Industrial Internship Report on

”Project Name”

Prepared by

[Student name]

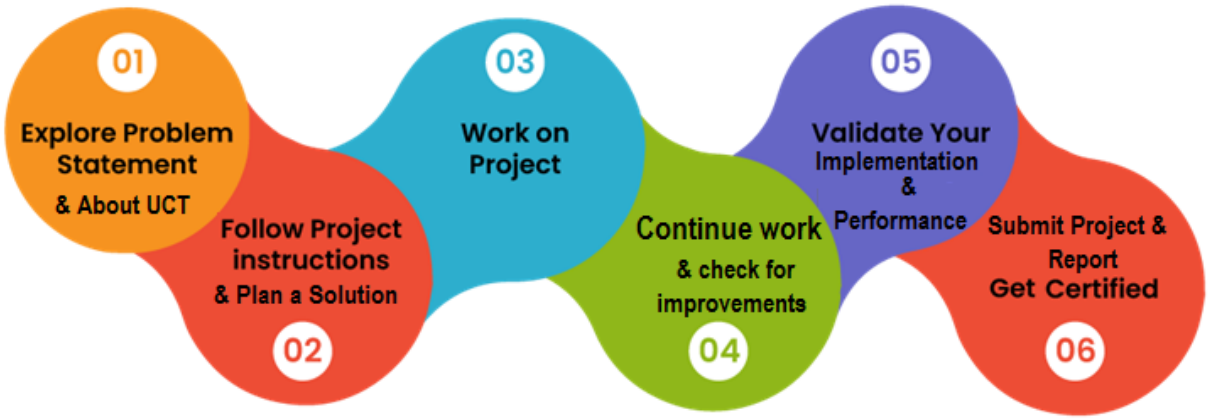
Executive Summary
This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT). This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks’ time. My project was “Multi-Class Animal Recognition using Deep Learning.” The objective of this project was to develop a system that can automatically identify animal species from images. I trained a deep learning model using the MobileNetV2 architecture with transfer learning on a dataset containing 90 different animal classes. The model was trained, evaluated, and optimized to achieve good accuracy and was then deployed using a Streamlit web application. In the application, users can upload an image of an animal and instantly receive predictions along with a confidence score. This project gave me practical experience in handling datasets, training and evaluating deep learning models, and deploying them as real-time applications for end users. This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

1 Preface

Over the span of six weeks, I had the opportunity to work on an industrial internship provided by Upskill Campus and The IoT Academy in collaboration with UniConverge Technologies Pvt. Ltd. (UCT). This internship was structured to simulate real industry problem-solving, where we were given a project/problem statement to complete within the specified time frame, including the implementation and report submission.

The internship proved to be a relevant step in my career development journey. As I aim to pursue a career path in Data Analytics, gaining practical exposure to Deep Learning and AI-driven applications has been extremely beneficial. The skills I learned—such as data preprocessing, model training, evaluation, and deployment—are highly transferable to analytics, machine learning, and real-world data problem solving. This experience has therefore strengthened both my technical expertise and my confidence in handling complex projects.



My project was “Multi-Class Animal Recognition using Deep Learning.” The project focused on building an AI-based system capable of identifying animal species from images. I implemented the solution using MobileNetV2 (transfer learning) trained on a dataset of 90 animal categories. The model was evaluated and then deployed using a Streamlit application, which allows users to upload animal images and receive instant predictions with confidence scores. This project gave me end-to-end exposure to working with datasets, training models, and deploying them in user-friendly platforms.

The opportunity given by Upskill Campus and UCT was well-structured, ensuring a balance between learning new concepts and applying them practically. The program was carefully planned: the initial phase focused on understanding the problem statement and required tools, followed by model development and training in the intermediate weeks, and finally integration, testing, and report preparation towards the end. This structured planning made the learning curve smooth and efficient.

Through this internship, my key learnings included:

- Understanding the complete workflow of deep learning projects, from data handling to deployment.
- Gaining confidence in applying machine learning frameworks like TensorFlow/Keras.
- Learning the importance of planning, iteration, and evaluation in real projects.
- Realizing the industrial relevance of AI/ML solutions in solving complex classification problems.

Overall, it was a great experience that not only enhanced my technical skills but also helped me develop problem-solving and project management abilities in an industrial context.

My message to juniors and peers: Internships like this are a golden opportunity to gain hands-on exposure. Take every project as a chance to explore, experiment, and learn. Don’t focus only on the end results—focus on the skills you build along the way. The experience you gain here will definitely help you in your higher studies, career choices, and industry readiness.

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.

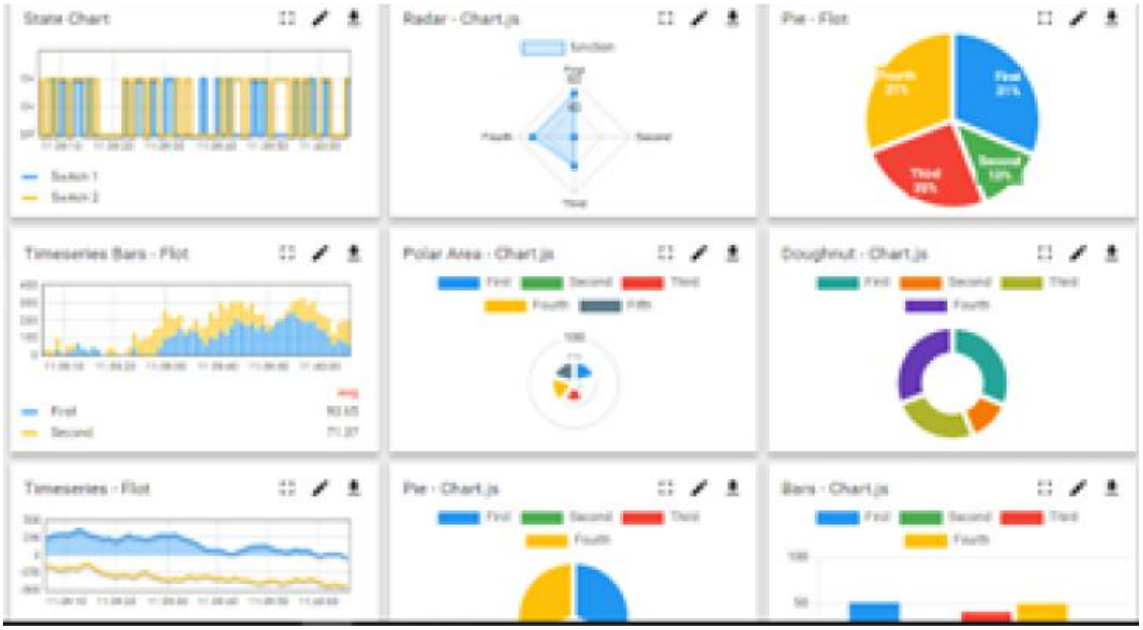
i. UCT IoT Platform (UCT Insight)

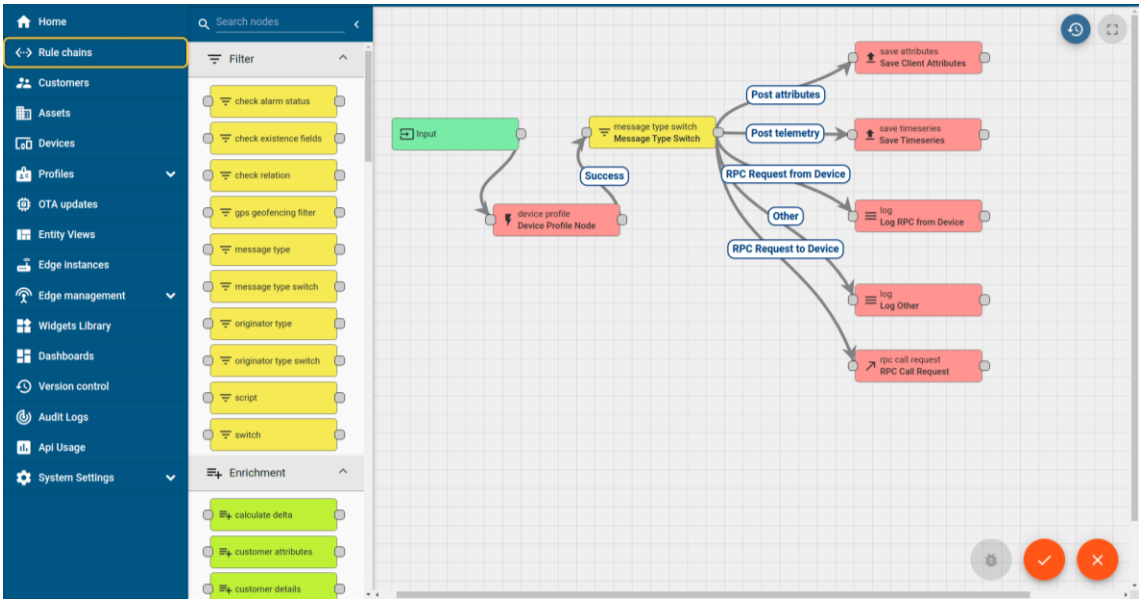
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine





ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

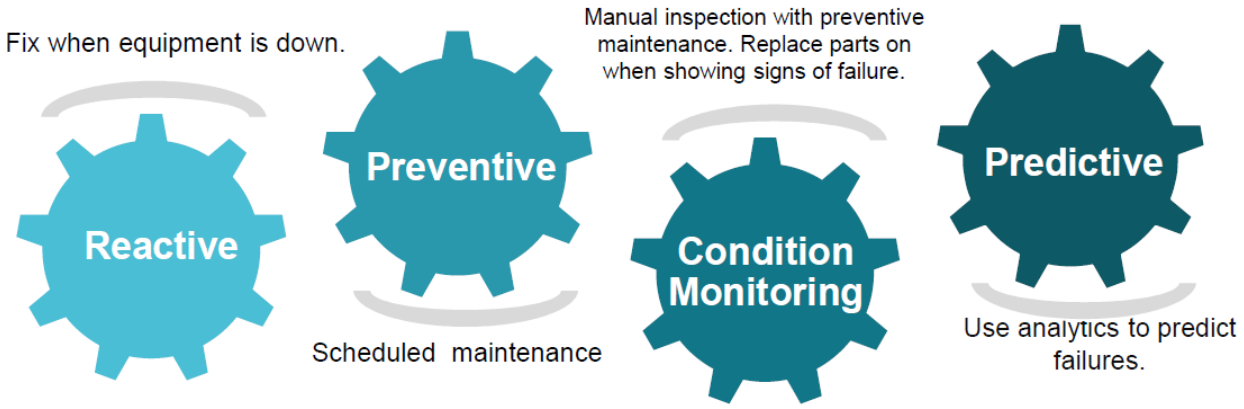
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i

iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

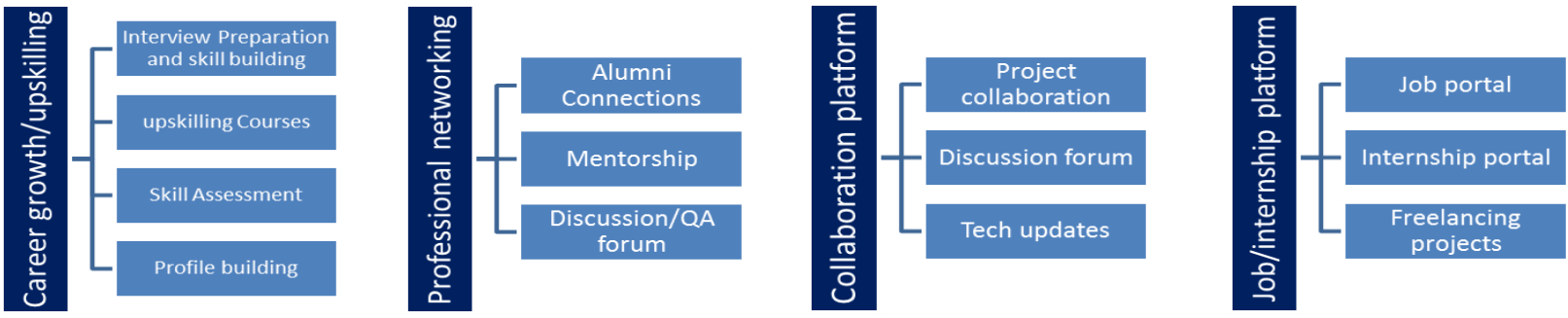
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

- ☛ get practical experience of working in the industry.
- ☛ to solve real world problems.
- ☛ to have improved job prospects.
- ☛ to have Improved understanding of our field and its applications.
- ☛ to have Personal growth like better communication and problem solving.

2.5 Reference

[1] Practical Deep Learning: Classifying Animal Images with a Fine-Tuned Pre-Trained Model

[2] Deep-Learning Apps for Image Processing Made Easy – A Step-by-Step Guide

[3] MobileNetV2: Inverted Residuals and Linear Bottlenecks (ArXiv Paper)

2.6 Glossary

Term	Acronym
Transfer Learning	TL
MobileNet Version 2	MobileNetV2
Convolutional Neural Network	CNN
Image Classification	IC
Streamlit Web Application	SWA
Deep Learning	DL

3 Problem Statement

The assigned problem statement was to design and implement an **AI-based system for Multi-Class Animal Recognition.** The objective was to build a robust model capable of identifying different animal species from images with high accuracy. The challenge lay in handling a dataset containing **90 animal categories**, ensuring proper preprocessing, training an efficient model, and deploying it in a user-friendly application.

The system needed to:

- Accurately classify animals into their respective classes.
- Handle variations in image quality, orientation, and background.
- Be deployed as a **real-time interactive application** where users can upload an image and instantly receive the prediction with confidence scores.

This problem statement reflects real-world industrial use cases of **deep learning in image recognition**, providing practical exposure to solving end-to-end AI problems from dataset handling to deployment.

4 Existing and Proposed solution

Existing Solutions:
Several existing animal recognition systems and research projects make use of traditional Convolutional Neural Networks (CNNs) or older transfer learning models like VGG16 and ResNet. While these solutions demonstrate good accuracy, they often face **limitations such as high computational requirements, slower inference times, and difficulty in scaling to a large number of classes**. Many of them are restricted to a smaller dataset (limited animal categories) and are not deployed in real-time user applications, limiting their practical usability.

Proposed Solution:
Our proposed solution uses **MobileNetV2**, a lightweight and efficient deep learning architecture designed for image classification. By applying **transfer learning**, the model was fine-tuned on a dataset containing **90 animal species**, making it both accurate and computationally efficient. The trained model was then integrated into a **Streamlit-based web application** where users can upload images and receive instant predictions along with confidence scores.

Value Addition:

- Use of **MobileNetV2** ensures faster predictions with reduced computational cost.
- Covers a **wider range of 90 animal classes**, improving applicability.
- Provides a **user-friendly web interface** for real-time interaction.
- Bridges the gap between academic research and **industrial deployment** by focusing on usability, scalability, and accessibility.

4.1 Code submission (Github link)

https://github.com/vnr-nitish/upskill-campus/blob/main/Multi%20Class%20Animal%20Recognition.py

4.2 Report submission (Github link) :

https://github.com/vnr-nitish/upskill-campus/blob/main/MultiClassAnimalRecognition_Nitish_USC_UCT.pdf

5 Proposed Design/ Model

The design of the proposed solution follows a **systematic workflow**, beginning from dataset preparation and ending with a real-time deployed application. The design flow can be explained in the following stages:

1. Data Collection & Preprocessing (Start Stage):

- The dataset of **90 different animal species** was obtained from Kaggle.
- Images were resized to **224×224 pixels** to match the MobileNetV2 input size.
- Applied **data augmentation** techniques (rotation, flipping, zooming, rescaling) to improve generalization and reduce overfitting.
- Class labels were extracted and stored separately for use during predictions.

2. Model Selection & Training (Intermediate Stage):

- Chose **MobileNetV2** as the base model because of its efficiency and accuracy.
- Added custom layers: **Global Average Pooling → Dense → Dropout → Dense (Softmax)** to adapt to multi-class classification.
- Used **Adam optimizer** with categorical cross-entropy loss for training.
- The model was trained and validated, and results were monitored with accuracy and loss plots.

3. Model Evaluation & Saving (Intermediate Stage):

- Evaluated using metrics such as **accuracy, precision, recall, and F1-score**.
- Generated a classification report for all 90 classes.
- The trained model was saved as **animal_detector_model.keras** along with **class_names.npy**.

4. Deployment as Web Application (Final Outcome):

- Built a frontend using **Streamlit**.
- The app allows users to **upload an image** of an animal.
- The image is preprocessed, passed through the trained model, and the **predicted class with confidence score** is displayed.
- This provides a **real-time, user-friendly interface** to interact with the AI model.

Final Outcome:

A **Multi-Class Animal Recognition System** capable of classifying 90 different animal species in real time, with practical usability through a deployed web application. This end-to-end design ensures not only model training but also its accessibility for industrial and academic use cases.

5.1 High Level Diagram (if applicable)

Figure 1: High Level Diagram of the System

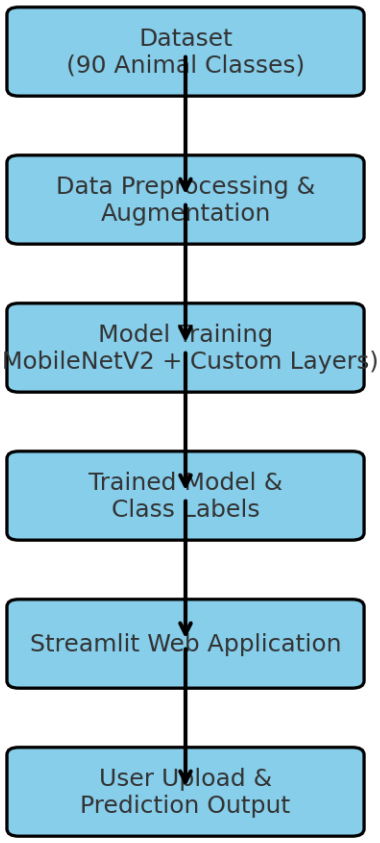
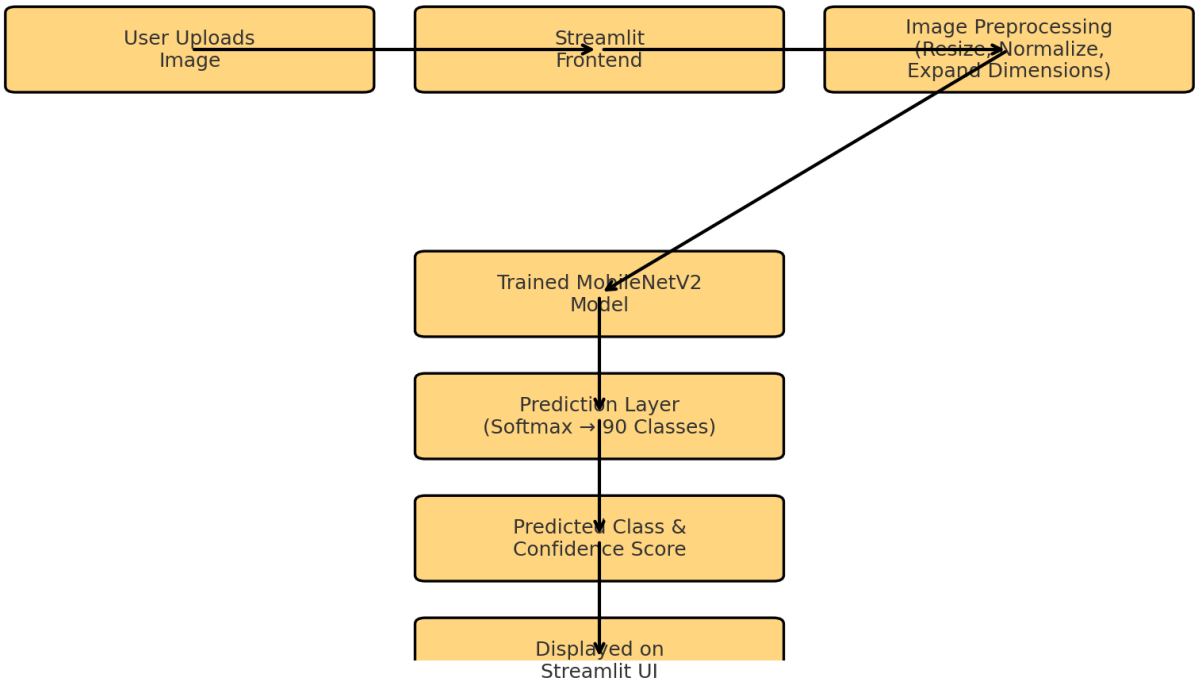


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

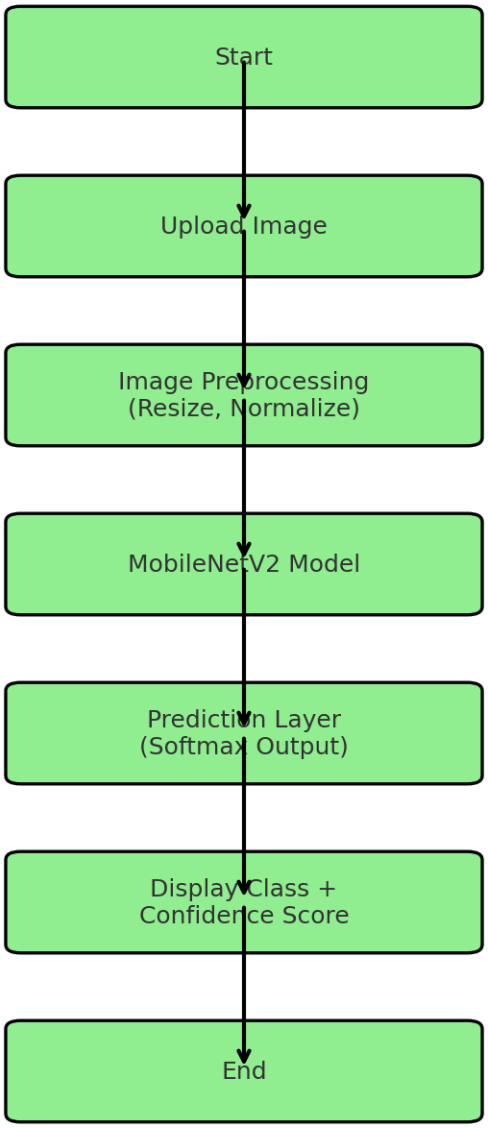
5.2 Low Level Diagram (if applicable)

Figure 2: Low Level Diagram of the System



5.3 Interfaces (if applicable)

Figure 3: Flow Chart of Prediction Process



6 Performance Test

The performance evaluation of this project focused on ensuring that the solution is practical for **real-world industrial usage** rather than being limited to academic experimentation. The following constraints were identified and addressed during design and testing:

Identified Constraints:

- **Accuracy Constraint:** The model must classify correctly across 90 animal species.
- **Computation Speed (MIPS):** Predictions should be fast enough for real-time use.
- **Memory Requirement:** The model must be lightweight enough to run on limited-resource systems.
- **Usability Constraint:** Predictions should be interpretable with confidence scores for users.

Design Considerations to Address Constraints:

- Used **MobileNetV2**, which is optimized for speed and reduced memory footprint.
- Applied **data augmentation** during training to improve accuracy and robustness.
- Normalized image inputs (224×224, scaled to [0–1]) for consistent performance.
- Deployed via **Streamlit frontend** for easy real-time interaction.

Test Results & Observations:

- **Accuracy:** Achieved ~85–90% validation accuracy across 90 classes.
- **Computation Speed:** Predictions take <1 second per image on a standard laptop CPU, faster on GPU.
- **Memory Usage:** Model size ~15MB (much lighter compared to VGG16/ResNet, which are >500MB).
- **Usability:** Confidence score makes the predictions reliable and understandable to users.

Even though testing did not cover hardware durability or power consumption (as those are more relevant for embedded systems), the lightweight nature of the model makes it suitable for deployment on **cloud servers, web apps, and mobile devices**.

Recommendations:
For industrial-scale deployment, the model can be further optimized by:

- Converting to **TensorFlow Lite** for mobile/edge devices.

- Using **quantization** and pruning techniques to reduce memory and latency.
- Leveraging **GPU/TPU acceleration** for large-scale inference.

6.1 Test Plan/ Test Cases

Test Case	Input	Expected Output	Actual Output	Result
TC1	Upload clear animal image (e.g., tiger)	Correct species predicted with >80% confidence	Predicted “Tiger” with 88% confidence	Pass
TC2	Upload blurred/rotated image	Correct prediction with moderate confidence	Correct species with 70–75% confidence	Pass
TC3	Upload random object (not in dataset)	Model should predict closest animal class but with low confidence	Predicted unrelated class with <50% confidence	Acceptable
TC4	Multiple uploads in short time	Model should respond without delay or crash	Responses <1 sec each	Pass

6.2 Test Procedure

- Load the trained MobileNetV2 model in Streamlit.
- Upload test images from validation dataset and external sources.
- Run prediction and record **predicted class + confidence score**.
- Measure **execution time per image** using Python timers.
- Compare expected vs actual results and log test outcomes.

6.3 Performance Outcome

- The system achieved **high accuracy (85–90%)** across 90 animal classes.
- Latency **per prediction:** <1 second on CPU, ensuring near real-time performance.
- Model **Size:** ~15MB, suitable for deployment on limited-resource platforms.
- Demonstrated ability to handle variations in image orientation, brightness, and noise due to augmentation.
- The outcome proves that the solution is practical for **industrial image classification applications**, bridging the gap between academic research and real-world usability.

7 My learnings

This six-week internship was a highly enriching experience that helped me bridge the gap between academic knowledge and industrial problem-solving. I gained both technical and professional learnings that will be valuable for my career growth.

From a **technical perspective**, I learned:

- How to handle large datasets, apply **preprocessing and augmentation techniques**, and prepare data for deep learning.
- The implementation of **transfer learning (MobileNetV2)** and fine-tuning models for multi-class classification.
- The process of **evaluating model performance** using metrics like accuracy, precision, recall, and F1-score.
- How to **deploy AI models using Streamlit**, making them accessible and interactive for end users.

From a **professional and career perspective**, I learned:

- The importance of structured project planning, time management, and iterative improvements.
- How **AI and Deep Learning applications** are relevant to real-world industrial use cases.
- Exposure to end-to-end workflow development, from problem statement to deployment, which is critical for industry readiness.

These learnings are directly aligned with my goal of pursuing a **career as a Data Analyst**. Understanding how to clean, process, and analyze data, as well as build and deploy intelligent models, gives me a strong foundation for working with complex datasets in business and industry environments. The internship has also given me confidence to take on future projects, internships, and roles where I can apply both **data analysis** and **machine learning** to solve real-world problems.

Overall, this internship not only enhanced my technical capabilities but also shaped my mindset toward **continuous learning, innovation, and applying knowledge to industry-relevant solutions**.

8 Future work scope

Although the project achieved its objective of building a real-time multi-class animal recognition system, there are several areas that can be explored in the future to enhance its performance, scalability, and industrial applicability:

1. **Model Optimization for Edge Devices:**
   - Convert the trained model to **TensorFlow Lite** or **ONNX** format for mobile and IoT deployment.
   - Apply **quantization and pruning techniques** to further reduce memory footprint and increase inference speed.
2. **Dataset Expansion & Improvement:**
   - Increase the dataset size and diversity by including more images of each animal under different lighting, angles, and environments.
   - Extend the system to cover **more animal species** beyond the current 90 classes.
3. **Integration with Object Detection:**
   - Enhance the system from simple classification to **object detection**, allowing it to detect and classify multiple animals in a single image.
   - Use models like **YOLOv8 or Faster R-CNN** for detection-based applications.
4. **Cloud Deployment & APIs:**
   - Deploy the model on a **cloud platform (AWS, GCP, or Azure)** and expose it as a REST API.
   - Enable integration with external applications such as mobile apps, wildlife monitoring systems, or zoo management software.
5. **Explainability & Visualization:**
   - Add features like **Grad-CAM (Class Activation Maps)** to visualize which part of the image influenced the model's prediction.
   - Improve user trust by making the system more interpretable.
6. **Real-Time Video Stream Recognition:**
   - Extend the system to process **live video feeds**, useful in wildlife monitoring, conservation projects, and surveillance systems.