Object Detection-based Automatic Waste Segregation using Robotic Arm

Azza Elsayed Ibrahim[1], Rasha Shoitan[2], Mona M. Moussa[3], Heba A. Elnemr[4], Young Im Cho[5], Mohamed S. Abdallah[6]

Computer and Systems Department, Electronics Research Institute (ERI), Cairo, Egypt[1, 2, 3]
Faculty of Computer and Software Engineering, Misr University for Science and Technology, October City, Egypt[4]
Department of Computer Engineering, Gachon University, Seongnam, Republic of Korea[5]
Informatics Department, Electronics Research Institute (ERI), Cairo, Egypt[6]
AI Lab, DeltaX Co., Ltd., 24 Namdaemun-ro 9-gil, Jung-gu, Seoul, Republic of Korea[6]

***Abstract*—Today's overpopulation and fast urbanization present a significant challenge for developing countries in the form of excessive garbage generation. Managing waste is essential in creating sustainable and habitable communities, but it remains an issue for developing countries. Finding an efficient smart waste management system is a challenge in current research. In recent years, robots and artificial intelligence have influenced a wide range of industries, especially waste management. This research proposes a waste segregation system that integrates the robot arm and YOLOv6 object detection model to automatically sort the garbage according to its type and achieve real-time requirements. The proposed algorithm utilizes the pros of the hardware-friendly architecture of YOLOv6 while keeping high detection accuracy in detecting and classifying garbage. Moreover, the proposed system creates a 3D model of a 4 DOF robotic arm by CAD tools. A new approach based on a geometric method is proposed to solve the inverse kinematics problem in order to precisely calculate the proper angles of the robot arm's joints via a unique solution with less computational time. The proposed system is evaluated on a modified TrashNet dataset with seven garbage classes. The experiments reveal that the proposed algorithm outperforms the other recent YOLO models in terms of precision, recall, F1 score, and model size. Furthermore, the proposed algorithm consumes approximately fractions of a second for picking up and placing a single object in its proper basket.**

***Keywords*—Smart recycling; inverse kinematics; object detection; 4 DOF robotic arm; YOLOV6**

I. Introduction

The amount of municipal solid garbage produced each year globally is 2.01 billion tonnes, with at least 33% of it needing to be handled in a way that protects the environment. Worldwide garbage is anticipated to increase worldwide to 3.40 billion tonnes by 2050, more than double the population growth over that time. Even if there are many ways to dispose of the garbage gathered today, the ecological system's sustainability and safety are nevertheless negatively impacted. Reusing and recycling as much trash as you can is thus the best option. In certain nations, the sorting process is primarily manual. Human sorters are manually stationed beside the material-transfer conveyor belts to identify the material type. One of its typical issues is that manual sorting mainly relies on visually examining the mixed garbage moving on the conveyor. A material surge might occur, leaving a sorter with insufficient time to understand all of the items handed to them. Moreover, health concerns, including skin difficulties, are unavoidable [1]. Therefore, manual sorting suffers from low productivity and increased health risks. Recently, robotic technology has replaced the time-consuming human garbage sorting system with an automated one. The robot is integrated with deep learning technology for detecting and classifying recyclable objects on the conveyer belt and picking and placing these objects in the appropriate baskets. Different convolutional neural networks (CNN) have been proposed for waste classification.

In [2], five deep learning architectures were applied for classifying Trashnet dataset, including: DenseNet121, DenseNet169, InceptionResnetV2, MobileNet, and Xception. Currently, Gondal et al. proposed a hybrid technique for smartly classifying real-time waste [3]. This technique applied to two machine learning methods: a multilayer perception and a multilayer convolutional neural network (ML-CNN). The former classifies the waste into metal or non-metal waste, while the latter specifies the class of non-metal waste (paper, plastic, rubber, cotton, and wood). The camera is positioned in front of a conveyor belt to take an image of each trash item. After image categorization, an automatic holder is used to pick up the trash item and place it into the assigned bucket. Although some of these CNN models achieve better classification accuracy, they are limited to classifying one object per image. If the image has many objects as the images captured from the conveyer belt, CNN model classifies the most dominant object in the image. On the contrary, object detectors are used in this case to both localize and classify various objects in the same image. Several methods are proposed for object detection, and their designs are based on two approaches: one-stage object detection and two-stage object detection. One-stage detectors are distinguished by their high inference speeds because they consider computing speed and combine object localization and classification to output. However, two-stage detectors are characterized by high localization and identification accuracy because they employ independent calculations for object localization and classification, increasing the accuracy and speed calculations. SSD and YOLO are models of one-stage detectors, and RCNN, Fast RCNN, and Faster RCNN are models of two stages

detectors. In waste recycling applications, speed is considered essential for picking up many objects in real-time. Therefore, most of the research in this application applies the one-stage detectors in their systems.

WEN MA et al. [4] propose a Lightweight Feature Fusion Single Shot Multibox Detector (L- SSD) algorithm for waste detection and classification. L-SSD is an enhanced SSD with a lightweight and a feature fusion module to improve its performance and detect waste with a small volume. The L-SSD is trained on a collected dataset consisting of five classes cardboard, glass, paper, plastic, and metal. Daniel Octavian Melinte et al. [5] fine tune the Single Shot Detectors (SSD) architecture with MobileNetV2 base network on the TrashNet dataset using appropriate training hyper parameters to be carried out on autonomous robot system.

Berardina De Carolis [6] detect and report the presence of abandoned waste through real-time analysis of video streams based on an improved YOLOv3. A new dataset containing four classes: garbage bag, garbage dumpster, garbage bin, and blob, is used to fine-tune the improved YOLOv3. Saurav Kumar et al. [7] apply YOLOv3 in the waste segregation application to detect six classes of trash objects (namely: cardboard, glass, metal, paper, plastic, and organic waste). Aria Bisma Wahyutama et al. [8] design a trash bin that is competent for automatically separating and collecting recyclable trash utilizing YOLOv4 object detection. The YOLOv4 model is installed on Raspberry Pi. As the trash object is detected, a servo rotates the trash bin cover to disclose the correct room for the user to throw away the trash. Additionally, the study [9] submitted a smart waste detection utilizing YOLOv3, YOLOv4, YOLOv3-tiny, and YOLOv4-tiny models. A nature-inspired searching strategy was applied to fine-tune the learning rate of YOLO structures. The results reveal that YOLOv3 provides the best results.

Anbang Ye et al. [10] propose a YOLO model with Variational Autoencoder (VAE) to increase the detection accuracy and decrease the model size for edge devices. The model has been trained on a generated dataset from the 2020 Haihua AI Challenge (2020 HAC), a garbage sorting competition. Andhy Panca Saputra et al. [11] propose YOLOv4 and YOLOv4-tiny for garbage detection and train the model on a modified version of the TrashNet dataset with a smaller number of classes and a higher number of images.

Deep Patel et al. [12] introduce a comparative study for applying five object detector techniques which are EfficientDet-D1, SSD ResNet-50 V1, Faster R-CNN ResNet-101 V1, CenterNet ResNet-101 V1 and YOLOv5M on a custom garbage dataset collected from the internet. The comparison results demonstrate that YOLOv5M has reliable and precise predictions. Sylwia Majchrowska et al. [13] localize and classify the litter through two networks, EfficientDet-D2 for localization while EfficientNet-B2 for classification. The algorithm is applied to new benchmark datasets, namely detect-waste and classify-waste merged from different datasets annotated similarly for the seven waste categories.

Other researchers, on the other hand, concentrate on the issue of designing and determining the angles of the joints of the robot arm manipulator. Indra Agustian et al. [14] propose the forward kinematics DH and the inverse kinematics Pseudoinverse Jacobian method to determine the right angle of each joint of the manipulator links. Adnan Rafi Al Tahtawi et al. [15] use inverse kinematics to design a small-scale three-degree of freedom (3-DoF) robot arm for a pick-and-place mission. Lately, the robot development time has been shortened, and the speed and quality of the robot design have been improved by designing robots based on SolidWorks 3D CAD [16]. Doo Sung Ahn et al. [17] present a platform that integrates Solidworks and Simscape tools for designing control algorithms of robot manipulators. Simscape Multibody imports 3D models and creates bodies, constraints, actions, and joints with parametrization by mathematical expressions described in MATLAB using data from SolidWorks.

Over and above, some researchers are directed toward integrating the robot arm with object detection techniques for waste segregation. Xuebin Yue et al. [18] propose a lightweight object detection model YOLO-GD (Ghost Net and Depthwise convolution) for empty-dish recycling robots to recycle dishes in restaurants and canteens automatically. The catch point coordinates of the various types of dishes are extracted using a catch point computation based on image processing. The target dishes are recycled using the coordinates by manipulating the robot arm. Qisong Song et al. [19] propose an improved YOLOv5 to achieve more precise positioning and recognition of objects for grasping robots using the wooden block image dataset. Jinqiang Bai et al. [20] present a moving pick-up robot that automatically moves on grass for garbage cleaning. The robot is designed based on a navigation algorithm that uses SegNet and ResNet to segment, classify, and localize objects. If the trash is detected, the manipulator picks it up and places it in the trash container. Jaeseok Kim et al. [21] integrate deep learning with the industrial robotic arm to classify garbage according to its material. First, the points cloud is processed utilizing the Kinect. Following this step, grasping tools on the robot arm are used to grab the objects. The items are then seated in front of an RGB camera, which categorizes them, based on their composition, using a modified LeNet model into two main classes: carton and plastic. Eventually, all the collected items are put in a box beside the manipulator.

Although all of the approaches mentioned above have made some progress in waste segregation, there is still the issue of satisfying high detection accuracy and real-time segregation simultaneously in practical applications. Thus, this research proposes a real-time waste segregation system for sorting garbage. The system is designed to integrate two modules: the waste segregation module and the robotic module. The waste segregation module exploits the advantages of the hardware-friendly architecture of YOLOv6 [22] for detecting and localizing the garbage with high detection accuracy and in real-time. In the robotic module, a robot arm is designed, and its design overcomes the inverse kinematics problem by accurately computing the angles of the arm's joints based on a simple geometric method. The advantage of this geometric method is to solve the inverse kinematics problem using a unique solution for each required joint configuration, which reduces the computational time and accelerates the response.

The robot arm architecture is created in CAD software (SolidWorks). The main contributions of this work can be summarized as follows:

1) Design a waste segregation system for categorizing garbage with high precision and in real time.

2) Build a garbage dataset consisting of 3217 images divided into seven classes: cardboard, glass, metal, paper, plastic, battery, and foam. These classes are usually found in trash bins.

3) Apply YOLOv6, which is characterized by its hardware-friendly design, low inference time and high detection accuracy for detecting and localizing garbage.

4) Design a 3D model of a 4 DOF robotic arm using CAD software and adopts a simple geometric method to calculate the angles of the arm's joints for picking up and locating the objects in their dedicated basket with smooth motion trajectories in a slight time.

5) Develop an automatic waste segregation robot system based on the designed robot arm and the proposed segregation system.

II. Proposed System Architecture

In this research, a waste segregation system is proposed to make the waste much easier to recycle, meaning less garbage goes to landfills and positively impacts health and the environment. The architecture of the proposed system consists of two modules: the waste segregation module and the robotic module. The waste segregation module comprises the YOLOv6 model for detecting the waste, while the robotic module picks up the objects according to their type and places them in their dedicated basket. The proposed system architecture is shown in Fig. 1. Moreover, the details of each module are described in the following subsections.

A. The Waste Segregation Module

This module is responsible for detecting various objects in the captured image and building a queue of information about each object identifying its current location and type. To perform this task, YOLOv6 is utilized. YOLOv6 is created for industrial applications with high-performance and hardware-friendly architecture. It makes different improvements to the network architecture and the training plans of the conventional YOLOv5 [23]. Conventional YOLOv5 consists of three main parts: backbone, neck, and head. The backbone primarily affects how well features can be represented, but because it performs most of the computational cost, its structure significantly impacts inferences performance. The neck is responsible for combining the low-level and high-level semantic features to construct a pyramid feature map. Then, these combined features are fed to the head, which consists of several convolutional layers, to predict the objects. YOLOv6 replaces the backbone and the neck networks in the conventional YOLOv5, which is designed based on CSPNet [24], with more efficient networks: EfficientRep as Backbone and Rep-PAN as Neck that is designed using RepVGG style [25]. The new network structure overcomes the drawbacks of increasing latency and decreasing the utilization of the memory bandwidth of the GPU. Second, YOLOv6 reduces the delay in the conventional YOLOv5 while preserving accuracy by designing an efficient and simplified decoupled head. Third, YOLOv6 attempt to improve the detection accuracy by improving the training strategies by using anchor free paradigm, SimOTA [26], as a label assignment method, and SIoU [27] and GIoU [28] as a bounding box regression loss function. In the next subsections, the networks architecture, and the training strategies of YOLOv6 will be explained in detail.

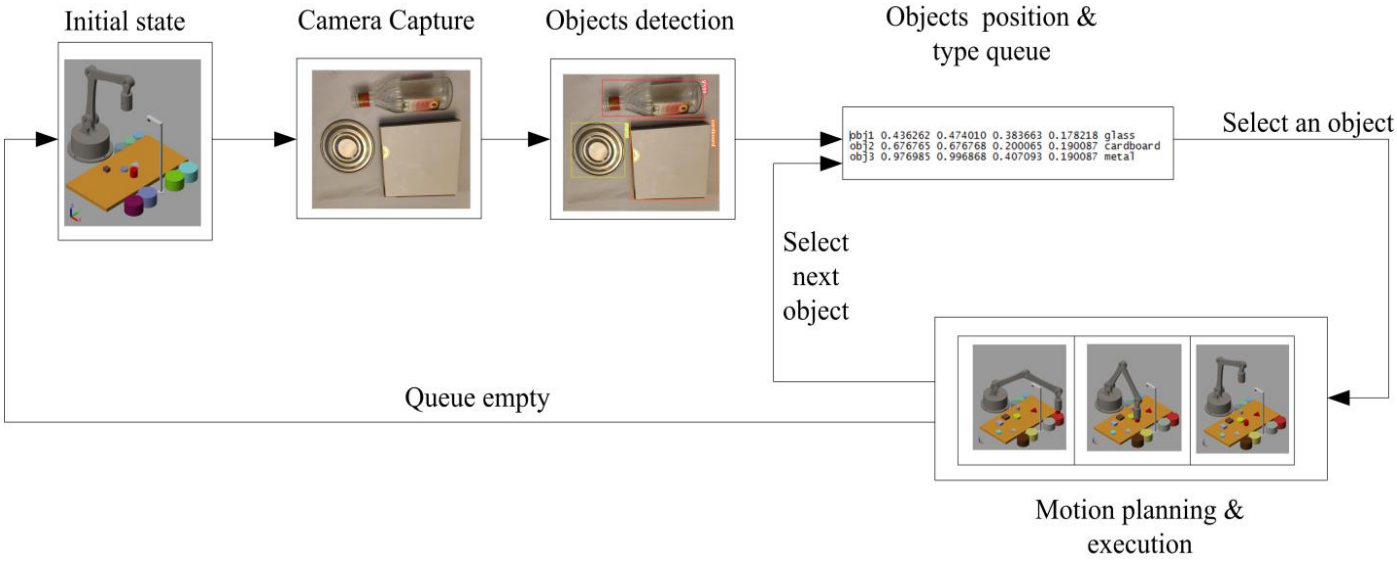


Fig. 1. The proposed waste segregation system architecture.

1) Backbone: The effectiveness and efficiency of the detection model are significantly influenced by the backbone network's design. EfficientRep is designed as a backbone in YOLOv6 to efficiently exploit the computational power of the hardware architecture to reduce the inference latency. The EfficientRep is designed based on the RepVGG structure to utilize the pros of multi-branch topology within the training process for achieving improved classification efficiency. However, this structure avoids the inference latency of the multi-branch topology by using the re-parameterization structure in the inference process to fuse the multi-branch into a single convolutional layer by converting its parameters within the deployment process.

2) Neck: The Rep-PAN neck design is built upon the idea of a hardware-aware neural network by fusing the features at multiple scales. The PAN topology [29] used in YOLOv4 [30] and YOLOv5 is modified to be the base of the Rep-PAN. RepBlock [25] or CSPStackRep block is also used in place of the CSP- Block utilized in YOLOv5.

3) Head: In YOLOv5 models, the classification and box regression heads share the same features. In YOLOx [26], the head is decoupled, which means that the network separates between the features of the classification and box regression heads and adds two additional 3x3 convolutional layers for each one. Although this has been empirically demonstrated to increase performance, it also causes a slight increase in network latency. Based on the Hybrid Channels approach, the decoupling head design of YOLOv6 has been streamlined, and a more effective decoupling head structure has been developed by eliminating the number of middle 3x3 convolutional layers to one. These changes result in lower processing costs and decreased inference delay.

4) Training strategies:

- Anchor-free

YOLOv6 reduces the delay results from transferring massive detections between the hardware stages by replacing Anchor-based detector with anchor-free detectors [26]. The anchor-free paradigm has recently gained significant popularity because of its superior generalization capabilities and simplicity. YOLOv6 adopts one of the anchor-free detectors called the anchor point-based paradigm, which predicts the distances from the bounding boxes' four sides to the anchor point.

- SimOTA label assignment strategy

One of the factors affecting the detection accuracy is the label assignment process. In this process, each predetermined anchor is assigned a label during the training phase. Previous YOLO versions used the static assignment method, which cannot be modified during network training. Recently, numerous techniques based on dynamic label assignment have emerged, allocating positive samples in accordance with the network output through the training procedure to enable the generation of more high-quality positive samples, which in turn improves the network optimization. YOLOv6 used one of the dynamic assignment methods which is SimOTA. This method finds the best match between the samples using the Top-K approximation technique, which significantly accelerates training speed.

- SIoU bounding box regression loss.

IoU, GIoU, and SIoU are proposed in recent researches as bounding box regression losses to adapt the network learning and improve the detection accuracy. These loss functions are calculated according to these aspects: the percentage of the overlap between the predicted and the target boxes, the aspect ratio, the distance between the center points, and the matching between the predicted and the target box directions. GIoU and SIoU loss functions are selected experimentally to apply to different versions of YOLOv6.

B. The Robotic Arm Module

Robots are used to handle complex, dangerous, and tedious tasks. The use of robotic arms also helps to relieve human workers of tasks that pose a risk of bodily harm [31]. Thus, pick-and-place robots are commonly used in modern industrial environments [32]. The pick-and-place process automation reduces cycle time, increases productivity, and decreases material handling costs [33]. Pick and place robots come in a variety of shapes and sizes. A two-degree-of-freedom robotic arm picks up and moves objects in a single plane. The Cartesian robotic arm works in multiple planes and moves along three orthogonal axes using Cartesian coordinates (X, Y, and Z). The Delta robot is frequently used in applications where robots pick items in groups and place them in assembly patterns or containers; they have heavy motors attached. Collaborative robots help humans by directing them to appropriate locations and guiding them through each task. In this research, a Cartesian robot arm is designed to sort the waste according to the locations and types obtained from the segregation module. The robot arm design and its motion planning and execution are described in the following subsections.

1) Solid 3D CAD Modeling of a robotic arm and its workspace: An autonomous robot platform based on SolidWorks, MATLAB Simulink, and Simscape Multibody software tools is utilized in this research to design the arm's structure and its workspace. The robot manipulator was initially created as a 3D CAD model using the SolidWorks tool to design and build the robot completely. It creates fast and accurate 3D models, which turn ideas into reality with the ability to run the concept design through many scenarios and make modifications as necessary in the design development process [34]. As shown in Fig. 2(a), the manipulator structure has four links and four revolute joints. Joint1 is used as a base joint, while joint2 connects link1 to link2. Joint3 connects link2 to link3, and joint4 is used for the robot end effector (robot hand). Fig. 2(b) depicts the simulated workspace which consists of various types of waste objects, an RGB camera placed in front of the robot arm's starting location, ultrasonic sensor, and a group of baskets for collecting categorized items. In order to control the motion of the arm, the robot assembly is imported into a Simscape Multibody model to simulate the multi-object systems by utilizing blocks that

describe system parts, joints, forces, and external restrictions [34]. The bodies are distinguished by their geometry. Accordingly, inertial properties, forces and moments can be applied to bodies. Besides, contact constraints can be defined. Moreover, the object's CAD model can be imported with all its physical properties and the dynamics of the system can be observed. Simscape Multibody software's primary use is to analyze and visualize system functioning and control design in Simulink [35], [36]. Fig. 2(c) shows the block diagram of the robotic arm in Simscape Multibody to be utilized for simulation trials in evaluating the proposed algorithm.

2) Robotic arm task scheduling: This research proposes a planning process to address the issue of computing the robot arm's movements during the object picking and placing assignment. Fig. 3 illustrates the complete cycle of the robotic arm pick and place task. Depending on the type of object material, the robot's destination is specified and modified dynamically. The motion plan is constructed based on several factors depending on the relation between the end-effector, the target destination, and the object's current location. The motion phases are summarized as follows:

- Object reaching: the manipulator moves to place the end-effector at the first object of the received queue. Once the end-effector has arrived at the (x, y) position of the object on the horizontal plane, the ultrasonic sensor is activated to determine the height that the end-effector needs to descend to reach the object.
- Picking up the object: the end-effector gets in touch with the object using a suction force to pick it up. The manipulator must continue to provide a suction force to hold the object until it reaches the proper basket.
- Lifting: the object is lifted vertically away from the table after the end-effector sucks it, enabling it to move as they are rigidly connected.
- Basket reaching: Depending on the object type, the manipulator moves to the appropriate basket location while holding the object.
- Placing the object: the suction force is switched off to release the object into the basket once the manipulator reaches it.
- Arm reset: the robot returns to the beginning state (home position) for starting another task when the object queue becomes empty.

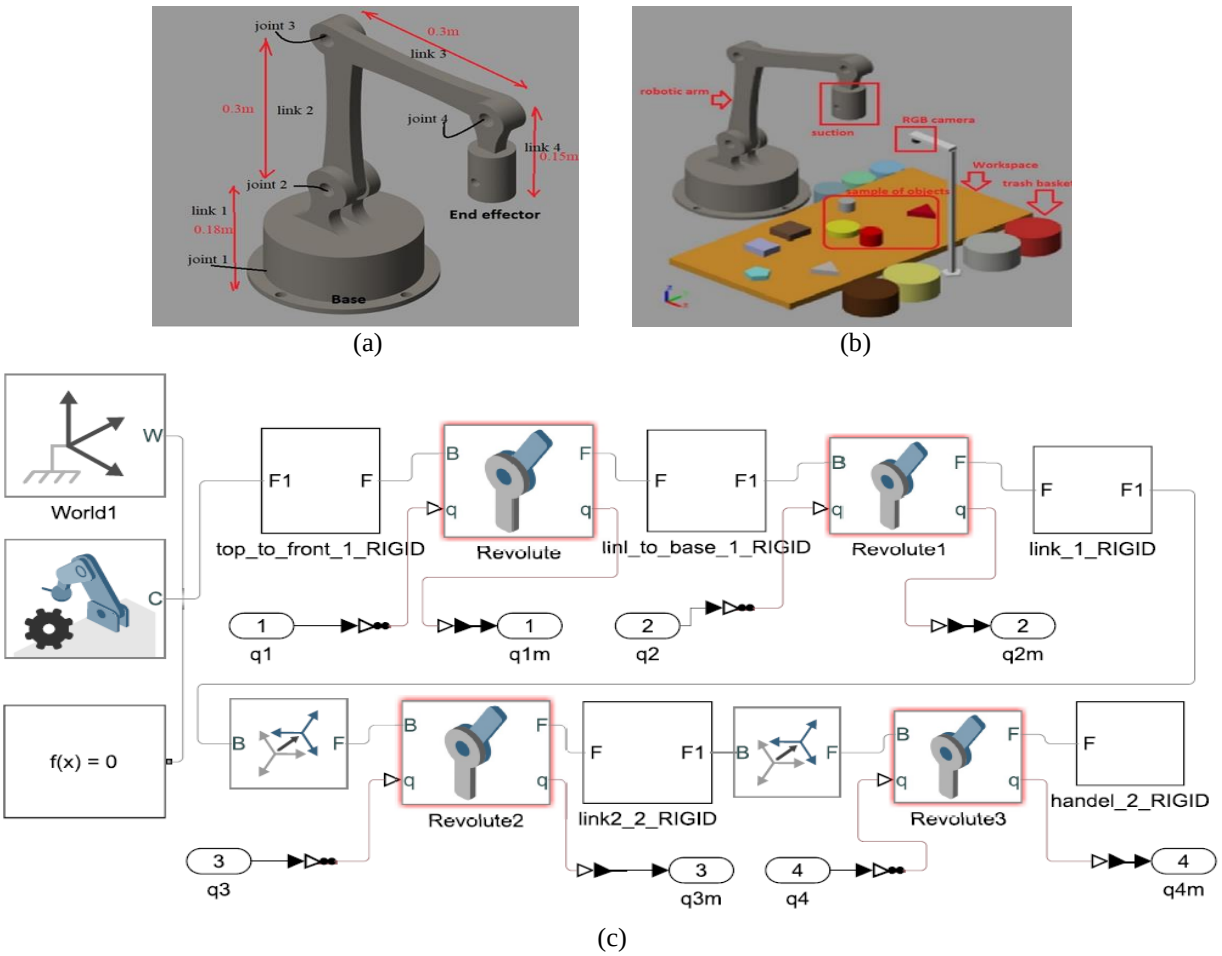


Fig. 2. Robot arm manipulator (a) arm structure (b) robotic workspace (c) Simscape Multibody block diagram of robotic arm.

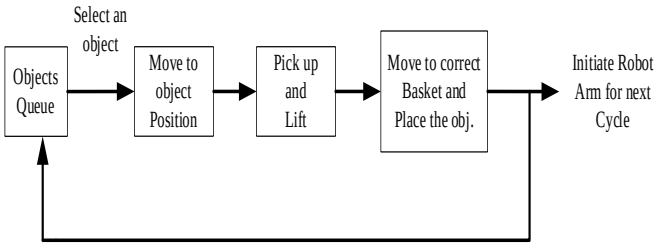


Fig. 3. Complete cycle of pick and place task.

3) Robot motion planning: Fig. 4 illustrates how the robotic arm motion is controlled from its starting point to various goals. First, the object is picked by a suction gripper with force chosen based on the heaviest expected waste object. The time for switching on/off this force is calculated from the generated trajectory. After that, trajectory planning is carried out to find a suitable route for the robot movement in an area free of obstacles. A robotic motion task is defined by determining a path for the robot to follow. A path is a set of points that can be defined in task coordinates (end-effector coordinates) or joint coordinates. The problem of trajectory generation is to compute the desired reference joint or end-effector variables as functions of time for the control system so that the robot follows the desired path [37]. In this study, a cubic polynomial trajectory that is constructed based on chosen waypoints and their associated time points is used to calculate the appropriate route between the source and destination. Waypoints are chosen so that the arm can move smoothly while considering the joint angle limitations. Inverse kinematics is proposed to calculate the appropriate joint configuration (joint rotation angles) for moving the robotic arm from its starting position to various destination points.

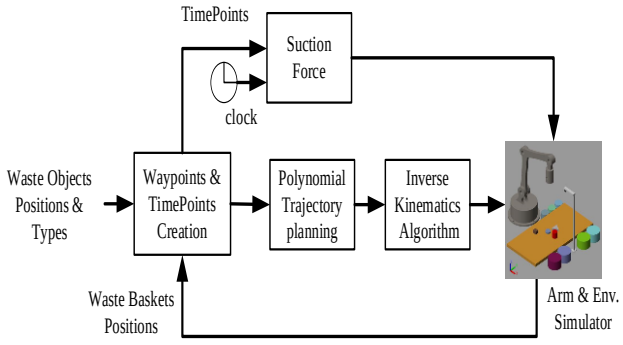


Fig. 4. Motion planning and movement execution process.

- Proposed Geometric Inverse Kinematics

The study of motion without considering its causes, such as forces and torques, is known as kinematics. Using kinematic equations, inverse kinematics (IK) can predict how a robot will move to arrive at a specific place [38]. Fig. 5 declares the concept of forward and inverse kinematics approach. Inverse kinematics is a transformation method from Cartesian space to joint space. Nevertheless, it is a somewhat complex nonlinear problem. In addition to its nonlinearity, the kinematics matrix's sine and cosine functions also have non-unique solutions, which makes the issue worse [39]. This study proposes a geometric analytical solution idea to compute the inverse kinematics of 3D space. The system will then follow the intended route determined by the trajectory of the end effector. The geometric solution is much more efficient during calculation (when compared to iterative methods).

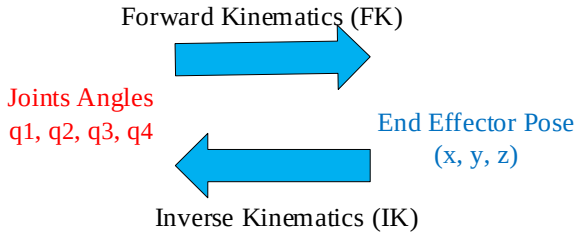


Fig. 5. Inverse kinematics vs forward kinematics.

The proposed method for determining the joints configuration (joints rotation angles) in 3D space for the designed four-link robot arm relied on a novel geometric notion. Since the numerical solution of inverse kinematics yields, a less accurate result, the analytical solution of inverse kinematics is preferred. The target point is indicated by the Cartesian coordinates P (x, y, and z) in the generalized case of the robot arm. The position of the target point can be converted to cylindrical coordinates (Θ, ρ, and z). If the base joint is turned at an angle Θ, as indicated in Fig. 6, the robot arm construction coincides in the x-z plane. The second joint position at point "a" is connected to the end-effector joint pose at point "p" by the line "c". Line "k" is thus drawn parallel to line "c" from the location of the base joint. The angles of the joints are obtained by applying the cosine formula as shown in equations (1–5). The polar coordinates of the target point projection in the x-y plane are determined by Equation (1).

$$\rho = \sqrt{x^2 + y^2} \quad , \quad \theta = \tan^{-1}(y/x) \tag{1}$$

The length of the line "c" is calculated as:

$$c = \sqrt{(z - l_1)^2 + \rho^2} \tag{2}$$

Equation (3) calculates the angle between line "k" and the x-axis.

$$\delta = \tan^{-1}((z - l_1)/\rho) \tag{3}$$

The following equations explain the angles α and β in the triangle (a-P-d).

$$\alpha = \cos^{-1}\left(\frac{l_2^2 + l_3^2 - c^2}{2 l_2 l_3}\right) \tag{4}$$

$$\beta = \cos^{-1}\left(\frac{l_2^2 + c^2 - l_3^2}{2 l_2 c}\right) \tag{5}$$

In order to move the arm's end-effector to a target point, the joints rotate at the angles calculated in equations (6-9) considering that the rotation angle of each joint frame is measured w.r.t. the previous link frame and counterclockwise.

$$q_1 = \theta \tag{6}$$

$$q_2 = 270 + \beta + \delta \tag{7}$$

$$q_3 = 180 + \alpha \tag{8}$$

To keep the end-effector pointed vertically

$$q_4 = 90 - (\alpha + \beta + \delta), \tag{9}$$

where, q1, q2, q3, and q4 are the angles values needed to rotate each joint to change the configuration of the joints such that the end effector reaches the desired position.

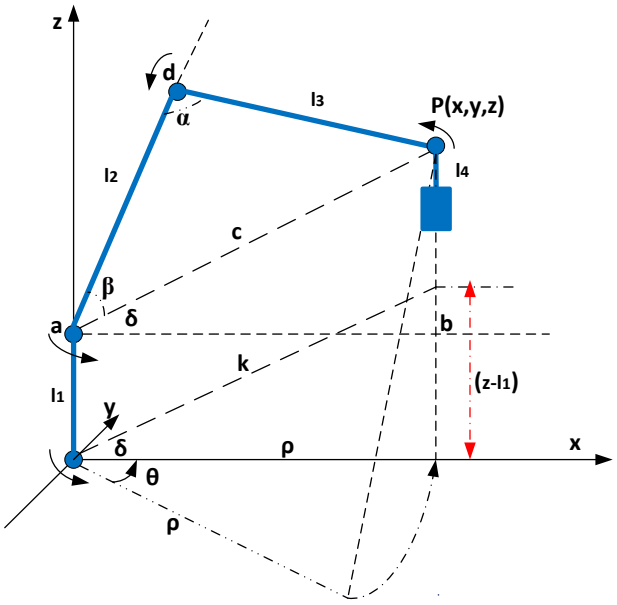


Fig. 6. The schematic diagram of the proposed geometric inverse kinematics approach.

III. Experiments and Results

This section presents the conducted experiments and shows the results of the proposed methodology for waste segregation and robotic modules. In the following subsections, each module's outcomes are described in detail.

A. Waste Segregation Module

In this module, YOLOv6 is evaluated numerically and compared to recent YOLO models: YOLOv7 [40] and YOLOR [41]. The dataset, performance metrics, and training parameters are also comprehensively explained.

1) Dataset: The proposed waste segregation performance model is assessed by conducting different experiments on a modified version of the TrashNet dataset. The TrashNet dataset [42] consists of 2527 images of waste divided into six classes: 501 glasses, 594 paper, 403 cardboard, 482 plastic, 410 metal, and 137 other trashes. The trash class is omitted in the modified version of the dataset, and two new classes are added, foam and battery. The authors also attempt to balance the number of the other classes' images by adding new ones. The new images are downloaded using Google Images Download software [43]. Then, some augmentation techniques are applied, namely: flipping, rotation, and resizing. Afterwards, the images are annotated using Ybat software [44], then the duplicated ones are deleted. At the end of the preprocessing and the annotation process, the dataset becomes 3217 images divided into seven classes: cardboard, glass, metal, paper, plastic, battery, and foam. The description of the modified TrashNet is presented in Table I, and samples from each class are shown in Fig. 7.

*2) Performance metrics:*To evaluate the performance of pre-trained YOLO models, three evaluation metrics namely, precision (AP), recall (AR), and the F1 score (F1) are applied on the model's detection output. Mathematically, precision, recall and F1 can be calculated as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{10}$$

$$\text{Re}call = \frac{TP}{TP + FN} \tag{11}$$

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

where, TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. Therefore, precision measures the ratio of the correctly detected objects to the total number of detected objects. Recall measures the percentage of true predictions among the total number of class objects, and F1 evaluates the model's performance based on the harmonic mean of the precision and recall.

TABLE I. Image Classes Distribution and Description in the Dataset

Class (type)	Description	Quantity
Glass	bottle, jar, cups	500
Paper	plates, posters, envelopes, receipts	591
Cardboard	packing box, mailing box, cardboard sheet	457
Plastic	bottles, boxes, jars, medicine packs, food packs	479
Metal	soft drink cans, food cans, foil sheets, plates	468
Foam	packing box, food box, plates, cups	382
Battery	AA, AAA, C, D, 9 volts	326

Fig. 7. Sample images from the modified TrashNet dataset with their corresponding class.

3) Training : The modified TrashNet dataset is split into 70% train, 30% for validation and test. Thus, according to these percentages, the number of images is 2239 in the training set, 541 in the validation set, and 423 in the test set. Training the model from scratch with randomly initialized parameters results in under-fitting due to the small number of images in the dataset. Therefore, the pre-trained YOLOv6 model, which is trained on the COCO dataset, is fine-tuned on the modified TrashNet dataset to exploit the advantages of transfer learning. Stochastic gradient descent (SGD) is used

during the fine-tuning process to optimize the network parameters. The number of epochs, the initial learning rate, the batch size, the weight decay, and the momentum are set as 300, 0.0032, 16, 0.00036, and 0.843, respectively. The confidence score is selected empirically, and a confidence score of 0.6 gives the best model performance. All the experiments are conducted on an Intel(R) Core (TM) i7-11800H.

4) Simulation results: The performance of the YOLOv6 model is compared with the recent YOLO models, which are YOLOv7 and YOLOR, in terms of the F1, precision, recall, inference time, and model size. YOLOv7 and YOLOR are trained on the COCO dataset and fine-tuned on the modified TrashNet. In the proposed research, two model architectures of YOLOv6 are used, in which the architectures vary considering the model size for a better accuracy-speed trade-off. The model size of YOLOv6n is smaller than YOLOv6s. The training process of the four models is performed several times according to different data shuffles, and the results are presented in Table II. It can be seen from the table that the highest precision value is for YOLOv7 in the third run, whereas the best recall and F1 values are for YOLOv6s in the first run. However, the recall and F1 values for YOLOv7 in the third run are less than that of YOLOv6s in the first run, while the precision value of YOLOv6s is slightly less than that of YOLOv7. Thus, YOLOv6s from the first run is adopted in this work. Besides, Table III compares the YOLOv6s versus the other models regarding precision, recall, F1, inference time, and model size. The presented values of precision, recall, and F1 are the average values of the three runs. It can be noticed from Table III that the YOLOv6s model has better performance with reference to the average values of precision, recall, and F1.

Furthermore, it can be observed that YOLOv6n is the smallest model size, and YOLOR is the largest model size and inference time. Thus, YOLOv6 has better performance and meets the real-time requirements. Using YOLOv6s or YOLOv6n depends on the application's requirements; if the application needs high accuracy, YOLOv6s is the best choice. However, if the application needs a small-size model with acceptable accuracy, YOLOv6n is recommended. Fig. 8 presents the predictions of YOLOv6 on real images captured by the authors using the RGB camera of the Samsung Galaxy S20 with 12 MP for testing the model's performance. It can be seen from the figure that the model predicts all the objects with high confidence scores.

On the other hand, the incorrect classifications on the test set are also statistically investigated to find the reasons behind the model misclassifications for some objects. Table IV analyses the classification output based on the confusion matrix. It can be observed from Table IV that there is a slight confusion between cardboard and paper, and also, there is confusion between plastic, glass and paper. This confusion is because, in some cases, the plastic and glass bottles or cups are very similar in shape; similarly, the supermarket flyers in paper category and the box packaging in the cardboard type. It can be observed from Table V that plastic has the lowest recall value, 90% because plastic is misclassified as the other categories multiple times. It can be noticed from the confusion matrix that foam and battery are not misclassified; however, their precision and recall values are not 100%. Some foam and battery objects are undetected, or the background is classified as foam or battery. Moreover, the glass and paper categories have the lowest precision, meaning several objects are misclassified as glass and paper. Furthermore, experimental results find that the number of undetected and misclassified objects is 39 objects for YOLOv7 and 32 objects for YOLOR from 508 objects in the test set compared to 19 objects for YOLOv6. Fig. 9 visualizes some illustrations of the classification results for YOLOv6, YOLOv7, and YOLOR. As can be noticed, YOLOv6 outperforms YOLOv7, and YOLOR in most of the cases.

TABLE II. THE EFFECT OF THE DATASET SHUFFLING ON THE MODEL'S PERFORMANCE

	Models	Precision	Recall	F1-score
1st train	**YOLOR**	94.82	93.7	94.26
	YOLOv6n	95.19	93.5	94.34
	YOLOv6s	**96.07**	**96.26**	**96.17**
	YOLOv7	94.75	92.32	93.52
2nd train	**YOLOR**	95.31	95.11	95.21
	YOLOv6n	95.05	93.9	94.47
	YOLOv6s	95.48	94.7	95.01
	YOLOv7	95.87	94.5	95.18
3rd train	**YOLOR**	95.65	94.48	95.06
	YOLOv6n	95.63	94.07	94.85
	YOLOv6s	95.67	94.68	95.17
	YOLOv7	96.47	94.89	95.67

TABLE III. THE AVERAGE PERFORMANCE COMPARISON OF THE YOLOV6 VERSUS YOLOV7 AND YOLOR

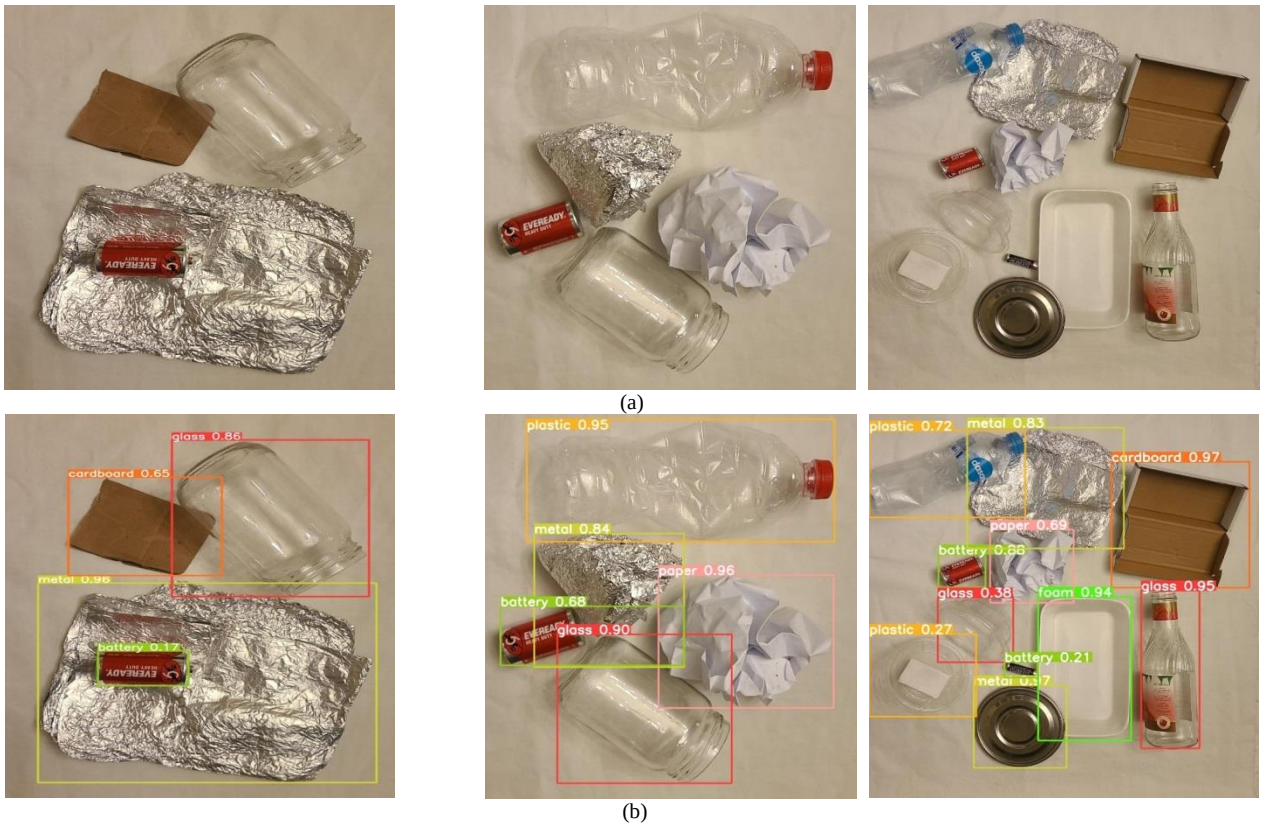
Models	F1-score	Precision	Recall	Inference time	weight size
YOLOR	94.84	95.26	94.43	24 sec	289 M
YOLOv6n	94.55	95.29	93.82	14 sec	**9 M**
YOLOv6s	**95.45**	**95.74**	**95.2**	13 sec	37 M
YOLOv7	94.79	95.7	93.9	**11 sec**	73 M

TABLE IV. CONFUSION MATRIX FOR EVALUATING THE CLASSIFICATION ACCURACY OF EACH MATERIAL TYPE

		Predicted						
		Glass	**Paper**	**Cardboard**	**Plastic**	**Metal**	**Foam**	**Battery**
Actual	**Glass**	64	0	0	1	0	0	0
	Paper	0	78	0	0	0	0	0
	Cardboard	0	3	58	0	0	0	0
	Plastic	3	3	0	57	0	0	0
	Metal	2	0	0	1	71	0	0
	Foam	0	0	0	0	0	60	0
	Battery	0	0	0	0	0	0	101

TABLE V. THE PERFORMANCE OF YOLOV6 FOR EACH CATEGORY IN THE DATASET IN TERMS OF PRECISION, RECALL AND F1

Class Type	**Precision**	**Recall**	**F1**
Glass	93	98	95
Paper	92	100	96
Cardboard	100	95	97
Plastic	95	90	92
Metal	99	95	97
Foam	98	94	96
Battery	97	99	98



(a)

(b)

Fig. 8. YOLOv6 waste detection results (a) test images (b) the model predictions.

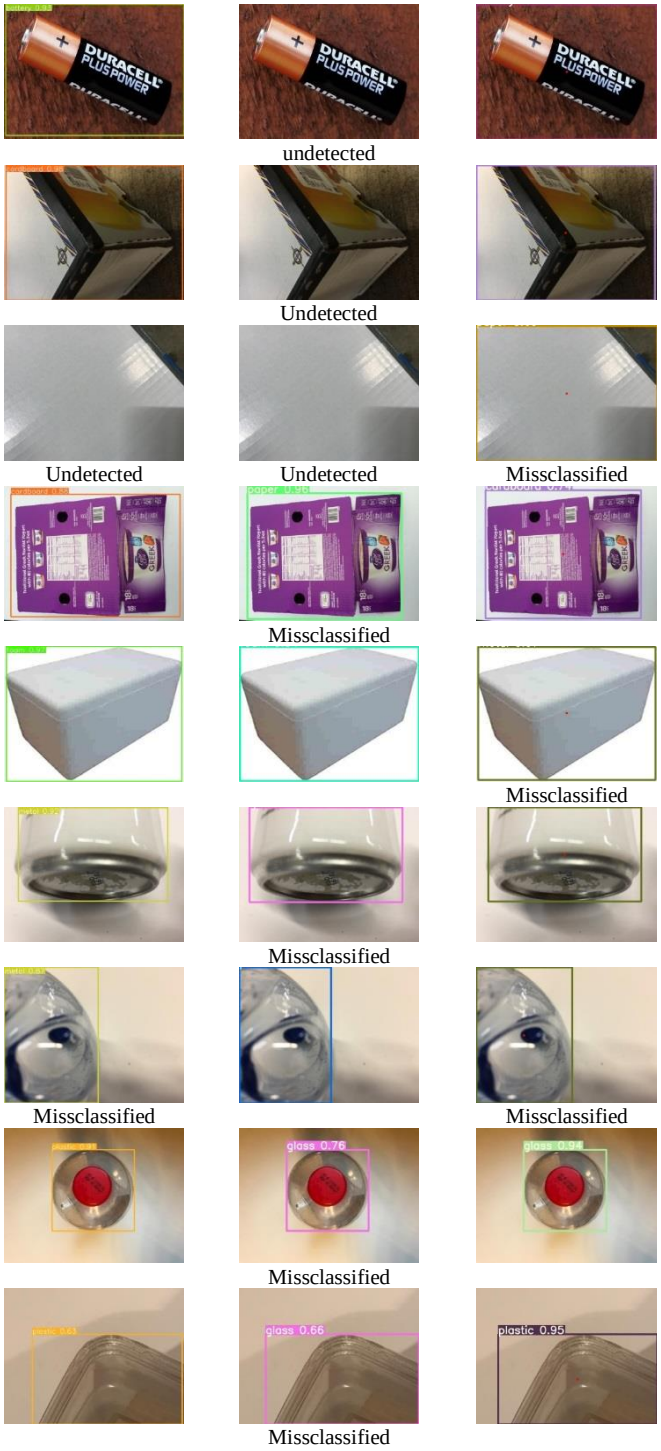


Fig. 9. Success and failure classifiction results of different test waste objects (a) YOLOv6s (b) YOLOv7 (c) YOLOR.

B. Robotic Arm Simulation Results

In the robot module, the location and material types of the waste objects that were detected in the waste segregation module are sent to the robotic arm for motion planning and execution. The proposed algorithm finds a solution for all four phases and moves the robot arm within its structure limits (joints and links limits) to the target position. The robot begins at its default position and the camera captures an image of the workspace. The image is sent to the computer for processing and detecting the objects to define their locations and material types. Then this data is formed in a queue and sent to the robot. The proposed algorithm utilizes these objects information to compute the appropriate joint configurations, allowing the arm simulator to perform all required tasks. Fig. 10 shows the motion sequence to place the objects in the right destination. Fig. 10(a) depicts the robot in its initial state while it is waiting to receive data from the object detection module. The necessary joints configuration is computed using the previously mentioned equations in Section II, then the robot successfully navigates to the chosen item (in this case, the red cylinder), and then the controller triggers the suction force to pick up the object, as illustrated in Fig. 10(b). The robot lifts the object and defines the position of the dedicated basket based on its type. The basket position is considered the new target point for the robot arm; thus, the trajectory and the correct joint angles are computed using inverse kinematics. When the arm reaches above the basket, the controller turns off the suction force to drop the object in the basket. Afterward, the next object is selected (in this case, the yellow disc), the arm considers its position to be the new target, computes the required joints rotation angles once more, and executes the pick and place task. The process is repeated until all objects are placed in their proper baskets; thus, the robot returns to its initial state. Fig. 11 shows the timeline of the process and the execution times for each task, such as the time the robot takes to go to the object's location, lift it, and finally drop it in the designated basket. It can be noticed from the figure that the processing time of delivering one object is 0.8 sec. Further, the trajectories of motion in joint space for the previous process sequences are illustrated in Fig. 12, pointing out that all trajectories are smooth, and there hasn't been any rapid damage or significant change made to the arm's rotation joints. The complete modeling and simulation of a robotic arm pick and place system with MATLAB Simscape Multibody and Solidworks is shown in Fig. 13.

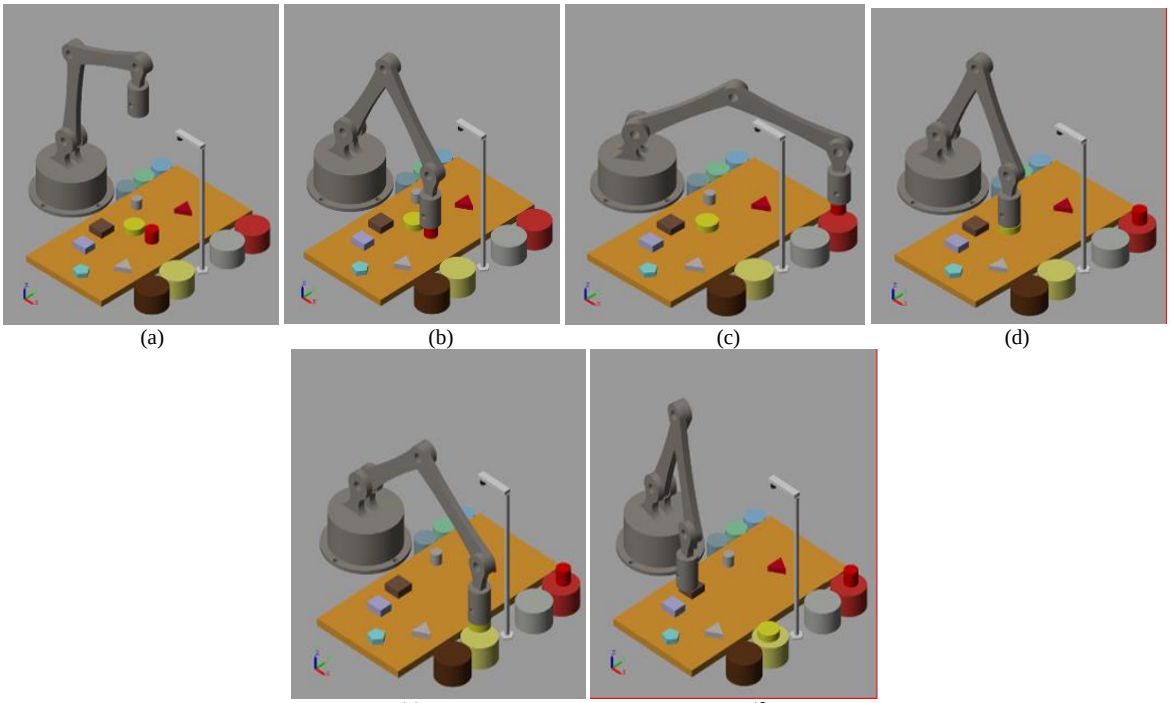


Fig. 10. The motion sequences required to complete the mission. (a) intial state (b) pick up object1 (c) place object1 in the dedicated basket (d) pick up object2 (e) place object2 in its basket (f) repeat the process for all objects in queue.

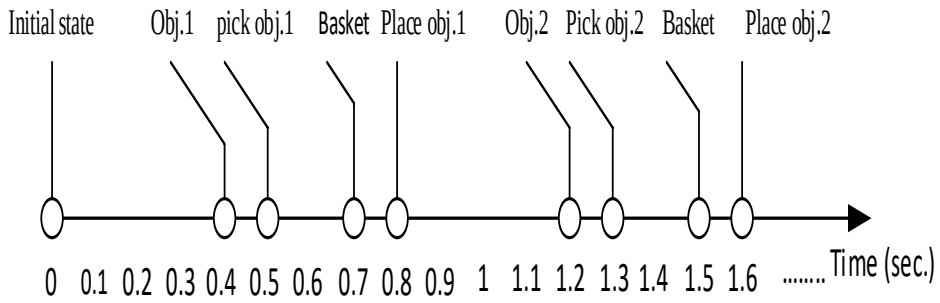
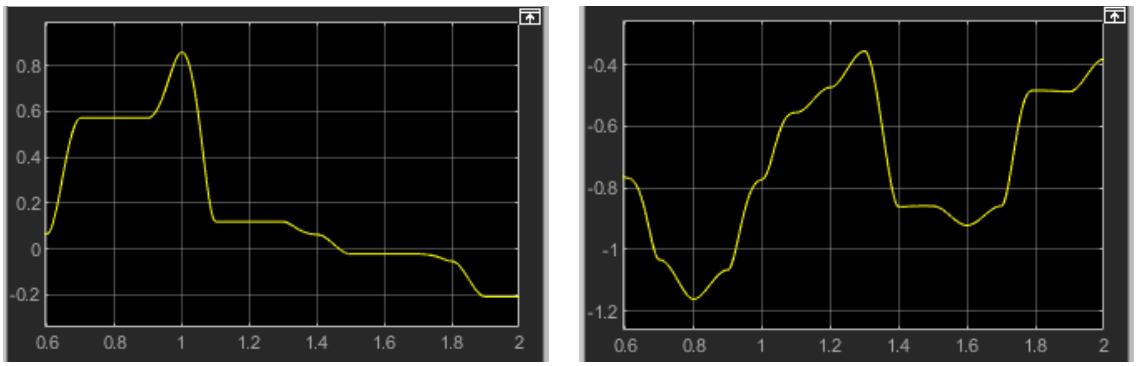
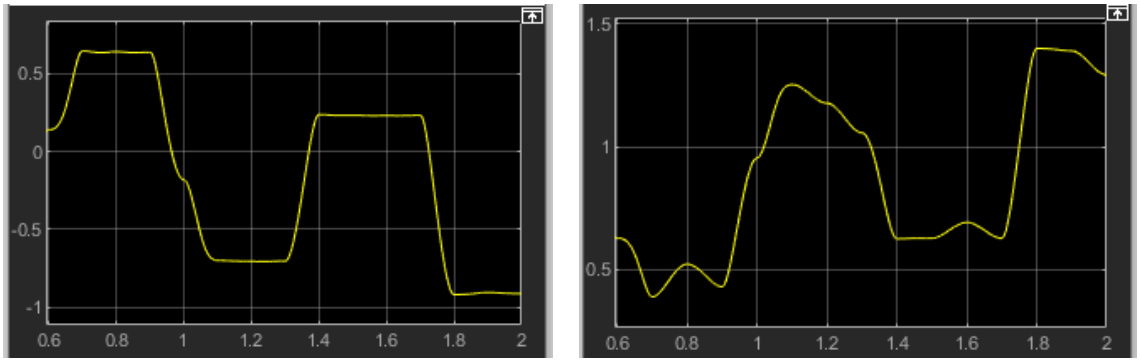


Fig. 11. Time line of pick and place mission for multi-objects.





(c) (d)

Fig. 12. The joints angles trajectories (in rad) (a) q_1 (b) q_2 (c) q_3 (d) q_4

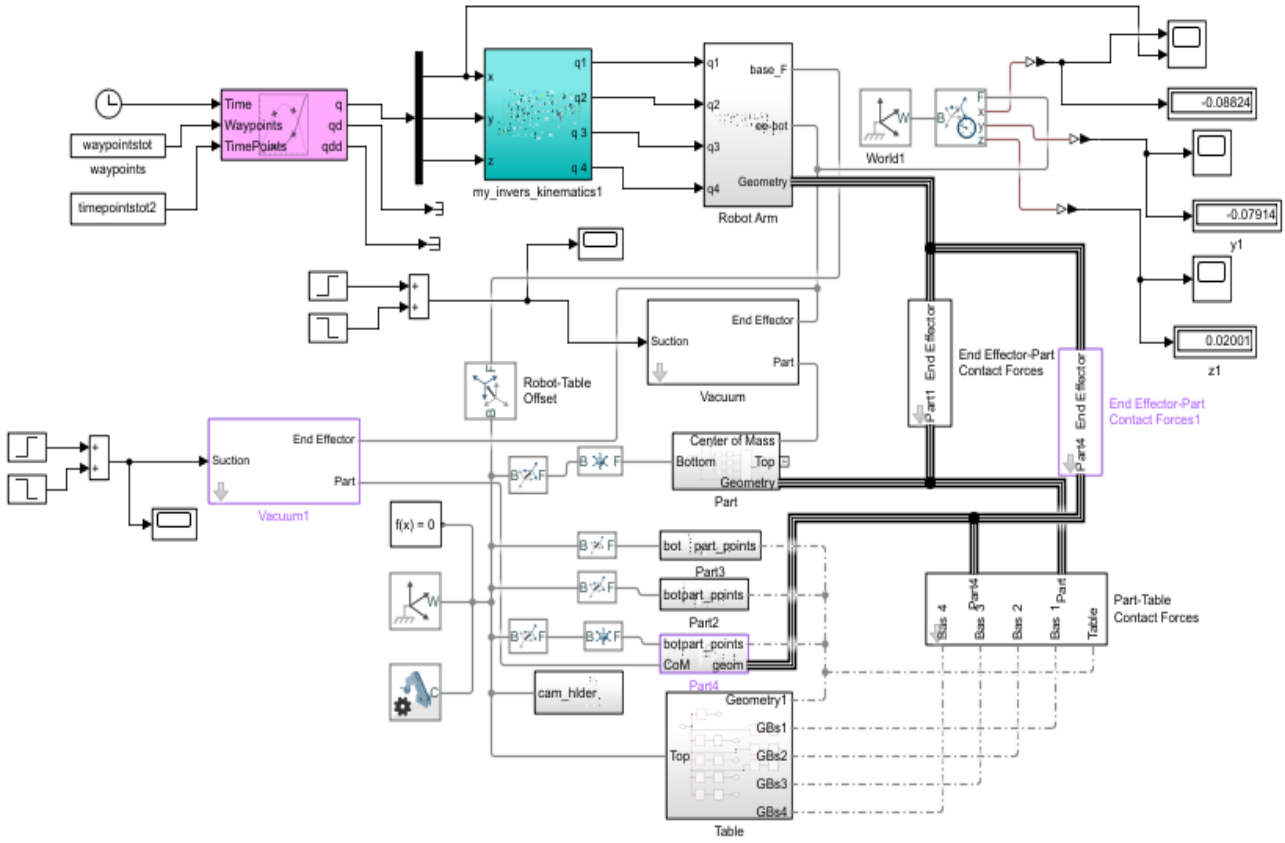


Fig. 13. The overall motion control robotic arm system simulator.

IV. CONCLUSION

This paper presents an automated waste segregation technique relying on blending an object detection system and robotic arm design. YOLOv6 is applied to detect and classify waste items. Over and above, CAD software is employed to design the robot arm that strives towards utilizing a simple geometric approach to compute the angles of the arm's joints accurately. To signify the efficacy of the proposed system, a TrashNet dataset has been modified and exploited for assessment. The suggested system proved high effectiveness in detecting and segregating waste items into distinct categories. Moreover, the system revealed high efficiency in picking the waste items, controlling the robot arm movement to the appropriate basket location, and placing the object in the proper basket. Furthermore, the adopted object detection approach is compared to the recent YOLO models: YOLOv7 and YOLOR. The obtained results illustrate that the submitted technique surpasses these techniques regarding F1, precision, recall, inference time, and model size. Over and above, the designed robot arm has been proven to consume a fraction of a second for picking up and placing a single object in its appropriate basket. In future work, new items will be added to the modified dataset, and the proposed simulation robot arm will be practically implemented.

V. FUNDING

This research was supported by Korea Institute of Marine Science & Technology Promotion (KIMST) funded by the Ministry of Oceans and Fisheries, Korea(G22202202102301), and by Korea Agency for Infrastructure Technology Advancement (KAIA), project title is the development of smart city governance structure and framework (project number is 22DEAP-B158922-0312982076870003).

REFERENCES

[1] R. A. E. David Edgar, Fantastic Recycled Plastic: 30 Clever Creations to Spark Your Imagination. New York: Lark Books, 2009.

[2] R. A. Aral, S. R. Keskin, M. Kaya, and M. Haciömeroğlu, "Classification of TrashNet Dataset Based on Deep Learning Models," Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018, pp. 2058–2062, 2019, doi: 10.1109/BigData.2018.8622212.

[3] A. U. Gondal et al., "Real time multipurpose smart waste classification model for efficient recycling in smart cities using multilayer convolutional neural network and perceptron," Sensors, vol. 21, no. 14, pp. 1–15, 2021, doi: 10.3390/s21144916.

[4] W. Ma, X. Wang, and J. Yu, "A lightweight feature fusion single shot multibox detector for garbage detection," IEEE Access, vol. 8, pp. 188577–188586, 2020, doi: 10.1109/ACCESS.2020.3031990.

[5] D. O. Melinte, A. M. Travediu, and D. N. Dumitriu, "Deep convolutional neural networks object detector for real-time waste identification," Appl. Sci., vol. 10, no. 20, pp. 1–18, 2020, doi: 10.3390/app10207301.

[6] B. De Carolis, F. Ladogana, and N. MacChiarulo, "YOLO TrashNet: Garbage Detection in Video Streams," IEEE Conf. Evol. Adapt. Intell. Syst., vol. 2020-May, 2020, doi: 10.1109/EAIS48028.2020.9122693.

[7] S. Kumar, D. Yadav, H. Gupta, O. P. Verma, I. A. Ansari, and C. W. Ahn, "A novel yolov3 algorithm-based deep learning approach for waste segregation: Towards smart waste management," Electron., vol. 10, no. 1, pp. 1–20, 2021, doi: 10.3390/electronics10010014.

[8] A. B. Wahyutama and M. Hwang, "YOLO-Based Object Detection for Separate Collection of Recyclables and Capacity Monitoring of Trash Bins," Electron., vol. 11, no. 9, 2022, doi: 10.3390/electronics11091323.

[9] I. E. Agbehadji, A. Abayomi, K. H. N. Bui, R. C. Millham, and E. Freeman, "Nature-Inspired Search Method and Custom Waste Object Detection and Classification Model for Smart Waste Bin," Sensors (Basel)., vol. 22, no. 16, 2022, doi: 10.3390/s22166176.

[10] A. Ye, B. Pang, Y. Jin, and J. Cui, "A YOLO-based Neural Network with VAE for Intelligent Garbage Detection and Classification," ACM Int. Conf. Proceeding Ser., 2020, doi: 10.1145/3446132.3446400.

[11] A. P. Saputra, "Waste Object Detection and Classification using Deep Learning Algorithm: YOLOv4 and YOLOv4-tiny," Turkish J. Comput. Math. Educ., vol. 12, no. 14, pp. 1666–1677, 2021.

[12] D. Patel, F. Patel, S. Patel, N. Patel, D. Shah, and V. Patel, "Garbage Detection using Advanced Object Detection Techniques," Proc. - Int. Conf. Artif. Intell. Smart Syst. ICAIS 2021, pp. 526–531, 2021, doi: 10.1109/ICAIS50930.2021.9395916.

[13] S. Majchrowska et al., "Deep learning-based waste detection in natural and urban environments," Waste Manag., vol. 138, no. June 2021, pp. 274–284, 2022, doi: 10.1016/j.wasman.2021.12.001.

[14] I. Agustian, N. Daratha, R. Faurina, A. Suandi, and S. Sulistyaningsih, "Robot Manipulator Control with Inverse Kinematics PD-Pseudoinverse Jacobian and Forward Kinematics Denavit Hartenberg," J. Elektron. dan Telekomun., vol. 21, no. 1, p. 8, 2021, doi: 10.14203/jet.v21.8-18.

[15] A. R. Al Tahtawi, M. Agni, and T. D. Hendrawati, "Small-scale robot arm design with pick and place mission based on inverse kinematics," J. Robot. Control, vol. 2, no. 6, pp. 469–475, 2021, doi: 10.18196/jrc.26124.

[16] R. Sam, K. Arrifin, and N. Buniyamin, "Simulation of pick and place robotics system using solidworks softmotion," Proc. 2012 Int. Conf. Syst. Eng. Technol. ICSET 2012, no. September, 2012, doi: 10.1109/ICSEngT.2012.6339325.

[17] H. H. K. Doo Sung Ahn, Ill Yeong Lee, "Integrated SolidWorks & Simscape Platform for the Model-Based Control Algorithms of Hydraulic Manipulators," J. Drive Control, vol. 12, no. 4, pp. 41–47, 2015.

[18] X. Yue, H. Li, M. Shimizu, S. Kawamura, and L. Meng, "YOLO-GD: A Deep Learning-Based Object Detection Algorithm for Empty-Dish Recycling Robots," Machines, vol. 10, no. 5, pp. 1–20, 2022, doi: 10.3390/machines10050294.

[19] Q. Song et al., "Object detection method for grasping robot based on improved yolov5," Micromachines, vol. 12, no. 11, 2021, doi: 10.3390/mi12111273.

[20] J. Bai, S. Lian, Z. Liu, K. Wang, and D. Liu, "Deep Learning Based Robot for Automatically Picking Up Garbage on the Grass," IEEE Trans. Consum. Electron., vol. 64, no. 3, pp. 382–389, 2018, doi: 10.1109/TCE.2018.2859629.

[21] J. Kim et al., "An Innovative Automated Robotic System based on Deep Learning Approach for Recycling Objects," in Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO), 2019, pp. 613–622. doi: 10.5220/0007839906130622.

[22] C. Li et al., "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications," arXiv Prepr. arXiv2209.02976, 2022, [Online]. Available: http://arxiv.org/abs/2209.02976.

[23] J. Glenn, "YOLOv5 release v6.1," https://github. com/ultralytics/yolov5/releases/tag/v6, 2022.

[24] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., vol. 2020-June, pp. 1571–1580, 2020, doi: 10.1109/CVPRW50498.2020.00203.

[25] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVgg: Making VGG-style ConvNets Great Again," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., no. 2017, pp. 13728–13737, 2021, doi: 10.1109/CVPR46437.2021.01352.

[26] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," arXiv:2107.08430, pp. 1–7, 2021, [Online]. Available: http://arxiv.org/abs/2107.08430.

[27] Z. Gevorgyan, "SIoU Loss: More Powerful Learning for Bounding Box Regression," arXiv Prepr. arXiv2205.12740, pp. 1–12, 2022, [Online]. Available: http://arxiv.org/abs/2205.12740.

[28] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 658–666, 2019, doi: 10.1109/CVPR.2019.00075.

[29] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 8759–8768, 2018, doi: 10.1109/CVPR.2018.00913.

[30] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv Prepr. arXiv2004.10934, 2020, [Online]. Available: http://arxiv.org/abs/2004.10934.

[31] V. A. Knights, M. Stankovski, N. Stojance, and O. Petrovska, "ROBOTS FOR SAFETY AND HEALTH AT WORK," 2015.

[32] K. Ghadge, S. More, P. Gaikwad, and S. Chillal, "Robotic ARM for pick and place application," Int. J. Mech. Eng. Technol., vol. 9, no. 1, pp. 125–133, 2018.

[33] S. Surati, S. Hedaoo, T. Rotti, V. Ahuja, and N. Patel, "Pick and Place Robotic Arm: A Review Paper," Int. Res. J. Eng. Technol., pp. 2121–2129, 2021, [Online]. Available: www.irjet.net.

[34] D. Cekus, B. PosiadaŁa, and P. Warys, "Integration of modeling in solidworks and matlab/simulink environments," Arch. Mech. Eng., vol. 61, no. 1, pp. 57–74, 2014, doi: 10.2478/meceng-2014-0003.

[35] M. Pozzi, G. M. Achilli, M. C. Valigi, and M. Malvezzi, "Modeling and Simulation of Robotic Grasping in Simulink Through Simscape Multibody," Front. Robot. AI, vol. 9, no. May, pp. 1–14, 2022, doi: 10.3389/frobt.2022.873558.

[36] M. Siwek, L. Baranowski, J. Panasiuk, and W. Kaczmarek, "Modeling and simulation of movement of dispersed group of mobile robots using Simscape multibody software," AIP Conf. Proc., vol. 2078, no. March 2019, 2019, doi: 10.1063/1.5092048.

[37] E. D. W Khalil, "Trajectory generation," in Modeling, Identification and Control of Robots, 2002, pp. 313–345.

[38] O. H. and J. Šedo and Additional, "Forward and Inverse Kinematics Using Pseudoinverse and Transposition Method for Robotic Arm DOBOT," in Kinematics, London, United Kingdom: IntechOpen, 2017. [Online]. Available: https://www.intechopen.com/books/advanced-biometric-technologies/liveness-detection-in-biometrics.

[39] S. Parman and A. Machmudah, "Waveform based Inverse Kinematics Algorithm of Kinematically Redundant 3-DOF Manipulator," Int. J. Innov. Technol. Interdiscip. Sci., vol. 3, no. 2 SE-Articles, pp. 407–428, 2020, doi: 10.15157/IJITIS.2020.3.2.407-428.

[40] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv Prepr. arXiv2207.02696, pp. 1–15, 2022, [Online]. Available: http://arxiv.org/abs/2207.02696.

[41] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "You Only Learn One Representation: Unified Network for Multiple Tasks," arXiv Prepr. arXiv2105.04206, pp. 1–11, 2021, [Online]. Available: http://arxiv.org/abs/2105.04206.

[42] M. Yang and G. Thung, "Classification of Trash for Recyclability Status," CS229Project Rep., pp. 1–6, 2016.

[43] J. O. Yicong, "Google-Image-Scraper." https://github.com/ohyicong

[44] D. Sun, "Ybat - YOLO BBox Annotation Tool." https://github.com/drainingsun/ybat.