Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/170615

Conveyor Belt Object Identification: Mathematical, Algorithmic, and Software Support

Vladyslav V. Yevsieiev[1], *Igor S. Nevliudov*[1], *Svitlana S. Maksymova*[1,*], *Murad A. O. Omarov*[1], *and Oleksandr M. Klymenko*[2]

[1]Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine
[2]«Kapelou» LLC, Kyiv, Ukraine

Received: 12 Oct, 2023, Revised: 15 Oct, 2023, Accepted: 16 Oct, 2023.
Published online: 1 Nov. 2023

Abstract: This article is devoted to the development of a package identification system on a mixed conveyor sorting line with a vertical lift from Kapelou. In the conditions of modern Warehouse 4.0 systems requirements, a tasks number arise associated with automatic objects sorting in real time. One of the most common is QR codes using, as it is the most cost-effective. But the introduction of automated systems for identifying objects on a conveyor line based on QR codes causes a number of tasks that are associated with the dynamic parameters of identifying the location of the package in the recognition zone, determining and localizing the location of the QR code, for further reading and decoding information about the package with further adoption decisions about its movement in the sorting system. The authors propose a solution to this problem by developing a module for identifying and recognizing objects on a conveyor line based on computer vision using a Raspberry Pi 4 Model B single-board computer with a developed method for processing a QR code image, the system structure, algorithmic and mathematical support have been developed. To check the correctness of proposed decisions, software was developed, and a number of natural experiments were carried out with different parameters (conveyor speeds, illumination, packaging feeding angles for the computer vision system) for the developed object identification system on the Kapelou sorting conveyor line, which showed a high processing speed and decoding data from a QR code.

Keywords: Conveyor Belt, Object Indentification, Object Recognition, Raspberry Pi , QR-code.

1 Introduction

Modern industry is characterized by the use of new technologies in production processes. Industry 4.0 technology consists in the full automation of production and the enterprise while managing production processes in real time, taking into account all the conditions that can affect the processes. To achieve these goals, there is a need for full or partial automation of the warehouse. This is achieved due to modern technologies, which in the complex transform an ordinary warehouse into a smart one - Warehouse 4.0 [1]. The creation of modern Warehouse 4.0 systems should solve the following tasks [2]:

- Efficient Communication Scheduling;
- Accurate and Robust Localization;
- Multi-Robot Collaboration, robots consist of various sensors, precise actuators and powerful processors;
- Human Activity Recognition.

Conveyor lines are a very common automation device for moving objects in enterprises. Conveyors are used in almost every enterprise and are divided into [3]:

roller - the surface of this type of conveyor belt consists of rollers that are selected according to production requirements, such as the weight or the required products movement speed that will move along the belt;

flat - uses a series of drive pulleys to move a continuous flat belt, which can be made of natural material or synthetic fabric (e.g. polyester, nylon);

chevron - belt conveyors with a wedge are designed with vertical wedges or barriers. These spikes can keep bulk materials safe on slopes, keep items evenly spaced, and more.

In order to improve work, reduce time spent on processes and improve the quality of identification and objects sorting in production, conveyor lines are additionally equipped with different types of identification systems, depending on the objects that need to be identified. In today's world automatic identification and data capturing technology (Automatic Identification and Data Capturing – AIDC) exists to perform such tasks [4]. This term refers to methods of automatic identification of objects, collection of data about them and their transfer directly to computer systems without human intervention. Identification data is obtained externally during image processing or

[*]Corresponding author e-mail: svitlana.milyutina@nure.ua

electromagnetic waves from readers. The converter converts the received information, which is further processed by the computer by comparing the received information with already existing information, or independently identifies the object.

Identification of objects can be carried out on the basis of the use of various modern comparison methods, which are presented in Table 1.

Table 1: Comparison of Object Identification Methods on the Conveyor Line

Parameters	Identification Method		
	Computer Vision System (QR-codes)[5]	Marks Radio Frequency Identification (RFID) [6]	Near Field Communication (NFC) [7]
1	2	3	4
Contactless	+	+	+
Restrictions in the operation line	absent	absent	absent
Identification distance	~ 30-40 cm	~ 20 cm	no more than 10 cm
Ability to read a mark from a moving object	+	+	+
Simultaneous reading of several marks	+	+ (in the presence of an anti-collision function)	–
Speed and accuracy of identification	+	+	+
Ability to retrieve data from a damaged or partially damaged mark	+	–	–
Possibility of storing additional information on the mark and rewriting it	–	+	+/–
Price	Thermal printer Xprinter XP-420B printer of labels, stickers and barcodes 108 mm ~90-120$ [8]	Thermal transfer polypropylene RFID labels (1pc)~1-1.20$.	NFC tag, sticker, sticker NTAG213 ISO 14443A 144 bytes 13.56 MHz (1pc) ~1-1.20$.
	Sticker barcodes (1 pc.) – 3 cent.	Polyester RFID label with digital printing in CMYK ~ 1.20-1.60$	Set of NFC tags Broadlink NFC Tag SRN1 (1pc) ~ 1.34-1.70$.

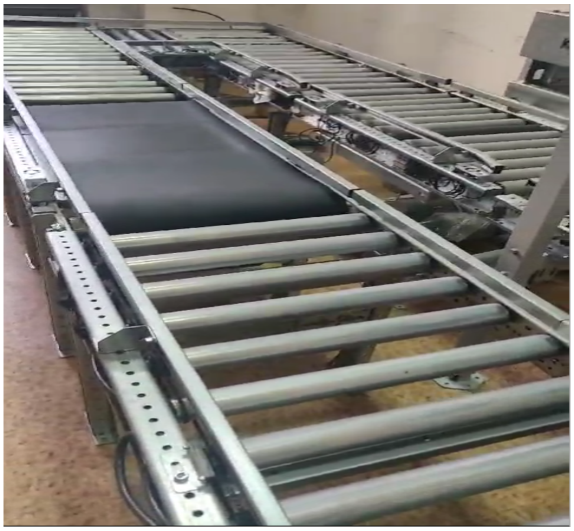
As can be seen from Table 1, both a barcode system and RFID and NFC tags can be used to implement object identification tasks on the conveyor line, but it is worth noting that the use of RFID/NFC tags will cost much more per unit product due to the price of the RFID tag from $1–1.60 (NFC $1–1.70) for 1 piece. Even if the RFID/NFC tags are damaged, it will be impossible to renew the information, unlike the use of barcodes, which makes it possible to partially restore fragments of information, which will allow the object to be identified, instead of sending it to the operator for visual recognition. It is also worth noting that when using RFID/NFC tags on a conveyor line to sort packages, it is necessary to maintain the distance between objects, or reduce the distance for identification to a minimum, which requires adjusting the height of the scanner depending on the size of the package each time.

Since the automatic sorting system sorts of products according to the corresponding QR code, fast and accurate QR code classification is the key to the supply system. How to quickly and accurately determine the QR code depends on the efficiency of the sorting system, as well as on the capacity and scale of the warehouse. However, the difficult and changing lighting environment often results in uneven illumination of the QR code. This uneven illumination affects the binarization and makes it difficult or inaccurate to quickly and accurately identify the QR code. Thus, it increases unnecessary time spent on warehouse management systems.

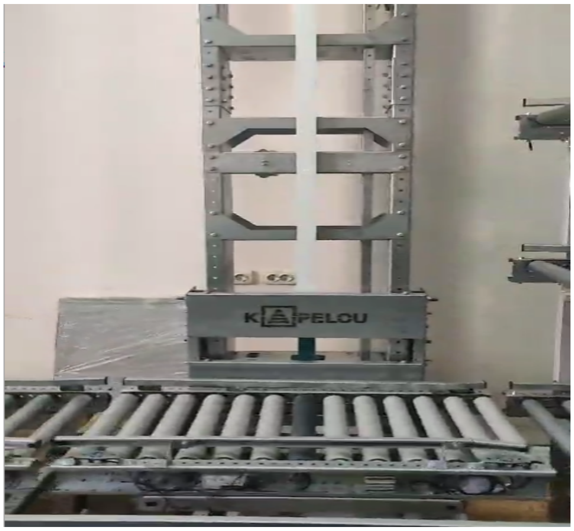
In recent years, many attempts have been made to solve this complex problem. Yi-Juan Di et al. developed an improved local Wellner's algorithm for identifying a QR code label on a package in an autosort system [9]. The principle of operation of the algorithm is simple, while it will spend more time on processing, and the effect of processing unevenly lit QR code images in difficult lighting conditions is far from the best. Yao, S. et al presented an algorithm that combines Niblack's improved algorithm with Otsu's algorithm [10]. The algorithm is faster but does not have good adaptive ability because it cannot adapt to the different illumination unevenness of the QR code images. Duan, Z. and others. proposed a method suitable for non-uniform illumination of barcodes [11], but the algorithm is more complex and does not cope well with non-uniform illumination boundaries. Wu, Y. et al. came up with a method that divides the QR code into several small blocks using block truncation coding (BTC) [12], but the steps of the algorithm are long, and the results are sensitive to the block window size. Yang, L et al presented an improved local thresholding method based on Bernsen's method [13], but the parameters need to be determined manually. In the work of Rongjun Chen et al. proposed a QR code scanning method based on the adaptive binarization approach, based on the features of the QR code symbol [14]. This solution allows to scan and obtain information from the QR code in uneven lighting, in the static mode of receiving the QR code frame, that is, when the package does not move along the conveyor, which allows to fully evaluate the effectiveness of the proposed method. As a result, it is possible to conclude that the study of automated object sorting systems based on real conveyor lines using computer vision is relevant.

2 Software and Hardware Modules Interaction Structural Block Diagram Development.

Based on this, within the framework of these studies, for the Kapelou industrial packaging sorting conveyor line [15] with a vertical elevator, which is presented in Figure 1, the authors propose to develop an identification control system based on the use of QR codes. This will allow sorting objects to use already existing codes on packages. As a part of the executive device, there will be a vertical elevator for moving packages between floors.



a) a fragment of an industrial roller conveyor with a belt zone



b) vertical industrial elevator for moving packages between floors

Fig. 1: Industrial Conveyor Line for Sorting Packages with a Vertical Elevator by Kapelou

The general scheme of an industrial conveyor line for sorting packages with a vertical elevator of the Kapelou company is presented in Figure 2. We will indicate the following main elements of the conveyor line on the diagram, 1 – a vertical elevator for moving packages between floors (Fig. 1b) will act as an executive mechanism for sorting identified packages on floors of transportation; 2 – rollers, as an element of the roller conveyor, serves to move the package in space; 3 – the zone of the conveyor with a belt element - a table, in this zone, the presence and identification of the package will be recognized with the help of a computer vision system; 4 – vertical conveyor on 3 floors, for sorted packages; 5 – packaging on the conveyor line. It is also necessary to consider the main events that affect the packaging, for the convenience of visualization we will present them in the form of red arrows. 6 – the arrival of an unidentified object on the conveyor line from the accumulator; 7 – identification of the object using the computer vision system; 8 – movement of the identified object along the conveyor; 9 – sorting of the identified object using a vertical elevator by levels; 10, 11 – movement of the identified object to the specified storage.

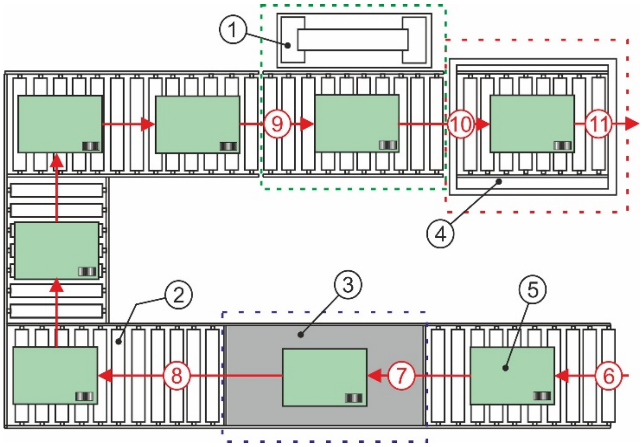
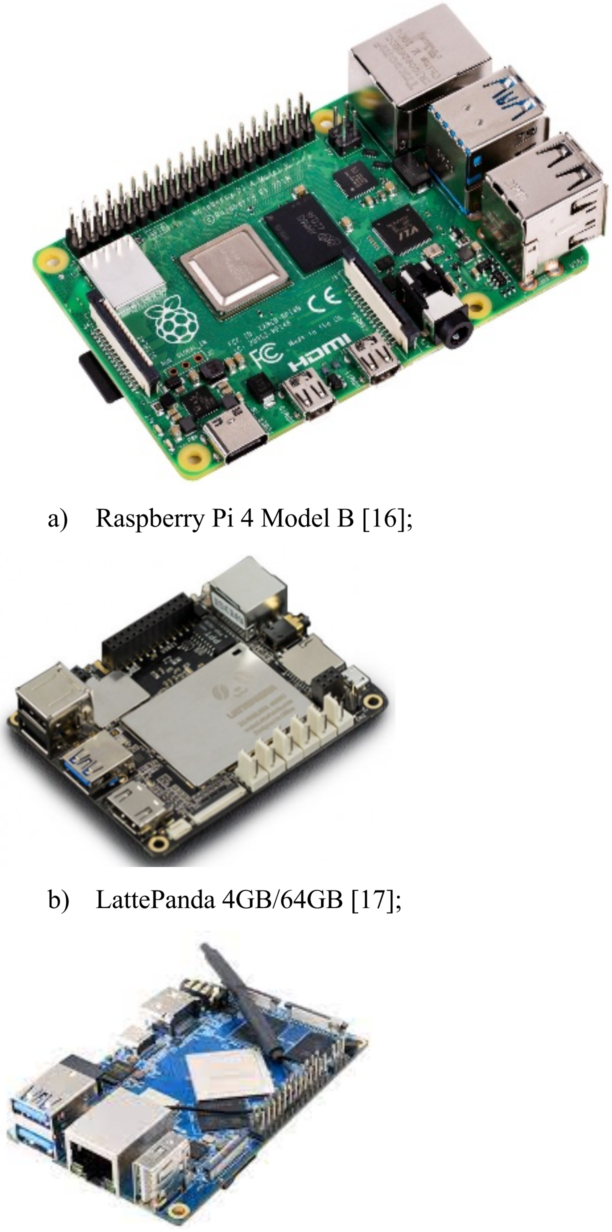


Fig. 2: Scheme of an Industrial Conveyor Line for Sorting Packages with a Vertical Elevator by Kapelou

Based on the technical features of the Kapelou industrial packaging sorting line with a vertical elevator, it is possible to start the selection of hardware modules for the implementation of a computer vision system for object identification, to simplify the selection of modules, we introduce some restrictions:

- small overall dimensions;
- support for wireless data transmission technologies (Wi-Fi, Bluetooth);
- a general-purpose input/output interface for connecting and controlling executive devices (vertical lift);
- support for Web-camera connection.

Based on the above limitations, the following single-board computers are suitable for the implementation of a computer vision system for the identification of objects on a conveyor line, which are presented in Figure 3, and a comparison of their technical characteristics is given in Table 2.



a) Raspberry Pi 4 Model B [16];

b) LattePanda 4GB/64GB [17];

c) Orange Pi 4 4Гб DDR4/16Гб EMMC [18].

Fig. 3: Single-Board Computers for the Object Identification Implementation

Table 2: Technical Characteristics of Single-Board Computers for the Object Identification System Implementation

Parameters	Single-Board Computers		
	Raspberry Pi 4 Model B [16]	LattePanda 4GB/64GB [17]	Orange Pi 4 4Гб DDR4/16Гб EMMC [18]
CPU	BROADCOM BCM2711	Intel Cherry Trail Z8350 Quad Core	Rockchip RK3399
Number of processor cores	4	4	4
CPU frequency	1.5 GHz	1.8GHz	2 GHz
RAM	8 GB	4 GB	4 GB
Graphics controller	VideoCore VI c OpenGL ES 1.1, 2.0, 3.0	Intel HD Graphics	Mali-T864 GPU with OpenGL support
Networking	Ethernet 10/100/1000 WiFi 2.4G/5.8G 300 mb/s	Ethernet 10/100/1000 WiFi 2.4G/5.8G 300 mb/s	Ethernet 10/100/1000 WiFi 2.4G/5.8G 300 mb/s
Interfaces	4 USB	3 USB	2 USB
Power supply	5V 3A	5V 2A	5V 3A
General-purpose input/output (GPIO)	+	For chip Intel X-Z8300 and for ATmega32u4	+
Storage eMMC	microSD	Built-in memory 64GB + microSD	Built-in memory 16GB + microSD
Operating System	Raspberry Pi OS, Debian Linux	Windows 10, 64-bit	Raspberry Pi OS, Debian Linux
Dimensions (mm)	88*58	80*70	91*55.7
Price	~120$	~ 150 – 219$	~ 110$

As can be seen from Table 2, all single-board computers have similar parameters and price range, however, it is worth noting some features, LattePanda 4GB / 64GB is based on the Intel Cherry Trail Z8350 Quad Core processor, which is essentially a microprocessor system, unlike the Orange Pi 4 4GB DDR4/16GB and Raspberry Pi 4 Model B. As a result, the principle of working with the GPIO bus becomes more complicated on LattePanda 4GB/64GB when developing control systems in Python, and, as a result, based on the fact that Raspberry Pi 4 Model B has twicely more memory capacity than Orange Pi 4 4GB DDR4/16GB, then a hardware module to develop a computer vision system for identifying objects on a conveyor line for sorting will be developed based on

Raspberry Pi 4 Model B. (Attar et al., 2022) To obtain a video image within the framework of these studies, it is proposed to use Web-cam Logitech C310 HD (model 960-001065) [20] or Raspberry Pi Cam v1.3 [21].

The next step was to develop a block diagram of the interaction of software and hardware modules based on Raspberry Pi 4 Model B, which is shown in Figure 4.

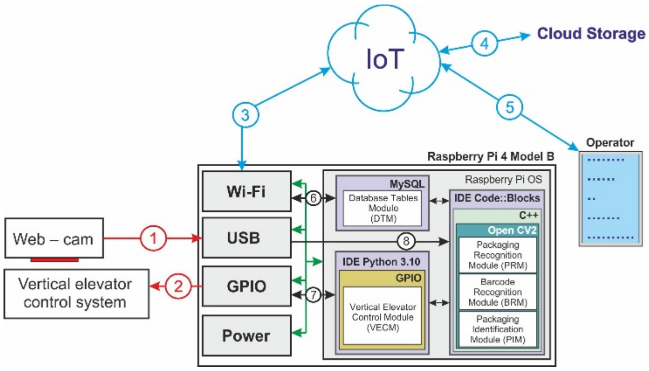

IoT
Cloud Storage
3
4
5
Operator
Raspberry Pi 4 Model B
Wi-Fi
USB
GPIO
Power
Web – cam
Vertical elevator control system
1
2
6
7
8
MySQL
Database Tables Module (DTM)
Raspberry Pi OS
IDE Code::Blocks
C++
Open CV2
Packaging Recognition Module (PRM)
Barcode Recognition Module (BRM)
Packaging Identification Module (PIM)
IDE Python 3.10
GPIO
Vertical Elevator Control Module (VECM)

Fig. 4: Structural Diagram of Interaction between Software and Hardware Modules

Let us describe the purposes of the block diagram main elements of the interaction between software and hardware modules in the developed system for identifying objects on an industrial conveyor line:

Web-cam - a camera that is installed directly above the conveyor area with a belt table, either the built-in Raspberry Pi Cam v1.3 camera or the remote Web-cam Logitech C310 HD camera (model 960-001065) can act as a camera, which is connected via a USB connector;

Vertical Elevator Control System - vertical elevator control system will be controlled by applying a control signal through the GPIO interface with Raspberry Pi 4 Model B;

Power - power supply module (power supply) of the single-board computer Raspberry Pi 4 Model B (DC 5V, 3A);

Wi-Fi - a hardware module that is integrated into the Raspberry Pi 4 Model B acts in this case for transmitting / receiving data from the Raspberry Pi 4 Model B from external storage devices (Cloud Storage), as well as for interacting with the operator using IoT technologies;

Raspberry Pi OS is a graphical operating system for controlling the Raspberry Pi 4 Model B, built on the basis of Debian, the Linux OS family [22];

IDE Code: Blocks is a cross-platform software development environment based on the wxWidgets library;

C++ is a high-level object-oriented programming language that acts as a language for developing an object identification system on an industrial conveyor line;

Open CV2 is an open-source computer vision library for image processing and general purpose numerical algorithms;

Packaging Recognition Module (PRM) - a module for recognizing the presence of packaging, on a conveyor line will allow to detect the contour of the package, to highlight its boundaries and identify its presence;

Barcode Recognition Module (BRM) - on the received image of the package localizes the location of the barcode;

Packaging Identification Module (PIM) - reads the data received from the barcode and identifies the object on the conveyor line;

IDE Python 3.10 - software development environment for the high-level object-oriented programming language Python;

GPIO - a special library (PRi.GPIO), which makes it possible to control external (vertical elevator control system) devices by writing control commands in Python;

Vertical Elevator Control Module is a program for transmitting command signals to a vertical elevator control system via the GPIO interface, which allows sorting objects by 3 levels.

MySQL is a relational database management system that stores information about objects of identification, specification and necessary information.

Database Tables Module - tables with data about sorting objects and necessary information for warehousing.

A general view of the computer vision system layout, an industrial conveyor line for sorting packages with a Kapelou vertical lift in the case is shown in Figure 5.

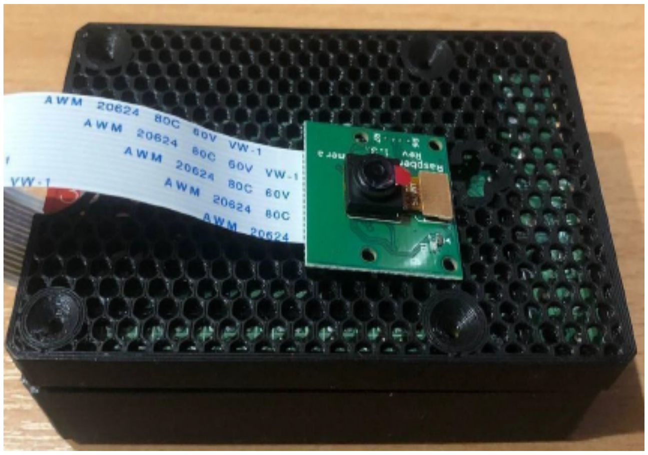
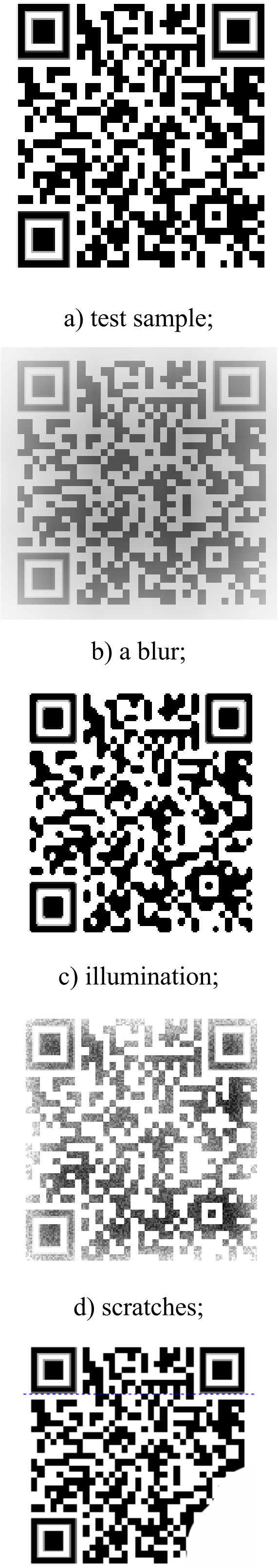


Fig. 5: Computer Vision System Layout

3 Received QR Code Image Frame Normalization Method Improvement

Considering that the developed system will be used for sorting packages within the conveyor line, there is a high possibility that the resulting image of QR codes may be damaged during transportation. An example of damage to QR codes is presented in Figure 6.



a) test sample;

b) a blur;

c) illumination;

d) scratches;

e) physical injury

Fig. 6: Example of QR Codes Image Frames on Conveyor Lines for Sorting Packages

All these damages have a bad effect on the identification time, and in many cases, if the system cannot identify the package, it falls into the sump, and then is manually recognized and sorted, as a result, the authors proposed to use the following sequence: analysis of the input QR-code image frame based on dynamic models, then a contrast-enhancing filter (to correct "blur", "illumination", "scratches" (Fig. 6b,c,d)) and improve the frame filtering method.

As part of this study, we will use two-dimensional discrete linear and non-linear models to analyze the image of the QR-code frame, which make it possible to describe the dynamics of the surface of the received image in space. As a result, the linear model can be described by the following system of equations:

$$\begin{cases} x(t+1) = Ax(t) + Bv(t), v(t) \approx N(0,Q); x(0) = x_0 \\ y(t) = Cx(t) + w(t), w(t) \approx N(0,R) \end{cases}, \quad (1)$$

where $y(t)$ - image vector, $x(t)$ - state vector of the dynamic model, A - matrix-operator of the model, B - moving average vector matrix, C - functional transformation matrix when measuring the values of elements of the QR-code image frame, $v(t), w(t)$ - vector of Gaussian centered noise with correlation functions Q and R, x_0 - vector of initial or boundary values.

Matrix-operator (A) (1) will have next structure:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_M & -a_{M-1} & -a_1 \end{bmatrix}, \quad (2)$$

where M - model order, a_i - model operator parameters.

Looking at the fact that the research is carried out with the QR code image frame, we will simplify the characteristics of the noise vectors, make the following assumption 3:

$$Q(t) \ = \ R(t) = \sigma^2 \delta(t), \quad (3)$$

where σ^2 - variance of Gaussian noise, $\delta(t)$ - delta function.

As a result, (1) can be represented as an autoregression with ARMA (AutoRegression and Moving Average) [23], then the model will take the following form:

$$u_{i,j} = -\sum_{m,n=0(m,n\neq 0)}^{P,Q} a_{m,n} u_{i-m,j-n} - \sum_{m,n=0}^{L,M} b_{m,n} w_{i-m,j-n}, \quad (4)$$

where $u_{i,j}$ - counts of the signal of the frame image QR-code, $a_{m,n}$ - linear prediction coefficients (LP), $w_{i,j}$ - white noise counts, P,Q - parameters specifying the order by coordinates (OX,OY), $b_{m,n}.L,M$ - coefficients and order of the moving average component.

As you can see, the resulting model 4 will consist of an AR (Autoregression) model, which makes it possible to describe the dynamic properties of the signal, and the classification uses the autoregression component and MA (moving average) errors as an aid to characterize the AR model. We assume that a frame is used to identify the packaging, which is converted from RGB to Crayscale in the process. This makes it possible to simplify model 4 to the following form:

$$u_{i,j} = -\sum_{m,n=0(m,n\neq 0)}^{P,Q} a_{m,n}u_{i-m,j-n} + \varepsilon_{i,j} \qquad (5)$$

where $\varepsilon_{i,j}$ - error counts with variance σ^2.

Based on the obtained model (5), the LP coefficient can be found using the method of least squares, which allows to adjust the sensitivity of the obtained model to minor fluctuations (Fig. 6b-d). This is achieved due to the invariance of the model parameters to the displacement of the frame base fragment of the QR-code relative to the coordinates in which the frame is set. In order for model (5) to be invariant to the shift of coordinates and at the same time to be adequate, it should perform the functions of approximation and extrapolation of the image signal data of the received QR-code frame. In cases where the obtained model will be inadequate, it is proposed to use a discrete nonlinear model in the form of a Voltaire series within the framework of these studies [24].

The criterion for the adequacy of the model can be the ratio of the signal strength of the QR code image frame to the error value and can be recognized through the Signal to Noise Ratio (SNR) [25]

$$SNR = 20\lg\frac{P(u)}{\sigma_\tau^2}. \qquad (6)$$

where $P(u)$ - signal strength of the basic fragment of the QR-code image frame, σ_τ^2 - error noise variance.

The next step for processing the received QR code frame is to use a median filter (MF). This makes it possible to sequentially process each point of the received frame, as a result of which a sequence of estimates is formed. Median filtering uses a filter aperture that usually has central symmetry, with the center located at the current filtering point [26]. The authors suggest in this case using a MF having a square aperture with a side equal to 3. If the MF reduces the local contrast of the QR code image frame, it is suggested to use a contrast-enhancing filter. The core of the contrast-enhancing filter has a value greater than 1 at the point (0,0) with the total sum of all values equal to 1.

Examples of a contrast-enhancing filter with a kernel given by the matrix M_n^{contr} are presented in 7.

$$M_1^{contr} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix};$$
$$M_2^{contr} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix} \qquad (7)$$

As can be seen from 7, the effect of the contrast increasing is achieved due to the fact that the filter emphasizes the difference between the intensities of neighbouring pixels, separating these intensities from each other.

As the last step of working with the received QR-code image frame, it is proposed to clean the frame image from various types of noise based on the improved Susan image filtering method [27,28]. Usually, when using the Susan method, a group of pixels of 5*5 elements is considered, the central pixel of this matrix is the one to be analyzed. The authors propose, within the framework of these studies, to reduce to 3*3 for the automated method of processing QR-code frames. Using a 3*3 pixel matrix somewhat slows down the system, but shows more accurate results needed to read and decode information from a QR-code.

During the check, the deviation of the pixel brightness from the average brightness is calculated. If the filter "decides" that such a pixel should not exist, its "loud" value is replaced by a new one calculated on the basis of surrounding pixels. The criterion for determining noise in this method is to consider n pixels that are included in the pixel matrix.

The sum of deviations of pixel brightness from the average value can be calculated as:

$$S = \sum_{i=1}^{n-1} \Delta_i \qquad (8)$$

where $\Delta_i = \left|\bar{b} - b_i\right|$; b_i - i-pixel value; $\bar{b}$ - the average value of the brightness, which is calculated by the expression:

$$\bar{b} = \sum_{i=0}^{n-1} \frac{b_i}{n} \qquad (9)$$

The relative contribution of the deviation Δ_k of the pixel to be analyzed to the value s is determined:

$$P_k = \frac{\Delta k}{S} \qquad (10)$$

where k - the number of the pixel to be analyzed.

It is likely that $\sum_{i=0}^{n-1} p_i = 1$.

If a uniform distribution of pixel brightness is observed in the considered image frame format, then the value p_i will not be significantly different from $1/n$. The brightness of a noisy pixel is significantly different from the average brightness of its surrounding pixels. The value Δk of such a pixel is greater than that of other pixels, which means that P_k will exceed $1/n$. This is the criterion for a noisy pixel, if $P_k > 1/n$, then the pixel k is noisy.

After detecting a noisy pixel, you need to decide what to do with it, there are two options:

- replace the noisy pixel with the average value $\overline{b}$;

- replace the noisy pixel with average values, taking into account the values of all pixels except the noisy one.

In this case, the authors propose to replace the noisy pixel with average values, calculated taking into account the values of all pixels that do not satisfy the noise selection criterion.

Within the framework of these studies, the following sequence of image processing of the QR-code frame is recommended:

1. using the basic fragment of the image determine the parameters of the model and the parameters characterizing the quality of the model (statistical parameters of error noise);
2. the image signal of the QR-code frame is processed using the model and the quality parameters are determined;
3. if the quality parameters correspond to the parameters of the basic fragment, then the QR-code image frame is recognized and can be read and decoded;
4. if the parameters do not correspond, then a contrast-enhancing filter is used to emphasize the differences between the intensities of neighboring pixels, removing these intensities from each other;
5. cleaning of the image of the frame based on the improved Susan image filtering method;
6. read and decode information from the QR code.

Let us implement the proposed sequence in the form of software for Raspberry Pi 4 Model B single-board computers.

4 Development of a Control Program for the Object Identification System on the Conveyor Line

Taking into account the features of software development for Raspberry Pi 4 Model B, and the requirements for the terms of reference and the limitations associated with the features of controlling the actuator, as part of the implementation of the tasks in this study to control the developed computer vision system for identifying objects, the following generalized algorithm was developed. It is shown in Figure 7.

At the beginning, we receive an image frame from the camera or video stream. Next, it is necessary to process the image frame in a certain way so that you can remove the excess and get the contours of the selected objects. There is a condition that the objects to be recognized are of the same color and different from the background color. For recognition, let us use the option of selecting objects by filtering the frame in shades of gray according to the limit values, which will bring the image to black and white, i.e. binary display format. To do this, we convert the frame from the RGB color palette to the grayscale palette, and then we determine the values of the shades of background pixels and object pixels to set the threshold values for processing.

After processing, our frame will have the data in a binary image, where black is the background and white is the object to be recognized.

The next step is to use the contour method and obtain contours from a binary image using the algorithm proposed by Satoshi Suzuki [28]. Contours are a useful tool for shape analysis, object detection and recognition. We will get contours in the form of a set of four dots. If the contours were obtained, then we can select extra objects based on the size of the selected contours area, for this we will find the size of the selected objects and filter them by size, that is, we will select small contours that may be garbage or glare.

For identification, according to the condition, it is determined that a QR-code will be used. This algorithm will be used only after the object recognition algorithm. At the beginning of the algorithm, we have an image frame in RGB color format, on which the presence of the object is precisely determined. The algorithm scans the incoming image frame and checks it for QR-codes. If a code or several codes are found, the data containing the code is decoded.

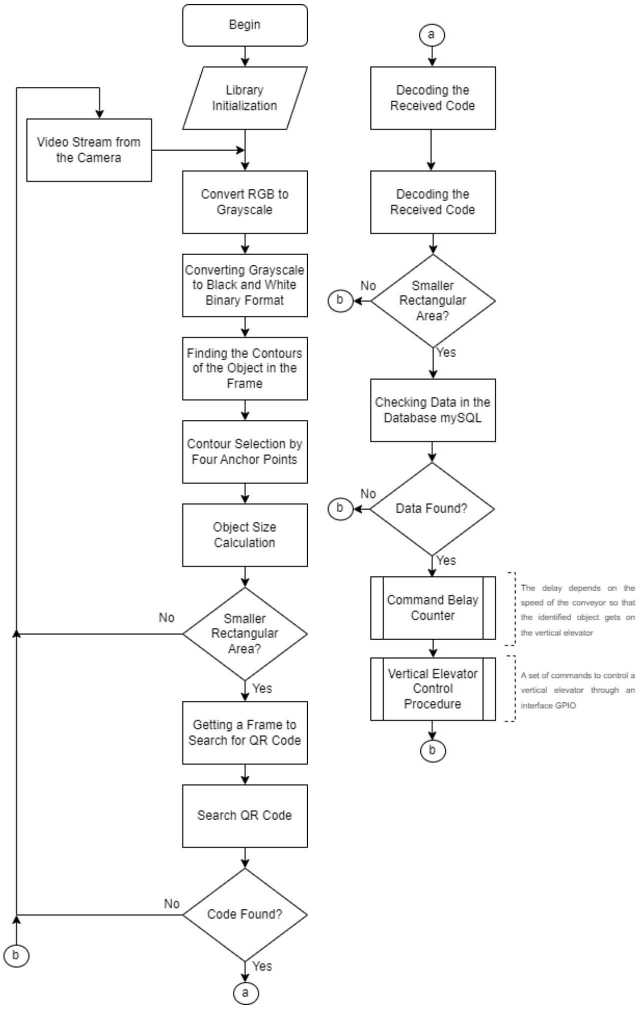

Begin
Library Initialization
Video Stream from the Camera
Convert RGB to Grayscale
Converting Grayscale to Black and White Binary Format
Finding the Contours of the Object in the Frame
Contour Selection by Four Anchor Points
Object Size Calculation
Smaller Rectangular Area?
No
Yes
Getting a Frame to Search for QR Code
Search QR Code
Code Found?
No
Yes
b
a
a
Decoding the Received Code
Decoding the Received Code
Smaller Rectangular Area?
No
b
Yes
Checking Data in the Database mySQL
Data Found?
No
b
Yes
Command Belay Counter
The delay depends on the speed of the conveyor so that the identified object gets on the vertical elevator
Vertical Elevator Control Procedure
A set of commands to control a vertical elevator through an interface GPIO
b

Fig. 7: Generalized Algorithm for Object Identification System based on Computer Vision.

Let's give a description of some moments of the program code that implement interesting solutions. We will start development by connecting the necessary libraries:

```
#include <iostream>
#include <opencv2/objdetect.hpp>
#include <opencv2/core.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/imgcodecs.hpp>
#include <opencv2/highgui.hpp>
```

Let's introduce several global variables used to improve the recognition algorithm.

```
uint32_t objectcounter = 0;
uint32_t objectTrycounter = 1;
bool foundObjectInZone = false;
bool qrCodeIsRead = false;
File *my_file;
```

The "objectcounter" variable is used to count recognized objects. The variable "objectTrycounter" will be used to improve the developed algorithm. The main purpose of this variable is to count attempts to read the code on a recognized object. The variable "foundObjectInZone" is the object recognition flag, the flag is set when the object is recognized, and removed when the object disappears from the working zone. The "qrCodeIsRead" variable is also a flag that is set when the QR code is successfully read. The last variable is a pointer to work with the file, the result of the system will be written to the file.

The main function of the program is performed by the "main" function. This function opens a data stream, it can be a video file or a stream from a camera. If the program failed to open the data stream, the program will terminate with an error. This function opens a file for recording the results, if the file does not exist, it will be created with the name "my_file". Next, in the "main" function, a loop begins, which will be executed as long as the program receives frames from the stream. If the frame was not received, the program will terminate the cycle and end the work.

```
int main()
{
VideoCapture cap("conveyourSlowMediumLight.MOV");
if (!cap.isOpened()) {
cout << "Error open" << endl;
return -1;
}
my_file.open("my_file.txt", ios::out);
while (1) {
Mat frame;
cap >> frame;
if (frame.empty())
break;
FindObjectAndDecode(frame, frame);
imshow("Base after detection", frame);
waitKey(5); // waits to display frame
}
cap.release();
destroyAllWindows();
return 0;
}
```

In the loop, a frame is received from the stream, and this frame is passed to a function called "FindObjectAndDecode" to process the frame, the main purpose of which is to recognize the object in the frame, and in case of recognition, search and decode the QR-code. Also, in the cycle, the frame is displayed on the monitor after processing.

Let's develop a function for processing a frame from a stream. The function is called "FindObjectAndDecode". According to the developed algorithms, this function converts the color palette from the RGB format to shades of gray and filters the obtained shades according to the limit values to obtain a representation of the frame in binary form. The part of the function where this image processing takes place is shown below:

```
vector<vector<Point>> contours;
vector<Vec4i> hierarchy;
Mat imgGrey;
Mat imgthreshold;
cvtColor(_img, imgGrey, COLOR_BGR2GRAY);
threshold(imgGrey, imgthreshold, 70, 255, THRESH_BINARY);
imshow("eee", imgthreshold);
findContours(imgthreshold, contours, hierarchy, RETR_EXTERNAL,
CHAIN_APPROX_SIMPLE);
vector<RotatedRect> minRect(contours.size());
```

An example of obtaining a frame before and after processing is presented in Figure 8 (a,b).

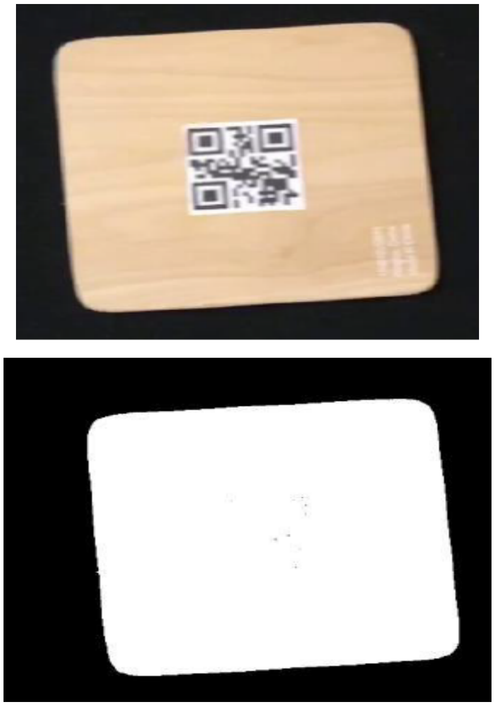


Fig. 8: Frame before and after Processing

Contours are also searched for on the processed frame, for this a function from the OpenCV library called "findContours" is used. This function searches for contours on the transmitted frame and stores contour points in a collection. To reduce memory usage, the function parameter "CHAIN_APPROX_SIMPLE" was used, which tells the function not to store all contour points, but only four as a rectangle.

Next, if the "findContours" function found contours, then we discard any low-area contours found so as not to process false contours. For post-contour processing, we add two boundary lines. If the object is completely between the lines, then you can start searching for the QR-code and decoding it. When the last dots cross the second line, the program decides that the object has left the working area. An example of the resulting frame with bounding lines and a highlighted object that was recognized is presented in Figure 9.

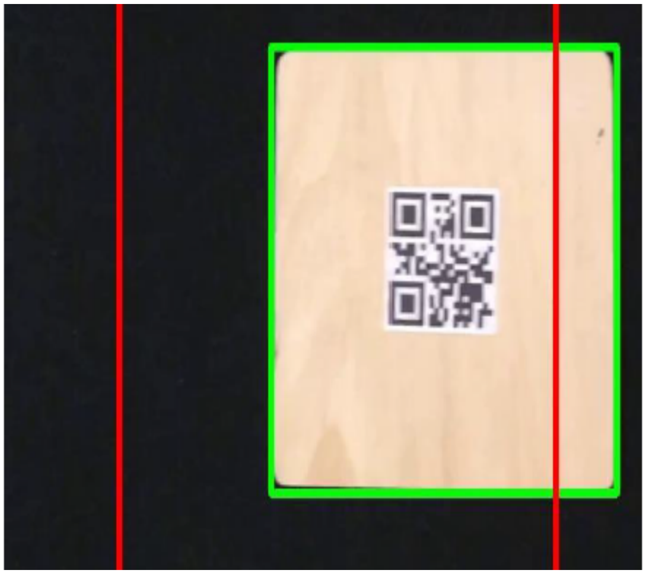


Fig. 9: Frame after Processing with Bounding Lines and Selection

Found Object

Next, we will develop a part of the function where decisions are made on object identification. In this part, global flags and the position of the object relative to the boundary lines are set and checked. If the object completely enters the restricted area for the first time, then this is written to a file, the frame is saved to the storage to the file system, and a function is called to find and decode the QR-code. If the code is found, then in the next frame, the search for the code will no longer take place. If the code was not found and decoded, and the object is still in the restricted area, then there will be an additional attempt, the number of additional attempts is limited only by the working area, that is, when the object leaves this area, no more attempts will be made. Also, in case of an additional attempt, this is recorded in the information file and a frame of each attempt is saved.

```
int xVals[4] = { rect_points[0].x , rect_points[1].x , rect_points[2].x , rect_points[3].x };
std::sort(xVals, &xVals[4]);
if (xVals[3] < 130) {
if (foundObjectInZone) {
foundObjectInZone = false;
qrCodeIsRead = false;
objectTrycounter = 1;
}
}
else if ((xVals[0] > 130) && (xVals[3] < 600) && !foundObjectInZone && !qrCodeIsRead) {
foundObjectInZone = true;
objectcounter++;
imwrite(format("Object%i.jpg", objectcounter), _dst);
my_file << "ObjectFound number:" << objectcounter << endl;
cout << "ObjectFound №" << objectcounter << endl;
Rect crop(130, 50, 470, 850);
Mat croppedImg = _img(crop);
qrCodeIsRead = FindQRAndRead(croppedImg);
}
else if ((xVals[0] > 130) && (xVals[3] < 600) && foundObjectInZone && !qrCodeIsRead) {
imwrite(format("Object%iTry%i.jpg", objectcounter,
objectTrycounter), _dst);
my_file << "ObjectFoundAgain number:" << objectcounter << endl;
cout << "ObjectAgain №" << objectcounter << endl;
Rect crop(130, 50, 470, 850);
Mat croppedImg = _img(crop);
qrCodeIsRead = FindQRAndRead(croppedImg);
objectTrycounter++;
}
else continue;
```

The OpenCV library is used to search and decode the QR-code. In the event of a successful search and decoding, the decoded data will be written to a file, and a vertical elevator control command will be given. If it could not be found or decoded, then a decoding error will be recorded in the file.

5 Conducting Experimental Studies and Analysis of the Obtained Objects Identification Results on Conveyor Line.

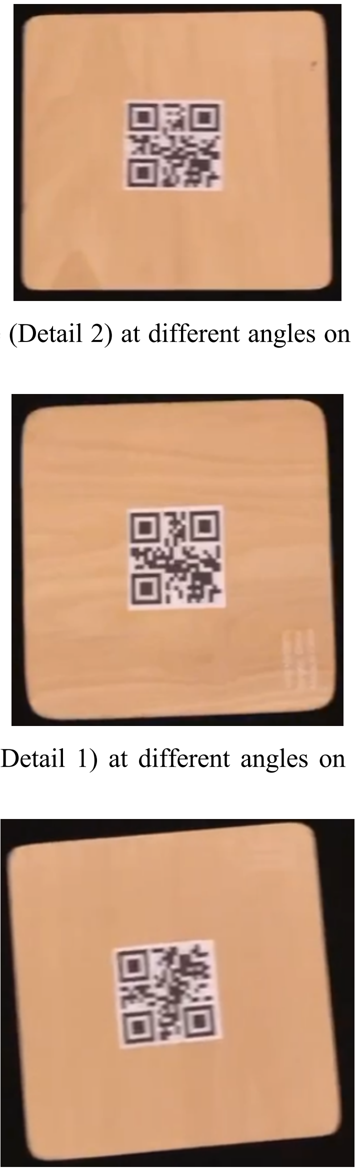
To test the developed object identification layout on an industrial conveyor line for sorting packages using a computer vision system, we will conduct a series of experiments based on the Kapelou equipment, which will allow us to determine the time of object identification (QR-code decoding) at different parameter values: conveyor line speed, working zone illumination and packaging orientation on the conveyor line, we will use Raspberry Pi Cam v1.3 as a camera.

An example of QR-code test for the experiment is presented in Figure 10.

a) QR-code (Detail 2) at different angles on the conveyor line

b) QR-code (Detail 1) at different angles on the conveyor line

c) QR-code (Detail 2) at different angles on the conveyor line



d) QR-code (Detail 2) at different angles on the conveyor line

f) QR-code (Detail 1) at different angles on the conveyor line

g) QR-code (Detail 1) at different angles on the conveyor line

Fig. 10: A set of Test QR-Codes for Conducting Experiments

As a result, we have 6 QR-codes test sets that simulate different angles of packaging orientation on the conveyor line, which complicates the identification process.

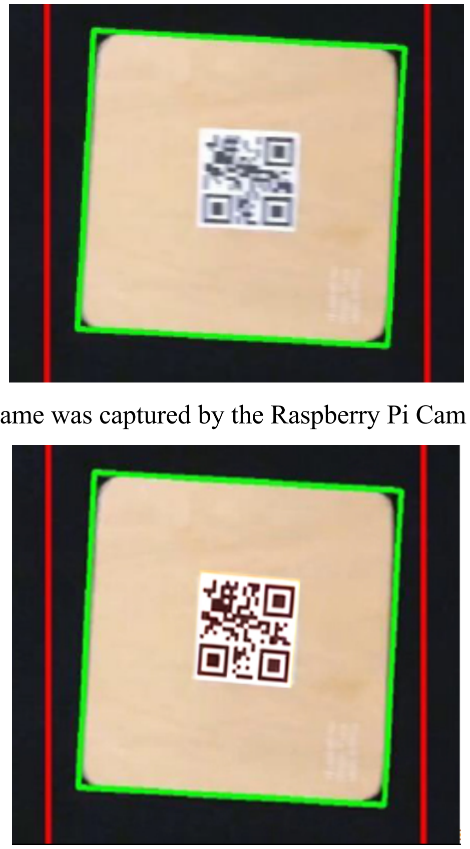
Let's conduct the first experiment with different objects movement speeds on a line. According to the conditions, we will conduct an experiment for a speed of 0.15 m/s and 0.3 m/s. Six objects with QR-codes will be used for the experiment. The obtained results are presented in Table 3.

Table 3: The Results of the First Experiment at Different Speeds of the Conveyor Line.

Object number	Objects movement speed on the conveyor line		
	0.15 m/s	0.20 m/s	0.30 m/s
Object № 1 (Fig.10a)	++	++	++
Object № 2 (Fig.10b)	++	++	++
Object № 3 (Fig.10c)	++	++	+–
Object № 4 (Fig.10d)	++	++	+–
Object № 5 (Fig.10f)	++	++	+–
Object № 6 (Fig.10g)	++	++	+–

++ - Recognized and decoded; +– - Recognized but not decoded

From the obtained data, it can be concluded that the speed of 0.30 m/s is too high for this system. Decoding does not occur due to the limited capabilities of the Raspberry Pi Cam v1.3 camera. An example of a frame obtained by a Raspberry Pi Cam v1.3 camera at a speed of 0.30 m/s is presented in Figure 10a, and for comparison, a frame obtained using a Web cam Logitech C310 HD (model 960-001065) is shown in Figure 11b.



a) the frame was captured by the Raspberry Pi Cam v1.3;

b) the frame was captured by the Logitech C310 HD

Fig. 11: An example of received frames at a speed of 0.30 m/s.

To conduct the second experiment, we will adjust the light flow and programmatically measure the time spent on decoding the code at a speed of 0.20 m/s. The time is measured from the recognition of the object in the frame to the receipt of data from the QR-code, the obtained data are presented in Table 4 and Figure 12.

Table 4: The Results Obtained during the Second Experiment.

Object number	Luminous flux (lux) (QR-code recognized/decoding time)		
	250	100	58
Object № 1 (Fig.9a)	+/0.059 s.	+/0.055 s.	+/0.057 s.
Object № 2 (Fig.9b)	+/0.043 s	+/0.044 s.	+/0.047 s.
Object № 3 (Fig.9c)	+/0.043 s	+/0.046 s.	+/0.045 s.
Object № 4 (Fig.9d)	+/0.069 s	+/0.087 s.	+/0.064 s.
Object № 5 (Fig.9f)	+/0.665 s	+/0.042 s.	+/0.068 s.
Object № 6 (Fig.9g)	+/0.886 s	+/0.044 s.	+/0.044 s.

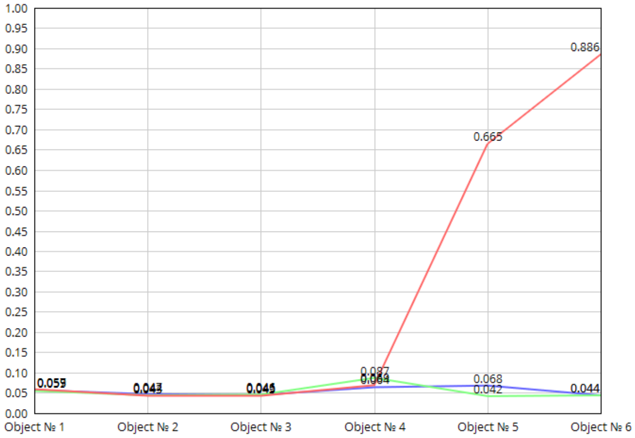


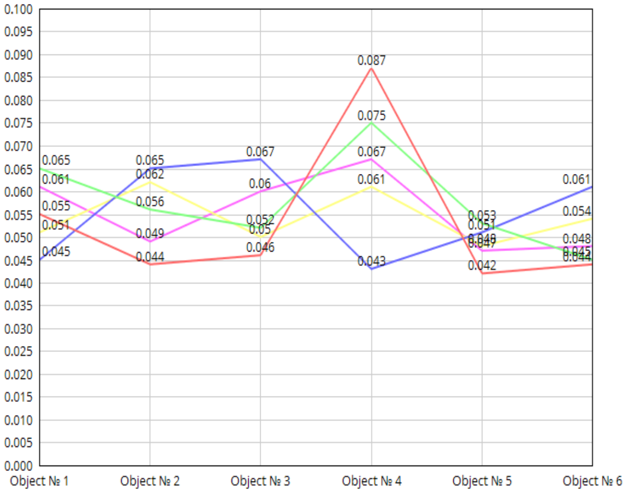
Fig. 12: The Time Spent on Decoding a QR-Code at a Speed of 0.20 m/s

As can be seen, from the results obtained at a light flux of 58-100 lux, all test QR-codes were recognized and decoded, but at a light flux of 250 lux, a delay in the decoding time was obtained for object №5 (Fig. 9f) and object №6 (Fig. 9g), this is related to the reflection of the light stream, the appearance of a light spot on the received QR-code frame.

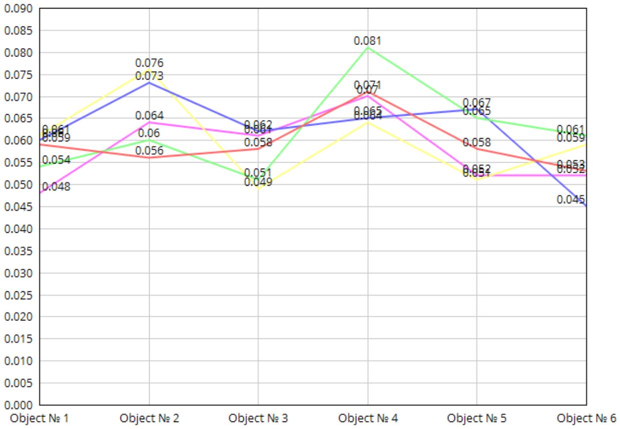
The last experiment is aimed at studying the code decoding time at a speed of 0.20 m/s, when there is no automatic system for aligning the orientation of the packaging on the conveyor line. To conduct this experiment, test QR-codes (Fig. 10) were used, with a rotation angle step of 300, the results of the conducted experiment are presented in Table 5, and in Figure 13a (from 0^0 to 120^0) and 13b (from 150^0 to 270^0).

Table 5: The Results Obtained during the Third Experiment.

Object number	Angle step (degrees)											
	0	30	60	90	120	150	180	210	240	270	300	330
Object №1	0.055	0.065	0.045	0.051	0.061	0.059	0.054	0.060	0.061	0.048	0.058	0.066
Object №2	0.044	0.056	0.065	0.062	0.049	0.056	0.060	0.073	0.076	0.064	0.056	0.065
Object №3	0.046	0.052	0.067	0.050	0.060	0.058	0.051	0.062	0.049	0.061	0.049	0.065
Object №4	0.087	0.075	0.043	0.061	0.067	0.071	0.081	0.065	0.064	0.070	0.071	0.071
Object №5	0.042	0.053	0.051	0.048	0.047	0.058	0.065	0.067	0.051	0.052	0.049	0.054
Object №6	0.044	0.045	0.061	0.054	0.048	0.053	0.061	0.045	0.059	0.052	0.049	0.053



a) QR code decoding time from 0^0 to 120^0



b) QR code decoding time from 150^0 to 270^0

Fig. 13: QR-Code Decoding Time at a Speed of 0.20 m/s, without Automatic System for Orientation Aligning for the Package on the Conveyor Line.

As can be seen from the obtained results of the third experiment, the proposed method, regardless of the different angles at which the QR code for the developed system was presented, the decoding time was from min 0.044 to max 0.087s, which allows us to affirm that the system works well, regardless of the orientation object on the conveyor belt.

To analyze the results obtained on the basis of the developed method, we will compare the average QR-code recognition time with Otsu's, Yao's, Niblack's, Di's and Chen's algorithms [29], [30]. The results of the comparison are given in Table 6.

As can be seen from Table 5, the speed of Otsu's, Yao's, and Chen's [29], [30] algorithms is greater, but the recognition efficiency is low. The indicators of these algorithms are given in the static mode of obtaining a QR-code. And for the developed method all the obtained results were in the dynamic mode of the conveyor lines at a speed of ~ 0.20 m/s. Compared with Niblack's and Di's algorithm, the proposed method has higher average calculation efficiency and recognition speed. In the future, the authors plan to improve the developed method to implement its work on the basis of small IIoT microprocessor devices with a built-in camera, ESP32 family [31,32,33,34,35].

Table 6: Comparison of the Average Calculation Time and QR-Codes Decoding Speed by Six Algorithms

Algorithm	Average Operation Time/s	Recognition Rate	
		~ Zxing	~ WeChat
Otsu's algorithm (Otsu, 1979)	~ 0.0013	32.5%	32.5%
Niblack's algorithm (Ranjitha& Shreelakshmi, 2021)	~ 1.6063	38.75%	86.25%
Yao's algorithm (Yao et al., 2018)	~ 0.0046	37.5%	42.5%
Di's algorithm (Duan et al., 2018)	~ 0.3516	53.75%	71.25%
Chen's algorithm (Chen et al., 2019)	~ 0.0428	88.75%	92.5%
Proposed method	~0.0564	93.42%	96.12%

6 Conclusion

To solve the problem of objects identification and recognition using QR-codes of the industrial conveyor line for sorting by Kapelou, a method of normalizing the QR-code image frame was developed. The Susan method was improved by reducing the pixel group to the size of 3*3 elements. The authors think, this decision makes it possible to speed up the time of decoding information from the QR-code. In order to check the correctness of the proposed solutions, a layout of the control system of the industrial conveyor line was developed using a computer vision system based on Raspberry Pi 4 Model B using a Logitech C310 HD Web-cam camera (model 960-001065) and Raspberry Pi Cam v1.3 . The results obtained during the natural experiment showed a high efficiency (Zxing - 93.42% and WeChat - 96.12%) of calculation and speed of recognition compared with analogues. The developed method can be used in difficult lighting conditions, and in case of slight damage to the QR-code, in automatic sorting systems in dynamic mode in real time conditions. The obtained scientific and practical results can also be applied to conveyor belts of other manufacturers.

References

[1] Maryam Abdirad, Krishna Krishnan, Industry 4.0 in Logistics and Supply Chain Management: A Systematic Literature Review. *Engineering Management Journal*, *33*(3), 187-201 (2021).

[2] Aravindaraj, K., & Chinna, P. R., A systematic literature review of integration of industry 4.0 and warehouse management to achieve Sustainable Development Goals (SDGs), *Cleaner logistics and supply chain*, 100072 (2022).

[3] Pihnastyi, O.M., Khodusov, V.D., Optimal Control Problem for a Conveyor-Type Production Line, *Cybernetics and Systems Analysis 54*, 744-753 (2018).

[4] Skoczylas, A., Stefaniak, P., Anufriiev, S., and Jachnik, B., Belt conveyors rollers diagnostics based on acoustic signal collected using autonomous legged inspection robot, *Applied Sciences*, *11*(5), 2299, 2021.

[5] Shi, X., Tang, K. and Lu, H., Smart library book sorting application with intelligence computer vision technology, *Library Hi Tech*, *39*(1), 220-232 (2021).

[6] Shakeeb Abdullah; Gaozhi Xiao; Rony E. Amaya., A Review on the History and Current Literature of Metamaterials and Its Applications to Antennas & Radio Frequency Identification (RFID) Devices, *Journal of Radio Frequency Identification*, *5*(4), 427-445 (2021).

[7] Dennis Giese, Kevin Liu, Michael Sun, Tahin Syed, Linda Zhang. Security Analysis of Near-Field Communication (NFC) Payments. *Computer Science. Cryptography and Security*, Security analysis of near-field communication (NFC) payments. *arXiv preprint arXiv:1904.10623*, (2019).

[8] Bar-Code Printer Xprinter XP-420B. Available: https://www.amazon.eg/-/en/Bar-Code-Printer-Xprinter-XP-420B-inch/dp/B0969BT7ZS

[9] Yi-Juan Di, Jian-Ping Shi and Guo-Yong Mao. A QR code identification technology in package auto-sorting system, *Modern Physics Letters B*. *31*(19-21), 1740035. (2017).

[10] Yao, S., Li, P., He, L., Li, Y., *Uneven Illumination Two-Dimensional Code Image Recognition Algorithm Research.* In Proceedings of the 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2043–2046, (2018).

[11] Z. Duan, Y. Zhang, S. Li, *Binarization of Nonuniform Illumination Barcode.* In Proceedings of the International Conference on Machine Vision and Applications, 18–21, (2018).

[12] Y. Wu, S. Yu, M. Yang, Quick Response Code Binary Research Based on Basic Image Processing, *Sens. Mater.*, *31*, 859–871 (2019)

[13] L. Yang, Q. Feng, *The Improvement of Bernsen Binarization Algorithm for QR Code Image.* In proceeding of the 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), 931–934, (2018).

[14] Chen, R.; Yu, Y.; Xu, X.; Wang, L.; Zhao, H.; Tan, H.-Z. Adaptive Binarization of QR Code Images for Fast Automatic Sorting in Warehouse Systems, *Sensors*, *19*(24), 5466 (2019).

[15] Informed choice of efficient warehouse logistics. Available: https://kapelou.com/en/

[16] Raspberry Pi 4. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[17] LattePanda V1 Selection Guide. Available: http://www.lattepanda.com:2006/lattepanda-v1

[18] Orange Pi 4 LTS. Available: http://www.orangepi.org/html/hardWare/ computerAndMicrocontrollers/ details/orange-pi-4-LTS.html

[19] Attar, H., Abu-Jassar, A. T., Amer, A., Lyashenko, V., Yevsieiev, V., & Khosravi, M. R., Control System Development and Implementation of a CNC Laser Engraver for Environmental Use with Remote Imaging. *Computational Intelligence and Neuroscience*, *2022*, 9140156 (2022)

[20] Logitech C310 HD Webcam. Available: https://www.amazon.co.uk/Logitech-Widescreen-Calling-Recording-Noise-Reducing/dp/B01BGBJ8RC

[21] Raspberry Pi Camera V1.3. Available: https://makersportal.com/shop/raspberry-pi-camera

[22] Raspberry Pi OS. Available: https://www.raspberrypi.com/software/

[23] G.E.P. Box and David A. Pierce, Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models, *Journal of the American Statistical Association*, *65*(332), 1509-1526 (1970).

[24] Abdul-Majid Wazwaz. A reliable treatment for mixed Volterra–Fredholm integral equations, *Applied Mathematics and Computation*, *127*(2-3), 405-414 (2002).

[25] Gabriela Czanner, Sridevi V. Sarma, Demba Ba and Emery N. Brown. Measuring the signal-to-noise ratio of a neuron, *PNAS*, *112*(23), 7141-7146 (2015).

[26] Youlian Zhu, Cheng Huang. An Improved Median Filtering Algorithm for Image Noise Reduction. *Physics Procedia, 25*, 609-616 (2012).

[27] Susan, S., Rachna Devi, K.M. Text area segmentation from document images by novel adaptive thresholding and template matching using texture cues, *Pattern Analysis and Applications*, *23*, 869–881 (2020).

[28] Satoshi Suzuki, Keiichi A., Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, *30*(1), 32-46 (1985).

[29] Otsu, N., A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, *9*(1), 62-66 (1979).

[30] P. Ranjitha, T.D. Shreelakshmi, *A Hybrid Ostu based Niblack Binarization for Degraded Image Documents.* In 2nd International Conference for Emerging Technology (INCET), 1-7, (2021).

[31] I. Nevliudov, V. Yevsieiev, S. Maksymova, N. Demska, K. Kolesnyk, O. Miliutina, *Object Recognition for a Humanoid Robot Based on a*

Microcontroller. In XVIII International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH). 61- 64, (2022).

[32] H. Attar, A. T. Abu-Jassar, V. Yevsieiev, V. Lyashenko, I. Nevliudov, and A. K. Luhach, Zoomorphic Mobile Robot Development for Vertical Movement Based on the Geometrical Family Caterpillar. *Computational Intelligence and Neuroscience*, *2022*, 3046116 (2022).

[33] Khalid, M. S., & et al. (2022). HMI Development Automation with GUI Elements for Object-Oriented Programming Languages Implementation. *International Journal of Engineering Trends and Technology*, *70*(1), 139-145.

[34] I. Nevliudov, V. Yevsieiev, J.H. Baker, M.A. Ahmad, V. Lyashenko, Development of a cyber design modeling declarative Language for cyber physical production systems, *J. Math. Comput. Sci., 11*(1), 520-542 (2021).

[35] A. T. Abu-Jassar, H. Attar, V. Yevsieiev, A. Amer, N. Demska, A. K. Luhach, and V. Lyashenko, Electronic User Authentication Key for Access to HMI/SCADA via Unsecured Internet Networks, *Computational Intelligence and Neuroscience, 2022*, 5866922 (2022).

Nevliudov Igor – Head of Department of Computer-Integrated Technologies, Automation And Robotics, Kharkiv National University of Radioelectronics, Ukraine. Dr. tech. Sciences, Professor, Honored Worker of Science and Technology of Ukraine, Laureate of the State Prize of Ukraine in the field of science and technology, Laureate of the State Prize of Ukraine in the field of Education. Nevliudov Igor is a head of Scientific School in the field of Technology and Automation of the Radio-Electronic Instrument production, in which 27 Candidates of Engineering Science (PhD) and five Doctors of Engineering Science were prepared. Under his leadership, the research projects at the expense of the state budget are successfully provided (9 projects, from 2002 to present) also the business projects for industrial enterprise’s orders. Fields of interest are production automation and control systems development.

Yevsieiev Vladyslav - Professor, Doctor of Engineering Science, Professor of Department of Computer-Integrated Technologies, Automation And Robotics, Kharkiv National University of Radioelectronics, Ukraine. Dissertation was about cyber-physical control ystems development. His fields of science interests are – Cyber-Physical Systems, Internet of Things, Industrial Internet of Things, Robotics. He has published research articles in reputed international journals of mathematical and engineering sciences.

Maksymova Svitlana - Associate Professor, Candidate of engineering science (PhD), Associate Professor of Department of Computer-Integrated Technologies, Automation And Robotics, Kharkiv National University of Radioelectronics, Ukraine. Her scientific interests are connected with Robotic from the robots development up to robot control system development. Her dissertation theme was connected with robot voice control system development. She has published research articles in reputed international journals of mathematical and engineering sciences.

Murad Anver Oglu Omarov – Vice-Rector on International Cooperation, Kharkiv National University of Radioelectronics, Ukraine. Member of Academic Council, Member of Specialized Scientific Council, Honored Worker of Science of the Republic of Azerbaijan, Doctor of Technical Sciences, Professor. In 1991he completed the postgraduate studies and defended his thesis in the field of fiber-optic transmission systems. In 2003 he defended his doctoral thesis on “Development of the Theory and Practice of Design of Electrodynamic Devices with Distributed and Quasiexperimental Nonlinear Elements”. Since 2011, M. A. Omarov is a member of the Advisory Council on the preparing of foreign citizens and citizens of Ukraine for entry into higher education institutions with the State Accreditation Commission of Ukraine, since January 2013 – member of the Specialized Council for doctoral theses (speciality 05.12.02 – telecommunication systems).

Klymenko Oleksandr – Managing Direktor LLC KAPELOU, post-graduate student at Department of Computer-Integrated Technologies, Automation And Robotics, Kharkiv National University of Radioelectronics, Ukraine. His fields of interests are production logistics and warehousing.