YOLO Multiobject Color Attack (YMCA)

Christian Cipolletta

Project Manager: Dr. Robi Polikar

Department of Electrical & Computer Engineering; Rowan University, Glassboro, NJ 08028

cipoll17@students.rowan.edu, polikar@rowan.edu

Problem Statement & Significance

- ***You Only Look Once (YOLO) architecture object detectors are being trusted to make critical decisions more than ever before (e.g. self-driving cars).***
- ***Colorants (e.g. paint, marker) are easily accessible ways for adversaries to change the appearance of objects.***
- ***We want to determine whether – and to what extent – YOLO is vulnerable to artificial coloring attacks.***
- ***It is hypothesized that an object that is colored differently than normal will decrease a YOLO model's ability to detect said object.***

Background / Prior Methods

There has been a lot of research focused on adversarial attacks against convolutional neural networks (CNNs). Focus was put on white-box attacks (e.g. Fast Gradient Sign Attack) where the attackers had full access to the model [1]. Black-box attacks, where the attacker has little knowledge, are more realistic. Current black-box attacks focus on issues like lighting, orientation, obstructions, stickers, and tape [2, 3].

Requirements / Deliverables

- ***Requirements***
  - Real-time object detection capability
  - Ability to perform well when faced with adversarial attacks (Robustness)
- ***Deliverables***
  - Algorithm and method for testing color-based vulnerabilities on custom datasets

Standards & Constraints

- ***Applicable engineering standards***
  - IEEE Standard for Robustness Testing and Evaluation of Artificial Intelligence (AI)-based Image Recognition Service [4].
  - IEEE Standard for Artificial Intelligence (AI) Model Representation, Compression, Distribution, and Management [5].
  - IEEE Recommended Practice for the Quality Management of Datasets for Medical Artificial Intelligence [6].
- ***Real-world constraints***
  - Real-time object detection is defined as 15 frames per second (FPS)
  - Compatibility with a Nvidia Jetson AGX Orin [7].
  - Current publicly available drone datasets are limited.

Design Impact Statement

- ***Impact on public health, welfare, safety***
  - If the model fails while in use it can cause serious injury or death. This is why robust testing is done on attacks most likely to be seen in real life.
- ***Impact on the environment***
  - Low power-consumption GPUs are used to lessen the amount of electricity needed to run the model in practice (which uses less fuel if on a vehicle).
- ***Impact on economic factors***
  - Fast GPUs are expensive, so YOLO models are used because they can get real time speeds on cheaper GPUs.
- ***Impact on social, cultural, global factors***
  - The adversaries are always advancing, so the model is well documented and can be easily re-trained on a growing dataset.

References & More Information

This QR code leads to a PDF with the references, the Github Repository of the project, the Gantt Chart, and the PDF of the poster.

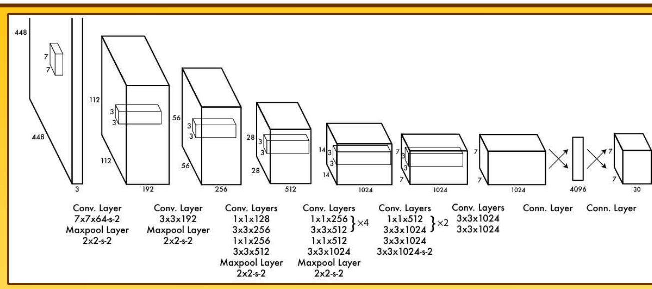

Figure 1. YOLO architecture which consists of 24 convolutional layers and 2 fully connected layers [11].

Alternative Approaches/ Optimization

- ***Using synthetic drone dataset***
  - Initial testing was done on our synthetic drone dataset.
  - The dataset only has 2 classes (drone, airplane).
  - The Common Objects in Context (COCO) dataset was used instead because of its 80 classes and standardization [8].
- ***Using only 6 colors and 5 opacities***
  - Originally testing was only done on the 6 primary and secondary colors, and 5 opacities.
  - This method did not generate enough data for any conclusions to be significant.

Approach / Methodology

1. Calculate the Mean Average Precision (mAP) score of a YOLO model on an unaltered dataset, shown in equation 3 [9].
2. Use segmentation labels or model to alter the color of the objects.
3. Calculate the mAP score of the same YOLO model on the altered dataset.
4. Repeat steps 2 and 3 for each color and opacity of interest.

$$Precision = \frac{CorrectPredictions}{TotalPredictions}$$

$$Recall = \frac{CorrectPredictions}{TotalGroundTruth}$$

$$mAP = \frac{1}{N}\sum_{k=1}^{k=N} AP_k$$

where N is the number of classes.

Equation 1, 2, 3

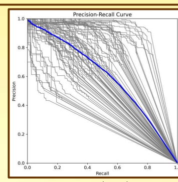

Figure 2. This is the Precision-Recall curve from the unaltered dataset. Each line is a different class. Average Precision (AP) is calculated by finding the area under the curve of a single class.

mAP50-95 Calculation:

1. AP is calculated for the intersection over union (IoU) threshold of 0.5 for each class.
2. Calculate the precision at every recall value (0 to 1 with a step size of 0.01), then it is repeated for IoU thresholds of 0.55,0.60,…,.95.
3. Average is taken over all 80 classes and 10 thresholds.

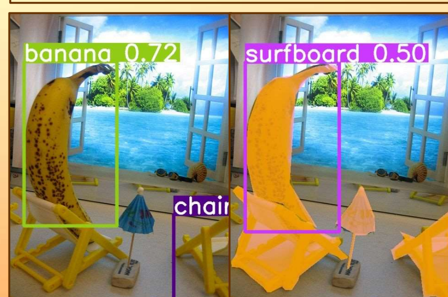

Figure 3. Model predictions when run on an unaltered image (left) and objects colored orange with an opacity of 80% (right).

Experimental Setup

- ***Experimental Setup***
  - Test the pretrained YOLOv8 model on the 5k validation images in the COCO dataset [10].
  - Then use segmentation labels of COCO to color each object and retest the model.
  - Repeat for 9 colors and 10 opacities.
- ***Evaluation and Validation***
  - The performance of the model was evaluated by using the mAP score.
- ***Datasets & Parameter Selection***
  - ***Common Objects in Context (COCO):*** Dataset created by Microsoft in 2017 with 300k training images, 5k validation images, and 80 object classes.
  - ***9 Colors:*** red, blue, green, yellow, purple, orange, gray, brown, white
  - ***10 Opacities:*** 10% to 100% in increments of 10%

Results

- All 9 colors resulted in worse model performance than the control
- 9 opacities decreased mAP score when compared to the control
- 10% opaque actually increased the mAP score
- Model achieves an average 52.18 FPS predictions when run on A6000 GPU
- Model runs on Jetson Orin, but speed testing was not done
- All requirements were met or exceeded

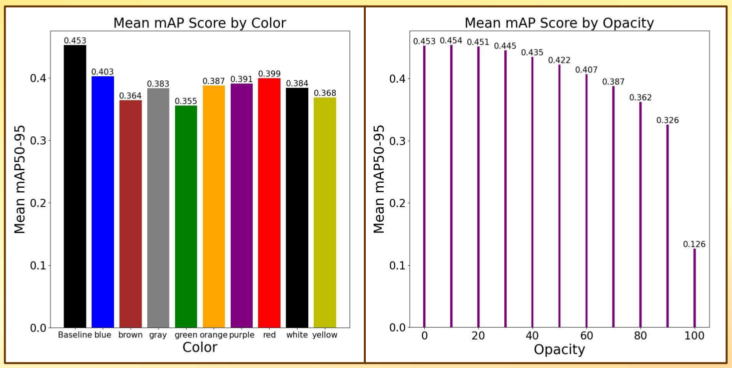

Figure 4. The mean mAP score grouped by color (left) and by opacity (left).

Conclusion & Future Work

- ***Conclusions***
  - Overall, we found that all 9 colors and most opacities will decrease the mAP score of the model by a varying amount.
  - The current model is effective against small perturbations shown by low opacities being an ineffective attack.
  - However, it is not common for a colorant to be very opaque meaning that this a large vulnerability to YOLO systems.
- ***Limitations***
  - Assumed that coloring objects digitally will have the same effect as coloring physical objects.
  - Many custom datasets do not have both segmentation and detection labels.
- ***Future Work***
  - Implement more robust training methods for the drone detection model (e.g. augmentations, mosaics).
  - Test the method on individual object classes rather than every object.