# Real-Time Localization Framework for Autonomous Basketball Robots

**Naren Medarametla**
*School of Computer Science Engineering*
*Vellore Institute of Technology*
*Chennai, India*
naren.medarametla2023@vitstudent.ac.in

**Sreejon Mondal**
*School of Electrical and Electronics Engineering*
*Vellore Institute of Technology*
*Chennai, India*
sreejon.mondal2023@vitstudent.ac.in

***Abstract*— Localization is a fundamental capability for autonomous robots, enabling them to operate effectively in dynamic environments. In Robocon 2025, accurate and reliable localization is crucial for improving shooting precision, avoiding collisions with other robots, and navigating the competition field efficiently. In this paper, we propose a hybrid localization algorithm that integrates classical techniques with learning-based methods, relying solely on visual data from the court's floor to achieve self-localization on the basketball field.**

***Keywords*—*Robot Localization, Autonomous Navigation, Neural Networks, Robocon***

## I. Introduction

**Section II** reviews existing methods and prior research related to this work. **Section III** provides a detailed description of our proposed algorithm, approach, and model architecture. **Section IV** provides the results obtained from our experiments. **Section V** evaluates the accuracy of our approach and discusses potential directions for future work.

## II. Related Work

## III. Methodology

Our approach is a two-step process that begins with **Preprocessing** the image, followed by passing it to the model for **Inference**.

### A. Preprocessing

The input image having dimensions (640 × 480 × 3) is converted from the RGB color space to the HSV color space, then the white regions are masked out using two predefined HSV ranges. The image is downsampled through a radial scan, flattened, and finally passed through the neural network.

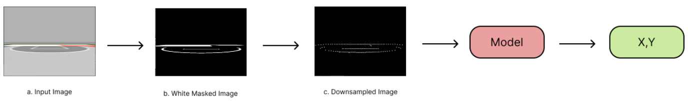

Figure I: preprocessing pipeline

**Algorithm 1** Downsampling

```
    Input: Image
 1: $H \leftarrow$ Image.height
 2: $W \leftarrow$ Image.width
 3: $R \leftarrow$ Black Image
 4: for angle ← 0 to 180 step 2 do
 5:     $lastPixel \leftarrow 0$
 6:     $cx \leftarrow$ W / 2
 7:     $cy \leftarrow$ H
 8:     for d ← 0 to max(H, W) do
 9:         $x \leftarrow cx + d \times \cos(angle)$
10:         $y \leftarrow cy - d \times \sin(angle)$
11:         if $0 \le x <$ W and $0 \le y <$ H then
12:             $pixel \leftarrow Image[y][x]$
13:             if lastPixel = 255 and pixel ≠ 255 then
14:                 $R[y][x] \leftarrow 255$
15:             end if
16:             $lastPixel \leftarrow pixel$
17:         end if
18:     end for
19: end for
20: Return R
```

### B. Model Architecture

The proposed model is a feedforward neural network consisting of a flattening layer followed by four fully connected layers. The first 200 pixels from the top of the image are removed to reduce the input size, as they do not carry useful information.

The resulting image with dimensions (640 × 280 × 1) is flattened into a vector of size 1,79,200 and passed through the first linear layer, followed by a ReLU activation function. The subsequent layers have sizes 1024, 256, and 64, each followed by a ReLU activation. The final layer outputs a 2-dimensional vector representing the predicted $x$ and $y$ positions of the robot.

The rationale for using this relatively simple model lies in the simplicity of the input images and to reducing inference time.

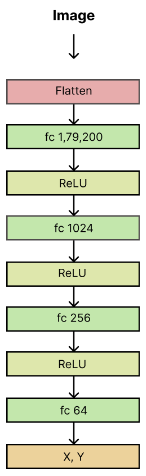

Figure II: architecture diagram

## IV. Results and Analysis

## V. Conclusion and Future Work

## VI. References