Object Weight Estimation from Image Using YOLOv8 & MiDaS

A Deep Learning Approach Without Dataset Dependency

Presented by: Selvapujith T

Abstract

This project presents a novel approach to estimate the weight of real-world objects directly from images without relying on a predefined dataset. The system integrates YOLOv8 for object detection, MiDaS for depth estimation, and classical computer vision techniques for shape detection. By calculating the 3D volume of an object and combining it with predefined density values, the model estimates the object's weight. This approach demonstrates how deep learning can be effectively combined with geometry and physics for practical applications.

Introduction

Accurate weight estimation of objects using only images is a challenging problem with applications in industries such as logistics, manufacturing, and retail. Traditional methods rely heavily on datasets, specialized hardware, or manual measurements. In contrast, this project explores a deep learning and computer vision pipeline that estimates object weight without dataset dependency. YOLOv8 is employed for object detection, MiDaS for depth estimation, and geometric shape approximation is used to infer 3D volume. The final weight is calculated using predefined material densities.

Objectives

- To design an end-to-end system that estimates object weight from a single 2D image.
- To integrate YOLOv8 for detecting objects of interest.
- To apply MiDaS for depth estimation and pixel-to-cm calibration.
- To approximate object shapes and compute their 3D volume.
- To calculate weight using known material densities.
- To evaluate limitations and suggest improvements.

Background & Literature Review

Object weight estimation is traditionally performed using physical scales or through dataset-driven machine learning models. Prior research focuses on regression-based weight prediction, 3D scanning, or multi-camera systems. However, these methods are often impractical for real-time applications due to cost, dataset dependency, or hardware requirements. The proposed method leverages YOLOv8, which provides robust object detection, and MiDaS, a powerful monocular depth estimation model. Combining these tools allows real-time weight estimation using only a single RGB image.

Proposed Methodology

The system workflow consists of several steps:

1. Object detection using YOLOv8.
2. Depth estimation using MiDaS.
3. Pixel-to-centimeter calibration using a reference object.
4. Shape detection (2D contours) and 3D shape prediction.
5. Volume computation based on detected shape.
6. Weight estimation using density values.

System Architecture & Workflow

The system follows a modular pipeline:

- Input image is provided with a reference object for calibration.
- YOLOv8 detects objects and provides bounding boxes.
- MiDaS predicts the relative depth of the detected object.
- Shape detection module identifies 2D shapes and infers possible 3D shape.
- Volume is calculated using geometric formulas.
- Weight is derived using material density values.

Implementation

The implementation was carried out in Python using PyTorch, OpenCV, and Ultralytics YOLO. The following key steps were followed in the code:

- Loading YOLOv8 pretrained weights for object detection.
- Loading MiDaS model for monocular depth estimation.
- Detecting 2D shape contours using Canny edge detection and polygon approximation.
- Converting pixel dimensions into real-world dimensions using calibration.
- Predicting 3D shape (cuboid, cylinder, triangular prism) and computing volume.
- Estimating weight using predefined density values.

Code Snippet:

```
import torch
import cv2
import numpy as np
```

```
from ultralytics import YOLO

# Load YOLOv8 model
yolo_model = YOLO('yolov8n.pt')

# Load MiDaS for depth estimation
midas = torch.hub.load('intel-isl/MiDaS', 'MiDaS')
midas.eval()
midas_transforms = torch.hub.load('intel-isl/MiDaS', 'transforms').default_transform

# Example analysis function
def analyze_image(image_path, ref_obj_width_px, ref_obj_width_cm):
    image = cv2.imread(image_path)
    results = yolo_model(image)[0]
    # Further steps: shape detection, depth estimation, volume, weight prediction...
```

Sample Output & Results

The system successfully detects objects, estimates their depth, and calculates approximate weight. For example, when analyzing an image with a reference coin of width 2.4 cm corresponding to 60 pixels, the system can estimate the dimensions of a detected object. A cuboid-like object may be estimated at a volume of 150 cm^3 with an approximate weight of 0.405 kg (assuming aluminum density).

Analysis & Discussion

The system demonstrates that combining YOLOv8 and MiDaS provides a feasible solution for object weight estimation without dataset dependency. However, accuracy depends on precise calibration, quality of input images, and correct shape detection. Complex or irregularly shaped objects may yield inaccurate results. The approach works best with simple geometric objects like cuboids, cylinders, and triangular prisms.

Limitations

- Accuracy depends heavily on calibration using reference objects.
- MiDaS provides relative depth, not absolute scale, which can lead to errors.
- The method assumes uniform material density.
- Irregular shapes and textures may reduce detection accuracy.
- Currently supports only basic shapes (cuboid, cylinder, triangular prism).

Future Work

- Integrating SAM (Segment Anything Model) for better object boundary segmentation.
- Expanding shape library to include irregular and composite shapes.
- Using multiple reference objects for improved calibration.
- Combining multi-view images or stereo vision for absolute depth estimation.
- Deploying the system as a real-time mobile application.

Output

```
Speed: 4.1ms preprocess, 233.8ms inference, 1.6ms postprocess per image at shape
Shape: triangular_prism
Volume: 2983.67 cm³
Estimated Weight: 60.644 kg
```

Conclusion

This project presents an innovative approach to object weight estimation from images using YOLOv8 and MiDaS. By combining object detection, depth estimation, and geometric approximation, the system achieves dataset-independent weight prediction. While limitations exist, the proposed pipeline is a step forward toward real-world applications in logistics, industrial automation, and e-commerce.