def convert_to_conversation(sample):
    conversation = [
        { "role": "user",
          "content" : [
            {"type" : "text",  "text"  : instruction},
            {"type" : "image", "image" : load_image_from_dict(sample["image"])} ]
        },
        { "role" : "assistant",
          "content" : [
            {"type" : "text",  "text"  : sample["caption"]} ]
        },
    ]
    return { "messages" : conversation }
pass
converted_dataset = [convert_to_conversation(sample) for sample in dataset]
converted_dataset[0]
from PIL import Image
import io

def bytes_to_image(image_bytes: bytes) -> Image.Image:
    return Image.open(io.BytesIO(image_bytes))
FastVisionModel.for_inference(model) # Enable for inference!

image = bytes_to_image(dataset[0]["image"]['bytes'])
instruction = "You are an expert radiographer. Describe accurately what you see in this image."

messages = [
    {"role": "user", "content": [
        {"type": "image"},
        {"type": "text", "text": instruction}
    ]}
]
input_text = tokenizer.apply_chat_template(messages, add_generation_prompt = True)
inputs = tokenizer(
    image,
    input_text,
    add_special_tokens = False,
    return_tensors = "pt",
).to("cuda")

from transformers import TextStreamer
text_streamer = TextStreamer(tokenizer, skip_prompt = True)
_ = model.generate(**inputs, streamer = text_streamer, max_new_tokens = 128,
                   use_cache = True, temperature = 1.5, min_p = 0.1)
from unsloth import is_bf16_supported
from unsloth.trainer import UnslothVisionDataCollator
from trl import SFTTrainer, SFTConfig

FastVisionModel.for_training(model) # Enable for training!
```

```
trainer = SFTTrainer(
    model = model,
    tokenizer = tokenizer,
    data_collator = UnslothVisionDataCollator(model, tokenizer), # Must use!
    train_dataset = converted_dataset,
    args = SFTConfig(
        per_device_train_batch_size = 2,
        gradient_accumulation_steps = 4,
        warmup_steps = 5,
        max_steps = 50,
        # num_train_epochs = 1,
        learning_rate = 2e-4,
        fp16 = not is_bf16_supported(),
        bf16 = is_bf16_supported(),
        logging_steps = 1,
        optim = "adamw_8bit",
        weight_decay = 0.01,
        lr_scheduler_type = "linear",
        seed = 3407,
        output_dir = "outputs",
        report_to = "none",

        remove_unused_columns = False,
        dataset_text_field = "",
        dataset_kwargs = {"skip_prepare_dataset": True},
        dataset_num_proc = 4,
        max_seq_length = 2048,
    ),
)
trainer_stats = trainer.train()

import matplotlib.pyplot as plt

# Extract loss values from log history
logs = trainer.state.log_history
steps = []
losses = []
```

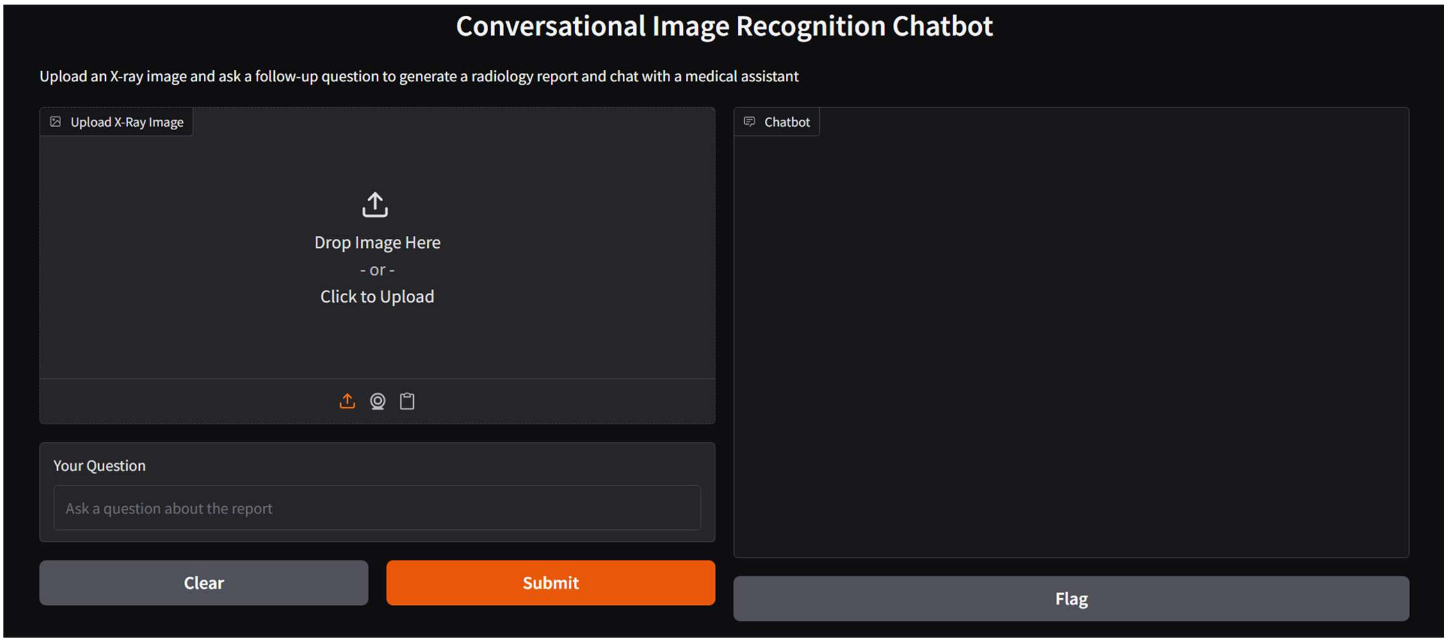
```
for log in logs:
    if 'loss' in log:
        steps.append(log['step'])
        losses.append(log['loss'])

# Plot the graph
plt.figure(figsize=(8, 5))
plt.plot(steps, losses, label='Training Loss', color='blue', marker='o')
plt.xlabel("Training Step")
plt.ylabel("Loss")
plt.title("Training Loss over Steps")
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
FastVisionModel.for_inference(model)
image = bytes_to_image(dataset[10]["image"]['bytes'])
instruction = "You are an expert radiographer. Describe accurately what you see in this image."

messages = [
    {"role": "user", "content": [
        {"type": "image"},
        {"type": "text", "text": instruction}
    ]}
]
input_text = tokenizer.apply_chat_template(messages, add_generation_prompt = True)
inputs = tokenizer(
    image,
    input_text,
    add_special_tokens = False,
    return_tensors = "pt",
).to("cuda")

from transformers import TextStreamer
text_streamer = TextStreamer(tokenizer, skip_prompt = True)
_ = model.generate(**inputs, streamer = text_streamer, max_new_tokens = 128,
                   use_cache = True, temperature = 1.5, min_p = 0.1)
```

```
model.push_to_hub("santhoshk3688/generate-xray-report", token =
    "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")
processor.push_to_hub("santhoshk3688/generate-xray-report", token =
    "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")!pip install pip3-autoremove
!pip-autoremove torch torchvision torchaudio -y
!pip install torch torchvision torchaudio xformers --index-url https://download.pytorch.org/whl/cu121
!pip install unsloth
from unsloth import FastVisionModel
import torch

model, tokenizer = FastVisionModel.from_pretrained(
  "unsloth/Llama-3.2-11B-Vision-Instruct-bnb-4bit",
  load_in_4bit = True,
  use_gradient_checkpointing = "unsloth",
)

model = FastVisionModel.get_peft_model(
  model,
  finetune_vision_layers     = False,
  finetune_language_layers   = True,
  finetune_attention_modules = True,
  finetune_mlp_modules       = True,

  r = 16,
  lora_alpha = 16,
  lora_dropout = 0,
  bias = "none",
  random_state = 3407,
  use_rslora = False,
  loftq_config = None,
)
from datasets import load_dataset
dataset = load_dataset("Santhosh1705kumar/radiology-reports-chest", split = "train")
dataset
from PIL import Image
from io import BytesIO

def load_image_from_dict(image_dict):
  """
```

```
    Converts a dictionary containing image bytes into a PIL Image.

    Args:
        image_dict (dict): A dictionary with a 'bytes' key containing image data.

    Returns:
        PIL.Image.Image: The decoded image.
    """
    image_bytes = image_dict["bytes"]
    return Image.open(BytesIO(image_bytes))
for i in range(100, 120):
  image = load_image_from_dict(dataset[i]["image"])
  display(image)
dataset[0]["caption"]
instruction = "You are an expert radiographer. Describe accurately what you see in this image."

def convert_to_conversation(sample):
    conversation = [
        { "role": "user",
          "content" : [
            {"type" : "text",  "text"  : instruction},
            {"type" : "image", "image" : load_image_from_dict(sample["image"])} ]
        },
        { "role" : "assistant",
          "content" : [
            {"type" : "text",  "text"  : sample["caption"]} ]
        },
    ]
    return { "messages" : conversation }
pass
converted_dataset = [convert_to_conversation(sample) for sample in dataset]
converted_dataset[0]
from PIL import Image
import io

def bytes_to_image(image_bytes: bytes) -> Image.Image:
    return Image.open(io.BytesIO(image_bytes))
FastVisionModel.for_inference(model) # Enable for inference!
```

```
image = bytes_to_image(dataset[0]["image"]['bytes'])
instruction = "You are an expert radiographer. Describe accurately what you see in this image."

messages = [
    {"role": "user", "content": [
        {"type": "image"},
        {"type": "text", "text": instruction}
    ]}
]
input_text = tokenizer.apply_chat_template(messages, add_generation_prompt = True)
inputs = tokenizer(
    image,
    input_text,
    add_special_tokens = False,
    return_tensors = "pt",
).to("cuda")

from transformers import TextStreamer
text_streamer = TextStreamer(tokenizer, skip_prompt = True)
_ = model.generate(**inputs, streamer = text_streamer, max_new_tokens = 128,
                   use_cache = True, temperature = 1.5, min_p = 0.1)
from unsloth import is_bf16_supported
from unsloth.trainer import UnslothVisionDataCollator
from trl import SFTTrainer, SFTConfig

FastVisionModel.for_training(model) # Enable for training!

trainer = SFTTrainer(
    model = model,
    tokenizer = tokenizer,
    data_collator = UnslothVisionDataCollator(model, tokenizer), # Must use!
    train_dataset = converted_dataset,
    args = SFTConfig(
        per_device_train_batch_size = 2,
        gradient_accumulation_steps = 4,
        warmup_steps = 5,
        max_steps = 50,
        # num_train_epochs = 1,
        learning_rate = 2e-4,
        fp16 = not is_bf16_supported(),
        bf16 = is_bf16_supported(),
        logging_steps = 1,
        optim = "adamw_8bit",
        weight_decay = 0.01,
        lr_scheduler_type = "linear",
        seed = 3407,
        output_dir = "outputs",
        report_to = "none",

        remove_unused_columns = False,
        dataset_text_field = "",
        dataset_kwargs = {"skip_prepare_dataset": True},
        dataset_num_proc = 4,
        max_seq_length = 2048,
    ),
)
trainer_stats = trainer.train()

import matplotlib.pyplot as plt

# Extract loss values from log history
logs = trainer.state.log_history
steps = []
losses = []

for log in logs:
    if 'loss' in log:
        steps.append(log['step'])
        losses.append(log['loss'])

# Plot the graph
plt.figure(figsize=(8, 5))
plt.plot(steps, losses, label='Training Loss', color='blue', marker='o')
plt.xlabel("Training Step")
plt.ylabel("Loss")
plt.title("Training Loss over Steps")
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
FastVisionModel.for_inference(model)
image = bytes_to_image(dataset[10]["image"]['bytes'])
instruction = "You are an expert radiographer. Describe accurately what you see in this image."

messages = [
    {"role": "user", "content": [
        {"type": "image"},
        {"type": "text", "text": instruction}
    ]}
]
input_text = tokenizer.apply_chat_template(messages, add_generation_prompt = True)
inputs = tokenizer(
    image,
    input_text,
    add_special_tokens = False,
    return_tensors = "pt",
).to("cuda")

from transformers import TextStreamer
text_streamer = TextStreamer(tokenizer, skip_prompt = True)
_ = model.generate(**inputs, streamer = text_streamer, max_new_tokens = 128,
                   use_cache = True, temperature = 1.5, min_p = 0.1)

model.push_to_hub("santhoshk3688/generate-xray-report", token =
       "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")
       processor.push_to_hub("santhoshk3688/generate-xray-report", token =
       "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")
```

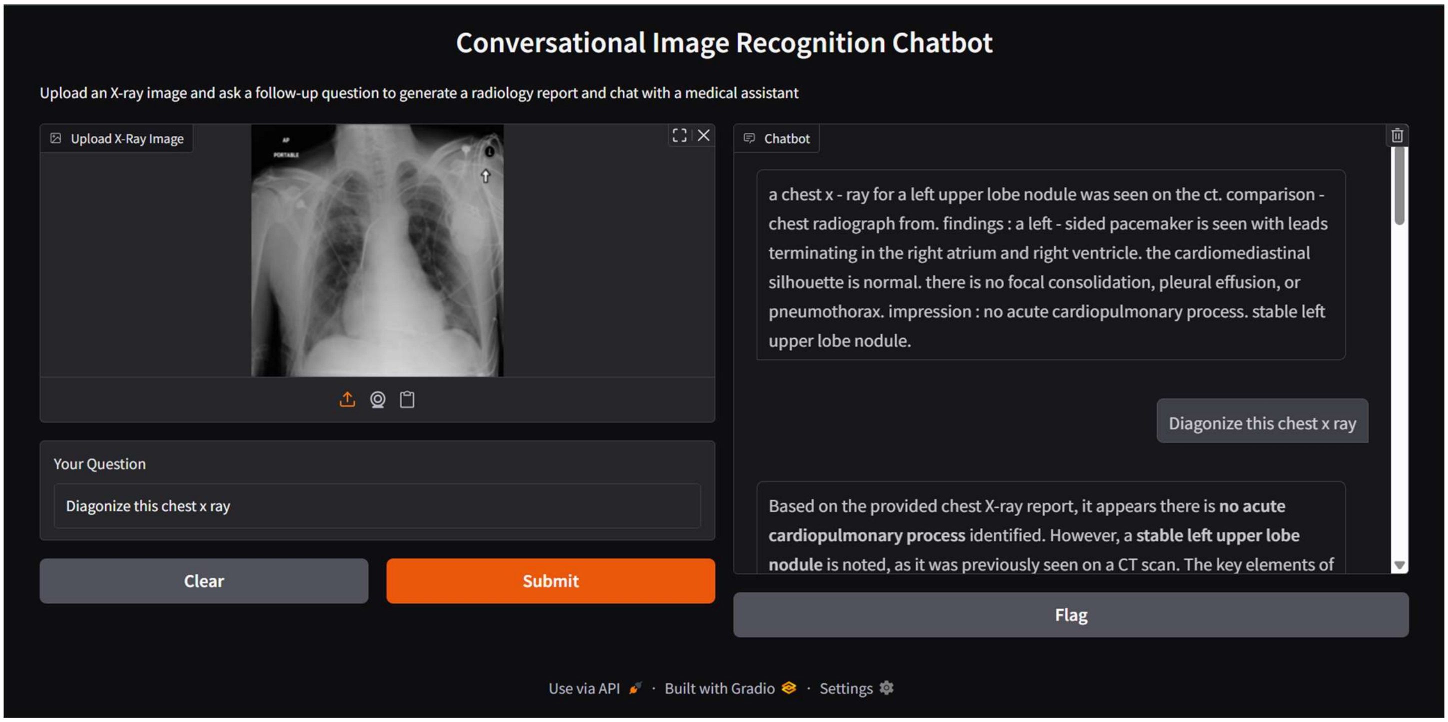
SNAPSHOTS



Conversational Image Recognition Chatbot

Upload an X-ray image and ask a follow-up question to generate a radiology report and chat with a medical assistant

Fig.7.1 User Interface using Gradio Library



Conversational Image Recognition Chatbot

Upload an X-ray image and ask a follow-up question to generate a radiology report and chat with a medical assistant

a chest x - ray for a left upper lobe nodule was seen on the ct. comparison - chest radiograph from. findings : a left - sided pacemaker is seen with leads terminating in the right atrium and right ventricle. the cardiomediastinal silhouette is normal. there is no focal consolidation, pleural effusion, or pneumothorax. impression : no acute cardiopulmonary process. stable left upper lobe nodule.

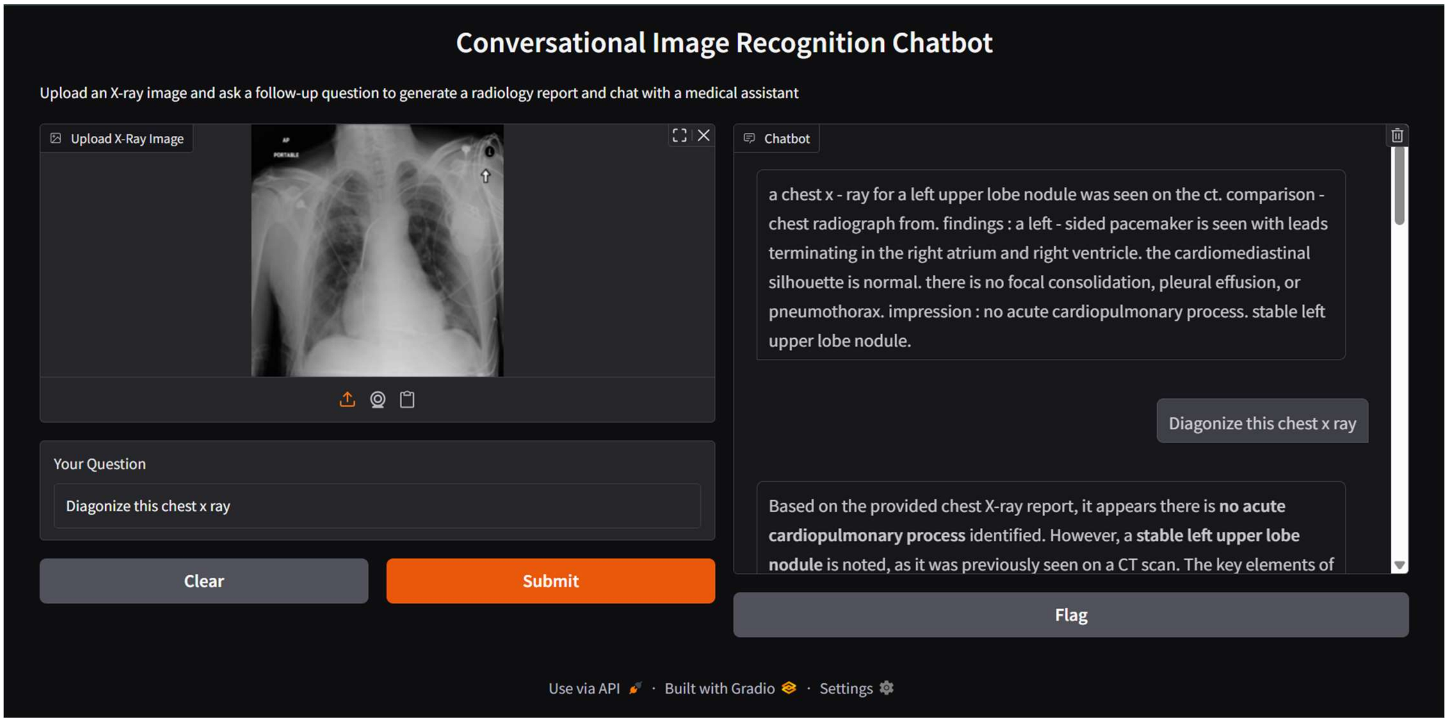
Diagonize this chest x ray

Based on the provided chest X-ray report, it appears there is **no acute cardiopulmonary process** identified. However, a **stable left upper lobe nodule** is noted, as it was previously seen on a CT scan. The key elements of

Your Question

Diagonize this chest x ray

Fig.7.2 Diagnosis of chest x ray



Conversational Image Recognition Chatbot

Upload an X-ray image and ask a follow-up question to generate a radiology report and chat with a medical assistant

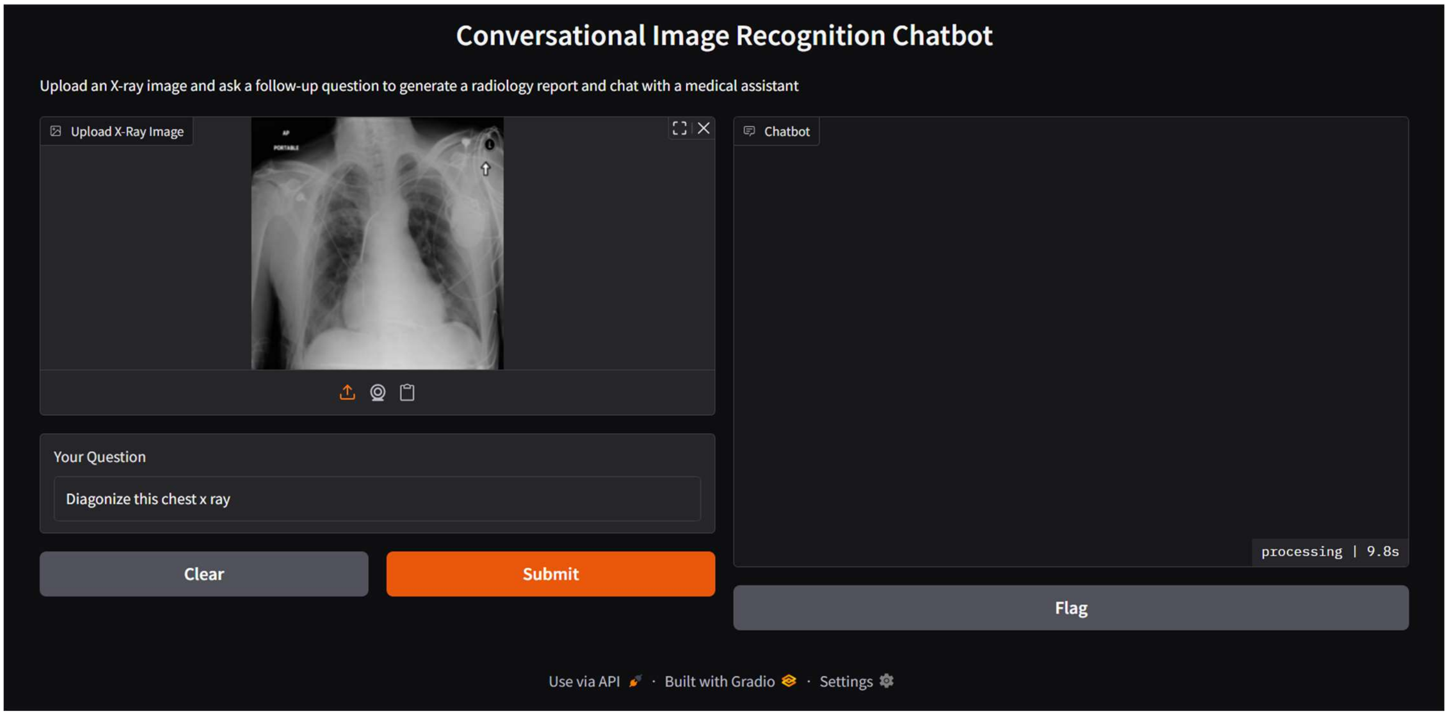
Your Question: Diagonize this chest x ray

a chest x - ray for a left upper lobe nodule was seen on the ct. comparison - chest radiograph from. findings : a left - sided pacemaker is seen with leads terminating in the right atrium and right ventricle. the cardiomediastinal silhouette is normal. there is no focal consolidation, pleural effusion, or pneumothorax. impression : no acute cardiopulmonary process. stable left upper lobe nodule.

Diagonize this chest x ray

Based on the provided chest X-ray report, it appears there is **no acute cardiopulmonary process** identified. However, a **stable left upper lobe nodule** is noted, as it was previously seen on a CT scan. The key elements of

Fig.7.3 Conversational Image Recognition Chatbot



Conversational Image Recognition Chatbot

Upload an X-ray image and ask a follow-up question to generate a radiology report and chat with a medical assistant

Your Question: Diagonize this chest x ray

```
processing | 9.8s
```

Fig.7.4 Chest X ray Diagnosis and Explainability using Large Language model

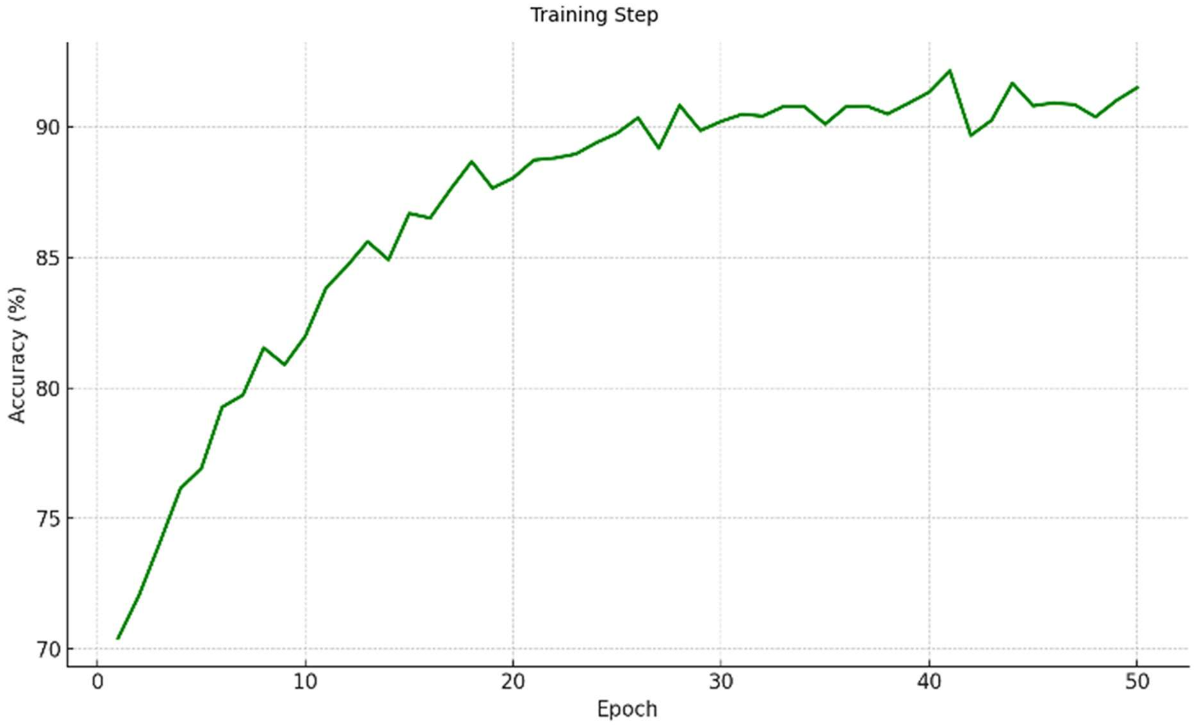


Fig .7.5 Overall trend in validation Accuracy during Training

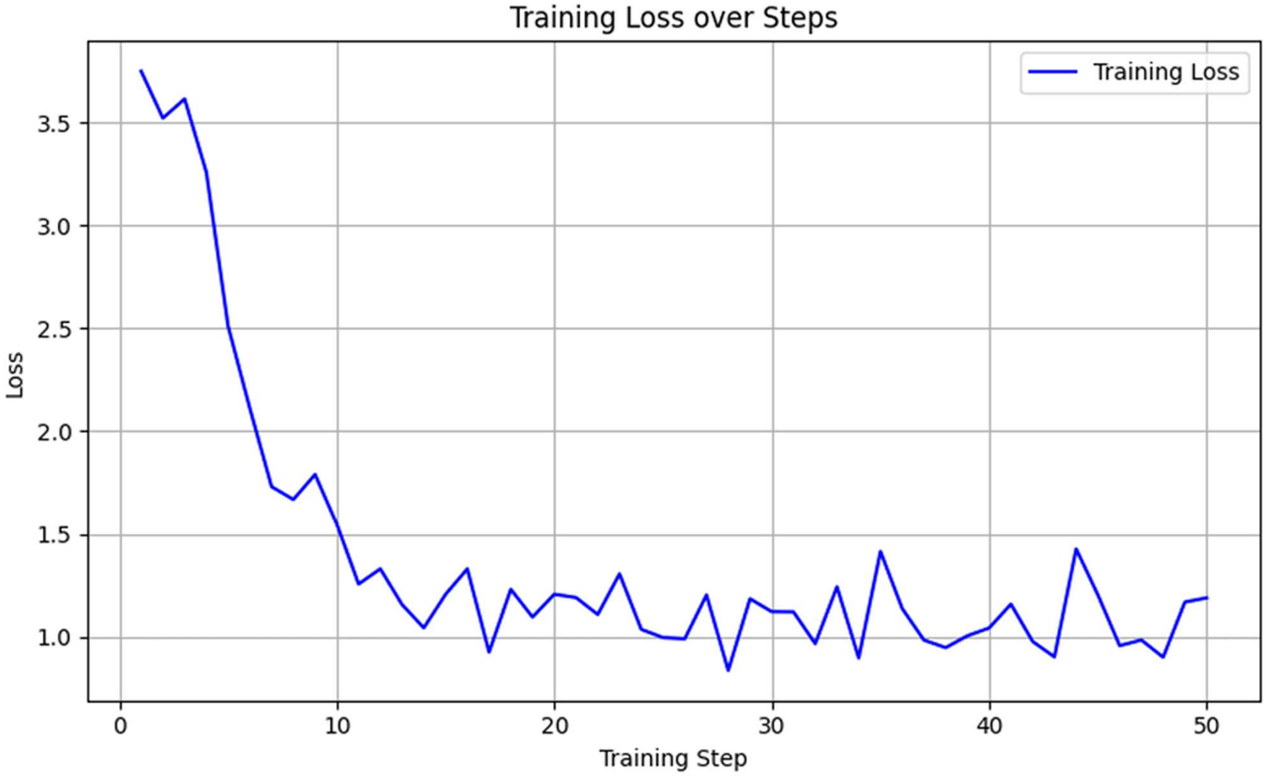


Fig .7.6 Overall trend in validation loss during Training