SOMAIYA VIDYAVIHAR UNIVERSITY

Automated Description Generation for Jewellery Images using Deep Learning

Submitted In Partial Fulfillment of Requirements
For the Degree Of

Bachelor of Technology
(Information Technology)

By

Tanisha Ashish Mangaonkar
Roll No: 16010422200

Prachi Sanjay Gandhi
Roll No: 16010422233

Chandana Ramesh Galgali
Roll No: 16010422234

Mahek Jaladhi Thakkar
Roll No: 16010422235

Guide

Prof. Avani Sakhapara

SOMAIYA VIDYAVIHAR UNIVERSITY

Somaiya Vidyavihar University
Vidyavihar, Mumbai - 400 077
2022-26

Somaiya Vidyavihar University

K. J. Somaiya College of Engineering

Certificate

This is to certify that the dissertation report entitled **Automated Description Generation for Jewellery Images using Deep Learning** submitted by Tanisha Mangaonkar, Prachi Gandhi, Chandana Galgali and Mahek Thakkar at the end of semester VII of LY B. Tech is a bona fide record for partial fulfillment of requirements for the degree Bachelor of Technology (Information Technology) of Somaiya Vidyavihar University.

Guide

Head of the Department

Principal

Date: 28/11/2025

Place: Mumbai-77

Somaiya Vidyavihar University

K. J. Somaiya College of Engineering

DECLARATION

We declare that this written report submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

P.Gandhi Signature of the Student 16010422233 Roll No.	Signature of the Student 16010422234 Roll No.
Signature of the Student 16010422200 Roll No.	Signature of the Student 16010422235 Roll No.

Date: 28/11/2025

Place: Mumbai-77

Somaiya Vidyavihar University

K. J. Somaiya College of Engineering

Certificate of Approval of Examiners

We certify that this dissertation report entitled **Automated Description Generation for Jewellery Images using Deep Learning** is a bona fide record of project work done by Tanisha Mangaonkar, Prachi Gandhi, Chandana Galgali and Mahek Thakkar during semester VII.

This project work is submitted at the end of semester VII in partial fulfillment of requirements for the degree of Bachelors of Technology in Information Technology of Somaiya Vidyavihar University.

Internal Examiners

A. S. Nazini

External/Internal/Expert Examiners

Date: 28/11/2025

Place: Mumbai-77

Abstract

The rapid growth of online jewellery retail has created a demand for automated systems that can accurately classify and describe jewellery items from images. Manual cataloging is time-consuming, inconsistent, and heavily dependent on domain expertise. This project proposes a deep learning–based system that automates jewellery detection, classification, and caption generation from user-uploaded images. The system utilizes a VGG-16 encoder to identify jewellery items such as earrings and necklaces and extract visual attributes, including metal colour and gemstone presence. A GRU-based decoder is then used to generate natural-language descriptions capturing the appearance and characteristics of the jewellery. The model is trained on a curated and augmented dataset of 32,190 jewellery images, achieving a classification accuracy of 94.62% and a description accuracy of 90.44%. The entire pipeline is integrated into a React-based user interface for seamless interaction. The proposed system provides a fast, reliable, and scalable solution for digital jewellery cataloging, helping jewellers, e-commerce platforms, and designers generate consistent and meaningful descriptions automatically.

Key words**:** Jewellery Detection, Deep Learning, Image Classification, VGG-16, GRU Decoder, Image Captioning, Feature Extraction, Metal Colour Detection, Gemstone Identification, Automated Description Generation, Computer Vision, React UI, PyTorch.

`

Contents

`

`

List of Figures

`

List of Tables

`

Nomenclature

ABS	Adaptive Background Subtraction
AI	Artificial Intelligence
API	Application Programming Interface
CNN	Convolutional Neural Network
CORS	Cross-Origin Resource Sharing
DL	Deep Learning
F1	F1 Score (Harmonic mean of precision and recall)
FPS	Frames Per Second
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Unit
HSV	Hue, Saturation, Value
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IoU	Intersection over Union
JSON	JavaScript Object Notation
LIBS	Laser-Induced Breakdown Spectroscopy
LSTM	Long Short-Term Memory
mAP	Mean Average Precision
ML	Machine Learning
OS	Operating System
PIL	Python Imaging Library

`

R-CNN	Region-based Convolutional Neural Network
REST	Representational State Transfer
RF	Random Forest
RNN	Recurrent Neural Network
ROI	Region of Interest
RPN	Region Proposal Network
SQL	Structured Query Language
SRS	Software Requirements Specification
UI	User Interface
UUID	Universally Unique Identifier
VGG-16	Visual Geometry Group (16-layer architecture)
YOLO	You Only Look Once

`

1. Introduction

Jewellery identification plays a crucial role in cataloging, retail management, cultural preservation, and digital documentation. Traditional manual processes are slow, subjective, and prone to human error. Additionally, most existing systems identify jewellery only at a basic object-level, without understanding features such as metal colour, gemstone type, or stylistic attributes. With recent advancements in Deep Learning, particularly in feature extraction and neural captioning, it has become possible to automate detailed jewellery understanding from images.

This project presents an AI-based system that detects jewellery items (earrings or necklaces) from an input image and generates natural-language descriptions covering their visual properties such as metal colour and gemstone presence. The model uses VGG-16 as an encoder and a GRU-based decoder for caption generation. The overall objective is to automate jewellery description generation with high accuracy, enabling faster cataloging and smarter retail tools.

1.1 Problem Definition

The problem addressed in this project is to develop an automated system capable of detecting, classifying, and describing jewellery items from an input image.

The system:

- Detects whether the item is an earring or necklace
- Extracts features such as metal colour, gemstone presence and gemstone type
- Generates a natural-language caption describing the jewellery
- Produces structured output usable for cataloging or digital archiving

1.2 Motivation

Manual identification of jewellery is slow and often inconsistent. Retailers, online stores, and cataloging systems require quick and accurate descriptions to maintain large digital inventories. Existing systems mostly focus on object detection but do not provide detailed semantic descriptions, such as metal type or gemstone colour. Deep Learning offers the ability to extract fine-grained visual details and generate meaningful captions, making jewellery classification more efficient, scalable, and user-friendly. The motivation also extends to:

- Reducing human workload in jewellery cataloging
- Providing descriptive metadata for e-commerce applications
- Preserving traditional jewellery designs through AI-generated digital descriptions
- Improving searchability in digital archives

1.3 Salient Contributions

This project provides the following key contributions:

1. Automated Jewellery Detection

A Deep Learning model using VGG-16 identifies whether the uploaded image contains an earring or a necklace.

2. Feature Extraction for Metal Colour & Gemstones

The encoder extracts visual attributes such as:

- type of metal (gold/silver)
- gemstone type and presence

3. Caption Generation using GRU Decoder

A GRU-based language model produces descriptive sentences, such as:

"A round necklace with emerald green gemstones set in yellow gold."

4. High Accuracy Performance

The system achieves a jewellery classification accuracy of 94.62% and a description generation accuracy of 90.44%.

5. User-Friendly Interface

A simple React-based UI allows users to upload jewellery images and view descriptions.

These combined features create a complete pipeline that moves beyond object detection to automated semantic understanding.

1.4 Scope of the Project

Functional Scope

- Accepting image input of jewellery (earring/necklace)
- Detecting and classifying jewellery type
- Extracting features such as metal colour and gemstone presence
- Generating detailed natural-language captions

- Providing structured descriptions as output
- Offering a user interface for uploading images and displaying results

Non-Functional Scope

- **Accuracy:** Achieve high precision in detection and captioning (94.62% and 90.44% accuracy obtained in jewellery classification and description generation respectively).
- **Response Time:** Generate output within 1–2 seconds.
- **Usability:** Interface must be simple and suitable for non-technical users like jewellers.

1.5 Organization of the Report

Chapter 1 provides an introduction to the project, outlining the problem definition, motivation, scope, and salient contributions. It establishes the need for an automated system that classifies jewelry images and generates high-quality descriptions using deep learning and language models.

Chapter 2 presents a comprehensive literature survey, reviewing existing work in image classification, CNN-based feature extraction, product description generation, and automated cataloging systems. It also identifies research gaps that this project aims to address.

Chapter 3 describes the Software Project Management Plan, detailing the overall project overview, deliverables, project organization, selected software process model, roles and responsibilities, tools and techniques, and the complete project schedule. It also outlines tasks, milestones, risks, contingencies, and resource requirements for successful project execution.

Chapter 4 contains the Software Requirements Specification (SRS), defining the system's functional and non-functional requirements. It includes the product overview, specific feature requirements, external interfaces, user and hardware interfaces, software attributes, communication protocols, and database specifications.

Chapter 5 focuses on the Software Design Description, explaining the system architecture, design overview, requirement traceability matrix, and detailed component designs. It also covers the user interface design, including screen layouts and interface descriptions that guide implementation.

Chapter 6 presents the Software Test Document, describing the test approach, features to be tested, features not tested, testing tools, test environment, and detailed test cases used to verify the system's correctness, performance, and reliability.

Chapter 7 concludes the report by summarizing the project outcomes, highlighting the system's performance in jewelry classification and description generation, and suggesting future enhancements such as improved fine-grained attribute detection, multilingual descriptions, and deployment for real-time e-commerce applications.

2. Literature Survey

This chapter presents a detailed review of existing work that supports the development of the proposed jewellery detection, classification, and description generation system. The literature survey is divided into four major areas: Jewellery Detection and Identification, General Object Detection and Tracking, Gemstone and Material Classification, and Image Captioning.

The first section discusses research focused on identifying jewellery items such as earrings, necklaces, and rings using deep learning models. The second section reviews modern object detection approaches, such as YOLO, Faster R-CNN, and CNN-based detectors, that form the foundation of visual understanding. The third section examines gemstone and metal classification methods used for identifying material attributes crucial to jewellery analysis. The final section explores encoder–decoder architectures and captioning techniques capable of generating natural-language descriptions from image features.

This chapter highlights limitations in existing research, such as restricted jewellery classes, lack of semantic description generation, and limited attribute extraction, justifying the need for an integrated system that performs both jewellery classification and descriptive caption generation from a single image input.

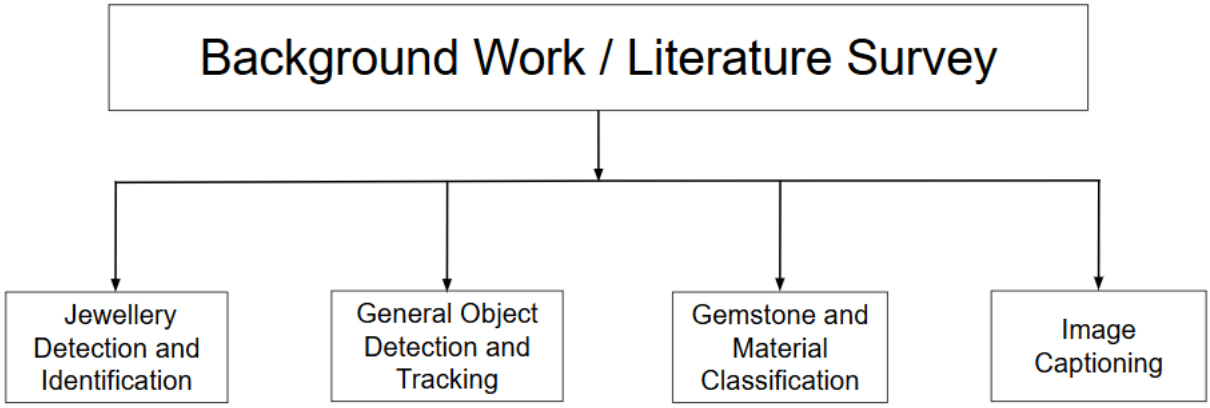

Figure 2.1: Key Areas Explored in the Literature Survey

Jewellery Detection and Identification

This section examines research focused specifically on identifying jewellery items in images. Several works employ transfer learning, custom CNN models, and YOLO-based architectures for categories such as earrings, necklaces, rings, and accessories. These studies demonstrate that convolutional neural networks can detect and classify jewellery efficiently; however, the scope is

often limited to identification without generating descriptive information.

Existing systems also depend heavily on curated datasets and struggle with fine-grained differences between designs, shapes, and colours. This creates a clear gap in automated domain-specific description generation.

Table 2.1: Summary of Research Papers on Jewellery Detection

Year	What is the paper about? (Aspects)	Methodology (Steps in 2-3 lines)	Datasets (Size, Type, etc.)	Results (Validation Metrics)	Advantages	Limitations	Reference No.
2024	Uses image captioning for jewellery classification in e-commerce.	Used VGG-16 and MobileNet as CNN encoders, GRU and LSTM as RNN decoders to generate multi-level jewellery captions, followed by classification using parsed caption attributes.	Created a comprehensive jewellery image database from local jewellery stores.	Achieved high accuracy using VGG-16 + GRU with strong F1-scores.	Accurate and robust jewellery detection with end-to-end automated recognition, handling diverse styles and occlusions effectively.	Performance drops on similar-looking items like bracelets, relies on high-quality labeled datasets, and is limited to predefined jewellery categories.	[1]
2023	Applies CNNs and Faster R-CNN for accurate image/video detection.	Used Faster R-CNN with a CNN backbone to extract features from Yakshagana images, applied region proposal networks (RPN) to locate small jewel regions, followed by ROI pooling and classification layers for precise jewel detection.	No detailed dataset info given; tested on public jewellery item datasets (implied).	Achieved high mAP and precision using Faster R-CNN for small jewel detection.	Faster R-CNN offers high precision, excels at detecting small jewellery items in complex images, and leverages deep learning adaptability for reliability.	Computationally intensive, needs large datasets, and is limited to Yakshagana-specific images.	[2]
2021	Automates jewellery tagging via transfer learning and live feeds.	Created an image repository of jewellery, labeled images by category, trained a transfer learning model, and used OpenCV for real-time classification via live camera feed.	Used a manually created image repository of jewellery articles for training and validation.	Achieved accurate, real-time jewellery recognition with strong validation metrics.	Automates real-time jewellery tagging and classification using transfer learning, reducing manual errors, manpower, and enabling accurate live camera-based recognition.	Requires a large, well-labeled image repository, with performance heavily influenced by image quality and lighting, and may struggle with unseen jewellery types without retraining.	[3]

2025	Proposes neural network for automatic jewellery description generation.	Used computer vision and various image captioning architectures, especially encoder-decoder models, trained on a comprehensive jewellery image database.	Built a large jewellery image dataset to train image captioning models.	Achieved high captioning accuracy with detailed descriptions of diverse jewellery.	Assists non-experts with detailed jewellery insights, generating accurate, hierarchical captions across varied styles.	Depends on image database quality, struggles with rare designs, and requires intensive training for captioning models.	[4]
2024	Develops vision method to track gold necklaces for theft prevention.	Improved traditional Gaussian Mixture Model (GMM) with adaptive background subtraction (ABS) for enhanced detection and tracking of gold necklaces.	Used video/image sequences from gold shops for detection and tracking evaluation.	Achieved high frame tracking accuracy, outperforming the standard GMM method.	Tracks small, deformable objects effectively with ABS-enhanced accuracy, aiding theft prevention in gold shops.	Limited to shop settings, sensitive to occlusion and fast movement, and reliant on lighting and camera quality.	[5]
2023	Presents mobile-friendly FC-YOLOv4 for fashion item detection.	Developed and compared a custom FC-YOLOv4 model with YOLOv3 and YOLOv4 using a dataset of 13,689 images across 10 categories, evaluated on mobile devices.	Dataset of 13,689 images covering five fashion and five accessories categories.	Achieved high mAP and IoU with reduced size and mobile efficiency.	Achieves extremely high accuracy (99.84% mAP), optimized for low-RAM mobile devices, and ensures faster detection for efficient e-commerce product categorization.	Limited to predefined categories, needs large labeled datasets, and performance varies across smartphone hardware.	[6]
2023	Proposes jewellery retrieval using local HSV color histograms.	Extracted feature vectors from five local regions in HSV space, applied a classification module, and matched similarity scores for jewellery retrieval.	Used publicly available jewellery item retrieval datasets: ringFIR and Fashion Product Images.	Outperformed baselines on ringFIR and Fashion datasets in retrieval accuracy.	Effective in handling occlusion and shape deformation Lightweight and color-focused feature extraction Performs well on real-world jewellery datasets	Limited to HSV color space features May struggle with grayscale or low-color-contrast images Less robust compared to deep learning-based retrieval methods	[7]

General Object Detection and Tracking

General object detection research provides the computational foundation for jewellery detection. Survey papers and comparative studies show the evolution of modern detectors like YOLOv3–YOLOv7, Faster R-CNN, and transformer-based models. These detectors extract

robust spatial features, making them suitable for detecting detailed objects such as jewellery.

However, these methods typically output bounding boxes and class labels only. They do not generate semantic interpretations (e.g., “silver bracelet with white gemstones”), demonstrating the need to extend from detection to captioning.

Table 2.2: Summary of Research Papers on Object Detection

Year	What is the paper about? (Aspects)	Methodology (Steps in 2-3 lines)	Datasets (Size, Type, etc.)	Results (Validation Metrics)	Advantages	Limitations	Reference No.
2023	Compares YOLOv5–v7; YOLOv6 excels.	Created a custom jewellery dataset, applied data augmentation, and trained multiple YOLO versions to compare their small object detection performance.	Used a custom dataset of jewellery images captured from a jewellery store with data augmentation.	YOLOv6 outperformed YOLOv5/YOLOv7 in accuracy, F1, recall, and mAP.	Targets small object detection using a real-world jewellery dataset, comparing multiple YOLO versions and highlighting YOLOv6’s superior performance.	Dataset covers only three jewellery classes, limiting generalization, and lacks full exploration of real-time deployment challenges.	[8]
2023	Proposes CNN-YOLOv7 for jewellery in smart stores.	Uses a CNN-based YOLOv7 model trained on a custom jewellery dataset for accurate detection and localization of small jewellery objects in smart store surveillance.	Used a unique dataset curated specifically for smart store surveillance focused on jewellery.	Achieved strong metrics on custom data using YOLOv7 for lightweight surveillance.	Designed for detecting small, intricate objects in surveillance, this lightweight model delivers high accuracy and real-time efficiency on custom jewellery datasets.	Primarily focused on jewellery, limiting generalization; may struggle in cluttered or low-light settings and needs a specialized dataset for training.	[9]
2022	Reviews YOLO/CNN for real-time detection.	Surveyed and analyzed YOLO algorithm versions and CNN architectures for real-time object detection and feature extraction.	No original dataset; it’s a review of YOLO and CNN models applied in literature.	Reported higher mAP and FPS, showing YOLO's real-time detection advantage.	Offers high accuracy and real-time speed with efficient CNN-based detection, enabling broad industrial applicability.	May underperform on small or overlapping objects, needs significant computational resources, and relies on high-quality training data.	[10]

2023	Proposes YOLO-based system for ring and earring detection in smart shops.	Trained and validated a YOLO-based object detector on a custom dataset of rings and earrings for real-time monitoring in smart shop surveillance systems.	Used a customized dataset containing rings and earrings images.	Achieved real-time, accurate detection with strong mAP and localization metrics.	Enables real-time jewellery monitoring in smart shops with high accuracy using a lightweight, efficient YOLO architecture.	Limited to only two jewellery classes (rings and earrings) May require retraining for different store layouts or lighting conditions	[11]
2021	Surveys DL models, datasets, and edge suitability.	Reviewed and compared deep learning-based object detection models using benchmark datasets, evaluation metrics, and backbone architectures, including lightweight models for edge deployment.	No specific dataset used; it's a survey paper reviewing existing models and benchmarks.	Compared detectors using mAP, FPS, and parameters for accuracy and efficiency.	Covers modern object detection models, benchmark datasets, and metrics, with insights on lightweight models and performance comparisons.	Lacks original experiments, relies solely on literature analysis, and may miss post-2021 advancements.	[12]
2023	Compares CNN and transformer models for object detection.	Conducted a comparative analysis of CNN and transformer architectures for object detection, focusing on design, performance, and attention mechanisms.	Literature review; no dataset used.	Provided literature-based insights without experimental metrics or validation.	Provides a thorough overview of CNN and transformer-based detectors, highlighting the shift to attention models and outlining emerging research trends.	Purely literature-based without experimental validation, lacks quantitative benchmarks, and may miss the latest transformer model developments.	[13]

Gemstone and Material Classification

Gemstone and metal recognition research explores how CNNs and hybrid architectures can classify minerals, stone colours, shine, and metal types. These works confirm that colour-space analysis, CNN feature extraction, and ensemble classifiers can identify gemstone categories with reasonable accuracy.

While relevant, these studies focus only on classification of materials or stones rather than full jewellery recognition—highlighting another gap that the proposed project addresses through integrated attribute extraction and description generation.

Table 2.3: Summary of Research Papers on Gemstone Classification

Year	What is the paper about? (Aspects)	Methodology (Steps in 2-3 lines)	Datasets (Size, Type, etc.)	Results (Validation Metrics)	Advantages	Limitations	Reference No.
2023	Proposes CNN-RF model for gemstone ID using curated images.	Used CNN for feature extraction from gemstone images and integrated a Random Forest classifier for final classification, trained on a 6265-image dataset with a 70:30 split.	Dataset of 6,265 gemstone images with 70:30 train-test split.	Achieved strong classification accuracy for effective gemstone identification.	Combines strengths of deep learning and traditional ML Works well on a moderate-sized dataset Applicable to geological and mineralogical domains	Accuracy (~74.76%) leaves room for improvement Performance may degrade on unseen gemstone types Limited dataset diversity may affect generalization	[14]
2024	Uses CNN-LSTM with LIBS to classify jewellery rocks.	Applied CNN layers for feature extraction from LIBS data and LSTM layers for sequence modeling, with interpretability analysis and Lasso feature selection.	Used laser-induced breakdown spectroscopy (LIBS) data from different jewellery rock samples.	Achieved high accuracy in classifying jewellery rocks using deep learning and LIBS.	Combines spectroscopy with interpretable deep learning High accuracy in classifying diverse jewellery rock types Provides layer-wise model interpretability	Requires specialized LIBS equipment May be limited to types of rocks studied Computationally intensive due to hybrid CNN-LSTM architecture	[15]

Image Captioning

Image captioning provides the backbone for generating natural-language descriptions of jewellery. Encoder–decoder architectures using CNN encoders (such as VGG-16 or MobileNet) and RNN/GRU/LSTM decoders can convert image features into descriptive text.

Recent works apply captioning techniques specifically to fashion items and jewellery, showing promising accuracy. Yet, most models require large datasets and often fail to capture fine-grained attributes unique to jewellery designs.

The integration of classification and captioning—particularly for jewellery—is still emerging, reinforcing the novelty of this project.

Table 2.4: Summary of Research Papers on Image Captioning

Year	What is the paper about? (Aspects)	Methodology (Steps in 2-3 lines)	Datasets (Size, Type, etc.)	Results (Validation Metrics)	Advantages	Limitations	Reference No.
2023	Explores how image captions enhance multimodal datasets for vision-language training.	Used synthetic captions from image captioning models and mixed them with raw web data, evaluating different strategies on a 128M image-text dataset.	Used large-scale web-scraped image-text datasets (128M and 1.28B pairs).	Outperformed prior filters on ImageNet, 38 tasks, Flickr, and MS-COCO.	Improves dataset quality without sacrificing diversity Boosts performance across multiple benchmarks Demonstrates scalable benefits on 1.28B image-text pairs	Synthetic captions may have limitations at very large scales Standard captioning benchmarks don't predict real training utility Image curation becomes increasingly critical with dataset size	[16]

Outcomes of Background Work:

1. Detection vs Description Gap

- Most existing studies focus on detecting or classifying jewellery but do not generate detailed natural-language descriptions.
- This gap motivates the integration of both tasks in this project.

2. Need for Fine-Grained Feature Extraction

- Research shows that jewellery contains small, delicate features.
- Deep learning models handle these well, but require domain-specific datasets.

3. Limited Jewellery-Specific Captioning Research

- Very few works attempt caption generation for jewellery.
- Existing captioning systems are general-purpose.

4. Integration Opportunity

The literature strongly supports combining the following into a single coherent:

- CNN-based feature extraction
- Jewellery classification
- Attribute recognition
- GRU-based caption generation

3. Software Project Management Plan

3.1 Introduction

This chapter provides an overview of the project structure, the expected outcomes, and the organizational approach followed during development. It outlines the key deliverables, the workflow model adopted, and the roles undertaken by team members to ensure systematic execution.

3.1.1 Project Overview

The project focuses on developing an automated system that can detect jewellery items (earrings or necklaces) from an image, extract their visual characteristics—specifically metal colour and gemstone presence—and generate meaningful natural-language descriptions.

Leveraging deep learning models such as VGG-16 for feature extraction and a GRU-based decoder for caption generation, the system addresses the limitations of existing jewellery identification processes, which are largely manual, time-consuming, and error-prone.

A user-friendly React-based interface allows users to upload jewellery images and instantly receive detailed descriptions. The solution aims to streamline catalogue creation for jewellers and e-commerce platforms by introducing automation, accuracy, and consistency.

3.1.2 Project Deliverables

The major deliverables of the project include:

1. **Jewellery Detection Model**

- A trained deep learning model using VGG-16 to classify images as earrings or necklaces.

2. **Feature Extraction Module**

- Identification of metal colour and gemstone presence using feature representation from VGG-16.

3. **Caption Generation System**

- A GRU-based decoder producing descriptive sentences such as:
  "A round necklace with emerald gemstones set in yellow gold."

4. **Dataset (Curated + Augmented)**

- A combined dataset of 32,190 images consisting of earrings and necklaces, prepared

through preprocessing and augmentation.

5. **Frontend Application (React.js)**

- User interface for uploading images and viewing generated descriptions.

6. **Backend API (Flask + PyTorch)**

- Handles model inference, file processing, and communication with the frontend.

7. **Documentation & Reports**

- Project report, literature survey, system design diagrams, implementation details, results, and references.

3.2 Project Organization

This section describes the structural arrangement of the project development process, including the software process model adopted, team responsibilities, and the tools and technologies employed.

3.2.1 Software Process Model

The project follows the Iterative and Incremental Development Model, suitable for AI and machine learning projects due to continuous experimentation and refinement.

Reasons for Selecting This Model:

- The model supports progressive dataset creation and augmentation.
- Deep learning training requires multiple iterations of tuning and validating.
- Feature extraction, caption generation, and UI integration benefit from parallel and incremental development.
- Allows incorporating feedback when accuracy improvements or UI changes are required.

Phases Followed:

1. **Requirement Analysis**

- Identified the need for automated jewellery detection & captioning.

2. **Dataset Preparation (Initial Iteration)**

- Curated raw images, applied augmentation, structured labels.

3. **Model Development (Multiple Iterations)**

- Encoder (VGG-16) training
- Decoder (GRU) tuning

- Feature extraction refinements

4. **Integration**

- Connecting backend APIs with the frontend UI.

5. **Evaluation & Improvement**

- Achieved 94.62% jewellery classification accuracy and a description accuracy of 90.44% after several optimization cycles.

6. **Final Deployment & Documentation**

- Complete pipeline delivered with UI + backend + dataset.

3.2.2 Roles and Responsibilities

The responsibilities were divided among team members to ensure smooth execution:

1. Data Collection & Preprocessing Team

- Gathered datasets from Hugging Face and additional sources.
- Performed augmentations such as rotations, flips, brightness adjustments.
- Created the final dataset of **32,190 images**.

2. Model Development Team

- Implemented **VGG-16 encoder** for feature extraction.
- Built **GRU decoder** for caption generation.
- Trained and validated the detection and captioning models.

3. Backend Development Team

- Built API endpoints using **Flask**.
- Handled image uploads, predictions, and JSON responses.
- Integrated PyTorch inference pipeline.

4. Frontend Development Team

- Designed **React** interface.
- Implemented upload form and results display panels.

5. Documentation & Testing Team

- Prepared project report chapters, diagrams, and result summaries.
- Conducted UI testing, model accuracy testing, and validation.

3.2.3 Tools and Techniques

The project employs a combination of deep learning frameworks, frontend technologies, and backend tools as described below. (Based on PPT Technologies section)

1. Development Tools

- Python 3.10 – Core language for model development
- PyTorch 2.0 – Used for training VGG-16 + GRU model
- Torchvision 0.15 – Pretrained model utilities

2. Frontend Technologies

- React.js – UI for image upload and result visualization
- HTML/CSS/JavaScript – UI styling and interaction

3. Backend Technologies

- Flask – API development
- CORS – Communication between frontend and backend
- PIL (Pillow) – Image handling
- UUID, OS Libraries – File handling and storage

4. Deep Learning Techniques

- Transfer Learning (VGG-16 Encoder) – For jewellery detection
- GRU Decoder – For caption generation
- Image Preprocessing – Resize (224×224), normalization, tensor conversion
- Augmentation – Rotation, brightness change, flips, etc.

5. Dataset Management Tools

- CSV Annotations
- Custom image folder organization
- Programmatic train/validation split (80/20)

3.3 Project Management Plan

3.3.1 Tasks

3.3.1.1 Description

The project is structured into the following major tasks:

1. **Dataset Preparation:**
   Collect datasets from Hugging Face, clean and augment them, and apply preprocessing for model training.
2. **Jewellery Detection & Feature Extraction:**
   Use the VGG-16 encoder to detect earrings/necklaces and extract visual features such as metal colour and gemstone presence.
3. **Classification & Caption Generation:**
   Implement the GRU-based decoder to classify the jewellery type and generate descriptive captions.
4. **Frontend Development:**
   Build a React-based interface for image upload and visualization of results.
5. **Backend Development:**
   Develop server logic to handle model execution and integrate frontend with the ML pipeline.
6. **Testing & Evaluation:**
   Validate detection accuracy and evaluate generated descriptions.
7. **Documentation & Final Presentation:**
   Prepare the final synopsis, report, and demonstration materials.

3.3.1.2 Deliverables and Milestones

- Dataset completed and augmented
- Preprocessing pipeline operational
- VGG-16–based detection model implemented
- GRU caption generator trained
- Frontend and backend integrated
- Working prototype with sample outputs
- Testing results and performance metrics completed
- Final synopsis + presentation + report prepared

3.3.1.3 Resources Needed

- Hardware: GPU-enabled system (recommended), local development machines

- Software: PyTorch, Torchvision, Python, React, VS Code
- Datasets: Jewelry Vision dataset + caption dataset from Hugging Face

3.3.1.4 Dependencies and Constraints

Dependencies:

- Availability of pretrained VGG-16 model
- Access to Hugging Face datasets
- Stable environment for frontend/backend integration

Constraints:

- Limited caption dataset
- Visual similarity between jewellery types
- Real-time response requirement (1–2 seconds)
- GPU dependency for efficient training

3.3.1.5 Risk and Contingencies

- **Dataset Imbalance:**

  Mitigation: Use augmentation and balancing strategies.
- **Overfitting in Caption Generation:**

  Mitigation: Apply regularization and expand training data.
- **Incorrect Feature Detection:**

  Mitigation: Improve preprocessing and colour extraction.
- **Integration Issues:**

  Mitigation: Conduct early and iterative integration testing.
- **Performance Bottlenecks:**

  Mitigation: Optimize model loading and API calls.

3.3.2 Time table

Task	Timeline
Problem Statement Refinement & Literature Review	July 14 – July 31
Dataset Collection & Curation	August 1 – August 15
Image Preprocessing	August 16 – August 31
Jewellery Detection and Classification Model Training	September 1 – September 30
Description Generation Model Training	October 1 – October 15
Frontend Integration	October 16 – October 31
Backend Integration & Testing	November 1 – November 20
Deployment on Vercel & Documentation	November 21 – December 5

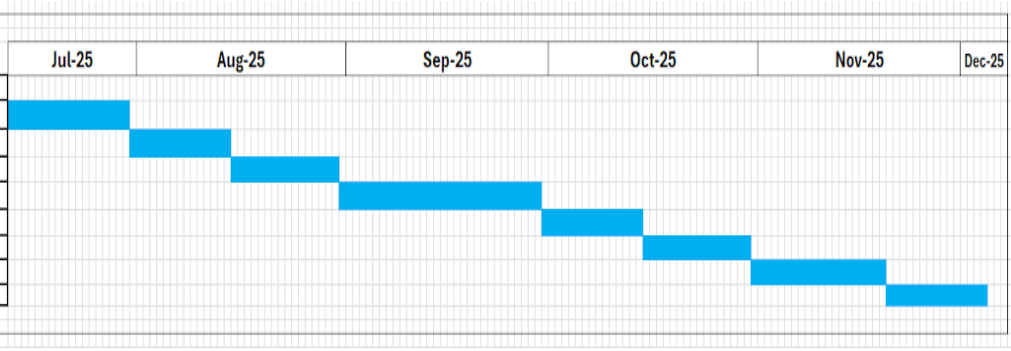

Figure 3.1: Project Implementation Timeline and Schedule (Gantt Chart)

4. Software Requirements Specification

4.1 Introduction

The Software Requirements Specification (SRS) defines the complete functional and non-functional requirements of the system "Automated Description Generation for Jewellery Images using Deep Learning."

This document describes how the system processes an input image containing a jewellery item, detects whether it is an earring or a necklace, extracts key visual features such as metal colour and gemstone presence, and generates a natural-language description using an encoder–decoder deep learning architecture.

The SRS ensures a clear understanding between developers, testers, and users by specifying system behaviour, constraints, performance expectations, and interfaces. It serves as a foundation for design, development, testing, and validation activities.

This chapter outlines the overall product perspective, the system's major capabilities, and the operating environment in which the software functions.

4.1.1 Product Overview

The product is an AI-powered jewellery detection and captioning system that automates the task of identifying jewellery items and generating detailed descriptions. The system consists of three major components:

1. Jewellery Detection Module
2. Feature Extraction & Caption Generation Module
3. User Interaction Interface

1. Jewellery Detection Module

This module identifies whether the uploaded image contains an earring or necklace using a fine-tuned VGG-16 model. It analyzes visual structure, contours, and textures to classify the jewellery type with 94.62% accuracy.

2. Feature Extraction and Caption Generation Module

Once classified, the system extracts visual attributes such as:

- Metal colour (e.g., yellow gold, white silver)
- Presence and colour of gemstones

- Shape-based cues when available

These features are fed into a GRU-based decoder, which produces a natural-language caption (e.g., “A yellow-gold necklace decorated with green gemstones.”).

3. User Interface (Frontend)

A simple React-based interface allows users to upload images and receive results. The UI displays:

- Jewellery type (earring/necklace)
- Generated caption
- Classification details extracted by the model

Product Perspective

The system functions as an integrated AI solution combining:

- Computer vision for feature extraction
- Deep learning for classification
- Natural language generation for descriptions
- A web interface for user interaction

Product Functions (High-Level)

- Accept user-uploaded jewellery images
- Detect jewellery category
- Extract relevant visual cues
- Generate descriptive captions
- Display results in a structured interface

User Characteristics

Intended users include:

- Jewellers
- E-commerce sellers
- Designers documenting inventory
- Research and archival teams

Users require no technical knowledge, as the system is fully automated and user-friendly.

General Constraints

- Requires images with clear visibility of jewellery
- Deep learning inference dependent on GPU/CPU performance

- Backend must process uploaded images securely

Assumptions

- The jewellery item in the input image is clearly visible
- Only one primary jewellery item is present per image

4.2 Specific Requirements

The system processes an uploaded image, detects jewellery (necklace/earrings), extracts visual features, and generates a descriptive caption using deep learning models.

4.2.1 External Interface Requirements

4.2.1.1 User Interfaces

- A simple React-based web interface for image upload.
- Display area for:
    - Uploaded image
    - Detected jewellery item (earring/necklace)
    - Generated caption
- Buttons for Upload, Process, and Retry.
- Clean and intuitive layout suited for jewellers and catalog creators.

4.2.1.2 Hardware Interfaces

- Works on any standard laptop/desktop.
- GPU-enabled machine required only during model training, not during inference.
- No special external devices needed.

4.2.1.3 Software Interfaces

- Frontend: React
- Backend: Python (Flask/FastAPI/Django REST)
- Model Framework:
    - PyTorch 2.0+
    - Torchvision (VGG-16 pretrained model)

- Model Components:
    - Encoder: VGG-16 for jewellery detection + feature extraction
    - Decoder: GRU for caption generation
- APIs:
    - REST APIs for processing images and returning results
    - Communication using JSON data format

4.2.1.4 Communications Protocols

- HTTP/HTTPS for communication between frontend and backend
- REST API for inference requests (image upload, output retrieval)
- multipart/form-data for sending images
- JSON for sending processed results (type + caption)

4.2.2 Software Product Features

1. **Jewellery Detection**

   Identifies whether the uploaded image contains a necklace or earrings.
2. **Jewellery Classification**

   Labels the detected item (earring/necklace).
3. **Visual Feature Extraction**

   Identifies metal colour (gold/silver), gemstone presence and colour.
4. **Caption Generation**

   Produces descriptive text such as:

   "A pair of gold earrings with emerald green gemstones."
5. **User Interaction**

   Simple upload-and-view process for non-technical users.
6. **Fast Processing**

   Generates results within 1–2 seconds.

4.2.3 Software System Attributes

- Performance:
    - Jewellery detection accuracy = 94.62%

    - Caption generation within 2 seconds
    - Description generated accuracy = 90.44%
- Usability:
    - Minimal design, easy to operate for all users
- Reliability:
    - Works for varying lighting/backgrounds
    - Handles incorrect input gracefully
- Portability:
    - Can run on any modern browser
    - Backend deployable on Windows/Linux systems
- Security:
    - Safe handling of user-uploaded images
    - No image storage unless explicitly required

4.2.4 Database Requirements

No database is used in this project.

The system processes images in real time and returns results directly to the user interface.

Thus:

- No storage of images
- No storage of captions
- No storage of metadata
- No need for SQL/NoSQL systems

The system operates entirely using in-memory processing and on-the-fly inference, making it lightweight and suitable for prototype deployment.

5. Software Design Description

5.1 Introduction

The design of the Automated Jewellery Description system emphasizes modularity, accuracy, and ease of use. The AI pipeline utilizes deep learning models—specifically a VGG-16 encoder for feature extraction from jewellery images and a GRU-based decoder for generating natural language captions. The architecture supports reliable jewellery type classification and detailed descriptive outputs for cataloging and e-commerce applications.

5.1.1 Design Overview

The system integrates three main modules:

- Jewellery Detection: Uses a VGG-16 model to classify images as earrings or necklaces.
- Feature Extraction & Caption Generation: Extracts visual features such as metal colour and gemstone presence, then generates a descriptive caption using the GRU decoder.
- User Interface: A React-based frontend for image upload and results display.

A Flask backend links the frontend to the model inference pipeline via REST APIs, maintaining smooth communication between components.

5.1.2 Requirement Traceability Matrix

The key requirements are traced as follows:

- Upload jewellery image → User Interface (React) and Backend API
- Detect jewellery type (earring/necklace) → VGG-16 Encoder
- Extract features (metal colour, gemstone) → Encoder feature extraction
- Generate caption → GRU Decoder
- Present results → UI display
- Usability for non-technical users → Simple UI with upload and result visualization

5.2 System Architectural Design

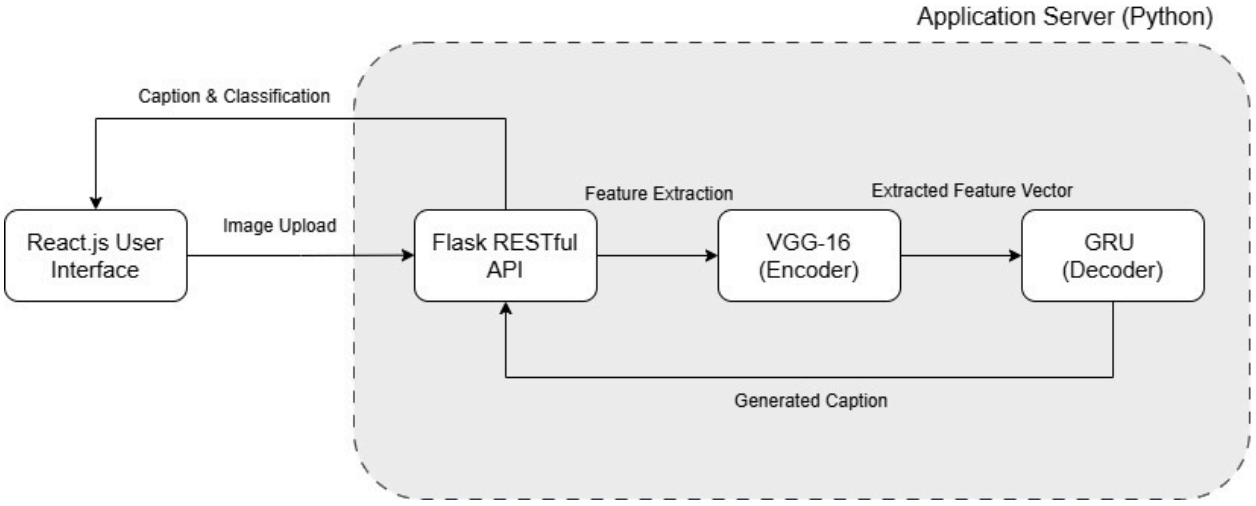

Figure 5.1: System Architecture Design

5.2.1 Chosen System Architecture

The system follows an encoder-decoder architecture:

Frontend: React.js user interface for uploading images and displaying results.

Backend: Flask-based RESTful API handling image uploads and returning predictions; runs the PyTorch models.

Model: VGG-16 encoder extracts features, followed by a GRU decoder which generates captions.

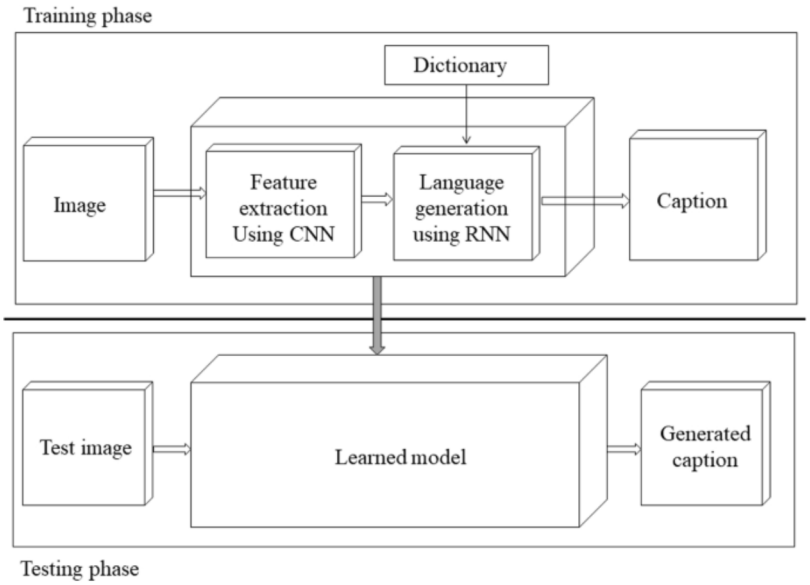

Figure 5.2: Training and Testing Workflow of the Proposed Model

Data Flow: User uploads image → frontend sends to backend → backend processes via models → caption and classification returned → frontend displays output.

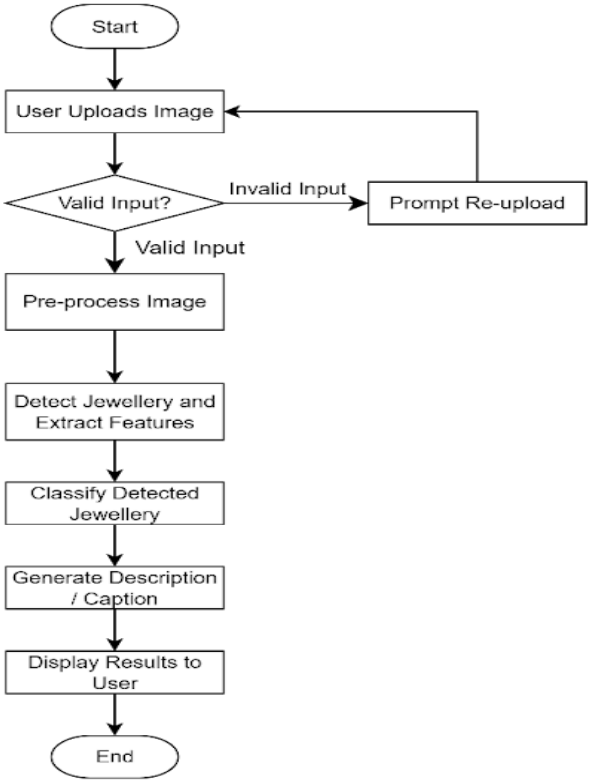

Figure 5.3: Data Flow for Caption Generation

5.2.2 Discussion of Alternative Designs

To determine the optimal architecture for jewelry classification and captioning, we conducted a chronological investigation of several models and training strategies. The evolution of our design choice is detailed below:

- **Initial Feature Extraction Exploration (MobileNet & Faster R-CNN):**
    - **Approach:** We evaluated MobileNet for its lightweight efficiency and Faster R-CNN for its object detection capabilities.
    - **Observation:** While effective, Faster R-CNN was computationally heavy for our specific classification needs, and MobileNet lacked the depth required for fine-grained jewelry feature extraction.
- **Advanced Captioning Attempt 1 (ResNet-101 + Bahdanau Attention):**
    - **Approach:** We implemented a state-of-the-art architecture using a ResNet-101

backbone coupled with Bahdanau Attention. We utilized Teacher Forcing during training to accelerate convergence.

- **Observation:** While the model captured details well, the heavy reliance on Teacher Forcing led to "Exposure Bias," causing the model to struggle with repetition loops during inference (e.g., repeating "a pair of...").

- **Refining Training Strategy (Scheduled Sampling):**
  - **Approach:** To fix the repetition issues found in the Attention model, we shifted from pure Teacher Forcing to Scheduled Sampling. This involved gradually introducing the model's own predictions during training to teach it error recovery.
  - **Observation:** This improved coherence but added significant training complexity and computational overhead.
- **Stability Comparison (ResNet-50 + LSTM):**
  - **Approach:** We implemented a standard ResNet-50 + LSTM encoder-decoder as a stability baseline to compare against the attention-based models.
  - **Observation:** This provided stable results but lacked the specific spatial interpretability we sought for the jewelry dataset.
- **Final Selection (VGG-16 + GRU):**
- **Approach:** We ultimately selected the VGG-16 (Encoder) + GRU (Decoder) architecture.
- **Rationale:** This hybrid CNN-RNN architecture offered the best trade-off. The VGG-16 encoder provided rich feature maps that the GRU could effectively sequence for both classification and captioning. It demonstrated optimal accuracy (~90%), faster inference speed, and superior resource efficiency compared to the heavier ResNet-Attention models.

5.2.3 System Interface Description

The system interfaces include:

- REST APIs using HTTP/HTTPS for communication between frontend and backend
- Image uploads via multipart/form-data
- JSON format for transmitting classification and caption data
- Backend loads PyTorch models for inference processing and sends results back to the UI.

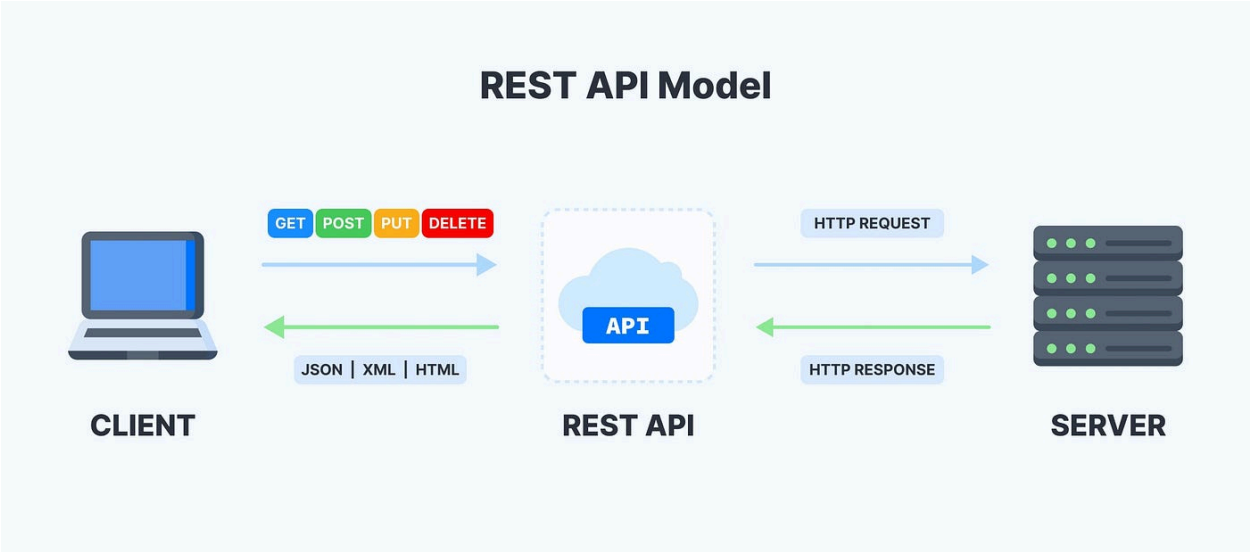

Figure 5.4: Client-Server Communication via REST API

5.3 Detailed Description of Components

Dataset Management: Curated and augmented dataset organized with CSV annotations and train-validation splits.

Image Preprocessing: Resizing to 224×224 pixels, normalization, tensor conversion, and data augmentation like rotation and brightness adjustments.

VGG-16 Encoder: Fine-tuned to extract jewellery-specific features such as shapes, metal colours, and gemstone presence.

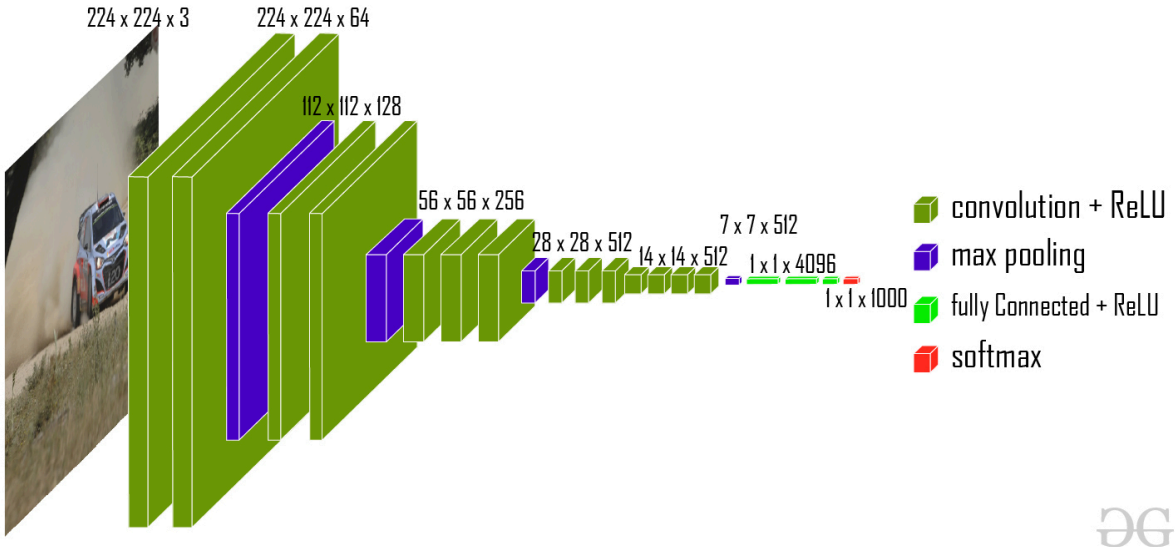

Figure 5.5: VGG-16 Architecture Used for Feature Extraction

GRU Decoder: Generates descriptive, natural language captions from encoder outputs.

User Interface Module: React.jsbased interface with an upload panel and result display sections optimized for ease of use.

Backend API: Flask-based server for safe, efficient handling of image uploads, model execution, and response delivery.

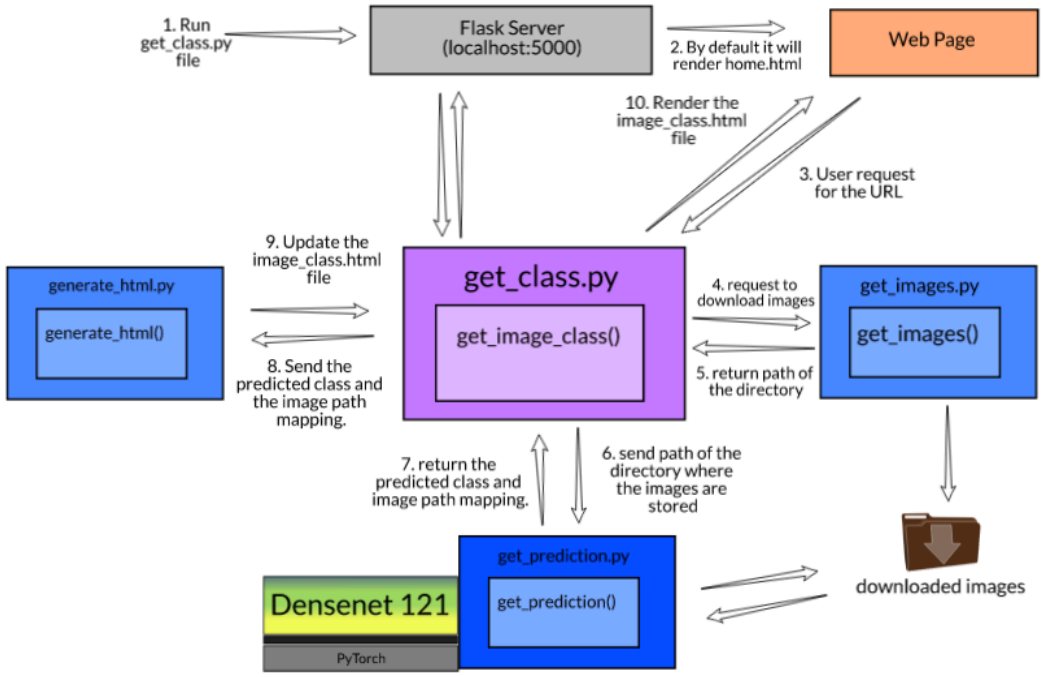

Figure 5.6: Backend Component Interaction and Data Flow

5.4 User Interface Design

5.4.1 Description of User Interface

- A simple, clean React frontend allows users to upload a single jewellery image at a time.
- Displays classification results (earring or necklace) and the corresponding descriptive caption.
- Provides buttons to Upload, Process, and Retry image submissions.
- Layout is intuitive and designed to be used easily by non-technical audiences such as jewellers and catalog managers.

5.4.1.1 Screen Images

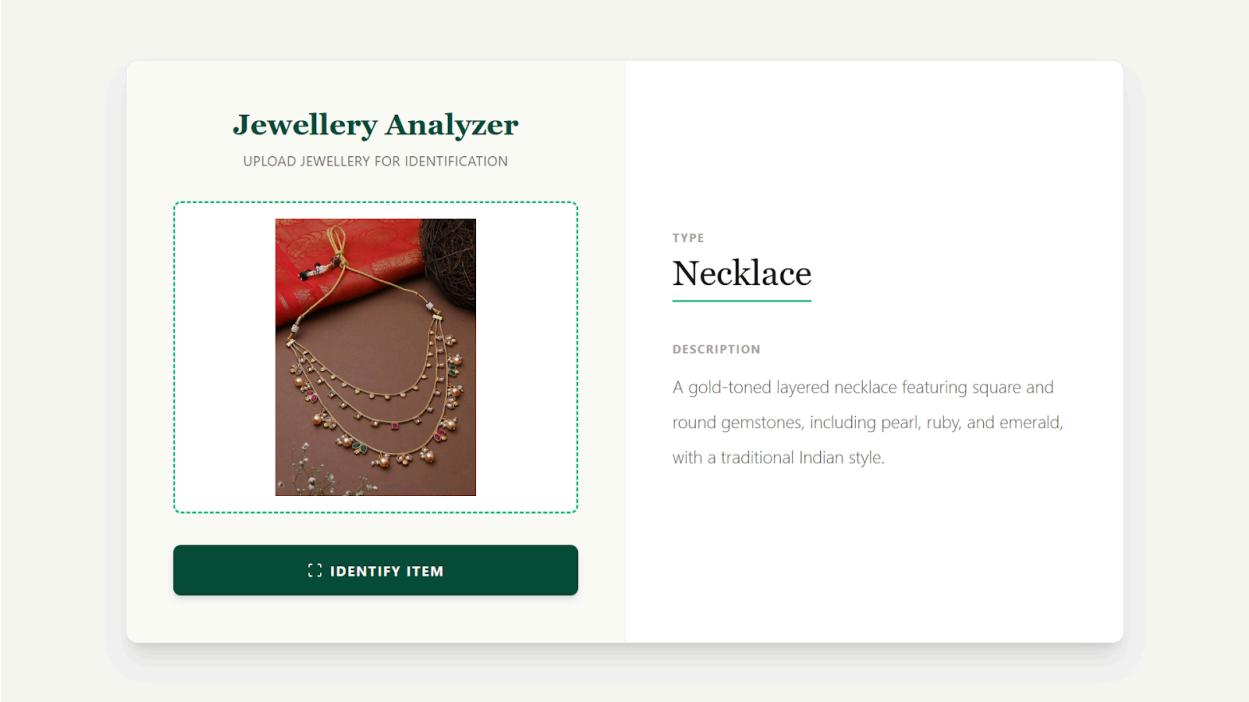

Figure 5.7: Classification and Description Result for Necklace Category

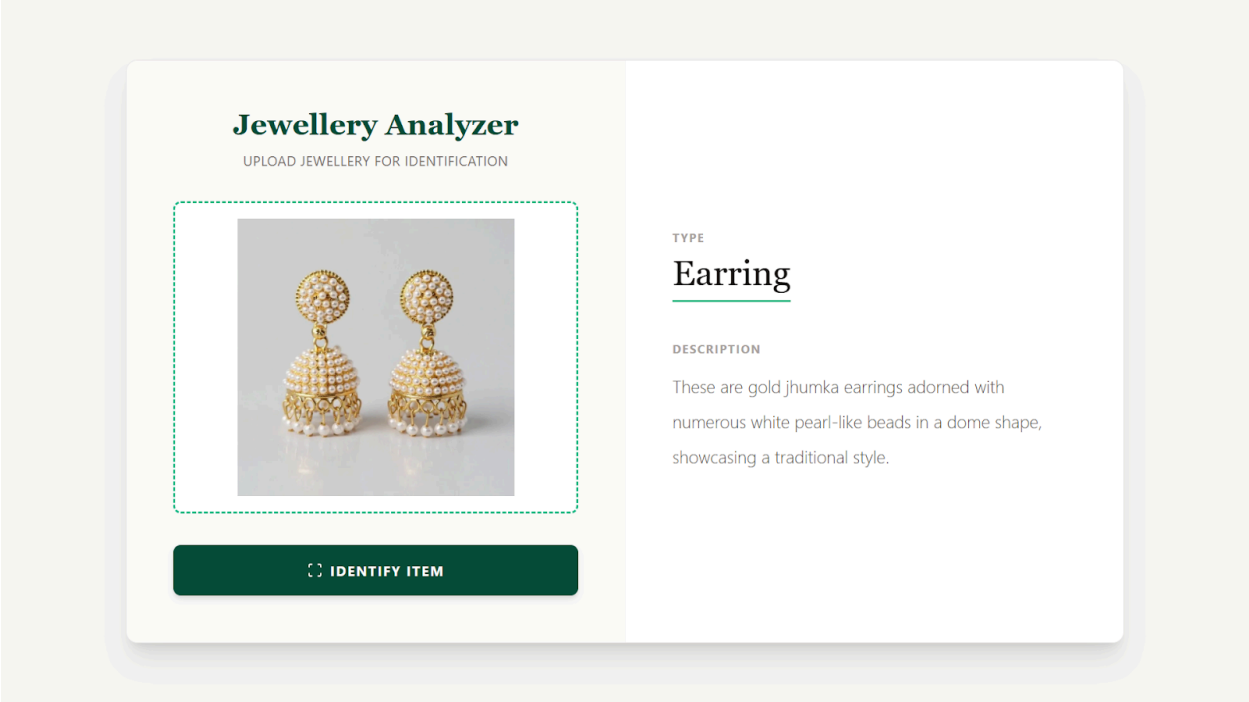

Figure 5.8: Classification and Description Result for Earring Category

6. Software Test Document

6.1 Introduction

This chapter describes the testing strategy and approach used to verify the correctness, performance, and reliability of the automated jewellery identification and description system. Testing ensures the system meets functional requirements such as accurate detection and caption generation while maintaining usability and stability.

6.1.1 System Overview

The system consists of a React-based frontend for image upload and display, a Flask backend that serves REST APIs for model inference, and deep learning models (VGG-16 encoder and GRU decoder) for jewellery detection and caption generation. Testing covers the entire pipeline from image upload to output display.

6.1.2 Test Approach

The test approach combines functional testing of key features with performance and usability validation. Tests include unit tests for backend APIs, integration tests for frontend-backend communication, and system-level tests that evaluate output accuracy against labeled data. Automated scripts and manual testing are both used.

6.2 Testing Details

6.2.1 Features to be Tested

- Jewellery image upload functionality via the React UI.
- Correct classification of jewellery type (earring or necklace) by the VGG-16 model.
- Accurate extraction of visual attributes such as metal colour and gemstone presence.
- Generation of meaningful natural-language captions by the GRU decoder.
- API responses return correctly structured JSON data.
- UI correctly displays classification results and captions.
- Performance targets including response time within 1–2 seconds per request.

6.2.2 Features Not to be Tested

- Database interactions, as the system does not use persistent storage.
- Non-jewellery images or images with multiple jewellery items, as the system assumes one main jewellery item per image.
- Detailed testing of React UI styling and responsiveness beyond core upload/result functionality.

6.2.3 Testing Tools and Environment

- Backend: Python unittest framework and Postman for API testing.
- Frontend: Manual testing in various modern browsers (Chrome, Firefox).
- Hardware: GPU-enabled machine for model training; testing mostly done on CPU-enabled systems for inference.
- Environment: Local development machines with Python 3.10, PyTorch 2.0, Flask server, React 18.

6.3 Test Cases

- Upload a valid jewellery image and verify correct classification (earring/necklace).
- Upload an image with known metal and gemstone attributes and verify generated caption accuracy.
- Test REST API endpoints for correct response codes and JSON format under normal and error conditions.
- Validate UI displays correct results after image processing.
- Measure API response time to ensure compliance with performance goals.

Table 6.1: Test Execution Report

TestID	Test Case	Description	Input	Expected Output	Actual Output	Status
TC01	Valid Image Upload (single)	Upload valid jewellery image	Jewellery image	UI shows preview, type & caption	Preview + type + caption	Pass
TC02	Invalid File Type	Upload wrong file format	PDF/TXT/CSV etc.	UI error message	Error message	Pass
TC03	Empty Upload	No file uploaded	Empty request	Clear error message	Error message	Pass
TCO4	Response Time	Check processing time	Valid image	≤ 10 seconds	Within limit	Pass
TC05	Multilingual Captions (e.g., Hindi)	Generates captions in Hindi and other languages.	Gold ring with red stone	सोने की अंगूठी लाल पत्थर के साथ		
TC06	Mobile Live Camera Mode	Real-time detection using mobile camera.	Live camera pointed at a silver necklace	Detected: Necklace Description: Silver necklace with pendant		
TC07	User Feedback Loop (corrections)	Users correct results to improve system.	System predicts: Earring User corrects: Pendant	Label updated to *Pendant* and stored for retraining		
TC08	Visual Search / Similar Items	Finds visually similar jewelry items.	Image of a diamond ring	visually similar diamond rings from database		

7. Conclusion

The project successfully developed an automated system for jewellery identification and descriptive caption generation using deep learning techniques. The system combines a VGG-16 encoder for detailed feature extraction with a GRU-based decoder to generate natural-language descriptions of earrings and necklaces. It achieves a high jewellery detection accuracy of 94.62%, demonstrating robustness in classifying jewellery types and a jewellery description generation accuracy of 90.44%.

The React-based user interface provides a simple and effective way for users, especially jewellers and e-commerce catalog managers, to upload images and receive detailed, structured descriptions quickly. Backend integration through Flask ensures smooth communication and real-time processing with low response times (1–2 seconds).

Future enhancements could involve expanding the model to detect finer-grained attributes, supporting multilingual caption generation, and deploying the system for real-time use in e-commerce and digital cataloging applications.

References

1. S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, “A Survey of Modern Deep Learning based Object Detection Models,” arXiv preprint, arXiv:2104.11892v2, May 2021.
2. M. Mafaz, “Identification of Jewellery Article using Transfer Learning and Image Repository,” International Journal of Advance Research, Ideas and Innovations in Technology (IJARIIT), vol. 7, no. 4, 2021.
3. V. Viswanatha, R. K. Chandana, and A. C. Ramachandra, “Real Time Object Detection System with YOLO and CNN Models: A Review,” Journal of Xi'an University of Architecture & Technology, vol. 12, no. 4, 2020.
4. L. Yang, “Investigation of You Only Look Once Networks for Vision-based Small Object Detection,” International Journal of Advanced Computer Science and Applications (IJACSA), vol. 14, no. 4, 2023.
5. W. Ni, “Implementation of a CNN-based Object Detection Approach for Smart Surveillance Applications,” IJACSA, vol. 14, no. 12, 2023.
6. A. Murthy, P. Devadiga, and N. P. S., “FASTER R-CNN Approach for Detecting Smaller Jewels in Yakshagana Image,” International Research Journal of Modernization in Engineering, Technology and Science (IRJMETS), vol. 5, no. 6, Jun. 2023.
7. W. Xu and Y. Zhai, “A YOLO-based Object Monitoring Approach for Smart Shops Surveillance System,” Journal of Optics, vol. 53, pp. 3163–3170, 2024.
8. Y. Thwe, N. Jongsawat, and A. Tungkasthan, “Accurate Fashion and Accessories Detection for Mobile Application based on Deep Learning,” International Journal of Electrical and Computer Engineering (IJECE), vol. 13, no. 4, pp. 4347–4356, Aug. 2023.
9. T. Nguyen, S. Oh, S. Y. Gadre, G. Ilharco, and L. Schmidt, “Improving Multimodal Datasets with Image Captioning,” in Proc. 37th Conf. Neural Information Processing Systems (NeurIPS), 2023.
10. S. Shah and J. Tembhurne, “Object detection using convolutional neural networks and transformer-based models: a review,” Journal of Electrical Systems and Information Technology, vol. 10, no. 1, 2023.
11. A. M. Shoib, S. Jabeen, C. Wang, and A. Tassawar, “Content-based Jewellery Item Retrieval using the Local Region-based Histograms,” arXiv preprint,

arXiv:2305.07540v1, May 2023.

12. Yashu, V. Kukreja, K. Madan, A. Singh, and D. Kumar, “GemID: A Hybrid CNN-Random Forest Approach for Accurate Gemstone Identification,” in Proc. 3rd Int. Conf. Smart Generation Computing, Communication and Networking (SMART GENCON), Dec. 2023, pp. 1–6.
13. J. M. Alcalde-Llergo, E. Yeguas-Bolívar, and A. Fuerte-Jurado, “Jewellery Recognition via Encoder-Decoder Models,” arXiv preprint, arXiv:2401.08003v1, Jan. 2024.
14. A. Thanakrirkphon and S. Mruetusatorn, “Detection and Tracking Shape-Shifting Gold Necklaces Using Computer Vision Techniques,” in Proc. IEEE Conf., 2023.
15. P. Khalilian et al., “Jewellery rock discrimination as interpretable data using laser-induced breakdown spectroscopy and a convolutional LSTM deep learning algorithm,” Scientific Reports, vol. 14, Art. no. 5169, 2024.
16. J. M. Alcalde-Llergo et al., “Automatic Identification and Description of Jewellery Through Computer Vision and Neural Networks for Translators and Interpreters,” Applied Sciences, vol. 15, no. 10, Art. no. 5538, 2025.