Chessy3D: A study of 3D Chessboard Detection and Pose Estimation

Luca Villani
matr. 200412
University of Modena and Reggio Emilia
269419@studenti.unimore.it

Alessandro Mezzogori
matr.
University of Modena and Reggio Emilia
271617@studenti.unimore.it

Davide Della Casa Venturelli
matr.
University of Modena and Reggio Emilia
xxxxx@studenti.unimore.it

***Abstract*—In this paper, we present a novel framework for chessboard corners detection and chess-pieces object detection. Our approach tackles the significant challenges posed by the variability of the photos taken of the chessboard and sizes, color and differences of the chess pieces. We propose a new method for chessboard keypoint detection that utilizes a combination of deep learning and geometric constraints to accurately identify the keypoints on the chessboard and the squares within it. This method is robust to variations in lighting, angle, and chessboard designs, making it suitable for real-world applications. Additionally, we developed a chess-piece object detection model that can accurately detect and classify chess pieces in various conditions. Our study delves into the intricacies of chessboard detection, addressing the challenges of different chessboard designs and piece variations. The proposed approach using YOLOv8 for chess-piece detection demonstrates superior performance in terms of accuracy and robustness compared to existing methods. We used two datasets for training and testing our models: ChessRed2K and another small dataset found on Roboflow. After the object detection we will propose a FEN annotation method for converting the chessboard state into a FEN (Forsyth-Edwards Notation) string, which is a standard notation for describing the state of a chess game. Additionally we show a method for retrieving similar images using one hot encoding and a similarity search algorithm.**

I. Introduction

The detection and analysis of keypoints in images have become crucial tasks in the field of computer vision, with applications ranging from object recognition and tracking to augmented reality and autonomous driving.

Identifying corners in images is a fundamental problem in computer vision, with applications in object recognition, image stitching, and 3D reconstruction. Chessboard detection is a specific case of keypoint detection that has gained significant attention due to its importance in camera calibration, pose estimation, and robotics. The chessboard pattern provides a regular and structured grid that can be easily detected and used to estimate the camera's intrinsic and extrinsic parameters. In this paper, we present an approach for chessboard corners detection and chess-pieces object detection using deep learning and geometric constraints. Our method is designed to handle the challenges posed by the variability of chessboard photos, including different lighting conditions, angles, and chessboard designs.

We have been through a series of steps to achieve our goal:

- We started with the chessboard corners detection, using a combination of deep learning and geometric constraints to accurately identify the keypoints on the chessboard.
- We then moved on to chessboard squares detection, which involves detecting the individual squares on the chessboard.
- Next, we focused on chess-pieces object detection, using YOLOv8 to detect and classify chess pieces in various conditions.
- After that, we developed a FEN annotation method for converting the chessboard state into a FEN string.
- Finally, we implemented an image similarity search method using one hot encoding and a similarity search algorithm.
- We also conducted a series of experiments to evaluate the performance of our approach, comparing it with existing methods and demonstrating its effectiveness in real-world scenarios.

We used two datasets for training and testing our models: ChessRed2K and another small dataset found on Roboflow. The rest of the paper is organized as follows:

- Section II discusses the related work in chessboard detection and keypoint detection.
- Section III describes the proposed approach for chessboard corners detection.
- Section IV presents the chessboard squares detection method.
- Section V details the chess-pieces object detection using YOLOv8.
- Section VI explains the FEN annotation method.
- Section VII discusses the image similarity search method.

II. Chessboard Corners Detection

- Davide

III. Chessboard Squares Detection

- Davide

IV. Chess-Pieces Object Detection

- Luca graphicx

We initially approached the problem of chess-pieces object detection using DETR (Detection Transformer), with a ResNet-50 backbone, used for object detection tasks. The

DETR model was pre-trained on the COCO dataset and fine-tuned on a custom dataset of 2000 chess images, which included various chess pieces in different lighting conditions, angles, and designs annotated in the COCO format with 12 classes representing the different chess pieces (e.g., white-pawn, white-knight, white-bishop, etc.). However, we encountered several challenges with this method, particularly in terms of performance and accuracy. Despite tuning and training the model for 100, 150 and 200 epoch, DETR consistently underperformed. The DETR model struggled with the variability of chess pieces in different lighting conditions, angles, and designs, leading to suboptimal detection results. We found out that the model was not able to converge properly, and the loss function did not decrease as expected during training. This led us to explore alternative approaches for chess-piece object detection.

We then turned our attention to YOLOv8, a state-of-the-art object detection model that has shown promising results in various applications. We trained YOLOv8 on the same custom dataset of 2000 chess images, which included various chess pieces in different lighting conditions, angles, and designs. The training process was more successful with YOLOv8, as the model was able to converge properly and achieve a lower loss function. The YOLOv8 model demonstrated superior performance in terms of accuracy and robustness compared to DETR.

The model was trained from a pre-trained checkpoint found online and fine-tuned on our custom dataset.[ref to github repo] We conducted three training runs with different parameters like:

- Data augmentations (mosaic augmentation)
- HSV shifts
- Horizontal and vertical flips

The training was performed using PyTorch 2.1 on an NVIDIA RTX 3090 GPU with 24GB of VRAM.

The three training runs were conducted with the two datasets in this order:

- YOLOv8m run 1: A small dataset found on Roboflow, which contains 100 images of chess pieces in different lighting conditions and angles.
- YOLOv8m run 2: ChessRed2K dataset, which contains 2000 images of chess pieces in various conditions.
- YOLOv8m run 3: Starting from run 1 we used ChessRed2K dataset for finetuning the model.

The YOLOv8m model performed well, achieving a mean Average Precision (mAP) of approximately 0.87 at IoU threshold 0.5 and a mAP of approximately 0.77 at IoU thresholds from 0.5 to 0.95.

As we can see in the table below, the model achieved high precision and recall values across the three training runs, indicating its effectiveness in detecting and classifying chess pieces. All the runs showed converged approximately in 200 epochs.

In terms of loss behavior, we decided to use three different loss functions:

TABLE I
KEY PERFORMANCE METRICS FOR THE THREE YOLOV8M TRAINING RUNS.

Run	Precision (B)	Recall (B)	mAP@0.5	mAP@0.5:0.95
YOLOv8m Run 1	$\sim$0.88	$\sim$0.93	$\sim$0.87	$\sim$0.75
YOLOv8m Run 2	$\sim$0.90	$\sim$0.95	$\sim$0.89	$\sim$0.78
YOLOv8m Run 3	$\sim$0.87	$\sim$0.91	$\sim$0.85	$\sim$0.72

- Classification Loss: it measures the accuracy of the model predictions in classfiying the class inside the bounding box. For this loss YOLOv8m uses a binary cross-entropy loss function.
- Box Loss: it measures the accuracy of the model in predicting bounding box positions and sizes. It's calculated with a combinantion of IoU (Interection over Union).
- Distribution Focal Loss: this loss function is an advanced feature of YOLOv8. It measures the accuracy of the model in identifying the presence of objects inside the image. It's used to handle better small objects.

TABLE II
FINAL LOSS VALUES FOR THE THREE YOLOV8M RUNS. ALL VALUES ARE APPROXIMATE AND TAKEN AT CONVERGENCE (STEP $\sim$200).

Run	Classification Loss (cls_loss)	Box Loss (box_loss)	DFL Loss (dfl_loss)
YOLOv8m Run 1	$\sim$0.06	$\sim$0.35	$\sim$1.10
YOLOv8m Run 2	$\sim$0.05	$\sim$0.32	$\sim$1.05
YOLOv8m Run 3	$\sim$0.07	$\sim$0.38	$\sim$1.15

Compared to earlier training runs and attempts with DETR that showed high loss values and poor convergence, YOLOv8m performed significantly better in terms of loss values and detection robustness. We decided to use the third run of YOLOv8m as our final model for chess-piece object detection, as it showed the best performance in terms of precision, recall, and mAP values.

V. FEN ANNOTATION METHOD

- Alle Luca

VI. IMAGE SIMILARITY SEARCH

- Alle

VII. CONCLUSION

- Luca

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[4] K. Elissa, "Title of paper if known," unpublished.
[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

images/PR-runs-obj.png

Fig. 1. Precision-Recall curve for YOLOv8m. Yellow: Run 1, Green: Run 2, Red: Run 3.

[8] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” 2013, arXiv:1312.6114. [Online]. Available: https://arxiv.org/abs/1312.6114

[9] S. Liu, “Wi-Fi Energy Detection Testbed (12MTC),” 2023, gitHub repository. [Online]. Available: https://github.com/liustone99/Wi-Fi-Energy-Detection-Testbed-12MTC

[10] “Treatment episode data set: discharges (TEDS-D): concatenated, 2006 to 2009.” U.S. Department of Health and Human Services, Substance Abuse and Mental Health Services Administration, Office of Applied Studies, August, 2013, DOI:10.3886/ICPSR30122.v2

[11] K. Eves and J. Valasek, “Adaptive control for singularly perturbed systems examples,” Code Ocean, Aug. 2023. [Online]. Available: https://codeocean.com/capsule/4989235/tree