OBJECT SORTING IN MANUFACTURING INDUSTRIES USING IMAGE PROCESSING

Manoj Sabnis¹, Vinita Thakur ², Rujuta Thorat², Gayatri Yeole², Chirag Tank²
¹Assistant Professor, ²Student, Department of Information Technology V.E.S Institute of Technology, Mumbai, India

ABSTRACT

In manufacturing industries, there arises a need to sort objects. The objects may be of similar or different types. The system should be able to detect the objects and then differentiate the objects from each other based on their properties. Objects may have different shapes or different colours. The objects may be of same shape and same colour but different texture. Thus, different objects and different conditions require different type of processing. Our aim is to classify objects using different image processing algorithms on the parameters like colour, shape and texture. The input for the system will be a video, which will be converted into frames and then processed for detecting the colour, shape or texture.

Keywords: Object detection, Sorting, Segmentation, Edge Detection, Video Processing

[1] INTRODUCTION

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which the input is an image or a video and output may be image or characteristics associated with that image.

It is among the rapidly growing technologies today, with its applications in various aspects of business. Image Processing forms a core research area within engineering and computer science disciplines too. Visual images are probably the most important means by which human beings experience their environment. However the information required by the human beings is often only a small part of the data in the image. Under some circumstances it becomes extremely inefficient in extracting information.

Thus there is a demand for techniques that process the image after capture but before presentation to the human being in a manner that makes it easy for us to extract the information we require. Industrial automation and robotics are at a high demand in the industry as both of them directly affect the growth of the industry. The system is developed with the purpose to optimize the productivity, minimizing the cost of labour and make no human mistakes. The main aim of the system
OBJECT SORTING IN MANUFACTURING INDUSTRIES USING IMAGE PROCESSING

is to increase the accuracy for sorting objects according to the manufacturer’s need. The objects should be classified as per the parameter selected by the user and statistics of the number of objects should be displayed.

MATLAB has the most powerful toolbox for image improving, enhancing and categorizing different images. This toolbox provides a comprehensive set of reference-standard algorithms, functions, and apps for image processing, analysis, visualization, and algorithm development. One can perform image analysis, image segmentation, image enhancement, noise reduction, geometric transformations, and image registration. Many toolbox functions support multi-core processors, GPUs, and C-code generation.

[2] RELATED WORK

Colour Sorting by a Robot:
Colour is the most common feature to distinguish between objects, sorting, recognizing and tracking. Generally robot is mounted with a camera or the camera is mounted in the workspace to detect the object. This technology can be used in material handling in logistics and packaging industry where the objects moving through a conveyer belt can be separated using a colour detecting robot. An algorithm is written and executed to identify the object and send the appropriate commands to the microcontroller using serial communication for the robot to perform the sorting operation.

Hardware Implementation:
The hardware implementation deals in:
1) Drawing the schematic on the plane paper according to the application
2) Testing the schematic design over the breadboard using the various IC”s to find if the design meets the objective.
3) Designing the PCB layout of the schematic tested on the breadboard.
4) Finally preparing the board and testing the designed hardware.

Development of the System
A webcam was mounted on the Robot which was connected to the USB port of the PC. The specifications of the camera are as follows:
- CMOS camera with plug and play USB connection (with driver software)
- Video data format: 24 bit RGB
- 30 fps max.

A simple approach for developing object reorganization system is shown below:
1) Decide the ideal position of the object with respect to the camera.
2) The distinguishing feature of the object to be picked is to be figured out.
3) Deciding the robots movement as planned.

The colour of the object will serve as the distinction on the basis of which it will be identified. In order to have accurate colour detection a few live images of the object must be captured and the pixel values for different colours should be noted. Taking a mean of the pixel values obtained through various images would be fair and justifiable. This threshold range will then be used to mark all pixels containing the object as „1” and all other pixels as „0”.
[3] PROPOSED WORK

The main aim of the system is to increase the accuracy for sorting objects according to the manufacturer’s need. The objective can be summarized as below:

- A video needs to be given as input to the system. The video will contain, objects moving on the conveyor belt. The camera should be placed at the top, so as top view of the objects is captured in the video.
- This video will then be processed and converted into images. After video processing, one should be left with one image of each object.
- The images should be classified as per the parameter selected by the user and statistics of the number of objects should be displayed.

Step 1: Video Processing:

A video consisting of objects of different shapes and colours is captured. The video is taken by placing the camera in a way such that the top view of objects is captured. This video is then converted into a series of images. These images contain noise, which must be removed. MATLAB provides various filters for noise removal. Some of the functions available for noise removal are: imadjust, medfilt2

As the number of images is quite large, redundant images need to be removed. There would exist two types of redundancies:

1) There may be images in which the object is only seen partially or is touching the boundary of the image. Such images need to be removed as they cannot be used for sorting. These images are detected by scanning the boundaries of the images. If pixels are found at the boundary, it indicates that the object is touching the boundary and hence such an image needs to be deleted. In MATLAB, there a function named imclearboundary. This function when applied to a black and white image, clears the objects that are touching the boundary. The original image is then compared with the cleared border image. If both images are same, it means that the original image need not be deleted. If both images are different, then delete the original image. This way, images which have objects that touch the boundary, are deleted. Image comparison can be done by counting the number of black pixels or white pixels in the images.

2) The remaining images are now images of the complete object. But, these images contain the same object getting repeated many a times. Multiple images containing the same object should be removed and only one image should be stored. Images can be compared by calculating the extent and determining the colour of objects contained in the image. Extent is the ratio of the area of an object to the area of its bounding box. Extent has been described in detail, in the shape detection algorithm.

3) These images are then given as an input to the shape, colour and texture algorithms.

Step 2: Applying algorithms

Shape Detection Algorithm:

For shape detection of objects, the algorithm uses Bounding box method. The shapes defined for classification are Rectangle, Square, Circle, Hexagon and Triangle.

Step 1: Read/capture image
The image is in RGB format which is a true colour format for an image. In MATLAB, the captured or imported RGB image is three dimensional and each pixel is represented by an element of a matrix whose size corresponds to the size of the image.

**Step 2: Converting RGB image to Black and White Image**

This process is done in two steps. The RGB image is first converted to a two dimensional grayscale image. The grayscale image is nothing but a matrix that holds the luminance (Y) values of the image. We obtain the luminance values of the image by combining the RGB values using the NTSC standard equation that multiplies the primary colors (red, green and blue) with coefficients based on the sensitivity of our eyes to these colours as shown:

\[ Y = 0.3 \times R + 0.59 \times G + 0.11 \times B. \]

The luminance image is then converted to black and white (binary) image by a process called thresholding. MATLAB provides a function, `BW = im2bw(I, threshold)` that converts the grayscale image `I` to a binary image.

**Step 3: Recognize boundaries of objects**

The image is now a two dimensional array with binary elements. Boundaries of the objects are recognized by first setting a single pixel on the object-background interface as a starting point and moving in a clockwise or counter-clockwise direction and searching for other object pixels.

**Step 4: Finding areas of objects and area filtering**

Once the object boundaries have been recognized, the area of that object can easily be calculated by summing the number of pixels within the boundary extent. The image is filtered to remove small, isolated noise pixels by inverting it.

**Step 5: Finding bounding box of the object**

The bounding box of an object is an imaginary rectangle that completely encloses the given object and its sides are always parallel to the axes. Figure below illustrates the concept of a bounding box. It is worth noting that due to the various angles of inclination of an object, the dimensions of the bounding box change accordingly.

However, to make the shape recognition independent of the rotation of the object, the dimensions of the bounding box must be constant. This is because the area of the bounding box is an important parameter which we will be using to classify the shape of the object.

**Step 6: Finding ratio of areas for given object**

The next step involves finding the ratio of the area of an object to the area of its bounding box. In MATLAB, this ratio is known as the *Extent* and is a very useful parameter:
**Extent = Area of the object/ area of bounding box**

For circles, this value is around 0.7853, irrespective of the radius. The corresponding value for rectangles and squares is approximately 1.0000, provided the sides are parallel to the axes and the bounding box and sides overlap. The objects were first made independent of rotation. As a result, the value of *Extent* is seen as constant. Thus, whenever an object that has value of Extent approximately equal to 0.7853 is a circle. Further, whenever an object is encountered that has value of Extent approximately equal to 1.0000, the object may be a square or a rectangle. The decision between square and rectangle can be made on the basis of the dimensions of the object. The object with equal sides is obviously a square. The object with 2 unequal pairs of sides is obviously a rectangle. Similarly, extent for triangles and hexagons can be calculated.

![Figure 3.3 Procedure for Shape Recognition](image)

**Colour Detection Algorithm:**
For colour detection of objects, the algorithm uses RGB colour space. The colours defined for classification are Red, Green, Blue and Black. 

**Step 1 and Step 2** are same as Shape Detection Algorithm

**Step 3: Colour Recognition**

In the Colour Recognition Algorithm, centroid pixel of the object whose colour is to be detected is extracted. A MATLAB Function, `P = impixel(I,x,y)` is used which returns the values of pixels in the specified image, I, where x and y are the centroid values. Now in the 3D matrix of RGB colour space, if the value of the specified centroid pixel has more value in Red space than blue or Green, then consider that pixel Red. Similar goes with Green and Blue.

**Texture Detection Algorithm:**

**Step 1:** Create Texture Image

Use entropyfilt to create a texture image. The function entropyfilt returns an array where each output pixel contains the entropy value of the 9-by-9 neighbourhood around the corresponding pixel in the input image. Entropy is a statistical measure of randomness.
Step 2: Create Rough Mask for the Bottom Texture
Threshold the rescaled image to segment the textures. A threshold value of 0.8 is selected because it is roughly the intensity value of pixels along the boundary between the textures. The segmented objects in the binary image are white. If we compare binary image with input image, we notice the top texture is overly segmented (multiple white objects) and the bottom texture is segmented almost in its entirety. We can extract the bottom texture using bwareaopen. Use imclose to smooth the edges and to close any open holes in the object. A 9-by-9 neighbourhood is selected because this neighbourhood was also used by entropyfilt.

Step 3: Use Rough Mask to Segment the Top Texture
Compare the binary image rough mask to the original image. The mask for the bottom texture is not perfect because the mask does not extend to the bottom of the image. However, rough Mask is used to segment the top texture.

Step 4: Obtain Textured image
Use entropyfilt to calculate the texture image. This detects the texture within an image.

Procedure Summary:

![Procedure Summary Diagram]

Figure 3.4 Procedure Summary

[4] RESULTS AND DISCUSSIONS

![Sorting Parameters]

Figure 4.1 Sorting Parameters
Figure 4.2 Detection of Green Object
Figure 4.3 Detection of Blue Object
Figure 4.4 Detection of Black Object
Figure 4.5 Detection of Red Object
Figure 4.6 Color Analysis
Figure 4.7 Shape Analysis
[6] CONCLUSION

The Object Sorting Algorithms stated above detect the objects and classifies them on different parameters. The Automatic Object Sorting System is developed with a view to decrease the human effort and make wider use of such systems in Manufacturing and Packaging Industries where there is a need to sort objects and then perform operations on them. The system also proves to be cost efficient since it eliminates the manpower required to manage the object queue and also to sort the objects.

[7] REFERENCES


6. Object Sorting System in Matlab using Robotic Arm

7. Object sorting system using robotic arm
