Designing of an Automatic Object Identifier Based on Probabilistic Machine Learning: An Experimental Study

Md Kamrul Hasan Chowdhury, Basudeb Biswas
Department of Mechanical Engineering,
Rajshahi University of Engineering and Technology
Rajshahi, Bangladesh
Email: iitpeterson736@gmail.com, basumech89@gmail.com

Spandan Basak Payel, Md. Tanvir Siraj
Department of Mechanical Engineering,
Bangladesh University of Engineering and Technology
Dhaka, Bangladesh
Email: spandanbasak@gmail.com, tanvir25392@gmail.com

Debashis Biswas
Department of Electrical Engineering,
National Institute of Technology
Silchar, India
Email: debashis.bs@hotmail.com

Abstract

In response to the growing need for autonomous systems that can interpret complex visual information akin to human cognition, this study introduces a probabilistic machine learning approach for visual object detection, demonstrating significant strides in identifying images through a computational model that emulates human cognitive precision. The method begins with capturing images via video cameras, followed by an algorithmic learning phase that employs Blob Analysis to discern diverse object features such as color, shape, and region. These attributes are cataloged in a database, facilitating the matching and recognition of objects against a learned dataset. This technique refines the focus-of-attention mechanism, ensuring high probability that non-target regions do not contain the object of interest. The proposed system shows exceptional promise in optical character recognition and content-based image indexing, with experimental results underscoring its efficiency and accuracy in object detection. This advancement could significantly impact robotic vision systems, enhancing their interpretative capabilities in varied and complex environments.

Keywords
Probabilistic Machine Learning, Visual Object Detection, Blob Analysis, Focus-of-Attention, Image Indexing

1. Introduction
The advent of digital technology has revolutionized the field of computer vision, particularly in object detection, where the imperative is not just to perceive but to understand imagery through identification and localization of objects (Russakovsky et al. 2015). This technology, grounded in processing, analyzing, and understanding high-dimensional image data, is integral to numerous applications, from enhancing space exploration to refining public security

measures. Despite remarkable progress, the quest for systems that parallel human cognitive capabilities in recognizing and interpreting objects under varied conditions remains a formidable challenge.

Object detection, a subset of computer vision, involves distinguishing and pinpointing specific object categories such as humans, structures, and vehicles within digital images and videos (Rajesh Kumar et al. 2022). This process is the cornerstone for other nuanced areas, including face and pedestrian detection, with far-reaching implications for both practical and innovative applications—ranging from image-based search engines to autonomous vehicle navigation.

However, object detection encapsulates more than mere image analysis; it requires a sophisticated interpretation of data to yield a high-level understanding, akin to human perception (Salari et al. 2022). In the context of robotics, this implies identifying an object from a known set, irrespective of the image capture conditions—viewpoint, illumination, or occlusion—which is an area where substantial advancements are still to be made.

Image processing is the foundational step in this journey, with digital images serving as arrays of finite binary words, manipulated to various ends: generating new images, extracting measurements, or formulating high-level descriptions. Such manipulation often involves selective operations on regions within an image to enhance specific features, such as suppressing motion blur or improving color rendition.

With the escalation of autonomous processes in daily life, the importance of object detection has surged, particularly in robotics, where detection accuracy and reliability are crucial. The current landscape of algorithms spans from edge matching, which is resilient to changes in lighting and color, to more complex divide-and-conquer search strategies and greyscale matching (Shinde et al. 2014).

Despite the plethora of methods and their applications in diverse fields, including outer space sciences, automated parking systems, and visual tracking, a conspicuous gap persists in the automatic exploration and identification of objects based on partial regions of interest (ROIs). This gap is underscored by the limitations in current methodologies that struggle with partial occlusion and varied viewpoints, hindering the realization of truly autonomous robotic vision systems.

This research aims to bridge this gap by introducing a novel approach to object detection that hinges on probabilistic exploration. The goal is to develop an algorithm that not only detects but also learns from partial object features, constructing a robust feature database, and leveraging this information to ascertain the probability of an object's identity. This method promises to refine the precision of object detection, particularly in robotics, where such capabilities are paramount.

The contributions of this research are multifaceted. Firstly, it advances the understanding of automatic object detection by introducing a probabilistic model that accounts for partial ROIs, thereby expanding the scope of object detection under challenging conditions. Secondly, it proposes a novel feature extraction method that enhances the current state-of-the-art algorithms. Lastly, it incorporates these advancements into a robot vision system designed for practical applications, thus demonstrating the real-world utility of the research.

1.1 Objectives

The objectives of this study are:

- To extract features from partial objects and construct a comprehensive database.
- To develop a probabilistic model that can accurately compute the likelihood of object identities based on their features.
- To implement a control system for camera movement that responds to the calculated probabilities, enhancing object detection in robotics.

The scope of this research extends to a variety of domains, where the developed algorithm and system can be crucial. These include, but are not limited to, Mars exploration robots, OCR, content-based image indexing, and automated vehicle systems. By addressing the identified research gap, this study lays the groundwork for significant advancements in object detection and robotic vision, potentially catalyzing further innovation in the field.

2. Literature Review

The computational mechanisms underlying the human visual attention system have been the subject of extensive research. An early architecture proposed by Koch and Ullman (1985) laid the groundwork by suggesting that a scene is dissected into multiple feature maps, culminating in a singular saliency map to direct attention towards regions of highest activity. This foundational model has since spurred a multitude of enhancements and variants.

Subsequent researchers, including Hu et al. (2004), Privitera and Stark (2000), and Itti et al. (1998), have introduced modifications to this model, each proposing iterations that refine the original concept. Itti and Koch (2001) provide a comprehensive review of these techniques, charting the evolution of visual attention models and their increasing sophistication.

Building on these physiological models and leveraging advancing technologies, various systems have been designed to emulate visual attention for object detection in scenes. Tagare et al. (2001) took inspiration from these models to develop a maximum-likelihood decision strategy, which represented a significant step towards practical application in known-object identification within images.

The advent of robotic platforms gave rise to active vision systems, marking a departure from the passive analysis of static scenes. Ude et al. (2003) introduced a humanoid system that utilizes both peripheral and foveal vision to detect and track a known object, illustrating the dynamic capabilities of these systems. This concept was further explored by Björkman and Kragic (2004), who proposed a system for object recognition and pose estimation, indicating a trend towards more complex and interactive visual processing systems. Orabona et al. (2005) contributed to this dialogue with an attention-driven system for visual exploration, aiming to identify the most salient object in a static scene.

Li et al. (2020) developed a novel adaptive attention mechanism that improved object detection by incorporating channel-wise, spatial-wise, and domain attention units into YOLOv3 and MobileNetv2, leading to enhanced performance on KITTI and Pascal VOC datasets. Similarly, Lu et al. (2023) combined self-attention with YOLOv4 to create SwinT-YOLOv4, which uses the Swin Transformer for more accurate traffic scene analysis. This model showed improved precision and mAP, particularly for detecting cars and pedestrians, despite a marginal increase in missed detections.

The chronological progression of these studies illustrates the field's trajectory towards increasingly interactive and sophisticated visual attention systems, with each study building upon the insights of its predecessors. The transition from static image analysis to dynamic, robotic vision reflects the field's ongoing endeavors to replicate human-like perception in machines. These collective efforts underscore the importance of integrating physiological insights with technological advancements to enhance object detection and recognition capabilities.

3. Theoretical Background

3.1 Blob Detection

Blob detection is a critical process in image analysis where the goal is to identify and isolate various objects, or 'blobs', within an image. Blobs, essentially binary large objects, are recognized as clusters of connected pixels that stand out due to their distinct brightness or color compared to their surroundings. Each blob is characterized by features such as area and centroid, calculated from its pixel composition. The common approach for blob detection involves creating a model of the object of interest, computing its features, and then comparing these to the features of each detected blob in the image to determine matches (see Figure 1).

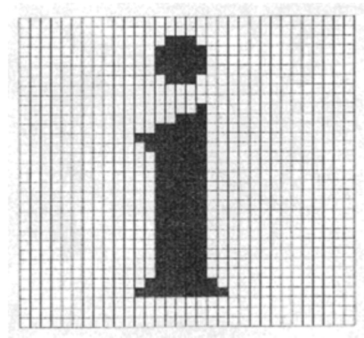


Figure 1. Blob selected.

3.2 Blob Detection Algorithm

To perform blob detection in an image, the algorithm begins by creating a duplicate of the original image for processing. The base case conditions dictate that if a pixel is not activated ('off') or if it lies outside the image boundaries, it is assigned a value of zero. In the recursive case, for an activated ('on') pixel, the algorithm first deactivates the pixel to prevent reprocessing and then calculates the blob size by adding one (for the current pixel) to the sum of the values returned from the recursive calls to its four neighboring pixels.

3.3 Blob Detection Process

The Blob Detection process aims to isolate each object in an image by creating a list of pixels that belong to it. This is achieved through connected component analysis or region growing, where defining connectivity—whether 4-connected or 8-connected—determines the pixel neighbors (see Figure 2). In a binary image, the process starts at a seed point and employs the 'Grassfire' technique, effectively erasing pixels as they are visited to avoid revisiting (see Figure 3). The algorithm continues until all connected neighbors, according to the defined connectivity, have been visited and accounted for.

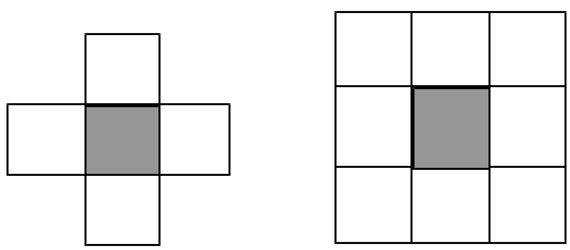


Figure 2. 4-connected (left) and 8-connected (right) pixel

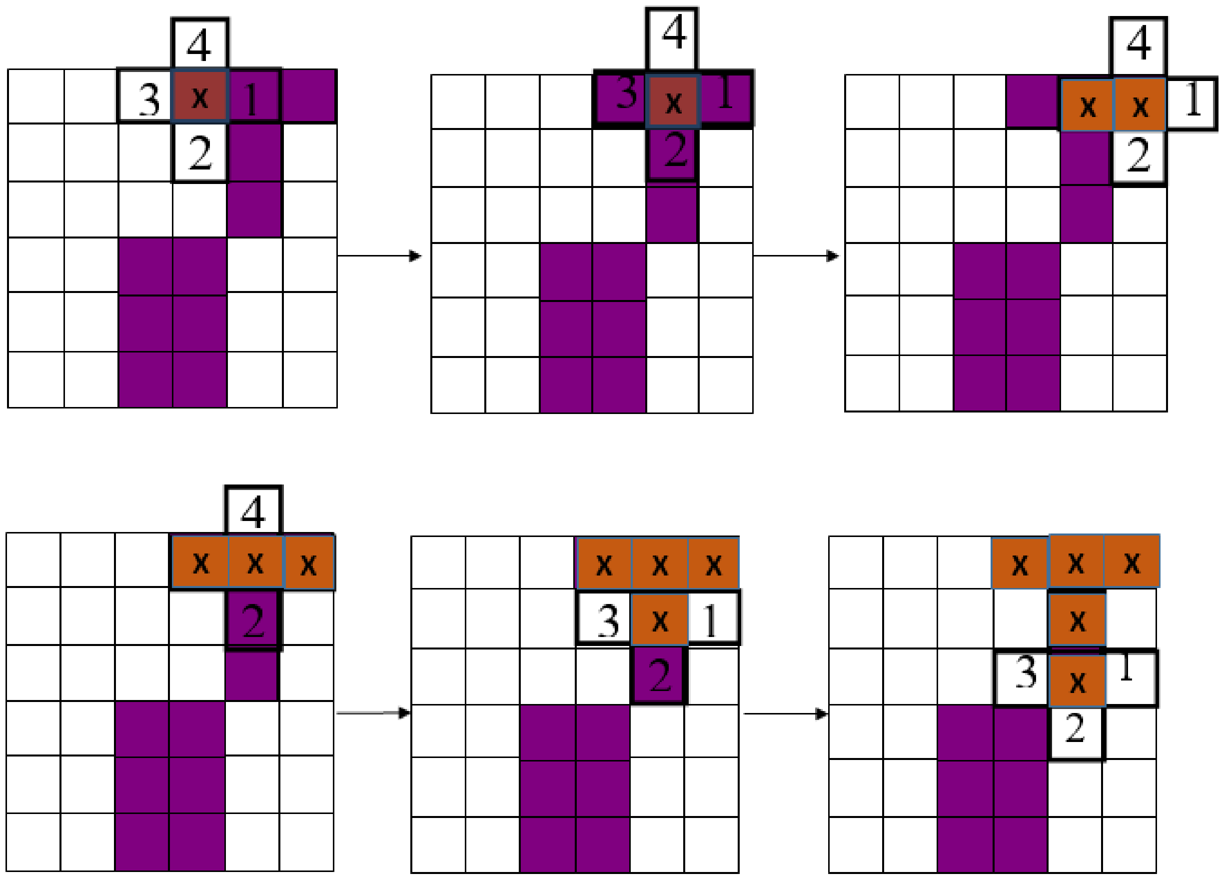


Figure 3. Connected component analysis.

3.4 Feature Extracting Process

A feature refers to a distinct element within an image, such as a corner, blob, edge, or line. Feature extraction is the process of deriving descriptors or feature vectors from these detected elements. The Computer Vision System Toolbox provides a suite of tools for this purpose, including corner detection methods like Shi & Tomasi, Harris, and FAST, blob and region detection via SURF and MSER, and the extraction of descriptors such as SURF, FREAK, and simple pixel neighborhood descriptors. Additionally, it facilitates the visualization of features, emphasizing their location, scale, and orientation.

In pattern recognition, entities such as numbers that represent size and shape can be generated for use. This process involves transforming image operations into mathematical operations, where an input list of pixel positions is processed to yield a feature vector. The initial step in this transformation is to filter out blobs that are too small, too large, or located at the borders, ensuring that only relevant features are extracted for recognition tasks (see Figure 4).

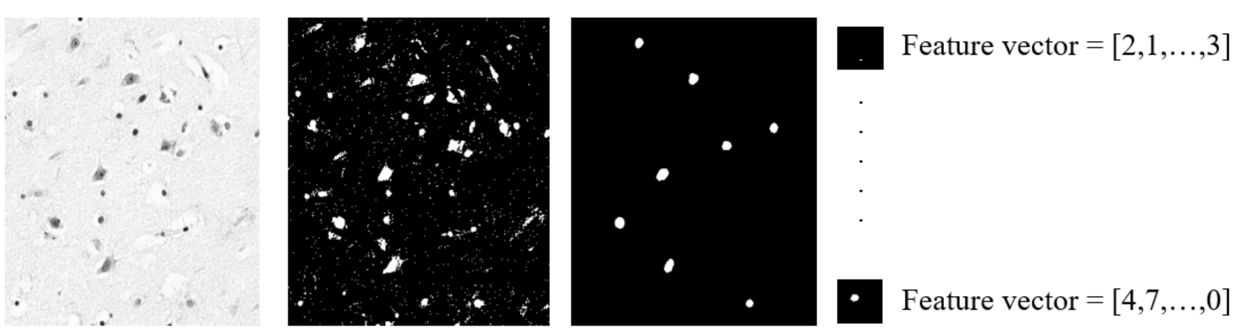
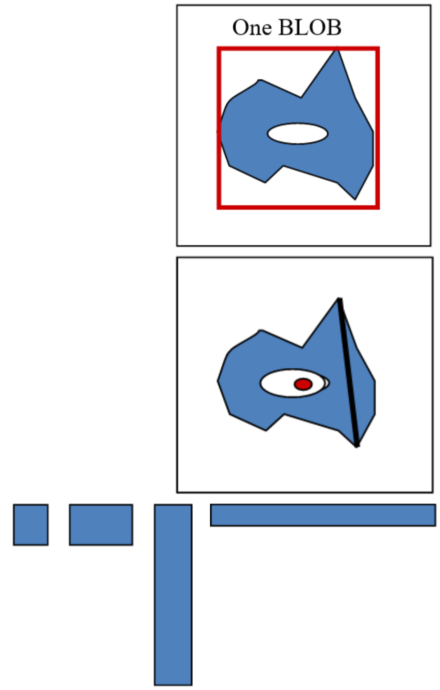


Figure 4. Feature extraction

The features are-

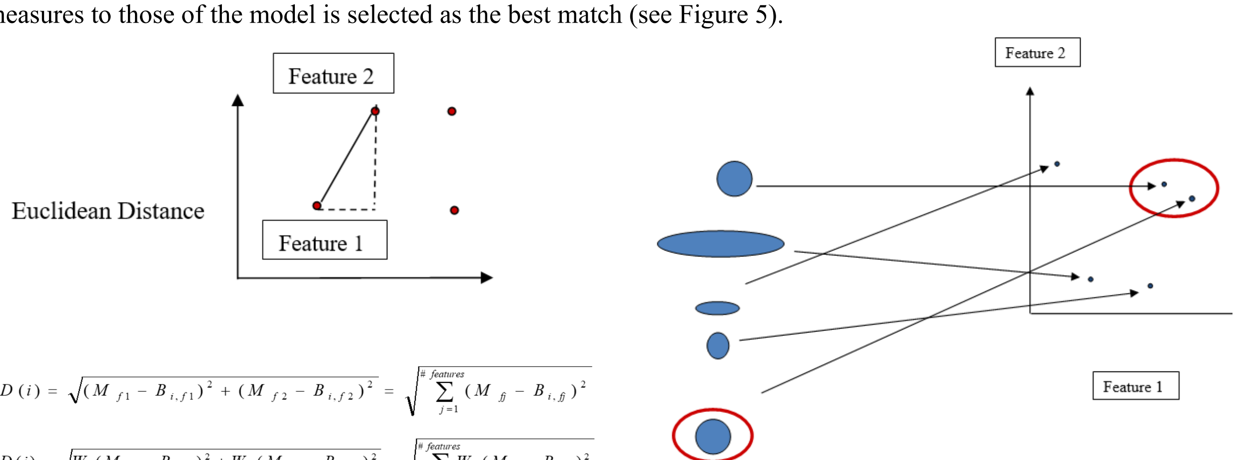
- Area (number of pixels)
  - Used to remove noise (small/big objects)
- Number of holes in the object
- Holes' area
- Total area = area + holes' area
- Perimeter = length of contour
- Bounding box
  - Upper left corner
  - Height and width of bounding box
- Centre of mass (x_m, y_m)
  $x_m = \sum_{i \in \text{object}} x_i, y_m = \sum_{i \in \text{object}} y_i$
- Compactness
  - Minimum for a disc = $\frac{Perimeter2}{Area}$
  - Minimum for a rectangle = $\frac{Area}{Width.Height}$
- Circularity = $\frac{Perimeter}{2\sqrt{\pi}.\ area}$
- Shapes
  - Bounding box ratio: Height / Width
  - How well the object fits into a rectangle.
    - Can be derived from: Area and Perimeter
  - How well the object fits into an ellipse.
    - Can be derived from: Area and Perimeter



3.5 Feature Matching

Feature matching involves comparing feature descriptors from different images to establish point correspondences. The Computer Vision System Toolbox facilitates this through various matching metrics like Sum of Absolute Differences (SAD), Sum of Squared Differences (SSD), and normalized cross-correlation. It also incorporates Hamming distance for binary feature comparison and provides several matching methods such as Nearest Neighbor Ratio, Nearest Neighbor, and Threshold. Moreover, the toolbox is optimized with multicore support, enhancing the speed of execution for large sets of features.

To determine the closest match to a model in blob detection, the distance of each blob from the model in feature-space is measured, with the shortest distance indicating the most similar blob. Distance is defined within a 2-dimensional feature space, considering attributes like area and circularity. The blob that exhibits the closest area and circularity measures to those of the model is selected as the best match (see Figure 5).



$$D(i) = \sqrt{(M_{f1} - B_{i,f1})^2 + (M_{f2} - B_{i,f2})^2} = \sqrt{\sum_{j=1}^{\#\ features} (M_{fj} - B_{i,fj})^2}$$

$$D(i) = \sqrt{W_1(M_{f1} - B_{i,f1})^2 + W_2(M_{f2} - B_{i,f2})^2} = \sqrt{\sum_{j=1}^{\#\ features} W_j(M_{fj} - B_{i,fj})^2}$$

Figure 5. Distance matching

However, the problem with measuring blob features like area in large units (e.g., $10m^2$) and circularity on a different scale ([0,1]) is that area can disproportionately influence the distance measure. To mitigate this, ratios (such as width to height) can be utilized as more balanced features, or features can be normalized to the same interval, typically [0,1]. Additionally, when certain features are considered more reliable than others, such as perimeter measurements for small objects, weighting can be adjusted to reflect the varying levels of trust in these features.

4. Methods

4.1 Study Framework

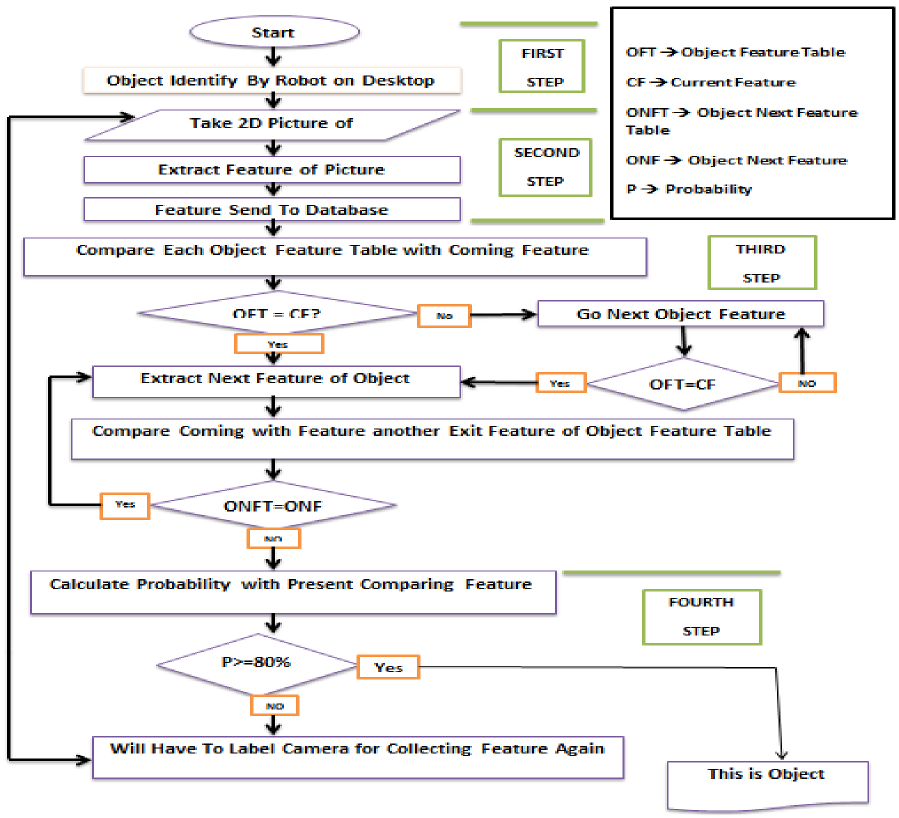


Figure 6. Methodological flowchart of the study

The flowchart of this study is shown in Figure 6 which outlines a four-step process for object identification by a robot. Initially, the robot captures a 2D picture and extracts its features, which are then sent to a database. The second step involves comparing the extracted features with existing ones in the object feature table. If a match is found, the process seeks the next set of features; if not, it moves to the next object's features. The third step calculates the probability of a match with the current features. If this probability is over 80%, the object is recognized. Otherwise, the camera is relabeled to collect features again, indicating the process's iterative nature. This systematic approach emphasizes the importance of feature comparison and probability calculation in robotic object identification.

In this study, the process for object identification by a robot involves several key steps: starting with the robot recognizing an object placed on a desktop, capturing a 2D picture of the object, and extracting its features. These features are then compared against a database of known object features. The system calculates the probability of a match, and if this probability exceeds 80%, the object is successfully identified, and its name is called out. Initially, the system is designed to recognize four distinct objects: a computer, a mobile phone, a pen, and a pen drive, by taking their 2D images and applying a predetermined flowchart methodology to ascertain their identities based on different extracted features.

The tools and platform for the project include OpenCV 2.4.5, a real-time computer vision library developed by Intel and primarily written in C++, which is cross-platform and BSD-licensed. Code:Blocks 10.05 serves as the cross-platform IDE, facilitating OpenCV configuration. CMake 2.8 manages compiler-independent build processes, supporting complex directory hierarchies and multiple library dependencies, compatible with build environments like Make, Xcode, and Visual Studio. The setup also incorporates a camera/webcam for image capture and operates on a Windows system.

4.2 Experimental Setup

The setup for our project encompasses several main components crucial for its operation (see Figure 7). A laptop serves as the central processing unit, offering portability and integrating essential features like a webcam and microphone for real-time vision and sound capture. The webcam, connected typically via USB, allows for live video streaming and various interactive applications. For precise movement control, a stepper motor is employed, working in unison with a rack and pinion system to convert rotational to linear motion. Power is supplied through both 12 V and 5 V adapters, ensuring adequate energy for all components. USB cables facilitate data and power transfer, adhering to the universal industry standard. Finally, a microcontroller acts as the brain, managing input/output peripherals, memory, and processing tasks for embedded applications, optimizing for both power efficiency and performance.

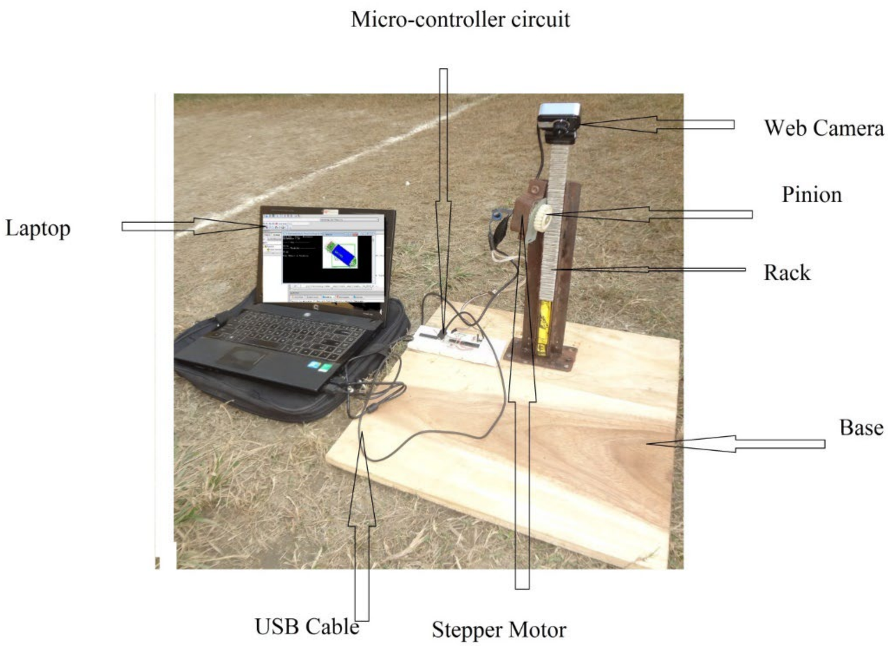


Figure 7. Experimental setup

4.3 Experimental Steps

Our program has been tested on an image, referred to as Figure 8, containing seven distinct blobs. It successfully identified each blob, drawing bounding boxes and calculating the area for them, with Figure 8 showcasing these results. Further analysis provided detailed outputs like bounding box coordinates, inter-blob distances, center positions, densities, and free spaces. Additionally, Figure 8 demonstrates the program's capability to process practical images, specifically analyzing a mobile phone and a computer, which have been the two main objects of study in this research.

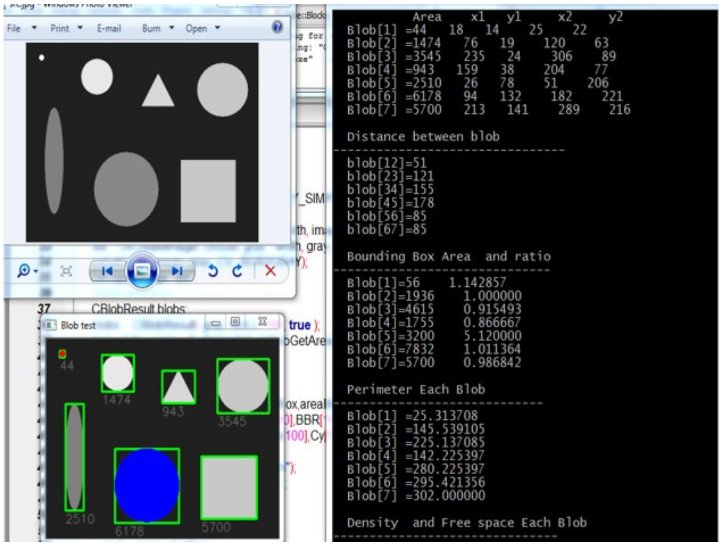


Figure 8. Finding various blobs and values of test images using OpenCV

In this paper, we focus on two objects: a mobile phone and a computer. Utilizing a practical image of a mobile phone as input, shown in Figure 9, we apply a blob detection program to identify various blob features within the mobile image. The resulting output, depicted in Figure 9, illustrates the detected blobs and their characteristics, such as bounding box areas, coordinates, distances between blobs, ratios, center positions, densities, and free spaces. These extracted values, constituting the mobile's database, are then used to calculate the probability of the mobile against a broader objects database.

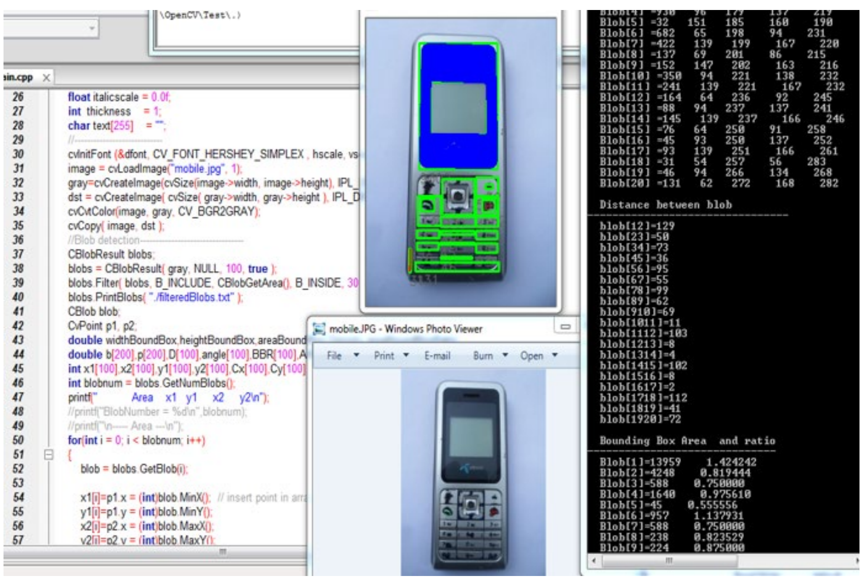


Figure 9. Finding various blobs and values of mobile image using OpenCV

Expanding on our practical analysis, we processed an image of a computer, depicted in Figure 10, through our blob detection program to identify different blobs. The output, shown in Figure 10, illustrates the computer's various blob features. These include bounding box dimensions, coordinates, inter-blob distances, area ratios, central positions, densities, and available space within each blob. These metrics form the computer's data profile in our database. We then use this data to compute the probability of the computer's match within our comprehensive object database.

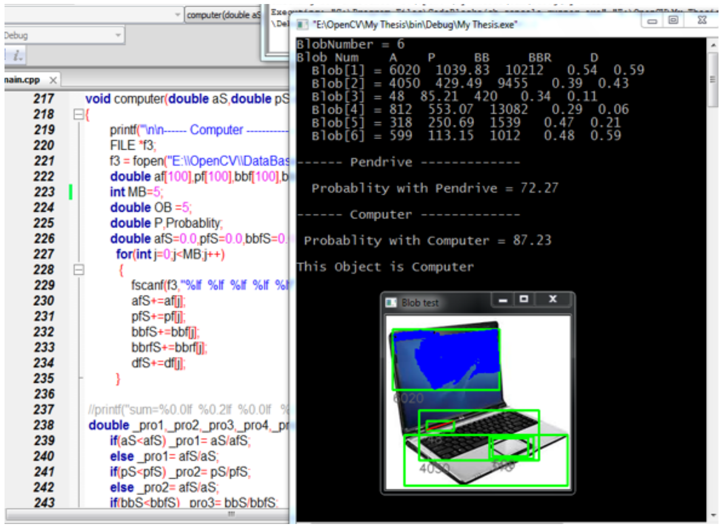


Figure 10. Finding various blobs and values of computer images using OpenCV

5. Performance Analysis and Discussions

5.1 Data Collections and Formulations

We have compiled a database for all objects, storing each object's blob values in separate files. We gathered all blob values from the objects and compared them with the values in the database for each corresponding object. This process has enabled us to create a comprehensive dataset.

We have used the following formulas to calculate the probability.

Probability of Area:

Total area of database object A1= $\sum_{i=1}^{i=no.\ of\ blobs} ai$

Total area of input object A2= $\sum_{i=1}^{i=no.\ of\ blobs} ai$

IF (A1<A2) THEN A=(A1/A2)

ELSE A=(A2/A1)

Probability of Perimeter:

Total perimeter of database object P1= $\sum_{i=1}^{i=no.\ of\ blobs} pi$

Total perimeter of input object P2= $\sum_{i=1}^{i=no.\ of\ blobs} pi$

IF (P1<P2) THEN P=(P1/P2)

ELSE P=(P2/P1)

Probability of Bounding Box:

Total perimeter of database object B1= $\sum_{i=1}^{i=no.\ of\ blobs} bi$

Total perimeter of input object B2= $\sum_{i=1}^{i=no.\ of\ blobs} bi$

IF (B1<B2) THEN B=(B1/B2)

ELSE B=(B2/B1)

Probability of Bounding Box Ratio:

Total perimeter of database object BR1= $\sum_{i=1}^{i=no.\ of\ blobs} bri$

Total perimeter of input object BR2= $\sum_{i=1}^{i=no.\ of\ blobs} bri$

IF (BR1<BR2) THEN BR=(BR1/BR2)

ELSE BR=(BR2/BR1)

Probability of Density:

Total perimeter of database object D1= $\sum_{i=1}^{i=no.\ of\ blobs} di$

Total perimeter of input object D2= $\sum_{i=1}^{i=no.\ of\ blobs} di$

IF (D1<D2) THEN D=(D1/D2)

ELSE D=(D2/D1)

Probability = $(\frac{A+P+B+BR+D}{N})$ *100

Here N is no. of parameter.

5.2 Results and Discussions

We analyzed the performance of our model by comparing the image of a pen, a computer, and a mobile. The results can be seen in Table 1 to 3.

Table 1. Probability between mobile and pen

Parameter	Object	Total	Probability

Area (A)	mobile	7745	0.14
	pen	1099	
Perimeter (P)	mobile	3797	0.14
	pen	546	
Bounding Box (B)	mobile	13958	0.38
	pen	5337	
Bounding Box ratio (BR)	mobile	131.808	0.03
	pen	4.148	
Density (D)	mobile	17.786	0.05
	pen	0.881	

Table 1 displays the probability calculations comparing features such as area, perimeter, and density between a mobile phone and a pen. Using the formula, the probability is computed as the average of these comparative values, resulting in 14.2%. This low probability indicates that the input image does not match the mobile phone profile in the database, leading to the conclusion that the input image is not that of a mobile.

Table 2. Probability between mobile and computer

Parameter	**Object**	**Total**	**Probability**
Area (A)	mobile	7745	0.65
	computer	11847	
Perimeter (P)	mobile	3797	0.65
	computer	2469	
Bounding Box (B)	mobile	13958	0.39
	computer	35720	
Bounding Box ratio (BR)	mobile	131.808	0.02
	computer	2.518	
Density (D)	mobile	17.786	0.11
	computer	1.992	

Similarly, this low probability (36.4%) found in Table 2 indicates that the input image does not match the mobile phone profile in the database, leading to the conclusion that the input image is not that of a mobile.

Table 3. Probability between mobile and mobile

Parameter	**Object**	**Total**	**Probability**
Area (A)	mobile	7745	0.98
	mobile	7912	
Perimeter (P)	mobile	3797	0.95
	mobile	3632	
Bounding Box (B)	mobile	13958	0.96
	mobile	14502	
Bounding Box ratio (BR)	mobile	131.808	0.94
	mobile	140.025	
Density (D)	mobile	17.786	0.91
	mobile	16.268	

Incorporating the probability calculation for an input image against the mobile profile in our database, we derive a probability of 95% for a mobile (see Table 3). Since this value exceeds the 80% threshold, we can confirm that the input image is indeed that of a mobile phone. This high probability demonstrates the effectiveness of our method in accurately identifying objects by comparing extracted feature values against a predefined database.

We perform a linear comparison by matching the input image's values with the values of all images in the database, resulting in a linear time complexity for the calculation method, which is denoted as O(n) time. This denotes that the time taken for the comparison is directly proportional to the number of images in the database.

6. Conclusion

The implications of this study are multifaceted and far-reaching. By successfully demonstrating a probabilistic machine learning approach to visual object detection, this research can significantly influence the development of autonomous systems capable of complex visual interpretation. The methodology's accuracy in identifying objects through features like area, perimeter, and density paves the way for more intuitive and interactive robotics, particularly in fields requiring precision such as manufacturing and quality control. Furthermore, the application in optical character recognition and image indexing can revolutionize how machines process and organize visual data, offering improvements in areas such as digital archiving, surveillance, and user-interface design. The study's linear comparison approach, with its O(n) time complexity, offers a scalable solution for real-time object detection, which is critical for systems requiring immediate response and decision-making capabilities.

This study presents a significant leap forward in the field of computer vision, particularly in the domain of object detection. By introducing a probabilistic approach that leverages feature extraction from partial objects, this research moves towards closing the gap between current object detection capabilities and the nuanced requirements of real-world applications. The developed model not only detects but also learns from various object features, thereby enhancing the precision and adaptability of robotic vision systems. Through this work, we have taken a step closer to replicating human-like perception in machines, demonstrating potential applications in space exploration, public security, and beyond. The future of autonomous systems looks promising, with such advancements heralding a new era of intelligent, perceptive robotics equipped to handle the complexities of their environments.

References

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L., Imagenet large scale visual recognition challenge. International journal of computer vision, 115, 211-252, 2015.

Rajesh Kumar, G., Srinivasa Rao, D., Rajasekhar, N., Ramesh Babu, C., Renuka, K., Koteswara Rao, M., ... & Mangathayaru, N., Object Detection Using Deep Learning Approaches. In International Conference on Intelligent Computing and Communication (pp. 717-727). Singapore: Springer Nature Singapore, November 2022.

Salari, A., Djavadifar, A., Liu, X., & Najjaran, H., Object recognition datasets and challenges: A review. Neurocomputing, 495, 129-152, 2022.

Shinde, A. K., & Shukla, M. Y., Crop detection by machine vision for weed management. International Journal of Advances in Engineering & Technology, 7(3), 818, 2014.

Koch, C., & Ullman, S., Shifts in selective visual attention: towards the underlying neural circuitry. Human neurobiology, 4(4), 219-227, 1985.

Hu, Y., Tao, V., & Croitoru, A., Understanding the rational function model: methods and applications. International archives of photogrammetry and remote sensing, 20(6), 119-124, 2004.

Privitera, C. M., & Stark, L. W., Algorithms for defining visual regions-of-interest: Comparison with eye fixations. IEEE Transactions on pattern analysis and machine intelligence, 22(9), 970-982, 2000.

Itti, L., Koch, C., & Niebur, E., A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on pattern analysis and machine intelligence, 20(11), 1254-1259, 1998.

Itti, L., & Koch, C., Feature combination strategies for saliency-based visual attention systems. Journal of Electronic imaging, 10(1), 161-169, 2001.

Tagare, H. D., Toyama, K., & Wang, J. G., A maximum-likelihood strategy for directing attention during visual search. IEEE Transactions on pattern analysis and machine intelligence, 23(5), 490-500, 2001.

Ude, A., Atkeson, C. G., & Cheng, G., Combining peripheral and foveal humanoid vision to detect, pursue, recognize and act. In Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No. 03CH37453) (Vol. 3, pp. 2173-2178). IEEE, October 2003.

Bjorkman, M., & Kragic, D., Combination of foveal and peripheral vision for object recognition and pose estimation. In IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 (Vol. 5, pp. 5135-5140). IEEE, April 2004.

Orabona, F., Metta, G., & Sandini, G., Object-based visual attention: a model for a behaving robot. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops (pp. 89-89). IEEE, September 2005.

Li, W., Liu, K., Zhang, L., & Cheng, F., Object detection based on an adaptive attention mechanism. Scientific Reports, 10(1), 11307, 2020.

Lu, K., Zhao, F., Xu, X., & Zhang, Y., An object detection algorithm combining self-attention and YOLOv4 in traffic scene. PLoS one, 18(5), e0285654, 2023.

Biographies

Md Kamrul Hasan Chowdhury completed his bachelor's in mechanical engineering at the Rajshahi University of Engineering and Technology, Bangladesh. He has several years of industry experience as an engineer. His research interests include Industrial Management, Industrial Automation, and Environmental Sustainability.

Basudeb Biswas completed his bachelor's in mechanical engineering at the Rajshahi University of Engineering and Technology, Bangladesh. He has several years of industry experience as an engineer. His research interests include Industrial Management, Industrial Automation, and Environmental Sustainability.

Spandan Basak Payel completed his Bachelor of Mechanical Engineering at the Bangladesh University of Engineering and Technology He has several years of industry experience as an engineer. His research interests include Industrial Management, Industrial Automation, Decision Science, Optimization, and Environmental Sustainability.

Md. Tanvir Siraj completed his Bachelor of Mechanical Engineering at the Bangladesh University of Engineering and Technology He has several years of industry experience as an engineer. His research interests include Industrial Management, Industrial Automation, Decision Science, Optimization, and Environmental Sustainability.

Debashis Biswas completed his bachelor's in electrical engineering at the National Institute of Technology, India. He has several years of RMG industry experience as an engineer. His research interests include Industrial Management, Industrial Automation, and Environmental Sustainability.