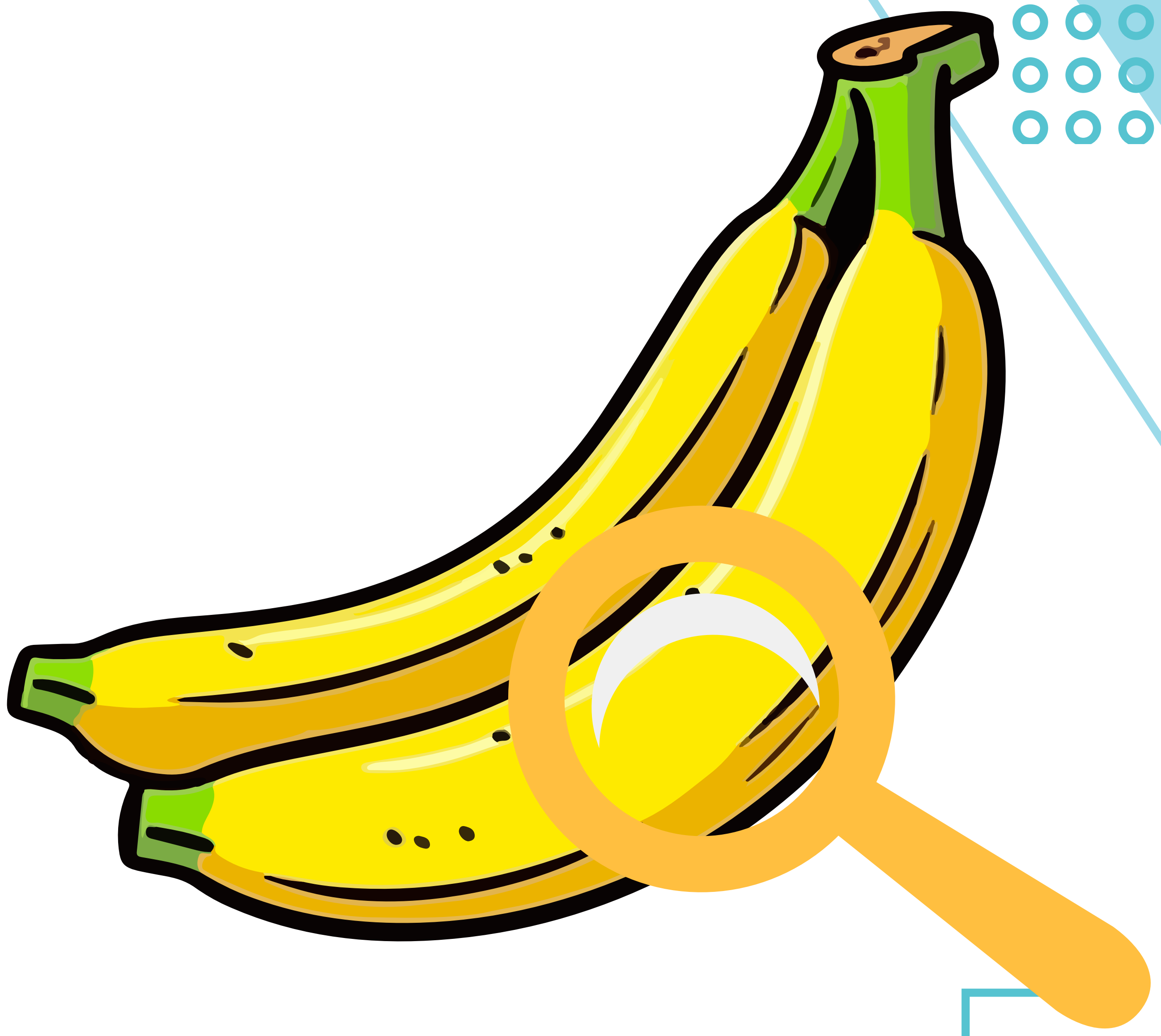
Automatic Fruit Quality Detection using Deep Learning



Group-08

Adithyan Manoj- MDL21EC009

Rohith M S- MDL21EC099

Ruben Davis Saji- MDL21EC102

Muhammad Hadhi V M- LMDL21EC134

Guided by
Ms. Rashida K
Assistant Professor
Dept of EC

Contents

Introduction

- Global demand for efficient and accurate fruit quality control
- Critical need for automated, scalable, and reliable classification systems
- Addresses challenges in the fruit industry with innovative solutions

Objectives

- Real-Time Classification
- Enhanced Accuracy through Deep Learning
- Category-Based Sorting
- Improving Environmental Conditions
- Enable Automated Sorting Using Conveyor Belt

Applications

- Automated sorting and grading in packaging facilities.
- Supply chain optimisation and post harvest management
- Research and development in agriculture technology, export quality assurance.

Scope:

1. Real-time fruit quality classification
2. Deployed on Jetson Nano (edge computing)
3. Conveyor belt integration for automation
4. Suitable for small-to-mid scale sorting units
5. Reduces manual effort and inconsistency

Novelty

- Implement Jetson Nano for efficient, real-time fruit quality analysis and sorting.
- Combines classification, and automated physical sorting.
- Utilizing MobileNetV2 for deep learning model.

LITERATURE REVIEW

SL NO	YEAR	TITTLE & SOURCE	OBJECTIVE	METHODOLOGY	RESULTS/CONCLUSIONS	LIMITATIONS
1	2023	A General machine Learning model for accessing fruit quality using deep image features	• A general machine learning model using Vision Transformers was introduced for fruit quality assessment. • The model focuses on identifying fruit quality based on visual appearance. • It eliminates the need for fruit-specific models by providing a universal solution.	• A Vision Transformer (ViT) model was developed and trained for fruit quality assessment using images. • Images were divided into patches and encoded. • These patches were processed through transformer blocks with multi-head attention and multilayer perceptron layers	• The model achieved equal or higher accuracy for most fruits compared to dedicated models. • It struggled with specific fruits like bananas and pomegranates. • The study emphasized the need to consider factors beyond just visual appearance for accurate fruit quality assessment.	• The model only considers visual features, potentially missing other important quality factors like internal defects or ripeness. • It struggles with certain fruits, like bananas and pomegranates, indicating it may not fully capture the unique characteristics of all fruit types. • Vision Transformers require significant computational resources, which might limit the model's practical application in resource-constrained environments.

2	2020	A Novel Transfer Learning Approach for Pomegranate growth detection	• To develop a transfer learning-based approach for accurately classifying pomegranate growth stages. • To compare the proposed method with traditional models, showcasing its superior accuracy.	• Collect and preprocess pomegranate growth stage images. • Develop models using CNN and transfer learning-based feature engineering. • Evaluate model performance with cross-validation and compare with existing methods.	• CNN achieved moderate accuracy (up to 65%) but with high error rates in early growth stages. • Proposed RF model with new features reached 98% accuracy, outperforming traditional methods. • Cross-validation confirmed the robustness of the proposed model across different pomegranate growth stages.	• Traditional CNNs showed suboptimal accuracy and high error rates in classifying early growth stages. • The RF model, despite high accuracy, required longer training times compared to other techniques. • Performance heavily relied on the newly proposed features, which may limit generalization to different datasets or conditions.
3	2023	Real time Oil Palm Grading system using Mobile and Yolo4	• Improve YOLOv4 model accuracy for detecting and classifying oil palm FFB ripeness from smartphone images. • Develop an Android app for real-time ripeness classification of oil palm FFB using the optimized model.	• Collect and preprocess FFB images with data augmentation. • Train and evaluate YOLOv4 models for ripeness classification. • Convert the best model to TensorFlow Lite and deploy it on an Android app.	• Achieved high accuracy in ripeness classification using YOLOv4. • Successfully reduced model size and latency with TensorFlow Lite conversion. • Implemented real-time ripeness detection on Android with minimal latency.	• Limited to detecting only predefined ripeness stages. • Performance varies with different banana varieties and conditions. • Requires high-quality images for accurate detection.

4	2021	Fruit Quality Recognition using Deep Learning Algorithm	To construct a fruit classification model using a convolutional neural network (CNN) to classify fruits into categories	Preprocessing: • Convert images from RGB to Gray. • Apply thresholding and segmentation. • Use median filtering to reduce noise and smooth images. Feature Extraction • Utilize convolutional layers for feature extraction. • Apply ReLU layers for non-linear activation. • Reduce dimensionality with pooling layers. • Flatten 2D data into a 1D column for classification.	• Provides a detailed description of the system's performance, capturing both common and rare occurrences in fruit grading. • Allows for nuanced distinctions and the detection of ambiguity in results. • Results are visually represented in the output image	The research acknowledges that no single CNN architecture can be declared superior to others, indicating that the performance of different CNN models may vary depending on the specific application or design.
5	2023	A General Machine Learning Model for Assessing Fruit Quality Using Deep Image Features	1. Develop a versatile ML model for fruit quality assessment across multiple fruit types. 2. Evaluate model scalability and generalizability. 3. Reduce the need for fruit-specific models. 4. Demonstrate the effectiveness of deep image features.	1. Employed Vision Transformer (ViT) for feature extraction. 2. Trained on a diverse fruit dataset covering various types. 3. Performed comparative analysis with dedicated models. 4. Evaluated using accuracy, F1-score, and confusion matrices.	1. High accuracy (up to 100%) on fruits like apples and strawberries. 2. Consistent performance across multiple fruits. 3. Demonstrated superior generalizability over fruit-specific models. 4. Potential for real-world applications in quality control.	1. Lower accuracy for visually complex fruits (e.g., guavas, lemons). 2. Limited to visual assessment, ignoring non-visual quality factors (e.g., taste, texture). 3. Requires extensive image data for training. 4. Computationally intensive, impacting real-time application.

6	**2024**	Enhancing Fruit Quality Detection with Deep Learning Models	• The study aims to improve fruit quality detection using deep learning techniques. • Specifically, it focuses on identifying defects, assessing ripeness, and sorting fruits based on visual cues. • The objective is to enhance post-harvest processes by automating qualit assessment.	• The researchers propose an EfficientB2 convolutional neural network (CNN) model. • This model extracts deep features from a dataset of processed fruit images. • The study addresses the limitations of previous approaches by considering both accuracy and robustness. .	• The proposed model outperforms previous methods in terms of efficiency and accuracy. • Fruit quality assessment benefits from this comprehensive solution. • By leveraging deep learning, the system achieves reliable detection and classification.	• Data Availability: Limited annotated fruit quality datasets may impact model performance. • Generalization: The model's effectiveness across various fruit types and conditions needs validation. • Real-World Deployment: Practical deployment may face challenges related to hardware constraints and environmental variations.

7	**2022**	Determination of Fruit Quality by Image Using Deep Neural Network	• Develop a robust DNN-based system to accurately determine fruit quality using image data, addressing the need for automated and reliable quality assessment in modern agriculture. • Target key quality metrics such as ripeness, surface defects, and overall condition, ensuring the system provides actionable insights for practical agricultural applications like sorting and grading.	• Employed a deep neural network, likely a convolutional neural network (CNN) variant, trained on a comprehensive dataset of fruit images collected from various sources, annotated with quality labels. • Processed high-resolution images to extract features for quality classification.	• Achieved high accuracy in identifying ripeness levels across tested fruits. • Effectively detected visual defects, supporting quality control in agriculture.	• Focused solely on visual quality attributes, overlooking non-visual factors such as taste, aroma, or internal defects (e.g., rot or pest damage), which are critical for comprehensive quality evaluation. • Lacked emphasis on real-time deployment, unlike edge-based systems.

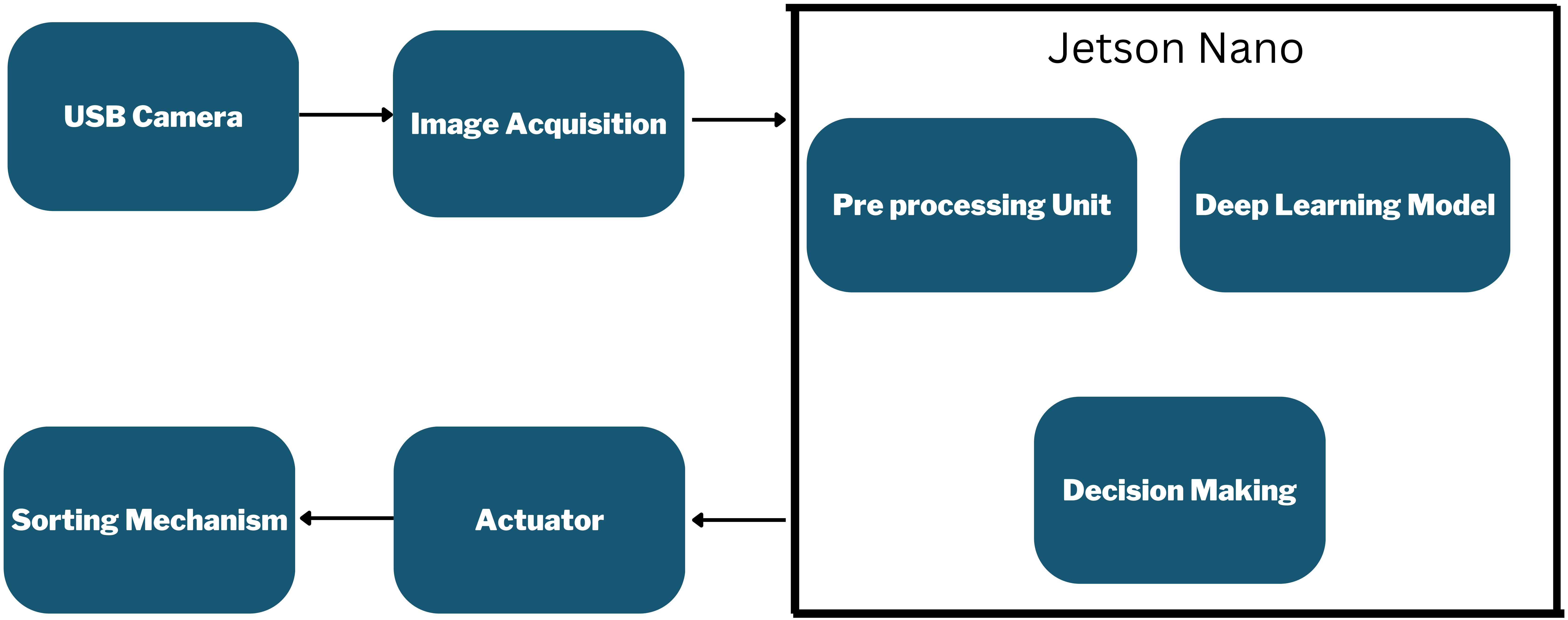
8	**2021**	Fruits and vegetables quality evaluation using computer vision	• The article aims various computer vision methods for assessing the quality of fruits and vegetables. • Specifically, it focuses on methods related to color, texture, size, shape, and defect detection	• The authors discuss different stages in quality inspection: image acquisition preprocessing, segmentation, feature extraction, and classification. • Various algorithms and techniques are explored, including color space transformations, thresholding, clustering, and machine learning	○ Researchers have used computer vision to address quality aspects such as appearance, defects, and grading. ○ Algorithms based on color, texture, and shape features have been proposed for fruit and vegetable quality assessment	• Challenges include handling variations in lighting, background, and fruit shape. • Some methods may be sensitive to environmental conditions or require extensive computational resources.

9	**2022**	Fruit Quality Assessment with Densely Connected Convolutional Neural Network	1. Automate the process of fruit quality assessment. 2. Improve accuracy in fruit classification and quality grading. 3. Develop a deep learning model for real-time applications. 4. Address fine-grained classification issues in fruit quality.	1. Applied DenseNet201, a deep learning model, for feature extraction and classification. 2. Performed data augmentation to handle limited data. 3. Employed the FruitNet dataset, comprising 19,526 images of six different fruits across three quality grades. 4. Fine-tuned the model using transfer learning techniques.	1. Achieved high accuracy of 99.67% in overall classification. 2. Effectively classified fruits and their quality levels. 3. Demonstrated robust performance in fine-grained quality detection tasks. 4. The model was tested on both raw and preprocessed data to evaluate consistency.	1. Model performance affected by imbalanced dataset. 2. Difficulty in distinguishing subtle differences in fruit quality. 3. Limited by the dataset's scope, which includes only six types of fruits. 4. Real-time deployment challenges due to hardware constraints.

10	**2017**	Deep fruit detection in orchards	• The study aims to develop an accurate and reliable image-based fruit detection system for orchards. • Specifically, the focus is on detecting fruits such as mangoes, almonds, and apples. • The system's purpose includes supporting yield mapping, robotic harvesting, and efficient resource utilization.	• The researchers use the Faster R-CNN (Region-based Convolutional Neural Network) framework for fruit detection. • Ablation studies are conducted to understand practical deployment, training data requirements, and variability capture • Data augmentation techniques significantly improve performance by reducing the number of required training images • Transferring knowledge between orchards doesn't contribute significantly to performance gains	• The study achieves the best detection performance for apples and mangoes compared to previous works. • An F1-score of > 0.9 is achieved for these fruits, demonstrating high accuracy.	• The study focuses on specific fruit types (mangoes, almonds, and apples) and may not generalize well to other fruits. • The tiling approach, while effective, may have limitations in terms of computational efficiency and scalability.

System Overview

3.1 Block Diagram



Feasibility study-Hardware

Feature	Jetson Nano	Jetson Xavier NX	Raspberry Pi 4 + Coral USB Accelerator
Processor	Quad-core ARM Cortex-A57 CPU, 128-core Maxwell GPU	6-core ARM v8.2 CPU, 384-core Volta GPU, 48 Tensor Cores	Quad-core ARM Cortex-A72 CPU + Edge TPU Accelerator
RAM	4 GB LPDDR4	8 GB or 16 GB LPDDR4x	4 GB or 8 GB LPDDR4 (Raspberry Pi)
Connectivity	Gigabit Ethernet, USB 3.0, GPIO, CSI	Gigabit Ethernet, USB 3.0, PCIe, GPIO, CSI	Gigabit Ethernet, USB 3.0, GPIO
Community & Resources	Strong support from NVIDIA's Jetson community	Strong support, enterprise-level resources	Huge Raspberry Pi community, limited TPU-specific support
Cost-effectiveness	Very cost-effective for basic AI tasks (~27000)	High performance-to-cost ratio (~116850)	Very cost-effective (~19000 for both Pi and Coral USB)

Feature	Raspberry Pi Camera Module 3	Arducam IMX477 High-Quality Camera	LG VC23GA
Resolution	12 MP (4056 x 3040 pixels)	12.3 MP (4056 x 3040 pixels)	1080p (1920 x 1080 pixels)
Image Quality	High-quality images with HDR and low-light support	High-quality, RAW image capture	Good image quality for video streaming
Frame Rate	1080p at 60 fps, 720p at 120 fps	1080p at 60 fps, lower for higher resolutions	1080p at 30 fps
Interface	MIPI CSI-2 (Compatible with Jetson Nano)	MIPI CSI-2 (Compatible with Jetson Nano)	USB 2.0 (Plug-and-play with Jetson Nano)
Size	25 x 24 x 9 mm	38 x 38 mm	94 x 24 x 29 mm
Cost	2923	8268	3681

Feasibility study-Deep Learning Models

Feature	MobileNetV1	MobileNetV2	MobileNetV3 small
Architecture	Depthwise separable convolutions	Inverted residuals & linear bottleneck additional to MobileNetV1	Depthwise separable convolutions and SE blocks.
Accuracy	Lower for the same computational complexity	Higher accuracy for similar computational complexity	High accuracy while significantly reducing computational cost.
Efficiency	Good for simpler tasks	Better efficiency and performance, especially for complex tasks	Optimized for fast inference
Use Case	Suitable for general mobile applications	Ideal for mobile and edge applications requiring better accuracy without much increase in resources	Real-time image classification on mobile and embedded devices.

Design Methodology

System Architecture.

- Conveyor Belt System:
  - Two conveyor belts driven by DC motors
  - Push Back Mechanism for sorting

- Image Capture Module:
  - USB Camera Mounted Above Conveyor for Real-Time Image Capture

- Processing Unit:
  - Jetson Nano
- Actuator Mechanism
  - Motor rotation and servo actuation for sorting

Implementation Steps

1. Data Collection and Preprocessing:

- Data Source: High-resolution images of banana of classes - good, bad and intermediate were obtained from

Kaggle-

1000 images per class in jpg format

Roboflow Universe

2000 images per class in jpg format

Mendely Datasets.

4000 images per class in jpg format

- Python library Augmentor was used.

2.Model selection and training

- Classification Model: MobileNetV2 → Good performance and low latency on edge devices
- Framework: PyTorch
- Transfer Learning →
  1. Freeze base layers
  2. Fine-tune fully connected layers
  3. Reduces training time

Model Parameters

- Learning Rate: 5x10^(-3)
- Optimizer: SGD
- Loss Function: CrossEntropyLoss
- Batch size:8

3.Deployment and Execution in Jetson Nano:

Environmental Setup:

- Installation of Pytorch, OpenCV and necessary libraries-

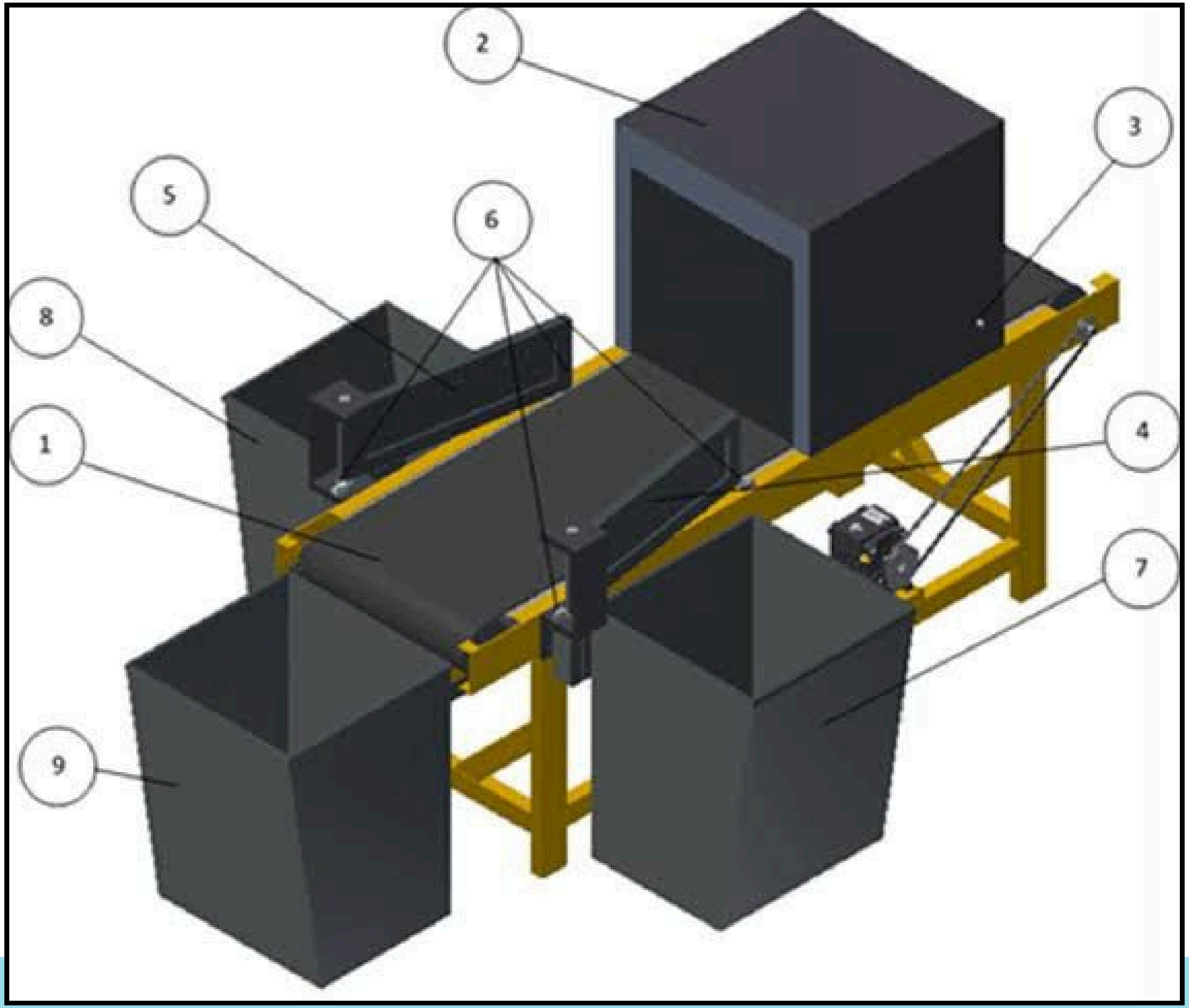
	Jetson Nano	Latest Version
Python	3.6.9	3.13
Torch	1.8	2.5
Torchvision	0.9	0.21
OpenCV	4.5.1	4.11

- Model Deployment:
  - Model trained using Pytorch.
  - After training the model is transferred to Jetson Nano.

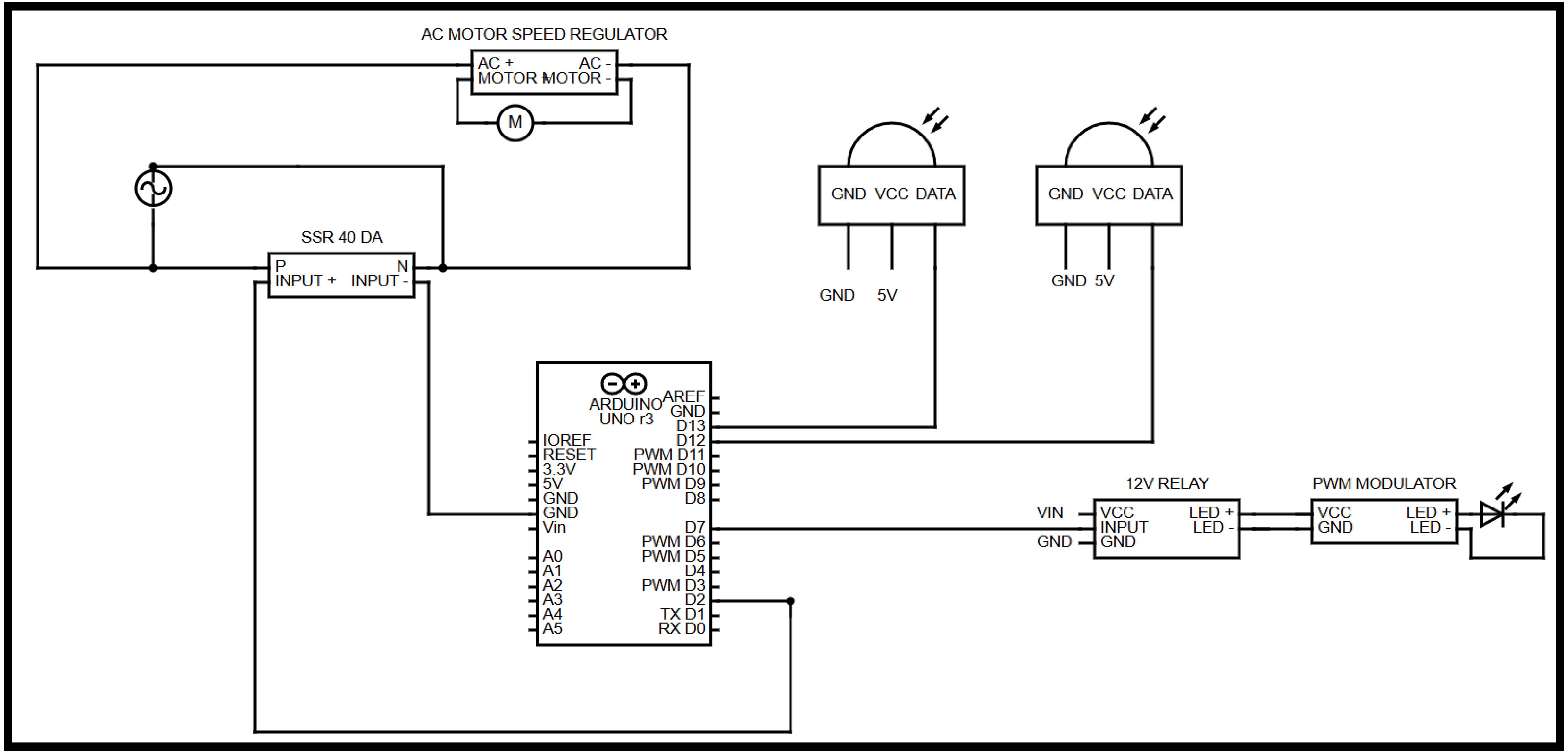
4. Conveyer Belt and Actuator Control Logic:

- Good quality: Moved to second conveyer belt.
- Intermediate quality: Moved to second conveyer belt.
- Bad quality: Push mechanism pushes away using the sample.

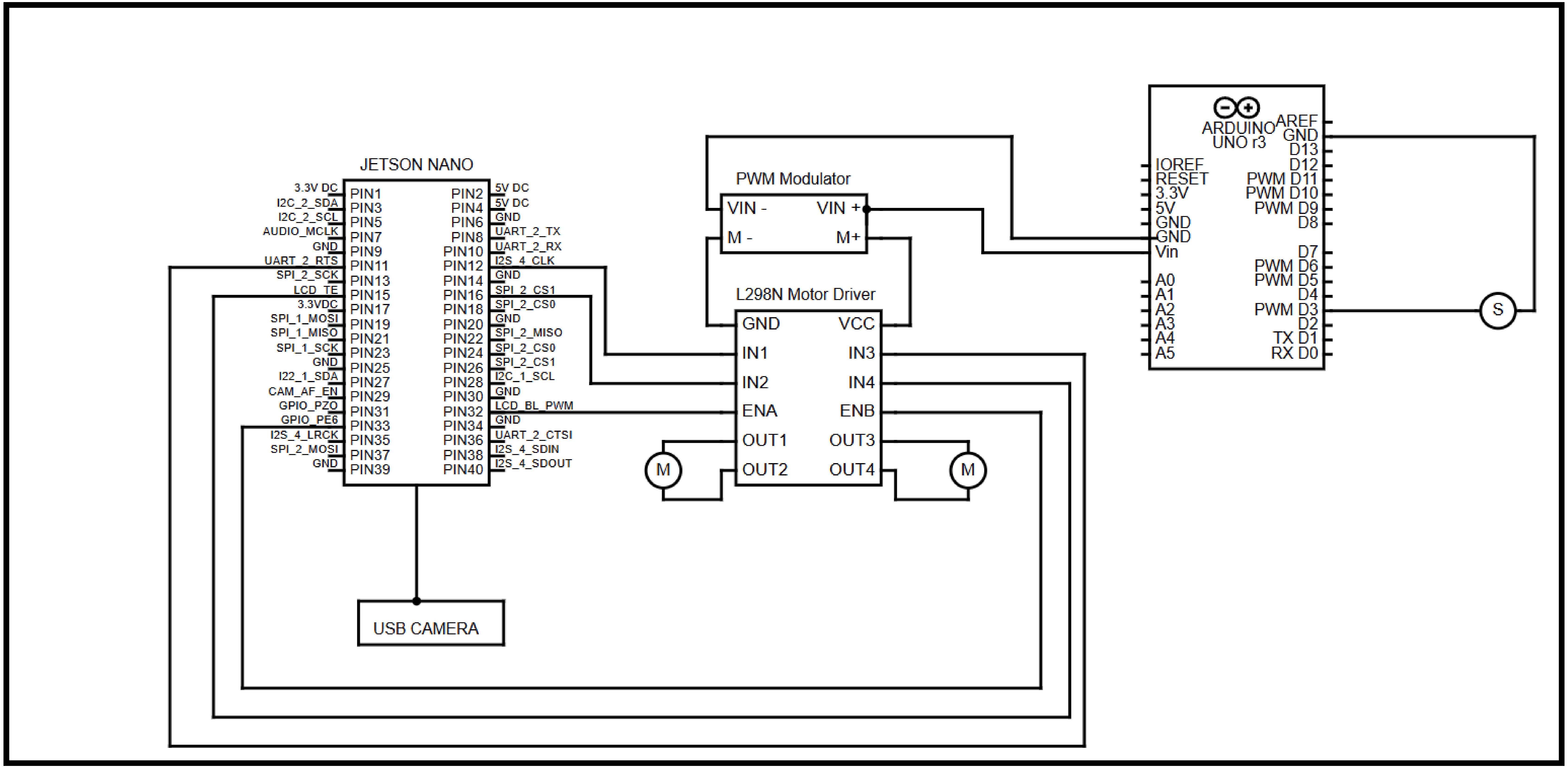
Hardware design



First Conveyer Belt System-



Second Conveyer Belt System-



Tools Required-

- Jetson Nano
  - RAM-4GB
  - Operating Voltage-5V
  - Minimum Current-2A
- Arduino Uno
  - Input Voltage-7-12V
  - Output Voltage- 0-5V
  - Maximum Current-500mA
- 12V Relay
  - Input Voltage: 12V DC
  - Contact Voltage: Up to 250V AC / 30V DC
  - Contact Current: 10A
- AC Motor Speed Regulator
  - Input Voltage: 220V AC
  - Output Voltage: 0-220V AC
  - Maximum Current: 10A

PWM Modulator

- Input Voltage: 5-12V DC
- Output Voltage: PWM Signal (0-5V)
- Frequency Range: 1 Hz - 20 kHz

IR Proximity Sensor

- Operating Voltage: 3.3V - 5V DC
- Detection Range: 2 cm - 30 cm
- Output Voltage: 0-5V (Digital)

SSR 40DA (Solid State Relay)

- Input Voltage: 3-32V DC
- Output Voltage: 24-380V AC
- Load Current: 40A

LG VC23GA Webcam

- Resolution: 2 MP (1920 x 1080)
- Frame Rate: 30 fps
- Lens Type: Fixed Focus

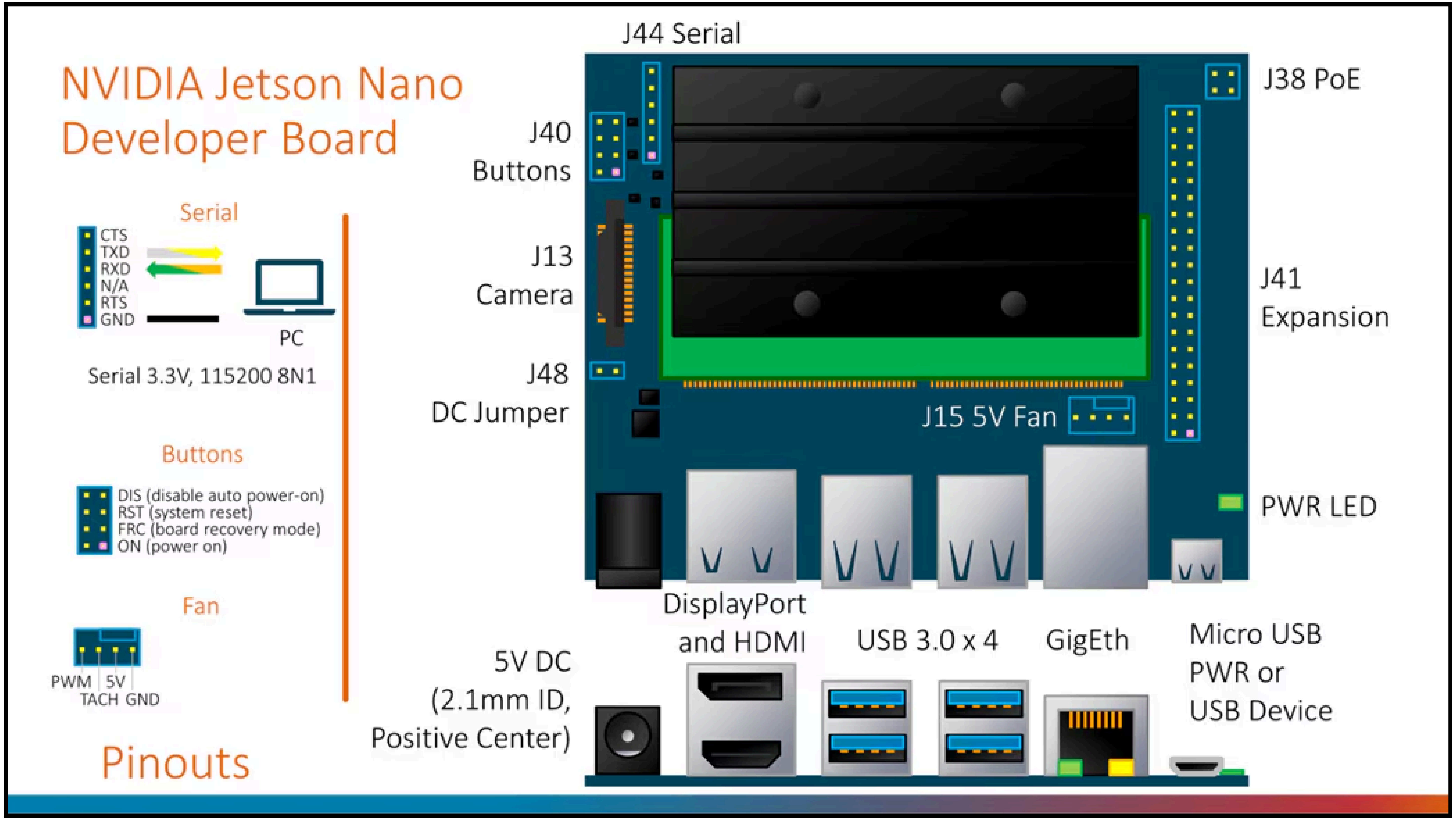
Servo Motors

- Operating Voltage: 4.8-6V DC
- Operating Speed: 0.1 sec/60° at 5V
- Stall Torque: 2.5 kg·cm

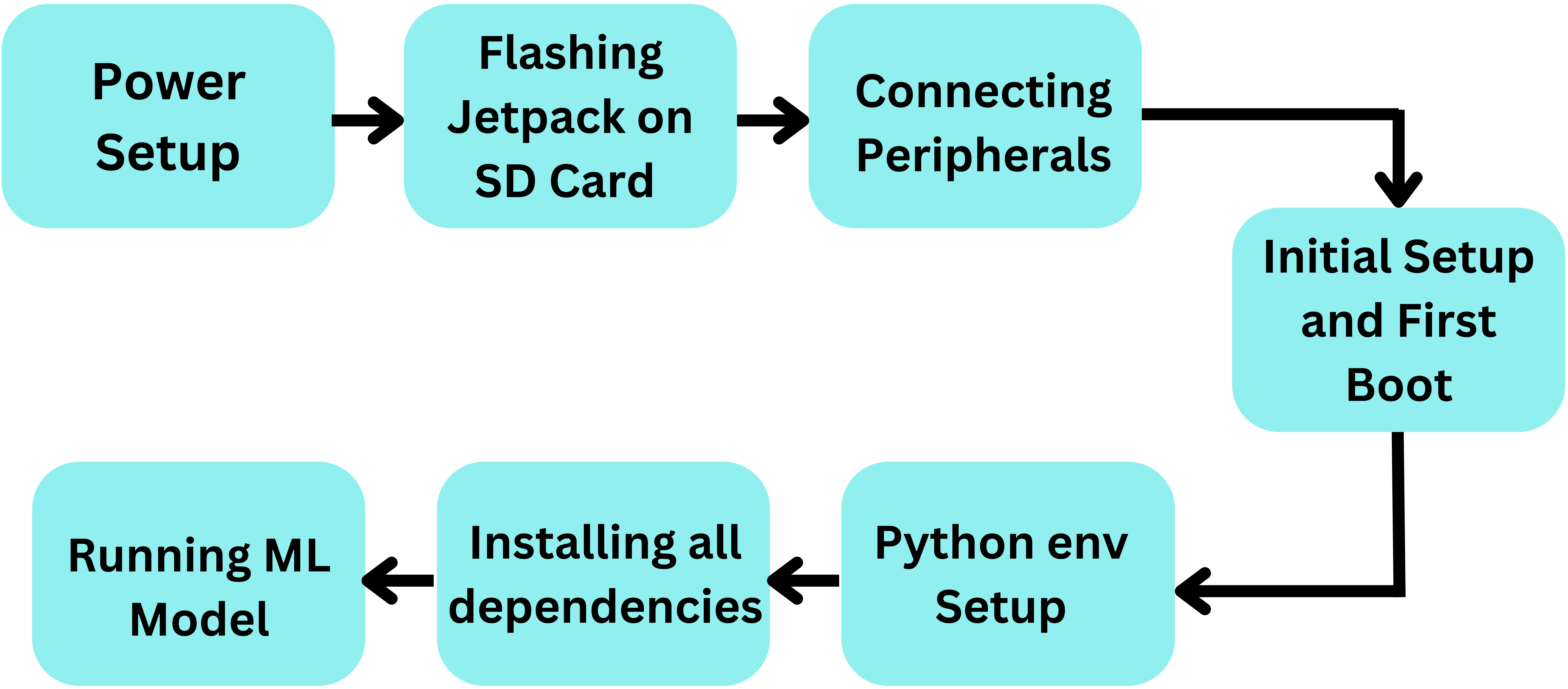
L298 Motor Driver

- Operating Voltage: 5-35V DC
- Output Voltage: 0-12V DC
- Maximum Current: 2A per channel

JETSON NANO


NVIDIA Jetson Nano
Developer Board
Serial
CTS
TXD
RXD
N/A
RTS
GND
PC
Serial 3.3V, 115200 8N1
Buttons
DIS (disable auto power-on)
RST (system reset)
FRC (board recovery mode)
ON (power on)
Fan
PWM
5V
TACH
GND
Pinouts
J44 Serial
J40
Buttons
J13
Camera
J48
DC Jumper
J38 PoE
J41
Expansion
J15 5V Fan
PWR LED
5V DC
(2.1mm ID,
Positive Center)
DisplayPort
and HDMI
USB 3.0 x 4
GigEth
Micro USB
PWR or
USB Device

Setting up environment in Jetson Nano:



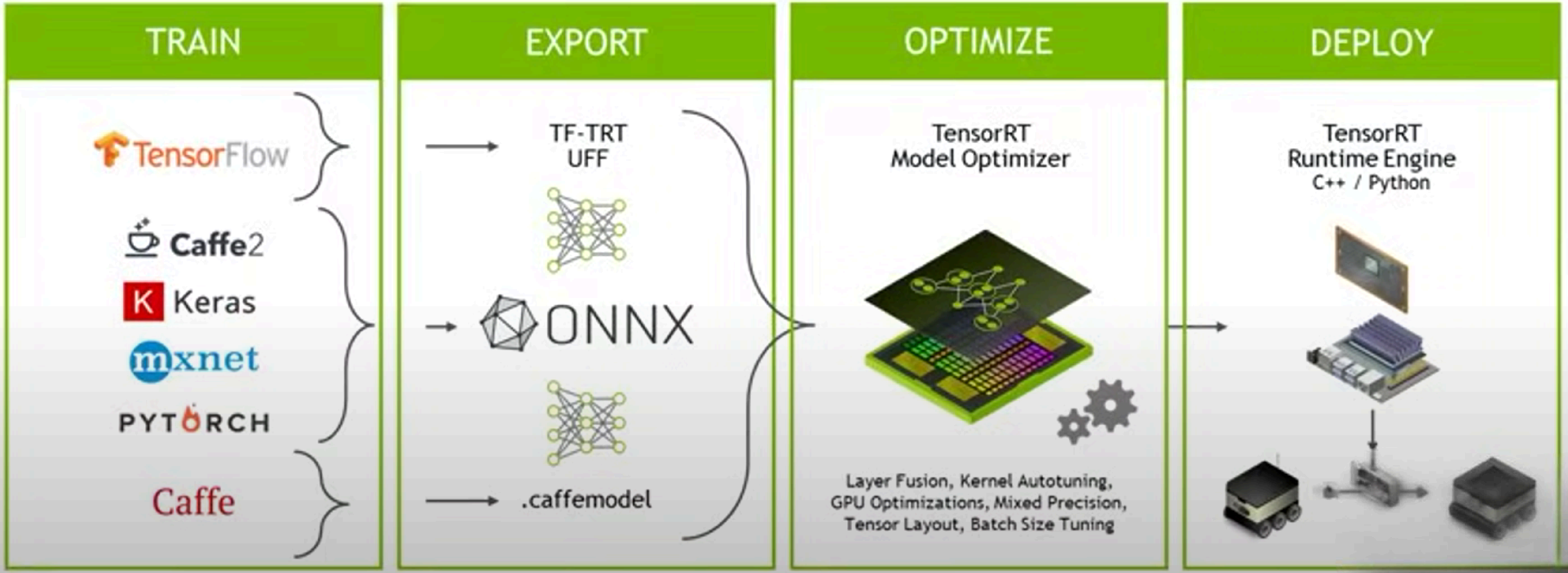
Comparison on deployment in cloud and Jetson Nano

Feature	Cloud Deployment	Jetson Nano Deployment
Latency	Moderate	Low
Throughput	High	Moderate
Power Consumption	Depends on cloud provider	Low to Moderate
Scalability	Highly Scalable	Limited Scalability
Data Privacy	Depends on Provider	High
Use Cases	- Large-scale image or video processing - Applications requiring integration with other cloud services	- Real-time image or video processing - IoT and edge applications

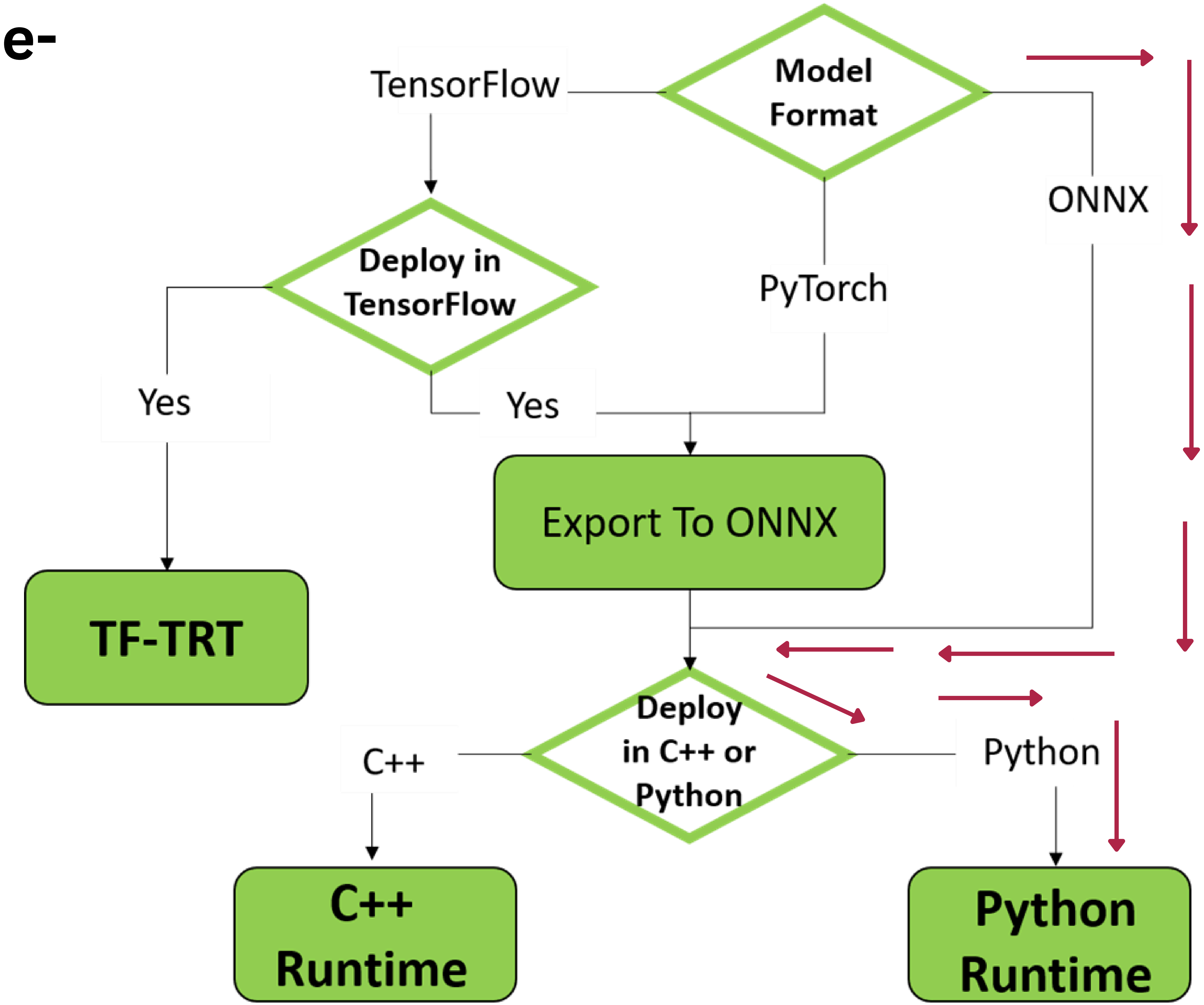
TensorRT-

- Accelerates deep learning inference using NVIDIA GPUs
- Converts models from PyTorch
- Uses CUDA for fast GPU execution
- Supports real-time deployment on Jetson Nano

NVIDIA TensorRT



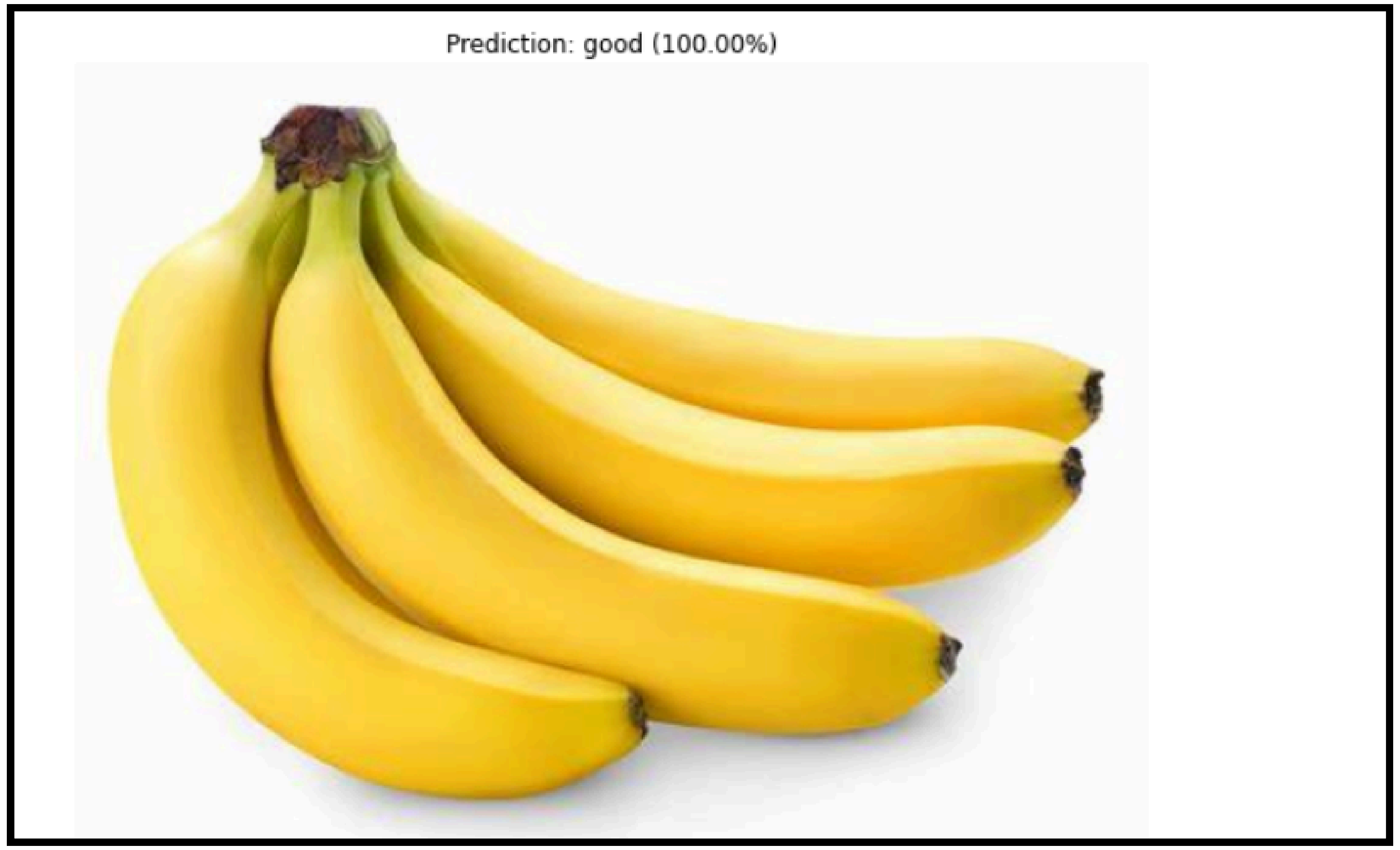
Deployment Pipeline-



Result

Classified Output of MobileNetV2

Shows banana quality classes: "Good", "Intermediate", "Bad"



Using device: cpu
Model loaded successfully from banana_mobilenet_best.pt
Class names: ['bad', 'good', 'inter']
Image: OIP(1).jpg
Prediction: good (Confidence: 100.00%)
Class probabilities:
  bad: 0.00%
  good: 100.00%
  inter: 0.00%

Prediction: good (100.00%)

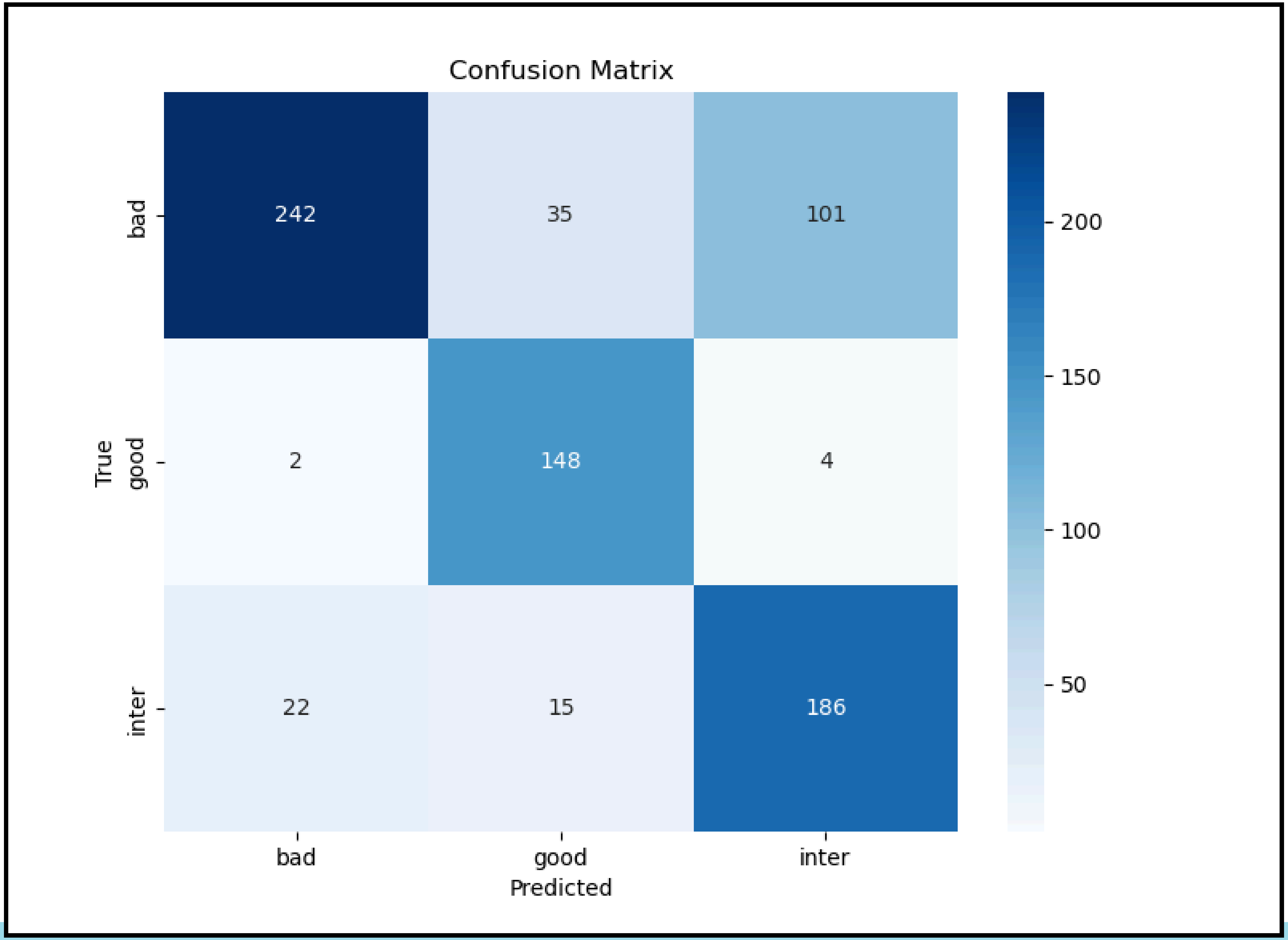
Using device: cpu
Model loaded successfully from banana_mobilenet_best.pt
Class names: ['good', 'inter', 'bad']
Image: Deshi-0083.jpg
Prediction: inter (Confidence: 82.26%)
Class probabilities:
  good: 15.04%
  inter: 82.26%
  bad: 2.70%

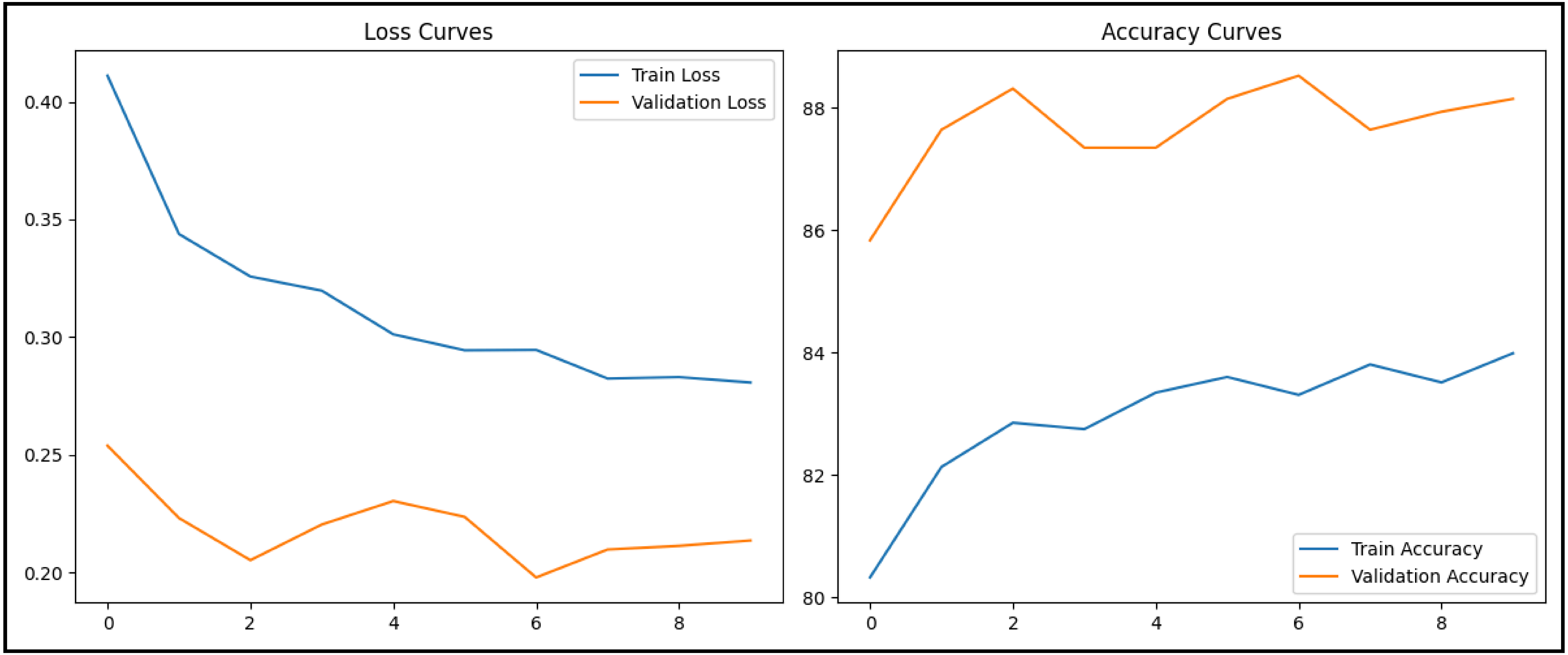
Prediction: inter (82.26%)

Using device: cpu
Model loaded successfully from banana_mobilenet_best.pt
Class names: ['bad', 'intermediate', 'good']
Image: rotten.jpg
Prediction: bad (Confidence: 100.00%)
Class probabilities:
  bad: 100.00%
  intermediate: 0.00%
  good: 0.00%

Prediction: bad (100.00%)

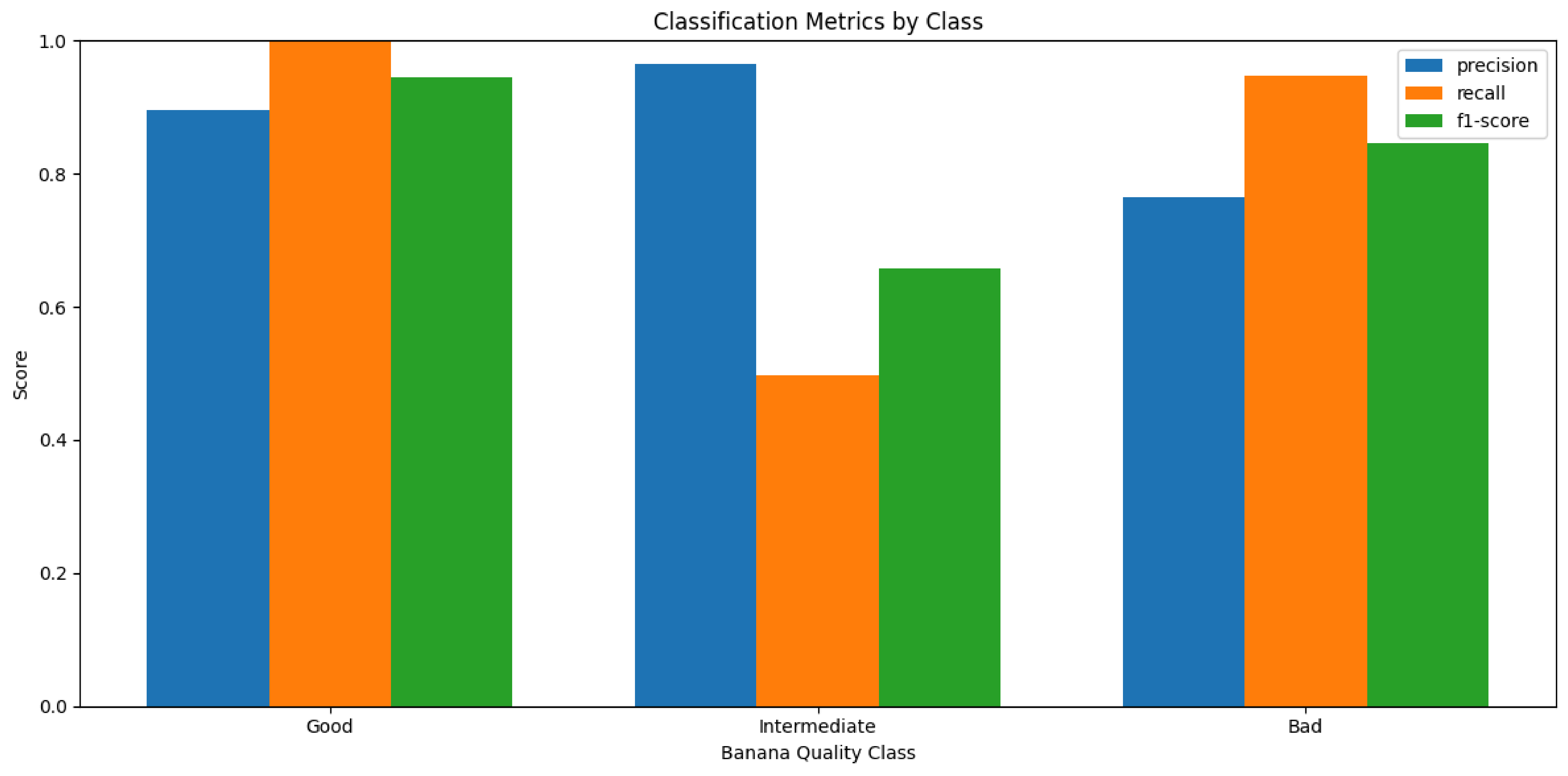
Confusion Matrix and Validation Graph of MobileNetV2



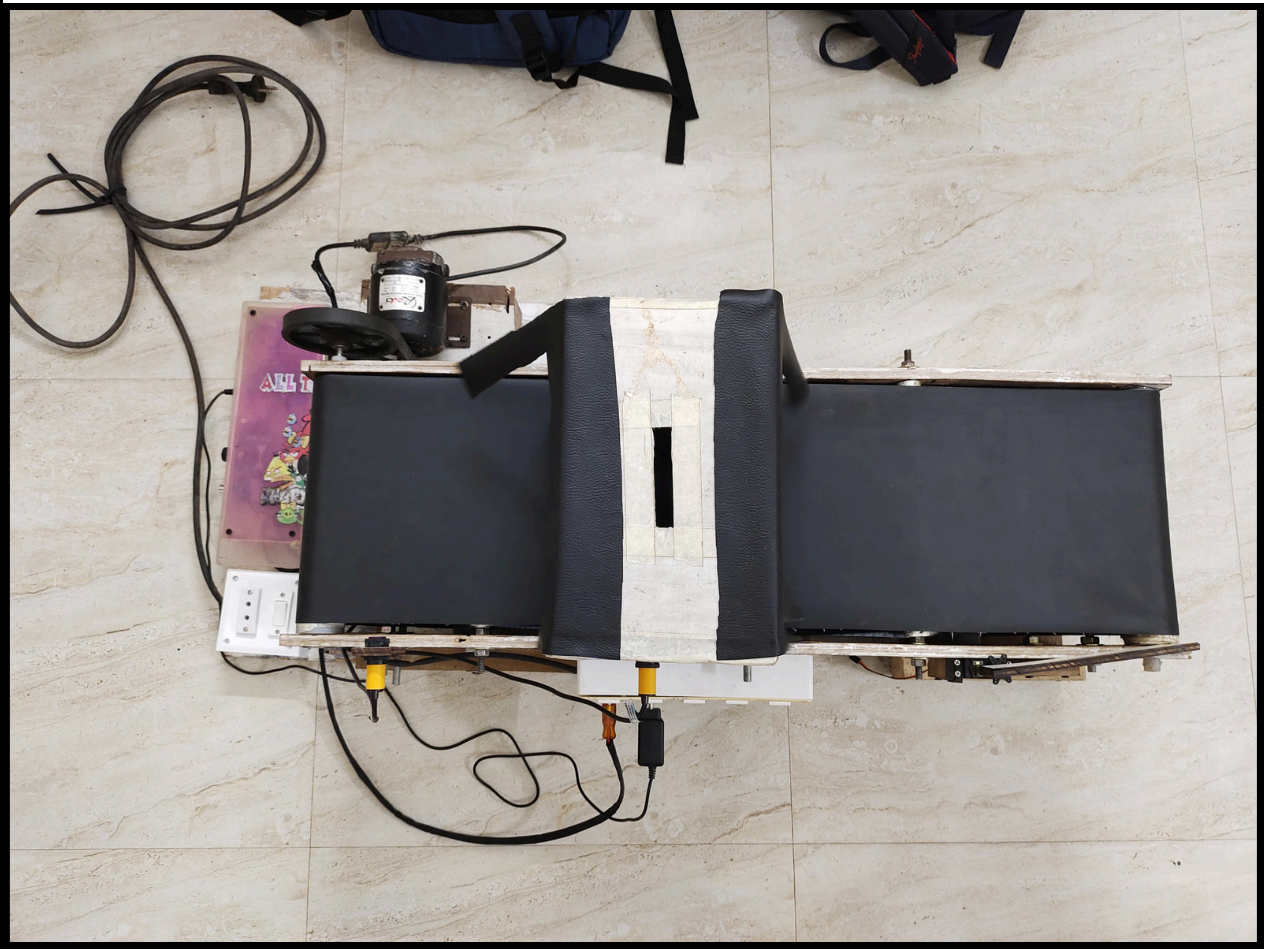


Loss Curves
Train Loss
Validation Loss
0.40
0.35
0.30
0.25
0.20
0
2
4
6
8
Accuracy Curves
88
86
84
82
80
Train Accuracy
Validation Accuracy
0
2
4
6
8

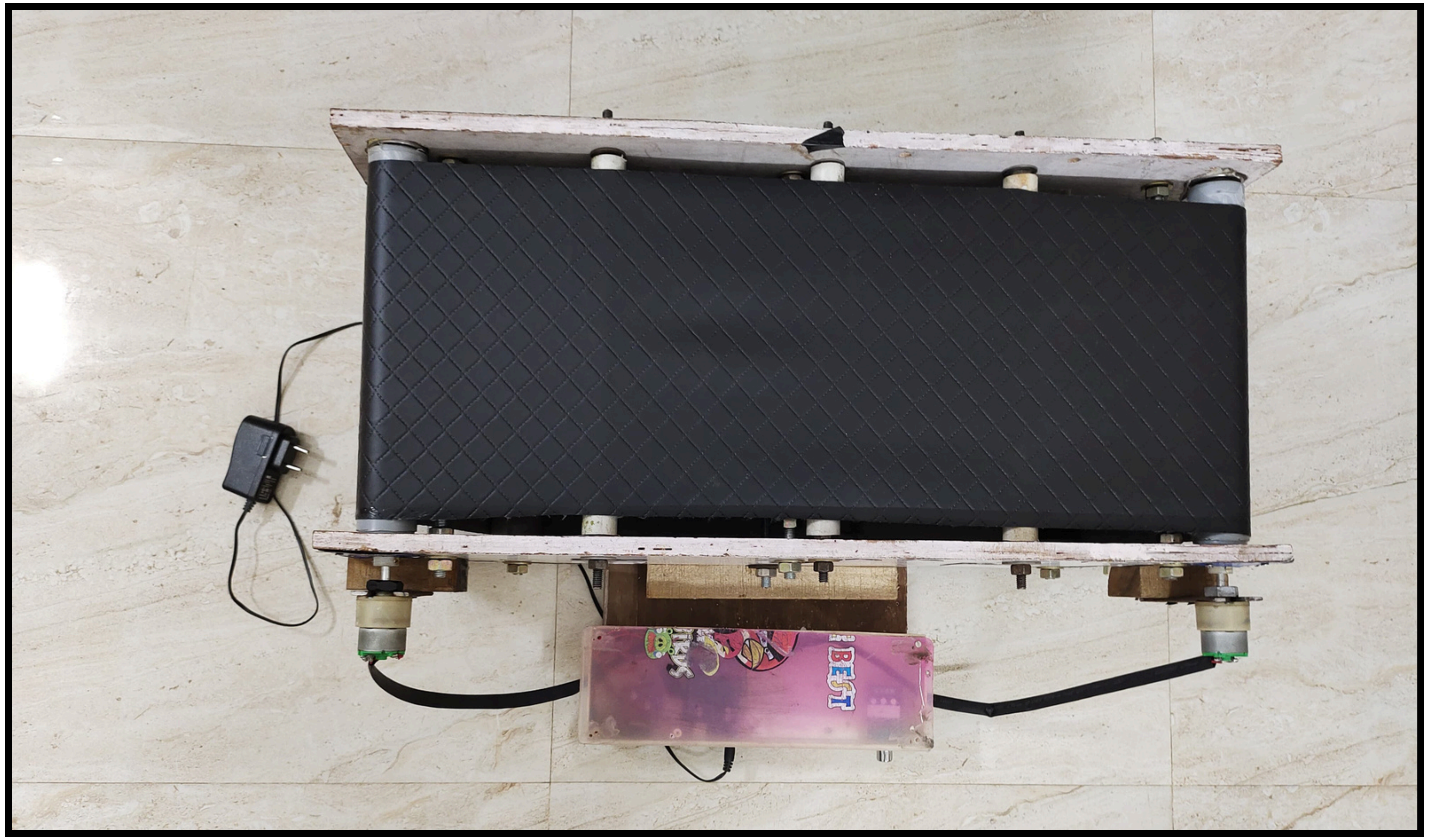
Class	Precision	Recall	F1-Score	Support
Good	0.90	1.00	0.94	154
Intermediate	0.97	0.50	0.66	223
Bad	0.76	0.95	0.85	378
Accuracy		0.83		755
Macro Avg	0.88	0.81	0.82	755
Weighted Avg	0.85	0.83	0.81	755



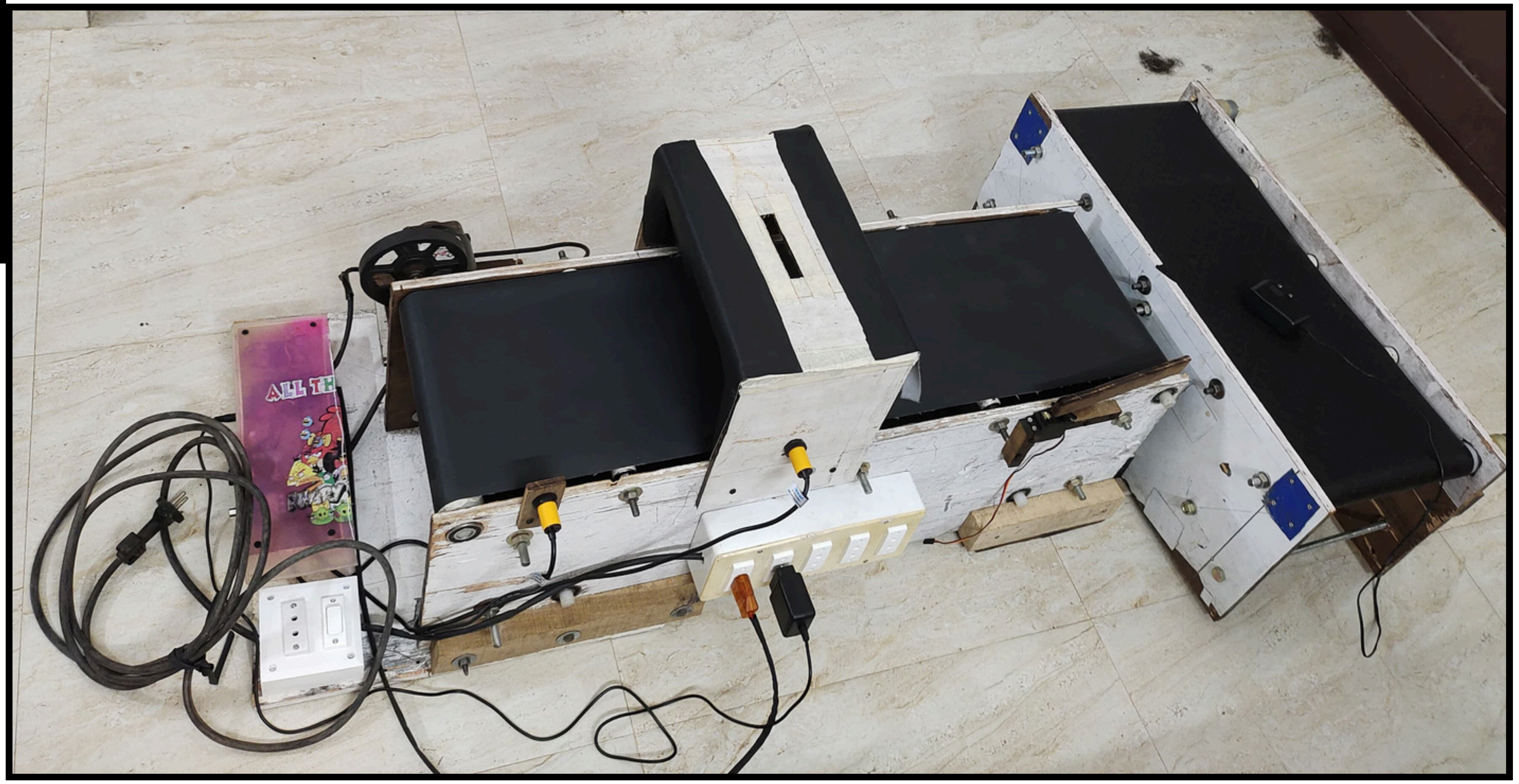
First Conveyer Belt with Imaging Box



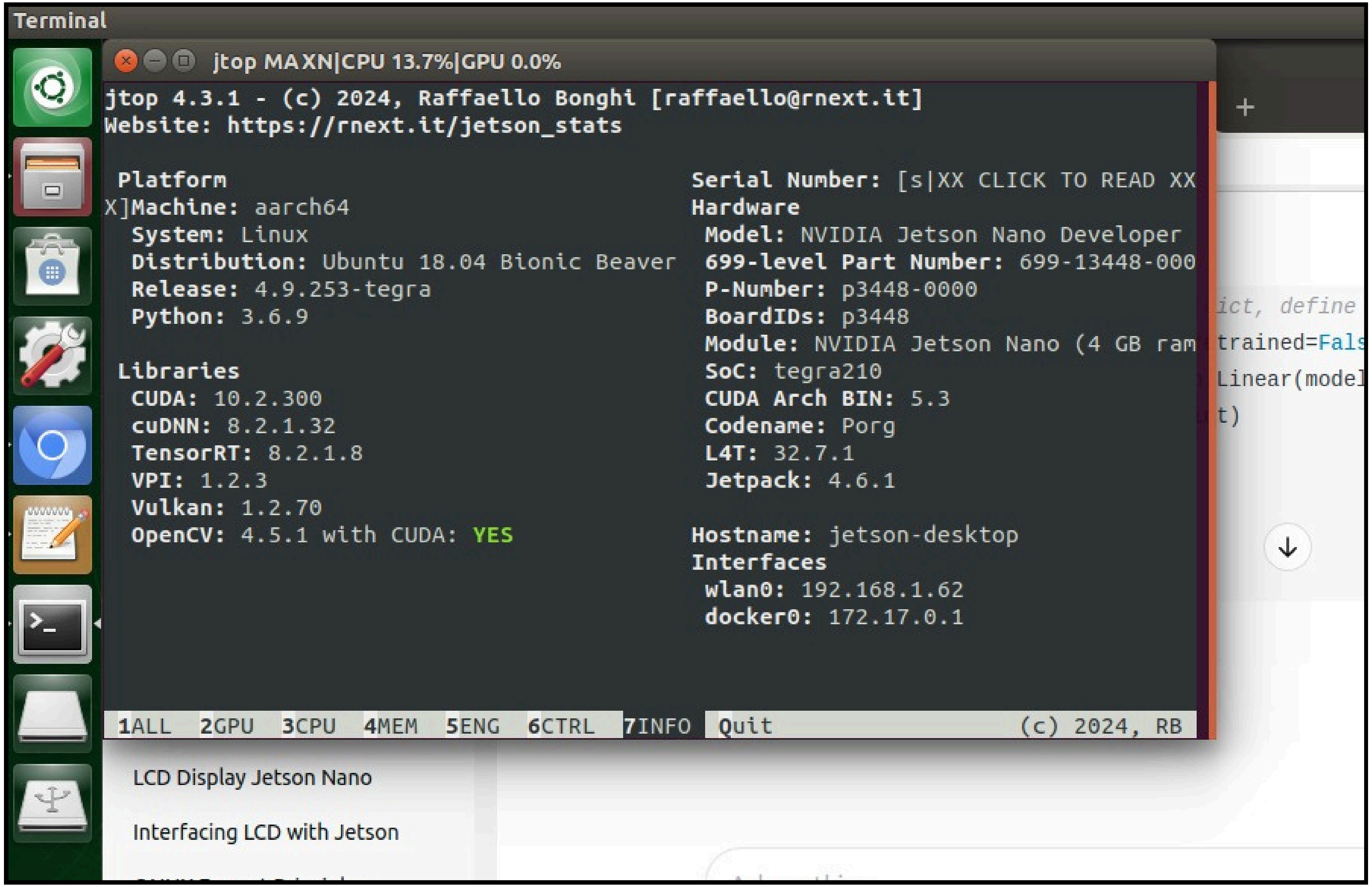
Second Conveyer Belt-



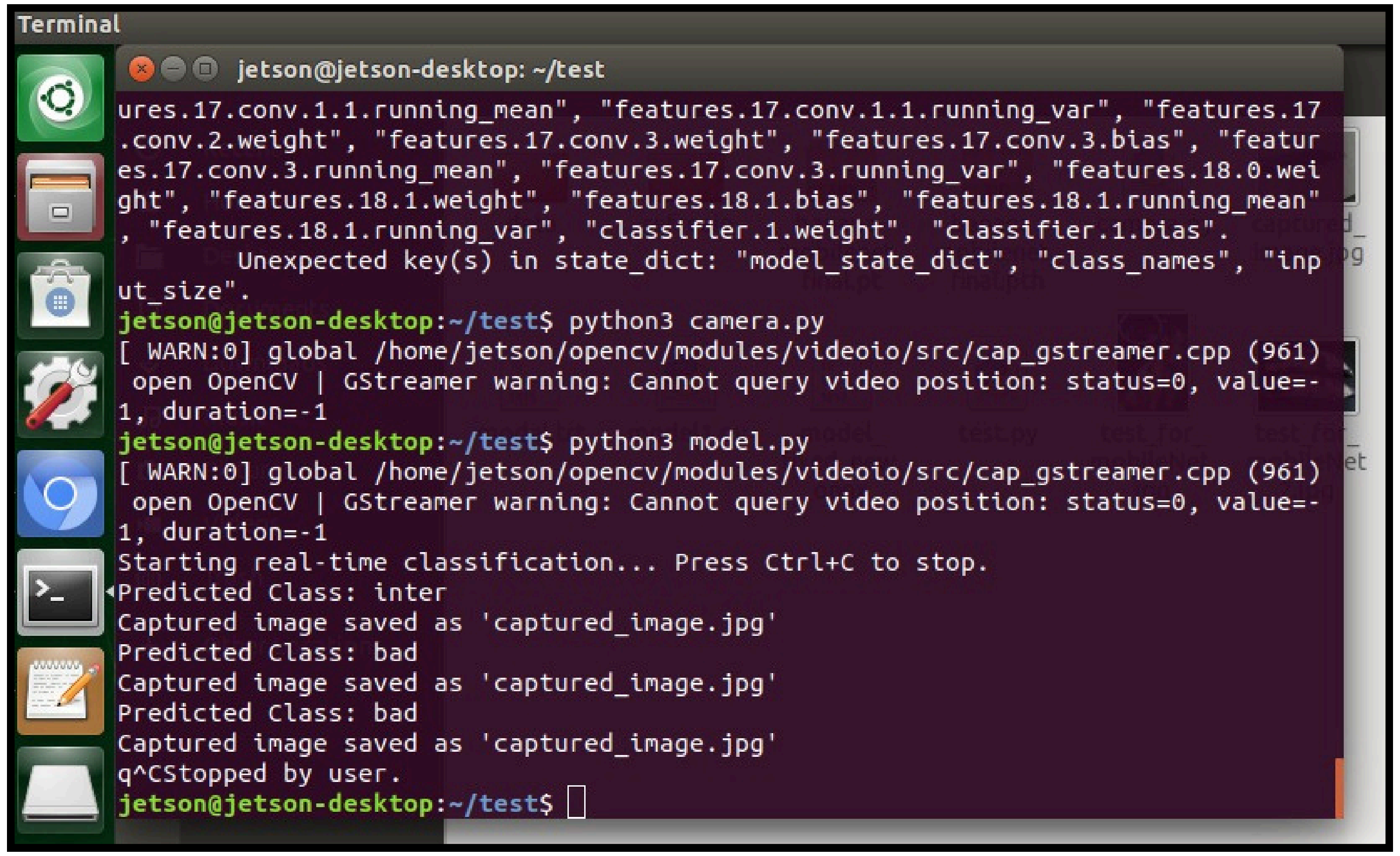
Overall System



Environment Setup for Jetson Nano-



Model Deployed on Jetson Nano-



Result-Bad

Sample image given through USB Camera

Conclusion

- The system integrates a conveyor system, Jetson Nano, and deep learning model(MobileNetV2) to enable real-time, high-precision sorting of bananas by quality.
- The system’s modular design—including a camera, LED lighting, and pushing mechanism—ensures efficient, automated sorting and adaptability for various grading needs.

Future Scope-

- Extend the system to support multiple fruit types.
- Add automated sorting using actuators or robotic arms.
- Integrate cloud for remote monitoring and analytics.
- Develop a mobile app for on-field quality checks.
- Optimize model for faster real-time performance on edge devices.

Challenges Faced-

- Python and PyTorch versions on Jetson Nano were outdated.
- Jetson Nano was difficult to set up with other hardware.
- Models trained on new systems didn’t run easily on Jetson Nano.
- Hard to sync detection with the moving conveyor.
- Real-time detection needed fine-tuning to avoid delays.
- AC motor for the conveyor belt was hard to control.

Bibliography

- Chen, Y. Zhang, and S. Wang, "Determination of Fruit Quality by Image Using Deep Neural Network," in Computers and Electronics in Agriculture, vol. 192, pp. 106689, 2022, doi: 10.1016/j.compag.2022.106689.
- Patel, R. Singh, and M. Kumar, "Fruits and Vegetables Quality Evaluation Using Computer Vision," in Journal of Food Engineering, vol. 301, pp. 110626, 2021, doi: 10.1016/j.jfoodeng.2021.110626.
- Johnson, M. Liu, and E. Davis, "Deep Fruit Detection in Orchards," in IEEE Transactions on Geoscience and Remote Sensing, vol. 55, pp. 1250-1261, 2017, doi: 10.1109/TGRS.2016.2618362.
- H. Nguyen, A. Zhou, and C. Patel, "Fruit Quality Assessment with Densely Connected Convolutional Neural Network," in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, pp. 1124-1135, 2022, doi: 10.1109/TNNLS.2022.3154578.
- J. Lee, T. Wong, and H. Zhang, "Enhancing Fruit Quality Detection with Deep Learning Models," in IEEE Transactions on Image Processing, vol. 33, pp. 567-579, 2024, doi: 10.1109/TIP.2024.0123456.

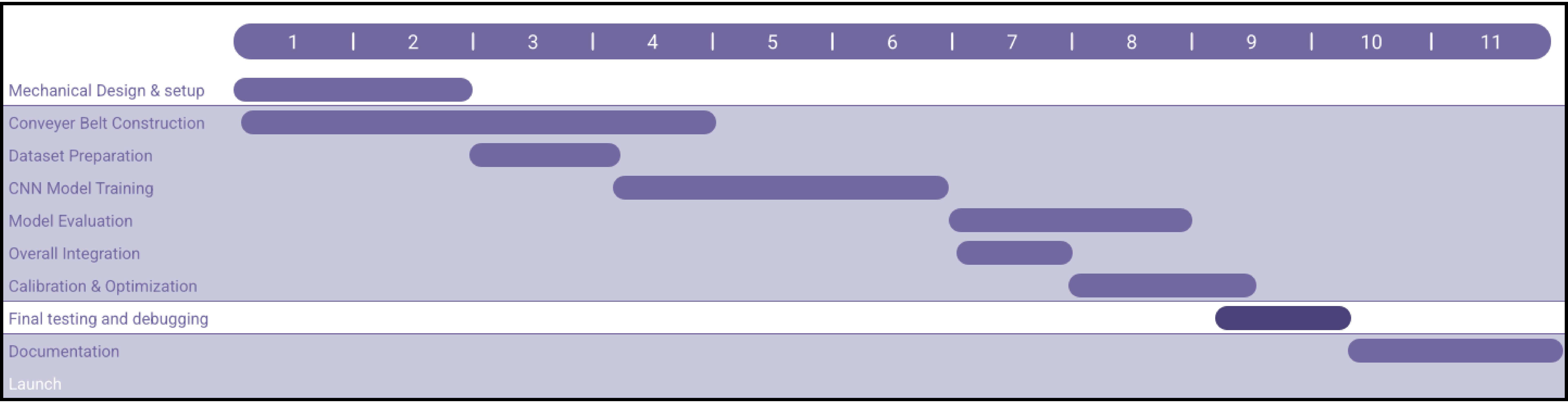
- A. Kumar, S. Singh, and P. Patel, "A General Machine Learning Model for Assessing Fruit Quality Using Deep Image Features," in IEEE Transactions on Computational Agriculture, vol. 10, pp. 1123-1134, 2023, doi: 10.1109/TCAG.2023.3105678.
- A. Sharma, R. Verma, and M. Singh, "A Novel Transfer Learning Approach for Pomegranate Growth Detection," in Journal of Agricultural Informatics, vol. 8, pp. 231 241, 2020, doi: 10.1109/JAI.2020.4025671.
- Patel Vamshi Sharma, P. Gupta, and N. Mehta, "Fruit Quality Recognition Using Deep Learning Algorithm," in International Journal of Computer Applications, vol. 175, pp. 20-27, 2021, doi: 10.5120/ijca2021912239
- Kumar, R. Patel, and M. Singh, "A General Machine Learning Model for Assessing Fruit Quality Using Deep Image Features," in Journal of Machine Learning Research, vol. 24, pp. 123-135, 2023, doi: 10.5555/12345678.
- Lee, T. Wong, and H. Zhang, "Enhancing Fruit Quality Detection with Deep Learning Models," in IEEE Transactions on Image Processing, vol. 33, pp. 567-579, 2024, doi: 10.1109/TIP.2024.0123456

Work Division

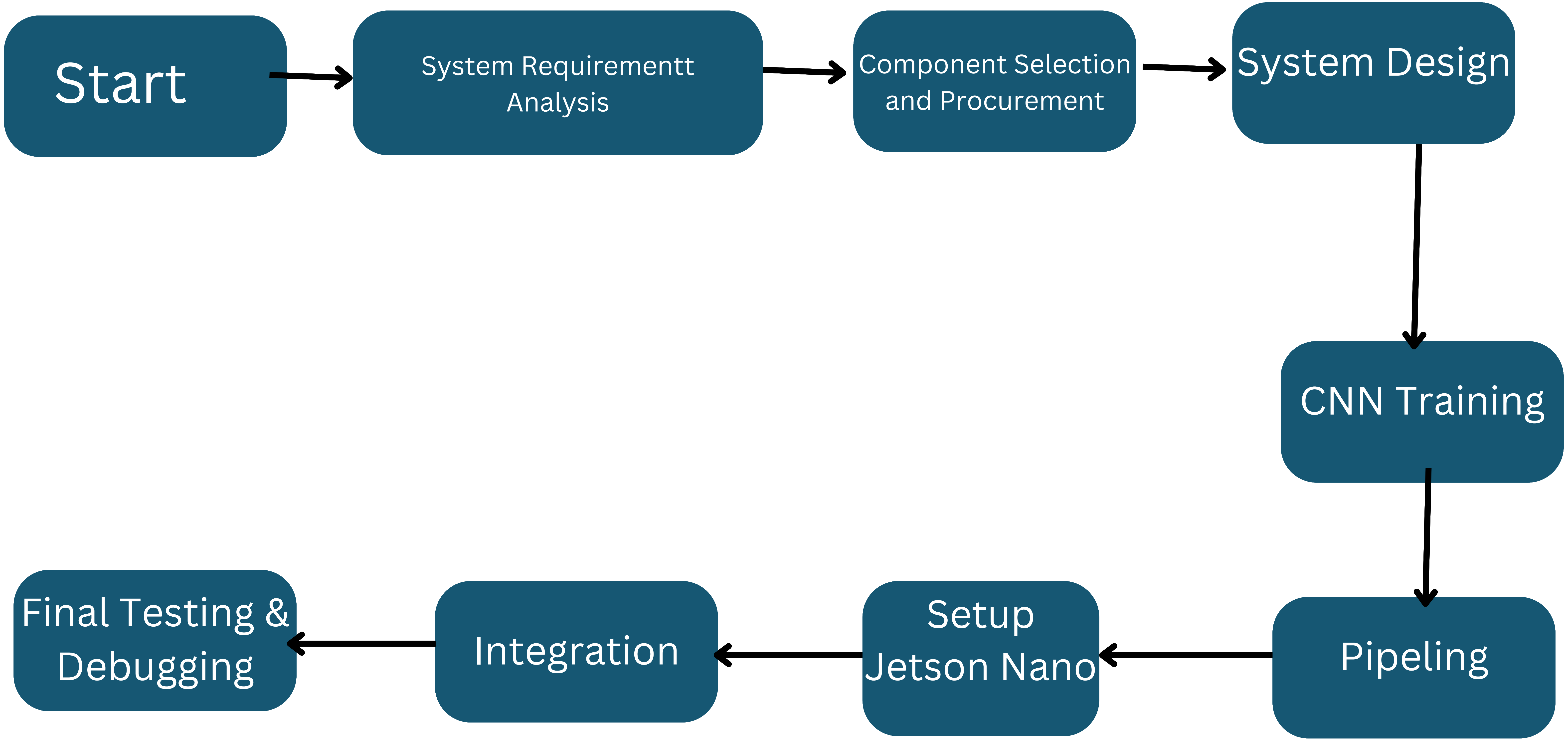
TASK	ASSIGNED TO
Presentation and Demo	
Define Goals	Team
Presentation & Demo	Team
Hardware	
Research , Planning and Mechanical Design	Muhammed Hadhi V M
Making small prototype for testing	Muhammed Hadhi and Adithyan
Assembly and Integration	Muhammed Hadhi and Adithyan
Testing Conveyer Belt	Muhammed Hadhi V M
Setting up the Jetson Environment	Rohith M S
Setting up the environment for real time analysis	Ruben
Setting up the sorting mechanism	Muhammed Hadhi V M
Optimizing the ML model into TensorRT format	Rohith M S and Ruben
Testing and Validation	Muhammed Hadhi V M

Software	
Dataset collection and augmentation	Ruben Davis Saji
Develop CNN Model	Adithyan Manoj
Integration of models and testing	Ruben and Adithyan
Testing in Jetson Nano	Adithyan and Rohith

Gantt Chart-


1
2
3
4
5
6
7
8
9
10
11
Mechanical Design & setup
Conveyer Belt Construction
Dataset Preparation
CNN Model Training
Model Evaluation
Overall Integration
Calibration & Optimization
Final testing and debugging
Documentation
Launch

Pert Chart


Start
System Requirementt Analysis
Component Selection and Procurement
System Design
CNN Training
Pipeling
Setup Jetson Nano
Integration
Final Testing & Debugging

Overall Budget

Component	Quantity	Price
Jetson Nano	1	15000
DC Motor	5	1000
LED Strip	1	165
USB Camera	1	1200
Bearing	20	1200
OLED Display	1	200
Motor Driver	2	200

Component	Quantity	Price
Nylon Roll	1	300
Ultraasonic Sensor	3	360
PWM Modulator	2	400
Accessories(screw,bolt,thread,washer)		1500
Total		**21,525**

Milestone Completed

✓ M1: First conveyor belt plus imaging box setup completed
✓ M2: MobileNetV2 training completed (Banana quality classification)
✓ M3: Setting up environment for Jetson Nano
✓ M4: Deployment in Jetson Nano
✓ M5: Real time testing
✓ M6: Second Conveyer Belt Construction
✓ M7:Pushing Mechanism construction
✓ M8: TensorRT engine conversion

Milestone to be Completed

- ✓ M1: TensorRT deployment
- ✓ M2: Working of pushing mechanism
- ✓ M3: Combined working as overall system

THANK YOU