



Article

# An FPGA-Based Hardware Low-Cost, Low-Consumption Target-Recognition and Sorting System

Yulu Wang <sup>1</sup>, Yi Han <sup>1</sup> , Jun Chen <sup>1</sup>, Zhou Wang <sup>2,\*</sup> and Yi Zhong <sup>1</sup>

<sup>1</sup> School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China; 315348@whut.edu.cn (Y.W.); hanyi@whut.edu.cn (Y.H.); hatake@whut.edu.cn (J.C.); zhongyi@whut.edu.cn (Y.Z.)

<sup>2</sup> SAIC GM Wuling Automobile Company Limited, Liuzhou 545007, China

\* Correspondence: zhou.wang@sgmw.com.cn

**Abstract:** In autonomous driving systems, high-speed and real-time image processing, along with object recognition, are crucial technologies. This paper builds upon the research achievements in industrial item-sorting systems and proposes an object-recognition and sorting system for autonomous driving. In industrial sorting lines, goods-sorting robots often need to work at high speeds to efficiently sort large volumes of items. This poses a challenge to the robot's real-time vision and sorting capabilities, making it both practical and economically viable to implement a real-time and low-cost sorting system in a real-world industrial sorting line. Existing sorting systems have limitations such as high cost, high computing resource consumption, and high power consumption. These issues mean that existing sorting systems are typically used only in large industrial plants. In this paper, we design a high-speed, low-cost, low-resource-consumption FPGA (Field-Programmable Gate Array)-based item-sorting system that achieves similar performance to current mainstream sorting systems but at a lower cost and consumption. The recognition component employs a morphological-recognition method, which segments the image using a frame difference algorithm and then extracts the color and shape features of the items. To handle sorting, a six-degrees-of-freedom robotic arm is introduced into the sorting segment. The improved cubic B-spline interpolation algorithm is employed to plan the motion trajectory and consequently control the robotic arm to execute the corresponding actions. Through a series of experiments, this system achieves an average recognition delay of 25.26 ms, ensures smooth operation of the gripping motion trajectory, minimizes resource consumption, and reduces implementation costs.

**Keywords:** autonomous driving system; product design; development and prototyping; motion and path planning; computer vision for manufacturing



**Citation:** Wang, Y.; Han, Y.; Chen, J.; Wang, Z.; Zhong, Y. An FPGA-Based Hardware Low-Cost, Low-Consumption Target-Recognition and Sorting System. *World Electr. Veh. J.* **2023**, *14*, 245. <https://doi.org/10.3390/wevj14090245>

Academic Editor: Zonghai Chen

Received: 13 August 2023

Revised: 28 August 2023

Accepted: 1 September 2023

Published: 4 September 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Industrial robots were first created in the 1950s and have since been developed for about 60 years. Due to their ability to replace humans in performing relatively simple and repetitive tasks, they are widely used in various major fields [1]. In recent years, FPGAs (Field-Programmable Gate Arrays) have proven to be advantageous in the field of image processing due to their ability to perform real-time pipeline operations and achieve high real-time performance. FPGAs are widely utilized across industries for their programmability, enabling rapid prototyping, iterative development, and easy adaptation to changing requirements. The trend of employing FPGAs for image processing in machine vision is increasing. Currently, machine vision utilizing neural networks for target recognition and sorting is extensively adopted in sorting lines within automated factories. GPUs (Graphics Processing Units) are designed to accelerate image and video processing. While most neural networks run on GPUs, their high power consumption and cost are limiting factors [2], making them unsuitable for extensive use in large automated factories. However, in various

scenarios, such as small platforms, the substantial power consumption and high cost are not feasible for users. Neural network-based machine vision for target recognition and sorting often demands significant resources, such as CNNs (Convolutional Neural Networks) [3], which not only consume substantial resources but also slow down processing speed due to the multitude of multiplication units required for convolutional operations.

After the industrial robot system completes the recognition task, ensuring the robot's accuracy in locating the grasping target and improving the robustness of the design method remain challenging tasks. The traditional approach employing robotic arms, which use predetermined movement routes and grasping positions for task execution, exhibits limited adaptability [4]. In contemporary industrial production, time-optimal planning algorithms are utilized to complete corresponding paths in the shortest time, such as PSO (Particle Swarm Optimization) algorithms and genetic algorithms [5]. In multi-objective optimal trajectory planning, the accuracy of late curve control tends to diminish as runtime increases, resulting in reduced accuracy of the robotic arm system and increased overall mechanical loss [6]. This scenario is less than ideal for practical engineering applications. Considering the aforementioned factors, the demand for designing a low-cost, low-power consumption, low-resource consumption, high real-time item identification and sorting system with tangible economic benefits and practical value is particularly prominent for small and medium-sized enterprises or individuals. The system designed in this paper possesses the following advantages.

- The hardware configuration of the entire system includes an FPGA development board, a vision camera, and a robotic arm grasping system. This setup eliminates the need for high power consumption and a large computer mobile platform, making it well suited to the requirements of small enterprises or individuals;
- The entire system is built on an FPGA, utilizing fundamental components such as counters, registers, and LUT (Look-Up Table) modules. This design significantly reduces system resources and power consumption when compared to GPUs;
- In terms of algorithm implementation, the vision component employs traditional morphology for real-time parallel processing through pipeline operations, while the grasping segment utilizes an enhanced cubic B-spline algorithm for trajectory planning. This comprehensive system exhibits strong real-time performance and high operational stability.

## 2. Analysis of Related Works

For items on industrial assembly lines, the application of industrial robots for recognition and grasping has garnered significant attention from researchers. In this section, we will provide a detailed analysis and summary, categorizing it into the visual component and the grasping segment.

### 2.1. Visual Section

The solutions in the vision section can be broadly divided into two categories: traditional morphological-recognition algorithms and neural network-based image-recognition algorithms [7]. Many researchers have proposed various algorithms and methods. Tang et al. [8] proposed a method to extract color features and shape features of items and used the BP (Backpropagation) neural network for the classification and recognition of items. The method has an accuracy rate of about 95%. The paper also compared the classification by selecting only color features and the classification by selecting only shape features; the accuracy rate was less than 84%, underscoring the importance of considering multiple features for classification judgments. In comparison to [8], Wang et al. [9] added the extraction of texture features, resulting in improved recognition accuracy to some extent. Both of the above methods are implemented based on software without hardware deployment, which imposes certain limitations on recognition speed. For image recognition, the deployment of neural networks in hardware generally involves GPUs, ensuring the algorithm's implementation and fast recognition speed, but at the cost of higher power and resource

consumption [1]. In [10], the neural network is deployed in hardware, and image segmentation is performed using a bounding box regression algorithm. Deploying the neural network in hardware implementation can effectively enhance the real-time performance of the system.

In contrast to the neural network-based image-recognition algorithm, the traditional morphological-recognition algorithm does not require a large number of computing units as in a neural network, resulting in reduced resource consumption.

## 2.2. Grabbing Section

Numerous domestic and international experts and scholars have conducted extensive in-depth studies and research on various aspects of robotic arm trajectory planning, including time-optimal, energy-optimal, impact-optimal, and integrated optimal approaches.

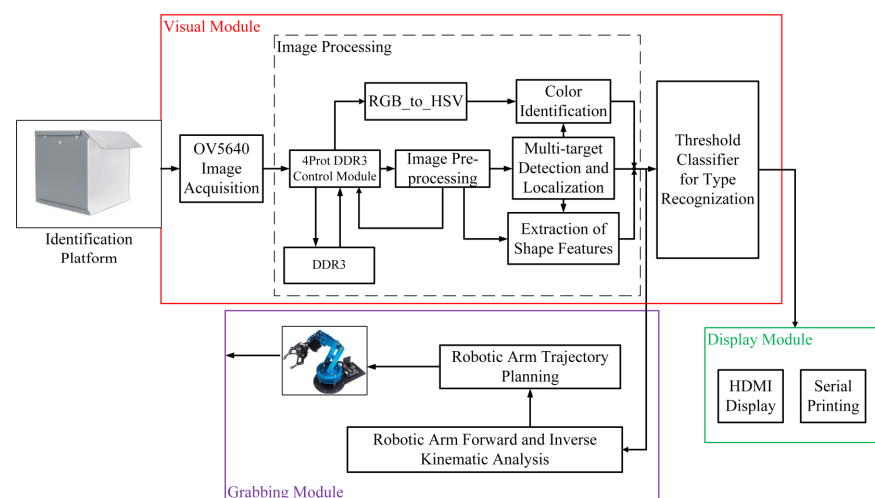
Chowdhury et al. [11] reduced the multivariate interpolation problem to a joint polynomial approximation issue, thereby performing polynomial interpolation for robotic arm trajectory planning. However, this method requires specific constraints to be configured within each segmentation interval, leading to increased complexity in calculations, especially with a large number of nodes. Park and Kyung-Jo [12] employed Fourier series and polynomial functions for path design to mitigate vibration. Cao et al. [13] analyzed convergence based on B-spline curves but noted that errors in B-spline fitting could lead to end effector deviations.

Building upon the analysis of the aforementioned relevant works, this paper employs a threshold classifier, instead of a neural network, for classification judgment in the recognition component. This approach significantly reduces resource consumption and minimizes the required number of operations. For the grasping segment, the paper adopts the improved cubic B-spline curve for trajectory planning, benefiting from continuous second-order derivatives and adherence to the path points, thus satisfying the requirements of robotic arm trajectory planning.

In FPGA hardware implementation, the unique advantage of high-speed parallel computing is harnessed to achieve a rapid recognition system. While the speed may be slightly lower than that of GPU recognition, FPGA's power consumption is significantly reduced.

## 3. Overall System Design

The overall design of the proposed system adopts a morphological-recognition method. The system's overall block diagram is depicted in Figure 1. The framework comprises three main modules: the visual module, the grasping module, and the display module.



**Figure 1.** System framework of visual module, the grabbing module, and the display module.

In the visual module, the system initially captures the scene using an OV5640 camera. The OV5640, manufactured by OmniVision Technologies Inc. (Santa Clara, CA, USA),

500 W pixels, supports QSXGA photo function (up to  $2592 \times 1944$ ), video image output (up to 1080P), autofocus function, and automatic exposure control (AEC). To extract the shape features of fruit objects, the system computes the background frame difference when the scene changes. This algorithm is implemented using a four-channel DDR3 (Double Data Rate 3) control module, employing the morphological filtering algorithm to eliminate noise and smooth the image curve of the items. DDR3, a type of synchronous dynamic random-access memory (SDRAM), can be used in computer systems and succeeded DDR2. It was prevalent in personal computers, laptops, servers, and other computing devices. Subsequently, multi-target detection and localization algorithm enable the segmentation, position extraction, and quantitative detection of multiple items. Meanwhile, the color identification of objects is achieved through HSV (Hue, Saturation, and Value) color space transformation combined with location information. The HSV color model is widely utilized in computer graphics, image processing, and color-related applications due to its intuitive and perceptually uniform color representation. Extracted object shape features, along with color features, facilitate object recognition using a pre-trained threshold classifier. Ultimately, the visual module identifies the object and transmits the corresponding object coordinate data to the six-degrees-of-freedom robotic arm for further grasping function.

In the grasping module, the FPGA completes the forward and inverse kinematic analysis of the robotic arm and the trajectory planning. Subsequently, the robotic arm receives signals from the FPGA, enabling the sorting of corresponding objects on a simulated assembly line.

The display module showcases information from both the visual and grasping modules. The HDMI display presents details such as color, type, quantity, coordinate information, and whether the fruit has undergone sorting.

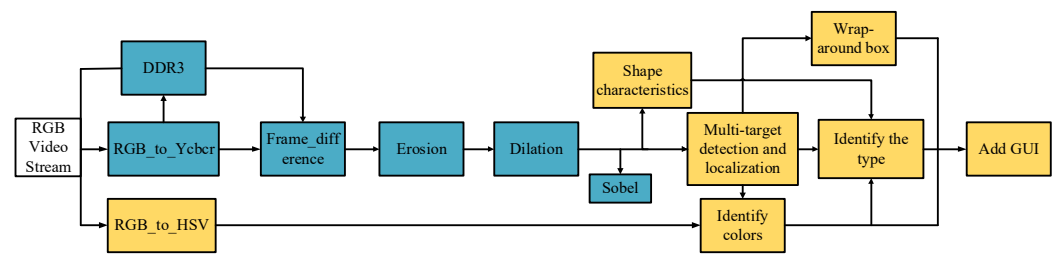
This system is designed around the characteristics of FPGA to fully utilize its computation and extension abilities. The image processing algorithm is implemented on an FPGA, with the OV5640 and the robotic arm serving as an external device to capture signals and perform the sorting operation. Once image processing is completed, a threshold classifier which is implemented in FPGA is utilized to determine the object's category and color. The positional information is processed by FPGA to perform forward and inverse kinematic analysis and robotic arm trajectory planning to grab objects.

Through system architecture design and overall debugging, the system meets the expected requirements. The system is applicable to a large range of target categories. Because of the wide variety of fruits with different colors and shapes, the system utilizes fruits as representative objects for identification and sorting purposes. The system encompasses the following major steps in the overall operation, and their details are represented in the following four subsections.

- Image processing;
- Threshold classifier;
- Robotic arm forward and inverse kinematic analysis;
- Robotic arm trajectory planning.

### 3.1. Image Processing

The image processing module is the core of this system, as shown in Figure 2. After real-time detection by the OV5640, the grayscale  $Y$  required for image processing is obtained by RGB-to-YCbCr conversion. The grayscale of the background frame is saved to DDR3 memory, and the grayscale data of multiple fruits placed against the background is obtained by computing the difference with the current frame. The difference grayscale data of the background frame are then sequentially filtered through erosion and expansion [14]. Sobel edge detection is also applied to the filtered image in the image preprocessing section to obtain the perimeter of the fruit [15]. After the frame scan localization, the position information of each fruit is utilized to add frames around each fruit. Simultaneously, the color of each localized fruit is identified using the HSV color model via the RGB-to-HSV algorithm.

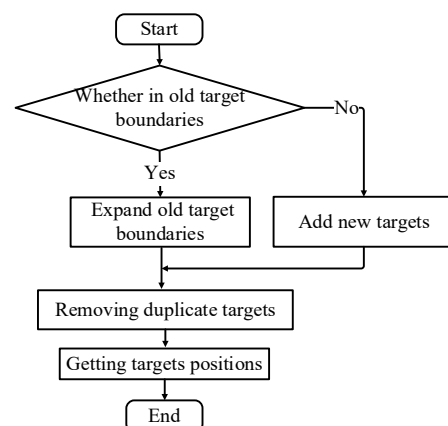


**Figure 2.** Image processing (Green parts stand for image processing parts, yellow parts stand for identification parts).

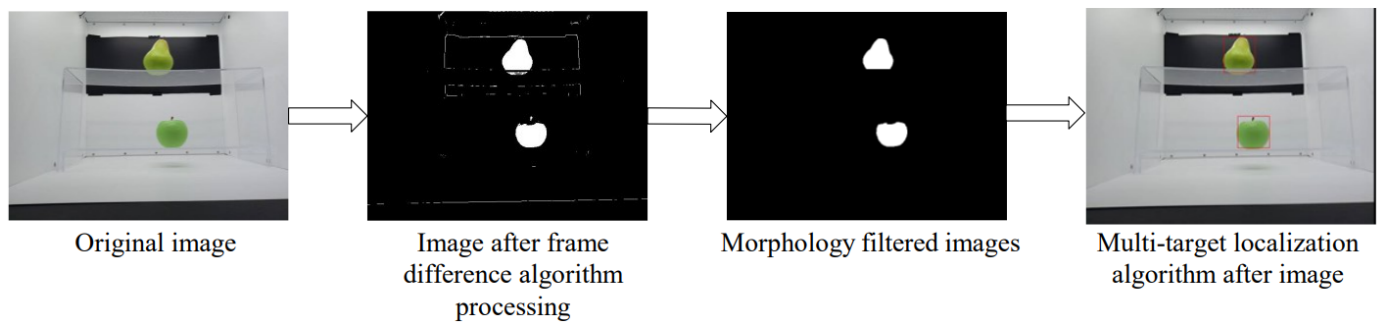
The boundary information, edge pixel points, and internal pixel points of each fruit obtained after image preprocessing are tallied to acquire shape features such as the long axis, short axis, perimeter, and area of each fruit. Further calculations yield second-order features like roundness, eccentricity, and body ratio. Ultimately, the acquired color features (Hue, Saturation, and Value) and shape features (Roundness, Eccentricity, and Body Ratio) are combined to ascertain the fruit type using a trained threshold classifier (as discussed in Section 3.2).

Following morphological filtering, target segmentation and quantity statistics are achieved by assessing the distances between each target. The fruits positioned on the detection platform are assessed, and the position coordinates of each fruit are real-time extracted based on the assessment outcome. The image is also binarized. Upon scanning the first white pixel, if the current target list is of black pixels, the pixel is automatically appraised during the subsequent white pixel scan. Upon detection, it first establishes whether the pixel pertains to an existing target. If not, it is identified as a new target; if it does, it is classified as an old target, and the target’s boundary is expanded according to the current pixel position information.

After observation, when the line field scan completes a frame, some specially shaped fruits, such as bananas, may appear and be identified as two overlapping targets. Therefore, each target needs to be compared to determine whether there is overlap. If overlap is detected, it is considered a duplicate target, and its boundary is merged to eliminate the duplication. The algorithm’s flowchart is depicted in Figure 3. The images of apples and pears are input to the image processing module according to the flow, and the verification of the image processing effect is displayed in Figure 4.



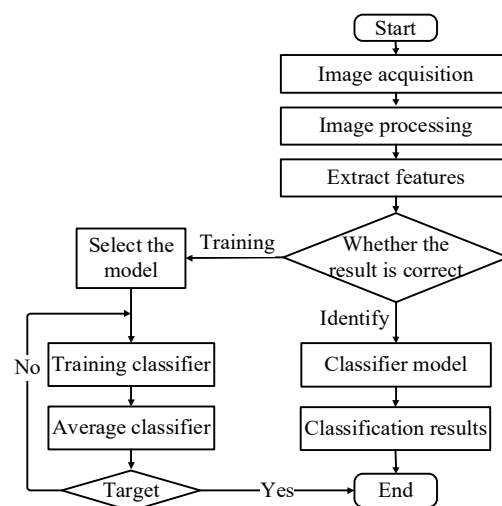
**Figure 3.** Multi-target detection and localization algorithms.



**Figure 4.** Effect of image processing.

### 3.2. Threshold Classifier

After preprocessing the captured fruit images, feature extraction is also required. Here, the color features and shape features of fruits are primarily extracted. The color features are obtained through the color-recognition module, allowing the system to perform an initial classification of fruits based on color. However, the core of fruit recognition lies in extracting the shape features of fruits. Figure 5 provides a comprehensive flowchart of fruit-type recognition.



**Figure 5.** Pattern recognition flowchart.

As evident from Figure 5, pattern recognition primarily encompasses two aspects, training and testing, with the training of images being particularly crucial. The training algorithm plays a pivotal role in handling the training data. The recognition-classification algorithm, which involves training and testing the extracted features, constitutes the essence of the fruit-recognition system. The thresholds employed for recognition are established through a threshold classifier's training. In this system, each threshold parameter is derived from multiple training sessions involving different fruits. The training outcomes are detailed in Section 4.1.

### 3.3. Robotic Arm Forward and Inverse Kinematic Analysis

In the context of robotics and kinematics, 'D-H' represents Denavit–Hartenberg. The Denavit–Hartenberg (D-H) method is a widely employed mathematical technique for illustrating the relationship between successive links in a robotic manipulator or robotic arm. It aids in defining the coordinate frames and kinematic parameters for each joint, facilitating efficient and systematic kinematic analysis. In this paper, a modified D-H coordinate system is utilized to depict the positional and angular relationship between two sets of jointed connecting rods, and a mathematical model along with a coordinate system



for the robotic arm is established. In the modified D-H coordinate system, the linkage length  $a_{i-1}$ , four parameters of linkage twist  $\alpha_{i-1}$ , linkage offset  $d_i$ , and joint rotation angle  $\theta_i$  are employed to depict the system, and the D-H parameter table is constructed based on the robotic arm model employed in the experiment.

After determining the D-H parameters and the homogeneous transformation matrix, a mathematical model of the kinematic equations of the robot arm was established. In the forward kinematic analysis, the orientation of the end effector relative to the coordinate system is obtained from the linkage parameters in Table 1, and the positional coordinate matrix of connecting rod  $i$  and connecting rod  $i - 1$  is shown in Equation (1).

$${}_{i-1}T_i(\theta_i) = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -\sin \alpha_{i-1} d_i \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & \cos \alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

**Table 1.** Mechanical arm modified D-H parameters.

Joint Number $i$	$a_{i-1}$	$\alpha_{i-1}$	$d_i$	$\theta_i$
1	0	0	$d_1$	$\theta_1$
2	0	$90^\circ$	0	$\theta_2$
3	$l_3$	0	0	$\theta_3$
4	$l_4$	0	0	$\theta_4$
5	$l_5$	0	0	$\theta_5$
6	0	0	0	$\theta_6$

In this table,  $l_3 = 105$  mm,  $l_4 = 90$  mm,  $l_5 = 170$  mm,  $d_1 = 100$  mm.

To achieve the grasping of the object, it is necessary to ensure that the robotic arm is in the desired position. The position of the end effector is determined by sequentially multiplying the position coordinate matrices of each linkage [16]. The desired position of the robotic end effector is shown in Equation (2). The  $n_x$ ,  $n_y$ , and  $n_z$  are the three-direction cosine of the end effector's  $x$ -axis of coordinates, with respect to the base-fixed coordinate system. The  $o_x$ ,  $o_y$ , and  $o_z$  are the three-direction cosine of the end effector's  $y$ -axis of coordinates, with respect to the base-fixed coordinate system. The  $a_x$ ,  $a_y$ , and  $a_z$  are the three-direction cosine of the end effector's  $z$ -axis of coordinates, with respect to the base-fixed coordinate system. The symbol 'p' indicates the end effector's coordinates, positioned relative to the base-fixed coordinate system. Additional details are provided in [16].

$${}^0T_6 = {}^0T_1(\theta_1){}_1T_2(\theta_2){}_2T_3(\theta_3){}_3T_4(\theta_4){}_4T_5(\theta_5){}_5T_6(\theta_6) = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

In inverse kinematics, the goal is to determine the angles of each joint of the robotic arm based on the known positional coordinates. In this paper, the analysis is focused solely on the main body of the robotic arm. The coordinates of the end point P are defined as ( $P_x$ ,  $P_y$ ) in the XOY plane, and these coordinates are obtained according to the D-H model, as shown in Equations (3) and (4).

$$P_x = l_3 \cos \theta_3 + l_4 \cos(\theta_3 + \theta_4) + l_5 \cos(\theta_3 + \theta_4 + \theta_5) \quad (3)$$

$$P_y = l_3 \sin \theta_3 + l_4 \sin(\theta_3 + \theta_4) + l_5 \sin(\theta_3 + \theta_4 + \theta_5) \quad (4)$$

In the inverse kinematics solution, the coordinates of the end point P are known, and  $l_3$ ,  $l_4$ , and  $l_5$  are the inherent parameters of the D-H model. Through calculations, Formula (5)

can be derived, and similarly, other angles can be obtained. These angles can be used to control servos and achieve coordinated position control.

$$\begin{aligned}\sin \theta_3 &= \frac{(-b \pm \sqrt{b^2 - 4ac})}{2a} \\ a &= m^2 + n^2 \\ b &= \frac{-n \cdot (l_3^2 - l_4^2 - m^2 - n^2)}{l_4} \\ c &= \left( \frac{l_3^2 - l_4^2 - m^2 - n^2}{2l_4} \right)^2 \\ m &= l_3 \cos(\theta_3 + \theta_4 + \theta_5) - P_x \\ n &= l_3 \sin(\theta_3 + \theta_4 + \theta_5) - P_y\end{aligned}\quad (5)$$

### 3.4. Robotic Arm Trajectory Planning

To maintain a stable state during the movement of the robotic arm and at the same time achieve efficient gripping, the trajectory of the robotic arm needs to be planned. This paper focuses on the variation of position, velocity, and acceleration of the robotic arm in the joint space, as well as the trajectory planning of the rotating joint over time. Trajectory planning involves the time series planning of velocity and acceleration during the movement of the robotic arm to ensure that the trajectory curve of the robotic arm at any moment is smooth and the speed is controllable, avoiding unnecessary sudden changes in position, velocity, and acceleration. At the same time, the motion trajectory of each joint of the robotic arm is planned so that the end effector can complete the task under the work requirements during the motion process. By controlling the roles of each joint at the same time, it ensures that the end effector performs the task by the optimal path. Reasonable motion trajectory planning helps to improve the accuracy of the end position of the robotic arm, thus ensuring the operational efficiency of the robot in high-precision motion. Then, a smooth path point position function is derived using the following two steps.

- Path planning to establish the geometric profile of the path, involving the determination of spatial curves such as space curves or other complex NURBS curves [17];
- Interpolation, where parameters like time or distance are utilized. Interpolation helps densify the parameters to obtain the intermediate points along the motion path.

In recent years, for the problem of free curve interpolation points in the field of robotic arm research, studies have been conducted on interpolation using the B-spline curve algorithm, which is relatively more advantageous than the polynomial interpolation algorithm, albeit due to the fact that the B-spline fitted curves do not necessarily pass through all the given interpolation points. In order to ensure that the B-spline curve passes through all control points, this paper employs an improved cubic B-spline interpolation algorithm for point-to-point trajectory planning [18]. This approach mitigates the impact of interpolation operations in both joint and Cartesian spaces, ultimately enhancing motion accuracy and quality. The expression for  $L_{3i+1}$  concerning time 't' is depicted in Equation (6), and the same principle applies to  $L_{3i+2}$ ,  $L_{3i+3}$ .

$$L_{3i+1} = \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}^T \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 3 & 0 \end{bmatrix} \begin{bmatrix} D_i \\ D_{i+1} \\ D_{i+2} \\ D_{i+3} \end{bmatrix}\quad (6)$$

In Equation (6),  $L_{3i+1}$  represents a segment of the trajectory curve of the cubic B-spline, while  $D_i$ ,  $D_{i+1}$ ,  $D_{i+2}$ , and  $D_{i+3}$  serve as the control points along the trajectory curve. The provided Equation (6) smoothly fits the given path, ensuring adherence to velocity, acceleration, and the duration of the movement. This enables the planning of an optimal path.

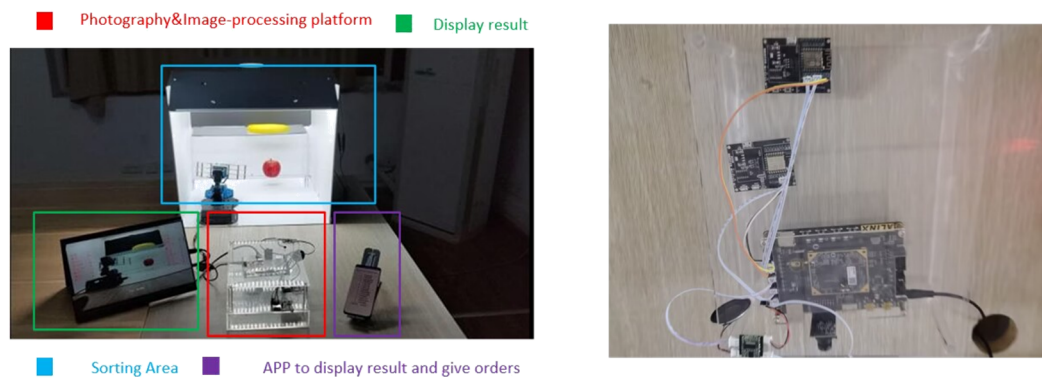


## 4. Experimental Results

Image processing and object recognition are pivotal technologies in this research. The morphological-recognition method and image segmentation techniques employed in this study can find application in image processing for the surrounding environment of vehicles, encompassing tasks like road, vehicle, and pedestrian detection. The system we have designed possesses the ability to recognize a wide array of objects. Through a straightforward replacement of these objects with road, vehicle, or pedestrian categories and the provision of suitable training, the system can be tailored for autonomous driving applications. Furthermore, by incorporating the enhanced motion trajectory planning algorithm to govern the actions of the robotic arm, it can be seamlessly integrated with path planning and vehicle control within autonomous driving systems, thereby enabling object-grasping and placing operations.

### 4.1. Experimental Setup and Data

Based on the above design plan, the physical system was fabricated and built, as shown in Figure 6 below, and testing of the system was started.



**Figure 6.** Overall system architecture (Left), and photography and image-processing platform (Right).

For this study, the color features and shape features of fruits are the primary focus. The color-recognition module takes in the color characteristics of fruits, enabling the identification of fruits with distinct colors such as kiwi fruits characterized by their brown hue, and facilitating an initial classification of fruits based on color. The color thresholds, derived from pertinent experiments, are detailed in Table 2.

**Table 2.** Fruit color H, S, V thresholds.

Fruit Type	Hue	Saturation	Value
Red Apple	0~15, 300~359	200~225	70~105
Green Apple	70~100	165~185	125~140
Banana	40~70	40~60	130~140
Yellow Mango	40~70	40~60	130~175
Green Mango	70~100	200~230	70~90
Green Pear	70~100	150~165	150~165
Yellow Pear	40~70	150~165	180~200
Orange	0~15, 300~359	220~240	150~170
Dragon Fruit	0~15, 300~359	170~195	80~100
Grape	0~15, 300~359	100~140	30~60
Kiwifruit	15~70	200~220	30~60
Fig	5~30	120~130	70~90
Mangosteen	0~15, 300~359	110~130	10~40

The shape characteristics of the fruit constitute a fundamental basis for fruit species identification. Various shape attributes contribute to this, including fruit size, fruit circumference, fruit area, fruit roundness, and fruit eccentricity. These shape features play a pivotal role in the identification process within the design.

Given that first-order shape features are susceptible to effects from rotation, scaling, and translation, it becomes imperative to extract shape features with RST (Rotation, Scaling, and Translation) invariance. Consequently, second-order shape features such as body ratio, roundness, and eccentricity are predominantly employed. The calculations for Equations (7)–(9) are provided below.

$$O = \frac{4\pi S}{L^2} \quad (7)$$

$$f = \frac{H}{L} \quad (8)$$

$$e = \frac{\sqrt{H^2 - W^2}}{H} \quad (9)$$

where  $O$  represents roundness,  $f$  denotes body ratio, which is the ratio of the long axis to the short axis,  $e$  stands for eccentricity,  $S$  signifies area,  $L$  represents perimeter,  $H$  corresponds to the long axis, and  $W$  represents the short axis. In the discrimination process, the measured roundness is normalized in the actual code by expanding the roundness value from 1 to 256. This normalization simplifies discrimination and enhances the accuracy of the threshold. Table 3 presents the data obtained for each feature's measurement.

**Table 3.** Shape feature thresholds.

Fruit Type	Roundness	Eccentricity	Body Ratio
Red Apple	0.74~0.80	0.34~0.51	1.01~1.13
Green Apple	0.73~0.79	0.35~0.52	1.00~1.13
Banana	0.01~0.09	0.96~0.99	5.00~6.00
Yellow Mango	0.31~0.37	0.89~0.95	2.50~3.20
Green Mango	0.39~0.43	0.84~0.94	1.90~2.10
Green Pear	0.69~0.77	0.32~0.42	0.89~0.94
Yellow Pear	0.67~0.76	0.33~0.42	0.90~0.92
Orange	0.71~0.86	0.12~0.21	0.98~1.26
Dragon Fruit	0.05~0.11	0.71~0.85	1.68~1.88
Grape	0.90~0.99	0.09~0.16	0.98~1.04
Kiwifruit	0.82~0.90	0.64~0.76	1.20~1.50
Fig	0.59~0.67	0.54~0.62	0.81~0.94
Mangosteen	0.49~0.58	0.43~0.52	0.84~0.92

#### 4.2. Experimental Results

Once the experimental setup and training data have been completed, the next step involves conducting a robot arm grasping experiment to test the system's robustness, calculate resource consumption, and measure recognition speed.

As the parameters of each joint of the robotic arm used in this paper serve as direct inputs and the system requires real-time capabilities, the trajectory planning method chosen is joint space. The LeArm six-degrees-of-freedom high-performance programmable robotic arm studied in this simulation is a six-axis structured rotary joint manufactured by Hiwonder. The arm structure comprises six digital servos with alloy grippers, an aluminum alloy stand, and an all-metal rotating base. The body is equipped with six high-precision digital servos, providing six degrees of freedom. The robotic arm's gripper structure is modeled using Solidworks software and produced using a 3D printer.

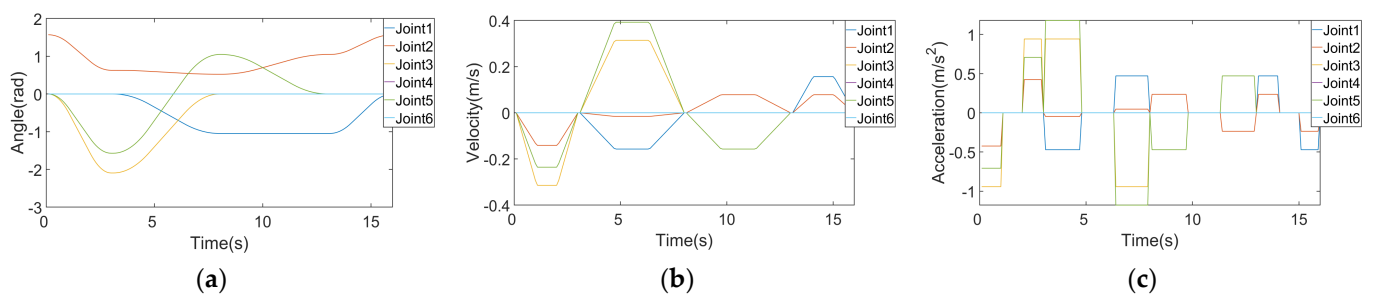
Based on the established mathematical model and D-H parameter table, this system conducts simulated object grasping experiments in the Matlab simulation environment. It adopts the improved cubic B-spline interpolation algorithm for trajectory planning

and selects four path points as experimental constraints. The constraint points and joint parameters are presented in Table 4.

**Table 4.** Constraint point and joint order list.

Constraint Point $P_{i-1}$	Joint 1 (Rad)	Joint 2 (Rad)	Joint 3 (Rad)	Joint 4 (Rad)	Joint 5 (Rad)	Joint 6 (Rad)
$P_0$	0	$\pi/2$	0	0	0	0
$P_1$	0	$\pi/5$	$-\pi \times 2/3$	0	$-\pi/2$	0
$P_2$	$-\pi/3$	$\pi/6$	0	0	$\pi/3$	0
$P_3$	$-\pi/3$	$\pi/3$	0	0	0	0

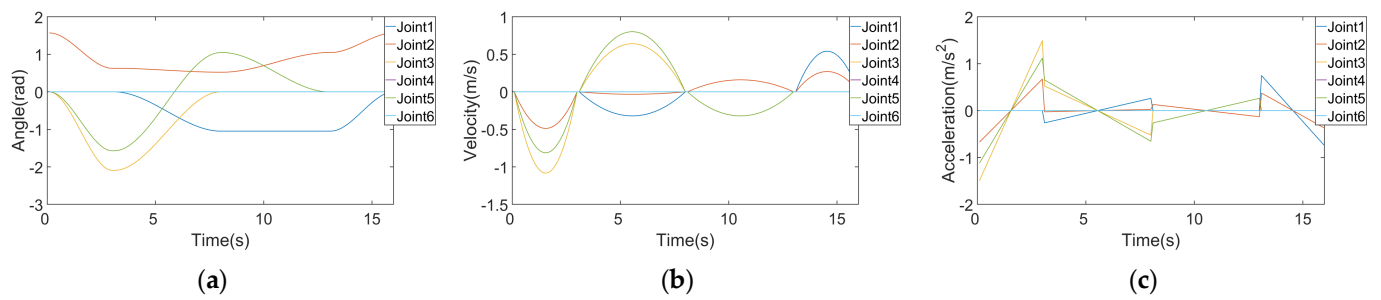
In this paper, the time of joint motion is set at each of the four constraint points.  $P_0$  represents the initial random position of the robotic arm,  $P_1$  signifies the position of the robotic arm after a reset,  $P_2$  corresponds to the position of the target object for the robotic arm, and  $P_3$  denotes the arrival position of the sorting robotic arm. These points collectively form the entire process of robotic arm grasping and sorting. The polynomial interpolation algorithm and the modified cubic B-spline interpolation algorithm are utilized to determine the pose of the robotic arm, encompassing displacement, velocity, and acceleration, as illustrated in Figure 7.



**Figure 7.** Polynomial interpolation algorithm motion simulation. (a) Displacement (b) velocity (c) acceleration.

Figures 7a and 8a display the positional changes of the robotic arm's joints during the gripping of an object using the two algorithms. These figures illustrate the trajectory trends of the robotic arm's joints over time. The two trajectories exhibit substantial consistency. Comparing Figures 7b and 8b, it becomes apparent that the cubic B-spline curve results in higher maximum velocity and acceleration. This adjustment is made to complete trajectory planning within the same timeframe. However, the improved cubic B-spline interpolation algorithm yields smoother curves, effectively reducing impact during the motion process. It addresses sudden parameter changes in the motion of each joint, enhancing stability and accuracy. Notably, an increase in the number of interpolation functions corresponds to a proportional rise in overall system computation. Thus, the improved cubic B-spline interpolation algorithm is a stable and efficient trajectory planning approach.

By recognizing and sorting different fruits multiple times at various locations within the recognition area, we achieve high accuracy in both recognition and sorting. The majority of fruits exhibit a high recognition and sorting accuracy, resulting in an impressive overall recognition success rate of 97.69% and a sorting success rate of 96.46%. These outcomes showcase the system's robustness, as depicted in Table 5.



**Figure 8.** Motion simulation of improved cubic B-spline interpolation algorithm. (a) Displacement (b) velocity (c) acceleration.

**Table 5.** Robustness test results.

Fruit Type	The Exact Number of Identification/Total Number	The Exact Number of Sorting/Total Number
Red Apple	99/100	100/100
Green Apple	100/100	100/100
Banana	99/100	95/100
Yellow Mango	98/100	98/100
Green Mango	97/100	100/100
Green Pear	98/100	98/100
Yellow Pear	93/100	95/100
Orange	100/100	100/100
Dragon Fruit	98/100	95/100
Grape	96/100	88/100
Kiwifruit	98/100	98/100
Fig	96/100	92/100
Mangosteen	97/100	96/100

As displayed in Table 5, both recognition accuracy and sorting accuracy remain consistently above 96%. In terms of the recognition component, it can be observed that higher accuracy is still achieved for items with brighter colors or more distinct shapes. For instance, the yellow pear may encounter recognition judgment errors due to its shape similarity to the green pear, despite the minor difference in their actual colors. Concerning the sorting aspect, superior grasping accuracy is observed for items closely resembling spheres or squares, while smaller objects like grapes and figs may incur grasping errors. This is attributed to the utilization of the traditional morphological-recognition approach, which significantly reduces system resources compared to neural network schemes.

Furthermore, the entire system is constructed on an FPGA development board utilizing fundamental components such as adders, multipliers, counters, registers, and Look-Up Tables (LUTs). Consequently, the system demonstrates substantially reduced resource utilization and power consumption. Although the CNN-based recognition system [19] and the YOLOv3-based recognition system [2] exhibit higher accuracy rates of approximately 99% and a mean Average Precision (mAP) of about 90.78%, respectively, compared to our method, the resource usage of our approach amounts to only about 10% of that employed in [2,19].

In addition, when considering recognition speed, the average recognition speed of the hardware-deployed CNN recognition system is approximately 900 ms, while the YOLOv3 neural network recognition system, deployed to hardware and optimized, achieves an average recognition speed of about 54.76 ms. In contrast, our system demonstrates an average recognition speed of approximately 25.26 ms. A comprehensive performance comparison is presented in Table 6. Compared to the systems introduced in [20,21], our system attains comparable or slightly higher accuracy while demanding fewer resources (FF, LUT). Furthermore, when contrasted with the system outlined in [22], our approach offers a substantial advantage in terms of recognition speed. Additionally, in comparison

with the system described in [23], our system achieves relatively higher accuracy without compromising recognition speed. Moreover, when evaluated against the system detailed in [24], our approach significantly reduces resource consumption while effectively fulfilling the intended functionality.

**Table 6.** Resource Usage, Speed, and Accuracy.

Equipment	FF (Flip-Flop)	LUT	Average Recognition Time/ms	Average Accuracy
Ours	13 K	19 K	25.26	97.69%
Yan et al. [19]	148 K	118 K	900	99.6%
Yin et al. [2]	154 K	71 K	54.76	90.78% (mAP)
Kojima et al. [20]	40 K	27 K	Not provided	96.33%
Kojima [21]	66 K	39 K	Not provided	More than 90% (mAP)
Wei et al. [22]	Not provided	Not provided	1582	Not provided
Hao et al. [23]	Not provided	Not provided	28.2	69.4% (mAP)
Takasaki et al. [24]	14 K	126 K	Not provided	Not provided

## 5. Conclusions

FPGAs offer significant application value in image processing due to their ability to execute real-time pipeline operations with the utmost efficiency. The proposed system in this paper leverages FPGA development boards to achieve enhanced recognition outcomes while optimizing resource utilization and effectively guiding the robotic arm to accomplish sorting duties. Moreover, when juxtaposed with recognition approaches like neural networks, the presented system boasts rapid recognition speed, minimal resource consumption, and exceptional portability. With its cost-effectiveness and adaptability across diverse scenarios, this system exhibits promising potential for small-scale production. Its implementation stands to substantially bolster production automation and markedly amplify sorting efficiency.

Given the attributes of low power consumption, high real-time performance, and minimal resource utilization inherent in this devised system, this study underscores its viability in addressing relatively straightforward sorting tasks across an array of application settings. By employing image recognition and trajectory planning, the system adeptly executes requisite operations while minimizing power consumption. A prime example of its utility lies in tasks like container sorting by port forklifts or material handling at construction sites.

**Author Contributions:** Conceptualization, Y.W., Y.H. and Z.W.; methodology, Y.W. and J.C.; software, Y.W., Y.H. and J.C.; validation, Y.W., Y.H., J.C. and Z.W.; formal analysis, Y.W. and J.C.; investigation, Y.W., J.C. and Y.Z.; resources, Y.W., J.C. and Z.W.; data curation, Y.W. and J.C.; writing—original draft preparation, Y.W., Y.H. and J.C.; writing—review and editing, Y.H. and Z.W.; visualization, Y.W., Y.H. and J.C.; supervision, Y.H.; project administration, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Research Project of Wuhan University of Technology Chongqing Research Institute (No. YF2021-06), the National Natural Science Foundation of China (Grant No. 61801341), and the 2022 Independent Innovation Research Fund of School of Information Engineering, Wuhan University of Technology (No. 2022-XX-B1-08).

**Data Availability Statement:** The simulation data used to support the findings of this study are available from the corresponding author upon request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Yang, Y.Z.; Wu, C.Y.; Hu, X.D. Study of Web-based integration of pneumatic manipulator and its vision positioning. *Zhejiang Univ. Sci. A* **2005**, *6*, 543–548.
2. Yin, H.; Hong, H.; Liu, J. FPGA-based Deep Learning Acceleration for Visual Grasping Control of Manipulator. In Proceedings of the 2021 IEEE International Conference on Real-Time Computing and Robotics (RCAR), Xining, China, 15–19 July 2021; pp. 881–886.

3. Yanagisawa, H.; Yamashita, T.; Watanabe, H. A study on object detection method from manga images using CNN. In Proceedings of the 2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, Thailand, 7–9 January 2018; pp. 1–4.
4. Zhu, L.; Lin, H.; Wu, W.J. Machine vision-based workpiece positioning for industrial robots. *J. Chin. Comput. Sci.* **2016**, *8*, 1873–1877.
5. Xu, W.F. Trajectory planning of vibration suppression for rigid-flexible hybrid manipulator based on PSO algorithm. *J. Control Decis.* **2014**, *29*, 632–638.
6. Lee, A.Y.; Jang, G.; Choi, Y. Infinitely differentiable and continuous trajectory planning for mobile robot control. In Proceedings of the 2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, Republic of Korea, 31 October–2 November 2013; pp. 357–361.
7. Wu, X.; Tan, L.; Xu, X. Three-dimensional recognition method of fruit target based on PointNet++. In Proceedings of the 2022 5th World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM), Ma’anshan, China, 18–20 November 2022; pp. 496–500.
8. Tang, Y.; Zhang, Y.; Zhu, Y. A Research on the Fruit Recognition Algorithm Based on the Multi-Feature Fusion. In Proceedings of the 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Harbin, China, 25–27 December 2020; pp. 1865–1869.
9. Wang, X.; Huang, W.; Jin, C.; Hu, M.; Ren, F. Fruit recognition based on multi-feature and multi-decision. In Proceedings of the 2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems, Shenzhen, China, 27–29 November 2014; pp. 113–117.
10. Latha, R.S.; Sreekanth, G.R.; Rajadevi, R.; Nivetha, S.K.; Kuma, K.A.; Akash, V.; Bhuvanesh, S.; Anbarasu, P. Fruits and Vegetables Recognition using YOLO. In Proceedings of the 2022 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 25–27 January 2022; pp. 1–6.
11. Chowdhury, M.F.I.; Jeannerod, C.P.; Neiger, V.; Schost, É.; Villard, G. Faster Algorithms for Multivariate Interpolation with Multiplicities and Simultaneous Polynomial Approximations. *IEEE Trans. Inf. Theory* **2015**, *61*, 2370–2387. [[CrossRef](#)]
12. Kyung-Jo, P.; Park, Y.S. Fourier-based optimal design of a flexible manipulator path to reduce residual vibration of the endpoint. *J. Robot.* **1993**, *11*, 263–272.
13. Cao, S.; He, X.; Zhang, R.; Xiao, J.; Shi, G. Study on the reverse design of screw rotor profiles based on a B-spline curve. *J. Adv. Mech. Engineering.* **2019**, *11*, 10. [[CrossRef](#)]
14. Qin, Q.; Sun, L.; Feng, Y. Real-Time Image Filtering and Edge Detection Method Based on FPGA. In Proceedings of the 2022 IEEE 5th International Conference on Electronics Technology (ICET), Chengdu, China, 13–16 May 2022; pp. 1168–1173.
15. Kim, J.R.; Jeon, J.W. Sobel Edge based Image Template Matching in FPGA. In Proceedings of the 2020 IEEE International Conference on Consumer Electronics Asia (ICCE-Asia), Seoul, Republic of Korea, 1–3 November 2020; pp. 1–2.
16. Cho, B.; Lee, C.J. Time-optimal trajectory planning for a robot system under torque and impulse constraints. In Proceedings of the 30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004, Busan, Republic of Korea, 2–6 November 2004; Volume 2, pp. 1058–1063.
17. Ma, H.; Wang, Y.; Yang, S.; Ming, L. Interpolation method of NURBS and its offset curve. *J. Mech. Sci. Technol.* **2022**, *41*, 433–438.
18. Li, X.; Gao, X.; Zhang, W.; Hao, L. Smooth and collision-free trajectory generation in cluttered environments using cubic B-spline form. *J. Mech. Mach. Theory* **2022**, *169*, 104606. [[CrossRef](#)]
19. Yan, P.; Xiang, Z. Acceleration and optimization of artificial intelligence CNN image recognition based on FPGA. In Proceedings of the 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 4–6 March 2022; pp. 1946–1950.
20. Kojima, A.; Nose, Y. Development of an Autonomous Driving Robot Car Using FPGA. In Proceedings of the 2018 International Conference on Field-Programmable Technology (FPT), Naha, Japan, 10–14 December 2018; pp. 411–414.
21. Kojima, A. Autonomous Driving System implemented on Robot Car using SoC FPGA. In Proceedings of the 2021 International Conference on Field-Programmable Technology (ICFPT), Auckland, New Zealand, 6–10 December 2021; pp. 1–4.
22. Wei, K.; Honda, K.; Amano, H. FPGA Design for Autonomous Vehicle Driving Using Binarized Neural Networks. In Proceedings of the 2018 International Conference on Field-Programmable Technology (FPT), Naha, Japan, 10–14 December 2018; pp. 425–428.
23. Hao, C.; Sarwari, A.; Jin, Z.; Abu-Haimed, H.; Sew, D.; Li, Y.; Liu, X.; Wu, B.; Fu, D.; Gu, J.; et al. A Hybrid GPU + FPGA System Design for Autonomous Driving Cars. In Proceedings of the 2019 IEEE International Workshop on Signal Processing Systems (SiPS), Nanjing, China, 20–23 October 2019; pp. 121–126.
24. Takasaki, K.; Hisafuru, K.; Negishi, R.; Yamashita, K.; Fukada, K.; Wakaizumi, T.; Togawa, N. An autonomous driving system utilizing image processing accelerated by FPGA. In Proceedings of the 2021 International Conference on Field-Programmable Technology (ICFPT), Auckland, New Zealand, 6–10 December 2021; pp. 1–4.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.