ML YOLOv7 Project Proposal

1st Andrew Serra
Dept. Computer Science (MS)
Rochester Institute of Tech.
Rochester, USA
acs8929@rit.edu

2nd Ayo Owolabi
Dept. Computer Science (MS)
Rochester Institute of Tech.
Rochester, USA
oo8583@rit.edu

3rd Daniel Gebura
Dept. Computer Science (MS)
Rochester Institute of Tech.
Rochester, USA
djg2170@rit.edu

4th Wendy Wang
Dept. Computer Science (MS)
Rochester Institute of Tech.
Rochester, USA
hw3555@rit.edu

***Abstract*—This paper explores and proposes potential improvements to the YOLOv7 model. After discussing the original paper's task, dataset and metrics, the architecture of the machine learning model is explained in depth. Finally, with proof of repeatable results from the paper, further improvements to the model will be proposed.**

I. Task Definition, Evaluation Protocol, and Data

A. Task Definition

The paper [1] we have chosen aims to set a new benchmark in real-time object detection with the introduction of YOLOv7, an advanced version of the YOLO (You Only Look Once) [2] family of real-time object detectors. The authors attribute their impressive accuracy improvements to two key enhancements applied to the traditional YOLO model: the use of a "Trainable Bag-of-Freebies" and an overall architectural improvement.

The "Trainable Bag-of-Freebies" refers to a series of techniques that enhance the model's accuracy without increasing inference cost. These techniques include architectural re-parameterization and optimization strategies that make the model more efficient during the training phase [3].

The second enhancement is a new network architecture that balances the trade-off between speed and accuracy. The architecture includes components such as Extended Efficient Layer Aggregation Networks (ELAN) and planned re-parameterized convolutional layers [1]. ELAN improves the network's learning ability by aggregating layers of different depths without significantly increasing the number of parameters.

B. Dataset

The dataset used for training this model was the Microsoft COCO dataset [5], a large-scale object detection dataset containing images of everyday scenes labeled with common objects in their natural context. The training set consists of over 118,000 images containing about 860,000 labeled objects across 80 categories. Additionally, a validation set of 5,000 images was withheld to evaluate model performance during development.

C. Metrics

The paper includes several metrics to evaluate model performance, including Average Precision (AP), specifically AP_{50} and AP_{75}, which measure the model's precision when the predicted bounding box overlaps with the ground truth by at least 50% and 75% Intersection over Union (IoU), respectively. Average Recall (AR) was also used to assess the model's ability to detect objects across images. Frames Per Second (FPS) was used as a measure of the model's efficiency, which is critical for real-time applications where processing speed is as important as accuracy. For overall model analysis, the model size was indicated in terms of parameters used, and computational cost was measured in the number of Floating Point Operations (FLOPs).

II. Machine Learning Model

A. General Architecture

YOLOv7's architecture is a Deep Convolutional Neural Network that integrates advanced components such as Extended Efficient Layer Aggregation Networks (ELAN), Cross Stage Partial Networks (CSP) [5], and Feature Pyramid Networks (FPN) [6]. These components work together to improve feature extraction, aggregation, and object detection capabilities.

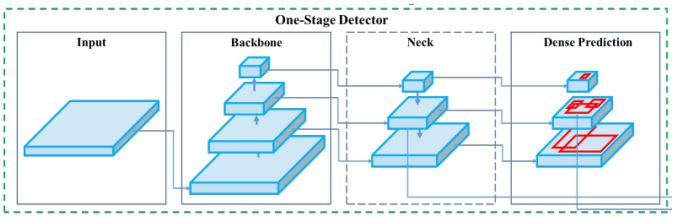

Fig. 1. YOLOv7 overall architecture.

B. Backbone

The backbone network serves as the feature extractor, transforming input images into rich feature representations. CSP structures are incorporated to enhance learning efficiency and reduce computational cost [5]. By partitioning feature maps and merging them through cross-stage connections, CSP networks improve gradient flow. Additionally, ELAN extends the concept of layer aggregation to improve the network's learning ability without significantly increasing parameters. It enables the network to learn more diverse and comprehensive features by aggregating layers of different depths.

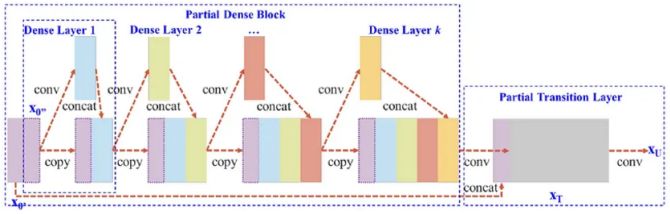

Fig. 2. Cross Stage Partial Network Architecture. CSP separates feature maps of the base layer into two parts. One part will go through a dense block and a transition layer, while the other part is then combined with the transmitted feature map to the next stage.

C. Neck and Head Layers

The neck and head of YOLOv7 are responsible for aggregating features at multiple scales and performing final object classification and localization. The neck utilizes FPN to combine feature maps from different stages of the backbone network [6]. This is achieved through lateral connections, which merge feature maps from different layers to create multi-scale representations, and a top-down pathway, which combines upsampled higher-level semantic features with lower-level features to refine detection capabilities [7]. YOLOv7 employs multiple detection heads corresponding to different scales, allowing the model to detect small, medium, and large objects. These heads use anchor boxes to predict bounding boxes, as well as some anchor-free mechanisms to enhance flexibility [9].

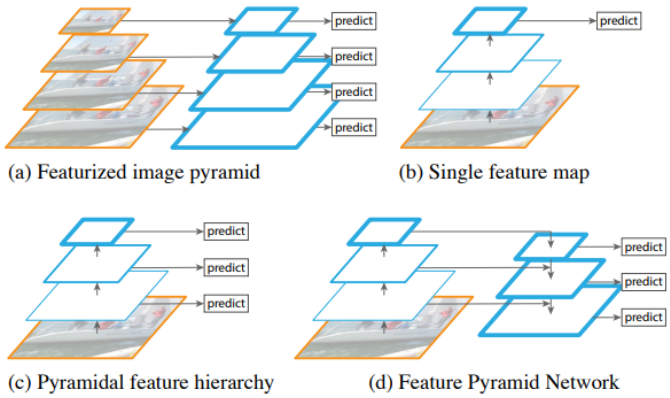

Fig. 3. Visualization of varying techniques to construct feature pyramids. (a) Using an image pyramid. (b) Single scale features. (c) Reuse the pyramidal feature hierarchy as if it were a featurized image pyramid. (d) Feature Pyramid Network (FPN)

D. Trainable Bag-of-Freebies

Lastly, YOLOv7 introduces several training techniques under the concept of the "Trainable Bag-of-Freebies," which enhance model accuracy without increasing inference costs. These techniques include data augmentation methods to create new training samples and improve generalization, and regularization techniques, such as label smoothing to prevent overconfident predictions and DropBlock regularization to promote robustness by randomly dropping entire regions of feature maps [8]. Additionally, dynamic label assignment algorithms are used to better match predictions with ground truth, optimizing the matching process.

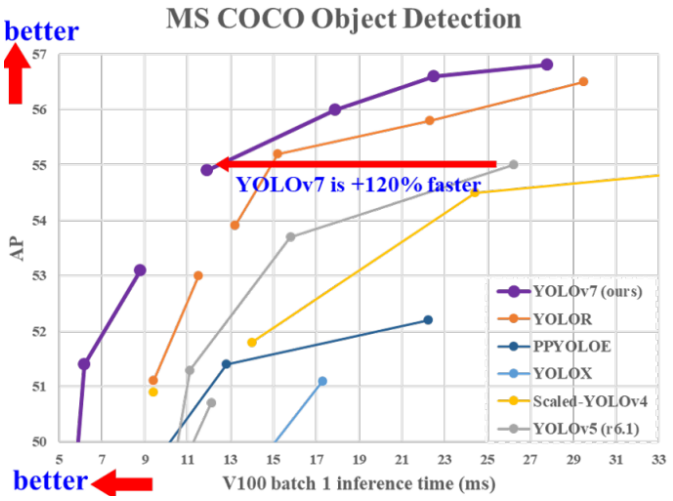

Fig. 4. Comparison with other real-time object detectors, YOLOv7's proposed methods achieve state-of-the-art performance.

III. Proposed Improvements

To explore alternative methods and assess their impact on the YOLOv7 model, we propose two significant modifications:

1) Implement a different data augmentation method by integrating **CutMix** augmentation into the training pipeline.
2) Replace the default **Leaky ReLU** activation functions in YOLOv7 with the **Mish** activation function to investigate its effect on model performance.

These modifications aim to enhance the model's generalization capabilities and learning efficiency. Below, we provide a detailed explanation of each modification, including the rationale, implementation details, and expected outcomes.

A. Integrating CutMix Data Augmentation

1) Current Data Augmentation: YOLOv7 employs a range of data augmentation techniques under the "Trainable Bag-of-Freebies" concept to improve model generalization without increasing inference cost. The primary augmentation methods used include:

- **Mosaic Augmentation:** Combines four training images into one mosaic image, allowing the model to learn from a richer context and improving detection of objects at varying scales and positions.
- **MixUp Augmentation:** Blends two images and their labels to create a new training sample, helping the model generalize better by learning from combined features.
- **Standard Augmentations:** Includes random horizontal flipping, color jittering, and random scaling and translation.

2) Proposed Modification: As a way to enhance the model's generalization and robustness, we propose adding **CutMix** augmentation to the existing data augmentation pipeline. CutMix is an advanced augmentation technique that generates new training samples by cutting and pasting patches among training images while adjusting the ground truth labels accordingly [10].

What CutMix Entails:

- **Image Mixing:** Randomly select two images from the training dataset and replace a random patch of the first image with a patch from the second image.
- **Label Mixing:** Adjust the labels to reflect the presence of objects from both images. The label for the mixed region is a combination of the labels from both images, weighted by the area of the patches.

3) Implementation Details:

- Modify the data loader to include CutMix augmentation, setting a probability p_{cutmix} for applying CutMix to each batch (e.g., 50%).
- Implement the CutMix algorithm within the training pipeline, ensuring labels are correctly adjusted for the mixed images.

B. Replacing Leaky ReLU with Mish Activation Function

1) Current Activation Function: YOLOv7 primarily uses the **Leaky Rectified Linear Unit (Leaky ReLU)** activation function throughout its architecture:

$$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha x, & \text{if } x < 0 \end{cases}$$

where α is a small constant (typically 0.1 or 0.01).

2) Proposed Modification: We propose replacing all Leaky ReLU activation functions with the **Mish** activation function [11]:

$$f(x) = x \cdot \tanh(\ln(1 + e^x))$$

Characteristics of Mish:

- Non-monotonic and smooth, providing better gradient flow due to its smooth nature.
- Potential to enhance feature representation and overall network performance.

3) Implementation Details:

- Replace all instances of Leaky ReLU in the YOLOv7 architecture with the Mish activation function.
- Adjust the learning rate or other hyperparameters if necessary to accommodate the new activation function.

C. Summary of Proposed Modifications

Table 1 (Bottom of document) summarizes these modifications accordingly.

D. Conclusion

The proposed modifications aim to enhance YOLOv7's performance by integrating advanced data augmentation techniques and exploring alternative activation functions. These changes are straightforward to implement within the existing framework and have the potential to yield significant improvements in object detection tasks.

By thoroughly evaluating these modifications against the baseline model, we can gain valuable insights into their effectiveness and contribute to the ongoing development of efficient and accurate real-time object detectors.

IV. Viability Test

To validate that the YOLOv7 model can be replicated and thus modified, we have ran the original model with its pre-trained weights on the original COCO dataset with success.

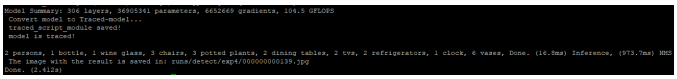

Fig. 5. Test results proving that the model works as provided in our framework

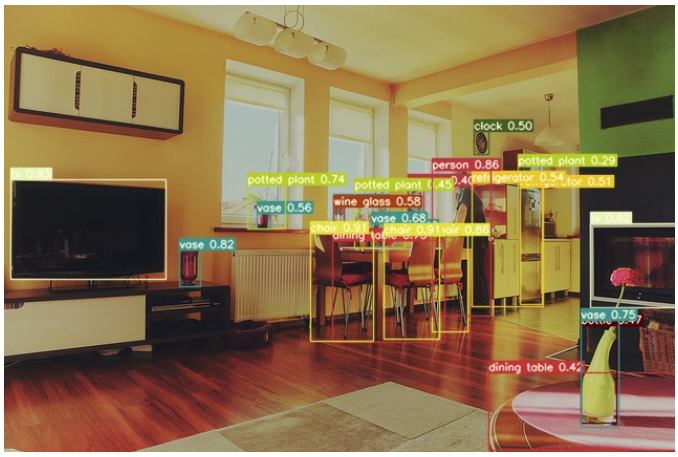

Fig. 6. Test results proving that the model works as provided in our framework

Additionally, we have been able to train the original model over 1 epoch of the complete training set. Training over 1 epoch took 1.648 hours on the provided CS servers, resulting in a model with 69% precision and 57% recall among 5,000 images with 36,335 labels in the validation set.

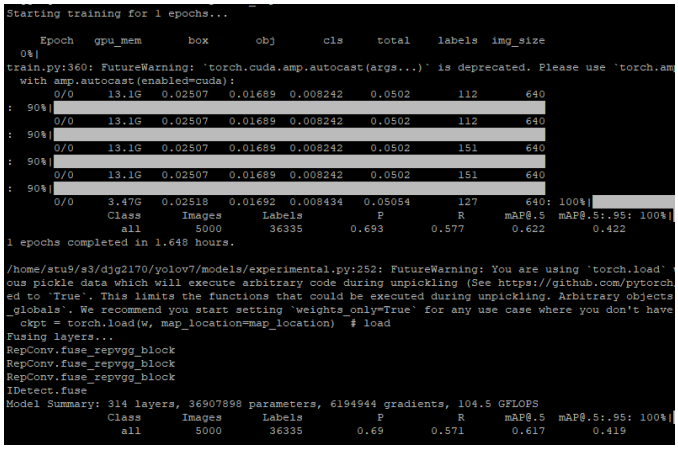

Fig. 7. Test results proving ability to train the model over 1 epoch over the dataset

References

[1] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

[2] Redmon, J. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[3] Ding, Xiaohan, et al. "Repvgg: Making vgg-style convnets great again." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.
[4] Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer International Publishing, 2014.
[5] Wang, Chien-Yao, et al. "CSPNet: A new backbone that can enhance learning capability of CNN." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020.
[6] Lin, Tsung-Yi, et al. "Feature pyramid networks for object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
[7] Liu, Shu, et al. "Path aggregation network for instance segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
[8] Ghiasi, Golnaz, Tsung-Yi Lin, and Quoc V. Le. "Dropblock: A regularization method for convolutional networks." Advances in neural information processing systems 31 (2018).
[9] Zhang, Shifeng, et al. "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.
[10] Yun, Sangdoo, et al. "Cutmix: Regularization strategy to train strong classifiers with localizable features." Proceedings of the IEEE/CVF international conference on computer vision. 2019.
[11] Misra, Diganta. "Mish: A self regularized non-monotonic activation function." arXiv preprint arXiv:1908.08681 (2019).

TABLE I
LIST OF PROPOSED MODIFICATIONS

Mod.	Aspect	Original Implementation	Proposed Modification
1	Data Augmentation	YOLOv7 employs Mosaic, MixUp, and standard augmentations (e.g., random flipping, color jittering) to enhance model generalization.	Integrate **CutMix** augmentation into the training pipeline by modifying the data loader to apply CutMix with a defined probability. This involves combining patches from different images and adjusting labels accordingly to improve model robustness and localization accuracy.
2	Activation Functions	YOLOv7 utilizes **Leaky ReLU** activation functions across all network layers, providing computational efficiency and mitigating the "dying ReLU" problem.	Replace all **Leaky ReLU** activation functions with the **Mish** activation function. Update the model architecture to incorporate Mish in place of Leaky ReLU in each convolutional layer, aiming to enhance gradient flow, feature representation, and overall detection accuracy.