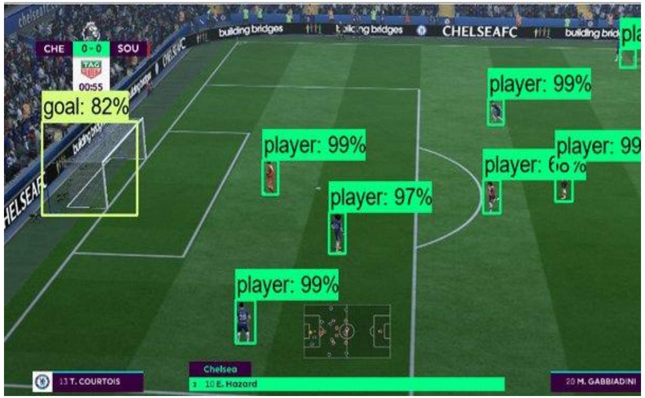
UNO Card Detection and Classification Using Yolo v5

Efe Tarhan 22002840 and **Ege Yüceel** 22003324

Bilkent University, Türkiye

CS 484/555 - Introduction to Computer Vision

Problem Definition and Motivation

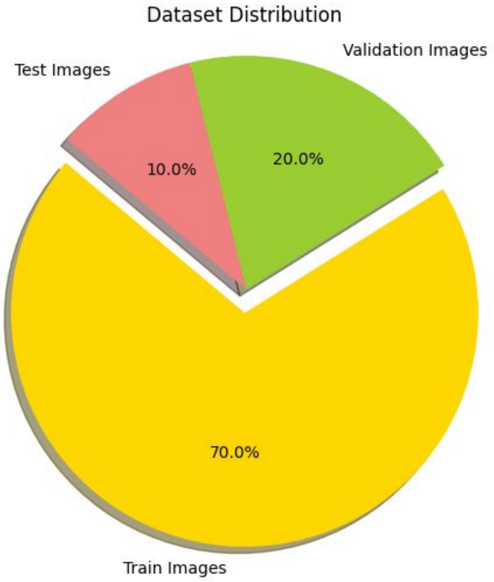


Our Problem of Interest

Computer Vision in Games

Dataset Description

- 8992 raw images
- 26.976 labeled images generated with combining raw images with various backgrounds
- 15 different card types and labels
- Labels consist of **coordinates of the upper left corner**, **width** and **height** of the box and **type** of the card



	Box 1	Box 2
Label	**14**	**13**
x-coord.	**261**	**276**
y-coord.	**213**	**227**
Width	**28**	**17**
Height	**21**	**16**

[1] A. Crawshaw, "Uno Cards Object Detection Dataset," *Roboflow*, July 24, 2022. [Online]. Available: https://public.roboflow.com/object-detection/uno-cards. [Accessed: 24 April 2024].

YOLO v5 Architecture

- CSPDarkNet53-based **backbone**
- Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) for the **neck**
- Anchor-based detection **head**.

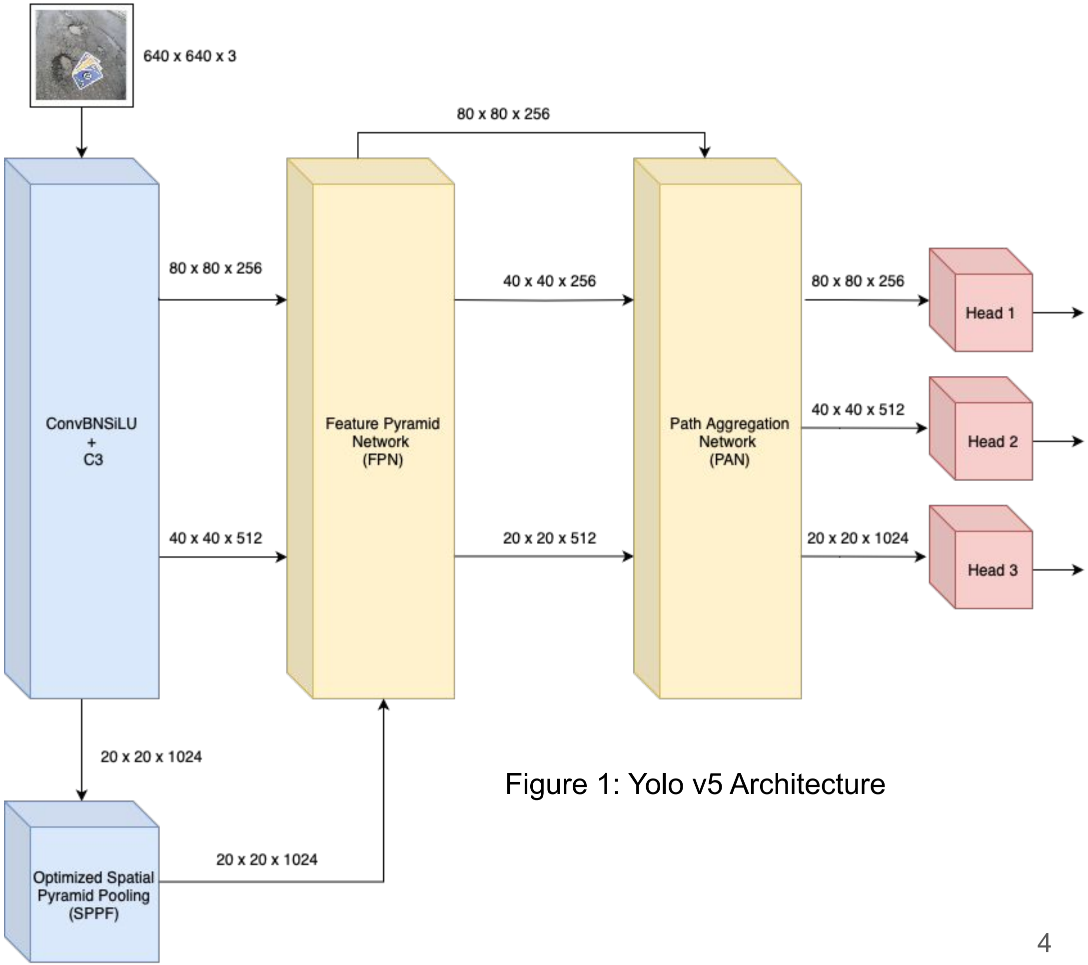


Figure 1: Yolo v5 Architecture

[2] Ultralytics, "Architecture Summary," *Architecture Summary - Ultralytics YOLOv8 Docs*,January 14, 2024. [Online]. Available: https://docs.ultralytics.com/yolov5/tutorials/architecture_description/. [Accessed: 24 April 2024].

YOLO v5 Backbone

- ConvBNSiLU layers for **feature extraction.**
- **Residual connections** in C3 blocks prevent information loss across layers.
- Optimized Spatial Pyramid Pooling layer for **multi-scale feature aggregation**.

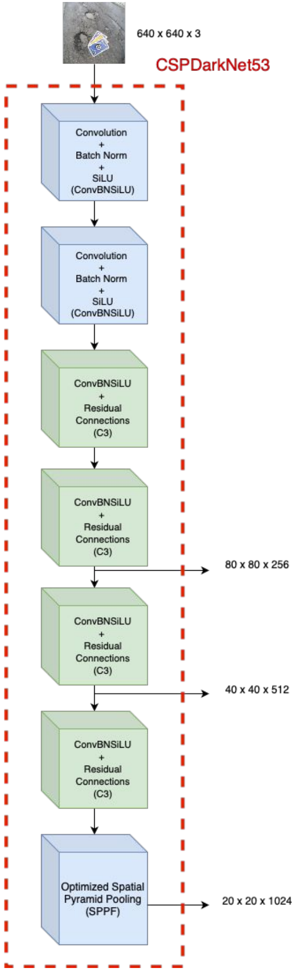


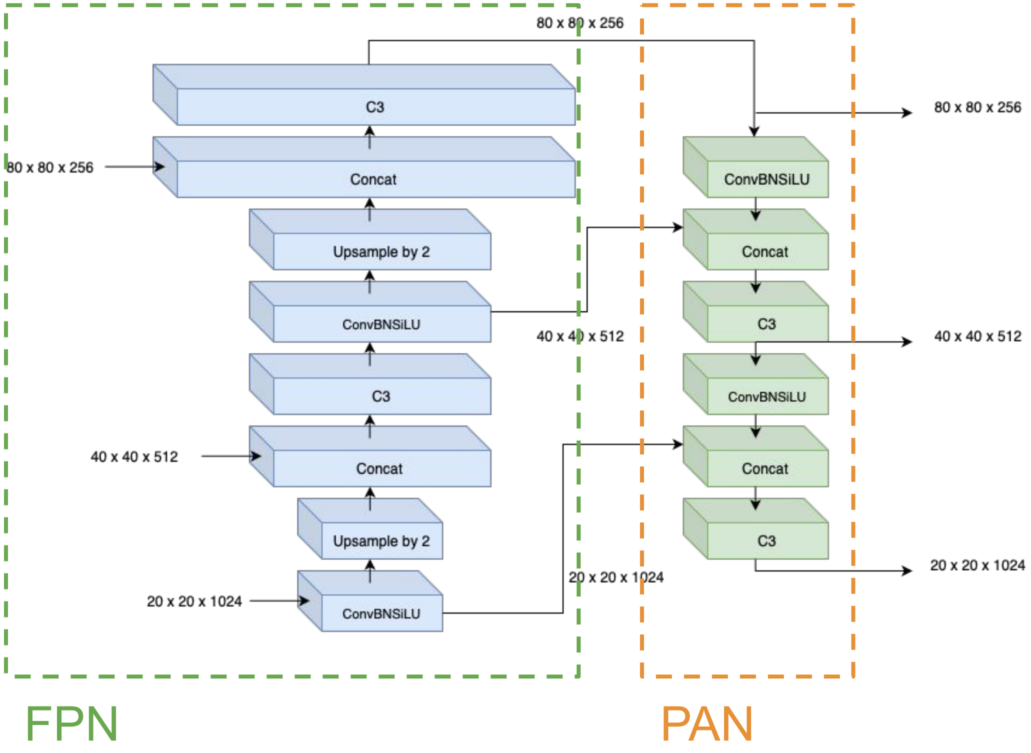
Figure 2: Yolo v5 Backbone

[2] Ultralytics, "Architecture Summary," *Architecture Summary - Ultralytics YOLOv8 Docs*,January 14, 2024. [Online]. Available: https://docs.ultralytics.com/yolov5/tutorials/architecture_description/. [Accessed: 24 April 2024].

YOLO v5 Neck

- FPN generates **multi-scale feature maps.**
- Adds a **bottom-up path** to the FPN to improve feature hierarchy and representation.
- Facilitates the flow of **low-level features** to the top layers and **high-level features** to the bottom layers.

Figure 3: Yolo v5 Neck



[2] Ultralytics, "Architecture Summary," *Architecture Summary - Ultralytics YOLOv8 Docs*,January 14, 2024. [Online]. Available: https://docs.ultralytics.com/yolov5/tutorials/architecture_description/. [Accessed: 24 April 2024].

YOLO v5 Head

- **Multi-Scale Predictions:** Predicts on three scales to capture a wide range of object sizes.
- **Anchor Boxes:** Uses anchor boxes to initialize shape and size for bounding box predictions.
- **Non-Maximum Suppression (NMS):** Applies NMS to refine the predictions, keeping only the most accurate bounding boxes for detected objects.

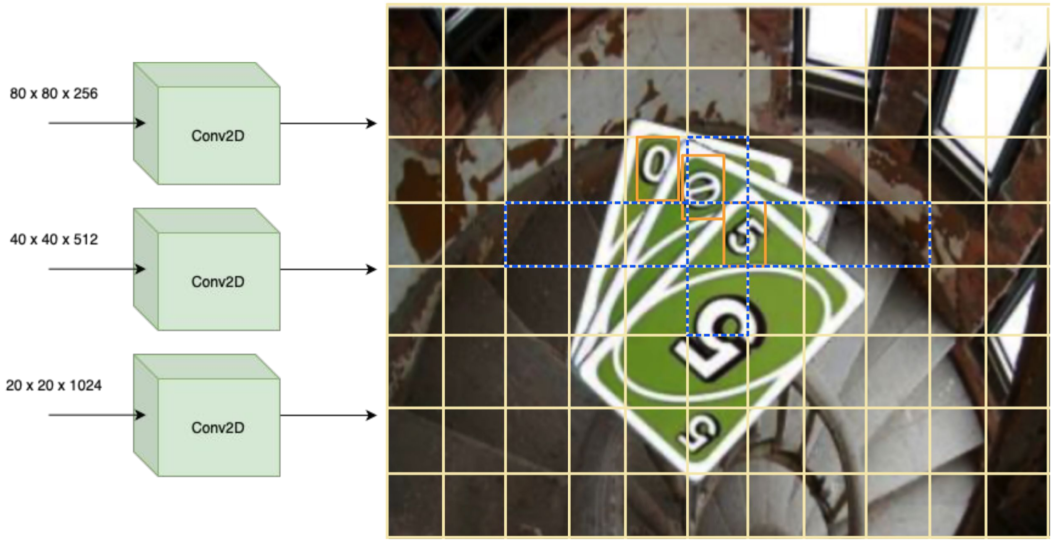
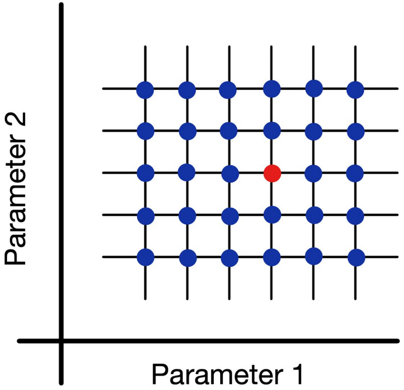


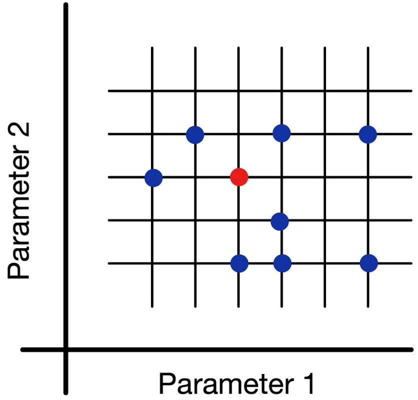
Figure 4: Yolo v5 Head

[2] Ultralytics, "Architecture Summary," *Architecture Summary - Ultralytics YOLOv8 Docs*,January 14, 2024. [Online]. Available: https://docs.ultralytics.com/yolov5/tutorials/architecture_description/. [Accessed: 24 April 2024].

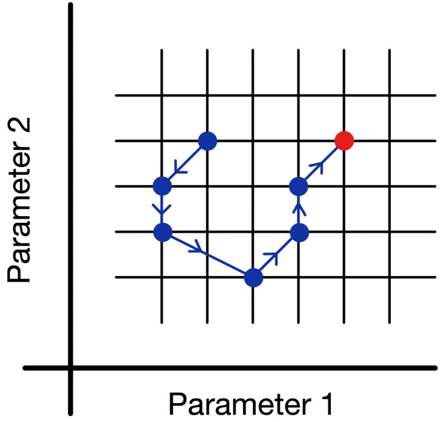
Hyperparameter Tuning



Grid Search



Random Search



Bayesian Optimization

Hyperparameter Objectives

- **Epoch Number**

  Ranging between 5 and 10.

- **Batch Size**

  8, 12 or 16

- **Learning Rate**

  Between 1e-4 and 1e-1

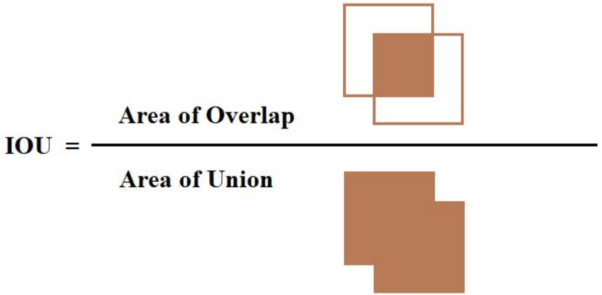
- **Momentum**

  Between 0.85 and 0.99

Optimization Metric: mAP_0.5:0.95

Mean Average Precision

- **Mean Average Precision (mAP):** Averages precision scores across classes and IoU thresholds, reflecting overall model accuracy.
- **Intersection over Union (IoU):** Measures the overlap between predicted and actual object boundaries, influencing precision calculation.



$$Precision = \frac{TP}{TP + FN}$$

$$Recall = \frac{TP}{TP + FP}$$

mAP_0.5:0.95 averages precision across IoU thresholds from 0.5 to 0.95.

Results

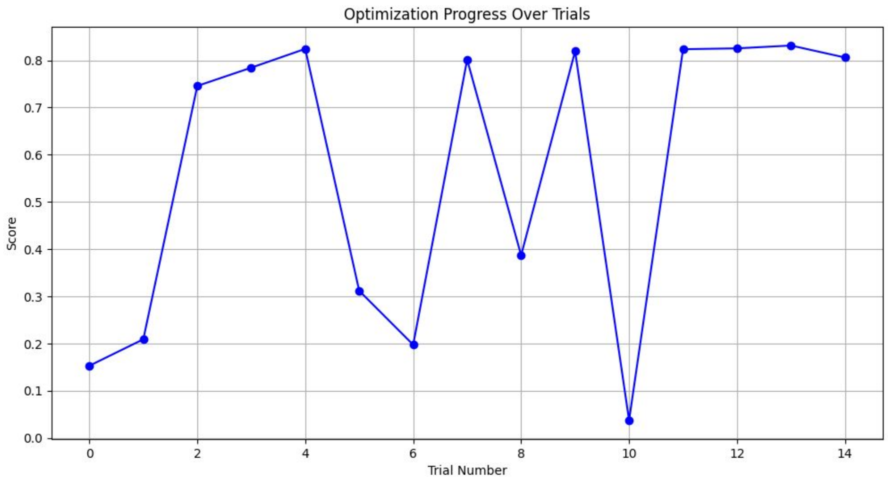


Figure 5: Progress over Trials

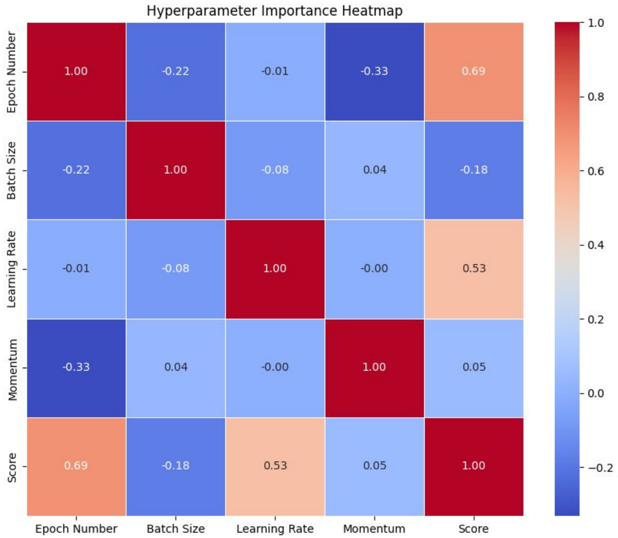


Figure 6: Importance Heatmap

- Accuracy reaching up to 83.5%

- Epoch - Score
- Learning Rate - Score
- Momentum - Epoch

Results Cont'd

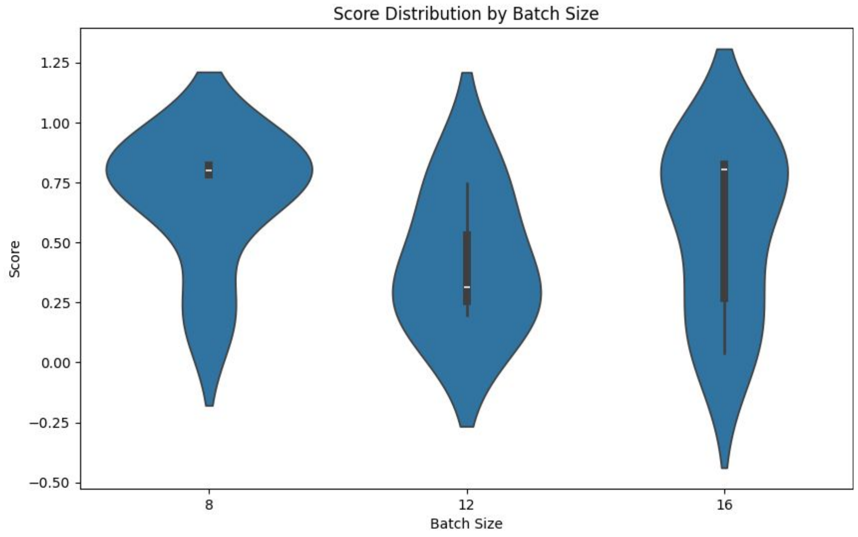


Figure 7: Score Distribution by Batch Size

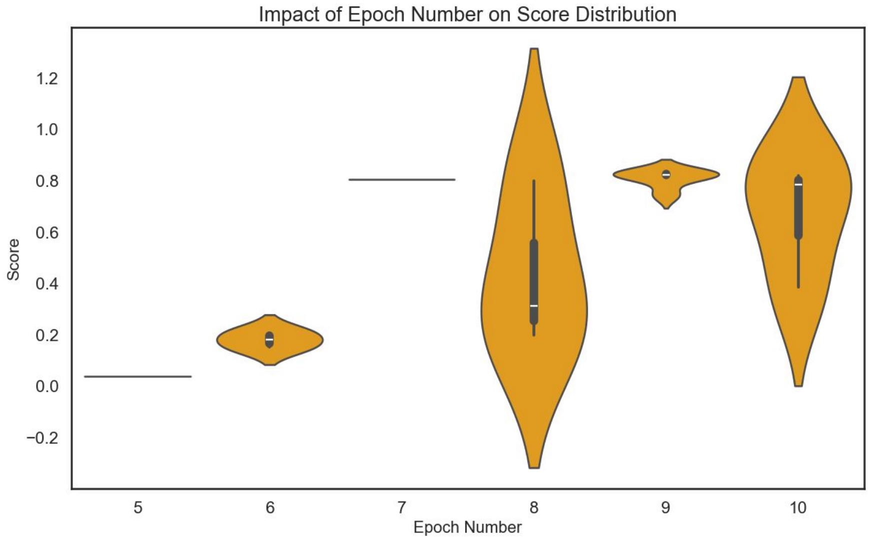


Figure 8: Score Distribution by Epoch Number

Results Cont’d - Best Attempt

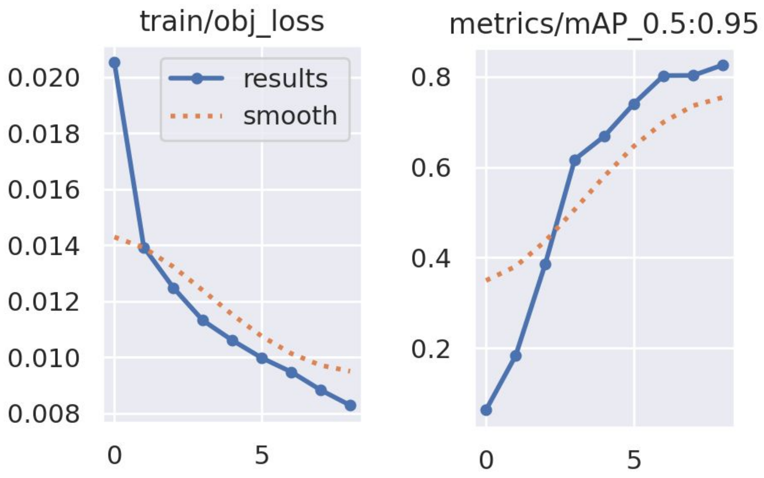


Figure 9: Progress of Best Train

Optimal Parameters

Epoch	Batch Size	Learning Rate	Momentum
10	16	0.0142893	0.9402136

Figure 10: Selected Examples (mAP Used)