Engineering Applications
https://publish.mersin.edu.tr/index.php/enap
e-ISSN 2979-9201

Identification and classification of fruits through robotic system by using artificial intelligence

Arbnor Pajaziti [*1], Fatmir Basholli [2], Ylber Zhaveli [1]

[1]*University of Prishtina, Department of Mechatronics, Kosovo, arbnor.pajaziti@uni-pr.edu; ylber.zhaveli@student.uni-pr.edu*
[2]*Albanian University, Department of Engineering, Albania, fatmir.basholli@albanianuniversity.edu.al*

Cite this study: Pajaziti, A., Basholli, F., & Zhaveli, Y. (2023). Identification and classification of fruits through robotic system by using artificial intelligence. Engineering Applications, 2 (2), 154-163

Keywords
Robotic system
Processing
Artificial intelligence

Research Article
Received:31.03.2023
Revised: 03.05.2023
Accepted: 14.05.2023
Published:26.05.2023

Abstract

The purpose of this research paper is to facilitate the work of industries that deal with the classification of different types of products, specifically in our case from the fruit and vegetable industry using Artificial Intelligence. The most important part of this work was the processing of images for fruit identification. For the classification to be as accurate as possible, it was necessary to use more samples of different types of fruits. To achieve the desired results, a total of 350 samples were needed, where to reach the number of these samples we photographed several types of fruits at different angles. Using the Python programming language, which has many libraries that are open sources and can be advanced by various professionals in the field of programming, facilitates image processing. For this purpose, the OpenCV library was used to generate photos so that fruits can be identified more easily. TensorFlow is used as a platform to teach machines to adapt our samples to the language that the robotic system understands.

1. Introduction

As the population grows, so do the demands for the production of various products. Along with production comes competition, causing industries to compete for quality, quantity, and precision. To achieve all these people, need automated work. The devices that perform automated work are robotic systems.

Robotic systems are mostly used in the manufacturing industry. Every day industries strive to have greater accuracy while saving time, while simultaneously being able to double production and increase productivity at work. To achieve this success, the right combination is required through the fields of programming, electronics, and mechanical parts, in other terms known as the field of Mechatronics.

Industrial automation reduces operating expenses by eliminating the costs of health care, paid leave, and vacation that comes from excluding a human operator. Furthermore, additional employee advantages, such as bonuses, pension coverage, and so on are reduced with industrial automation. Above all, despite the high initial cost, it saves the monthly wages of workers by drastically reducing their number, resulting in significant cost savings for the company.

In industrial automation, the used machines have lower maintenance costs as they're less probably to fail. Only computer and maintenance engineers are authorized to repair it if it fails. Only computer and maintenance engineers are required to repair it if it fails. Industrial automation is gaining appeal in a range of industries due to its many advantages, including increasing productivity, quality, and safety at a cheap cost. Automated lines are replacing employee group work by increasing the efficiency of numerous operations that have been carried out by hand and increasing productivity.

An example of an automated line is the FruitTron product [1]. It is a novelty that automates the task of ranking fruits with higher accuracy and classification of a large number of fruits. A combination of rigorous imaging processing algorithms and hardware ensures that each fruit is comprehensively examined for all defects before exiting the inspection line. FruitTron uses a machine vision system capable of inspecting defects on the fruit surface; determining the shape and color of the fruit; measuring the weight and size of the fruit. Artificial intelligence is not used in this automated line to make machine vision processing.

The purpose of this work is to use AI to identify fruits from images taken by the camera and the actuation of the robotic system to make their placement in baskets according to the respective type.

2. Material and Method

2.1. Artificial intelligence and machine learning

Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) are all words that are sometimes used mutually, although each is a sub branch of the field. In short, AI is the attempt to automate intellectual activity performed by people under normal conditions. Instead of being explicitly coded, the ML system is taught. In other words, machine learning is a branch of AI. DL is a branch of ML, with Neural Networks (NNs) serving as the foundation of DL algorithms (Figure 1).

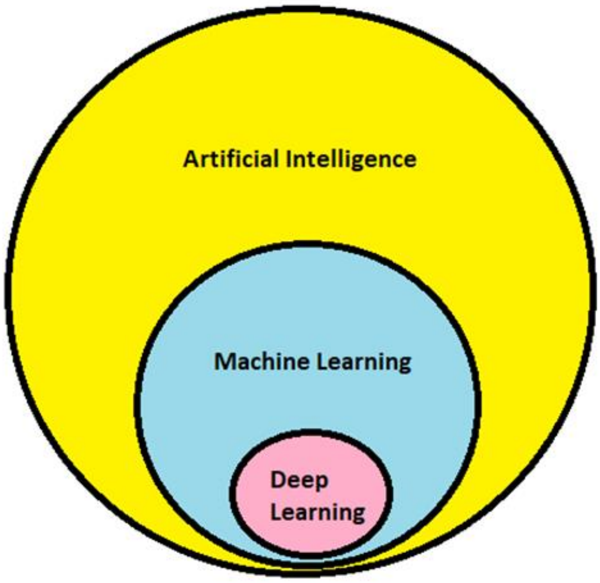

Figure 1. Diagram of the interconnection of AI with ML and DL

AI applications were at the center of the most successful areas of problem-solving computing in the 2010s, and have now become a common part of everyday life [2]. Online advertising targeting and recommendation systems, advanced search machines (such as Google Search) include all AI-based algorithms, such as the Control of internet traffic, Targeted advertising (AdSense, Facebook), virtual assistants (such as Alexa or Siri), facial recognition (Microsoft's DeepFace or Apple's Face ID), automatic language translation (Google Translate, Microsoft Translator), autonomous vehicles (such as drones and cars running independently), and lots of different applications [3].

In robotics, AI is widely used. Localization is the process through which a robot determines its location and builds the surrounding area around it. Robots can learn how to act effectively, not harming the environment they work in based on information they get from sensors and cameras [4, 5].

The robotic arm known as the Dobot Magician has been put to work in a variety of tasks, including research, education, and manufacturing. The following papers and publications on using AI with the Dobot Magician are pertinent.

In the article [6] is explored how robots like the Dobot Magician may be taught to do a variety of manipulation tasks using AI approaches like imitation learning and reinforcement learning. The authors evaluate recent advancements in this area, highlighting some of the difficulties and potential paths.

In the work [7] is shown a dual-arm robot system that can carry out assembly-related tasks using deep learning. The Dobot Magician served as the platform for the authors' studies, which demonstrated how their strategy may enhance the system's functionality and effectiveness.

The design and deployment of an intelligent robotic arm system that can detect and operate items using computer vision and AI algorithms are discussed in this article [8]. The Dobot Magician was employed as the system's robotic arm, and the authors showed off its skills to carry out a variety of duties like object identification and sorting.

These studies show the Dobot Magician's adaptability and potential in a variety of applications, particularly when combined with AI methods. The Dobot Magician can learn to identify objects and do activities on its own by using AI. Further research in this area is needed to explore the full capabilities of this technology and to overcome the challenges that arise when integrating AI and robotics.

2.2. Robotic arms and dobot magician

Robotic arms can be used in all types of processing and manufacturing industries where extremely precise, fast, and repetitive movements are required [9].

Robotic arms of all types are used today at every scale of production, from the detailed assembly of county plates to heavy-duty industries such as automotive production lines, as well as widespread application in pick-and-place process applications (from the conveyor belt). Dobot Magician is a multifunctional robotic arm that can be used on a desktop for hands-on education. It is manufactured to be able to cooperate with various additional devices. This robot can perform interesting functions such as 3D printing, laser cutting, vacuum swallowing, gripper grasping, block-programmed, and mouse-oriented writing and drawing.

There are also ports for attaching additional equipment. It has more than one implementation alternative, as well as over 20 programming languages that aid them, which facilitates studies and development and will increase creativity with no obstacles, Figure 2.

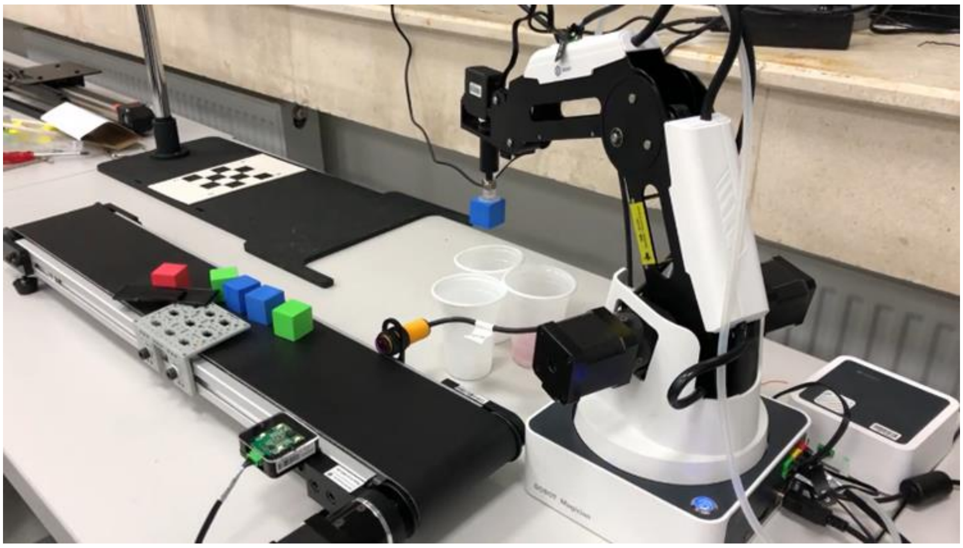

Figure 2. Dobot Magician with the conveyor belt, the color identification sensor and the ultrasonic sensor

The robot control software provides a user interface (Figure 3) very easily understandable. With the "Teaching & Playback" function being presented as the first icon in this interface, it serves to teach the robot the process that will perform by pressing the button found in Dobot and orienting it in coordinates that will be executed later.

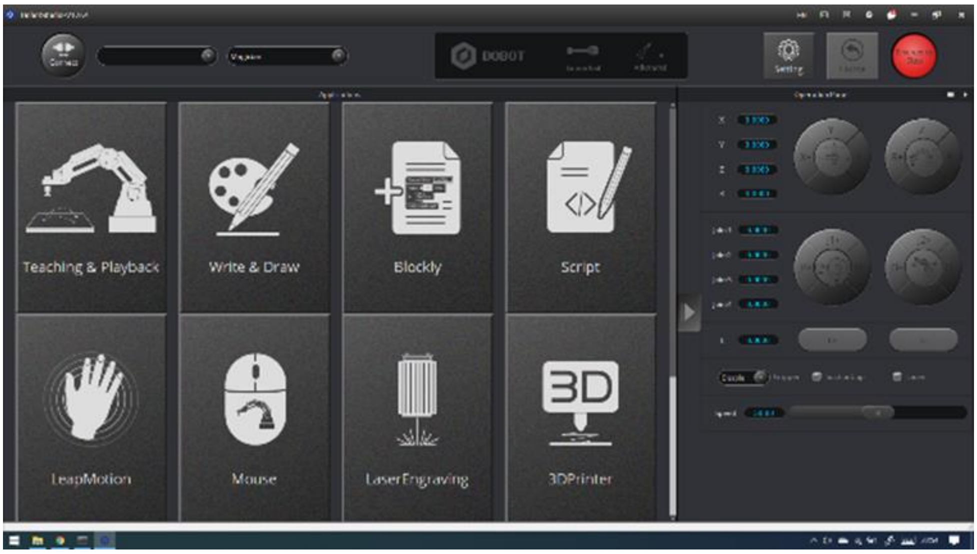

Figure 3. Software of the DOBOT Magician control with basic functions [10]

The robotic arm can be used with a range of equipment, including a linear railway set, and a conveyor belt (conveyor) with controllable speed to perform a range of practical tasks. Programming, Applications, Bluetooth, WiFi, Mouse, and other methods can be used to control it.

This robot also has the vision set that is configured in the Dobot system by creating research, and simulating the AI algorithm (Figure 4). Through Dobot and its additional equipment, all real manufacturing robot applications can be activated [11].

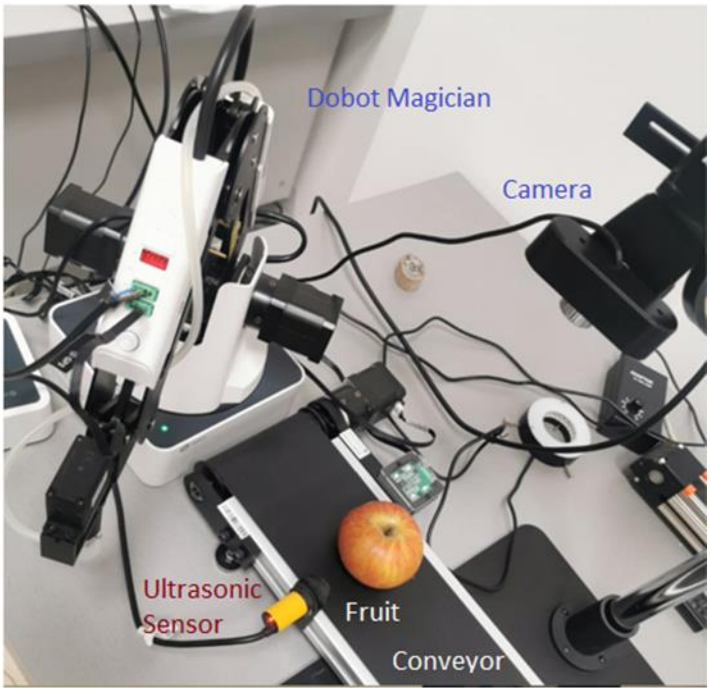

Figure 4. Use of the Dobot Magician equipment

Considering that Dobot's software is for educational purposes it is not possible to add other libraries to solve more advanced tasks. Due to the problem to complete the goal by processing images from the web camera, it was necessary to use the DLL software faults to run the code for processing images, by giving commands to Dobot and receiving inputs from it.

2.3 Python programming language and OpenCV platform

Python is an extensively used, interpreted, object-oriented, high-level programming language with dynamic semantics used for general-purpose programming. Many scientists have abandoned expensive applications and switched to Python. Many IT project testers have started using Python to perform repeatable test procedures.

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and ML software library. OpenCV provides a common infrastructure for computer vision applications to accelerate the use of machine perception in commercial products.

There are more than 2500 optimized algorithms in the library, which includes a complete set of classic and modern computer vision and ML algorithms. These algorithms can be used to detect and recognize faces, identify and classify fruits in our case, etc.

2.3.1. TensorFlow platform

TensorFlow is an end-to-end open-source platform for building ML applications. It permits builders to create ML programs through the usage of various equipment, libraries, and community assets.

It's called TensorFlow because it takes input as a multidimensional array of data, also known as tensors. So, using TensorFlow we can build a kind of flow graph of the operations (called tensor) that we want to perform on that input. Input goes in at one end and then flows through this system of multiple operations and comes out at the other end as output Figure 5.

TensorBoard is a significant feature of TensorFlow. It enables visual and graphical monitoring of what TensorFlow is doing [12].

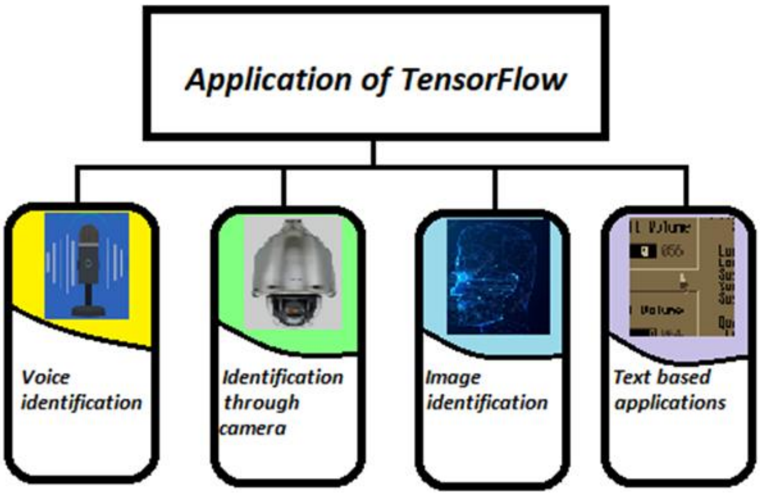

Figure 5. TensorFlow finds its application in everyday life

2.3.2. Anaconda and LabelImg

The advantage of Anaconda is the distribution of the Python programming language and serves for scientific computing, intended to simplify package management and deployment.

If we need to have a version higher than 2.0 of TensorFlow, then along with TensorFlow come other packages giving a warning whether this version is possible for your type of computer and do you want to accept the packages as well other depending on TensorFlow.

LabelImg is an open-source program for visual image labeling. It is developed in Python and has a graphical user interface built with QT (QuickTime). It's a quick and effortless way to identify several hundred images for your next object identification project. To work with LabelImg, we first take some images/samples of objects that we later need to identify.

For example, after taking about 300 images of different fruits in different positions, we classify them into two folders Train and Test. In the Train folder, we put 80% of the images, and in the Test, we put the remaining 20%.

The fastest approach to get LabelImg is to use pip, via Python 3. First, via the link *https://github.com/tzutalin/labelImg* get the LabelImg folder and proceed by opening CMD and going to the location where LabelImg was downloaded. In CMD type "pyrcc4 -o resources.py resources.qrc" to install all packages needed to open LabelImg. After installing the packages, open LabelImg by typing "python labelImg.py", also in CMD to access the labeling application.

This application converts images to the *.xml file type to be readable by TensorFlow and easier to process.

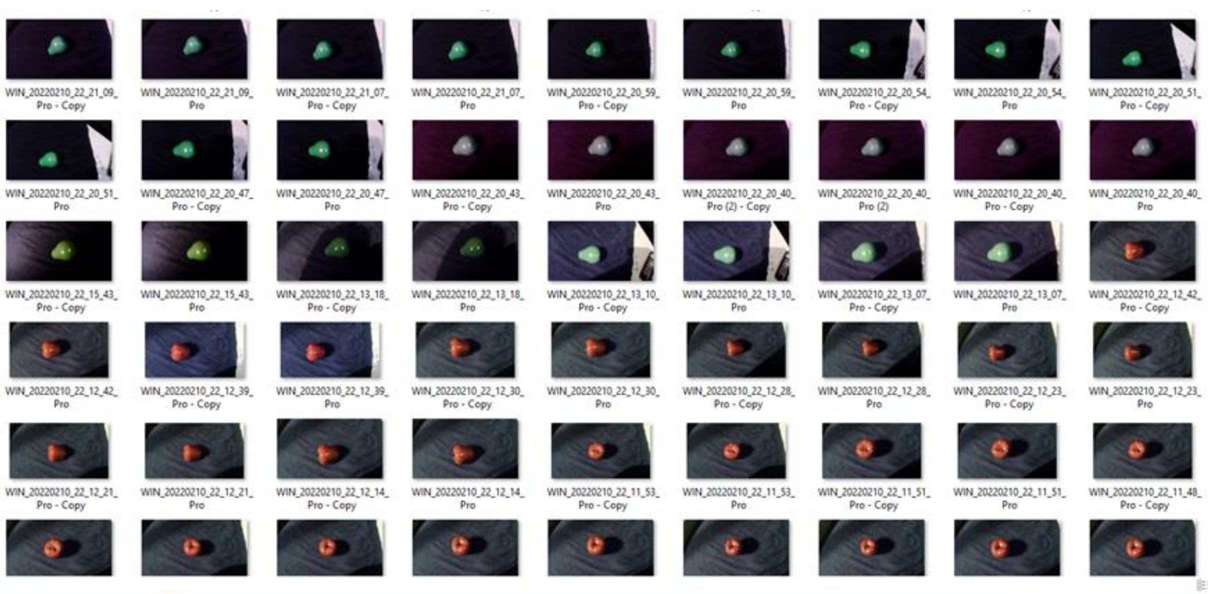

Figure 6. The view of the folder with about 350 images of fruits such as apples, pears, mandarins, lemons and strawberries

2.3.3. The Pseudo-Code used in the paper

In this section, the pseudo-code is presented, which was created for the process of fruit identification in real-time [13].

```
Importing the following libraries← os, cv2, numpy as np, tensorflow as tf, sys, os, shutil, base64, subprocess, time;
Initializing the files locations← sys.path.append("..");
Connecting the Dobot Vision:
 CON_STR = {".."};
Call api = dType.load();

if← "DobotConnect_NoError"
state = dType.ConnectDobot(api, "", 115200)[0];
print("Connection status:", CON_STR[state]);
end
        dType.SetEndEffectorParamsEx(api, 59.7, 0, 0, 1);
        dType.SetInfraredSensor(api, 1 ,2, 1);
Insert variables:
                     MODEL_NAME ←'inference_graph';
                   CWD_PATH ← os.getcwd();
Insert file location and number of objects:
PATH_TO_CKPT  ← os.path.join(CWD_PATH,MODEL_NAME,'frozen_inference_graph.pb');
PATH_TO_LABELS  ← os.path.join(CWD_PATH,'training','labelmap.pbtxt');
NUM_CLASSES = 3;
Training of images of TensorFlow:
              image_tensor ← detection_graph.get_tensor_by_name('image_tensor:0');
Input data:
              if← "VideoCapture(0)"
video = cv2.VideoCapture(0);
else
if ← "VideoCapture(1)"
    video = cv2.VideoCapture(1);
end
Fruit classification:
while
def clasify_fruit();
   if
   if ("Lemon" in a);
dType.SetPTPCmdEx(1:7);
dType.SetEndEffectorSuctionCupEx(1:3);
dType.SetPTPCmdEx(1:7);
dType.SetEndEffectorSuctionCupEx(1:3);
   end
   if ("Apple" in a);
dType.SetPTPCmdEx(1:7);
dType.SetEndEffectorSuctionCupEx(1:3);
dType.SetPTPCmdEx(1:7);
dType.SetEndEffectorSuctionCupEx(1:3);
   end
   if ("Mandarin" in a);
dType.SetPTPCmdEx(1:7);
dType.SetEndEffectorSuctionCupEx(1:3);
dType.SetPTPCmdEx(1:7);
dType.SetEndEffectorSuctionCupEx(1:3);
   end
   if ("Pear" in a);
dType.SetPTPCmdEx(1:7);
dType.SetEndEffectorSuctionCupEx(1:3);
dType.SetPTPCmdEx(1:7);
dType.SetEndEffectorSuctionCupEx(1:3);
   end
   if ("Strawberry" in a);
dType.SetPTPCmdEx(1:7);
dType.SetEndEffectorSuctionCupEx(1:3);
dType.SetPTPCmdEx(1:7);
dType.SetEndEffectorSuctionCupEx(1:3);
   end
```

```
    else
STEP_PER_CRICLE ← 360.0 / 1.8 * 10.0 * 16.0;
          MM_PER_CRICLE ← 3.1415926535898 * 36.0;
    continue
    pass;
When there is no fruit on the conveyer:
while True:
     if (dType.GetInfraredSensor(api, 2)[0]) ← 0;
      dType.SetEMotorEx(1:5);
      Percentage of the fruit identification above 90%:
      min_score_thresh←0.90)
      cv2.imshow('Object detector', frame)
     elseif
     (dType.GetInfraredSensor(api, 2)[0]) ! ← 0);
     dType.SetEMotorEx(1:5);
    end
    for number_of_readings +=1;
   if (number_of_readings == 5);
         print ([category_index.get(value) for index,value in enumerate(classes[0]) if scores[0,index] >  0.9]);
         a ← str([category_index.get(value) for index,value in enumerate(classes[0]) if scores[0,index] > 0.9]);
         classify_fruit();
         number_of_readings ← 0;
    else
    continue
    if cv2.waitKey(1) ← ord('q');
    break
```

3. Results

Knowledge from the field of mechanics, electronics, and programming is used to realize the entire process from arrangement to classification. To carry out the process of identification and classification of fruits, the following steps should be followed:

- First, the selection of fruits that we thought to classify was made, by considering their shape, the possibility to move on the Dobot conveyor, and their weight.
- Second, the images of the selected fruits were taken, images which were taken at every angle of the fruit so that the accuracy is as high as possible. Those fruits will later be classified separately, depending on which type they belong to.
- Third, the generated images are trained, the steps of which are explained in the aforementioned pseudo code, which serves as samples for the software that will do the classification. In this case, the classification software is ready to classify the fruits that have been considered.

After working with the software, it's time to activate the robotic system. The robot, together with the additional conveyor device, will carry out the process of carrying and placing them in different packages, where for each type of fruit there will be a separate package in which they will be collected. Dobot Magician software was used to obtain the coordinates of these packages and the conveyor. After reading the coordinates, we put those data in the code for each type of fruit, we put the coordinates of the package with the same fruit.

Before the entire process is set up into operation, the speed of the sliding belt must be determined and the coordinates of where the fruit to be sorted will be taken. Finally, all the preliminary steps are combined to realize the final process, Figure 6.

4. Discussion

Turn on the robot and all the devices that will participate in the process. The moment the belt starts to move, the fruits are placed on it. The sliding belt continues to move until the first fruit placed on it reaches the sensor for detecting the presence of objects (ultrasonic sensor). The moment the sensor signals the presence of fruit, the sliding belt stops moving. Being aware of the shortcomings that the software may have (which depends on the performance of the computer processor being used for the process) and to achieve the most accurate result, the same fruit is detected up to five times. In Figure 7, 8 and 9, it can be seen how the same fruit can vary in accuracy depending on the image captured by the camera.

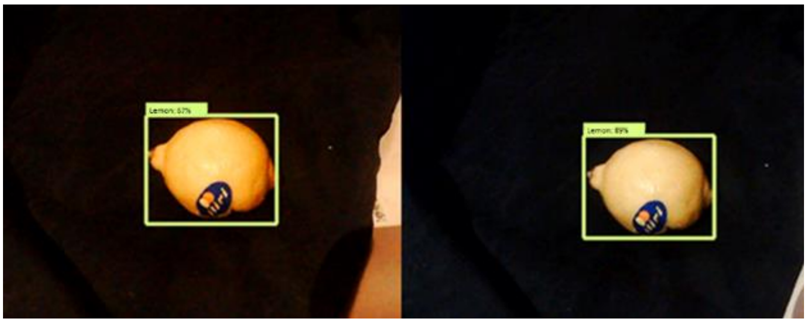

Figure 7. Identified lemon by using the TensorFlow platform

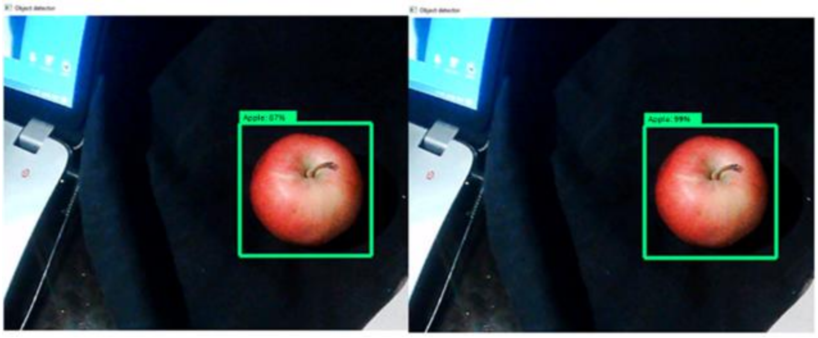

Figure 8. Identified apple by using the TensorFlow platform

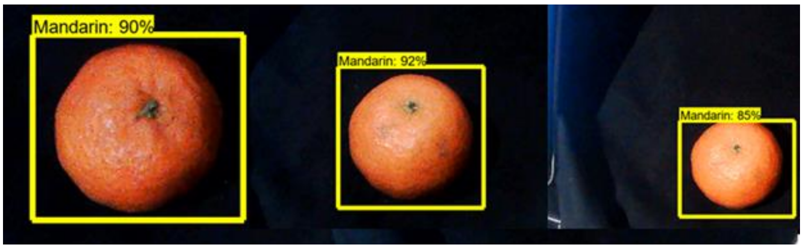

Figure 9. Mandarin identified at different distances and with different accuracy

After the five-fold detection is done, the last data is taken to classify the fruit, i.e., from the fifth time as the final result.

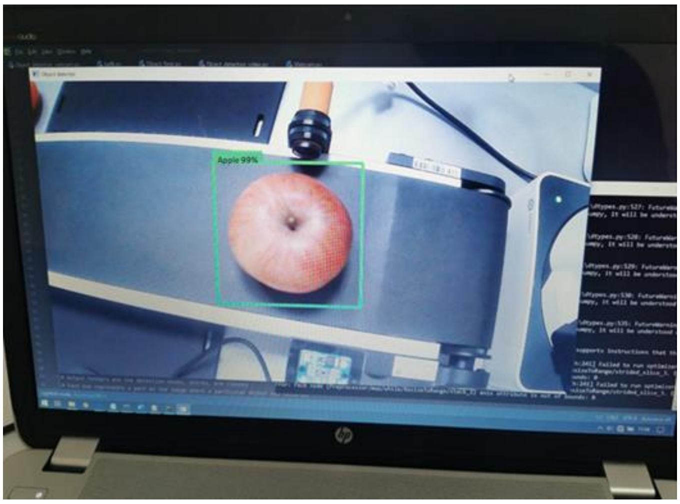

Figure 10. TensorFlow for fruits classification

After this step, the robot is given the command to grab the detected fruit. Based on the data obtained from the classification, the fruit that the robot has caught is placed in the coordinates defined for that type of fruit, Figure 10.

With the implementation of the given coordinates, the end of the fruit classification process is reached.

In this case, the robot returns to its zero point and the sliding belt starts moving, repeating the whole process for the classification of the next fruit.

If the object to be detected is not one of the selected fruits, the sliding belt continues to move, letting it know that there are no coordinates for this type of object, and that fruit moves off the conveyor without being picked up by the Dobot.

5. Conclusion

With this work, the primary goal was fulfilled, which was to develop an automated line for the identification and classification of fruits and vegetables using artificial intelligence. Using the robotic system and the software presented in this paper, with the help of TensorFlow fruit classification can be achieved with higher accuracy and quantity than would be achieved if the process is carried out manually.

The robotic system could be advanced by adding other robots, which are dedicated to a certain work process. Robots can be added for the transport of fruit packages, which will be sorted, robots that pack the fruits after sorting, and robots for their transport.

The robotic system can also make advances in classification, classifying a fruit according to the different sizes it can have, its weight, as well as its shape. But why not get rid of fruits that are defective or not ripe enough?

With all these advancements one can have a complete production line, not only for the agricultural industry but also for other industries in which product classification can took place.

Acknowledgement

This research has been carried out at the Laboratories of Mechatronics of the Faculty of Mechanical Engineering, University of Prishtina (FME-UP), and we would like to appreciate the support of the Staff of the FEM-UP, and the Faculty of Applied and Economics Sciences, Albanian University.

Funding

This research received no external funding.

Author contributions

Arbnor Pajaziti: Conceptualization, Methodology, Software **Fatmir Basholli:** Data curation, Writing-Original draft preparation, Software, Validation, Visualization **Ylber Zhaveli:** Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

1. https://zentronlabs.com/systems/optical-fruit-grading-and-sorting-machine
2. Luger, G. F. (2005). Artificial intelligence: structures and strategies for complex problem solving. Pearson education.
3. Russell, S. J. (2010). Artificial intelligence a modern approach. Pearson Education, Inc..
4. Nilsson, N. J. (2010). The quest for artificial intelligence: A history of ideas and achievements. Cambridge University Press.
5. Bishop, C. M. (2016). Pattern recognition and machine learning. Springer New York.
6. Wang, Y., Zhu, S., Zhang, Q., Zhou, R., Dou, R., Sun, H., ... & Zhang, Y. (2021). A visual grasping strategy for improving assembly efficiency based on deep reinforcement learning. Journal of Sensors, 2021, 1-11. https://doi.org/10.1155/2021/8741454.
7. Ying, K. C., Pourhejazy, P., Cheng, C. Y., & Cai, Z. Y. (2021). Deep learning-based optimization for motion planning of dual-arm assembly robots. Computers & Industrial Engineering, 160, 107603. https://doi.org/10.1016/j.cie.2021.107603
8. Shahria, M. T., Sunny, M. S. H., Zarif, M. I. I., Ghommam, J., Ahamed, S. I., & Rahman, M. H. (2022). A Comprehensive Review of Vision-Based Robotic Applications: Current State, Components, Approaches, Barriers, and Potential Solutions. Robotics, 11(6), 139. https://doi.org/10.3390/robotics11060139
9. Pajaziti, A., Buza, S., Gojani, I., Safaric, R., & Kopacek, P. (2009). Cost Oriented Robots for Kosovo. INHALT Seite, 58.

10. Dobot Magician Manuals. https://www.dobot.cc/dobot-magician/product-overview.html
11. Intelligent Robotic Arms Provider. DOBOT. https://www.dobot.cc/dobot-magician/product-overview.html
12. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.
13. TensorFlow (2021). Object Detection Camera Demo. https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/auto_examples/object_detection_camera.html#sphx-glr-auto-examples-object-detection-camera-py