Proceedings of the Fourth International Conference on Inventive Systems and Control (ICISC 2020)
IEEE Xplore Part Number: CFP20J06-ART; ISBN: 978-1-7281-2813-9

Identification and Sorting of Objects based on Shape and Colour using robotic arm

Mr. Lennon Fernandes
P.G. Scholar, Dept. of E&C Engineering
NMAM Institute of Technology, Nitte
lennfernz@gmail.com

Mr. Shivakumar B.R.
Assistant Professor, Dept. of E&C Engineering
NMAM Institute of Technology, Nitte
shivkumarbr@gmail.com

Abstract— **Robotic Automation is favored since human eyes are unable to constantly distinguish colors which will eventually reduce the work efficiency. In this work, a system that sorts objects based on their shape and color is proposed and implemented using a robotic arm. The system captures a real-time image from a web camera and is preprocessed by performing RGB to HSV conversion and noise removal by using a median filter. Colour detection of the objects is then carried out using the lower and upper HSV values. The system then performs an object's shape detection using the contour detection technique. In this technique the contours are identified first using the modified boundary fill approach and the shapes are then detected from the contours using the Douglas-Peucker algorithm. A Robotic arm is constructed to sort objects based on their identified shape and color. The system is tested for three object shapes: square, triangle, and rectangle.**

I. INTRODUCTION

Object detection and tracking is an active area of research and development in the field of computer vision which operates through a sequence of frames to identify, recognize and track objects [3]. In many parts of the knowledge society, object detection is commonly utilized, providing support wherever required. For computer systems, it is a challenge as to what people see as the simple process of recognition. Object identification algorithms rely on matching, learning or even pattern recognition techniques by making use of appearance-based or feature-based approaches [4]. Identifying objects based on their shapes is an important task in computer vision especially in understanding digital images. Many application areas depend on shape recognition that may include robotics, medical applications, assistance for the impaired, etc. In terms of object sorting, robotic automation is preferred since human eyes cannot continuously discern colors that ultimately reduce the quality of the job. Robotic Automation would minimize the responsibility of the operator on the site to be physically present to perform the duties assigned to him.

The aim of this paper is to design and implement a robotic system for industrial applications that can detect and sort objects based on their definite shape and color. The system detects objects in an image based on their shape and color by carrying out the color detection process using lower and upper HSV values. The shape of the objects is detected using a contour detection technique where contours are computed from the detected edges and shapes of the object are identified using the Douglas-Peucker algorithm.

The rest of the paper is organized as follows: Section II discusses the literature on the proposed system and its techniques, section III presents the methodology of the proposed system, section IV presents the design and implementation of the robotic arm, section V discusses the results obtained from the implemented system, and section VI presents the conclusion drawn from the obtained results.

II. RELATED WORK

Ferrari et al. [1] have presented a series of scale-invariant features of local shapes obtained by a large number of k connected roughly straight contour segments (kAS). kAS are utilized to encode pure fragments of the surface of the object by eliminating the nearby clutter. A new algorithm for identifying and tracking boundary contours in a binary image is presented by Ren et al. [2], which is widely used in digital image processing. A novel approach for identifying and locating objects of a visual category in most of the cluttered areas is proposed by Sun et al. [3]. This approach is mainly aimed at utilizing a highly flexible learned representation for object shape that can integrate the differences observed on various training examples of the Generalized Hough Transform (GHT). A general, trainable system for object detection in non-restricted, cluttered scenes is addressed by Papageorgiou et al. [4]. Their system utilizes a technique based on an over accomplished dictionary of Haar wavelets that obtains a certain amount of information regarding elements of the object class. The techniques that incorporate two complementary bottom-up approaches are discussed by Brox et al. [5] which comprises object edges and patches of texture to direct top-down image segmentation to higher accuracy.

The issues pertaining to visual object class identification and localization in natural images are discussed by Laptev [6]. Also, thorough analyses are carried out for certain image features. Validation of the method on recent benchmarks for object recognition shows its superior performance.

A generic location approach that is proposed by Seo et al. [7] is widely known for encountering a visual object of interest without the requirement of training. The proposed approach utilizes an object of interest to obtain similar objects, without having prior knowledge (learning) regarding objects being searched for, and without having any requirement for pre-processing steps. An adaptive object detection approach is introduced by Murguia et al. [8] to function without human intervention in complex backgrounds. The approach suggested is based on a neural-fuzzy design. The neural phase deals with the effective framework for object detection and the removal of shadows. A new convolutional neural network (CNN) is developed by Tian [9] to assimilate the communication technique for the detection of RGBD items. RGBD image is a combination of an RGB image along with its corresponding depth image. Numerous efforts have been made to design various low-level saliency cues for RGBD saliency detection, such as color and depth contrast features. A novel polygonal approximation approach for curve representation is developed by Lelin Li et al. [10]. This algorithm is based on the foundation of the Douglas-Peucker algorithm.

Huang et al. [11] used a structural feature selection template to introduce an object detection approach. This method first creates a class-specific codebook of contour parameters and then produces descriptors of structural attributes by integrating data about background shape information. Marie et al. [12] investigated two fundamental issues pertaining to computer vision namely contour detection and image segmentation. The contours are identified by utilizing multiple local cues and combining them into a globalization framework based on spectral clustering.

Feng et al. [13] have discussed techniques to develop a data structure that enables the utilization of approximation techniques to get polygonal boundaries of the foreground objects in an image. In addition, it provides a technique for modifying the regions encountered from the utilization of a segmentation technique. Pavlidis et al. [14] have discussed methods that integrate area expansion and edge detection used for image processing. A split-and-merge approach has been implemented in which the values are set so that an over segmented image is obtained. Yuh Tai-Liow [17] has presented a new algorithm that identifies the contours and traces them using the extended boundary principle that shifts the contour's inner boundary by a half-pixel to the right and bottom position. It is depicted that the new algorithm can be beneficial over the conventional techniques in addition to the traditional boundary representation: i. it incorporates more information in each contour, and ii. simpler and more efficient implementation.

III. METHODOLOGY

In this work, objects are detected based on shape and color and sorted accordingly using a robotic arm. Real-time images are captured using a web camera. The images captured from the web camera are initially in RGB form. The images in RGB form are not suitable for image processing to be carried out in real-time since in RGB form the color information cannot be separated from luminance. Whereas HSV color space obtains the color (hue) by separating it from saturation and pseudo-illumination. Therefore the images in RGB form are converted to HSV for efficient processing. This is the first step for image pre-processing. The second pre-processing step is to remove noise from the HSV image. The median filter is used for eliminating noise from the image. The median filter gives better performance as compared to other filters namely Gaussian filter, mean filter, etc. since the blurring intensity is better than that of the other filters and is hence more effective in removing noise from the raw image which makes it efficient for shape detection to be carried out.

Three colors are considered in this paper namely red, blue and yellow. To detect the colors initially the lower and upper HSV color values for red, blue and yellow are obtained. The lower and upper HSV values are used for each of these colors instead of fixed HSV values for each of the three colors since the color detection is carried out in real-time. The implementation is carried out using the *OpenCV* library on the python platform. The lower and upper HSV values for each of the three colors are obtained using *numpy* arrays. *Numpy* is a highly optimized library for numerical calculations with a MATLAB-style syntax. The Lower and Upper bounds are [1 x 3] arrays which contain the lower and upper values for Hue, Saturation and Value. After obtaining the lower and upper HSV values, the color masks are created separately for each of the three colors using *numpy* arrays. Masks are separate binary images that contain only black and white pixels. The black pixels will be the background while the desired color will be highlighted in white. Masking is performed in order to eliminate the unwanted regions of interest and maintain only the desired objects of interest.

After the color masks are obtained, the edges of the foreground objects are detected using a canny edge detection technique. From the edges detected the contours are obtained from the binary image using a modified boundary fill technique. There are two methods in which this boundary fill approach has been implemented namely Four-connected pixels and Eight–connected pixels. In four-connected pixels the four pixels adjacent to the current pixel are checked which include pixels towards the left, right, top and bottom whereas in eight-connected pixels not only the four adjacent pixels are filled, but also the adjacent diagonal pixel positions are filled. Eight-connected pixel technique is preferred since four connected pixel techniques cannot fill all the surrounding pixels, hence it is advantageous over four connected pixels.

After the detection of contours, the detection of shapes can be carried out with the help of the Douglas-Peucker Algorithm [10]. The main objective of the algorithm is, given a curve comprising of line segments, to obtain a similar curve with lesser points.

The overall methodology of the proposed system is summarized as follows:
Consider an RGB Image $Img_{(x,y,z)}$ captured using the system's web camera where (x, y) denotes the dimensions of the mask image and *z* denotes the color component.

i. Convert the RGB image to the HSV image and remove the noise using the median filter.
ii. Obtain the lower and upper HSV color values for red, blue and green color given by:
$$Lower_{HSV} = [H - 10,100,100] \quad (1)$$
$$Upper_{HSV} = [H + 10,100,100] \quad (2)$$
where $Lower_{HSV}$ and $Upper_{HSV}$ are [1x3] arrays in which the first element denotes the *hue* component, the second element denotes the *saturation* component, and the third element denotes the *value*. In the above equations, parameter H denotes the *hue* component.
iii. Obtain the mask Image $Img_{(x,y)}$ for each color separately using their lower and upper HSV values, where (x, y) denotes the dimensions of the mask image.
iv. Compute the edges of the mask image $Img_{(x,y)}$ using Canny edge detection technique as illustrated in the equations:
$$|G| = |Gx| + |Gy| \quad (3)$$
$$\phi(x, y) = arctan(\frac{Gx}{Gy}) \quad (4)$$
where, $|G|$ and $\phi(x, y)$ denote the gradient magnitude and direction, and $|Gx|$ and $|Gy|$ denote the Sobel masks in x- and y-direction.
v. Compute the area of the foreground object given by Green's formula:
$$A = \sum_{k=0}^{n} \frac{(x_{k+1}+x_k)(y_{k+1}-y_k)}{2} \quad (5)$$
where, n is the number of vertices, (x_k, y_k) denotes the initial point, on the contour and $(x_k + 1, y_k + 1)$ denotes the next point.
vi. Let the minimum area be denoted by *min_area* and the maximum area be denoted by *max_area*. *min_area* and *max_area* are user-defined as per the project requirements.
vii. Compute the centroid for each foreground object as illustrated in the equation:
$$centroid = (\frac{\mu_{(1,0)}}{\mu_{(0,0)}}, \frac{\mu_{(0,1)}}{\mu_{(0,0)}}) \quad (6)$$
where, μ_{10}/μ_{00} and μ_{01}/μ_{00} define the center of gravity.
viii. If object area > *min_area* and object area < *max_area*, obtain the contours of the foreground objects using a modified boundary fill approach.
ix. From the contours detected, compute the perimeter and shape the length of the foreground objects using the Douglas-Peucker algorithm.
x. If shape length = 3, then shape = triangle, where shape length denotes the sides of a particular shape. In this case, the shape is a triangle and the shape length is '3' indicating that the triangle has three sides.
xi. If shape length == 4 and 0.96<aspect ratio<1.05 then shape = square else shape = rectangle.

The overall methodology of the object detection system is depicted using a flowchart as shown in Fig 3.

IV. IMPLEMENTATION OF ROBOTIC ARM

The robotic arm is constructed with the aid of servo motors. The servo motors enable the robotic arm to identify, pick and sort objects. The robotic arm is made up of five servo motors namely base, shoulder, elbow, wrist, and gripper. Each of the servo motors has 3 pins i.e. ground, power and data link. The ground and control pins of all the five servos are connected to an existing 5V and 2A power source, while the data pins are wired to the PWM (Pulse Width Modulation) pins of Atmega328, where Atmega328 is a single-chip 8-bit microcontroller. Since the Atmega328 has PWM connectors, it is used to send data to the servo motors and is therefore programmed accordingly to control the servo motors.

The Atmega328 is attached serially to OdroidC2 which is a 64 bit quad-core ARM development board. The robotic arm is designed to identify and arrange objects based on their shape and color. To capture the image of the desired object, a web camera is utilized. Image processing techniques are utilized to identify objects based on shape and color and to send the command via serial communication to the Atmega328 microcontroller which is programmed to command the robotic arm to select and arrange objects as programmed by the user.

Once the objects are identified, the data is sent serially to Atmega328 which then programs the robotic arm to pick the object and sort them at the desired location that is already programmed. The base servo motor of the robotic arm is responsible for rotating the robotic arm to the desired position i.e. horizontal movements, while the shoulder, elbow, and wrist motors are responsible for the vertical movements of the robotic arm i.e. to lift and place the object at the desired location. Finally, the gripper servo is responsible for picking and dropping the object. Once the object is picked and sorted the control is sent back from the Atmega328 to the main program on Odroid C2 and the process continues for the remaining objects. The identification and sorting of objects at different locations (user-defined) using the robotic arm is displayed in Fig 4.

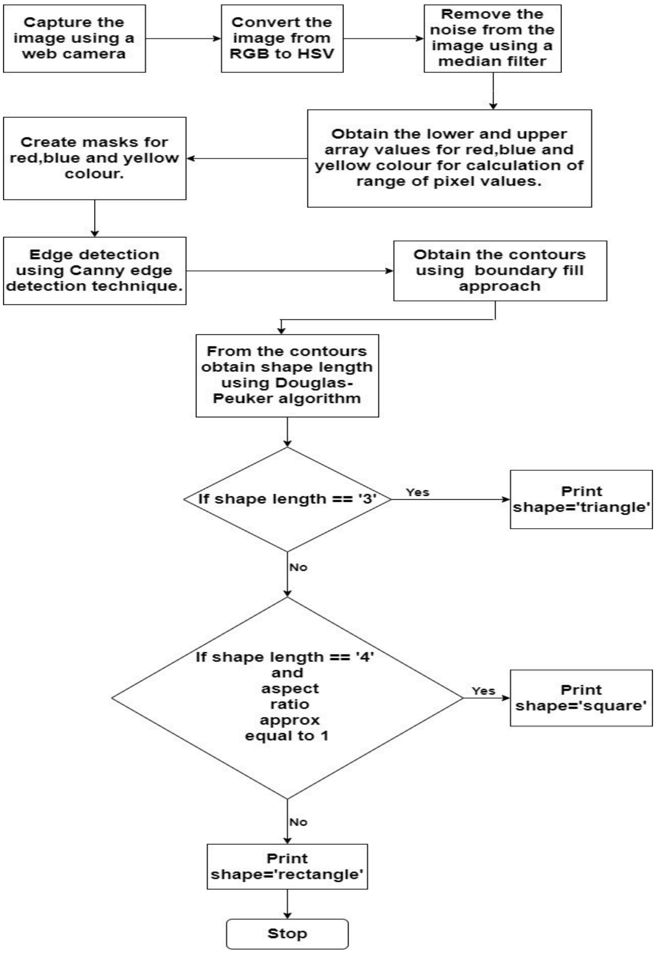


Fig 3 -: Flowchart of object detection based on shape using contour detection technique

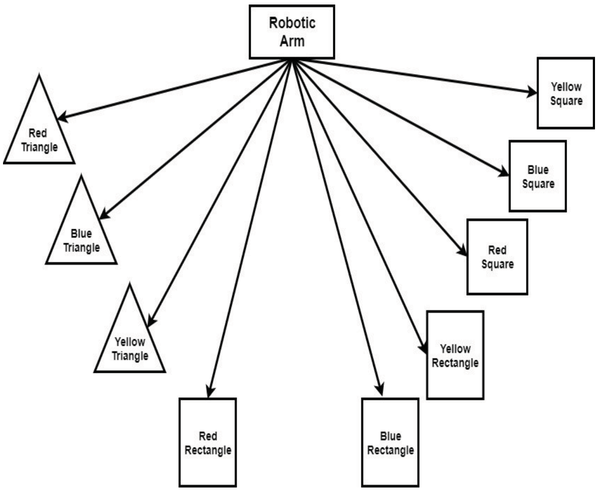
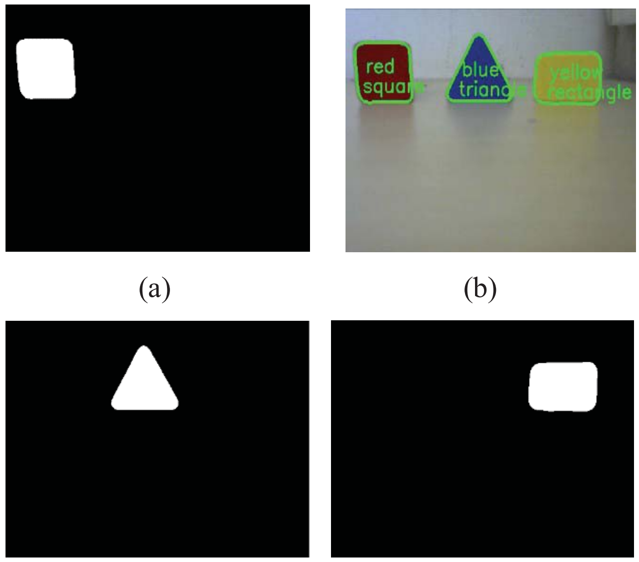


Fig 4: Objects sorted at different locations using a robotic arm.

V. RESULTS AND DISCUSSIONS

This section presents and discusses the results obtained during the study. Real time images captured using the web camera are processed on the python platform. The images which were initially in RGB form have been pre-processed to remove noise thus making it suitable for color and shape detection. Colour masks were used, separately, for red, blue and yellow color. Fig. 5 shows the detection of three different colored objects of three different shapes namely square, rectangle and triangle.

Fig. 5 shows the three different colored objects that are detected simultaneously in one frame along with their respective mask images shown separately in Fig 5(a),5(c) and 5(d). The mask image is obtained with the help of a *numpy* arrays in the python platform where the *numpy* arrays contain the lower and upper HSV values for each of the three colors detected. Masks for different colors are created separately.



(a) (b)

(c) (d)

Fig 5: Detection of a red square, blue triangular and a yellow rectangular object (a) Red squared mask image (b) A red square, blue triangular and a yellow rectangle original image (c) Blue triangular object mask image (d) Yellow rectangular object mask image.

Fig 5(a) shows the mask of the red square detected using *OpenCV library*. This mask is a binary image that has been created by considering the original RGB image that is converted to HSV along with the lower and upper HSV values for the red color. Similarly, the color masks for blue triangle and yellow rectangle as shown in Fig 5(c) and 5(d) are obtained using the lower and upper HSV values for blue and yellow colors. The Canny edge detection technique is implemented on the mask image in order to detect edges.

Contours are obtained from the edge image using a modified boundary fill approach. In order to carry out shape detection, the contours detected are taken into consideration. Shape detection then is carried out with the help of the Douglas Peucker algorithm.

A. Construction of Robotic Arm

The construction of the robotic arm is performed with the help of the servo motors and is shown in Fig 6. The base of the arm is made up of an acrylic sheet cut according to the required shape and dimension. The arm has a circular base that can be modified to a mobile platform if required. A rotating plate rests on top of this platform. The plate is balanced on four castor balls. The plate rotates with the help of a servo motor fixed to the circular base. The shoulder joint is mounted on the plate. The elbow's servo is joined to the shoulder's servo followed by the wrist's servo. To attain the ability to rotate an object, another servo is attached to the wrist perpendicular to the wrist joint. Finally, a gripper mechanism is attached to the rotating wrist's servo.

The gripper mechanism is constructed around a smaller servo, such that both the gripper-fingers are controlled by this single servo. The length of the acrylic links is critical to the mechanical design. If the links are too long, the servo motors will not have enough torque to lift any load at all. If the links are too short, it will reduce the total reach of the arm and the motors might not be able to move freely without hitting each other.

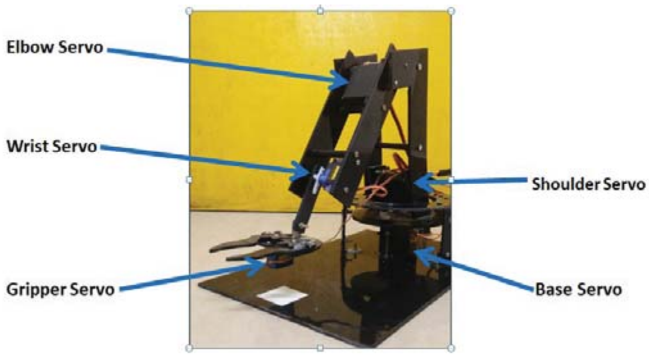
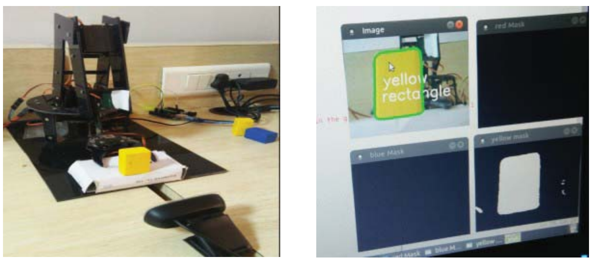


Fig 6: Robotic arm constructed for the study



(a) (b)

(c) (d)

Fig 7: Detection and sorting of a yellow rectangle. (a) Detection of a yellow rectangle. (b) Yellow rectangle detected using *OpenCV* on the python platform. (c) Picking of a yellow rectangle. (d) Sorting of a yellow rectangle.

The identification of a yellow rectangle using the *OpenCV* library and picking and sorting of the desired object using the robotic arm is shown in Fig 7. A C290 Logitech web camera is used to capture the image of the objects placed in front of it. Fig. 7 (a) shows a yellow rectangle placed in front of the web camera. Fig 7 (b) shows the yellow rectangle detected using the contour detection technique. A mask image of the yellow rectangle is also shown in Fig 7 (b). The yellow color is detected using the lower and upper HSV values for the yellow color, while the rectangle is detected using a contour detection technique. In this technique, the contours are identified first using a modified boundary fill approach and the shapes are detected using the Douglas-Peuker algorithm. After the detection of the yellow rectangle, the command is sent to the Atmega328 from Odroid C2 via serial communication that is programmed accordingly to pick and sort the object at the desired location. Atmega328 is used to control the servo motors connected to the robotic arm. Fig 7 (c) shows the robotic arm picking the yellow rectangle and placing it in the desired location. As seen in the figure, the gripper servo of the robotic arm is used to hold the object, while the wrist, elbow and shoulder servo is used for vertical movement of the arm, and the base servo is used for horizontal movement. Finally, Figure 7 (d) shows the robotic arm placing the object at the desired location.

VI. CONCLUSIONS

In this work, a system for the detection and sorting of objects based on their shape and color is designed and implemented. The system used three primary shapes; triangle, square, and rectangle and three primary colors: red, blue, and green during for implementation. The designed system successfully identified all the considered shapes and colors of the objects. Douglas-Peucker algorithm was used to identify the desired objects based on their shape. The Robotic Arm was programmed and integrated with the *OpenCV* platform in order to identify objects based on their shape and color and sort them accordingly. The robotic arm successfully sorted all the objects considered during the study. In the future, it is intended to improve the system further by including other colours and shapes of objects.

References

[1] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, "Groups of Adjacent Contour Segments for Object Detection (slides)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 1, pp. 36–51, 2013.

[2] Mingwu Ren, Jingyu Yang, Han Sun,"Tracing boundary contours in a binary image,"*Image and Vision Computing.,*vol 20, pp.125-131,2012

[3] Z. Sun, G. Bebis, and R. Miller, "Object detection using feature subset selection,"*Pattern Recognit.,* vol. 37, pp. 2165–2176, 2014.

[4] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection,"*Pattern Recognit.,* vol. 38, no. 1, pp. 15–33, 2010.

[5] T. Brox, L. Bourdev, S. Maji, and J. Malik, "Object Segmentation by Alignment of Poselet Activations to Image Contours,"*IEEE Trans. Pattern Anal. Mach. Intell.,* pp. 2225–223, 2012

[6] I. Laptev, "Improving object detection with boosted histograms," *Image Vis. Comput.*, vol. 27, no. 5, pp. 535–544, 2009.

[7] H. J. Seo, S. Member, and P. Milanfar, "Training-Free, Generic Object Detection Using Locally Adaptive Regression Kernels," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 32, no. 9, pp. 1688–1704, 2010.

[8] M. I. Chacon-murguia, S. Member, and S. Gonzalez-duarte, "An Adaptive Neural-Fuzzy Approach for Object Detection in Dynamic Backgrounds for Surveillance Systems," *IEEE Trans. Industrial Electronics* vol. 59, no. 8, pp. 3286–3298, 2012.

[9] J. Tian, L. Qu, S. He, J. Zhang, Y. Tang, and Q. Yang, "RGBD Salient Object Detection via Deep Fusion," *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 14, no. 8, 2017.

[10] Lelin Li, Wanshou Jiang, "An Improved Douglas-Peucker Algorithm for Fast Curve Approximation," *International Congress on Image and Signal Processing.,*pp. 1797-1802, 2010.

[11] S. Huang and B. Chen, "In Variable Bit Rate Video-Based,"*Pattern Recognit.*, vol. 24, no. 12, pp. 1920–1931, 2013.

[12] M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[13] H.Y.Franklin feng and Theodosios Pavlidin,"The Generation of Polygonal Outlines of Objects from Gray Level Pictures ,"*IEEE Transactions on Circuits and Systems*, vol.23, no.5, pp.101 -106 2017.

[14] Theo Pavlidis and Yuh-Tai Liow,"Integrating Region Growing and Edge Detection",*IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol.12, no.3, pp.225-233, 2015.

[15] Yuh -Tai Liow,"A Contour Tracing Algorithm That Preserves Common Boundaries between Regions ", *IEEE Trans. Pattern Anal. Mach. Intell,* vol. 53, no.3, pp. 313-321, 2010.