Automatic Detection for Plastic Bottles

Orgad Shlishman

Abstract

Recycling is a key component in reducing environmental pollution and conserving resources, with plastic bottles being a significant target due to their prevalence and impact. Automating the detection of plastic bottles can greatly improve the efficiency of recycling processes by streamlining their identification and sorting. In this project, a YOLO (You Only Look Once) neural network was utilized for real-time detection of plastic bottles in images, demonstrating high-speed and accurate performance.

To enhance the detection system's robustness, especially in cases of blurry images, an unsharp masking algorithm was integrated as a preprocessing step. This algorithm sharpens image details and highlights the contours of objects, enabling the YOLO model to perform effectively even under suboptimal imaging conditions. The combined approach ensures improved accuracy in plastic bottle detection, contributing to more effective recycling operations.

1 My Work

In this project, I developed and adapted the YOLO algorithm to enable object detection specifically for plastic bottle images. The work involved fine-tuning the model to ensure accurate recognition across diverse scenarios.

After implementing the code for detecting plastic bottles, I focused on addressing the challenge of suboptimal input images. To achieve this, I incorporated techniques for improving blurry images, allowing our automated methodology to effectively recognize bottles even in less-than-ideal conditions.

The code and additional details can be found on GitHub: https://github.com/OrgadShlishman/Environment-Engineering/tree/main

1.1 YOLO

In our project we chose to use YOLO for plastic bottles detection. YOLO (You Only Look Once) is a real-time object detection algorithm that stands out for its speed and accuracy. Unlike traditional methods that break the task into separate steps like region proposal and classification, YOLO treats object detection as a single regression problem. It divides the input image into a grid and predicts bounding boxes and class probabilities for each grid cell simultaneously.

The algorithm uses a convolutional neural network (CNN) to process the image in one pass, which makes it highly efficient for real-time applications. YOLO's unique architecture allows it to achieve a balance between precision and speed, making it a popular choice for tasks like detecting objects in videos, autonomous driving, and surveillance systems. By training on diverse datasets, YOLO can generalize well to detect objects in various contexts, including specific items like plastic bottles.

After enabling YOLO, it was partly working and we concluded that training/fine tuning is required for improving the results.

1.2 YOLO Examples

The YOLO model is doing a good job for plastic bottles detection, but in some cases, it cannot detect all of them, and sometimes it can't detect them at all. As stated before, one possible option is to use some more training to improve the results. See Figure 1.

Figure 1: YOLO - recognizing plastic bottles (The results are not perfect)

1.3 Unsharp Masking

Another way of improving results we were dealing with, is Unsharp Masking. The idea is simple: improve the input image so the network (YOLO) would have easier task of recognizing the plastic bottles.

The theory behind Unsharp Masking Algorithm: Unsharp masking is an image enhancement technique used to improve the perceived sharpness of an image. Despite its name, the method sharpens images by emphasizing edges and fine details. The algorithm works by subtracting a blurred version of the image (referred to as the "unsharp mask") from the original image.

The process involves three main steps:

1. **Blurring**: A low-pass filter (often a Gaussian blur) is applied to create a smoothed version of the image.

2. **Edge Detection**: The blurred image is subtracted from the original, highlighting the high-frequency components (edges and details).

3. **Recombination**: The edge information is scaled by a factor (strength) and added back to the original image, enhancing the edges.

Mathematically, if I is the original image, I_b is the blurred image, and k is the sharpening factor, the sharpened image I_{sharp} is given by:

$$I_{\text{sharp}} = I + k \cdot (I - I_b)$$

where $I - I_b$ represents the high-frequency components, and k controls the intensity of the sharpening effect.

The algorithm involves three key parameters:

- **Radius**: Determines the size of the blurring kernel, controlling how much of the edges are enhanced.

- **Strength (percent)**: Adjusts the weight of the high-frequency components.

- **Threshold**: Sets a minimum intensity difference to apply sharpening, preventing amplification of noise.

Unsharp masking is widely used in photography, medical imaging, and computer vision tasks to enhance image clarity and detail without introducing significant artifacts.

1.4 Unsharp Masking Examples

Unsharp Masking can improve YOLO results for bottles detection in some cases, especially when the input images quality is not so high. For some cases, using unsharp masking can degrade the results, so it should be used just in specific cases. See Figures 2 and 3.

Figure 2: Original Image - YOLO can't recognize the plastic bottle

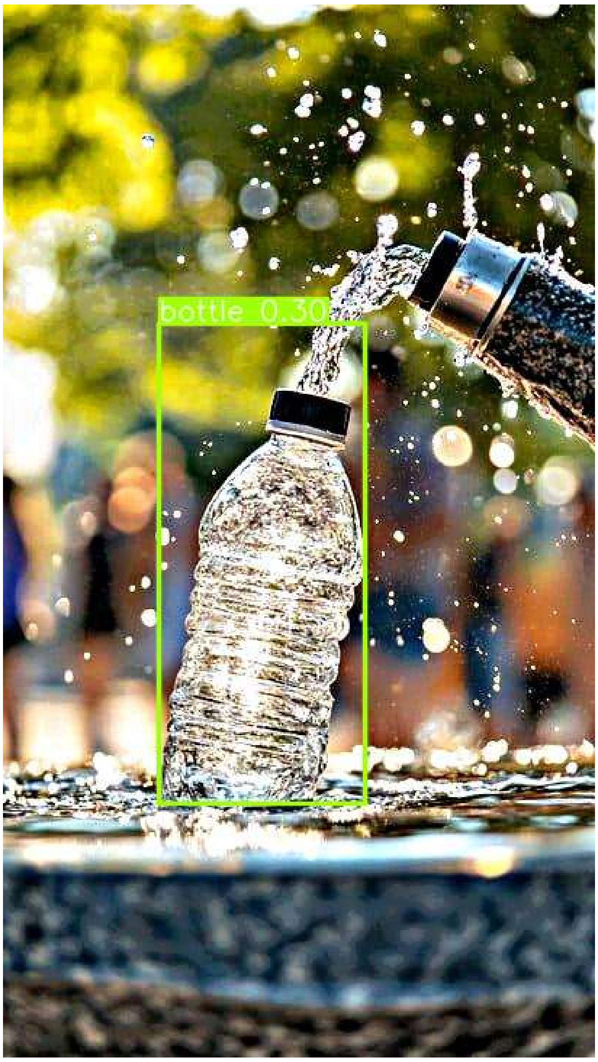

Figure 3: Sharp Image - YOLO can detect the plastic bottle

2 More Resources

YOLO

- https://pjreddie.com/darknet/yolo/
- https://www.v7labs.com/blog/yolo-object-detection
- https://docs.ultralytics.com/#where-to-start

Unsharp Masking

- https://en.wikipedia.org/wiki/Unsharp_masking
- https://www.cambridgeincolour.com/tutorials/unsharp-mask.htm
- https://docs.gimp.org/2.6/en/plug-in-unsharp-mask.html