IMAGE DETECTION OF HUMAN ACTIVITIES BASED ON YOLO V8

Chen Yang, 19th Dec 2023

1 Introduction

This project aims to leverage the advanced capabilities of YOLO V8 (Redmon et al., 2016), a state-of-the-art deep learning model, to identify and classify various human activities captured in images. (Bochkovskiy, 2020) The dataset, sourced from Kaggle and refined using the Labelme user interface, encompasses various human activities, providing a robust foundation for model training and testing.

The project involves meticulous dataset preprocessing, including data cleaning, normalization, and augmentation, to ensure optimal model performance. Key aspects of the YOLO V8 model are explored in detail, highlighting its architectural strengths and suitability for real-time activity detection. The training process adheres to rigorous standards set by Ultralytics documentation, with specific emphasis on configuration parameters like data settings, epochs, batch sizes, and device utilization. Model optimization forms a critical part of the study, where different iterations of the YOLO model are compared, and performance is analyzed through curve analysis. The testing phase addresses challenges in data acquisition and annotation, providing insights into the model's performance under varied conditions.

The report culminates in a discussion of the model's potential applications in fields such as smart home devices, human-computer interaction, sports performance analysis, and video surveillance.

2 Dataset

2.1 Dataset Selection

The selection of the dataset was a crucial step in our project, aimed at ensuring a comprehensive and realistic representation of human activities. We chose a dataset from Kaggle (Rapolu, 2022), renowned for its diversity and reliability in machine learning datasets. This particular dataset was selected due to its extensive range of human activities captured in varied environments, offering a rich source of data for training our model. Additionally, the Labelme (Wada, n.d.) user interface (Figure 1) was employed to refine and annotate the dataset further, allowing for more precise model training. This combination of a robust dataset from Kaggle and the meticulous annotation through Labelme ensured that our model would be trained on high-quality, diverse data, crucial for its success in accurately detecting human activities.

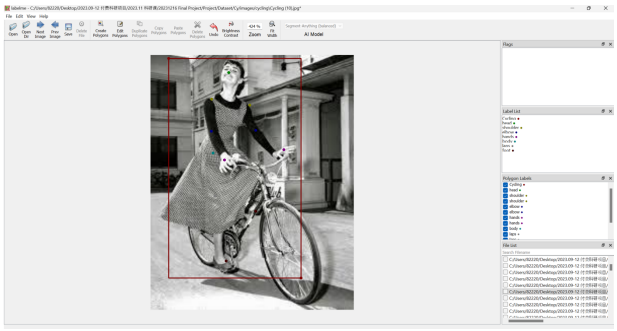

Figure 1: *Labelme user interface showing mark-up of key points.*

2.2 Data Pre-processing

Our data preprocessing involved several coding techniques to optimize the dataset for YOLO V8 training. We first cleaned the dataset, removing corrupt or irrelevant images. Image normalization followed, standardizing dimensions and pixel values. Manual annotation with Labelme and a custom script conversion ensured compatibility with YOLO V8. Finally, Python scripts were used to transform the .json file that include all boxes and key points data to .txt file, which can be recognized by YOLO model.(Figure 2, 3) We also splitted the dataset into balanced training, validation, and test sets, crucial for unbiased model training and evaluation.

3 Methods

3.1 Training

The training of our YOLO V8 model for human activity detection was a meticulous process, designed to optimize the model's accuracy and efficiency. We undertook several key steps, leveraging advanced coding techniques and deep learning principles.

We utilized GPUs to accelerate the training process, as they significantly reduce training time.(Chaudhari

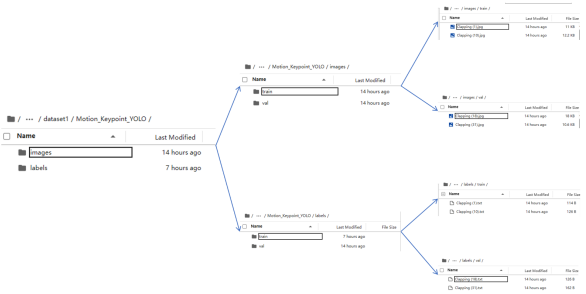

Figure 2: *Sort the data structure*

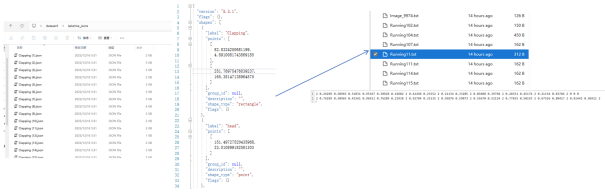

Figure 3: *A glance at the Python scripts to transform .json to .txt file*

et al., 2020) The training was conducted using Python, with the PyTorch framework, which offers extensive support for convolutional neural networks like YOLO.

The configuration of the model was aligned with the guidelines provided by Ultralytics. YOLO V8's architecture was employed, known for its ability to perform object detection tasks efficiently. (Chaudhari et al., 2020) Its architecture includes several convolutional layers that help in feature extraction and detection. We monitored the model's performance through each epoch using validation data. Loss functions, specifically a combination of mean squared error (for bounding box predictions) and cross-entropy loss (for class predictions), were used to calculate the error during training. Minimizing these losses was key to improving model accuracy.

Performance metrics such as precision, recall, and F1-score were used to evaluate the model after each epoch. These metrics provided insights into the model's detection capabilities. If the model's performance did not improve or plateaued, hyperparameters were adjusted.

Through these steps, we trained our YOLO V8 model to achieve a certain level of accuracy in detecting human activities in images. The process ensured that the model was not only accurate but also efficient and robust in various real-world scenarios.(Adeola, 2023)

3.2 Optimization

Optimizing the YOLO V8 model was a critical phase, focusing on enhancing its performance and accuracy in detecting human activities. We employed several strategies for this purpose:

We experimented with different hyperparameters, including learning rates, batch sizes, and the number of epochs, to find the optimal settings that minimized loss and maximized accuracy. We continuously monitored the model using precision, recall, and F1 scores. This iterative evaluation allowed us to identify areas needing improvement. A combination of loss functions was used to effectively train the model. We fine-tuned these functions to better align with our specific objectives in human activity detection. We also trained the data based on two YOLO models - one is YOLO v8n and the other one is YOLO v8x- for comparison and optimization. (Figure 4)

Through these optimization techniques, we significantly improved the model's ability to accurately and efficiently detect a wide range of human activities, ensuring robust performance under various conditions.

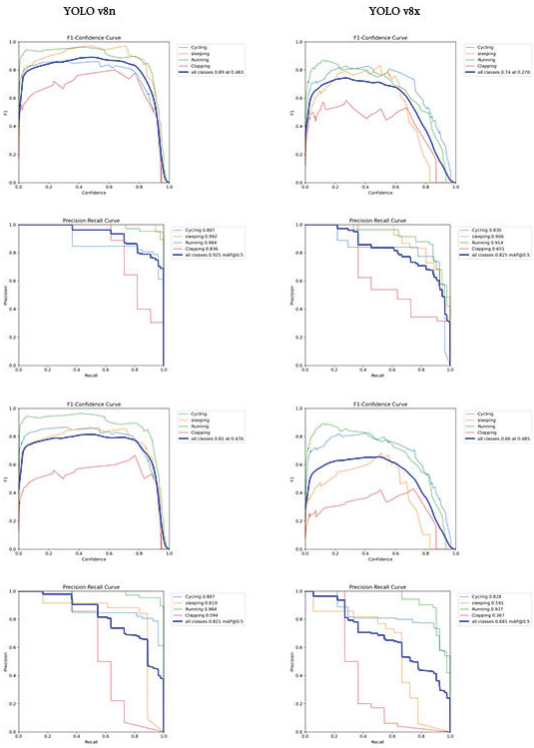

Figure 4: *Optimization of results comparison based on two YOLO models*

3.3 Test And Deployment

Testing the YOLO V8 model was a crucial stage to validate its efficacy in real-world scenarios. We employed a dedicated test dataset, distinct from the training and validation sets, to evaluate the model's performance. This dataset included a diverse range of images featuring various human activities, ensuring a comprehensive assessment. During testing, we focused on the model's ability to accurately detect and classify activities, paying close attention to its precision, recall, and

overall accuracy. We also encountered challenges such as varying image angles and lighting conditions, which provided insights into the model's robustness. And as Figure 5, 6, 7 shows, YOLO v8x and v8n were showing significant difference on the accuracy of the final results, where for the image on the column one and row 2, YOLO v8x successfully recognized it as a person clapping while YOLO v8n did now. The testing phase highlighted areas for further refinement and demonstrated the model's potential in practical applications. We also deployed our model on local computers to use cameras to run real-time human activities detection based on our trained model.(Figure 8)

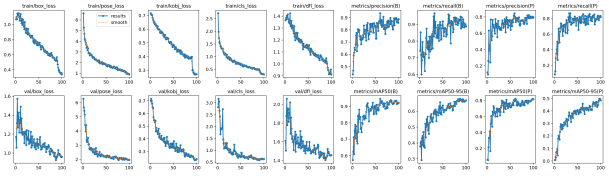

Figure 5: *Results of test on YOLO v8n*

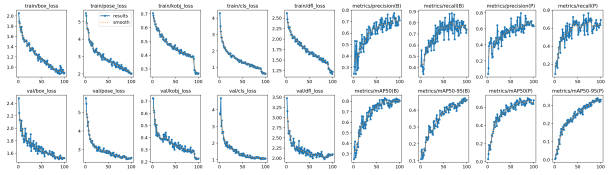

Figure 6: *Results of test on YOLO v8x*

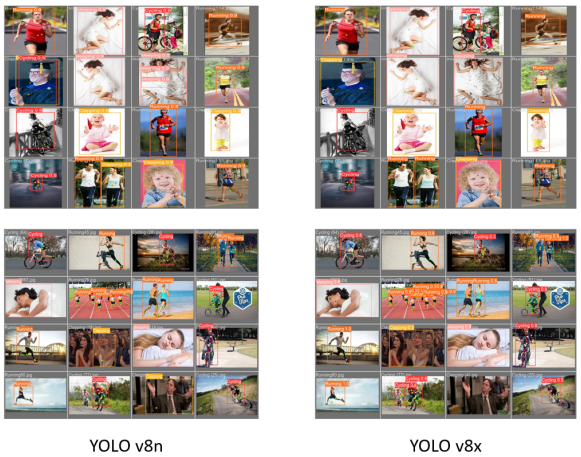

Figure 7: *Results of testing images and the model's confidence*

Figure 8: *Local deployment of the model to use computer cameras to detect real-time human activities.*

4 Future Prospect

The successful implementation of the YOLO V8 model for human activity detection opens a plethora of prospects in both academic and practical realms. In academic research, this project can act as a foundation for further exploration into more complex activity recognition and real-time processing challenges. Practically, the model has significant potential in enhancing smart home systems, enriching human-computer interaction experiences, and contributing to advanced sports performance analysis. Furthermore, its application in public safety and surveillance systems could revolutionize how environments are monitored, offering more efficient and accurate anomaly detection.(Redmon et al., 2016)

References

Adeola, O. (2023). "Development of a fake news detection model using decision tree algorithm". In: *Advances in Multidisciplinary Scientific Research Journal Publication* 2 (2), pp. 81–88. DOI: 10.22624/aims/csean-smart2023p10.

Bochkovskiy, A. (2020). "Yolov4: optimal speed and accuracy of object detection". In: DOI: 10.48550/arxiv.2004.10934.

Chaudhari, S. et al. (2020). "Yolo real time object detection". In: *International Journal of Computer Trends and Technology* 68 (6), pp. 70–76. DOI: 10.14445/22312803/ijctt-v68i6p112.

Rapolu, Shashank (2022). *Human action recognition dataset*. URL: https://www.kaggle.com/datasets/shashankrapolu/human-action-recognition-dataset/data.

Redmon, J. et al. (2016). "You only look once: unified, real-time object detection". In: DOI: 10.1109/cvpr.2016.91.

Wada, Kentaro (n.d.). *Labelme: Image Polygonal Annotation with Python*. DOI: 10.5281/zenodo.5711226. URL: https://github.com/wkentaro/labelme.