Specific Object Picking Robotic Arm Using Haar Cascades

Mohammad Junaid Malik¹, Mathamsetti Aditya¹, D Rohith Surya Teja Varma¹, Gandharla Sidhartha¹, P Sohan¹, S.Govindha Rao¹ and Myasar Mundher Adnan²

¹Gokaraju Rangaraju Institute of Engineering and Technology, Department of Computer Science, Hyderabad, Telangana, India.
²the Islamic University, Najaf, Iraq

Abstract. This project shows a robotic arm that can pick up objects. It was made with accuracy and speed in mind for use in factories. The robotic arm is very good at picking up metal nuts. It uses cutting edge technologies, like Haar cascade to find objects and inverse kinematics to figure out angles very accurately, to make its moves more exact and dexterous. A powerful computer vision method called Haar cascade is used to find metal nuts in the robotic arm's working environment. To do this, positive and negative pictures are used to train a Haar cascade classifier, which makes a model that can recognize the unique traits of metal nuts. The object recognition process makes sure that the nuts are located correctly, which lets the robotic arm deal with them in a controlled and precise way. The robotic arm uses inverse kinematics along with Haar cascading to figure out the exact joint angles needed for smooth, controlled movement. Inverse kinematics is very important for figuring out the joint setups that are needed to place the robotic arm's end-effector exactly on top of the metal nuts that have been found. Haar cascading for finding objects and inverse kinematics for accurate angle estimates work together to make sure that the robotic arm can pick up metal nuts in a variety of space arrangements. This combined system is a high-tech way to automate the picking of specific items in manufacturing and assembly lines. It shows how industrial automation can be used to improve accuracy, efficiency, and flexibility.

1 Introduction

More people are buying things and being more aware of quality and safety, which has made people want better quality in market goods. Costs need to go down because more people want customization and there is more competition. This can be done by making the goods better, cutting down on waste during production, letting customers make changes, and making the production process go faster.
After a certain amount of work, quality control by people is no longer useful. When output gets higher, it's important to have a machine that can act like a person. One way to look at the vision system is as a simulation system that combines the human eye (camera) with intelligence (computer). Machine Vision (MV) is the technology that lets you analyse pictures for things like automatic checking, process control, and robot guiding in the business world. Vision Sensors and Machine Vision Systems look at pictures to check things like appearance, character, placement, and for flaws. There are many ways that the machine vision systems can be used because they are flexible and have many different features. Several real-time limits are put on picture processing when vision systems are used for checking and motion control. However, vision measuring systems are better than traditional measuring systems because machine vision software and hardware are getting better and cheaper all the time. These vision systems can exactly measure things like size, angle, location, direction, colour, and more. The best thing about a machine vision-based system is that it can check things without touching them, which is useful when it's hard to take readings by touching something. Better efficiency and better-quality control are also helped by Machine Vision technology. This makes businesses that do not use vision systems face tough competition. Vision-based systems are not just used in the fields we've talked about here; they're also used in many more, like welding, where machine vision is used to find and label metal flaws in places where human review would not work well. In addition, it keeps people from contaminating clean rooms and keeps workers safe in dangerous places. As this area has grown, computer vision has grown to include systems that can recognize how people walk.

2 Literature Review

Object Sorting Robotic Arm Based on Colour Sensing[1]:
Robotic arm system utilizing an Arduino microcontroller and color sensor for object sorting based on color. It addresses the need for technological advancements in various applications, particularly in material transportation processes, where manual labor is time-consuming and hazardous. Through a review of related literature, it highlights the potential of robotic arms in mitigating these challenges. The proposed system employs an Arduino Uno board and a TCS3200 color sensor to detect RGB colors, enabling the robotic arm to classify and arrange objects efficiently. The implementation involves essential hardware components such as servo motors and cables. Flowcharts and graphics illustrate the system's operational steps, including color detection, object selection, transportation, and sorting. The study suggests diverse applications such as item detection and color-based sorting.

Design and Development of a Robotic Arm[2]:
Designing a safe robotic arm for feeding especially challenged or senior individuals. It addresses issues related to gripper positioning and kinematics for safe interaction with users. Previous designs utilizing string-driven or slider-crank mechanisms faced limitations. A 5-DOF articulated robotic arm is proposed, employing kinematic modeling in MATLAB for precise control from base to gripper. Arduino Mega2560 serves as the central control unit, with force detectors at the gripper and potentiometers at joints for object manipulation and motor position sensing, respectively. The aim is to create an
easily controllable and lightweight robotic arm suitable for assisting individuals with daily tasks.

**A Comprehensive Analysis of Image Edge Detection Techniques[3]:**

Edge detection is crucial in image processing, where sudden brightness or color changes signify object boundaries. This method extracts shape and reflectance information from images, vital in computer vision, pattern recognition, and human vision. Various techniques exist, influenced by factors like intensity, lighting, and noise. This paper compares colorful edge detection methods, analyzing their efficacy in identifying image edges. Edge detection plays a pivotal role in computer vision, aiding in object recognition and pattern analysis.

**Using Tkinter of Python to Create Graphical User Interface (GUI)[4]:**

Python was selected for GUI development based on prior experience with a similar language at LNLS. Py4Syn, a Python 3.4 library, facilitates synchronized bias control and data storage for analysis. It interfaces with EPICS IOCs and PVs via PyEPICS, simplifying complex calculations with NumPy and data visualization using Matplotlib. Despite CS Studio's recommendation for EPICS management, integrating Py4Syn proved inefficient. Consequently, a GUI was designed using Tkinter due to its ubiquity and ample online resources. LNLS's initial experience with Tkinter during DXAS beamline renovation (2014-2015) informed this decision, emphasizing its suitability for Python GUI development.

**A Comparison Analysis Of Pwm Circuit With Arduino[5]:**

The interface between control systems and power circuits is facilitated by Pulse Width Modulation (PWM), pivotal for various applications like Arduino, CPLD, and FPGA platforms. Arduino, an open-source prototyping platform, offers a user-friendly programming interface and digital I/O pins capable of PWM functions. It can interface with diverse sensors and control electronic devices such as motors and lights. Arduino's software environment, utilizing an enhanced C library, supports easy programming. For PWM generation, Xilinx FPGAs provide advantages like flexibility, quick circuit updates, and cost-effectiveness compared to fixed circuitry. FPGAs can generate PWM signals using VHDL, a hardware description language, or through layered processors. This enables efficient circuit design and rapid prototyping.

**Detection & Distinction Of Colors Using Color Sorting Robotic Arm In a Pick & Place Mechanism[6]:**

The TCS3200 color detector is a photodiode array sensor widely used for color sensing in robotics. It measures red, green, blue, and clear light intensity, allowing precise color detection. By illuminating objects with white light and analyzing reflected light, it provides RGB values proportional to each color's intensity. Interfacing with the PIC Microcontroller (18F452), the robotic arm initiates color sensing and receives RGB values. These values are compared to predefined thresholds or a color database to determine the object's color accurately. Based on this detection, the arm executes specific actions like picking up objects.
using servo and stepper motors. Detailed diagrams and mechanical designs illustrate how the system integrates color detection with the arm's movements. This robust integration facilitates precise object manipulation and transportation based on their colors, enhancing the arm's versatility and utility in various applications.

Object Sorting Robotic Arm Based On Coloursensing[7]:
Agricultural machinery such as rice, bean, and peanut sorters commonly employ color sorters for efficient sorting. These sorters find applications in various industries, including sorting colored nuts, bolts, plastic granules, and quartz sand, reducing manpower, expenses, and manual effort. To control the robotic arm, an LPC2138 32-bit RISC microprocessor on an ARM7 platform is utilized. The TCS3200 light intensity-to-frequency converter detects item colors and generates variable frequency square waves accordingly. The robotic arm is powered by four DC servo motors, enabling precise and efficient sorting operations across diverse applications.

3 Methodology

Inverse kinematics[1]:
Many fields, including robotics, computer graphics, and animation use inverse kinematics (IK) to figure out the joint setups of a robot or flexible structure that are needed to get the end-effector to the position and direction that is wanted. For easier understanding, it is the problem of finding out how to move the different parts (joints) of a system so that it can reach a certain point in space. Within the field of robotics, a robotic arm or manipulator is made up of several parts, each with its own range of motion. Given a target position and direction in space for the end-effector (for example, the robot's hand), the objective of inverse kinematics is to locate the joint angles that will exactly place the end-effector where it is needed.

Finding the end-effector's position and direction using known joint angles is what forward kinematics, which is the exact opposite of inverse kinematics, does. As opposed to inverse kinematics, which is easier to understand, it requires solving equations to find the joint angles that lead to a certain end-effector motion.

Different ways exist to resolve inverse kinematics problems, ranging from mathematical methods for more complicated and higher-dimensional systems to analytical methods for simpler systems. As a result of mathematical formulas, analytical answers directly connect joint angles to the position of the end-effector. Nevertheless, these are frequently limited to certain types of robotic systems and might not work in more complicated situations.

Numerical methods, like the Cyclic Coordinate Descent (CCD) algorithm or repetitive methods like the Jacobian transpose method, are often used for smaller problems. Iteratively changing joint angles in these methods closes the gap between the current position of the end-effector and the goal position. Although computer methods are flexible, they may need to be fine-tuned and depend on what the joint angles were thought to be at first.

In robots and computer graphics, specific jobs like trajectory planning, motion control, and animation require exact placement of end-effectors in order to be done quickly and
correctly.

\[
\begin{align*}
\theta &= \tan^{-1}\left(\frac{r}{d}\right) \\
g &= \sqrt{r^2 + b^2} \\
\delta &= \cos^{-1}\left(\frac{a^2 + g^2 - f^2}{2ag}\right) \\
\beta &= \theta + \delta
\end{align*}
\]

**Fig. 1.** Inverse Kinematics
Formula-1

\[
\begin{align*}
\gamma &= \cos^{-1}\left(\frac{a^2 + b^2 - g^2}{2ab}\right)
\end{align*}
\]

**Fig. 4.** Inverse Kinematics
Formula-4

\[
\begin{align*}
r &= \sqrt{x^2 + y^2} \\
x &= r \cos \alpha \\
\Rightarrow \alpha &= \cos^{-1}\left(\frac{r}{x}\right)
\end{align*}
\]

**Fig. 2.** Inverse Kinematics
Formula-2

\[
\begin{align*}
i &= \sqrt{i^2 + k^2} \\
\sigma &= \tan^{-1}\left(\frac{k}{i}\right) \\
\phi &= 90 - \sigma \\
\cos \phi &= \frac{i^2 - k^2}{2i} = g = \sqrt{i^2 + b^2 - 2ib\cos \phi} \\
\theta &= \cos^{-1}\left(\frac{f^2 + g^2 - i^2}{2fg}\right)
\end{align*}
\]

**Fig. 3.** Inverse Kinematics
Formula-3

**Haar cascade[2]:**

Haar cascades are a type of machine learning object recognition that can be used to find things in pictures or videos. Viola and Jones proposed this method in their important 2001 work. The idea behind Haar cascades is to show what an object is like with a set of simple, rectangle features called Haar-like features.

These traits that look like Haar come from the Haar wavelet, which is a mathematical idea used in signal processing. Each feature is a different pattern of changes in the image's strength. As an example, a Haar-like feature could show the difference between the sum of the pixel values in one area and the sum of the pixel values in an area next to it. A lot of positive and negative cases are needed to train a Haar cascade. Pictures of the target object are in positive examples, while pictures of the object are not in negative examples. Using Haar-like traits, the program learns to tell the difference between these two groups. After being taught, the Haar cascade can be used on new pictures or video clips to find the target object. The word "cascading" comes from the way the features are set up in a number of steps. A different set of features is added to the picture at each stage. If the area makes it through all the steps, it is called a discovery. One of Haar cascades' strengths is how quickly they can be computed. Because the cascade structure quickly gets rid of non-object parts, it can be used in real-time situations. Additionally, Haar cascades can work with some occlusions and are not affected by changes in lighting. For many years, Haar cascades have been used in many computer...
vision tasks, such as recognizing faces, people, and objects. The well-known computer vision library OpenCV has tools for teaching and using Haar cascades, which means they can be used in many situations.

**Arduino Uno[3]:**
Arduino is a popular open-source electronics platform for physical computing due to its simplicity and adaptability. Arduino uses microcontroller boards to read sensor inputs and operate lights, motors, and other actuators. The platform is popular in teaching, prototyping, and DIY.

Arduino is known for its easy-to-use IDE. The Arduino IDE facilitates board code authoring using a customized C++ programming language. This abstraction helps novices and experts learn programming without diving into low-level languages.

Arduino Uno boards include several digital and analog input/output pins. Electronic components link to these pins, allowing users to readily interact with the physical environment.

Arduino's hardware simplicity and broad ecosystem of sensors, actuators, and shields make it a versatile platform. Another Arduino strength is its community. Active and collaborative users contribute information, code, and project ideas on the site. This collaborative atmosphere has created a massive resource of tutorials, documentation, and forums to help users find assistance and inspiration for their projects.

Arduino has several libraries that facilitate component interaction in addition to its hardware. Arduino's wide library support simplifies development for internet connectivity, displays, and complicated algorithms. Also interesting is Arduino's openness. The hardware specs and source code are open source, enabling third-party manufacturers to create compatible gear. This openness has produced a variety of Arduino-compatible boards and shields for specialized uses.

In conclusion, Arduino is a leader in embedded systems and physical computing, providing a user-friendly platform for hobbyists, students, and professionals to realize their electrical ideas. Its affordable hardware, friendly community, and simple IDE have made it a popular choice for projects and learning.

**Fig. 5. Arduino Uno R3**

**Servo motor[4]:** There are a lot of hobbyists and robotics users who use the MG996R model of servo motor. It is specifically a metal-gearced servo, which is what the "MG" in its name stands for. The number "996" is its model number, and the letter "R" means that it has been
changed or improved. The MG996R is known for being long-lasting and well-built. Since it has a metal gear train, it is stronger and can be used in situations where stability and high power are important.

This servo motor is often used in robotic projects, remote-controlled cars, and planes. With a torque rate of about 10 kg/cm, it has enough power for jobs that need a lot of force. MG996R also works with a pulse-width modulation (PWM) signal, which means it can be used with many microcontrollers and integrated systems.

It is important to keep in mind, though, that the MG996R may have some backlash or play in its gears, which can affect its accuracy in some situations. Users often like how cheap it is, which makes it a good choice for students and artists who want to do robots and automation projects. The MG996R servo is famous in the maker and robotics groups because it has a good mix of speed, longevity, and price.

![MG996R Servo Motor](image)

**Fig. 6.** MG 996R Servo Motor

**Python GUI[5]:**

In conclusion, Tkinter, PyQt, and Kivy improve Python's GUI development. Tkinter, a standard library component, is popular because of its simplicity. It has several GUI development tools including buttons, labels, entry forms, and layout management tools like `pack()`, `grid()`, and `grid()` for widget placement. Event handling is crucial in GUI development, and Python's flexibility in real-time connection to events like button clicks and key presses is highlighted. Python's threading and asynchronous programming capabilities avoid GUI freezes during resource-intensive activities like file I/O and network connectivity. Python GUIs may be themable for consistent and attractive displays across devices. When combined with NumPy for scientific computing or Matplotlib for data visualization, Python GUI apps have expanded possibilities. Finally, Python's GUI development tools, especially Tkinter, allow developers to construct interesting and visually beautiful cross-platform apps. Python's flexibility and ability to interface with other tools make it excellent for building strong and attractive programs for a variety of operating systems.
4 Results

Fig. 7. DOF Robotic Arm

Fig. 8. DOF Robotic Arm

Fig. 9. DOF Robotic Arm
**Fig. 10.** Arduino UNO

**Fig. 11.** Classification of objects using Haar Cascades Classifier

<table>
<thead>
<tr>
<th>Object</th>
<th>Detected</th>
<th>Sorted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medium size nut</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Small size nut</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Large size nut</td>
<td>Yes</td>
<td>yes</td>
</tr>
<tr>
<td>Shiny nut</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Rusty nut</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Bolts</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 2. Classification of nuts detected using Haar Cascades.

<table>
<thead>
<tr>
<th>Case No</th>
<th>Placed</th>
<th>Detected</th>
<th>Sorted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10 nuts</td>
<td>10 nuts</td>
<td>10 nuts</td>
</tr>
<tr>
<td>2</td>
<td>12 nuts</td>
<td>11 nuts</td>
<td>11 nuts</td>
</tr>
</tbody>
</table>

5 Conclusion

In conclusion, project on a robotic arm with image processing for object handling was successful. Combining computer vision algorithms with the mechanical arm allows for accurate and efficient item recognition and handling. Real-time object recognition and localization demonstrate the system's grasping and transporting accuracy. We faced image processing issues such as capture, preprocessing, feature extraction, and object identification throughout the project. We designed a robust system that can handle items of various sizes, shapes, and color using modern methods and technologies. In particular, image processing technologies, such as Haar cascade exploration, have increased the robotic arm's autonomy and flexibility. Autonomous object identification lets the machine traverse dynamic settings without operator intervention. This degree of automation has intriguing applications in manufacturing, shipping, and healthcare, where precise object manipulation streamlines processes.

References