HELMET DETECTION

USING FAST AND SCALABLE VISION TECHNIQUES

RAGHAV JUNEJA, SHIVAANSHPANDE
E22CSEU1207, E22CSEU1215

THE RISE OF ACCIDENTS AND NON-COMPLIANCE OF HELMET LAWS

01 **PROBLEM**

High accident rates are often linked to riders not wearing helmets, posing serious safety risks and legal concerns.

02 **Manual Check Challenges**

Enforcing helmet compliance manually is labor-intensive, time-consuming, and often ineffective in real-time scenarios.

03 **Deep Learning Challenges**

hile deep learning can automate detection, it requires high costs, powerful hardware, and large labeled datasets to train accurate models.

OBJECTIVES

Real Time Helmet Detection

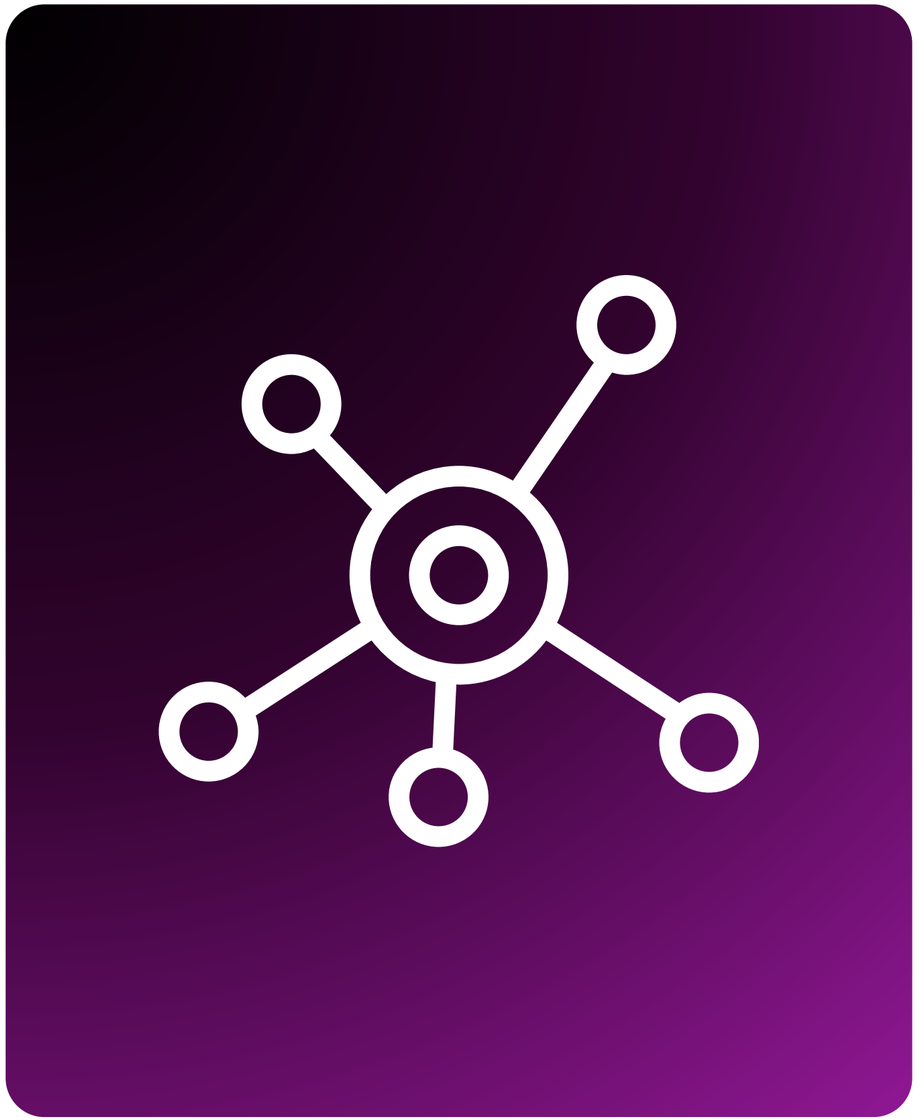
ims to instantly identify helmet usage from video feeds to support timely enforcement.

Reduce Computation Cost

Seeks lightweight models or traditional methods to run efficiently on low-resource devices.

Improve without GPU dependency

Focuses on alternatives to deep learning or optimized models that work well on CPUs.



OVERVIEW OF THE PROPOSED SYSTEM

01 INPUT MATCHING

02 YOLO FOR ROI

03 PREPROCESSING

04 FEATURE MATCHING

05 TEMPLATE MATCHING

06 OUTPUT

METHODOLOGY (STEP BY STEP)

01 **Data collection + Preprocessing**

02 **BRIEF based feature encoding**

03 **Template Matching using Hamming Distance**

04 **Spatial Index Table for fast search**

05 **M-estimator for Robustness**

PREPROCESSING TECHNIQUES

GRAYSCALE CONVERSION

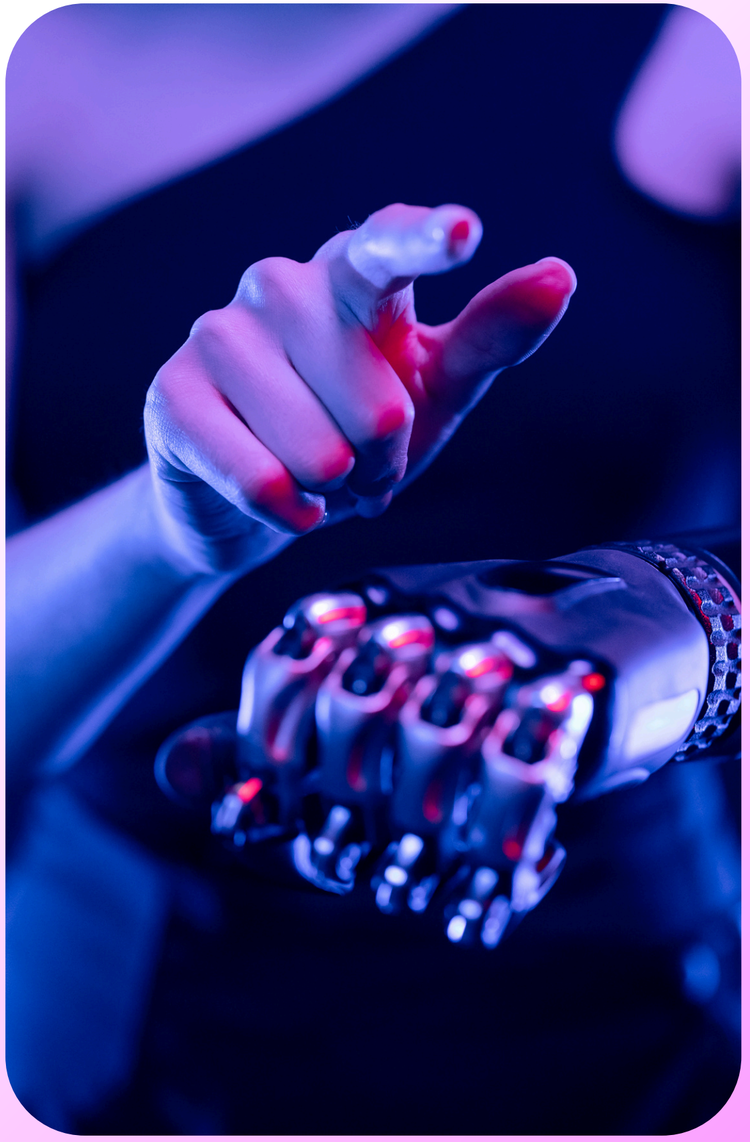
Converts a color image (RGB or BGR) to shades of gray, reducing it to a single channel.

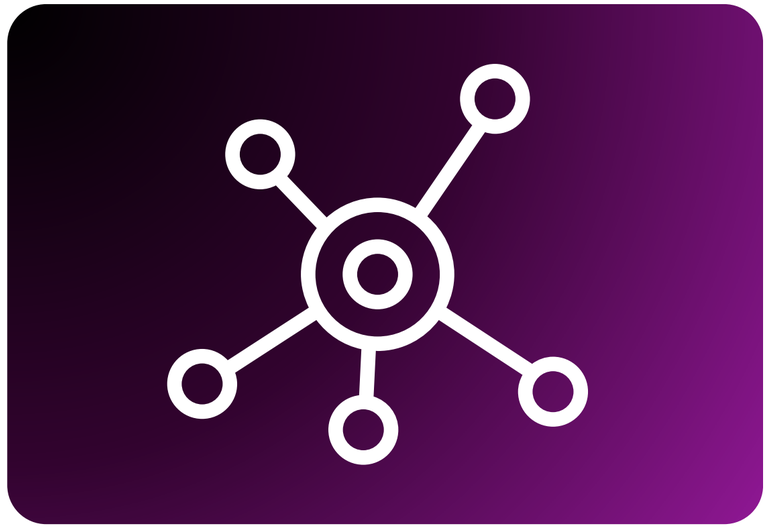
HISTOGRAM EQUALIZATION

Improves image contrast by spreading out the most frequent intensity values. Especially useful in low-contrast images to enhance detail visibility.

ADAPTIVE BINARIZATION

Converts an image to black and white using local thresholds for each region, making it effective for images with varying lighting conditions.
python
CopyEdit





TEMPLATE MATCHING AND SPATIAL INDEX TABLE

01 Template Matching with Keypoints

- Detect keypoints in both template and input image using a detector (e.g., FAST).
- Extract small patches around keypoints for matching.

02 BRIEF: Binary Feature Encoding

- Compare intensity values of point pairs in a patch.
- Each comparison outputs a bit (1 or 0), forming a compact binary string (e.g., 256 bits).
- Enables fast and memory-efficient matching.

03 Fast Matching via Hamming Distance

- Measures the number of bit differences between two BRIEF descriptors.
- Faster than Euclidean distance since it uses bitwise XOR and bit counting.
- Perfect for binary descriptors.

04 SPATIL INDEX TABLE (SPEED UP SEARCH)

- indexes keypoints into a spatial grid (e.g., 2D cells based on position).
- Limits matching to nearby regions in space.
- Reduces unnecessary descriptor comparisons across distant areas.

WEBSITE DEMO!!

Image Processing

Start Detection New Image Image 3 of 3

Detection Results

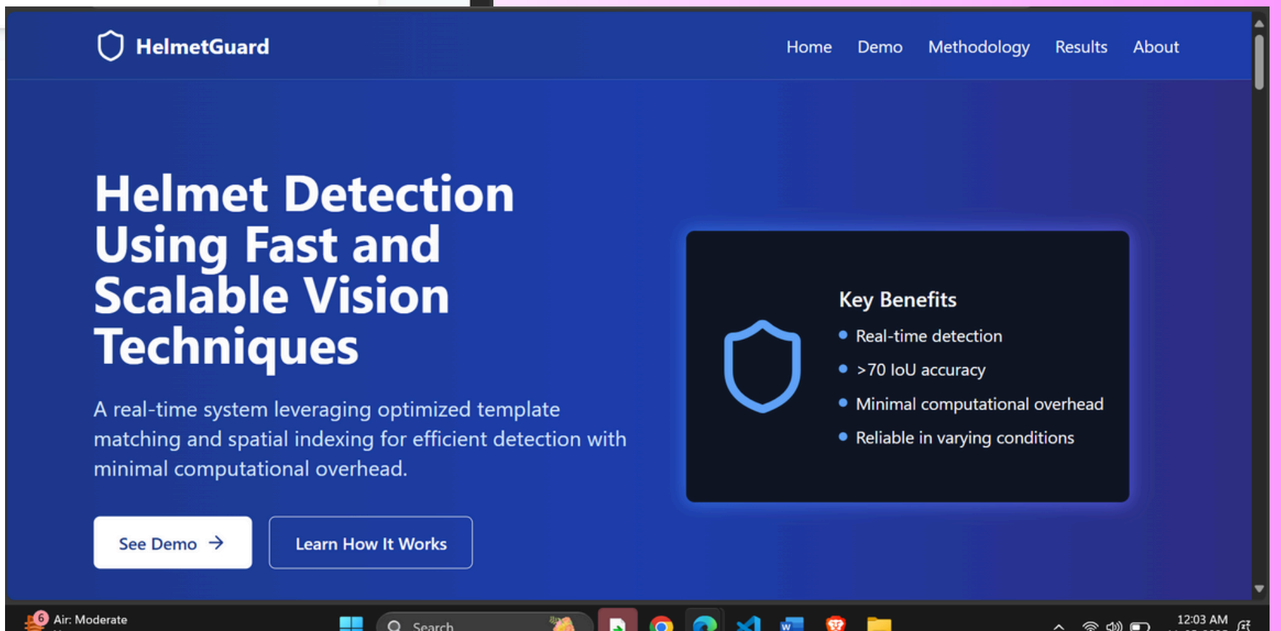
Status
Detection Complete

IoU Accuracy
80%

Processing Time
118 ms

Detection Summary

Helmet detected with high confidence. The system identified the helmet using template matching with binary feature encoding and spatial indexing.



THANK YOU!!!!!!!!!

A PROJECT BY
RAGHAV JUNEJA
E22CSEU1207
&
SHIVAANSH PANDE
E22CSEU1215