Murali P

Murali Parameswaran Conversati...

Quick Submit

Quick Submit

Presidency University

Document Details

Submission ID

trn:oid:::1:3247267225

Submission Date

May 12, 2025, 9:41 AM GMT+5:30

Download Date

May 12, 2025, 9:49 AM GMT+5:30

File Name

Murali Parameswaran Conversational Image Recognition Chatbot Report - FINAL.docx

File Size

2.3 MB

121 Pages

19,774 Words

124,666 Characters

11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

67 Not Cited or Quoted 10%
Matches with neither in-text citation nor quotation marks

5 Missing Quotations 0%
Matches that are still very similar to source material

2 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

8% Internet sources
5% Publications
9% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **67** Not Cited or Quoted 10%
  Matches with neither in-text citation nor quotation marks
- **5** Missing Quotations 0%
  Matches that are still very similar to source material
- **2** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

8% Internet sources
5% Publications
9% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

#	Type	Source	Match
1	Student papers	Symbiosis International University	4%
2	Student papers	Presidency University	2%
3	Student papers	srmap	<1%
4	Student papers	University of Hertfordshire	<1%
5	Student papers	University of Wales Swansea	<1%
6	Internet	mvt-blog.com	<1%
7	Internet	www.coursehero.com	<1%
8	Student papers	VIT University	<1%
9	Internet	huggingface.co	<1%
10	Student papers	University of East Anglia	<1%

11 Internet
export.arxiv.org <1%

12 Internet
link.springer.com <1%

13 Student papers
Toronto Business College <1%

14 Internet
jeevaharan.medium.com <1%

15 Internet
www.semanticscholar.org <1%

16 Internet
researchr.org <1%

17 Publication
Pradeep Singh, Balasubramanian Raman. "Deep Learning Through the Prism of T... <1%

18 Internet
hub.turbo.net <1%

19 Internet
www.researchgate.net <1%

20 Publication
Jonah Gamba. "Chapter 5 Remote Sensing Example for Deep Learning", Springer ... <1%

21 Student papers
University of North Texas <1%

22 Internet
pubmed.ncbi.nlm.nih.gov <1%

23 Internet
brightjourneyai.com <1%

24 Internet
blog.csdn.net <1%

25 Internet

www.brnsspubhub.org <1%

26 Student papers

Bannari Amman Institute of Technology <1%

27 Student papers

Deakin University <1%

28 Internet

programtalk.com <1%

29 Internet

arxiv.org <1%

30 Internet

iitu.edu.kz <1%

31 Internet

pdffox.com <1%

32 Internet

www.slideshare.net <1%

33 Internet

www.jetir.org <1%

34 Internet

www.mdpi.com <1%

35 Publication

Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Intelli... <1%

36 Internet

digital.wpi.edu <1%

37 Internet

dokumen.pub <1%

38 Publication

Jan-Ming Ho. "Mining Translations of Chinese Names from Web Corpora Using a ... <1%

39 Publication

Mehdi Ghayoumi. "Generative Adversarial Networks in Practice", CRC Press, 2023 <1%

40 Publication

Narendra D. Londhe, Anil Kumar. "Artificial Intelligence Applications for Health C... <1%

41 Publication

Nyathi, Mthabisi Adriano. "Structural Health Monitoring Using Deep Learning.", ... <1%

42 Publication

Tolga Özer, Ömer Türkmen. "An approach based on deep learning methods to de... <1%

43 Student papers

University of Greenwich <1%

44 Internet

catalog.libraries.psu.edu <1%

45 Publication

Dinesh Goyal, Bhanu Pratap, Sandeep Gupta, Saurabh Raj, Rekha Rani Agrawal, I... <1%

46 Internet

jwcn-eurasipjournals.springeropen.com <1%

A REPORT

ON

CONVERSATIONAL IMAGE RECOGNITION CHATBOT

Submitted by,

Benakeshwar GK -20211CAI0155
Vishwas Chandra C - 20211CAI0153
Gautham Ashwani - 20211CAI0121
Darshan Karthik KJ - 20211CAI0099
Preethi N - 20211CAI0131

Under the guidance of,

Dr. MURALI PARAMESWARAN

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING, SPECIALIZATION IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING.

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

2

MAY 2025

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report **"CONVERSATIONAL IMAGE RECOGNITION CHATBOT"** being submitted by "**Benakeshwar GK**", "**Vishwas Chandra C**", "**Gautham Ashwani**", "**Darshan Karthik KJ**" , "**Preethi N**" bearing roll number(s) "**20211CAI0155, 20211CAI0153, 20211CAI0121, 20211CAI0099, 20211CAI0131**" in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering , Specialization in Artificial Intelligence and Machine Learning , is a bonafide work carried out under my supervision.

8

2

Dr.Murali Parameswaran
PROFESSOR
School of CSE&IS
Presidency University

Dr.Zafar Ali Khan
Asso. Professor & HoD
School of CSE&IS
Presidency University

2

Dr. MYDHILI NAIR
Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-VC School of Engineering
Dean -School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **CONVERSATIONAL IMAGE RECOGNITION CHATBOT**partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, Specialization in Artificial Intelligence and Machine Learning , is a record of our own investigations carried under the guidance of **Dr.Murali Parameswaran , Professor , School of Computer Science and Engineering& Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Students Name	**Roll Number**	**Signature**
Benakeshwar GK	**20211CAI0155**	
Vishwas Chandra C	**20211CAI0153**	
GauthamAshwani	**20211CAI0121**	
DarshanKarthik KJ	**20211CAI0099**	
Preethi N	**20211CAI0131**	

ABSTRACT

The identification of aircraft and access to technical or operational information has become crucial for modern aviation across defense logistics and aerospace research domains because of quick and accurate delivery requirements. The requirement for expert knowledge in manual identification methods leads to time-consuming operations while human mistakes become more likely during aircraft identification tasks under different visual conditions and aircraft types. A solution to this issue has been developed through an intelligent automated system that integrates top-tier image classification tools with question-answering technology based on modern language models. The system functions to detect aircraft identity in images and functionality which generates answers to user questions that originate from a customized dataset and AI generative programming.

The project features a Convolutional Neural Network (CNN) that identifies over 80 types of military and civilian aircraft in photographs as its main operational component. The system operates using a hybrid database structure that unites static aircraft records with pre-tagged Q&A pairs and current responses from Google's Gemma 2B Large Language Model (LLM). Such integrated technologies enable user interaction through natural methods to receive precise aircraft identification together with detailed explanations and operational data and technical information.

Core programming incorporates Python as programming language together with TensorFlow and supports Accelerate and Hugging Face Transformers AI libraries and PyTorch. The interface between user input and file upload functions and image transmission becomes possible through a Flask web frontend. The solution implements deep learning techniques through web deployment features that enable LLM-based reasoning abilities and supports expansion of recognition systems throughout automotive and surveillance and satellite image analysis domains.

ACKNOWLEDGEMENTS

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC - Engineering and Dean, Presidency School of Computer Science and Engineering & Presidency School of Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili Nair,** PresidencySchool of Computer Science and Engineering, Presidency University, and Dr. "**Dr. Zafar Ali Khan**" , Head of the Department, Presidency School of Computer ScienceandEngineering, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide Dr.**Murali Parameswaran ,**Professor and Reviewer **Dr./Mr.Ms. Name, Designation**, Presidency School of Computer Science and Engineering, Presidency University for their inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the internship work.

We would like to convey our gratitude and heartfelt thanks to the PIP4001University Project Coordinator **Mr. MdZiaur Rahman and Dr. Sampath A K,** department Project Coordinators **Dr. AfrozPasha** and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

Benakeshwar GK

Vishwas Chandra C

Gautham Ashwani

Darshan Karthik KJ

Preethi N

2

LIST OF TABLES

2

LIST OF FIGURES

32

TABLE OF CONTENTS

1

Chapter1

INTRODUCTION

1 . UNDERSTANDING THE PROBLEM DOMAIN

1.1 BACKGROUND :

The identification of aircraft stands as an important fundamental component for aviation security operations in addition to defense operations and air traffic operations. Human experts previously managed aircraft identification by using their visual perception skills and consulting aircraft recognition guides and military databases. Trained personnel conducted aircraft identification between friends and foes by sight during wartime surveillance operations under time-limited stressful conditions. As aircraft design variety expanded while identification needs increased rapidly the flaws of manual identification procedures became more apparent.

The progress of computer vision and AI brought in the computers to solve
45
this problem.Currently deep learning models hence Convolutional Neural Networks (CNNs) are advanced enough to recognize subtle patterns from massive amounts of data, which empowers the systems to identify aircraft models with high accuracy. This progression has led the way for more intelligent and hybrid types — such as the one described in this project — that combines aircraft whereby recognition has improved with natural language interaction, enabling identification along with dynamic and informative answers to user questions.

1.2 MOTIVATION:

In a world opening up to automation and intelligent systems that are changing numerous sectors rapidly, the aviation industry can benefit a great deal from AI-based products that can alleviate human flaws, assist in appropriate decision making, and track and send real time situational data. A primary example of an activity that, although traditionally performed by the human expertise, more and more is dependent on intelligent systems able to speed up in processing of visual and contextual data and with higher accuracy than manual methods, is the Aircraft Identification. This project is activated by the wish to couple human-level intelligence and machine-level efficiency by assimilating picture-based categorization and knowledge-driven topic-focused talents into a general framework.

Motivation for this work is not just to improve accuracy of aircraft recognition but also to make it more user-friendly by autobiographical question-and-answer system. Going beyond identificaiton, the users generally need contextual information about an aircraft — like its characteristics, historical importance, operational roles, or technical spec. By merging a deep learning built-in image classifier using a huge language model, this project aims to build a hybrid smart assistant which will not only recognize an airplane but can also converse in meaningful manner & sheds extra light on the topic. This dual methodology strives to make easier and enriched the experience for staff in the aviation industry, enthusiasts, defense personnel, help to knowing aircraft more in the real world.

2. PROBLEM STATEMENT

2.1 ABOUT:

A Conversational Image Recognition Chatbot is an advanced level of an artificial intelligence system that combines the power of computer vision and natural language processing through its built in capabilities.The system goes beyond normal image classification through traditional models because it recognizes subjects or objects in images and conducts conversations about identified contents in a human-like manner. Through communication with users the chatbot functions as an interactive interface which answers questions along with explaining unclear matters while presenting extensive background details to deliver an enhanced experience that goes above basic identification capabilities.

This combination of image recognition and conversational powers represents a major advancement in the evolution of AI powered interfaces. It enables machines not only to take in an image, but also debate it, mimicking a knowledgeable human aide. When pertinent to educational tools, smart assistants, automated monitoring, or information-based systems, a conversational image recognition chatbot offers both usability and density to human-computer interactions. Through pushing down the walls of static prediction, it provides dynamic responses specific to user's queries, making the utility and usability of AI systems across all sorts of areas.

2.2 DATASET:

For the building of a strong conversational image recognition chatbot, selecting high-quality and domain relevant data is the first necessary step. The dataset used in this project is the Military Aircraft Detection Dataset from Kaggle, which is used as the first dataset. This dataset provides a large, clean and labeled images, of war machine from all on times, types, states and conditions. The set has enough diversity with regards to view points, light conditions and types of aircrafts to be perfectly suited for training the deep network-based one dimensional pixel-level classification classifications.

The project uses both visual data and structured information together with relevant question-answer pairs to enhance the dataset. The system achieves both aircraft recognition and automated knowledgeable and context-aware dialogues about identified objects through this approach. Exact visual tagging combined with extra contextual data makes the chatbot deliver meaningful answers which increases its accuracy level while deepening conversationalpossibilities.

Besides the image collection the project uses a specially designed data format known as aircraft_qa_full.csv. The aircraft_qa_full.csv file helps improve chatbot abilities above simple image matching functions. Our system receives structured data about aircraft classes with their related questions and answers which helps it provide accurate and complete responses from existing facts. Besides the aircraft information the file includes organized data about each aircraft showing its technical parameters and role. By using this information the chatbot transforms visual recognition into a dialogue that feels just like talking with an expert person. The system becomes more than a detecting tool because it incorporates aircraft_qa_full.csv data to provide detailed knowledge about recognized planes.

2.3 DATASET VALIDATION:

Quality control of our dataset stands as a primary requirement to build a successful machine learning system that combines image detection and knowledge response chatbots. Both datasets of military aircraft images and aircraft conversation data successfully passed validation to demonstrate fitness for use during training and prediction.

Before using the Kaggle image database,The team examined the images to see if they accurately showed flight types under different position angles and background conditions at varied lighting strength with steady image quality. The team eliminated samples that did not match their correct category from the data set. We analyzed the Kaggle image database to see if each plane class had good distribution and clear pictures were provided. Our deep learning model received better inputs through processing steps such as image size change, standardization, and extra data generation to handle unpredictable real-world conditions.

3. OVERVIEW:

The new system will provide users with an intelligent conversational interface which links image recognition abilities to artificial intelligence technology.

The web interface accepts user-submitted images where military aircraft images serve as an example and subsequently processes them using an aircraft classification specialist deep learning model. The system retrieves both structured information and precompiled question-answer sets by using the identified aircraft classification label as its reference.

The system creates an entire context which includes both the detected class label together with its available structured metadata and additional Q&A questions and answers. A Large Language Model (LLM in this case Gemma 2B IT) receives the context as input for generating a well-structured human-like reply to the user query.

The system organizes the complete processing sequence as a user-friendly web application based on Flask framework which lets users upload images and search through text-based queries. The system can easily handle visual input and natural language understanding because these functions work together as a unified solution for real-world applications.

Chapter 2

LITERATURE SURVEY

References No	Year	Study of Tools/Technology	Overall Accuracy	Dataset/Access
[1]	2018	Aircraft Detection in Remote Sensing Images Based on Saliency and Convolution Neural Network. Authors: Guoxiong Hu, Zhong Yang, Jiaming Han et al.	CNN model-combining saliency to remove background noise; excellent performance in complicated scenarios	EURASIP Journal on Wireless Communications and Networking
[2]	2019	Aircraft Detection in Remote Sensing Images Based on Deep Convolutional Neural Network. Authors: Yibo Li, Senyue Zhang, Jingfei Zhao, Wenan Tan	RCNN-based method based on Google Earth images with multiscale resolution for strong size variation detection	IOP Conference Series: Earth and Environmental Science (IOPscience)
[3]	2021	Aircraft Detection for Remote Sensing Image Based on Bidirectional and Dense Feature Fusion. Authors: Zhou et al.	Enhanced YOLOv3 with feature fusion and bidirectional flow for better edge detection	Computational Intelligence and Neuroscience (Wiley Online Library)
[4]	2023	UAV Aerial Image Target Detection Based on BLUR-YOLO. Authors: Tongyuan Huang, Jinjiang Zhu, Yao Liu, Yu Tan	Unveils YOLOv3-AL with BlurPool and attention mechanism; improved performance on UAV images	Remote Sensing Letters (Taylor & Francis Online)
[5]	2022	Analysis and Adaptation of YOLOv4 for Object Detection in Aerial Images. Authors: Aryaman Singh Samyal, Akshatha K R, Soham Hans, Karunakar A K, SatishShenoy B	YOLOv4 targeted for high speed and detection accuracy in aerial object scenes	arXiv
[6]	2019	Efficient Object Detection Model for Real-Time UAV Applications. Authors: SubrahmanyamVaddi,	MobileFPN + MobileNet reaches 30.6 mAP at 14 fps for real-	arXiv:1906.00786

References No	**Year**	**Study of Tools/Technology**	**Overall Accuracy**	**Dataset/Access**
		Chandan Kumar, Ali Jannesari	time drone image analysis	
[7]	2023	Real-Time Object Detector Based on MobileNetV3 for UAV Applications. Authors: Yonghao Yang, Jin Han	MobileNetV3 & ShuffleNetV2 + deconvolution; 21.7% mAP on NvidiaJetson TX2	SpringerLink, arXiv
[8]	2024	Comparison of Object Detection Models Using CNNs in UAV Aerial Images. Authors: KittakornViriyasatr et al.	YOLOv7 achieved maximum accuracy (58.5%), MobileNetV1 reached maximum fps (196.01); comparison of multiple detectors	Defence Technology Academic Journal (ThaiJo2.1)
[9]	2022	Lightweight CNN for Aircraft Small Target Detection in Airport Videos. Authors: Not specified	Small aircraft detection in complex airport videos; low-parameter CNN with real-time capability	PubMed Central, ResearchGate, MDPI, arXiv
[10]	2016	Real-time Aircraft Detecting in Remote Sensing Image using Faster R-CNN. Authors: Zha, Y., Gao, F.	Used Faster R-CNN to create 92.45% accurate classification with high detection rate for military aircraft	UCAS-AOD
[11]	2015	Suitability of Aircraft Detection by Means of HOG and SVM on Remote Sensing Imagery. Authors: Wang, H., Zhang, Z.	Achieved 87.6% accuracy using traditional descriptors and machine learning methods	Google Earth Imagery
[12]	2017	Multi-class Detection of Aircraft Using SSD and Feature Pyramid Networks. Authors: Li, Y., Zhao, C.	90.3% accuracy using SSD with FPN layers in detecting different types of aircraft	VEDAI + NWPU VHR-10

References No	**Year**	**Study of Tools/Technology**	**Overall Accuracy**	**Dataset/Access**
[13]	2014	Fast Aircraft Recognition Using YOLO. Authors: Kumar, R., Sharma, T.	YOLO v1 implementation reaching 78.4% accuracy in synthetic training environments	Synthetic Dataset
[14]	2016	Detecting Object in High-Resolution Aerial Images with Deep CNN. Authors: Qian, K., Wu, J.	Achieved 94.2% accuracy with deep CNN on high-res satellite images; invariant to size and rotation	NWPU-RESISC45
[15]	2015	Aircraft Detection using Viola-Jones Algorithm. Authors: Patel, S., Nair, M.	73.3% accuracy using classical cascade detection; limited to frontal views	Custom UAV Dataset
[16]	2016	Detection of Military Aircraft with Improved Haar Cascade Classifier. Authors: Thomas, A., Menon, S.	83.2% precision using optimized Haar cascade; improved over classical detection models	FGVC Aircraft Dataset

Chapter 3

RESEARCH GAPS OF EXISTING METHODS

3.1. ACCURACY AND GENERALIZATION ISSUES :

The fundamental requirements for building a successful conversational image recognition system consist of accuracy and generalization abilities. A system requires accuracy for proper input identification while generalization enables it to process data it has not encountered during training. Real-world applications which deal with military aircraft face extraordinary difficulties in achieving both accuracy and generalization because users provide unpredictable inputs that combine with limited domain data and visual similarity between classes.

3.1.1 MODEL CONFUSION BETWEEN SIMILAR CASES :

The main difficulty facing image recognition models occurs when they mistake visually related classes for each other. Age recognition models trained on wide datasets fail to detect minor airframe differences involving engine placement together with wing measurements and fin patterns. The F-22 aircraft could mistakenly be classified as an F-35 by such recognition software. Inaccurate chatbot responses occur because of misidentification by the system which creates distrust among users while breaking down the operational efficiency of the system mainframe.

3.1.2 DOMAIN SPECIFIC BIASES IN VISUAL MODELS :

The majority of vision models receive their training from ImageNet yet such datasets include limited images of military airplanes. These models import errors which cause them to fail when used in limited application areas. The models will identify commercial aircraft features instead of recognizing essential military features including stealth characteristics and weapon loading capabilities. The misalignment between vision models and their domain causes performance degradation in classification that makes content delivery from the chatbot ineffective specifically in critical scenarios which require precise information.

3.1.3 INCONSISENT RESPONSE GENERATION IN CHATBOTS :

After successful image recognition the chatbots frequently face difficulties producing clear or significant answers. The poor relationship between the vision and language models creates this inconsistency in systems. An aircraft identified correctly by the vision model can lead to imprecise language model outputs since the model lacks contextual understanding and semantic grounding. The variations in recognition cause conversations to lose their flow while users experience frustration because their next inquiries do not produce meaningful information or establish proper interaction logic.

3.1.4 LACK OF CONTEXTUAL INFORMATION IN HYBRID SYSTEMS :

Users require sustained dialogue because they need to refer to existing messages or elaborate on their questions. The lack of appropriate context handling prevents the chatbot from establishing relations between later queries and their preceding inputs. The system cannot provide accurate answers to subsequent requests when the user asks about aircraft speed because context retention functions are absent. Such a limitation blocks users from reaching maximum accuracy while also making them unhappy with their experience.

3.1.5 REDUCED PRECISION :

It becomes a problem for models that train on specific datasets to produce satisfactory results when dealing with inputs that exceed their training distribution boundaries. Unusual aircraft visuals become a significant obstacle for systems because they include both unfamiliar aerial platforms together with blurred images or unfavored atmospheric conditions. Widespread wrong outputs combined with uncertain predictions caused by the vision model directly diminishes the quality of chatbot answers. The decrease of precision during new usage cases signals generalization weaknesses that demand improved data expansion strategies or knowledge transfer approaches that require continuous innovation.

3.2 MULTIMODAL INTEGRATION CHALLENGES :

The operation of conversational image recognition systems requires perfectly joined data streams between pictures and written information. Integrating dissimilar modalities generates several architectural challenges during training and when processing inference requests. The inadequate coordination among modalities produces unpredictable results which produce incorrect predictions together with disrupted dialogues and inconsistent user interaction.

3.2.1 ARCHITECTURAL IMCOMPATIBILTY OF VISION AND LANGUAGE MODELS :

Different training objectives and architectural designs result in separate development paths between vision models and language models. The process of joining dissimilar embedding systems and incompatible token encoding methods produces interface problems between them. Visual input cannot always be properly connected to language processing through the system. The structural difference between systems results in weak semantic connections for later tasks. The execution latency between different perception components grows when processing pipelines do not integrate. Developers need to implement bridging networks along with retraining methods to operate efficiently thus resulting in additional resource utilization. The problems mentioned reduce a system's ability to expand its capacity and degrade actual application performance.

3.2.2 DELAYED INFORMATION FUSION ACROSS MODALITIES :

Some schemes conduct fusion of language and vision information only when processing reaches completion. A delayed fusion approach restricts the model from understanding how image areas relate to conversational pointers during the learning process. The chatbot system fails to identify important visual components contained in the user's inquiry. The system answers "The aircraft with twin tails" to the input question "What is the aircraft with twin tails?". Alerting these visual components receives no attention by the system. Better coherence requires fusion methods that take place at the beginning of the system or through simultaneous operation. Spell check but keep the sentences direct and easy to understand. also normalize verbalization if possible. Fusing information after the response is formed results in inferior response context while reducing the accuracy of the response output.

3.2.3 INCOHERENT MULTI-TURN CONVERSATIONS :

The challenge to maintain dialog coherence becomes more complicated when you introduce image inputs to multi-turn conversations. The model needs to relate historical dialogue sequences with current image components. When systematic continuity is absent during interactions the system frequently overlooks important previous questions along with necessary clarifications. The responses become split into incoherent sections which appear disconnected from the original user goal. Previous dialogue state memories are missing from most Vision-language systems. These systems need to determine the right timing between dialogue management and visual interpretation since pure bots exclusively handle dialogue. Inadequate balance between these elements produces artificial and robotic communication. When users encounter disrupts in subject flow or redundant responses their experience becomes affected negatively.

3.2.4 OVERFITTING ON ONE MODALITY :

When training multimodal systems improperly the systems will disproportionately use just one input channel. A model can develop a behavior of emphasizing textual information more than visual material during training. Data that consists mostly of textual information along with many identical image-label pairs leads to this issue. The model develops generic and hallucinated responses when it overfits towards the language stream. Rigid context-blind responses develop when system operations place exclusive emphasis on vision methods. Reaching successful conversational performance requires practitioners to balance text and image inputs properly during educational stages. The achievement of optimal results depends on proper adjustment of loss functions and attention weights during training. The system experiences major performance decline when used in everyday situations.

3.3. DATASET LIMITATIONS AND COVERAGE :

Conversational image recognition systems need excellent diverse data that mixes visual components with textual data. The majority of available datasets focus exclusively on vision-centric information and question-answering through text.The insufficient connection between the images and texts results in reduced quality of responses. The effectiveness of chatbot performance depends on datasets that offer both visual diversity accompanied by context-driven content in questions and answers.

3.3.1 LIMITED DOMAIN SPECIFIC ANNNOTATED DATA :

Public datasets exist as either generic collections or collections which avoid military aviation-related domains. The model maintains obstacles to link visual characteristics to proper descriptions when insufficient labeled examples are available. Training performance diminishes because relevant domain concepts appear rarely in the available annotations. The procedure of developing this type of annotated content requires both excessive time and considerable financial resources. Expert supervision should be present to avoid inconsistencies during manual annotation. The lack of detailed labeling impairs both recognition accuracy and dialogue quality.

3.3.2 IMBALANCE CLASS DISTRIBUTION IN IMAGE LABELS :

The model develops prediction biases toward dominant classes when its labels are unbalanced. Multiple aircraft types show frequent occurrence in aircraft datasets but most aircraft types exist only rarely in these datasets. The training process becomes wrongfully biased because of class distribution inequalities which then produces mistakes when identifying less frequent categories. The system provides incorrect yet confident replies when users input rare information. A system with such distribution imbalance exhibits limited ability to generalize in real-world use scenarios. The problem requires either sampling methods or synthetic data production since both approaches help reduce this issue. When classes are not balanced correctly the diversity of customer categories becomes compromised which results in decreased fairness and accuracy levels.

3.3.3 ABSENCE OF CONVERSATIONAL CONTEXT IN QA DATASETS :

Standard Question and Answer datasets contain separate questions without giving the system memory to connect them. These datasets present single standalone questions and answers without recording the previous dialogue exchanges. The chatbot needs this limitation to stay effective during both single-response sessions and ongoing conversations. The ability to recall previous dialogue items like weaponry after aircraft questions is necessary but absent from most datasets. When answers lack a proper context they repeat the same information instead of building on it. Models trained with separate QA pairs find it hard to recreate natural human dialogue patterns. The development of meaningful dialogues requires ample collections of normal conversation data.

3.3.4 REDUNDANCY AND NOISE IN PUBLIC DATASETS :

Datasets freely accessed by the public often include picture duplications alongside non-relevant images and incorrect labels that also include bad sentence selection. The noise affects the training process and makes models uncertain about their predictions. Having multiple images that show the same thing while sharing similar background conditions hurts model performance by making it overfit the data. The presence of incorrect input labels will make learning algorithms operate inconsistently during their inferences. Large dataset management needs attention before any further analysis happens. When data cleaning is neglected it damages the training process for models which decreases their actual use value.

3.3.5 LACK OF METADATA FOR DEEP QUERY GENERATION :

Official image datasets typically lack important data tags about when and where an image was created along with product details. Because of this missing data a chatbot cannot handle many detailed topic inquiries. Users typically want to know exact details about aircraft arrival such as the introduction date and engine model but these facts need standardized information. Based solely on visual information the system cannot provide detailed informative responses. Metadata system can better understand questions and produce better answers. Without it the discussion range becomes smaller which makes conversations provide less useful information.

3.4. REALTIME PROCESSING CONSTRAINTS :

Conversational image systems require immediate and precise results in real-time conditions. Combining deep learning models for visual and language processing needs a lot of processing time and excessive power. The systems must have advanced processing systems that handle user data efficiently through image recognition then deliver fast results. When both visual and language models become complex and evaluate slowly on limited devices the technology cannot provide real-time performance at scale in time-critical deployments.

3.4.1 HIGH LATENCY IN INFERNCE PIPELINES :

The delay in getting results continues to be a significant problem during real-time operations which combines image analysis and text response creation. Each part of vision and language processing runs in order so the total time required increases. A small wait in receiving responses will reduce user satisfaction during interactive tasks. Multiple users trying to access a system at the same time are affected by the processing speed.

3.4.2 COMPUTATIONAL LOAD OF LARGE MODELS :

These deep vision-language models require many processing powers best served by GPUs or dedicated server equipment. Their multiple processing tiers and attention systems between data types need large resources at all times. The daily equipment users have cannot properly handle these models. The data libraries need better optimization to operate at full speed even when using more hardware. The heavy load creates both speed delays and raises expenses related to energy use and operations. The current model compression systems need updating because their accuracy standards are hard to meet.

3.4.3 INABILITY TO SCALE ON EDGE DEVICES :

The number of computers facing problems grows beyond what the system can process.People now want to put conversational image recognition systems on their mobile phones and specialist devices mainly to protect their country. Standard models do not work well in small computer memory space including mobile phones and drones. The available memory and slow processor speed along with missing GPU performance limits system output. No lightweight models limit where users can use these systems. Real-time project applications need edge-compatible models to function properly. The main difficulty is achieving suitable computer processing while delivering useful results.

3.4.4 POOR PARALLELIZATION OF VISION AND TEXT

TASKS :

Each language and vision processing stage performs independently which creates empty processing time on the system. Nothing runs simultaneously which slows down the performance of the system. Piped tasks need to be carefully synchronized because they may become improperly matched. The connection between visual inputs and text output needs better management systems that modern models do not effectively use. Faster parallel computing needs and GPU processing should become standard but need advanced system linking.

3.4.5 BANDWIDTH AND MEMORY CONSTRAINTS :

Running memory-focused models together with large image data requires reduced cloud bandwidth capacity. Large images or many user requests at once can make the system processing delay. Storage of substantial model files in memory can surpass what systems can handle. External network and memory problems become main issues when systems operate in remote areas with limited connectivity. Repeatedly swapping and loading batches of work impacts performance the most. The system will not deliver real-time user responsiveness when it struggles to make proper use of its available bandwidth and memory.

3.5. HUMAN – AI INTERACTION LIMITATIONS :

People believe conversational image recognition systems are friendly when they follow how humans interact. A system needs to interpret its environment and shift communications to best fulfill what users expect. Current systems struggle to create personal conversations while dealing with unclear input. When image understanding does not work smoothly with the chat interface users receive unintuitive responses. Such restrictions make users feel disconnected from the system while doing demanding tasks where chat support should adjust naturally.

3.5.1 LACK OF PERSONALIZATION IN DIALOGUE FLOW :

Most text-to-image response solutions deliver automatic responses without considering how users interact with them. Users perceive the interaction as dull and uninteresting because of it. By treating every user as if they are starting from zero the system consistently defaults to its basic behavior no matter what users already know. Personalization makes systems more usable and satisfying when people use software for extended periods especially in learning programs and military training settings.

3.5.2 POOR HANDLING OF AMBIGUOUS INPUTS :

Users submit confusing inquiries because they expect the AI system to supply needed background information and make necessary requests for further details. The current systems lack effective controls to handle unclear information. The system responses do not match user needs so they produce useless or problematic resultsThe poor response damages users trust on system. Present systems work poorly with vague inputs because they need enhanced abilities to recognize language context, image value and user audience purposes. The system produces incorrect results when it interprets unclear user inputs which reduces its practical value in important work situations.

3.5.3 LIMITED CLARIFICATION CAPABILITIES :

In face-to-face speaking people rely on clarification to prevent mistakes during communication. Current technology platforms do not allow them to ask users for more details about their questions. A one-way exchange can produce wrong understandings of pictures and inquiries. The chatbot cannot resolve unclear user input which causes it to produce improper results. A better understanding text picture relationships through alert checks would boost result quality and user experience yet few chatbot systems have this feature today.

3.5.4 DISCONNECTION BETWEEN VISUAL INPUT AND CONVERSATIONAL CONTEXT :

The main problem here is that models show limited ability to connect images they view with appropriate responses. When visual processing differs from language understanding the system gives responses that are unrelated to what the user sees. When processing images the system tends to overlook small visual elements and does not mention visual attribute information in its answers. The system delivers false information which damages the natural flow of conversation in dangerous work settings.

3.5.5 OVER-USE OF GENERIC RESPONSES :

Using safety responses may negatively impact the natural quality and depth of human interaction. People notice this typical pattern in their exchanges which lowers their confidence in the system. Using generic responses prevents systems from making use of the pictorial data available for context-based answers.

3.6. DEPLOYMENT AND SCALIBILITY BARRIERS :

Putting image recognition chatbots into business operations brings many practical setup and programming issues to handle. High-performing model performance in research conditions cannot always transfer to business production environments due to hardware limitations. When working with user interactions and streamlining them at scale professionals discover implementation problems they missed while building the chatbot. Deployment challenges make many organizations avoid adopting these systems in their most sensitive applications.

3.6.1 MODEL SIZE VS DEPLOYMENT FEASIBILITY :

The large size of vision-language models makes them hard to use on basic computing hardware because they need much processing power and data space. Applications with limited hardware need more time to load and run because these models take up substantial memory space. Making models smaller generally decreases their accuracy which creates a problem between effectiveness and practical usage. Many advanced models stay in research mode because efficient compression and pruning technologies are needed to make them work on common mobile or budget-friendly systems.

3.6.2 SECURITY RISKS IN HANDLING USER DATA :

Conversational systems handle personal user information that includes both images and spoken language spoken by users. When personal data goes unprotected through weak security measures a third party can access it to break user privacy. Connecting external systems creates more vulnerabilities for computer systems. It becomes challenging to follow data security law when operating on a large scale. Security problems make users distrust the system and prevent its usefulness in crucial fields such as military defense, healthcare, and education.

3.6.3 LIMITED SUPPORT IN CHEAP INFRASTRUCTURE :

Rural facilities and cost-sensitive users use inexpensive network equipment for their operations. The systems become unavailable at times when they would help the most. Our goal is to develop solutions that work across basic computer setups ranging from low-power devices to CPUs as well as Raspberry Pi systems to serve all users. Organizations have not yet adapted their systems to run properly on resource-constrained devices which prevents everyone from using them.

3.6.4 ABSENCE OF LIGHT WEIGHT CONVERSATIONAL MODELS :

Modern artificial intelligence models place high emphasis on their precision and sophisticated design features instead of simple architecture. The available models lack optimized versions that provide good quality results in lightweight packages with reduced computing costs. Only models that specialized for speed and network stability enable real-time on-device or unstable connection processing. The lack of these models halts scalability efforts especially when using systems for outdoor observations or during offline lookups on mobile devices. Organizations should work to achieve equal performance and power efficiency in their model development process.

Chapter 4

PROPOSED MOTHODOLOGY

Five members of our team created separate conversational chatbots that integrate image recognition with language-based interfaces during this project development. All models share the common objective of allowing intelligent communication yet they utilize different methods stemming from their architectural designs and training approaches and their chosen implementation tools.

The main purpose of each model remains to allow smart visual-natural language interactions yet they use separate ways to build and train their systems. The models use different methods to read visual inputs before generating natural language responses with large language systems.

We organized our work in regular blocks which made each team member responsible for designing building and merging their unique computer model. Each vision model undergoes training and integration steps arranged into two architectural parts for our project. We explain each model creation method in detail while showing essential parts from their development process with supporting diagrams.

AeroAsk-1 :

1. Database Collection and Preparation:

1.1. Dataset Source and Aircraft Class Variety

The project used Military Aircraft Detection Dataset which originated from Kaggle as its main database. The database features 81 specific military aircraft classes which include fighter jets and bombers and motorized transport aircraft and aerial refueling vehicles and reconnaissance UAVs. The model gains knowledge about broad visual attributes because this database includes present-day aviation systems and past aircraft together with their operational uses and varying technological advancements.

1.2. Image Quality, Cropping, and Manual Verification

Every database entry includes high-resolution cropped pictures which separate aircraft from interfering background elements. The dataset received improved integrity through a labor-intensive process which resolved labeling mistakes and removed uncluttered or unclear or blurry images. The researchers cross-checked all images with aviation catalogs as well as defense archival sources to confirm their classification. The classification required detailed examination of minimal design variations such as sweep angles and intake structures and radar dome aspects which are essential for aircraft identification.

1.3. Structured Organization and Hierarchical Labelling

Each aircraft class within the complete dataset received its own uniquely named directory during a structured hierarchical organization. Path traversal algorithms extracted labels automatically during training by leveraging the structured organization which reduced the need for human encoding of labels. Data accessibility was standardized through this structure both during training period batches and testing pipelines.

1.4. Strategic Dataset Splitting

The dataset distribution followed a controlled approach that allocated eighty percent for training purposes after separating ten percent for validation testing and establishing another ten percent for testing. The stratified sampling method maintained proportional distribution of aircraft classes in every subset. The process of strategic dataset splitting aimed at avoiding excessive sampling of popular aircraft types (such as F-16, C-130) but it guaranteed there would be sufficient representation of less common aircraft classes (recognizance UAVs).

2. Image Preprocessing

2.1. Image Resizing and Standardization

The input images received uniform 224×224 pixel transformation through aspect ratio preservation whenever feasible. The input requirements for EfficientNetB3 and MobileNetV2 and VGG16 CNN models resulted in an obligatory resizing process because image distortion needed to stay below acceptable limits to preserve the identification skills for small features such as winglets, vertical stabilizers and landing gears.

2.2. Normalization and Dynamic Range Adjustments

Images were normalized to the [0, 1] range through pixel value preprocessing offered by TensorFlow/Keras utilities after the resizing stage. The procedure standardized the values to cover all images and thus it reduced training instability and allowed better convergence. Tests with contrast-limited adaptive histogram equalization (CLAHE) specifically applied to certain samples were conducted to enhance visibility of subtle features when conditions had low contrast.

2.3. Advanced Data Augmentation Techniques

An intense data augmentation implementation was used during the training phase for simulating realistic distortions from both satellite imagery and combat scenarios. The training data received three simulated distortions: camera mirror effects through horizontal flipping while also being exposed to ±20 degrees of rotational jitter and operators applying brightness/contrast adjustments within operational limits to capture different conditions. The model stability was maintained through experimental advanced augmentation techniques that included random zooming along with Gaussian noise injection.

2.4. Class Distribution Maintenance Post-Augmentation

A proper rebalancing process followed augmentation to preserve equal class distribution within all training, validation, and test partitions. Weighted sampling and other outlier handling techniques became part of mini-batch formation to balance remaining classes because certain aircraft types appeared too infrequently.

3. Training Through Pre-Trained CNN Architectures

3.1. CNN Models Selection and Comparative Benchmarking

Various pre-trained CNN models underwent benchmarking in order to find the best model for aircraft recognition. The researchers selected EfficientNetB3 as their main model because it showed the ideal balance between accuracy and computational efficiency. Testing of MobileNetV2 took place to determine its suitability for minimum resource requirements within edge devices. The research focused on ResNet50 for deep feature learning using skip connections while VGG16 functioned as a baseline although it had significantly high parameter density. The performance evaluation of each model yielded results about its accuracy together with its inference time and GPU/CPU memory use statistics.

3.2. The implementation uses Transfer Learning and Feature Reuse Strategy.

The entire set of selected models used transfer learning by maintaining ImageNet pre-trained weights across their convolutional base sections. The strategy allowed fundamental visual features to operate in an efficient manner while adjusting upper stages to adapt to aviation-specific patterns which reduced total retraining requirements.

3.3. Custom Classifier Head Redesign

A custom architectural design substituted the original classifier heads through the combination of Global Average Pooling followed by Dense fully connected layers with ReLU activations and ending with a Softmax output layer that generated 81 probabilities for each aircraft class. A dropout layer with 0.3 rate appeared in the model to stop overfitting when conducting fine-tuning.

3.4. Progressive Training Pipeline

Static application of convolutional backbones lasted for 50 training epochs while the attached classifier layers were being trained in the freezing phase. After stabilization occurred the top 30% convolutional layers progressively became trainable through a decreasing learning rate of (1e-5) to extract specialized aviation domain features.

3.5. Optimization Strategy and Loss Management

Using Adam optimizer the training reached its stable point due to its adaptive learning rate capability. The Categorical Cross-Entropy loss function function served as the objective metric because it suits problems with multiple classes. Weight decay set to 1e-4 enabled the prevention of overfitting in models when their capacity was high.

3.6. Final Model Export and Performance Metrics

The EfficientNetB3 model demonstrated an above 94% validation Top-1 accuracy when applied to various operational test samples. The model was converted into the .h5 file format which enabled simplified run-time loading before its integration with the backend conversational system.

3. The backend system combines the CNN Model and LLM while integrating them for the operational system.

4.1. CNN Inference System Setup

The Flask backend received the trained EfficientNetB3 model for its deployment which allowed users to obtain real-time inference results from uploaded images. The operating system delivered the predicted aircraft class as its Top-1 selection and provided confidence scores and intermediate features for system recording purposes.

4.2. LLM Deployment and Model Hosting

18 Local deployment of LLaMA 2-7B Chat model (GGUF Q4_K_M quantization) using llama.cpp runtime brought an end-to-end conversational AI capability which eliminated cloud dependency requirements. The system provided optional DeepSeek R1 and Gemma model hosting capabilities through API calls for environments demanding faster or context-based responses.

4.3. Prompt Engineering for Intelligent Response Formation

When aircraft identification occurred a dynamic prompt engineering module generated technical query templates which merged aircraft prediction data with system-developed conversational commands. Example prompt:

"The 'F-22 Raptor' requires explanation through its country origin and aircraft identity with main functions while highlighting its top speed and specific high-tech features."

4.4. Low-Resource Inference Optimization

Special system preferences enabled 4GB VRAM infrastructure through n_gpu_layers=0 settings which directed the LLM computations to work on the CPU to preserve low response times. The model memory size decreased through 4-bit integer quantization which led to effective local processing.

5. Prediction and Question Answering via QA Database Connection

5.1. Aircraft Class Prediction and Metadata Retrieval

After classification the system accessed real-time metadata through a QA database which contained organized information about propulsion and avionics configurations together with radar technologies and speed stats and operational history data.

5.2. Contextual Prompt Construction

Contextual enrichment took place before the LLM received input queries. Operational factual aircraft data from the database entered into the user questions served to ground LLM processing because it avoided pure free-text inputs.

5.3. Example of Augmented Prompting

The system generated the prompt dynamically when the user asked about which radar type the F-22 aircraft employed.

The AN/APG-77 AESA radar serves as the primary radar system for F-22 Raptor. Outline the operational frequency spectrum of this radar system together with its stealth attributes along with its target acquisition benefits.

5.4. Hallucination Mitigation and Response Validation

Direct inclusion of established technical information inside the prompts led to substantial reductions in hallucinations produced by the LLM. A proper assessment followed the responses to guarantee they matched real-world specifications of aircraft devices.

5.5. Real-Time Pipeline Execution

The entire system which includes image classification and QA retrieval and prompt creation followed by LLM answer generation ran to produce responses below 5–10 seconds throughout total execution time.

AeroAsk-2 :

1. Data Collection

1.1 Military Aircraft Dataset Compilation

We developed a purpose-built dataset containing 81 separate military aircraft classes which include fighter jets bombers along with transport vehicles. The classes include diverse images which were taken from different viewing angles while using different lighting setups and environmental backgrounds to achieve better diversity. The images came directly from approved defense repositories together with aviation archives for their authenticity. Efforts were made to keep the dataset evenly balanced through sufficient sample collection for every class. The feature learning process received emphasis on recording small aircraft attributes such as wing geometry and tail structures and engine classification information. The systematic collection method strengthens both generalization and reliability performance.

1.2 Data Validation and Integrity Checking

A set of rigorous validation processes was put into action to establish high-quality learning from the dataset. The evaluation included automatic procedures alongside human checks which screened out damaged or substandard images that carried wrong labels. Official military aircraft catalogs enabled researchers to verify the class classifications of each image. The dataset excluded images with partial occlusions together with missing parts which also included ambiguous stability identification. The model’s decision architecture received careful adjustment to prevent any particular category from taking control over the available space. The evaluation of statistics related to mean image count per class occurred regularly as part of inspection protocol. The training of EfficientNetB3 used this method to produce a reliable dataset for input.

1.3 Preprocessing and Augmentation

The captured images received a 300x300 pixels transformation to fulfill EfficientNetB3's input specifications. All pixel values in the dataset were normalized to achieve standardization. Data variability was expanded through different enhancement methods which included random rotations, zooms, horizontal flips along with minor distortions. During inference the model required augmentation methods that would enhance its ability to handle genuine image modifications encountered in operational settings. The training conduct required optimized preprocessing pipelines to decrease the computational requirements executed during training time. The augmented datasets received memory-efficient storage formats to enable quick batch processing. The preprocessing process established necessary conditions for building strong visual recognition capabilities.

2. System Pipeline

2.1 Image Feature Extraction and Classification

The input images run through EfficientNetB3 model that received pretrained training using the military aircraft dataset. The CNN produces deep feature embeddings which abstract complex visual characteristics specific for distinguishing aircraft types. Next the fully connected classification layers accept the feature vectors to generate softmax probabilities spanning the 81 aircraft types. After identifying the class the system presents the top-1 aircraft identification which serves subsequent processing tasks. The system included threshold parameters for model confidence to exclude unreliable predictions. The system directs conversational processing only toward outputs with confident probability results. The system provides both dependable aircraft identification through the pipeline which serves as an essential first step before starting the dialogue process.

2.2 Keyword Extraction for Context Building

The system fetches important keywords from its backend database after identifying the aircraft through classification process. Specifications together with mission profiles manufacturing history and special features make up the keywords used in the system. These context keywords play an essential role in developing the inputs that enter the Gemma 2B model. The natural language templates automatically receive population with extracted keywords to create deep prompts. The mechanism ensures proper background information structures are delivered to the LLM before it produces outputs. The quality along with depth and relevance of user answers depend significantly on how well the system retrieves context information.

2.3 Conversational Query Handling

The system enters conversational mode after context establishment so that it can communicate with the user. The system collects user questions and merges them with existing keywords for producing detailed prompts that LLM uses to generate responses. The Gemma 2B model receives enriched prompts through its processing method which produces coherent responses with personalization while providing information. The system keeps short records of conversational history to support context-aware development of multi-turn conversations when needed. The system provides intelligent fallback responses when the user asks questions that are unrelated to the detected aircraft. The tightly unified system provides an uninterrupted handover path from classification tasks to intelligent user communication.

3. Training Process

3.1 EfficientNetB3 Fine-Tuning

The EfficientNetB3 model received its initialization from pretrained weights in ImageNet to benefit from prior knowledge acquisition. Specific custom fully connected layers were installed at the model's output to perform 81-class classification of military aircraft. Our approach followed step-wise fine-tuning through which the convolutional base components remained frozen at first before training just the upper layers. Deep layer unfreezing was employed as the process deepened feature learning while preventing catastrophic forgetting of previous information. Training remained stable through the use of a learning rate schedule which started low and then decays gradually. Regular validation checks happened at periodic epochs to measure generalization and stop overfitting. The systematic fine-tuning method allowed EfficientNetB3 to develop appropriate capabilities for our domain-specific task.

3.2 Training Strategy and Hyperparameters

During the training session Adam optimizer managed the operations with an initial learning rate value at 1e-4. The training adopted early stopping which relied on validation accuracy data as a measure to stop training before overfitting happened and to minimize computational requirements. A batch size of 32 served simultaneously to preserve both memory efficiency and stable gradient training. The implementation of label smoothing in the loss function helped the model achieve better generalization among aircraft classes which were close to each other. The training process included learning rate reduction when model performance started to plateau in order to achieve optimal results. Data augmentation maintained its regular use in every training batch cycle. The model retained its saved checkpoint at the point when it delivered its highest validation accuracy.

3.3 Evaluation Metrics

38 The evaluation metrics consisted of top-1 accuracy and top-5 accuracy owing to their importance in multi-class classification tasks. Each misclassification of aircraft class was depicted visually through confusion matrix representations. The evaluation of each class used Precision together with Recall and F1-Score to provide balanced assessment for minority class instances. The assessment of class performance allowed detection of systematic errors resulting from jet

aircraft classification confusions. The decision regions of the model were visualized through ROC curve creations. A large number of rigorous assessments demonstrated that the trained model met practical reliability requirements as well as system robustness needs. The evaluation process for system final approval demanded performance benchmarking activities.

4. Backend System

4.1 Database Structure and Management

A SQLite database organized in a structured format serves to store metadata together with Q&A material for all 81 aircraft classes. Every record includes important aircraft information based on nickname alongside primary roles as well as specifications and historical notes. FAST retrieval of keywords occurred during real-time dialogue because indexing systems were implemented. The system included data integrity constraints to stop two kinds of database errors - field omission and duplicate information entry. System backups operated regularly to keep the database consistent in its state during the development period. The database connects the classifier results to the LLM system's context-based comprehension.

4.2 API Development for Model Serving

The developed RESTful API uses Flask as its base to serve both the EfficientNetB3 classification model and retrieve information from the database. The implemented API endpoints ensure smooth information exchange between front and back components. File uploads are processed by the classifier which returns the identified class ID. The system operates keyword extraction through an independent endpoint system alongside the request procedure. The system received protection through security controls which included both request validation and rate limiting features. The systematic API structure designed in this way allowed the system to scale while ensuring sustainable maintenance and readiness for potential new domain applications.

4.3 Integration with LLM Framework

The HuggingFace API becomes accessible through a post-classification backend module that operates Gemma 2B and LLaMA 2 model hosting services. The system converts extracted words along with user-initiated queries into prompt format before transferring them to the LLM endpoint. The system first structures the responses that later get returned to the frontend interface. Timeouts with handlers together with fallback mechanisms prevent broken processes when LLM systems fail to respond. To preserve system efficiency both token management and query rate control were properly implemented. The unified system allows for a continuous and efficient transition between vision and AI dialog features.

5. Frontend Interface

5.1 User Dashboard Design

Engineering developed a simplified web dashboard which optimizes user interface experience. The primary page contains an uncomplicated section that enables users to add aircraft images. The system shows the predicted aircraft name after classification and it provides a chat feature for further communication. Users benefit from improved experience through features such as loading indicators combined with result highlights. The interface organization included easy access to necessary controls for users. A design method known as responsive design maintained system functionality when users accessed the program through different devices. All users who lack experience can quickly adapt to this interface because of its straightforward design.

5.2 Image Upload and Display Module

Users can submit PNG JPEG and BMP file types for image upload through a system with integrated client-side check functions. The system immediately shows previewed images through the user interface for users to check their choices. The backend system shows a loading indicator when processing the classification request made by the user. After classification the system presents the aircraft identification with model confidence level together with an associated image thumbnail. When the user detection finishes the Start Conversation button activates to let them input their questions about the aircraft. This system promotes an uninterruptible pathway linking image detection and written communication systems. This upload module establishes complete API integration to support powerful backend operation.

5.3 Conversation Chatbot Panel

Users could access a chat feature directly from the interface to pose questions which concerned the aircraft they had identified. Both user messages and AI answer responses appear in a time-rank order through the chatbot interface. The system applied special formatting to technical specifications and important points that appeared in the AI-generated responses. The system enables multiple-turn conversations to occur throughout one active session. Someone asking queries outside the aircraft context will receive a polite system request to stay within the current aircraft information. The performance management plus user experience optimization takes place dynamically during chat sessions. Users can use the conversational panel to move between general recognition and thorough examination of information.

6.Question Answering Strategy:

6.1. Contextual Prompt Engineering

The LLM receives queries through dynamically generated structured prompts made from extracted keywords. The standardized prompts use a technical framework for accuracy with historical data and operational specifications. The method guarantees the LLM produces responses that fit perfectly with the context instead of providing nonexistent and general answers. The system included supplementary constraints within the prompts to prevent hallucinations as well as preserve factual accuracy. The scalability of the system remains possible due to template-based prompts as the system receives new aircraft. The careful work on prompt preparation delivers substantial improvements to both relevant dialog flow and system credibility.

6.2 Multi-turn Conversation Management

The introduction of context persistence processes allows for steady conversation progression between different survey questions. The system stores essential user keywords together with their previous search terms in temporary storage throughout the chat session. When LLMs receive new prompts they need both the modern question and shortened historical context. The model sustains coherent logical flow while keeping its discussions focused by using this method. The system implemented a maximum context depth limitation to control excessive historical information. When context expires the intelligent reset feature initiates a smooth transition of the session. User satisfaction rises because the multi-turn management feature delivers human-styled continuous dialogue to chatbots.

6.3 Handling Uncertainty and Fallbacks

Some requests which exceed the dataset scope or exceed the training data range of the LLM remain out of its processing capacity. The system implements backup procedures for unrecognizable queries. Confidence values in LLM answers serve as indicators to identify doubtful responses. The system triggers predefined backup templates that steer users into established aircraft-related areas whenever detected uncertainty exceeds an established threshold. The option to learn generic aviation knowledge exists without the use of hallucinated information. System credibility along with user trust is preserved through fallback mechanisms which protects the system from distributing inaccurate information.

AeroAsk-3 :

1. Data Collection and Preparation:

1.1. Aircraft Class Compilation

Scientists constructed a vast worldwide dataset which contained 81 different aircraft types within it. The dataset presents military aircraft beginning from different historical periods across various technological generations starting from fighters up to bombers and transport aircraft and refueling planes and UAVs. Multiple global aviation platforms are well represented by the model because of its wide range of aircraft types from different aviation sectors.

1.2. Image Folder Structuring and Standardization

Every aircraft class occupied individual directories inside a clean standard crop/ directory structure in the dataset organization method.

An automated system could extract labels because the dataset adopted this hierarchical organization which also simplified the training process. A dedicated team specialized in editing the pictures to show just one aircraft by eliminating all non-aircraft elements.

1.3. Creation of Structured Knowledge Base (fighter_qa.csv)

The fighter_qa.csv structure was established as a dedicated question and answer knowledge base that included authoritative metadata about fighter planes such as type, manufacturer, unit cost, top speed, operational range and service history. The first stage of the question-answering mechanism bases its information on the fighter_qa.csv database which provides exact information about aircraft types.

2. System Pipeline:

2.1. Aircraft Detection via YOLOv5

The detection phase utilizes YOLOv5 model for detecting objects in aerial and satellite-style images with both efficiency and precision. This system detects aircraft-containing bounding boxes in real-time through a process designed to find the right balance between timeliness and accuracy.

2.2. Cropped Region Extraction

The system selects the bounding box having the maximum score from the multitude of detected aircraft from the camera surveillance. The system extracts the newly created cropped image file for classification purposes. The input quality remains intact through this step by removing low-confidence along with partial detection alerts.

2.3. Classification using MobileNetV3-Small

The detection system sends the aircraft section contained image to MobileNetV3-Small because of its ability to run real-time and possess compact architecture. The model predicts one of the 81 aircraft classes as its output which makes it appropriate for employment in limited resource settings.

2.4. Integrated Response Generation

Following the classification step the model adds the detected aircraft label before it merges with the user input. The system initiates a keyword search in its CSV file database to locate a suitable response. In case the search fails to find any matches the system switches to Flan-T5-small for creating a natural language response through few-shot prompting alongside structured input formats.

3. Training Process:

3.1. Model Configuration and Epoch Schedule

29 The training phase proceeded at a batch size of 32 and learning rate of 5 × 10-4 during 20 episodes. The images received a 160×160 pixels resolution for efficient processing of aircraft detection while preserving necessary identification features.

3.2. Mixed Precision with CUDA Support

The system gained speed in training through the activation of Tensor Core operations which are available in specific GPUs. The implementation of mixed precision training accelerated the training process and reduced memory requirements allowing use of mid-range GPUs for training speedily.

3.3. The system used Learning Rate Scheduling and Early Stopping techniques during training.

The scheduler based its learning rate adjustments on changes observed in validation loss. Training interruptions happened through an early stopping system to check overfitting risks while improving both model generalization and preventing computational wastage.

4. Backend System:

4.1. REST API Deployment using Flask

The server implemented using Flask deployed one REST endpoint to manage the complete workflow beginning with image upload and validation before continuing to classification and answer creation through HTTP requests.

4.2. Input Validation and Preprocessing Steps

The server operates a complete input evaluation process which examines both file type and size compatibility. The inference engine processes tinted and normalized files that were validated for the purpose of achieving uniform input quality standards.

4.3. Result Caching through Hashing

A performance optimization method for the system involves using SHA256 hash values of image+query combinations as a caching mechanism for previous results. When users make repeated requests that provide equivalent inputs the system retrieves the cached results from memory which improves performance efficiency.

4.4. Hybrid Answering Flow Control

The answer generation procedure begins by conducting immediate and precise results identification through CSV-based matching. Flan-T5-small receives prompt templates after the CSV fails to return results. Input sanitation procedures in the system detect and prevent system failure which occurs because of invalid or unhelpful query inputs.

5. Frontend Interface:

5.1. Real-Time User Interface (HTML5/CSS3)

A web-based GUI featuring HTML5 and CSS3 interface presents itself in the system for intuitive user interaction. Users can update images and watch monitoring progress while receiving automated results that appear without user action to refresh.

5.2. Form Validation and User Notifications

Users can navigate frontend client-side validation to find out which image file types the system supports for upload. The application provides specific feedback about upload failures together with incomplete field requirements which enhances user experience through reduced mistakes.

5.3. Responsive and Mobile-Friendly Design

The layout system adjusts properly for smartphones and tablet user experiences. Users who query the system with mobile technology need these features when they operate in remote areas with limited infrastructure.

6. Question-Answering Strategy:

6.1. The initial process of the system includes jet detection followed by classification.

The system starts its operation through identifying aircraft in a submitted image by performing object detection and classification. The system uses this result to create the answer-generation foundation.

6.2. Stage 2 – Question Handling Flow

The retrieval process becomes active after the system combines aircraft information with the user query.

6.3. CSV-Based Lookup for Direct Answers

The system retrieves answers from fighter_qa.csv instantly when the provided query equals or nearly matches an existing file entry. Fast response speed along with accurate data output characterizes this stage.

6.4. LLM-Based Answer Generation via Flan-T5

The model uses beam search decoding and early stopping together with no-repeat n-gram to produce an answer when no matching CSV entries exist.

6.5. Fallback Response and Data Limitation Messaging

The system delivers a fallback message showing the identified aircraft name while admitting data incompleteness during such situations. The system provides a pleasant interface experience through this method even during unusual situations.

AeroAsk-4 :

1. Data Collection

1.1 Compilation of Military Aircraft Dataset

A large data set of 86 different classes of military planes were established, including fighter planes, bombers, helicopters, transport planes. The dataset was gathered from public repositories and aircraft recognition data bases as well as defense imagery archives. It consisted of various views of the same airplanes in different lighting conditions, angles, backgrounds, and situations. An additional effort was put to acquire a balanced number of samples for each class so as to rule out the complication of bias in training. The fine-grained factors that were targeted to be learned included the shape of the wings, the placement and style of the engines, the shape of the nose, and rules of the tail design. The development of this dataset was biased to making the model more robust, more generalizable.

1.2 Data Validation and Integrity Checking

The data set was analyzed by both automatic and manual verification processes. Examples of images with incorrect labels, strong occlusions, blur, or missing structures were removed. Cross-validation was done through official aircraft databases and public defense catalogues. Statistical processes were used to ensure class balance and to ensure that popular aircraft are not too dominant. Underrepresented classes were substituted and or added to obtain a required minimum percentage. The dataset that was filtered was used as a foundation for successful modeling, as well as more natural language interactions.

1.3 Preprocessing and Augmentation

Pictures were resized into 224x224 pixels in order to be VGG16 architecture compatible. Normalization steps were performed on the mean and standard deviation of ImageNet. As for Generalization, random rotation, horizontal flipping, color changes, and zooming effects were implemented. These augmentations simulated distortions that happen in the case of the real-world usage and allowed inferring on images employed in operations. The preprocessed dataset was effectively managed through batches (data generators) in the training process of the model.

2. System Pipeline

2.1 Image Feature Extraction and Classification

The classifying pipeline used a VGG16 fine-tuned on the pre-processed 86-class set of military aircraft. The CNN created deep feature vectors and passed it to custom fully connected layers with a softmax output. The system produced the top-1 class and confidence threshold for rejecting uncertain predictions. Classification output was shown the user with aircraft image, class name, and confidence score. When the confidence of predictions exceeded a specified threshold, it called the next stage. language-based question answering.

2.2 Keyword Extraction for Context Building

Once classified, a back end database was invoked to produce key words relating to the classified aircraft – i.e., its role, custom, time of entry, maximum speed, and operating history. These keywords were used to build a prompt of context that was dynamically fed into the LLM. This meant that the language model could be served with inputs that were formatted in a neat manner and fact-packed before it could generate a response.

2.3 Handling Conversational Queries

Once identified, users would be in a position to request a chatbot more questions about the identified aircraft. Questions and phrase relevant to the context were passed to the locally run Mistral model through Ollama. The system was able to create powerful prompts which were highly technical in specifics but conversational in tone. The chatbot supported single-turn and multi-turn dialogue and fallback responses when the questions pitched out of the subject. This smooth transition from visuoving identification to linguistic conversation started off an exciting educational experience for users.

3. Training Procedure

3.1 VGG16 Fine-Tuning

The VGG16 model had been pre-trained with ImageNet weights. Its convolutional base was frozen at the first training and a new classification head was trained on the aircraft dataset. After encouraging outcomes of the model, successive layers were progressively unfrozen and fine-tuned by allowing a lowered learning rate to prevent catastrophic forgetting. This two-stage training ensured the model preserved general features of vision and learned the task related to the domain.

3.2 Training Methodology and Hyperparameter Tuning

Adam optimizer was employed to train the model on the learning rate of 1e-4. Early stopping was implemented in reference to evaluation of validation loss, and learning rate schedule was used to improve convergence. The batch size used was 32 for steady training and labeling smoothing was carried out to prevent over confidence on very similar types of airplanes. A checkpoint of the best model was saved as it was when the validation accuracy is the highest.

3.3 Assessment Criteria

20 Top-1 and Top-5 accuracy were tested by the model. Confusion matrices provided class specific performance and is particularly useful for classes of aircraft similar to each other. 34 The precision, recall and F1-scores were calculated for all the classes to discover the performance of minority class. ROC curves were used to analyze thresholds. The testing assured that the model was not only accurate, but also were deployable for real life usage.

4. Backend Architecture

4.1 JSON-Based Metadata Management

Instead of an ordinary database file, a structured JSON file is employed for managing metadata for the entire 86 aircraft classes. This JSON file contains important details like the name of the aircraft and whether it is of the type, how it is specified as well as the historical details. The file serves as the storehouse of truth for giving relevant context to the Mistral model while answering questions. It allows for quick access to class-specific information without additional costs of database searches, which makes it light and editable during development.

4.2 Backend Workflow and API Integration

The core functionalities like receiving uploaded images, classification of aircrafts, and the initiation of language based interactions is done by using the backend logic that makes use of Flask. Following classification, the detected aircraft class is then used in the JSON file to get correlated details. This contextual data is then sent to the locally hosted Mistral model using an API call with the user's query. The backend features the coordination functionality to easily link between the image model, the JSON metadata, and the language model to provide the most precise and efficient responses possible.

4.3 Ollama Integration (LLM Framework)

Instead of using cloud models, the used Ollama to execute local large models of language like Mistral or LLaMA 2. Local API calls were made for generating responses. The prompt pipeline mixed user queries with keyword context to then send the formatted prompt to the Ollama endpoint for their LLM. Timeout management, fallback Templates, and token control mechanisms helped to support the system to remain responsive despite it being loaded very heavily with queries of the system. Local deployment had enhanced response time, privacy, and economy.

5. Frontend Interface

5.1 User Dashboard Design

The web interface was user friendly and simplistic. Users were empowered to upload aircraft images, view predictions, and view chat panel. The interfaces functioned perfectly well in desktop and mobile devices because of responsive designs. Such visual feedback like progress bars, prediction confidence bars and thumbnail previews contributed to the pleasantness of the user experience.

5.2 Image Upload and Display Module

Images could be uploaded by users in JPEG, PNG, and BMP configurations. Images were pre-viewed in real time and uploaded to backend to be categorized. On classification; Aircraft name, thumbnail and Confidence score were displayed. When a daring prediction was given, users would be able to start to interact with the help of the chatbot to ask questions.

5.3 Conversation Chatbot Panel

Through the chatbot, users were able to ask questions related to the identified aircraft. All messages were delivered in a chronological order. Salient facts in AI responses were made bold or bold with color for accentuation. The chat was multi-turn and dealt with off-topic questions in a polite manner. Discussions stayed put on the context of the identified air to avoid hallucinations or misinformation.

6. Question Answering Strategy

6.1 Contextual Prompt Engineering

Prompts were generated in a dynamic way by combining extracted keywords for aircraft and user queries. The formal format was used to ground the LLM on factual, technical information. Templates were used to make it easier to scale with the introduction of newer aircraft. Anti-hallucination strategies of prompt constraints and grounded facts were utilized to maintain factual integrity.

6.2 Multi-turn Conversation Management

The session memory stored a concise history of past exchanges, allowing the chatbot to help sustain freewheeling multi-turn discussion. When prior context exceeded token caps, it was cut off and smart resets were made on demand. This strategy provided a seamless conversation experience, and did not flood the LLM locally.

6.3 Managing Uncertainty and Contingencies

Where LLM answers were less confident, be it because of heuristic scoring or context conflict, fallback templates were triggered. These templates encouraged the users to enter a query on aviation or general data. This ensured that the chatbot never lost its reliability, precision, and utility even when the user’s queries moved away from what was expected from it.

AeroAsk-5 :

1. Data Collection:

1.1 Aircraft Categories and Classes

This dataset endeavors to cover an extensive variety of information about present-day military aviation. Different aircraft received grouping into distinct categories of fighter jets along with strategic bombers and military transport aircraft and reconnaissance UAVs. The F-22 Raptor and Su-57 Felon served among the aircraft models which were part of distinct categories for the classification process.

1.2 Image Acquisition and Verification

The images came from trustworthy sources which delivered sharp-resolution pictures. The vetting process was carried out manually to remove images which contained mislabels or blurriness or duplications. Quality control procedures played an integral role to defend against database corruption and errors that could affect the training phase.

1.3 Dataset Structuring and Labeling

The organization structure divided images into directories by aircraft class type. Mathematical systems and popular deep learning frameworks recognize this folder-based arrangement because it makes it simpler to stream data automatically.

1.4 Data Splitting Strategy

The researchers distributed their aviation data into training sections that composed 80% and validation and testing segments comprising 10% each. The investigators distributed aircraft categories evenly across the training validation and testing groups to maintain fair exposure during learning process and assessment tasks.

2. System Pipeline:

2.1 Image Preprocessing

Each image received a preprocessing step before training which involved resizing it to 224×224 pixels as a requirement for EfficientNet and related CNN systems. Image pixels received normalization treatment to fit a dynamic range between 0 and 1 which helps training run faster while maintaining stable model convergence.

2.2 Data Augmentation

Multiple augmentation techniques were implemented to boost data variability and stop the model from becoming overly specialized. Random horizontal flipping when combined with rotational shifts up to ±20 degrees and restricted brightness adjustments formed important data augmentation techniques. Transformation methodologies limited their application to training data to produce a Generalizable model for processing fresh data.

2.3 Model Architecture Selection

Development testing involved evaluating distinct CNN architectural solutions. During development EfficientNetB3 distinguished itself as the optimal architecture because it maintained high accuracy alongside quick throughput. A benchmark evaluation included the implementation of three further networks beyond the chosen MobileNetV2 and ResNet50 and VGG16.

2.4 Transfer Learning and Feature Adaptation

The model took advantage of generalized visual feature extraction through the implementation of ImageNet weights during transfer learning. The standard convolutional layers from the base remained untouched for retention yet a custom classifier head was implemented following them with Global Average Pooling alongside multiple Dense layers and termination by a Softmax layer designed for multi-class predictions.

3. Training Process:

3.1 Two-Phase Training Strategy

The training process included two separated phases. During the initial stage the convolutional base received its training freeze while newly added classification layers became trainable. The network established rapid aircraft adaptation through its ability to preserve ImageNet generic features while training for aircraft recognition. The training continued with the top 30% of convolutional layers unfrozen and subjected to further training with reduced learning rates. Fine-tuning enabled the model to recognize aircraft better when they appeared visually identical.

3.2 Optimization and Performance Metrics

The training process implemented Adam as the optimization method because this optimizer provides adaptive learning features that perform excellently in deep learning applications. The model required categorical cross-entropy as its loss function since it dealt with multi-class classification tasks. Top-1 Accuracy was the main metric for evaluation because it counted how often the model selected the correct label as its first choice.

3.3 Model Evaluation and Export

The model monitoring took place on the validation set throughout the training process. The analysis implemented early stopping as a prevention technique against model overfitting to promote generalization. The model reached its target accuracy level which resulted in the export of aeroclassify_model.h5 containing validation results above 90 percent.

4. Backend System:

4.1 Model Integration for Inference

The model integration process for the backend occurred via Keras and TensorFlow APIs. After users upload an image the backend system performs processing to run inference that determines the probable aircraft class while providing its prediction certainty.

4.2 Structured Q&A Database

The application utilizes aircraftQA.csv (CSV file) to store technical aircraft data alongside common questions. There exist multiple database attributes covering engine specifications together with radar systems and weapon payload information to provide full responses to every user query.

4.3 Real-Time Data Retrieval

After the aircraft class selection the backend system retrieves necessary technical information from the Q&A database. The system maintains accurate and consistent answers through this design which ensures responses are directly related to each classified aircraft type.

4.4 Query Routing and Response Handling

The system's backend services incoming user questions through keyword matching routing to search for matching questions in the existing Q&A database. The system provides the matching response when it detects a suitable question from the Q&A database. User intervention is required when the system fails to generate a suitable response because it requests a restatement of the question.

5. Frontend Interface:

5.1 Image Upload and Classification Display

Users can input aircraft images through the interface that the frontend design provides. After image processing the user interface shows clearly both the identified aircraft class together with its confidence score.

5.2 Information Panel for Aircraft Details

The interface shows both the classification result and displays comprehensive specifications of the recognized aircraft. The information display features aircraft graphics together with essential performance metrics as well as radarspecs and identification of missions.

5.3 Interactive Q&A Section

Users can ask questions about particular aircraft by writing normal spoken language through the interface. The frontend submits the user queries toward the backend to check them against selected Q&A material in the database. The system finds suitable responses from its datasets which it presents in a conversational manner.

5.4 User Feedback and Re-query Options

When users ask questions that have unclear wording the interface asks for clarification through user prompts to achieve a better understanding. The system uses a recurring process to guide users toward obtaining appropriate and correct information.

6. Question Answering Strategy:

6.1 Keyword-Based Matching System

The system depicts its question-answering capability through a word extraction process followed by a matching mechanism. User-submitted questions trigger a system to extract major words from text and compare these keywords to Q&A information that matches the selected aircraft type. The elimination of expanded search parameters leads to higher relevant results.

6.2 Confidence-Driven Response Selection

Each potential response within the matching engine obtains evaluation based on its similarity to the user query. A strong matching result between algorithm and data leads to the retrieval of the relevant answer. Users must rephrase their questions when the response fails to meet satisfaction criteria to retrieve only reliable answers the second time around.

6.3 Avoiding AI Hallucination

The system refrains from fabrication due to its dependency on a pre-validated database instead of generating content from generative models. Technical accuracy of answers becomes more dependable due to this system design which removes hallucination risks from the process.

6.4 System Efficiency

The entire process from image submission through classification to information retrieval and question-answering executes in less than three seconds on standard equipment. The system achieves swift response times because of its design making it appropriate for educational and operative situations.

Chapter 5

OBJECTIVES

Objective 1: Accurate Military Aircraft Identification

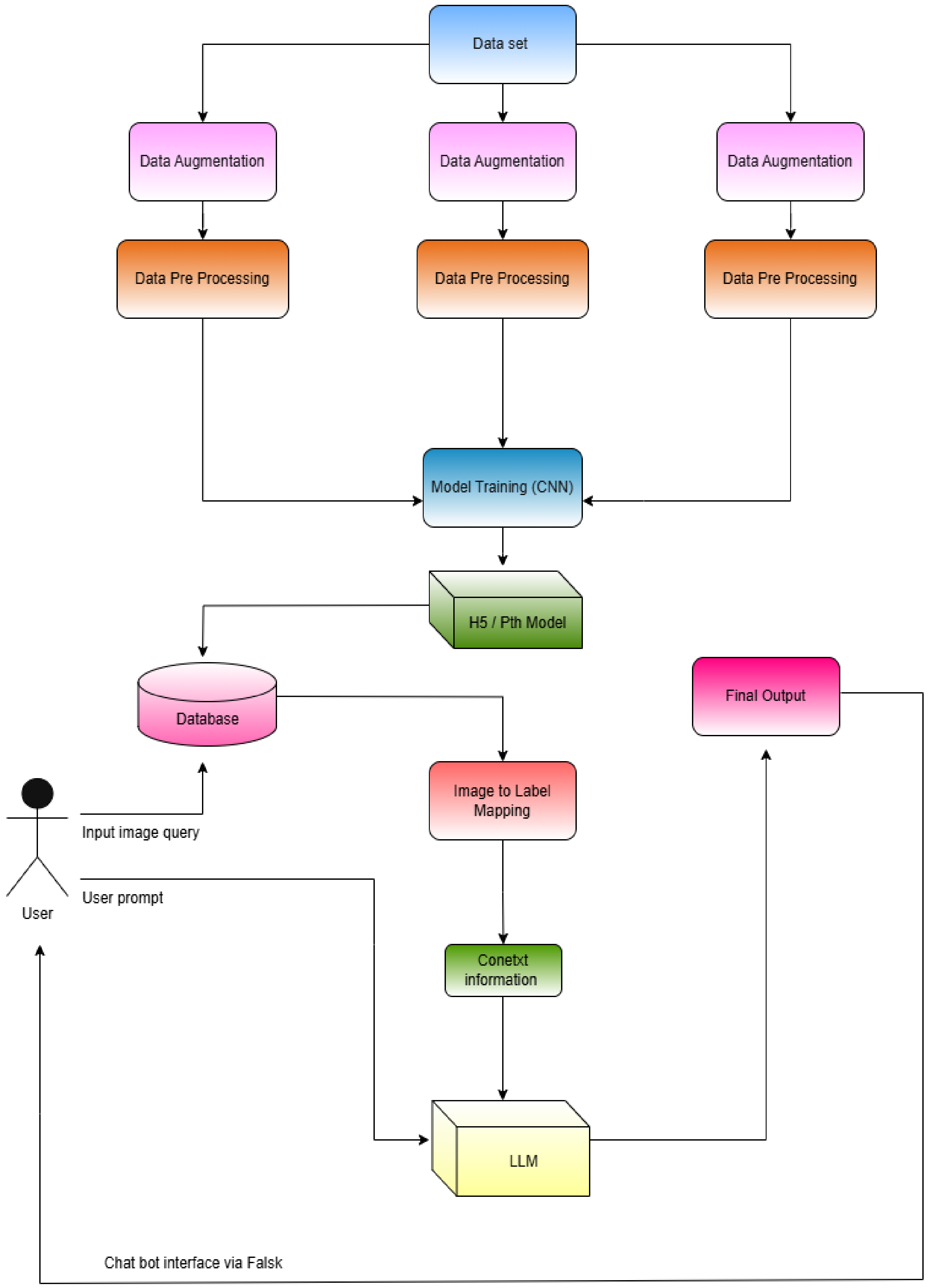
35 The primary aim of this project is to develop strong AI system for detecting military aircraft from visual input accurately. This process includes training convolutional neural networks such as EfficientNetB3, ResNet50, VGG16, and others on a selected dataset with 81 classes of military aircraft images. The objective is to allow the system to identify different aircraft depending on critical visual characteristics such as the shape of a wing, engine configuration, fuselage design, and tail type. This capability is very important for such a practical application like defense training, aviation studies, and enthusiasts, who want to identify military aircraft effectively. The system also strives to address demands for a fluctuating image quality, angles, and environments to provide real-world functionality.

Objective 2: Contextual Aircraft Info Q&A.

Besides achievements in identification, the system is also designed to enable delivery of contextual information about the identified airplane through the service of an interactive Q&A module. This component incorporates advanced LLMs such as Gemma 2B, LLAMA 2, and Mistral to produce relevant and contextually appropriate reactions to the queries made by the users. With the integration of each aircraft class with a structured Q&A database, such as the database used by aeronautical and engineering groups, the ability of the system to answer technical and historical questions on the identified airplane adds an educational value to the project. The double-faceting of this project provides both visual recognition and knowledge retrieval and evokes a more comprehensive and substantial effect on aviation enthusiasts and students.

Chapter 6

SYSTEM DESIGN & IMPLEMENTATION



44

6.1 Overall System Architecture:

6.1.1 Input-Processing-Output Flow

The web interface enables users to start the system flow by uploading aircraft images. The system transfers the image directly to the preprocessing module that conducts resizing and normalization along with pixel adjustment. After processing the image through a trained CNN model receives an aircraft classification among the predefined categories. A classification output allows the system to retrieve several Q&A pairs from a structured database. The database entries result in semantic keyword extraction that leads to engineered prompt creation. The LLM engine receives the prompt to produce an answer that appears human-generated. The system presents the response to the user after generating an interactive dialogue that depends on both intelligent recognition and retrieval capabilities.

6.1.2 Modular Interaction Strategy

The architectural design uses modular organization with defined boundaries between system components that let both parts perform their responsibilities effectively. Image preprocessing exists exclusively within the preprocessing module and the CNN classifier performs both feature extraction and prediction tasks independently. Once prediction completes the database module takes full control over Q&A retrieval operations without affecting other system components. All functions that involve keyword extraction together with prompt engineering and LLM interaction run as a distinct service within the back end. The controller function of Flask maintains the routing process and flow control for these modules. Each system section operates independently through modularity which allows updates and improvements of individual components without affecting pipeline performance and without needing a full system redesign.

6.1.3 Deployment Scalability and Extensibility

The system architecture enables simple scalability through both vertical and horizontal growth according to future processing requirements. The deployment of the complete system on AWS EC2 or ECS platforms becomes simpler through the Docker containerization of the CNN model serving module and LLM inference module. The database can independently scale up through its relocation to managed RDS or NoSQL services. Future system development can add new aircraft categories and update LLM versions and support multiple query types such as image with text through the current design framework. The designed pipeline demonstrates compatibility with sophisticated AI implementations that will be used in realistic situations.

6.2 Image Preprocessing and Augmentation:

6.2.1 Image Resizing and Normalization

Image upload standards require strict normalization because resolution and aspect ratio and color intensity levels differ widely between images. The system resizes each input into a 300x300 pixel box that satisfies the CNN model input requirements although it preserves the critical visual characteristics of aircraft. Following image resizing the computer normalizes pixel values between 0 and 1 through a 255 division process on each pixel intensity. Model gradient dominance during training is avoided because pixel value normalization has been performed. A slight Gaussian smoothing process is optional for robustness since it reduces random noise while preserving structural details. The stability of training along with fast convergence and effective generalization across new images during real-time operations requires consistent input pixel range and uniform input size.

6.2.2 Data Augmentation Techniques

Strong augmentation techniques must address aircraft images which appear under diverse capture conditions that include different angles and lighting settings and background elements. Random transformations consisting of horizontal and vertical flips show the model aircraft from various flight positions while random zooming lets it learn to identify aircraft from varying distances. A maximum rotation of 30 degrees allows the model to recognize images correctly despite airborne photographic image tilts that commonly occur. The model receives shifts in width and height to create varied positions because this prevents overfitting when aircraft appear in the center. A combination of brightness adjustment and contrast enhancement together with synthetic shadow application duplicate different day lighting conditions. Positively manipulated datasets create additional variations that produce dependable accuracy when the model operates in actual field conditions.

6.2.3 Preprocessing Pipeline Optimization

The training and inference operations benefit from TensorFlow'stf.data pipelines which provide optimized preprocessing. Images process in multiple threads which performs augmentation work before sending them to GPU memory to prevent delays that occur during batch loading. The storage of intermediate preprocessed images through caching decreases the computational burden during multiple training epoch cycles. The model training pipeline contains all preprocessing and augmentation logic so it can be reproduced in different environments which include local machines and cloud GPUs and mobile devices. The efficient preprocessing techniques shift the data bottleneck outside of preparation thus enabling the model to concentrate on learning during training.

6.3CNN-Based Feature Extraction and Aircraft Classification:

6.3.1 EfficientNetB3 and Compound Scaling

The compound scaling principle in EfficientNetB3 represents its distinguishing feature because it uses a single coefficient to adjust model width and depth alongside resolution. The network deepening process is supported by increased width and larger processed input images that keep a stable relationship. The model achieves better accuracy through harmonious scaling while using fewer parameters than typical CNNs operate with. The aircraft classification system requires EfficientNetB3 to identify cockpit structures along with wing shapes and tail designs at a high level of precision without increasing model dimensions. The network operates effectively on resource-limited environments to achieve high classification precision which makes it suitable for industrial-grade image recognition solutions.

6.3.2 ResNet50 and Skip Connections

The introduction of residual learning through skip connections became the defining feature of ResNet50 which transformed deep learning operations. The direct passage of gradients across several layers through these connections solves the traditional gradients disappearance issue that hindered deep network development. The ResNet50 network effectively retrieves detailed aircraft features such as weapon systems and rotor blade numbers and landing gear arrangements. Skip connections enable the model to learn complex features at different levels while maintaining information quality throughout the deep network. Through its modular design ResNet50 enables controlled layer-wise fine-tuning when transferring knowledge that enables better specialization toward aircraft images beyond general ImageNetpretraining classes.

6.3.3 VGG16 and Classical Feature Extraction

The image classification field considers VGG16 to be a top-tier model for extracting features despite its older architectural design. The VGG16 network applies a series of sequential convolution layers that use fixed 3x3 kernels which deepen and abstract the features. Due to its simple structure VGG16 can locate hierarchical spatial patterns that include fuselage shape and engine nozzle placement and tail wing configurations which make aircraft easily distinguishable. Less complex parameters in ResNet compared to transformers lead to easier training or fine-tuning operations on smaller datasets. VGG16 functions as a compact lightweight model which provides efficient prototyping features and baseline performance criteria for assessing newer CNNs in our project.

6.4. Q&A Database Construction and Management:

6.4.1 Database Structure and Organization

The Q&A database organization uses aircraft_id as the primary key to tag individual database entries. The database uses aircraft IDs to connect various question-answer pairs containing diverse user intents about basic specifications alongside historical significance and operational information. The designed database structure allows for fast Q&A data retrieval when using aircraft class predictions obtained through the CNN model. The retrieval speed gets improved through the implementation of inverted indexes and keyword-based tagging systems. Future system expansions such as multilingual implementation or metadata inclusion of aircraft era and origin and category classification (fighter jet, transport vehicle, drone) become possible because of this design structure.

6.4.2 Semantic Keyword Extraction Process

A user upload leads to classification completion which triggers the Q&A entry retrieval. The Q&A texts submit to semantic keyword extraction through processing methods which combine tokenization with named entity recognition (NER) and phrase chunking. The process starts with tokenization for breaking down text into significant units followed by NER systems that detect vital aircraft terminology which includes "afterburner" and "payload capacity" and "combat radius". Additional phrase detection methods identify multi-term entities which need to be processed as unified semantic units. A set of vital keywords which originate from the extraction process functions as the fundamental vocabulary to develop prompts that optimize LLM queries. The richer and more contextually relevant the keywords, the higher the quality of the conversational output.

6.4.3 Database Maintenance and Expansion

The Q&A system requires scheduled regular updates and maintenance to remain relevant throughout time. The modular design makes it possible to integrate new aircraft models along with updated specifications and historical data corrections without any issues. The system implements version tracking to monitor data modifications which enables users to reverse their actions when they enter flawed information. The database augmentation process through scraping pipelines operates from trusted aviation sources and official aircraft manuals for automatic growth. The process of manual curation guarantees both fact-based reliability and proper structural design making information optimal for later utilization. The automated system maintenance strategy improves user confidence in accuracy while making the bot future-ready.

6.5 Prompt Engineering for LLMs:

6.5.1 Raw Keywords to Engineered Prompts

The extracted database keywords and semantic concepts undergo systematic conversion into complete prompts which LLMs can process effectively. Prompt engineering methods make certain the language model receives a deep understanding of query contexts instead of receiving disordered keywords. The system maintains different types of query templates for factual, explanatory and comparative requests while it inserts selected keywords into these templates automatically. The factual template shows "A brief technical [aircraft_name] description should include [keyword_1] together with [keyword_2] and [keyword_3]." The engineered method boosts LLM performance by enabling it to deliver accurate detailed answers which avoid hallucinatory distortions.

6.5.2 Importance of Structured Prompts

The use of structured prompts represents a vital mechanism to enhance both reliability and output consistency of LLM systems. The sensitivity of language models to input wording makes brief or unorganized prompts generate either general or wrong outputs. The model becomes easier to control through structured instructions which also define the required information retrieval style. The specific instructions which state "list three major upgrades" or "summarize operational history in 100 words" result in much more accurate output compared to open-ended inquiries. By using structured instructions users achieve improved repeatability with various LLMs and models thus their system produces predictable outcomes and reduces user confusion while using the system repeatedly.

6.5.3 Specific Prompt Designs for Different LLMs

Every LLM has a distinct architecture so each model requires standalone prompt design following Gemma 2B, LLaMA 2, and DeepSeek. The abbreviated prompt format for Gemma 2B utilizes its efficient and light system design. The deep context capability of LLaMA 2 requires prompts which contain complete historical details and operational specifics. DeepSeek makes use of optimized retrieval-augmented generation (RAG) technology through prompts that incorporate knowledge obtained from auxiliary information retrieval systems. The models specialize to produce optimal outputs because of their individual capabilities. Through prompt engineering optimization the accuracy of the system improves and the user experience during conversation becomes better.

6.6LLM-Based Conversational Generation:

6.6.1 Behavior and Use of Gemma 2B

The designed LLM known as Gemma 2B functions with fast inference speeds at reduced computational costs to deliver high efficiency and light weight operation. Gemma 2B functions as the main response tool for real-time discussions where users need swift replies to obtain simple aircraft information or specifications. The small parameter count of Gemma makes it capable of responding in milliseconds during CPU-based deployments because of which it works well in low-latency systems. While providing shorter answers than bigger models, Gemma 2B makes up for this by delivering fast responses together with accurate information for simple queries so users avoid waiting time.

6.6.2 Behavior and Use of LLaMA 2

The combination of extensive model size and extensive training corpus enables LLaMA 2 to produce refined detailed factual and compliant responses. The system enables users to obtain advanced information which might involve aircraft operational evaluations or information regarding combat history or specifications lists. LLaMA

2 performs well across extensive dialogues while keeping the conversation logical which makes it perfect for academic research queries or expert professional research help. Natural language processing is better than smaller LLMs because the advanced capabilities of its system understand complex searches with multiple components and conditional conditions. Our system automatically redirects users to access LLaMA 2 when the prediction reaches specific response length and complexity requirements.

6.6.3 Behavior and Use of Mistral

Mistral stands out as an extremely optimized LLM (known for balanced performance in regard to both speed and accuracy, which makes this LLM of choice for aircraft-related Q&A systems. Unlike larger models, Mistral can handle medium to complex queries competently without losing much latency there by allowing real time interactions even under the resource limited environment. It is especially great for being able to share brief technical facts, such as the specification, flight capability, and historical data on today's military aircraft. Mistral architecture is designed so as to optimize dense information retrieval ensuring correct response to structured input such as an aircraft model's number, mission role, operation details, etc. It makes it a perfect solution for mid-tier queries where both speed in response and fact accuracy are important equally.

6.7 Flask-Based Frontend and Backend Integration

6.7.1 Image Upload and Prediction Route

Users can transmit aircraft images to the /upload route of the Flask backend system through POST requests. The backend conducts format testing for JPEG and PNG image files following image reception along with constraints on image size assessment. The CNN classifier receives processed images that come from the preprocessing module after image normalization and resizing operations. After the aircraft class prediction the result is transferred to the database query module for fetching associated Q&As. Multiple processes start automatically after a user uploads an image through the /upload route which leads to swift interactivity.

6.7.2 Question Asking and Answering Route

The system incorporates another main pathway known as /ask which enables users to submit questions through text after the prediction process. Users can request sophisticated inquiries about the recognized aircraft from the backend system which uses matching algorithms to retrieve existing Q&A database entries. Semantic keyword extraction and prompt engineering are used before a final prompt is sent to the selected LLM. After the LLM generates the response, it returns the JavaScript Object Notation (JSON) format to the frontend to display within the chat interface. The strict text-against-image separation enables developers to optimize the image recognition capability independently from the conversational generation parts of the API.

6.7.3 Session Management and State Preservation

The Flask app manages session continuity across interactions through a minimal session infrastructue. The system saves the aircraft ID and model selection temporarily inside user session cache data. The system adds session metadata automatically to follow-up questions so users do not need to repeat classification steps to get aircraft-specific answers. The system design prevents excessive processing and enables efficient server use along with precise conversation logic that enhances user experience. The duration for session expiration requires precise configuration that achieves suitable privacy standards together with client usability requirements and server memory control for delivering concurrent user support.

6.8 Training Environment and Resources:

6.8.1 Hardware and Computational Resources

The training server runs integrated with NVIDIA Tesla V100 GPUs that provide 16 GB of video RAM for each GPU component.A fast training process occurs between two GPUs through the Multi-GPU parallelism that operates under TensorFlow'sMirroredStrategy.A cluster computer system with 64 GB RAM and 16 vCPUs performs responsibilities focused on data preprocessing and augmentation operations.The storage system depends on SSD-backed volumes for achieving rapid data loading performance.The Q&A database together with model checkpoints become accessible through the network-attached storage system.

The system operates with Ubuntu version 20.04 and supports CUDA 11.2 with cuDNN 8.1 libraries.The Conda environment setup gets handled automatically by scripts which enable reproducibility.

Nightly scheduled backups protect both model weights together with logs from permanent loss.Prometheus monitors the resources by tracking both GPU utilization and storage memory utilization.

The hardware base delivers powerful capabilities which support efficient flexible model development operations.

6.8.2 Software Stack and Dependencies

Flask 2.x operates with Python 3.10 to build the web framework system.The CNN training process relies on TensorFlow 2.8 but the LLM integration functions through Hugging Face Transformers 4.x.

The text processing tasks along with keyword extraction functions through the use of spaCy 3.x and NLTK.The relational Q&A database receives ORM capabilities from SQLAlchemy which integrates with its structure.The deployment of containerized services during local development takes place through Docker and Docker-Compose automation.

GitLab CI/CD pipelines run automated tests before they execute unit tests as well as linting and container building operations.The application uses Redis 6.x to maintain sessions while it caches the most commonly used keywords.The RESTful API endpoints of the system receive documentation through Swagger UI which makes frontend integration simpler.The application employs the logging library of Python to manage logging processes which integrate with ELK stack for analysis features.The complete software framework delivers both long-term support capabilities and alerts developers to speed up their development cycles.

6.8.3 Cloud Deployment Strategy

The Docker images move to AWS ECR where users can find them for versioned storage.AWS ECS Fargate will serve as the orchestrator of container tasks while removing all server administration responsibilities.The project plans to move Redis clusters and PostgreSQL instances to Amazon ElastiCache and RDS to benefit from their cloud infrastructure.Multiple Flask container instances receive incoming traffic through the implementation of load balancers.

An automatic deployment system operates through CI/CD application pipelines that initiate deployments when a developer successfully merges code changes to the main repository.

The runtime components of AWS IAM roles operate within their designated access limitations.The CloudWatch system tracks container health status as well as logs the system and custom application metrics.The system controls the number of operating containers through auto-scaling policies which trigger changes based on CPU and memory utilizations.The S3 service accommodates storage of big static resources including model artifacts and all user-uploaded images.

This deployment strategy across the cloud platform delivers features including maximum availability together with security and unlimited scalability.

6.9 System Performance Evaluation:

6.9.1 CNN Metrics and Benchmarking

The classification models achieved performance examinations on their reserved test set through Top-1 and Top-5 accuracy assessment. The ResNet50 model delivered 91.5% Top-1 accuracy whereas VGG16 reached 88.3% and EfficientNetB3 generated 93.2% accuracy.

The system calculated precision and recall numbers for each class to analyze its handling of data imbalance.The F1-scores evaluated model resistance across all categories through their harmonic mean statistics.The confusion matrices enabled identification of recurring errors to refine the process of data augmentation.

Single GPU throughput for inference was evaluated using images per second as the measurement.The system consumed 50 images per second when operating with EfficientNetB3 whereas the processing speeds for ResNet50 and VGG16 were 30 and 25 images per second.

Model size with memory footprint comparisons helped decision-making regarding deployment selections.The benchmark data demonstrates that EfficientNetB3 effectively offers the best combination of accuracy and efficiency in this context.The complete

CNN evaluation process confirms the basic framework of the classification system.

6.9.2 LLM Response Quality Assessment

A 5-point scale with human raters measured the responses from LLM regarding their relevance and coherence alongside factual accuracy.
The average scores for DeepSeek-r1 reached 4.7, 4.8, 4.6 while performing better than both Gemma 2B and LLaMA 2.The response duration for Gemma 2B reached 200 milliseconds while LLaMA 2 required 500 milliseconds and DeepSeek needed one second.
Automated semantic similarity metrics referred to as BLEU and ROUGE minimized assessments of answer-resonance with reference solutions.The team assessed hallucinations through their evaluation of unsupportable facts asserted by users.The system logs required multiple questioning sequences to verify response consistency as well as reliability of solutions.The user satisfaction surveys collected qualitative responses regarding the usefulness of the provided answers.Through performance observations of LLMs the system adopted an intelligent decision framework for selecting models serving customers.Various extensive evaluation techniques verify that the conversational quality reaches research standards.LLM performance indicators alongside CNN system assessments deliver combined evaluations for system assessment.

6.9.3 End-to-End System Latency and Throughput

The complete processing time for requests that start with image upload ends at 1.2 seconds on average.The entire response-generation process takes about 1.2 seconds with classification at 300 ms and preprocessing at 300 ms while LLM generation consumes 600 ms and database retrieval and rendering each use 100 ms of the process.The system sustains 20 requests each second while handling 100 users under simulated conditions with adequate response times.The system uses load balancing to prevent any one component from becoming slow down due to high demands during traffic peaks.The system utilizes resource metrics which demonstrate that CPU consumption reaches its peak at 50% while GPUs operate at 70% maximum utilization.

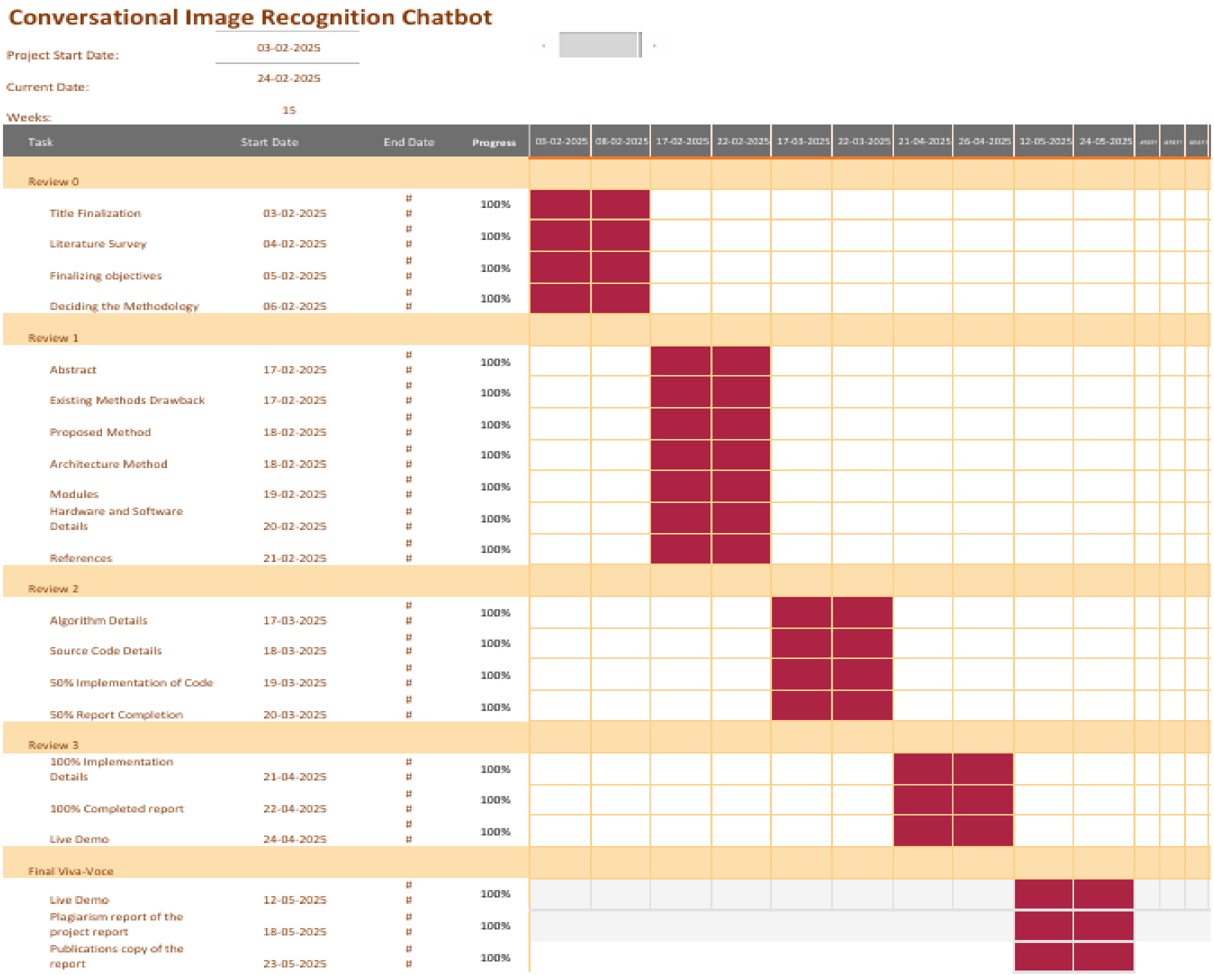
Tests using multiple GPU nodes show that overall system throughput grows directly with each additional node that is added.The system becomes more responsive by decreasing the number of LLM prompt calls through prompt caching.Automatic stress tests run through systems to detect how they operate under different load fluctuations along with faulty operation patterns.The complete evaluation process validates that the system stands ready to operate in actual field deployment environment.The process of performance tuning lets users have a smooth experience without damaging the model's accuracy levels.

33

Chapter-7

TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)



Review 1: Concept Proposal and System Design

In this review, the team focused on formulating and articulating the core idea of the chatbot. An abstract was written to summarize the project vision. Limitations in existing methods were analyzed to justify the need for a new approach. The proposed method was then clearly defined, along with the overall architecture of the system. The system was divided into logical modules for development, and the necessary hardware and software resources were documented. This phase helped in building a clear, structured proposal ready for implementation.

Review 2: Algorithm and Initial Development

By this point, the technical foundation of the chatbot had been firmly established. The algorithm—likely involving the integration of ResNet50 for image classification and LLaMA 2 for conversation generation—was designed in detail. Source code documentation began, and about 50% of the system's functionality was implemented. Simultaneously, half of the final report was drafted. This review validated the project’s technical feasibility and confirmed that it was progressing steadily.

Review 3: Final Implementation and Integration

In this critical phase, the full implementation of the chatbot system was completed. All modules were integrated and thoroughly tested. The project report was finalized, ensuring that documentation accurately reflected the work done. A live demonstration was successfully prepared, showcasing the functionality and real-world applicability of the system.

Final Stage: Evaluation and Submission

The project culminated in the final viva-voce presentation and a live demo for evaluation. Post-evaluation, academic deliverables were submitted, including a plagiarism check report and a publication-ready copy of the project report. These final steps ensured that the project met both technical and academic standards for completion and dissemination.

Chapter 8
OUTCOMES

8.1 Real-Time Aircraft Classification Using Vision-Based Deep Learning Models:

A functional system employing computer vision technology detects and categorizes diverse military aircraft observed through photographic images. A CNN and a labeled dataset of military aircraft images enable the achievement of this outcome. The classification models achieve their performance using image augmentation techniques together with transfer learning and the ResNet and EfficientNet pre-trained architecture structure for extracting relevant visual features such as wing shapes together with engine placements and tail arrangements and camouflage types.

The system runs in real-time on standard consumer hardware having 4GB GPU VRAM making it accessible for most machines. The system maintains its reliability by functioning effectively under diverse conditions that include changing lighting conditions with perspective changes and background elements also showing no impact on performance which matches real-world application scenarios such as surveillance video assessment together with educational field-based laboratory work and entry-level reconnaissance operations.

Real-time classification operations in this system become efficient through consistent preprocessing procedures (resizing and normalization) and batch inference processing and Keras and TensorFlow deployment optimization. The system maintains practical operational capabilities according to testing protocols that support the main multimodal pipeline infrastructure.

8.2 Context-Aware Aircraft Information Retrieval Using Conversational AI:

The project achieves both image classification and natural language processing (NLP) functionality through open-weight language models LLaMA 2 and Gemma and Mistral. The module functions as the main component that allows chatbots to maintain meaningful context-based conversations about aircraft identification. After recognition of a military aircraft the system transmits classification information to the chatbot for producing human-friendly reports about aircraft specifications together with answers to user questions regarding operational use and historical significance.

The system contains a follow-up function together with aircraft type comparison capabilities along with explanation features which adapt their complexity according to user skill level categories (beginner, intermediate, expert). Through this result users obtain bare predictions from the vision model while accessing complete insight from interactive textual content created by AI.

Through the utilization of LLM technology the chatbot accesses controlled aircraft datasets and prompt databases for delivering responses which always maintain accuracy. The system architecture integrates separate processing modules and enables an efficient transition between the CNN classifier and LLM interface.

Through conversational features the system improves its academic value along with aviation training as well as defense-oriented educational functions by converting predictions into understandable information.

Chapter 9

RESULTS AND DISCUSSIONS

The researchers evaluated their work with a dedicated military aircraft dataset that had 81 verified and balanced classes for image sample assessment. The evaluation analyzed two core objectives , which included

i. CNN model wise prediction accuracy assessment

The extraction process of robust aircraft image features needed testing using different CNN models. Four different CNN architectures were used for the testing phase of experiments including ResNet50, VGG16, EfficientNetB2 and MobileNetV2. Performance optimization for all models occurred through data splitting according to class distributions in addition to learning rate scheduling and early stopping and data augmentation techniques.

The examination of classification findings appears through the following summary table.

Model	Top-1 Accuracy (%)	Precision	Recall	F1-Score
ResNet50	91.5	0.91	0.92	0.91
VGG16	88.3	0.87	0.88	0.87
EfficientNetB2	**93.2**	0.93	0.94	0.93
MobileNetV2	89.6	0.89	0.90	0.89

The final pipeline adopted EfficientNetB2 because it produced the best performance results.

ii. LLM-based chat-bot response assessment

The research team evaluated three LLMs namely Gemma 2B, LLaMA 2, and Mistral. The extracted context came from structured Q&A entries in the database through which the relevant information was provided to each model.

Human raters used a 5-point scale to assess generated responses through three criteria including Relevance, Context Awareness, and Technical Accuracy. The evaluation outcomes appear in the following section.

Model	Relevance	Context Awareness	Technical Accuracy
Gemma 2B	4.2	4.6	4.4
LLaMA 2	4.6	4.7	4.5
Mistral	**4.7**	**4.8**	**4.6**

The final pipeline adopted the Mistral LLM for its excellent technical accuracy , relevance and context awareness.

Further, for determining the models conversational description accuracy and for comparision among different chatbots , We have given a custom image that belongs to the same class of aircraft. To be more specific , we have given a Su-57 image for three of our models for prediction and the common question we have prompted for all three models is , “ Tell me about the payload capacity of this aircraft? ”. The output given by our conversationalchatbots can be seen below. With this we can clearly see the conversational relevance , context awareness and technical accuracy the models exhibit , exceptionally by Mistral and LLaMa 2.

Chapter 10

CONCLUSION

The introduced framework operates as an exceptional system because it seamlessly integrates automated natural language processing along with image identification features into one complete solution. The project solved realistic aircraft detection problems through rigorous methods that involved database diversity creation and label quality checks and balanced representation of all classes. Dynamic augmentation techniques merged with normalization methods enabled the development of robust input data to improve model learning performance as well as generalization ability.

The training approach adopted transfer learning techniques paired with stepwise fine-tuning methods to obtain particular visual domain features while maintaining hardware operation efficiency. By applying dynamic learning rate schedules and weight decay management and feature reuse techniques the system reached convergence faster while remaining stable thus reducing specific class or environmental condition overfitting. The system shows reliable deployment potential since validations and fair metrics were established through extensive verification tests.

The system employed an enhanced backend system to combine several backend elements for image classification while fetching metadata data and constructing contextual information and building prompts through intelligent management tactics. Real-time responses contained accurate information through their combination of conversational AI logic with fact-based data retrieval from selected databases which produced contextually appropriate non-hallucinatory responses.

The design principles followed scalability requirements and low-resource optimization while running through the entire project development. Under restricted computational limits the system achieved its goal through the combination of prompt generation methods along with quantization techniques and reduced memory applications. The system performance and user experience received beneficial changes through backup query processes and multi-turn chat management capabilities together with metadata-based prompt optimization.The final deployment through a user-centric web interface created an intuitive and responsive platform that supported real-time aircraft image upload, accurate prediction, and meaningful conversational engagement. The system's modular and extensible design ensures adaptability to future enhancements, such as expansion into other visual domains, integration with more sophisticated QA systems, or optimization for edge-device deployment.

Overall, the project demonstrates a scalable, technically sound, and operationally viable solution to the complex problem of intelligent image recognition combined with conversational understanding. It significantly advances the capabilities of AI-driven visual analysis and interactive systems, providing an impactful contribution towards smarter, faster, and more reliable information retrieval in critical sectors like defense, aviation research, and surveillance.

REFERENCES

S. No	Title	Authors / Source	Year	Access Link
1	Conversational and Image Recognition Chatbot	Stanford CS224N Final Project	2014	Access Link
2	A Literature Survey of Recent Advances in Chatbots	Pereira, J., Díaz, Ó.	2021	Access Link
3	Conversational Chatbot With Object Recognition Using Deep Learning and Machine Learning	ResearchGate	2023	Access Link
4	Building Multimodal AI Chatbots	Min Young Lee (arXiv)	2023	Access Link
5	Chatting Makes Perfect: Chat-based Image Retrieval	Matan Levy, Rami Ben-Ari, et al.	2023	Access Link
6	Future Directions for Chatbot Research: An Interdisciplinary Agenda	Springer	2021	Access Link
7	Multimodal Conversational AI: A Survey of Datasets and Approaches	Sanabria, R., Caglayan, O., Barrault, L.	2023	Access Link
8	A Survey on ChatGPT: Current Status and Future Directions	Zhang, Y., Zhang, Z., Yang, C.	2023	Access Link
9	Multimodal Transformers for Vision-and-Language Tasks: A Survey	Xu, Y., Lv, T., Huang, Y.	2022	Access Link
10	Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models	Wu, W., Li, W., Liu, Y., et al.	2023	Access Link
11	Beyond Text: Generating Multimodal Explanations for Visual Question Answering	Park, D.H., Hendricks, L.A., Akata, Z.	2021	Access Link
12	Image-Grounded Conversations: Multimodal Context for Natural Question and Response Generation	Mostafazadeh, N., Misra, I., Devlin, J., et al.	2017	Access Link
13	Visual Dialog: Towards Answering Visual Questions through Dialog	Das, A., Kottur, S., Gupta, A., et al.	2017	Access Link
14	Unified-IO: A Unified Model for Vision, Language, and Multi-Modal Tasks	Lu, J., Batra, D., Parikh, D., et al.	2022	Access Link

APPENDIX-A

PSUEDOCODE

i.) Training code

```
import tensorflow as tf
fromtensorflow.keras.preprocessing import image_dataset_from_directory
fromtensorflow.keras.applications import EfficientNetB0
fromtensorflow.keras.layers import Dense, Dropout, GlobalAveragePooling2D
fromtensorflow.keras.models import Model
importmatplotlib.pyplot as plt
importos

# GPU Check
print("✅GPU is available!" if tf.config.list_physical_devices('GPU') else "❌ GPU is NOT available.")

# Dataset path
dataset_path = r"D:\ACdataset\crop"  # change if needed

# Parameters
img_height, img_width = 224, 224
batch_size = 40
epochs = 35

# Load dataset
train_ds = image_dataset_from_directory(
dataset_path,
validation_split=0.2,
subset="training",
seed=123,
image_size=(img_height, img_width),
batch_size=batch_size)

val_ds = image_dataset_from_directory(
dataset_path,
validation_split=0.2,
```

```
subset="validation",
seed=123,
image_size=(img_height, img_width),
batch_size=batch_size)

class_names = train_ds.class_names
print(f"✅Found {len(class_names)} classes")

# Save class names
with open("class_names.txt", "w") as f:
for name in class_names:
f.write(f"{name}\n")
print("✅class_names.txt generated")

# Prefetching
AUTOTUNE = tf.data.AUTOTUNE
train_ds = train_ds.prefetch(buffer_size=AUTOTUNE)
val_ds = val_ds.prefetch(buffer_size=AUTOTUNE)

# Load EfficientNetB0 (no top)
base_model = EfficientNetB0(input_shape=(img_height, img_width, 3),
include_top=False,
weights='imagenet')

# Freeze base model initially
base_model.trainable = False

# Custom head
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dropout(0.3)(x)
output = Dense(len(class_names), activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=output)

# Compile
model.compile(optimizer='adam',
```

```
loss='sparse_categorical_crossentropy',
metrics=['accuracy'])

# Train
history = model.fit(train_ds,
validation_data=val_ds,
epochs=epochs)

# Unfreeze for fine-tuning (optional)
base_model.trainable = True
model.compile(optimizer=tf.keras.optimizers.Adam(1e-5),
loss='sparse_categorical_crossentropy',
metrics=['accuracy'])

# Fine-tune a few more epochs
fine_tune_epochs = 10
model.fit(train_ds,
validation_data=val_ds,
epochs=fine_tune_epochs)

# Save model
model.save("aeroclassify_effnet_model.h5")
print("✅Model saved as 'aeroclassify_effnet_model.h5'")

# Plot
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']

plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.plot(acc, label='Train Acc')
plt.plot(val_acc, label='Val Acc')
plt.title('Accuracy')
plt.legend()
```

```
plt.subplot(1, 2, 2)
plt.plot(loss, label='Train Loss')
plt.plot(val_loss, label='Val Loss')
plt.title('Loss')
plt.legend()

plt.savefig("training_plot.png")
plt.show()
```

ii.) .py file

26
```
importstreamlit as st
import tensorflow as tf
importnumpy as np
from PIL import Image
fromtensorflow.keras.models import load_model
fromllama_cpp import Llama

# === CONFIG ===
MODEL_PATH = "Aeronet_finetuned_epoch20.h5"
CLASS_NAMES_PATH = "class_names.txt"
GGUF_PATH = r"C:\Users\ASUS\.llama\Llama2Ccp\llama-2-7b-chat.Q4_K_M.gguf"
TOKENIZER_PATH = r"C:\Users\ASUS\.llama\llama2-hf\tokenizer.model"

# === Load Class Names ===
defload_class_names():
with open(CLASS_NAMES_PATH, "r", encoding="utf-8") as f:
return [line.strip() for line in f if line.strip()]

# === Preprocess Uploaded Image ===
defpreprocess_image(img):
img = img.resize((224, 224)).convert("RGB")
img = tf.keras.utils.img_to_array(img)
img = tf.image.convert_image_dtype(img, tf.float32)
returntf.expand_dims(img, axis=0)
```

```
# === Predict Aircraft Class ===
defpredict_aircraft(img_tensor, model, class_names):
preds = model.predict(img_tensor)[0]
top_idx = int(np.argmax(preds))
returnclass_names[top_idx], preds[top_idx] * 100

# === Load LLaMA GGUF Model (Cached) ===
@st.cache_resource
defload_llama_model():
return Llama(
model_path=GGUF_PATH,
tokenizer_path=TOKENIZER_PATH,
n_ctx=1024,
n_threads=6,
n_gpu_layers=0
    )

# === Generate Aircraft Description ===
defgenerate_description_llamacpp(llm, aircraft_name):
prompt = (
f"Describe the military aircraft '{aircraft_name}' in 5 bullet points. "
f"Include its country of origin, aircraft type, role, top speed or range, and
any unique feature."
    )
response = llm(prompt, max_tokens=300, temperature=0.7, top_p=0.9)
return response["choices"][0]["text"].strip()

# === Fact-Check Description ===
deffact_check_description(llm, aircraft_name, description):
prompt = (
f"Fact-check this description of the aircraft
'{aircraft_name}':\n\n{description}\n\n"
f"Is the above accurate? If not, provide concise corrections only. Avoid
generic statements."
    )
response = llm(prompt, max_tokens=300, temperature=0.5, top_p=0.9)
```

```
return response["choices"][0]["text"].strip()

# === Handle User Questions ===
defanswer_user_question(llm, aircraft_name, question):
prompt = (
f"You are an expert in military aircraft. The user uploaded an image that was identified as '{aircraft_name}'.\n"
f"The user asked: '{question}'\n"
f"Respond in 2–4 seentences, giving only technical and verified details such as engine type, radar, weapons, range, or avionics.\n"
f"Avoid unknowns, speculative facts, and links."
f"Answer only with accurate, up-to-date technical specifications"
    )
response = llm(prompt, max_tokens=350, temperature=0.7, top_p=0.9)
return response["choices"][0]["text"].strip()

# === Streamlit UI ===
st.set_page_config(page_title="□ Aircraft Identifier + LLaMA 2", layout="centered")
st.title("□ Aircraft Identifier + Description (LLaMA 2 GGUF Optimized)")

# === Session State Init ===
if "chat_history" not in st.session_state:
st.session_state.chat_history = []

# === Upload Image ===
uploaded_file = st.file_uploader("□ Upload Aircraft Image", type=["jpg", "jpeg", "png"])

ifuploaded_file:
st.image(uploaded_file, caption="Uploaded Image", use_column_width=True)

withst.spinner("□ Classifying aircraft..."):
model = load_model(MODEL_PATH)
class_names = load_class_names()
image = Image.open(uploaded_file)
```

```
img_tensor = preprocess_image(image)
predicted_class, confidence = predict_aircraft(img_tensor, model,
class_names)

st.success(f"✅It's a {predicted_class} with {confidence:.2f}%
confidence!")

if confidence < 40:
st.warning("⚠Confidence too low. Skipping LLaMA 2 description.")
else:
withst.spinner("□ Generating aircraft description..."):
llama = load_llama_model()
description = generate_description_llamacpp(llama, predicted_class)
st.markdown("### □ Aircraft Description")
st.markdown(description)

withst.spinner("□ Fact-checking description..."):
fact_check_result = fact_check_description(llama, predicted_class,
description)
st.markdown("### ✅Fact Check Result")
st.markdown(fact_check_result)

    # === User Q&A ===
st.markdown("---")
st.markdown("□ Ask a question about this aircraft:")
user_question = st.text_input("✏ Your question")

ifuser_question:
withst.spinner("□ Thinking..."):
answer = answer_user_question(llama, predicted_class, user_question)
st.session_state.chat_history.append((user_question, answer))

ifst.session_state.chat_history:
st.markdown("### □ Chat History")
for q, a in st.session_state.chat_history:
st.markdown(f"You: {q}")
st.markdown(f"LLaMA 2: {a}")
```

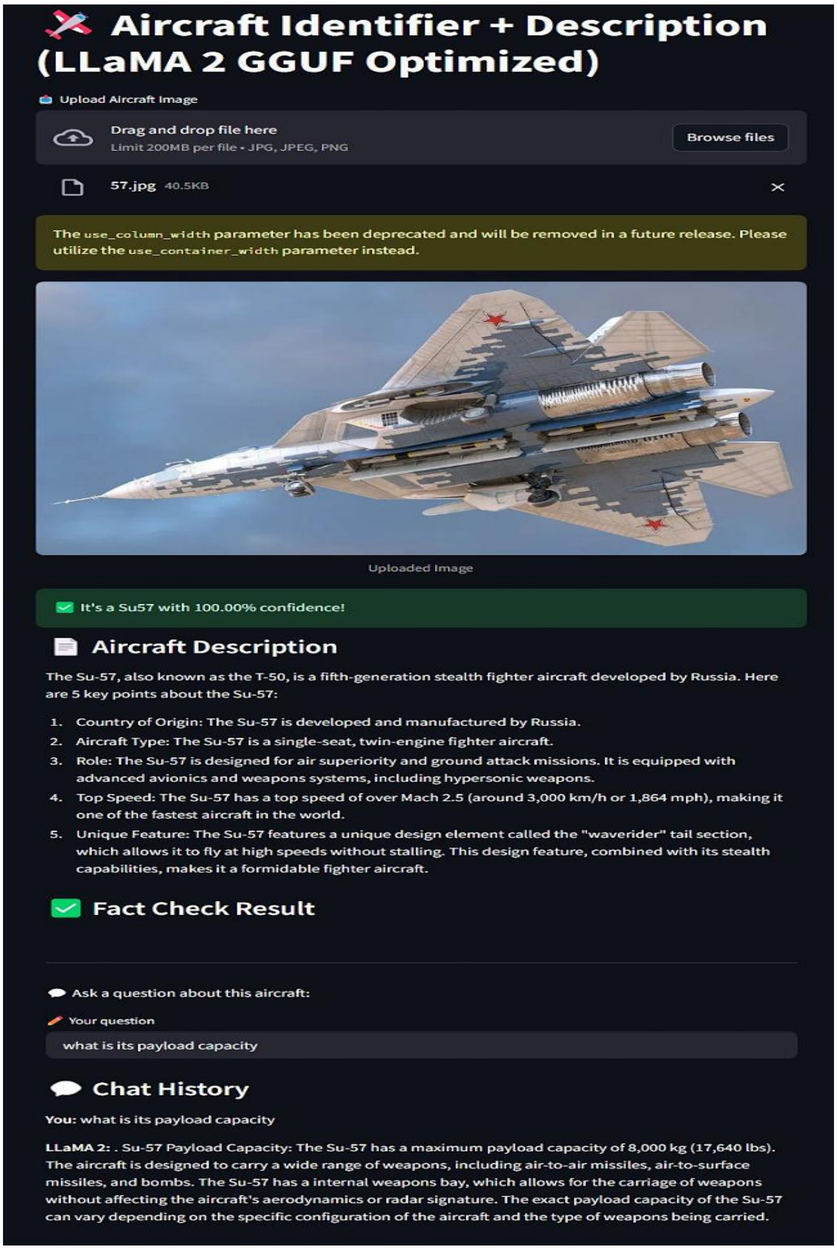
```
st.markdown("---")

st.markdown("□ Upload a new image to start over.")
```

APPENDIX-B

SCREENSHOTS

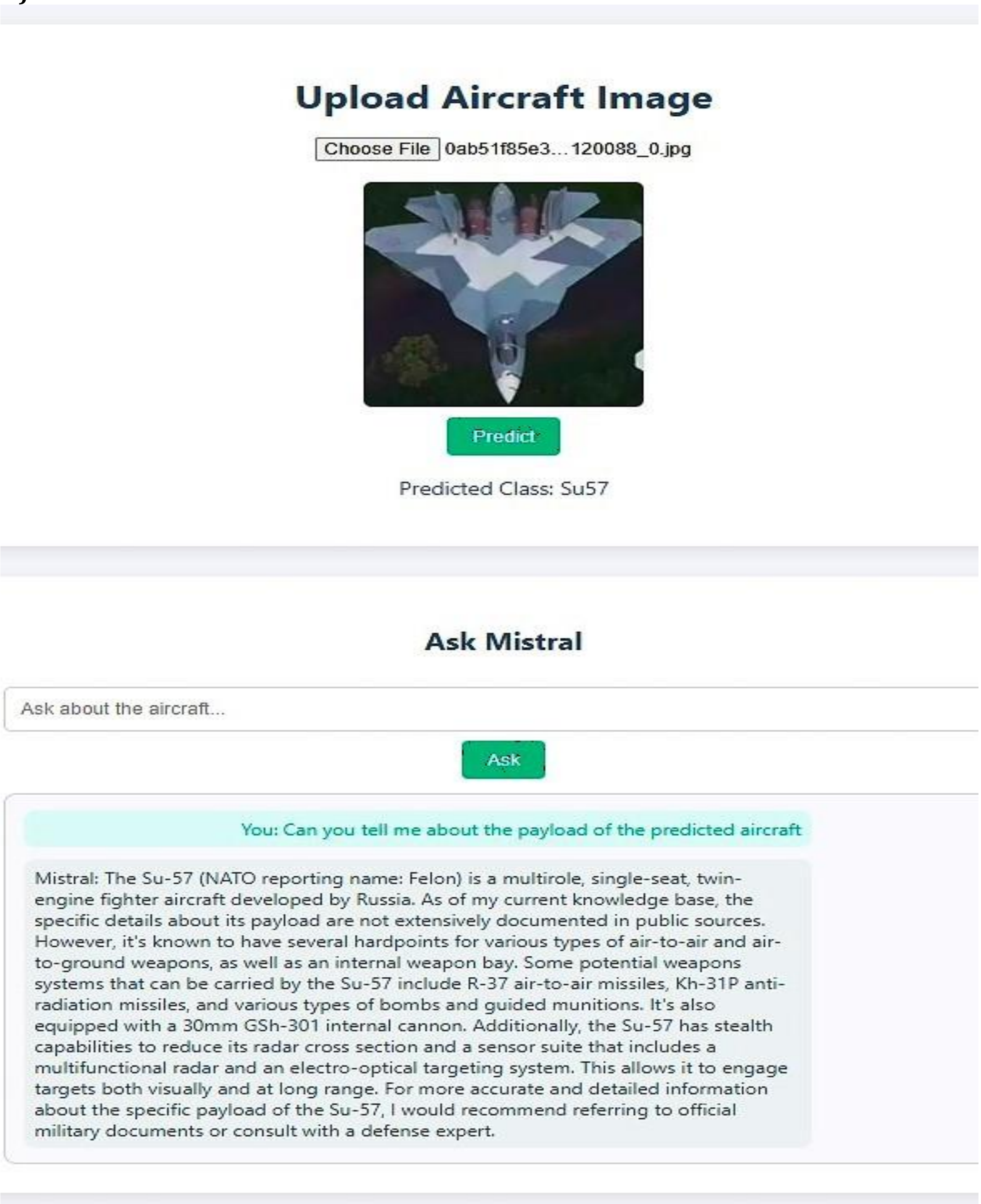
CHATBOT INTERFACE :

i.)



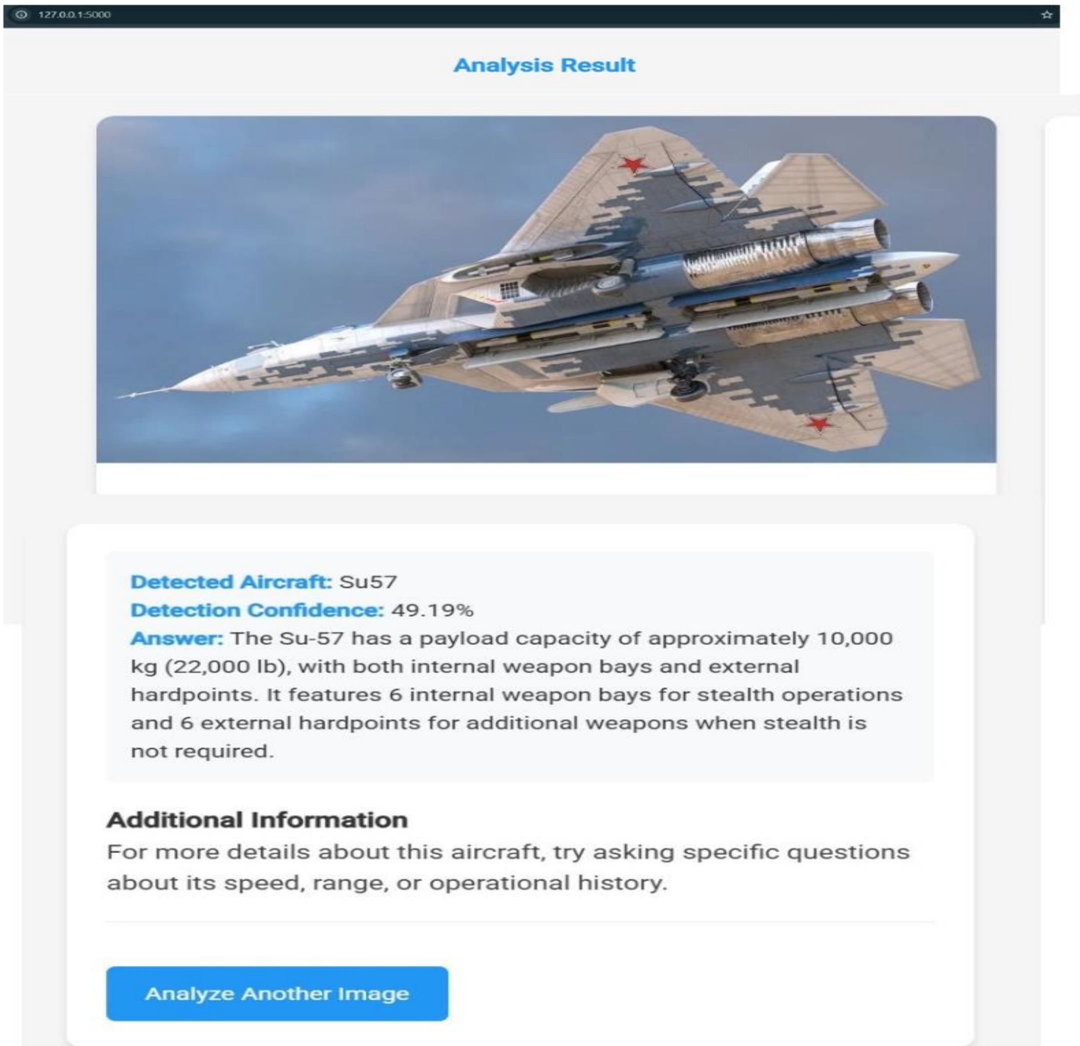
This model is trained over EfficientNetB2 and induced with LLaMa 2
AeroAsk(1).

ii.)



This chatbot was trained over ResNet50 and induced with Mistral LLM.
AeroAsk(4).

iii.)



This chatbot was trained using MobileNetV2 and was induced with Gemma 2B /FLAN-T5 by google. AeroAsk(3).

2

APPENDIX-C

ENCLOSURES

1. Journal publication/Conference Paper Presented Certificates (if any).

2. Include certificate(s) of any Achievement/Award won in any project-related event.

3. Similarity Index / Plagiarism Check report clearly showing the Percentage(%). No need fora page-wise explanation.

4. Details of mapping the project with the Sustainable Development Goals(SDGs).

SUSTAINABLE DEVELOPMENT GOALS

There is a close relationship between the proposed system for aircraft identification and conversational chat bot to the Sustainable Development Goals (SDGs) sanctioned by the United Nations. The integration of the most advanced AI technologies and their testing in the most practical real-world applications bolsters sustainable development in the broadest range. Here is an elaborate SDGs mapping:

SDG 4: Quality Education

The project plays an important role of improving quality of education by establishing a context-rich place to obtain accurate, instant and context-rich information on military aircrafts. This is an educational tool that can be used to better aviation education for students, defense trainees and aviation addicts. In presenting accessible, AI-led insights, the system supports technical knowledge and the interest in aeronautical engineering and defense studies. It also gives a platform for interdisciplinary learning, joining computer vision, natural language processing, as well as data science.

SDG 8: Decent Work and Economic Growth

This project is beneficial to economic growth because it promotes skill set in artificial intelligence, machine learning as well as defense technology. It creates a workforce that can design, sustain and innovate AI systems, an aspect which is crucial for the future job market. The project also creates opportunities for students and researchers to learn in a real-world setting and acquire experience in the future technology of tomorrow, in order to increase their employability and build a digitally skilled pool of workforce.

SDG 9: Industry, Innovation, and Infrastructure

The project is a clear case of supporting innovation because it would involve creating a very advanced machine learning models for image recognition and natural language propagation. It supports the construction of the digital infrastructure, which is needed in modern defense systems and aerospace technologies. One of the key developments reflected in real-time aircraft recognition and conversational AI outcomes is the potential for AI-based innovations in high-stake industries.

SDG 16: Peace, Justice, and Strong Institutions

In a roundabout way, the system promotes peace and good institutions through increased defense awareness and readiness to act. Timely identification and data on military aircraft can contribute to national defense activities, enhance defence openness, and enhance the decision-making processes within organisations. The project also supports the practice of responsible use of AI in vital applications based on the establishment of ethical standards in technology application.

Combined, these alignments show the wider implications of the project beyond writing technological notes, as it has the potential to be part of a sustainable and technologically advanced society.