

Playing Cards Object Detection Using YOLOv8

Teodor Kostadinov, student at
Faculty of Mathematics and Informatics, Sofia University

Abstract

This paper presents the development and evaluation of a machine learning model for detecting and identifying playing cards using the YOLOv8 architecture. The objective is to assist in card games by providing real-time card recognition. The study explores various approaches, including using synthetic and real datasets, augmentation techniques, and model fine-tuning.

1 Introduction

The goal is to develop an application capable of recognizing playing cards to assist in games like Blackjack, Bridge, Belot, and Poker. The project faced several challenges, including the lack of senior data science expertise and insufficient time to create a model from scratch or develop a comprehensive dataset.

2 Methodology

2.1 Initial Approach

To quickly implement the model, a pre-trained YOLOv8 model was chosen due to its accessibility through the Python Ultralytics library. However, the lack of a public dataset necessitated the use of a synthetic dataset.

2.2 Datasets

All playing cards datasets used during the training of the models are available in the project directory. All datasets are in YOLOv8 object detection format, split into *train*, *valid*, and *test* directories with *labels* and *images* subdirectories. All datasets fall under the CC0: Public Domain license.

2.2.1 The Synthetic Dataset

The synthetic dataset was copied from Kaggle - Playing Cards Object Detection Dataset. It includes 20,000 images synthetically generated and labeled with 52 classes. This dataset was used to train the YOLOv8m.synthetic model.

2.2.2 The "Real" Dataset

The real dataset was created by the author, Teodor Kostadinov, and includes 100 images shot and labeled by the author using Label Studio with 13 classes. This dataset was used to train the YOLOv8m_real and YOLOv8m_tuned models.

2.2.3 The "Real" Augmented Dataset

To address data scarcity, the real dataset was augmented using the `imgaug` library with the script `augment_dataset.ipynb`. This augmentation introduced 10 new augmented images for each image in the real dataset using various transformations, resulting in a total of 1,000 images. This augmented dataset was used to train the YOLOv8m_aug model.

2.2.4 The Combined Dataset

The combined dataset was created using the `imgaug` library along with the scripts `combine_datasets.py` and `transform_labels_in_dataset.py`. It combines the full real dataset with 10 times more images taken from the synthetic dataset. This combined dataset was used to train the YOLOv8m_comb model.

2.3 Training Procedure

The datasets were used to train various YOLOv8 models. The training was conducted using an NVIDIA RTX A2000 8GB GPU. The training details for each model are summarized in the following table:

Model	Dataset	Epochs	Training Time
YOLOv8m_synthetic	20,000 synthetic images	10	2 hours
YOLOv8m_real	100 real images	100	20 minutes
YOLOv8m_aug	1,000 augmented images	100	40 minutes
YOLOv8m_comb	100 real + 1,000 synthetic	100	50 minutes
YOLOv8m_tuned	100 real images	100	10 minutes

Table 1: Training details for YOLOv8 models

3 Results

3.1 YOLOv8_Synthetic Model

- **Training:** 20,000 images, 10 epochs, NVIDIA RTX A2000 8GB
- **Performance:** 99% accuracy on the test set

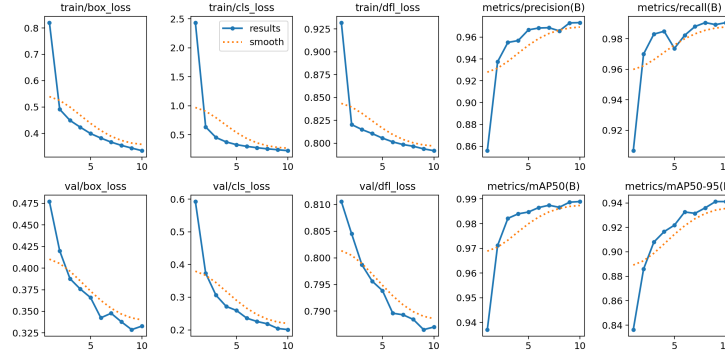


Figure 1: YOLOv8_Synthetic Model Results

3.2 YOLOv8_Real Model

- **Training:** 100 images, 100 epochs, NVIDIA RTX A2000 8GB
- **Performance:** Limited due to the small dataset

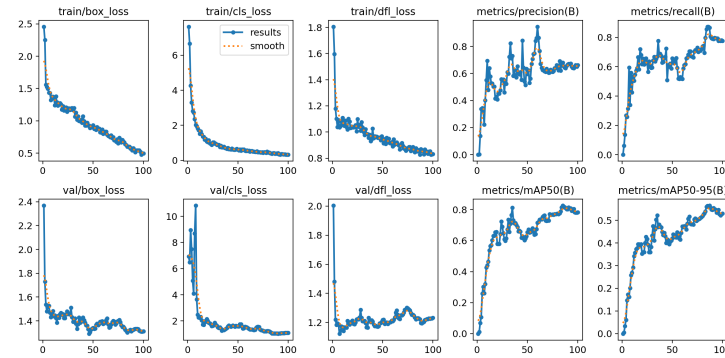


Figure 2: YOLOv8_Real Model Results

3.3 YOLOv8_Augmented Model

- **Training:** 1,000 images, 100 epochs, NVIDIA RTX A2000 8GB
- **Performance:** Similar to YOLOv8_Real

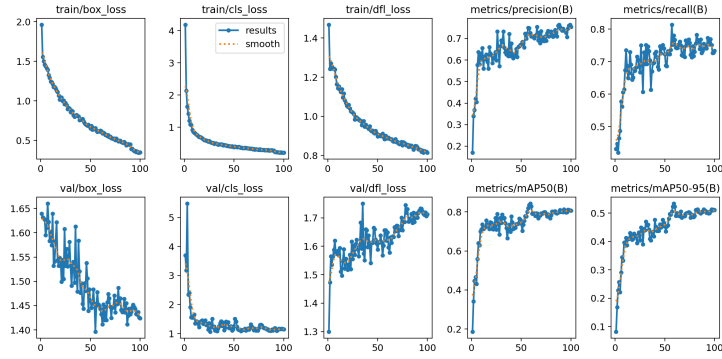


Figure 3: YOLOv8_Augmented Model Results

3.4 YOLOv8_Combined Model

- **Training:** 1,100 images, 100 epochs, NVIDIA RTX A2000 8GB
- **Performance:** Worse than synthetic-only model

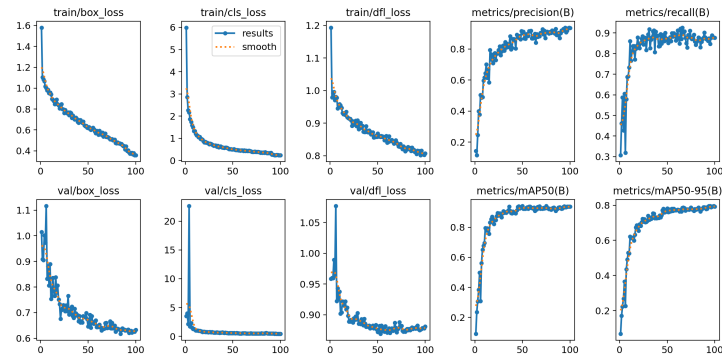


Figure 4: YOLOv8_Combined Model Results

3.5 YOLOv8_Tuned Model

- **Training:** Fine-tuned with 100 real images
- **Performance:** Improved detection of specific cards like Ace of Hearts but issues with suit differentiation

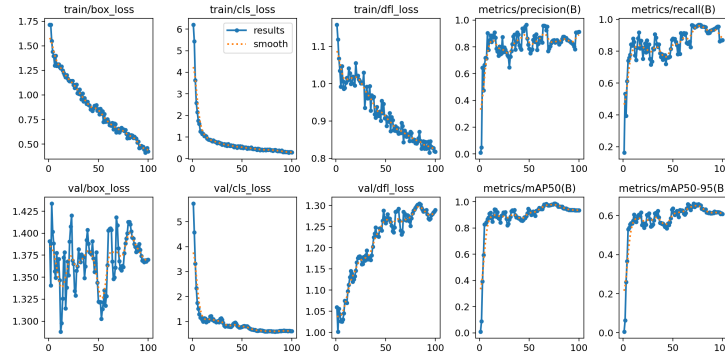


Figure 5: YOLOv8_Tuned Model Results

4 Discussion

The study demonstrates that using a synthetic dataset with YOLOv8 can yield high accuracy for card detection under controlled conditions. Real-life datasets require more data to achieve comparable results. Augmentation alone did not significantly improve performance. Fine-tuning synthetic models with real data shows promise, but further refinement and more data are necessary.

5 Conclusion

More data generally leads to better results in object detection tasks. Fine-tuning synthetic models with real-life data appears to be a viable strategy, though combining datasets effectively remains challenging. Future work should focus on increasing the dataset size and exploring better data augmentation and combination techniques.

6 References

- Playing Cards Object Detection Dataset on Kaggle
- DTD Dataset
- imgaug Documentation