Object Recognition and Motion tracking for Robotic pick and place application

AIMLCZG628T DISSERTATION

by

SAI NIKHIL ABINAS MOHANTY
(2022AA05245)

Submitted in partial fulfillment of
M. Tech. – Artificial Intelligent and Machine Learning

Under the Supervision of

(Mr. Arvind Raju)

(Principal Engineer, Intel, Bangalore)

BIRLA INSTITUTE OF TECHNOLOGY &

SCIENCE PILANI (RAJASTHAN)

(July 2024)

Table of Contents

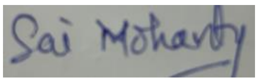
ABSTRACT

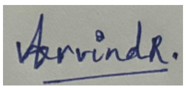
In modern manufacturing and logistics, automation plays a critical role in enhancing efficiency, accuracy, and speed. One of the key components of automation is the ability of robotic systems to handle and manipulate objects of various shapes and sizes. However, traditional robotic arms often struggle with identifying and picking different type of objects that are moving on a conveyor belt, especially when these objects vary in shape and orientation. This challenge necessitates the integration of advanced computer vision techniques to empower robotic arms with the ability to dynamically identify and pick moving objects.

The dissertation would provide a comprehensive study and implementation on leveraging computer vision, video analytics techniques to enable a robotic arm to pick up various shapes of moving objects on a conveyor system. The project will focus on using depth camera as an input and will develop an image pre-processing and feature extraction algorithm for feature extraction and image modification best suitable for our use case.

The project will Self Supervised model for object shape recognition and motion tracking for unlabeled videos to predict shape and location of the object. The meta data such as location of the object, speed will be published to a ROS2 environment for robotic system consumption.

To make this implementation production ready, the project will also use some of MLOps techniques during development of model such as quantization, model packaging, containerization of solution, model meta data store. Different model performance metrics will be used to evaluate the solution.





Signature of the Student

Name: Sai Nikhil Abinas Mohanty

Date: 26.07.2024
Place: Bangalore

Signature of the Supervisor

Name: Arvind Raju

Date: 26.07.2024
Place: Bangalore

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

Abbreviations	Meaning
ROS	Robot Operating System
MLOPS	Machine Learning Operations
ML/DL	Machine Learning/Deep Learning
BB	Bounding box
2D	2 dimensional
PCA	Principal Component Analysis
GPU	Graphical process unit
UI	User Interface
LAN	Local Area Network

LIST OF TABLES

LIST OF FIGURES

1. Chapter 1: Introduction

1.1 Problem Statement

Automatic pick and place of a moving object in a conveyor belt by a robotics system is a wide industrial use case that has been popular in industries like warehouse, material management, automotive manufacturing. The solution of this use case includes conventional methods such as mechanical and sensor-based systems.

Automatic pick-place using Sensor based method:

In this method the speed of the conveyor belt is synchronized using encoders. It is a close loop system where the real-time data from encoder gives the information on conveyor belt's speed and position of the object. This data is then sent to Robotics system using which the robotics arm acts.

This implementation has few advantages such as:

- The implementation is simple and time efficient.
- It is cost-effective.

These conventional approaches have limitations such as:

- The position of object w.r.t encoders should be very much precise.
- The encoder value can vary with time, which can be due to mechanical wear and tear, or sensor offset.
- It largely depends on the object shape. Any change in object shape the system might fail.

1.2 Objective/Goal of the Project

With recent advancements, computer vision-based techniques are used for object pick and place automation. A major part of this approach is to develop a model that can detect the object and predict the motion.

The object detection models can be developed using supervised and unsupervised technique.

Current Setup/Solution:

The current model available in our lab is a supervised model that is trained to detect a specific cube used to pick and place on a conveyor system by the robotic arm. The setup uses a RealSense camera which takes the video of the conveyor with object from the side view of the conveyor with object. So, in this design, the distance of the object from the camera becomes important to measure the distance and the coordinates. Hence, an additional parameter depth is also measured and calculated to accurately pick the object.

Another problem in this approach is as this is a supervised model that is trained on the specific type of cube used in the lab so the model cannot predict the object of any other type. This makes the approach not scalable and also time complex as the code needs to find out depth of the object as an additional parameter.

The objective of the project is to create a model and overall new approach that can be scalable and can detect object from 2D parameters. The approach should be able to detect any moving/static object irrespective of the shape of the object.

The goal is to solve the problem by utilizing Computer Vision, Video Analytics and ML/DL techniques. Instead of creating only the algorithm or model, the dissertation focuses on the developing an end-to-end application that can be used by an end user. Hence, MLOps techniques are used in the project to make the solution as a package solution and develop surrounding modules that can interact with the model. The project also aims to create a demo solution where the data predicted/generated by the models can be published to a ROS environment, which can be used by any robotics arm for pick-n-place application.

1.3 Project Scope

The scope of the project is to solve or create a new approach for the problem faced in the lab environment (Conveyor with robotics setup).

- The focus is to primarily resolve the limitations in the lab setup and work in future to make it compatible for other environments.
- The scope limits till we publish the meta data and frame to ROS environment of Robotics system.
- Scope is limited to only developing model registry and offline data store as part of MLOPS.
- Assumption is each image frame consists of one moving object, which is our case of interest.

2. Chapter 2: Literature Review

Paper1: https://www.jetir.org/papers/JETIR2202153.pdf

The paper describes image processing algorithm and the steps involved in the algorithm to detect an object. It uses edge detection method to detect an object and a CNN model to classify different objects.

Limitations:

Limitation of this approach is it detects all type of objects in the frame, which is not useful in our problem statement. If this approach is used then the model will detect all type of objects even which are not necessary along with the object of our interest, which is a cube here. It is difficult to classify the objects and find out the cube out of all detected objects.

Also, the paper does not talk about getting the motion of the object, which is a necessary parameter for a robotic arm to operate precisely.

Proposed Solution:

Instead of only detecting the objects in a frame, the proposed approach is to find out the moving object in the frame. As our setup always has one moving object at a time so detecting a moving object eliminates any foreign element/object or any static object that are part of the frame.

Paper2: https://www.ijert.org/motion-detection

This paper describes the methods used for human motion detection, human modelling and human activity tracking.

The paper uses an efficient method background subtraction for motion detection.

In addition to the flow described in the paper, we will be using additional image processing techniques such as edge detection and smoothing before we go ahead with motion detection for making it robust and make it work for all generic objects.

We will also use additional steps such as speed, orientation of the object of the motion object.

3. Chapter 3: Research Methodology

3.1 Design Solution

The robotics setup in the lab is a distributed system. The design focuses on developing the algorithm that will be light weight and faster enough to push the data to robotics system via LAN/5G. Hence, the focus is to choose a model/algorithm that will be fast enough to detect the model and can serve the purpose of object detection and motion detection of the specific setup that is available in lab.

In general, we need 3 parameters of the moving object for a robotic arm to precisely pick from the conveyor system and they are height, width and depth.

Existing Solution in the lab:

Current solution runs a model that is specifically designed/trained for picking up the specific size of cubes that are our object of interest now. The camera position is a side view angle and hence it needs to capture depth of the object for robotic arm to precisely pick the object, which is an additional logic to be run.

The above two are the drawbacks of the current solution as the model cannot be run for any kind of shape of the objects. Also, calculating the depth of the object also puts an overhead to the fast execution of the logic.

Proposed Design Solution:

- Proposed solution is to develop a generic model that can detect object and track the motion of the object irrespective of the shape of the object.
- To optimize the overall code for faster execution. Hence, position of the camera is changed from side view to top view. This change in angle removed the requirement to calculate/predict depth of the object.
- Aim is to create a packaged solution and make the model as a microservice.

The proposal is to create a unified solution such as the model as a microservice so that it can be readily used without any difficulty of compatibility in any number of different conveyor systems in a factory floor.

The final output both image and meta data associated is also needed to store in a datastore to use it for reference or to measure accuracy of the model.

The need is to also do a version management of the model to store it in the model registry as part of MLOPs flow.

3.2 Setup Comparison

Below is the setup block diagram for current solution. In this current setup, the camera is positioned to have a side angle view of the object. Edge device is a system where the camera stream is fed, and the model is run. The edge device is connected to the Robotics arm setup via LAN network. For fast communication, CAT8 communication is used between them. ROS environment is setup in both Robotics arm setup and edge device.

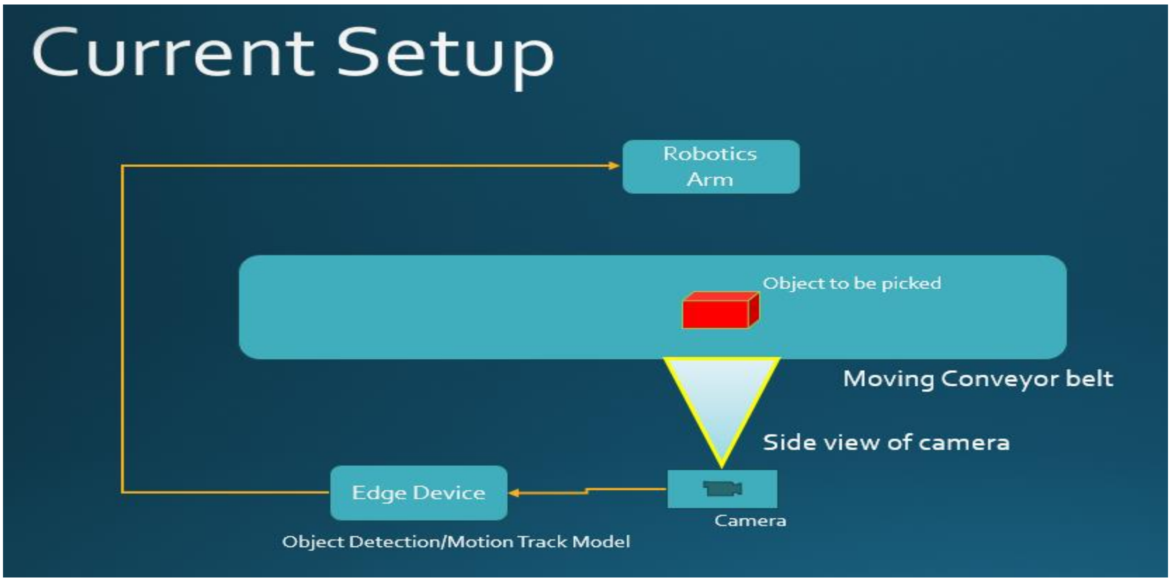


Figure 1: Current Setup Diagram

In the new setup the difference is the camera position which captures the top view of the object instead of side view that helps us to eliminate calculation of depth of the object.

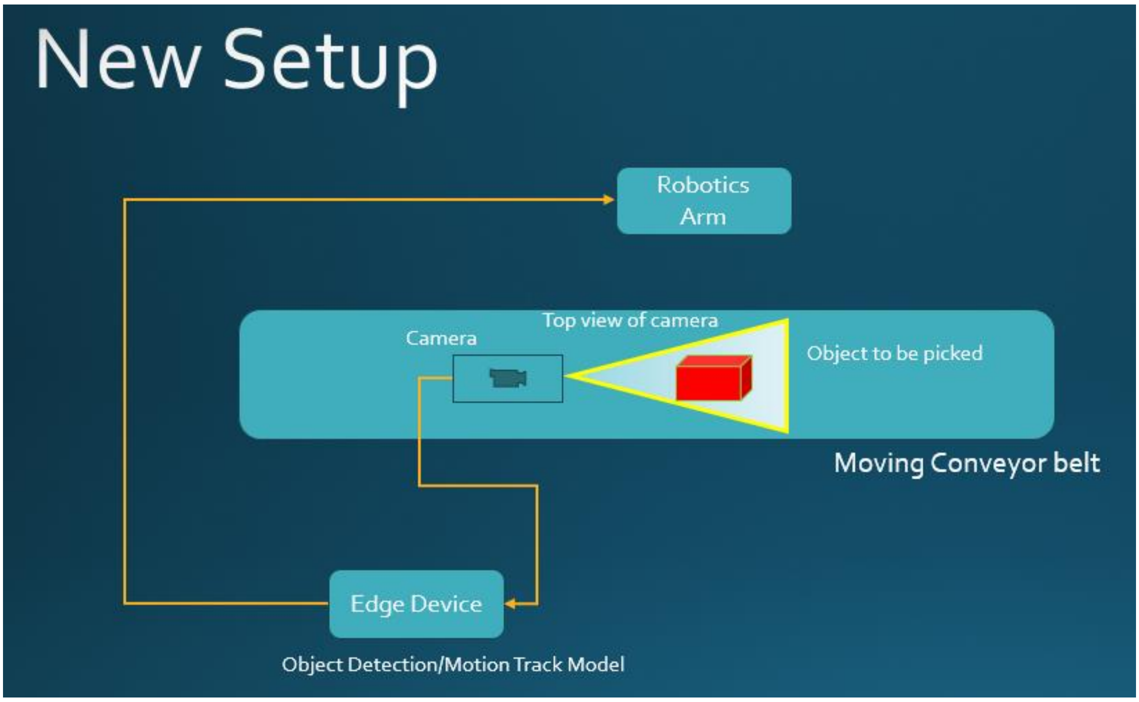


Figure 2: New Proposed Setup Diagram

Advantage of new setup:

- The new setup need only 2D coordinates to be predicted and send to robotics setup. Because the distance of the object from the camera always remains same so the depth of the object always remains constant.
- As we need to predict the speed of the object to have a synchronization of the belt with object. Current approach uses an encoder to calculate the speed of the conveyor belt and send it to robotics setup. Instead, we are using motion tracking model to predict the speed of the object which eliminates the need of another device, which is encoder for the conveyor belt.

3.3 Data Collection for Experiment

This is a use case where we do not want to train the model with a specific type of labelled object. Hence, simulated data is generated using python script.

There are two types of dataset:

- First one is to create a synthetic data or simulated data i.e conveyor with object using python script. The custom python script generates images with conveyor system with specified number of objects in it at different positions. Then a video file is created with using those image frames. This will be used while developing the model and as we do not have real data set, this serves the purpose of testing the model during development phase.

- Second dataset is a video taken of real setup i.e conveyor system with different type of objects in the lab. This will be used to feed the model once developed and validate the model for its accuracy and efficiency.

3.4 Research on Methodology and Approach

3.4.1 Study on Precise pick-n-place of object

For a robotic arm to precisely pick any moving object there are 3 parameters which robotic system show be aware of.

- One is the current position of the object on the belt. In our case only height and width are sufficient along with the center of the object as we are using top view so 2D coordinates can serves the purpose.
- Average speed of the moving object. Here we need to measure the speed at which object is moving. This can be done is two ways.
  a) First one is to measure the speed of the conveyor system, which is by adding an instrument called encoder to the conveyor system.
  b) Second one is to predict the speed of the moving object by comparing subsequent frames comparison and calculate the average speed of the object.

  We will use the second approach to predict the speed, which will eliminate the use of instrument and model can predict the motion.
- Third one is to find out the angle of the object so that robotic arm can be moved accordingly while picking up.

3.4.2 Study on Object Detection and Motion Tracking

For the object detection and motion tracking, the solution proposes to use image processing and motion detection algorithm to detect the object. The reason for this approach is we want to build a generic model where the background always remains same in a frame, but the type of objects can change. Hence, having a model that is supervised and trained with a specific object say cube may fail if we introduce an unknown object with different shape.

Hence, proposal is to use different image processing and motion detection techniques.

There are two model files or algorithm created for object detection with two different approaches.

1. The algorithm uses only image processing techniques such as Smoothing and edge detection to detect the boundary of the object and then find out the coordinates, orientation of the object.

   The approach is simple and effective in many of the cases. While testing with different static objects. But the limitation of this approach is if the coverage of focus of camera changes such that the camera captures other static objects such as some part of conveyor system (e.g: control buttons, motor system) then the model detects them also as an object. Hence, it is then hard to distinguish which is our object of interest.
2. This limitation of point 1 brings us to the different approach where we want to distinguish multiple detected objects from the real object of interest. Here comes the introduction to predict the movable object in the frame. As we know the other objects surrounding the belt that camera captures are static objects and does not change their position w.r.t time. The only moving object is on the conveyor belt, which is our object of interest. Hence, we detect the object in motion and from there we filter all other foreign objects that we consider them as noise. Then we predict the coordinate, orientation and average speed of the object.

Hence, we will use second approach for object detection and motion tracking.

3.4.3 Study on implementation of MLOPs techniques

For our solution, as we do not have labelled dataset, so we do not need to have a database for storing image dataset for offline training of the model. Here we are also not training the model with labelled data, so we are only focused on implementing below flows to implement MLOPs in our solution.

1. **Datastore:** We will implement a datastore module that will store both annotated object detection image data and the meta data associated with each frame. For the image dataset, we will be using minio database that will store the images as

blob. For image meta data, which will contain current position, center of the object, angle of the object and speed of the object for each frame in a relational database. For that we will be using influxdb to store meta data.

2. **Version Management:** For storing and version management of the models, we will be using MLFLOW, an open-source model registry tool. This provides basic model registry features such as versioning of model files, model searching and model packaging. These features are sufficient to serve our purpose.
3. **Containerization:** To make this solution easily deployable in any environment, we want to package the model into a docker image with all dependencies installed. Then this solution can be deployed as model as a microservice.
4. **Metrics measurement:** <TBD after mid-sem >

3.4.4 Study on implementation of Web Application

As discussed in abstract review, the suggestion was to develop a web application from where the user can be able to upload image or video file and run the model.

Here we studied to use Flask with Django framework for development of basic web application. This will have basic features such as upload image and video files buttons, button to run motion tracking model, a display placeholder to view input image/video file and the final detected object frame. A display placeholder to view metadata for each frame.

3.4.5 Study on integration of model output to ROS environment

The solution needs to create a ROS environment where the model is run and the model output is needed to publish to ROS environment to a topic, which the robotics setup will listen and fetch the meta data.

We will be using ROS2 environment in an ubuntu platform. We will create a ROS2 publisher script in the system where model is running. In the other hand, we will use python to create a subscriber script in the robotics setup. The end goal is to verify if the meta data gets published to the subscriber system.

4 Chapter 4: System Design and Implementation

4.1 System Requirements

The solution is developed in an Intel Core Edge platform with Ubuntu 22.04. This does not require an additional GPU or accelerator. The system should have below minimum requirements:

Requirements	Minimum Configuration
Platform	Intel Core Platform 11th Gen
RAM	32GB
HDD	256GB
OS	Ubuntu 22.04

Table 1: System Requirements

4.2 Architecture Diagram

Below shows the architecture diagram of overall solution. The solution shows Node1 as the system where all the logic runs. It will have 3 parts.

- Web Application
- Detection and tracking Model as a service
- MLOps

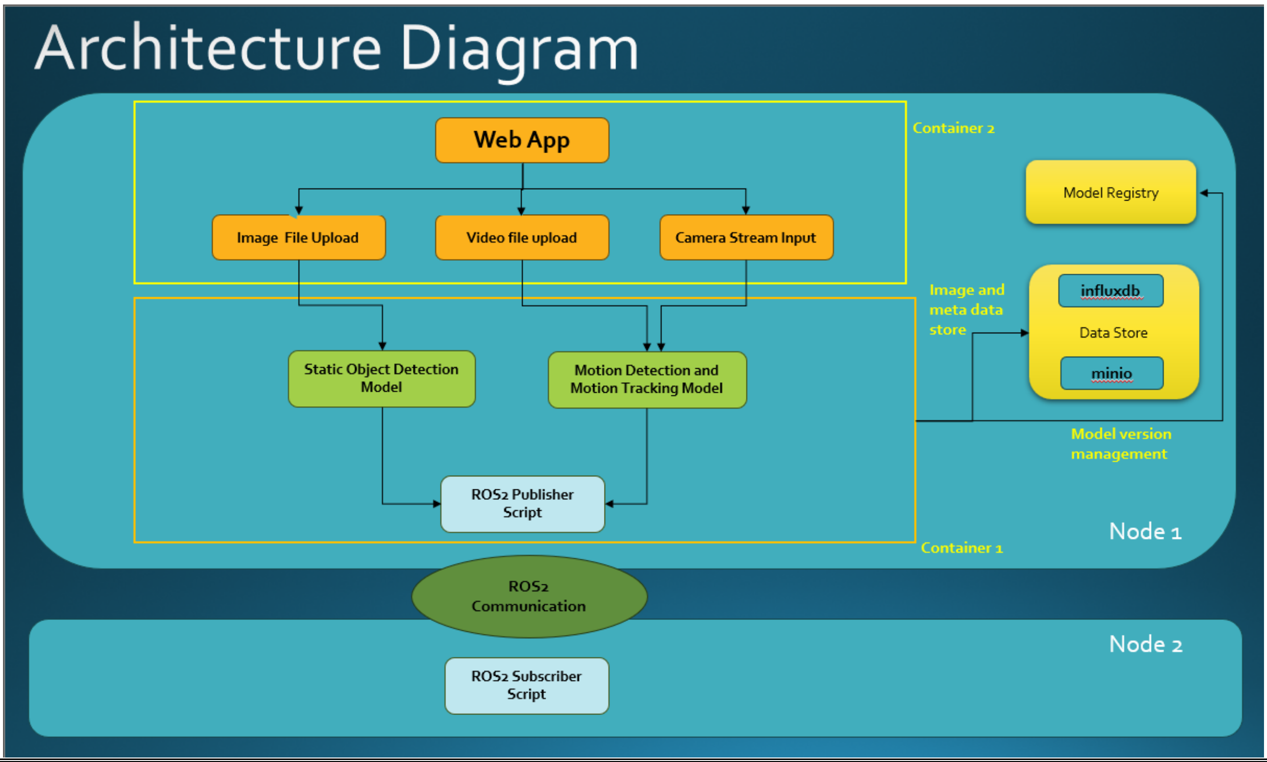


Figure 3: Architecture Diagram

Out of all the blocks, influxdb and miniodb are open-source components. Rest all are designed as part of the solution. Each component is described in the implementation section.

4.3 Implementation

The implementation is broadly divided into below categories:

- Developing script to generate dataset for testing
- Creating a User Interface application
- Development of Object Detection and Motion Detection model
- Developing ROS environment and publisher/subscriber logic
- MLOPs implementation

4.3.1 Generation of dataset

As this is a use case where we do not have labelled dataset and we do not want to train the model with some specific dataset, for testing the model during development, a python script was developed to create sequence of images/frames what will mimic top view of conveyor belt with an square object on it. The sequence of images will have different positions of same object to mimic the motion of the object. The position change of the object in sequence of object is kept uniform to show the conveyor has a fixed speed.

Then a video file was created using ffmpeg tool using those sequence of images.

Below shows the sequence of frames generated from the script for testing purpose.

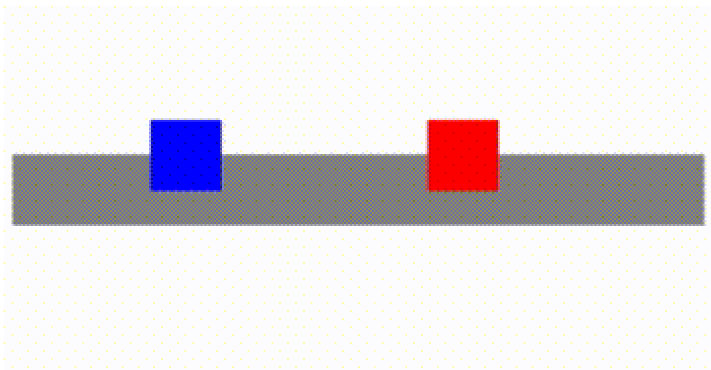


Figure 4: Sample generated frames for test dataset

4.3.2 Implementation of User Interface Application

The purpose of creating a user interface is to enable user to provide their choice of input, run the model and then view the displayed output image/video along with the predicted meta data.

The user interface has input fields as Image file input, video file input and RTSP camera stream URL input. User can upload an image file in .jpg or .png format or user can upload a video file or can enter RTSP stream URL for which the user wants to run the model.

Hence, the project supports running the model in an image file or video file or with camera (currently RTSP camera stream support is provided.).

The UI also provides button to run the object detection/motion tracking model. The framework has two python model files, one for static object detection model for image file and second for motion detection and tracking model for sequence of frames or video file/camera stream input.

The UI has placeholder to display input frames, post-processed image after detection and display meta data such as location of the object, orientation of the object, average speed of the object in motion.

Python Django framework is used with Flask to develop the web application running as server.

Below is the block diagram of Web Application Page:

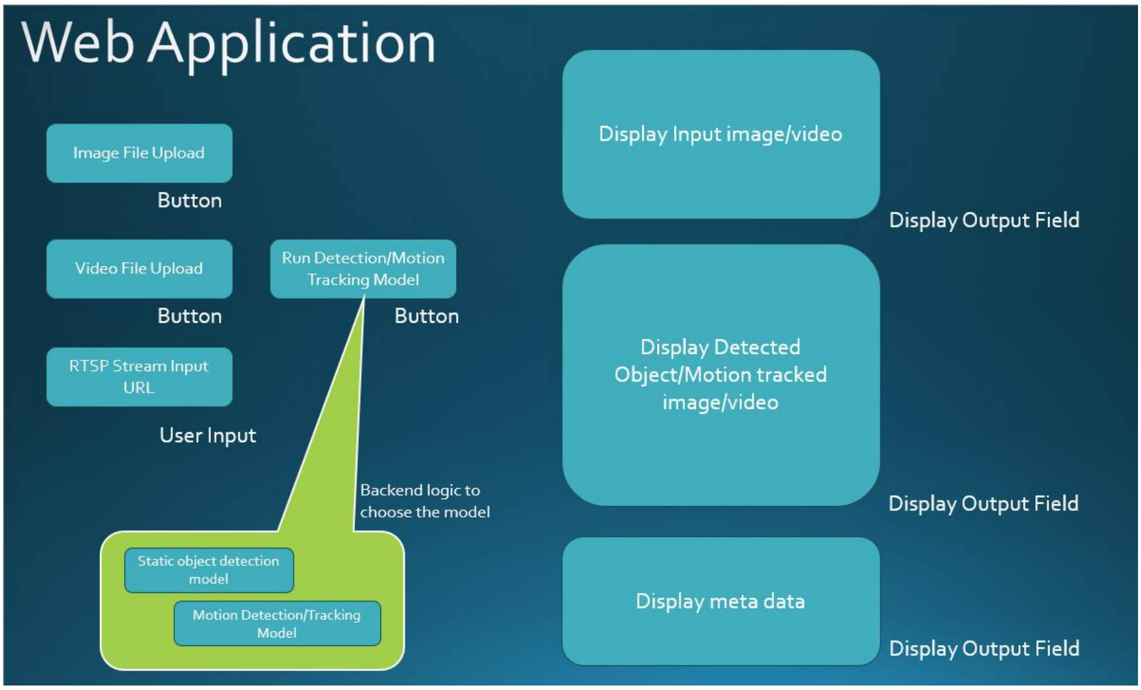


Figure 5: Web Application Page View

4.3.3 Object Detection/Motion Tracking Model

Two models are developed using python language. One is Static Object Detection model for a single image input and other one is motion detection model for sequence of frames.

Static Object Detection Model:

In this model, image file is used as an input source and then the image processing techniques are used -> conversion of image to 2D by converting to gray scale.

Image smoothing is done by using gaussian blur.

The next part is to find out the edges to locate the object. Hence, Canny edge detector is used to find out the edges. Contours are then detected and which in turn is used to calculate bounding box coordinates, center of object, orientation of the object using PCA.

Resizing is done to the annotated image to focus on the detected object.

Block diagram of Static Object Detection Model:

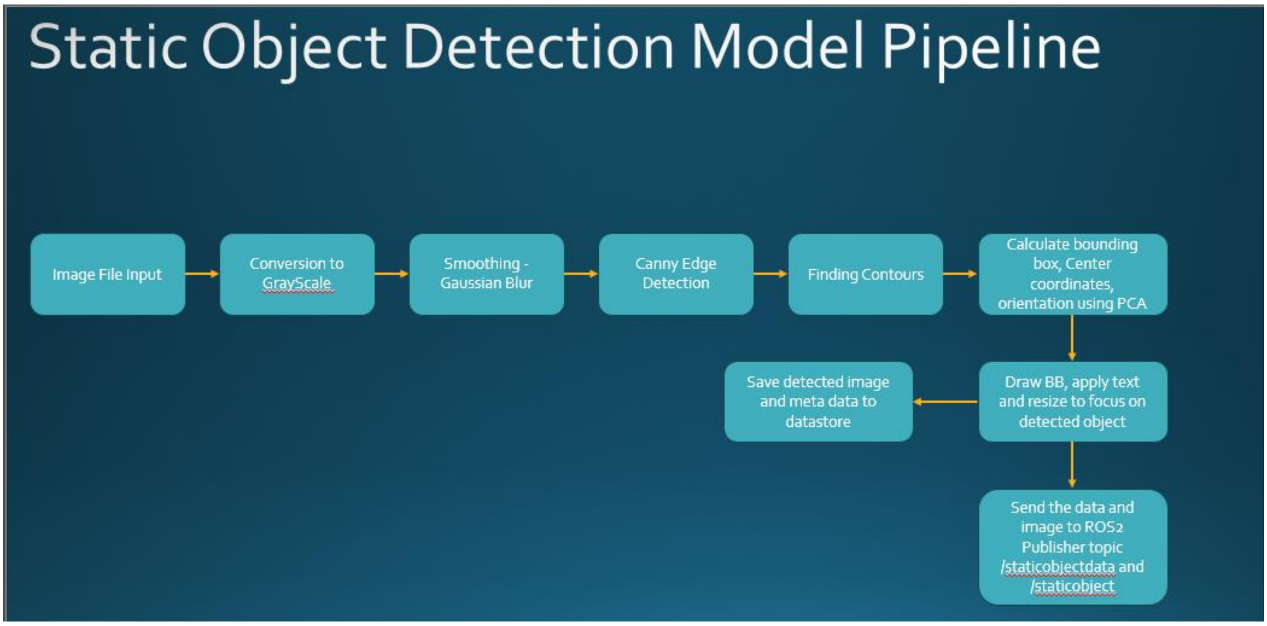


Figure 6: Static Object Detection Model Algorithm

Motion Detection and Tracking Model:

There are two methods explored to do motion detection.

- Background Subtraction
- Frame differencing

The problem with frame difference method is any gradual change in background may lead to reduced accuracy of detecting the object. It works very well where there are significant changes in subsequent frames.

Whereas the advantages of background subtraction method is it works pretty well where the camera position is static and the background is also constant.

Hence, background subtraction method is used in our project.

Python, opencv and numpy is used here to develop the model.

Below is the block diagram of developed Motion detection model flow:

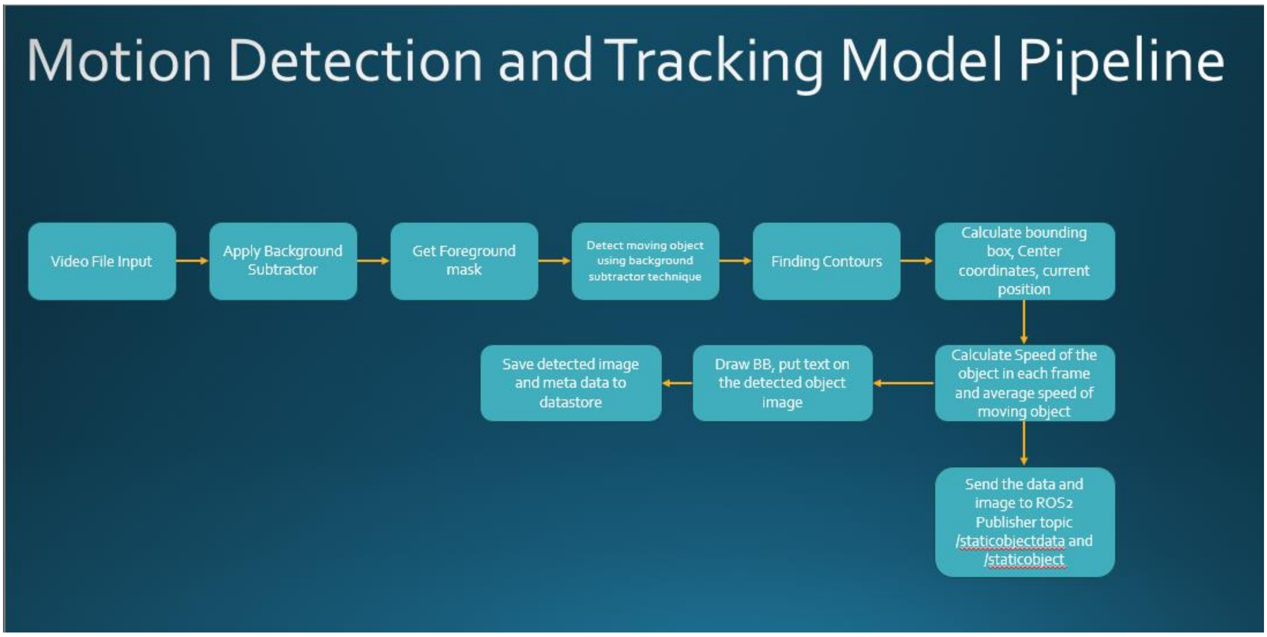


Figure 7: Motion Detection and Tracking Model Algorithm

4.3.4 Publisher and Subscriber Script for ROS2 environment

In this step, two ubuntu platforms are setup with ROS2 installed in it. Tested both the nodes communication via ROS2 with demo publisher and talker script.

Python publisher and subscriber script is to be developed with publisher and subscriber logic to send meta data in json format to Robotics setup via ROS2 protocol.

<To be implemented post mid-sem>

4.3.5 MLOPS Implementation

As part of implementing MLOPs into the project for making the solution deployable, three components need to be developed.

- Database with supporting blob and meta data storage
- Model registry implementation using MLFLOW APIs
- Containerization of the solution

<To be Implemented post mid-sem>

Below is the implementation to be done as part of MLOPS.

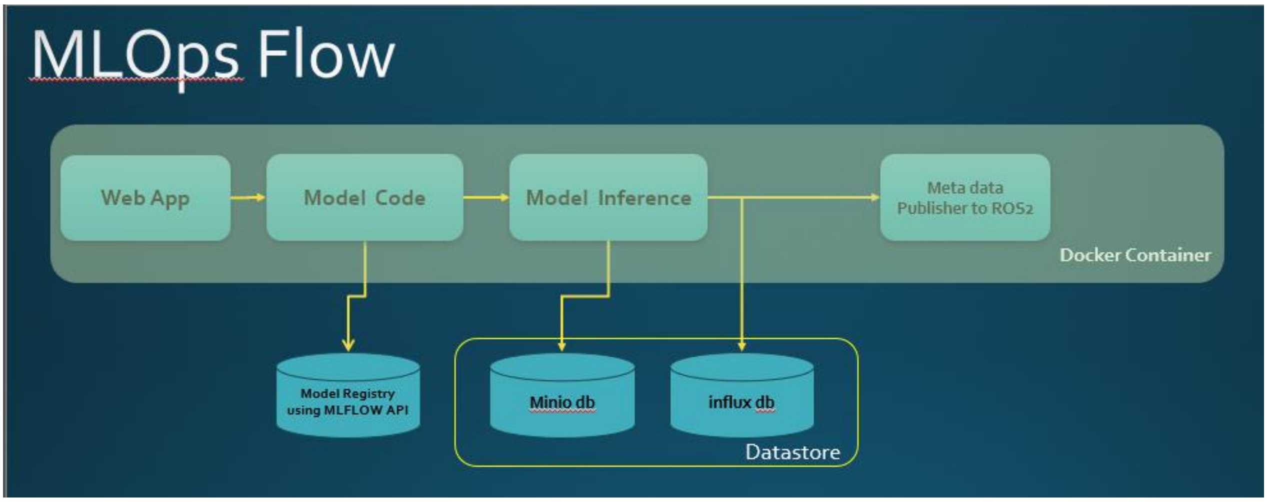


Figure 8: MLOPS Flow Diagram

5. Chapter 5: Plan of Work/Progress

Below table shows the plan of work and the progress so far.

Sl.no	Description of Work	Status
1	Build and prepare ROS2 environment in two nodes. - Install ROS2 environment in Ubuntu22.04 - Test demo talker and listener in the node	Completed
2	Install and hands-on with gazebo simulator	Completed
3	Development of data set generation script	Completed
4	Build Web GUI for uploading image/video file	Completed
5	Webapp for uploading image and video file	Completed
6	Development of Motion detection and tracking Algorithm	Completed
7	Development of Static Object Detection Algorithm	Completed
8	Initial Testing of the model with generated dataset and few sample images	Completed
9	Webapp button to trigger object detection model	In Progress
10	Webapp button to show annotated detected object	To be Started
11	Development of ROS2 Publisher and subscriber script	To be Started
12	Development of Datastore with minio and influxdb	To be Started
13	Development of Model Registry using MLFLOW API	To be Started
14	Development of WebApp screen to show annotated image and meta data	To be Started
15	Containerization of the solution	To be Started
16	Testing of the model with actual data in lab	To be Started
17	Measuring efficiency and accuracy of the model	To be Started

Table 2: Plan of Work and Progress

Apart from the above implementation work, major effort spent on studying the literature, various methods, exploring the code and testing the code during development.

6. Chapter 6: Results and Analysis

<To be completed>

7. References

The following are referred journals from the preliminary literature review.

[1] Qiguang Chen , Lucas Wan , Ya-Jun Pan , "Object Recognition and Localization for Pick-and-Place Task using Difference-based Dynamic Movement Primitives", IFAC PapersOnLine 56-2 (2023), https://www.sciencedirect.com/science/article/pii/S2405896323012454

[2] Chen, Q., Wan, L., Ravichadran, P., Pan, Y.J., and Chang, Y.K. (2022). "Vision-based impedance control of a 7-dof robotic manipulator for pick-and-place tasks in grasping fruits", 2022 CSME International Congress of Canadian Mechanical Engineering

[3] Dr.V.Raju, "feature extraction of object for a pick and place robot", https://www.jetir.org/papers/JETIR2202153.pdf

8. Semester Progress Evaluation Sheet

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
Work Integrated Learning Programmes Division
II SEMESTER 23-24

AIMLCZG628T DISSERTATION

(EC-2 Mid-Semester Progress Evaluation Sheet)

Scheduled Month : July

NAME OF THE STUDENT: SAI NIKHIL ABINAS MOHANTY

ID NO. : 2022AA05245

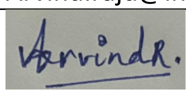
Email Address : 2022aa05245@wilp.bits-pilani.ac.in

NAME OF SUPERVISOR : Mr. Arvind Raju

PROJECT TITLE : Object Recognition and Motion tracking for Robotic pick and place application

EVALUATION DETAILS

EC No.	Component	Weightage	Comments *(Technical Quality, Originality, Approach, Progress, Business value)*	Marks Awarded
1.	Dissertation Outline	10%	Abstract focuses on solving a real problem in the project, has potential to scale it to other projects, covers real use case scenario	10%
2.	Mid-Sem Progress Seminar Viva Work Progress	 10% 5% 15%	Understanding of the problem and initial exploration of the solution was done well. Much time spent of learning the techniques and understanding the objectives. Implementation progress so far is up to the mark and on track to complete remaining tasks. Waiting for final test with real conveyor system and see the efficiency of the solution	9% 5% 14%

	Organizational Mentor	
Name	Arvind Raju	
Qualification	BE, Computer Science	
Designation & Address	Principal Engineer, Intel, Bangalore	
Email Address	Arvind.raju@intel.com	
Signature		
Date	26.07.2024	