

Descent Angle Calculation for UAVs using Monocular Camera

Dheeraj Komandur
Student

School of Computer
Engineering and Technology,
MIT World Peace University
Pune, India

Sagar Karki
Student

School of Computer
Engineering and Technology,
MIT World Peace University
Pune, India

Shebin Silvester
Student

School of Computer
Engineering and Technology,
MIT World Peace University
Pune, India

Gajendra Kashyap

Project Owner
CTO, NextLeap Aeronautics,
Bengaluru

ABSTRACT

In this paper, an approach for solid angle calculation for autonomous landing of UAVs/AVs using a monocular camera and 2D image of the helipad is introduced. Autonomous landing has always been a prime research field that is complex and unique in the field of automation. The proposed method that uses object detection and homography to generate relative flight angles - Yaw, Pitch, and Roll. First, an object detection algorithm is used to locate the landing pad in an image and extract the required region of interest (ROI). This ROI is then preprocessed using OpenCV and key corners of the landing pad are extracted. Further, these extracted coordinates are mapped to a reference image and relative homography is calculated. Decomposing the homography matrix results in Euler angles which give the angles of UAVs/AVs with respect to the landing pad.

General Terms

Object Detection, YOLO, Homography.

Keywords

Autonomous Landing, UAVs

1. INTRODUCTION

Autonomous landing has always been a complex and fascinating area of research in the AI(Artificial Intelligence) industry as well as in the Aviation industry. The complexity and limited computation power (in the case of lightweight UAVs) make the problem of autonomous landing more challenging and unique. This paper proposes a solution for autonomous landing using object detection and mathematical tools for real-time autonomous landing. The significant contribution of this paper is a novel framework for relative position estimation of the aerial vehicle from the identified helipad. The proposed method finds the angles of the aerial vehicle with respect to the stationary or moving landing pad using the 2D data.

The Object detection model trained on helipad images in all possible angles is used to detect the landing region and using different image processing techniques the angles and the relative positions are calculated.

2. RELATED WORKS

With the recent trends in autonomous landing, most of the

autonomous landing use aid of other parts of the aerial vehicle like the infrared vision and lidar for relative position estimation for the landing region. The method used by O. A. Yakimenko et al [1] addresses the problem of determining the relative position and orientation of an unmanned aerial vehicle with respect to a ship using three visible points of known separation. The images of the points are obtained from an onboard infrared camera. If the landing is in any form dependent on such methods, it will cause delays and may even lead to disastrous situations for the inability to land due to the failure of specific aiding systems. Although the experiments are performed using infrared vision on less amount of images using the segmentation method miss out on one image which may prove to be a problem when dealing in the real world. The method although impressive did not consider the possibilities of occlusion and the result of it in the experiment.

G.Anitha et al [2] proposed a method in which the position estimation of the vehicle is determined with an uncalibrated camera. The additional speedometer is also used as it is necessary for a safe touch down. Similar experiments done by J. Janousek et al [3] indicate the use of custom landing target design and significant use of infrared beams for target analysis by onboard computers. Pose estimation on the helipad is also a commonly used technique for identifying the helipad, rotation, and scaling. A similar methodology was presented by Sanchez-Lopez et al [4] where the principal component analysis (PCA) applied to the first seven Hu moments are performed for each blob contained in the image. Afterwards, the information obtained from the PCA processing is passed as input to an artificial neural network and it provides the classification of blobs.

In another method implemented by Premachandra C [5], the helipad detection is proposed using the camera and ultrasonic sensor. The data was processed on an onboard raspberry pi. Similarly, a monocular vision-based real-time target recognition was proposed by Lin S et al [6] in which the helipad contours were used to perform the point feature mapping and clustering. The algorithm is able to recognize the international landing target in a cluttered environment providing the 4 degrees of freedom pose of the drone with respect to the helipad.

A machine-learning based method was presented by Oliveira et al [7] to automatically detect two types of landing pads.

These pads consisted of the traditional characters “H” and “X” inside circles. Six classifiers based on Haar-like features and local binary patterns (LBPs) were trained to recognize the artificial helipad marks. The result showed that Haar is slower than LBP but more accurate. In another research done by Arora et al [8], researchers address the need for an infrastructure-free system for landing deck pose estimation. They presented a solution that uses complementary modalities of camera and lidar independently to track the ship deck.

3. MATERIALS AND METHODS

This paper proposes a technique which involves YOLOv4 [9] for detecting and localizing the helipad from the 2D images obtained from the camera mounted on the UAV/AV, which is followed by preprocessing, corner detection, and homography [10] from which the descent angle of aerial UAV/AVs is calculated.

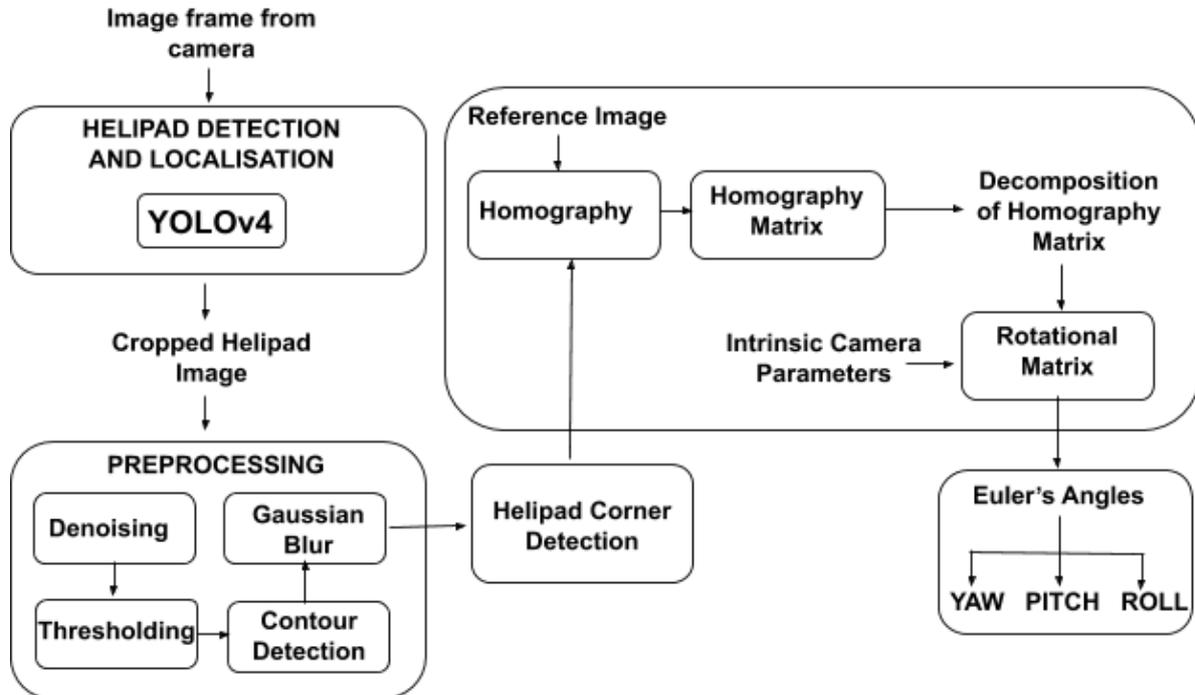


Fig 1: End-to-End System Architecture Diagram

3.1 Object Detection

Yolo is short for ‘You Only Look Once’. It is a single step detection model and is very fast as compared to two-step detectors. For detecting the ‘H’ of the helipad various models were trained. It was noted that YOLOv4 worked best for detection. Due to inbuilt heavy augmentations, while training

provided in YOLOv4, it facilitates training with fewer images feasible and provides better results when trained on a big dataset. The detection model was trained on helipad images with different angles and light conditions. The model is trained for 3000 steps on Nvidia Tesla K80 GPU.

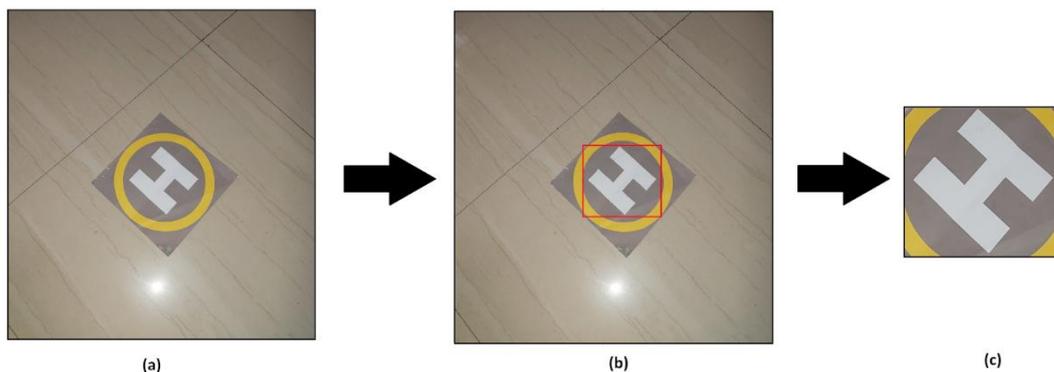


Fig 2: Object Detection of Helipad (a) Input image from the camera (b) YOLO detects helipad (c) Region of interest extracted

3.2 Image Processing

The helipad image obtained from YOLO is preprocessed before applying further image processing techniques. The image is converted to grayscale and the following methods are applied in sequence:

Step 1: Resizing - The extracted region of interest is resized

into a predetermined size which corresponds to the size of the reference image.

Step 2: Grayscale - The resized region of interest is converted to a grayscale image. This step reduces the storage of the file and improves the speed of processing. The image is converted to grayscale in OpenCV which uses the formula mentioned in Figure 3.

$$RGB[A]to\ Gray: Y \leftarrow 0.299.R + 0.587.G + 0.114.B$$

Fig 3: Formula used to convert the RGB image to grayscale image

Step 3: Denoising - The grayscale image is passed through a denoising algorithm to remove the unwanted disturbances in the image. The region of interest which is extracted from an aerial image has unwanted noise which can result in wrong estimations of the corners of the helipad. Further denoising can be followed by gaussian blur to smoothen the image. The cv::fastNlMeansDenoisingMulti [11] function in OpenCV was used because of its efficiency on image sequences where

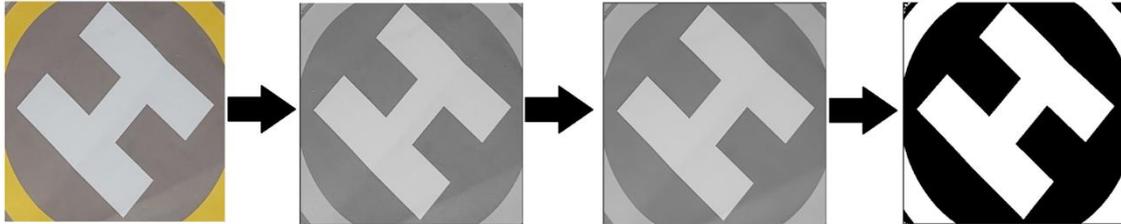


Fig 4: The Four steps of Image Processing

3.3 Corner Detection

After the image is processed, such that the features of the region of interest i.e only the “H” sign is extracted, a corner detection algorithm is implemented.

Step 1: Contour detection - The binary image has one major contour that is a shape “H”. There may be other contour depending upon the extracted region of interest. The method of contour detection introduced by Satoshi Suzuki et al. [13] is used to extract the coordinate points of all possible contours in the image.

Step 2: H contour detection - The list of all contours is passed through cv::approxPolyDP function that approximates a curve or a polygon with another curve/polygon with fewer vertices

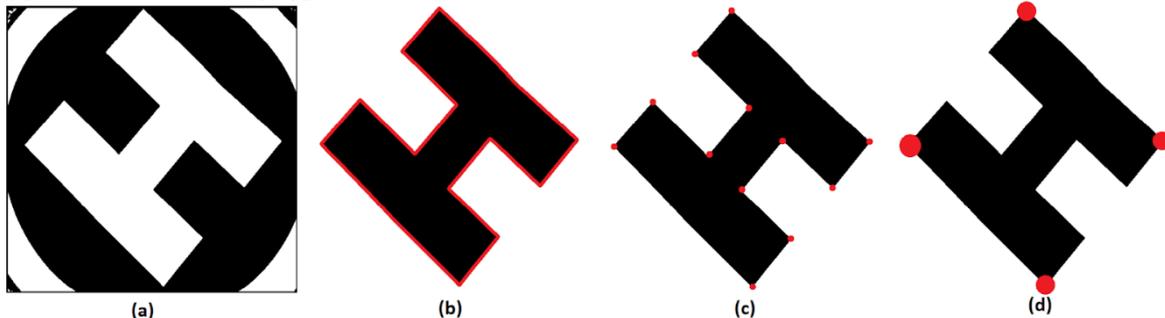


Fig 5: Corner Detection (a) Input image from image processing algorithm (b) Contour Extraction having shape of H (c) 12 corners detected from the contour (d) 4 extreme corners of helipad

3.4 Homography Estimation

A Homography is a transformation (3×3 matrix) that maps the points in one image to the corresponding points in the other image. If in one image a certain point is represented by (x₁, y₁) point and the same point in the second image is represented by (x₂, y₂), then the Homography H maps them as shown in Figure 6.

$$\begin{bmatrix} x1 \\ y1 \\ 1 \end{bmatrix} = H \cdot \begin{bmatrix} x2 \\ y2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x2 \\ y2 \\ 1 \end{bmatrix}$$

Fig 6: Homography matrix relation with respect to points

consecutive images have been captured within a small period.

Step 4: Thresholding - The next objective is to highlight the features of the helipad such that light shadows, color fluctuations, noise, etc can be eliminated and also to separate the foreground from the background. Otsu’s Thresholding [12] method was used in the system because it uses histogram plots and probability to fixate on pixel values that separate the two peaks of the histogram. This process for thresholding is highly efficient because, in most helipads, two contrasting colors mark the background and the helipad sign, for example, white helipad mark on black surface or white helipad mark on green surface.

so that the distance between them is less or equal to the specified precision. It uses the Douglas-Peucker algorithm [14]. This function returns the coordinates of the corners of a particular contour and hence if the function returns a list of 12 coordinates for a particular contour then it can be concluded that the contour belongs to the “H” which has 12 corners.

Step 3: Once the contour which represents the “H” is identified the four extreme corners are selected. This is done by wrapping a rectangle box with a minimum area around the contour and then using the Euclidean distance formula for all twelve edges. The four points that have minimum Euclidean distance from the corners of the wrapped rectangle would be the four extreme corners of the detected “H”.

Once the four extreme corners are detected by the corner detection algorithm, mapping of those points with the points of a reference image is processed. The reference image is the standard image of a helipad that is aligned. The coordinates of four corners of the reference image can be manually selected or can be computed using the corner detection algorithm used of the extracted region of interest.

These four points of the extracted processed region of interest and the reference image must be in the same order. These points are passed through cv::findHomography function which calculates the homography.

This decomposition of the homography matrix results in a

rotational matrix. For decomposing the matrix homography `cv::decomposeHomographyMat` function of OpenCV is used. This function requires a camera matrix [15] as an input. Each camera has a unique camera matrix which can be obtained by calibrating the camera in OpenCV.

3.5 Euler Angle Calculation

Rotation in 3D can be represented as three angles that specify three rotations applied successively to the X, Y, and Z axes. But this could be represented in the same rotation as three angles applied successively to Z, Y, and X axes. These angles are called Euler angles or Tait–Bryan angles. The common representation is Z-Y-X because that corresponds to yaw, pitch, and roll which are the standard angles required by most drones and UAVs.

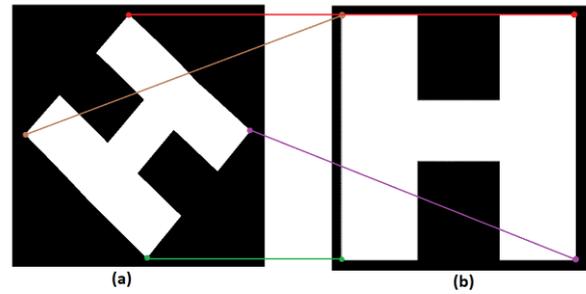


Fig 7: Corner matching of the extracted region of interest (a) and reference image (b)

A rotation R about any arbitrary axis can be written in terms of successive rotations about the Z, Y, and finally, X-axes using the matrix multiplication shown below. In this formulation θ_x , θ_y and θ_z are the Euler angles. Given these three angles, you can easily find the rotation matrix by first finding R_x , R_y , and R_z and then multiply them to obtain R .

Figure 9 shows the result of decomposing the homography matrix to a rotational matrix and thereafter converting to Euler angle. In Figure 9 the roll angle is -1.1 degree, pitch angle is 10.75 degree and yaw angle is 48.66 degree.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix} \quad R_y = \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \quad R_z = \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(a)

$$R = R_z \cdot R_y \cdot R_x$$

(b)

Fig 8: Relationship between Euler angle and rotational matrix

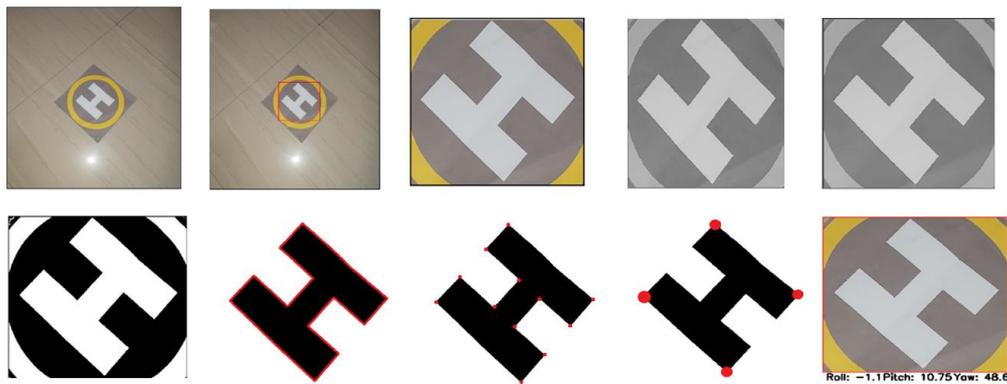


Fig 9: Process Overview from input to output (Output Angle: roll -1.1 degrees, pitch 10.75 degrees and yaw 48.66 degrees)

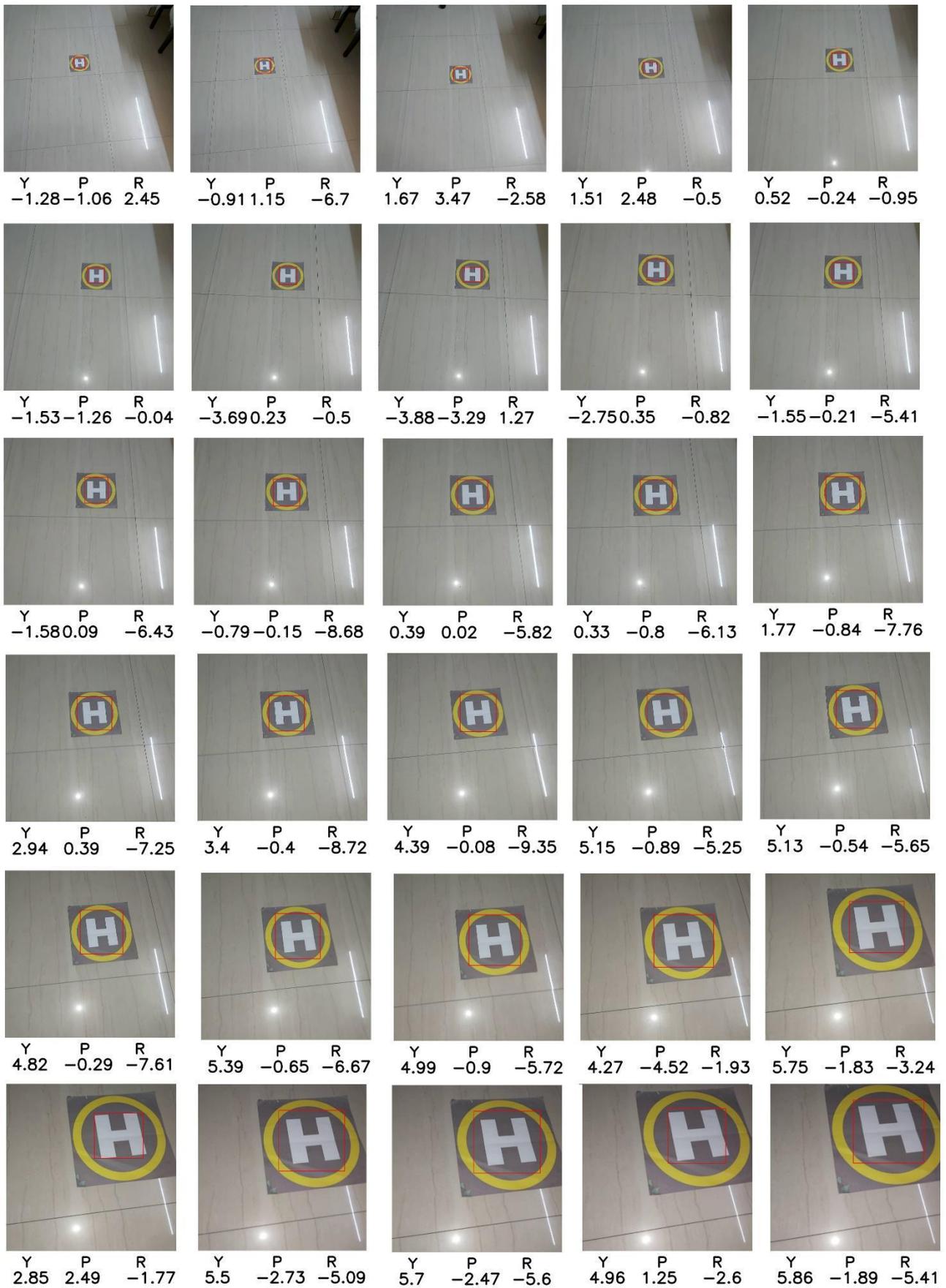


Fig 10: Few Frames of a video representing Roll (R), Pitch (P) and Yaw (Y)

4. FUTURE SCOPE

To improve the performance of the algorithm different object detection algorithms can be used like single shot multibox detectors (SSD) [16] or SSD mobilenets. Further, masked RCNN [17] can be used to extract the contour “H” itself and thereby eliminating the need for image processing and corner detection.

Accuracy can be increased by mapping all 12 corners of the region of interest extracted with 12 corners of the reference image. Other object detection models like EfficientDet [18] and HR Net can be used to further improve the performance of ROI extraction. These models can also be modified at the final layer to customize it for helipad detection. The tensorflow object detection APIs [19] mounted with single stage and two stage detection can further improve the quality of the results.

Parallel processing can be implemented to improve the speed of the system on an aerial UAV/AV. Further, high end GPUs can be used to enhance the inference time and reduce the speed of detection.

5. CONCLUSION

In this paper, a system using object detection and image processing is developed to detect the helipad and estimate its solid angle with respect to the UAV/AV. The system uses a fast object detection algorithm - YOLOv4 to detect the Helipad. Thus obtained image is preprocessed to remove the noise, find contours and to detect the extreme corners. This is followed by homography with a reference image to find the homography matrix which is decomposed to get the rotation matrix, from which the Euler’s angles are deduced. The system being efficient in terms of its speed as well as a lightweight at the same time can be used for the autonomous landing of the UAV/AV.

6. ACKNOWLEDGEMENT

This research project was proposed by NextLeap Aeronautics Pvt. Ltd. for studying methods and algorithms which could be useful in engineering a vision-based landing system for helicopters and VTOL aircrafts. We acknowledge the opportunity and guidance provided to us by Gajendra Kashyap, CTO NextLeap Aeronautics, Dr. Himangi Pande, Associate Professor, MIT World Peace University Pune, Dr. Lalit Kulkarni, Associate Professor, MIT World Peace University Pune.

7. REFERENCES

- [1] O. A. Yakimenko I. I. Kaminerw. J. Lentzp. A. Ghyzel, “Unmanned Aircraft Navigation for Shipboard Landing Using Infrared Vision”, August 2002.
- [2] G.Anitha, Dr. R.N Gireesh Kumar, “Vision Based Autonomous Landing of an Unmanned Aerial Vehicle”, September 2012.
- [3] J. Janousek and P. Marcon, "Precision landing options in unmanned aerial vehicles," 2018 International Interdisciplinary PhD Workshop (IIPHDW), Swinoujście, 2018, pp. 58-60, doi: 10.1109/IIPHDW.2018.8388325.
- [4] Sanchez-Lopez JL, Saripalli S, Campoy P, et al. Toward visual autonomous ship board landing of a VTOL UAV. In: International conference on unmanned aircraft

systems (ICUAS), Atlanta, GA, USA, 28–31 May 2013.

- [5] Premachandra C, Yoshida T and Kato K. A basic study of landing system for multicopters using Raspberry Pi. In: International symposium on consumer electronics (ISCE), Madrid, Spain, 24–26 June 2015.
- [6] Lin S, Garrat MA and Lambert AJ. Monocular vision-based real-time target recognition and tracking for autonomously landing an UAV in a cluttered shipboard environment. *Auton Robot* 2017; 41(4): 881–901. DOI: 10.1007/s10514-016-9564-2.
- [7] Oliveira, César S. de, A. Anvar, Marília Costa Rosendo Silva, Armando Alves Neto and Leonardo Amaral Mozelli. “COMPARISON OF CASCADE CLASSIFIERS FOR AUTOMATIC LANDING PAD DETECTION IN DIGITAL IMAGES.” (2015).
- [8] Arora, Sankalp, Sezal Jain, Sebastian A. Scherer, Stephen Nuske, Lyle Chamberlain and Sanjiv Singh. “Infrastructure-free shipdeck tracking for autonomous landing.” 2013 IEEE International Conference on Robotics and Automation (2013): 323-330.
- [9] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection”, 23 Apr 2020.
- [10] Homography - an overview | ScienceDirect Topics. <https://www.sciencedirect.com/topics/engineering/homography>
- [11] A. Buades, B. Coll and J. M. Morel, "Denoising image sequences does not require motion estimation," IEEE Conference on Advanced Video and Signal Based Surveillance, 2005.
- [12] OpenCV Reference Manual. <https://docs.opencv.org/>
- [13] Satoshi Suzuki and others. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.
- [14] ‘Ramer–Douglas–Peucker algorithm’ (2020) Wikipedia. Available at: http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm
- [15] Zhang Z. (2014) Camera Parameters (Intrinsic, Extrinsic). In: Ikeuchi K. (eds) *Computer Vision*. Springer, Boston, MA.
- [16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg, “SSD: Single Shot MultiBox Detector”, 2016.
- [17] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, “Mask R-CNN”, 24 Jan 2018.
- [18] Tan, Mingxing & Pang, Ruoming & Le, Quoc. (2019). EfficientDet: Scalable and Efficient Object Detection.
- [19] TensorFlow Object Detection API is a Machine Learning modeling library consisting of various pre-trained models. https://github.com/tensorflow/models/tree/master/research/object_detection