Object Tracking by Using Pattern Matching in LabVIEW

[1]Dr.Ravikumar A V, [2]Madhusudhan K V, [3]Guruprasad G, [4]Hareen N R,[5]Gagan H
[1]Associate Professor, [2345]Student
[1]Electronics and Communication,
[1]SJB Institute of Technology, Bengaluru, India

Abstract— **Tracking in real time is one of the major issue or task, which can be perform for variety of purposes like in surveillance for monitoring any kind of dynamic scenes for safety purposes. So here, in proposed system a object tracking system is developed using the NI vision development module [1] for the purpose of surveillance. The proposed system utilizes a inbuilt-camera to capture images of the object, which are then processed using pattern matching algorithms to identify and track the object's movement. The system's performance can be evaluated in terms of accuracy, speed, and robustness. The results demonstrate the effectiveness of the proposed system in tracking objects in real-time, with potential applications in robotics, surveillance, and automation. In the proposed system, we use the National Instruments, LabVIEW programming which is the main part of the proposed system. Here a software related to NI i.e., Vision Acquisition is taken from the video or image processing techniques and it performs various tasks like Set image conversion of the image data and grabbing of the video or modifying the resolution.**

Index Terms— Object Tracking, Image Acquisition, Region of Interest, Pattern Matching, Feature Extraction

I. INTRODUCTION

This project leverages LabVIEW to implement a real-time object tracking system for surveillance applications. The system captures a live video feed and processes individual frames to detect and track a Region of Interest (ROI). The primary objective is to enhance security by providing real-time visualization of tracked objects within a monitored area. The workflow begins with video acquisition using LabVIEW's Vision Acquisition tools, which extract frames from the live feed. Image processing techniques, such as color segmentation, edge detection, or template matching [3], are applied to identify the target object or ROI in the frame. Once identified, a tracking algorithm (e.g., centroid-based tracking or optical flow) is used to follow the ROI across subsequent frames.

To provide visual feedback, the detected ROI is overlay onto the processed frame in real time. This overlay highlights the tracked object and allows operators to monitor its movement effectively. The system is design to be scalable, enabling customization based on the specific requirements of the surveillance environment, such as tracking multiple objects or integrating alerts for unauthorized movements.

This project demonstrates the effectiveness of LabVIEW in building interactive, real-time video processing applications, displaying its potential in enhancing security systems.

II. MOTIVATION

The growing demand for automated monitoring and control systems drives the need for efficient object tracking solutions. **LabVIEW's** graphical programming and real-time processing capabilities offer a promoting alternative.. This project aims to leverage LabVIEW's strengths to develop an accurate and scalable object tracking system using **Pattern Matching** [3].To develop a tracking system that can enhance surveillance, robotics, automation, transportation and healthcare applications.. Effective object tracking can improve safety, efficiency and productivity. This project's outcome has significant potential for real-world impact.

III. OBJECTIVES

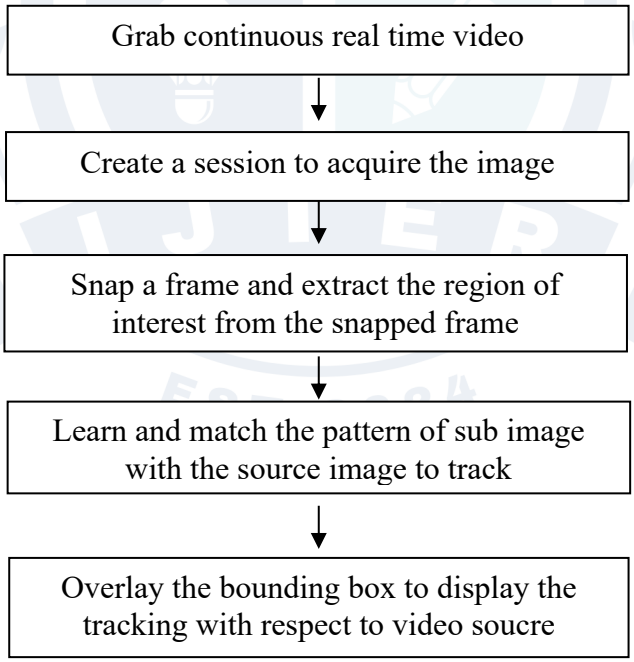
- Acquire the video scene by using machine vision LabVIEW tool.
- To perform morphological process by using LabVIEW vision tool.
- To determine and tracking of the object by using Pattern Matching algorithm.
- To extract the position of the tracked object.

IV. HARDWARE & SOFTWARE REQUIREMENTS

We have used the Inbuilt Camera System available in our PC .We can use the above system to be deployed using Webcam which has a support for FPGA or different digital Camera that can be interfaced with the system.

LabVIEW: LabVIEW 2023, developed by NI (National Instruments), is a programming environment used for designing, testing, and automating engineering and scientific systems. It offers an intuitive drag-and-drop interface, allowing users to create complex applications visually using function blocks and wiring. Key features include enhanced compatibility with modern hardware, improved data analysis tools, and integration with Python and MATLAB. LabVIEW 2023 supports advanced features for real-time data acquisition, image processing, and IoT applications, making it ideal for automation, control systems, and research. Its robust libraries and versatile tools streamline development for engineers and researchers across various industries.

V. BLOCK DIAGRAM OF PROPOSED SYSTEM



VI. IMAQ VISION ACQUISITION SOFTWARE

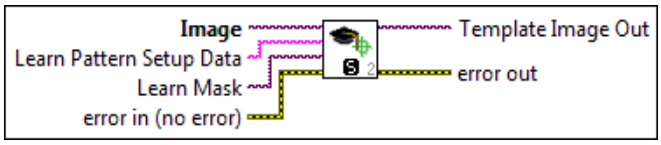
IMAQ Vision Acquisition Software, part of LabVIEW's Vision Development Module, is designed for high-performance image acquisition and processing [1]. It enables seamless integration with cameras and imaging hardware, supporting various standards like USB, GigE, and Camera Link. The software allows users to capture, display, and process images in real-time for applications such as object tracking, pattern matching, and edge detection. With its extensive library of vision functions and intuitive interface, IMAQ simplifies the

development of imaging systems for machine vision, automation, and surveillance. It is a powerful tool for engineers and researchers working on image-based projects.

A. Vision development module

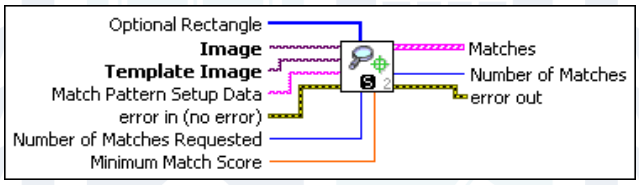
In we Vision development module we have different pallets for Search and Matching, Extract image and region of interest, Machine vision and our focus is on object tracking.

B. Learn pattern2 VI



This function generates a description of the template image to be used during the matching phase of pattern matching. The descriptor data is appended to the input template image. During the matching phase, the template descriptor is extracted from the template image and is used to locate the template within the inspection image

C. Match pattern2 VI



Searches for a pattern, or Template image in an inspection image. Execute the IMAQ Learn Pattern V2 VI before this VI to configure the template image.

VII. METHODOLOGY

Pattern matching is the process of finding the location of a sub image, called a template image. There are numerous methods involved to find identical images and its features. This section illustrate the Pattern matching algorithm for matching a sub image, which is a part of a larger image. Once the number of matching templates are found, their image center magnitude can be used as the corresponding control points to determine the matching attributes. Template matching involves comparing a given template image with windows of the same size in an image and recognizing the window that is most similar to the template [3].

Normalized Cross Correlation (NCC) [3] is the basic numerical approach to match sub images in a larger image. NCC is used for pattern matching or template recognition. A template can be considered from a reference image, and an image from a video can be considered as a source image. The objective is to establish the strong matches between the reference and source images.

We have four methods used in Pattern Matching Methods:
1) Sum of Squared Difference (SSD)
2) Normalized Sum of Squared Difference (NSSD)
3) Cross Correlation
4) Normalized Cross Correlation

The objective of the proposed method is to utilize the pattern-matching algorithm and increase accurate matching in LabVIEW. As a similarity measure, the SSD is the most popular and effectively used for matching applications. NCC is more robust against illumination changes than SSD nevertheless, NCC has delay than SSD and the equations for respective correlation is mentioned below.

1) Sum of Squared Differences:

$$R(x,y) = \sum_{u}^{v} \quad \big(T(u,v) - I(x+u, y+v)\big)^2$$

2) Normalized Cross Correlation:

$$R(x,y) = \frac{\sum_{u}^{v} \quad (T(u,v).I(x+u, y+v))}{\sum_{u}^{v} \quad T(u,v)^2 . \sum_{u}^{v} \quad I(x+u, y+v)^2}$$

The following three key steps are involved in implementing of the proposed method.

- Detection of reference image.
- Updating of suitable template.
- Tracking of required image from frame to frame in a video stream.

A. Matching Procedure

To identify a matching image, the method needs to compare a template image against a source image by moving or sliding through source image. By sliding the template image, the process can measure the similarity between the template image and a region in the video source image. At each location, a center metric is calculated so it represents how similar the sub image is to that particular area of the scene image. The process then finds the maximum or the minimum location in the resulting image depending on the measurement of given input image and video source.

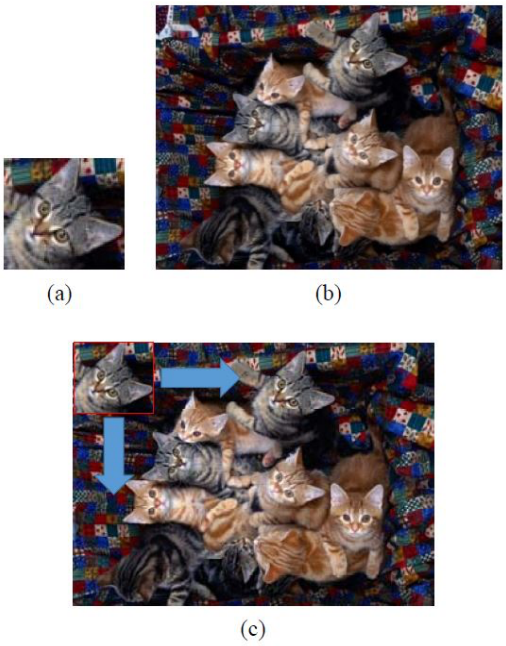


Fig 1. Example of pattern matching procedure

VIII. IMPLEMENTATION

A. Front Panel of Proposed System

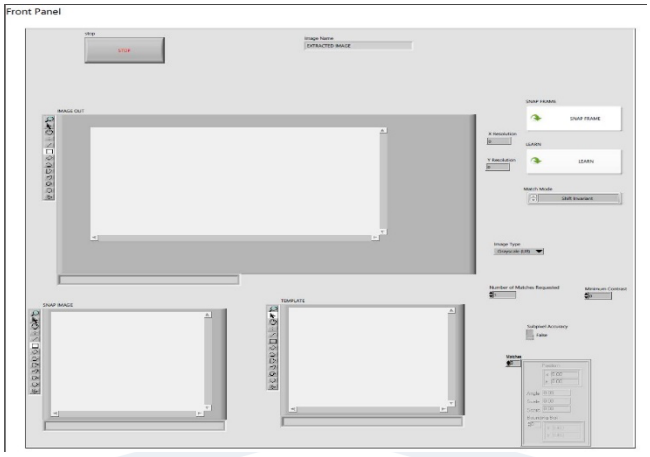


Fig 2. Front panel of Proposed system

B. Grab continuous video from camera source

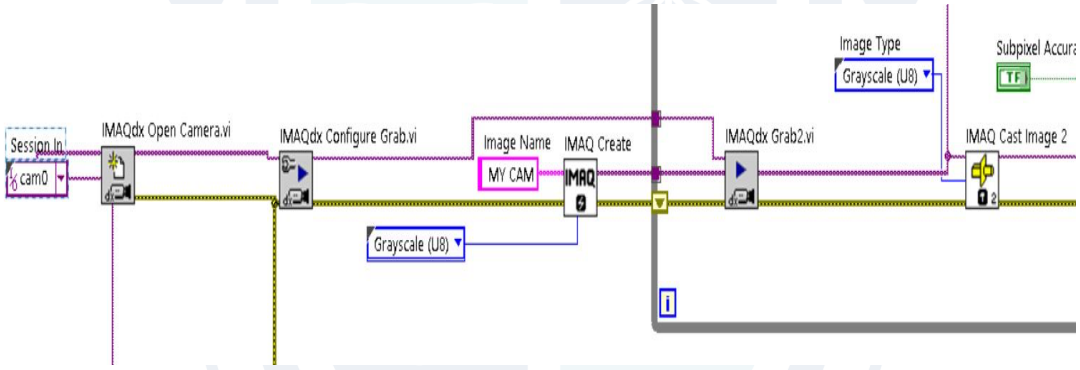


Fig 3. Block Diagram to grab continuous video source

- We have used started the Inbuilt PC Camera Session.
- It is given as input to the open camera vi to open the camera.
- Then we initialize the configure grab to start the real time video acquisition.
- We have used IMAQ create to provide a temporary memory for image. IMAQ Grab VI2 is used for real time video acquisition.

C. Snapping a frame from video source and extracting a subframe from snapped frame

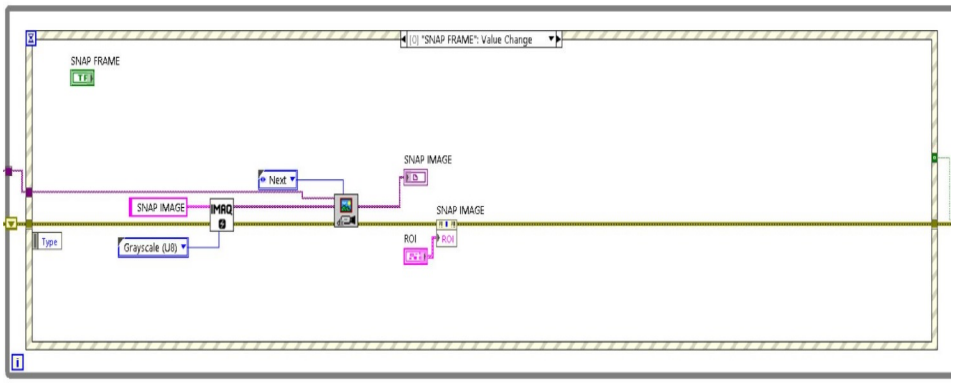


Fig 4. Block diagram to snap a frame from a continuous video source

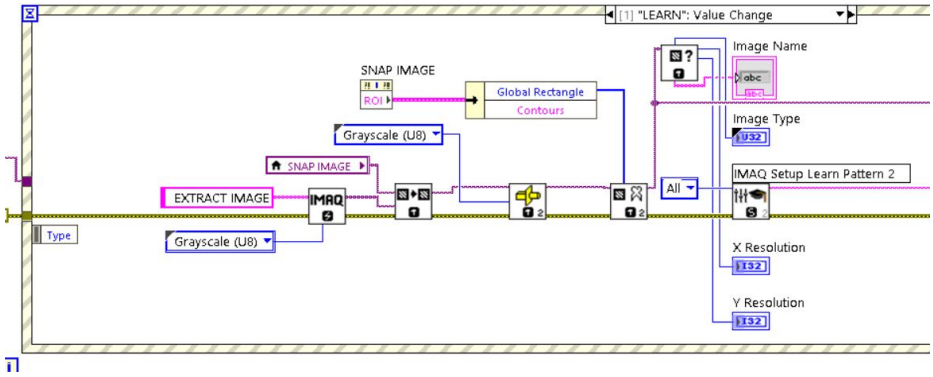


Fig 5. Block Diagram to snap and extract a frame from source image

- We have used while loop to continuously snap the frames of video.
- We have used an event structure to execute snapping a frame and extracting the portion of image from snapped frame as different events.
- We have used IMAQ get image to gets frame of image and display the same.
- We have created a property node to draw ROI around the snapped frame and provide the same as input to optional rectangle of IMAQ extract
- Firstly we create a local variable containing snapped image and its data and its given as input to the IMAQ copy to have a copy of snapped image and its grayscale using IMAQ cast image which is in turn given as input to the IMAQ Extract.
- The ROI drawn in the snapped frame is collected in the form of size of Global rectangle and its related contour, which is stored in the form of property node.
- It is given as input to optional rectangle node of IMAQ extract VI to extract the selected portion of snapped image.
- We can draw Global Rectangle or any contours such as (circle, oval) to define the pattern to be tracked
- The output image of IMAQ Extract VI is given as input to the IMAQ image info VI to know the resolution of image selected as pattern.
- IMAQ Get info VI gives the data of resolution of the image along with the image type.

D. Learn and match the frame in accordance with real time video

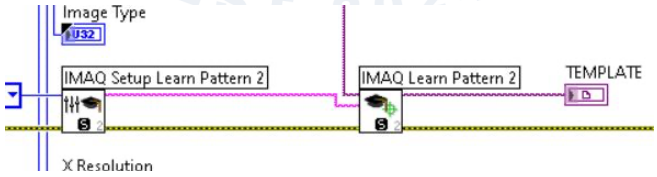


Fig 6. Block diagram to learn the pattern of extracted frame.

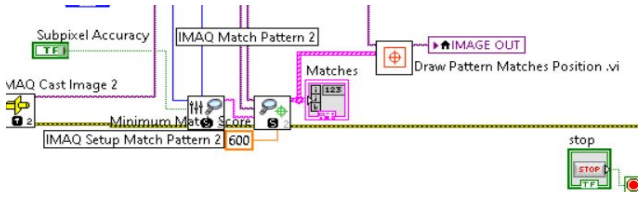


Fig 7. Block diagram to match the pattern of the extracted image.

- Extracted image is provided as input to the IMAQ Learn Pattern 2 where the pixel and contour data is read as per the mechanism of Learn pattern algorithm.
- Before that, we should configure the algorithm to start with Learn pattern by initializing the IMAQ Setup Learn Pattern 2.
- The pattern of the image is processed by the final VI is displayed as template image.
- This Template image is given as input to match the pattern with the real time video.
- We use the template image and give the same as input to the Match Pattern V2 to match the pattern in accordance with same pattern appearing in the real time video sequence.
- Before that, we should use the Setup Match Pattern VI to configure and Initialize the Match Pattern V2.
- The Match pattern vi should be provided with the input of real time video as reference to match the pattern
- Finally, we obtain the output of the Match Pattern VI as matches, which is a contour data, and the same is given as input to the Draw Pattern matching sub VI.
- To draw the overlay diagram of selected ROI we use the draw pattern matching VI.

E. Overlaying the bounding box at the time of real time video display.

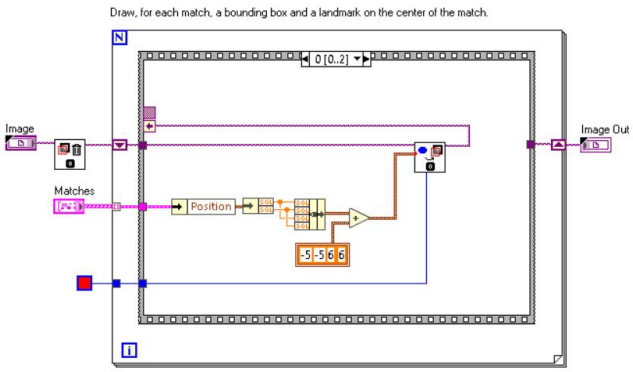


Fig 8. Block diagram to overlay the bounding box

- Finally, we use the last VI as Sub VI to draw the bounding box around the image, which is matched by the Match pattern VI.
- The challenging task is to draw the bounding box around the image after each shift and iteration in the real time video.
- Firstly after getting the matched data of the pattern image we use as data of array containing the contour data.
- The contour data is in the form of array, which has to be given as single element data.
- So we use Unbundle by element cluster element the divided the array elements.
- Here we have given the array of matched data to position element, which is reference in LabVIEW.
- This reference is use to define the new position of the image in real time video.
- IMAQ Overlay VI is used to overlay the bounding box drawn based the ROI data and match the same image.

IX. RESULTS & IMPLEMENTATION

This project demonstrates object tracking using pattern matching in LabVIEW. By utilizing the Vision Development Module, a predefined template is matched with real-time video feed to detect and track objects. The system processes images, identifies the target pattern, and continuously updates its position in the frame. Implementation includes setting up the camera, configuring pattern matching parameters, and integrating feedback for precise tracking. The result displays efficient and accurate object detection under varying conditions, proving the system's robustness and suitability for industrial automation, surveillance, and robotics applications.

- Firstly, we obtain the continuous video steam and display the same using Image display of vision acquisition module.

Fig 9. Continuous video capture using inbuilt pc camera

- Next, we snap the frame from the continuous video stream by clicking the snap frame button.

Fig 10.Snapped frame from video source

- Next, draw the ROI around the image and extract the frame.
- Once we draw, the ROI and click extract the frame we obtain the image on template image display.

Fig 11. ROI drawn to the snapped image

- We obtain the extracted portion of image in template image display after clicking the learn frame.

Fig 12.Template image obtained after extracting the desired frame.

- Next, we can see the processing of pattern matching algorithm and bounding box diagram on the matched image in image display.

A. 1st Test Result

Fig 13.1st instance of tracking image

B. Tracking of the object

Fig 14. Tracking of static object

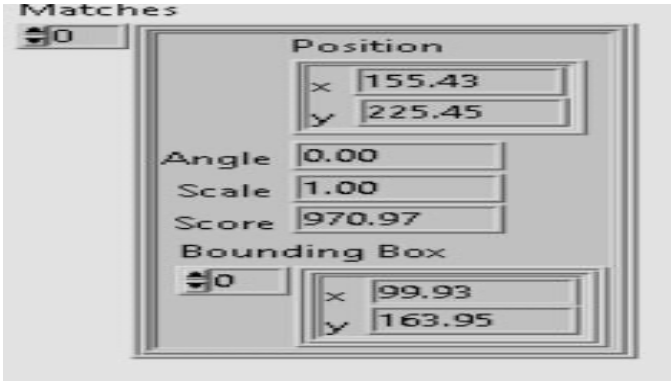


Fig 15. Tracking results of the static object

C. Tracking of multiple object occurrences

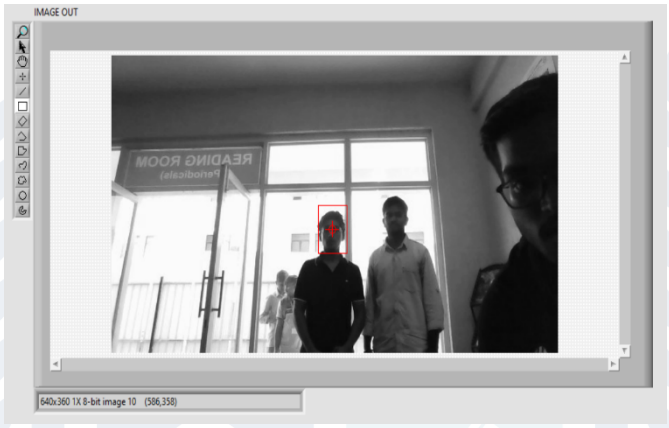
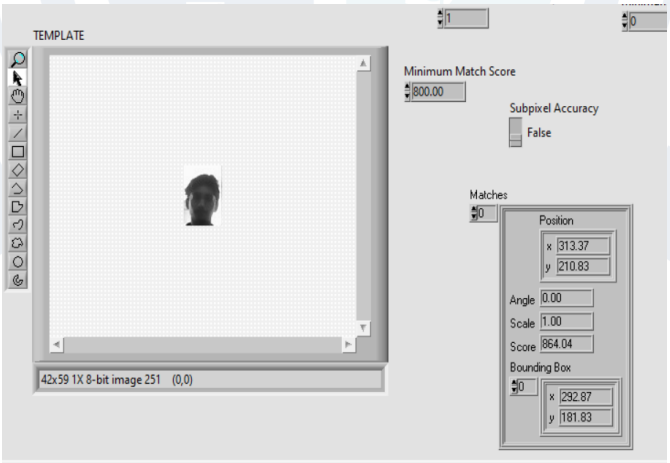


Fig 16. Tracking results of multiple facial occurrences



.

Fig 17. Tracking score of the above image.

D. Tracking of letters and structures in an image.

Fig 18. Tracking of letters in an image and display the same on video source

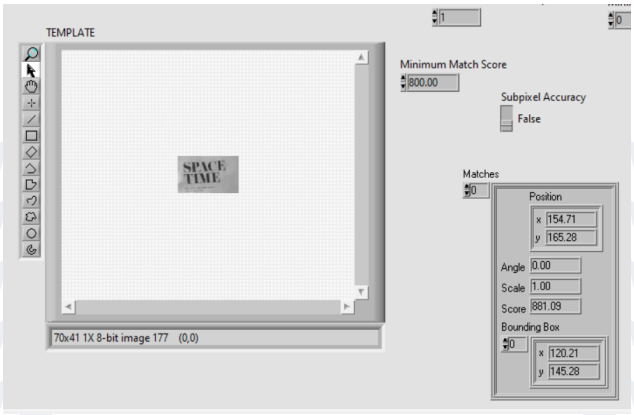


Fig 19. Tracking score of the above image.

E. Tracking of structure imbibed in an image.

Fig 20.Tracking of the structure in a book.

F. Tracking of a vehicle number plate.

Fig 21.Tracking of the vehicle number plate

G. Accuracy of matched results

In Each picture of the above matched results, we have table named "Matches" which gives the information about the X-axis and Y-axis data that provides the information about the position of the object in the kernel.

We have another parameter named Score, which suggests the accuracy of the object during the matching process. The minimum score, which is default, value while matching is 800. We can increase or decrease the score based on our desired results. We can use the score to calculate the accuracy of the matching process.

For our convenience to obtain the accuracy under 150% we have used the score value as 900 but the set score in the VI is 800.

To calculate the accuracy our formula is:

Accuracy=Obtained Score/Total Score * 100

TABLE 1. Data of accuracy of matched results

Result	Score	Accuracy/Percentage
1st Test Result	670	74%
2nd Test Result of Next Frame	897	99.6%
Tracking Image of an Object	970	107%
Tracking of Images or structures during multiple Instances	864	96%
Tracking of Structure on an Object	860.19	95.57%
Tracking of Facial Structures during multiple occurrences	821.50	91.27%
Tracking of objects or structures having Different Patterns, letters on a paper, letters on a Book	881.09 886.46	97.89% 98.49%

We have plotted the results for easy visualization of the data we have used in our tabular columns. The y-axis represents the combination of Matched Score and Accuracy. The score is represented along the bar chart and accuracy along the line plot.

Additional Information on ROI Descriptor: In LabVIEW, a **Region of Interest (ROI) Descriptor** is a critical tool used in image processing to define specific areas within an image for focused analysis. The ROI descriptor specifies the shape, size, and location of the region to be analyzed, such as rectangles, circles, polygons, or freehand areas. This is particularly useful in object tracking, pattern matching, and feature extraction.

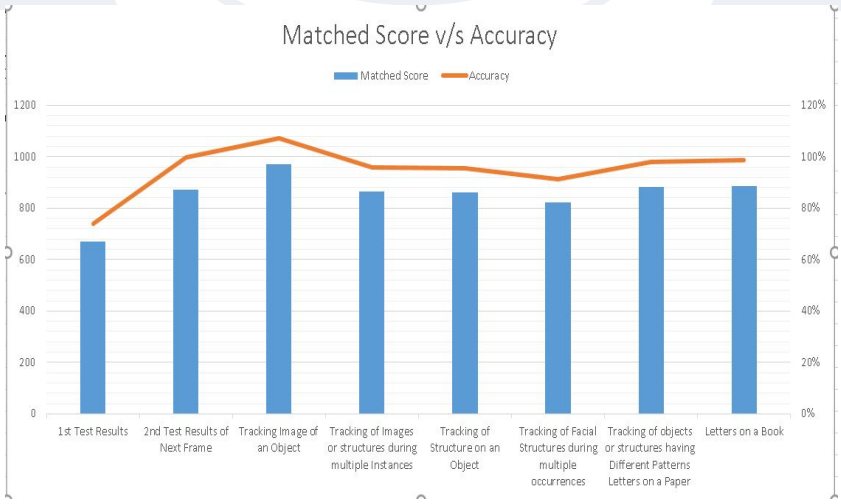


Fig 22. Graphical representation of matched score v/s accuracy

The ROI descriptor allows users to isolate parts of an image, minimizing computational effort and improving efficiency. It is often used with LabVIEW's Vision Development Module for tasks like measuring object size, detecting edges, or applying filters to specific areas. ROI descriptors can be created manually by drawing regions on the image or programmatically by specifying coordinates. This flexibility makes it a powerful tool for customizing image analysis workflows, ensuring accurate and efficient processing for a wide range of machine vision applications.

X. FUTURE SCOPE

We are in the era of computer vision and based systems, and is used for many applications. As a team, we have proposed the system as mini project, which can be further developed using color tracking virtual instruments using LabVIEW. By learning the image processing abilities of LabVIEW along with nourishing the skills of Hardware interfacing using LabVIEW such as "*NI myRIO module*" we develop it as object tracking system for highly Advanced Camera system where we can integrate the movement of Camera using Tilt Pan motor or servo motor controls based on the tracking of object. Such kind of systems are highly necessary in military surveillance and tracking purposes. We have different kind of filters to enhance the accuracy of the tracking such as kalman Filter, Particle filter to track the high motion target objects. Here we have proposed the object tracking system for single object we can integrate the system for multiple object using Harr features.

XI. CONCLUSION

"We conclude that after conducting a survey through information source available decided the project to be undertaken as part of our learning and explore the Image processing capabilities of LabVIEW and computer vision. Object tracking can be observed in different scenarios such as Traffic monitoring systems, Digital Cameras, Surveillance Systems that motivated us to develop a portion of the system, which is highly applicable in different Imaging domains. The challenging part is to draw the bounding box around the image in a real time video sequence and display the same. Hence, we have resolved it by creating a sub VI and making the system more efficient. Our next goal is to develop the system for highly advanced camera integrated with movement of object and develop the same using LabVIEW and learn hardware interfacing capabilities".

XXI. REFERENCES

[1] Divya .M., Dr.Ravikumar.A.V., "Single Object Tracking System by LabVIEW", IJRITCC, vol.4, no.5, pp.52-56, May 2016.

[2] Dr. Ravikumar.A.V., Dr. Nataraj.K.R., Dr.Rekha.K.R., "Morphological Real time Video Edge Detection in LabVIEW"., IJCSIT, vol.3, no.2, 2012.

[3] W. Chantara ., Ji-Hun Mun., Dong-Won Shin., "Object Tracking using Adaptive Template Matching.", IEIE Transactions on Smart Processing and Computing, vol.4, no.1, February 2015.

[4] M.S. Gururaj., M.H.Ramesh., J.A.Arvind., "A Review on Image Tracking Technique in LabVIEW", IJSDR, vol.1, Issue 6, June 2016.

[5] Sachin Gupta., Manoj Shukla., "Detection and Identification of object color using LabVIEW based Machine vision for vehicular Robots.", JETIR, vol.6, Issue 3, March 2019.

[6] L.Abdelhai., M.Haitham., B.Belkhier., "Novel LabVIEW Implementation of Freak Based Image Mosaicing and Object Tracking Algorithms.", ICEE, Istanbul, Turkey, pp. 1-6, 2020.