Object Detection and Classification for Clash of Clans

1st Andrei Pavlov
DS-01
Innopolis University
Innopolis, Russia
an.pavlov@innopolis.university

2nd Bulat Latypov
AAI-02
Innopolis University
Innopolis, Russia
b.latypov@innopolis.university

***Abstract*—In this paper authors suggest object detection and classification for computer games and its specifically usage in Clash of Clans, where computer vision methods can be used for identifying buildings structures of players' bases.**

***Index Terms*—Computer Vision, Object Detection, Object Classification, Computer Games, Clash of Clans**

I. Introduction

With the increasing popularity of computer vision (CV), its applications are expanding across various industries, including gaming. CV techniques offer innovative ways to enhance player experiences by enabling real-time analysis and interaction within virtual environments. In this paper, we explore the use of object detection and classification specifically within the context of the popular mobile strategy game Clash of Clans. By employing advanced computer vision methods, the system can accurately identify and classify structures in players' bases, providing new opportunities for automation and strategic planning.

II. Related Work

Object detection and classification has grown significantly over past years, especially with their combination with deep learning algorithms, such as YOLO. Old methods of object detection, such as SIFT, has shown poor results, especially when it comes to work with complex datasets.

The interest of object detection researchers to computer games is also growing. Computer vision is used to take an information from the screen and use it, for example, for calculating position of a car relative to turns in racing games or extracting actions of enemies in Dota 2 or StarCraft.

The lack of open-source datasets for gaming applications has been a bottleneck in this field. Researchers have often created their own custom datasets, a process that takes a lot of person-hours. For example, model given in website https://huggingface.co/keremberke/yolov5s-clash-of-clans has 13 classes of buildings, while their total amount is more than 1900.

In summary, while significant steps have been made in object detection, gaming remains an unexplored field with huge potential. Our work builds upon the principles stated by previous researches and gaining the strengths of having a complete dataset.

III. Method

The goal of our project is to detect and classify buildings in Clash of Clans player bases using computer vision techniques. We chose the YOLO (You Only Look Once) algorithm. It is suitable for analyzing and detecting images given in our dataset.

Since no public dataset is available for this task, we manually collected and annotated screenshots of Clash of Clans bases. Each building was labeled with its type, such as Cannon, Archer Tower, or Town Hall. This process ensured that the dataset included a variety of base layouts. Also, sprites of all towers was parsed from Clash of Clans fans website. After that, some actual players' bases were parsed and labeled manually.

The collected images were annotated individually to meet the requirements of the YOLO model.

The database is published on Roboflow. It can be imported to your project manually or downloaded using roboflow library:

```
from roboflow import Roboflow
rf = Roboflow(api_key="API_KEY")
project = rf.workspace("coc-2fz7c").project("coc-
    object-detection-ypfix")
version = project.version(4)
dataset = version.download("yolov11")
```

The YOLOv11 was trained on the annotated dataset. During training, the model learned to identify buildings and predict their bounding boxes and classes. mAP (mean Average Precision) is the main metrics to evaluate the model's performance. Actual version of the model is trained and published on Roboflow as website provides free computing resources. It can be imported to your project using docker or "inference" library in python:

```
model = inference.get_model(model_id="coc-object-
    detection-ypfix/4", api_key="API_KEY")
```

Examples of importing and usage are provided in example.ipynb file in Github repository.

Also, the model training can be applied on local machine using "ultralytics" and "roboflow" libraries:

```
!yolo train model=yolo11n.pt data={dataset.location
    }/data.yaml epochs=250 imgsz=640
```

The detailed code of training with metrics and examples is provided in training.ipynb file in Github repository.

IV. Github link

https://github.com/IAndermanI/coc_object_detection

V. Experiments and Evaluation

To evaluate the performance of the YOLOv11 model, we conducted experiments on a test set of 43 screenshots. These experiments measured the model's ability to detect and classify various building types in the game. Our evaluation focused on key metrics such as mAP (mean average precision), precision, and recall. Below are the main findings from our experiments:

1) **mAP:** The model achieved an impressive precision of 97.9%, successfully identifying the majority of buildings on different bases.
2) **Precision:** precision of 91.9% - model correctly assigned labels to most detected buildings, including difficult cases like overlapping structures or rotating objects, such as cannon or mortar.
3) **Recall:** The recall of 96% is also an exceptional result meaning that model is able to differentiate objects from each other.

VI. Analysis and Observations

We tested the model on a separate set of screenshots to measure its detection and classification accuracy. The results are demonstrating the effectiveness of our approach in identifying buildings quickly and accurately.

All metrics are on the appendix session (Figures 1-4).

VII. Timeline with each member individual task.

Each members tasks:

- Andrei Pavlov: downloading, labeling data and making dataset
- Bulat Latypov: applying and testing computer vision algorithms

References

[1] Clash of Clans wiki website: https://clashofclans.fandom.com/wiki/Defensive_Buildings/Home_Village
[2] Github page of the project: https://github.com/IAndermanI/coc_object_detection
[3] Project on Roboflow: https://universe.roboflow.com/coc-2fz7c/coc-object-detection-ypfix/model/4
[4] Previous research: https://huggingface.co/keremberke/yolov5s-clash-of-clans

VIII. Appendix

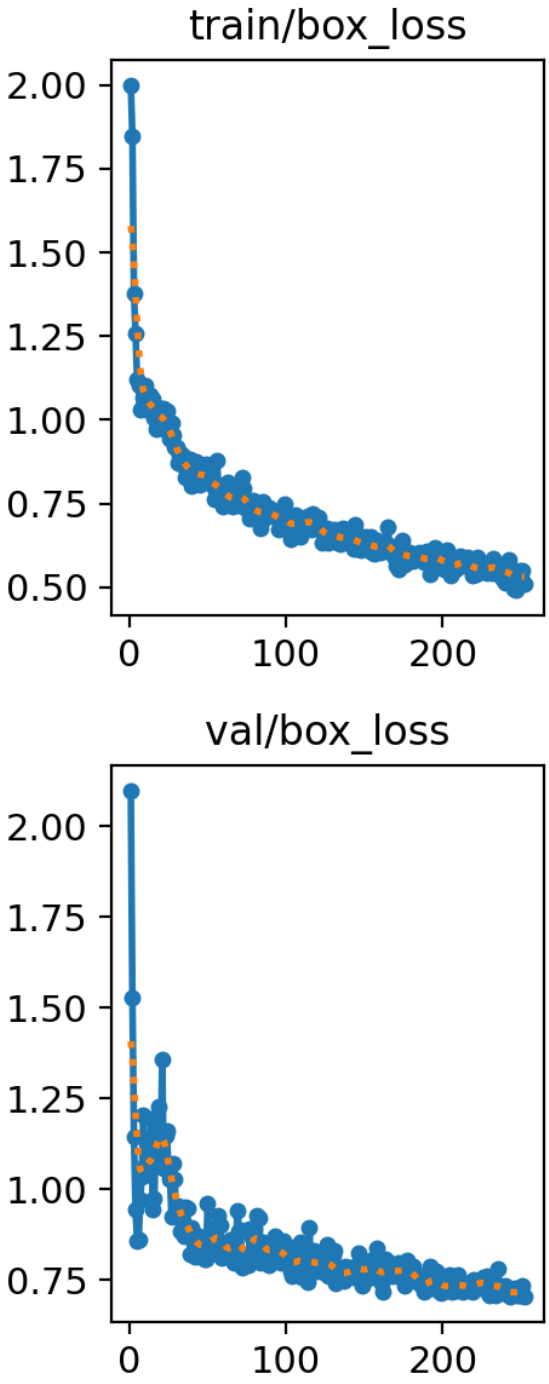


Fig. 1. Test and validation box loss

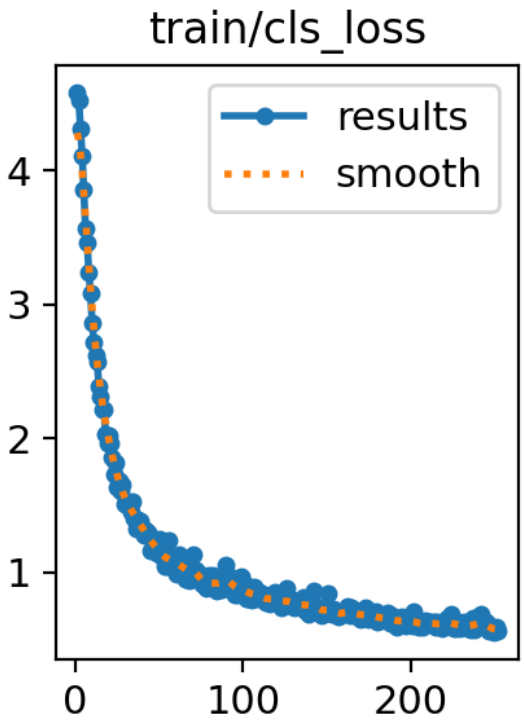
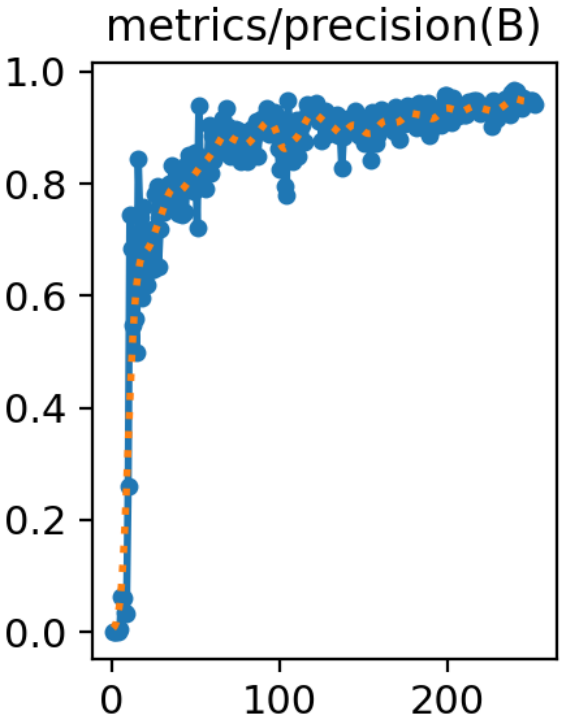


Fig. 2. Test and validation class loss



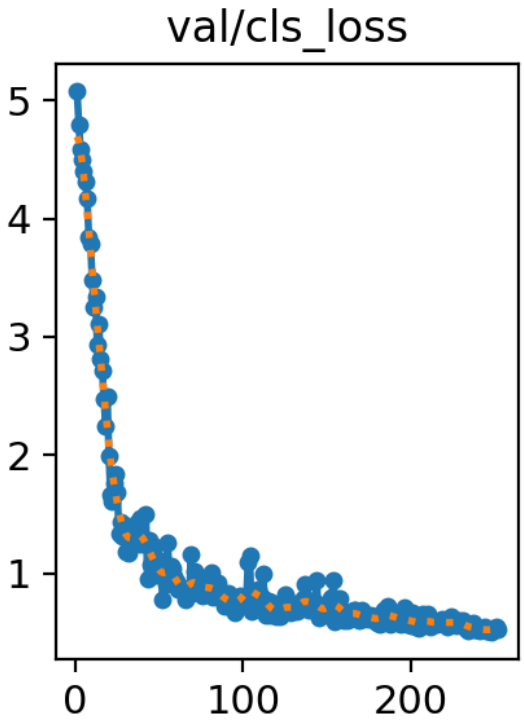
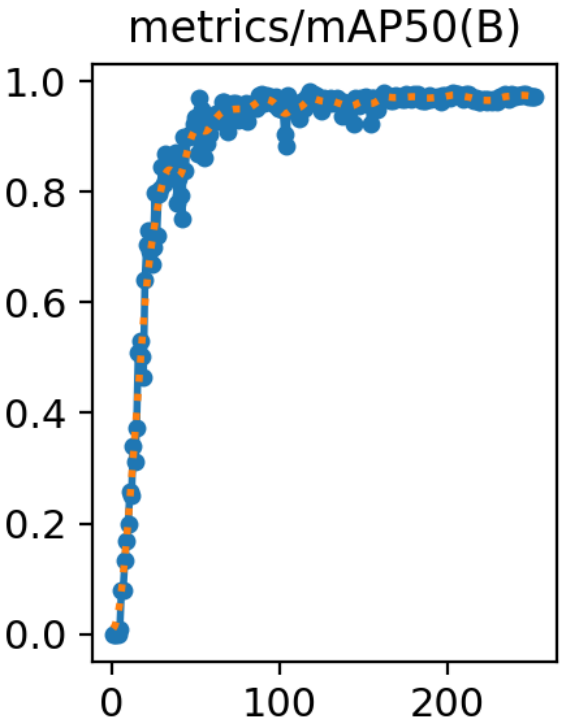


Fig. 3. Test and validation precision



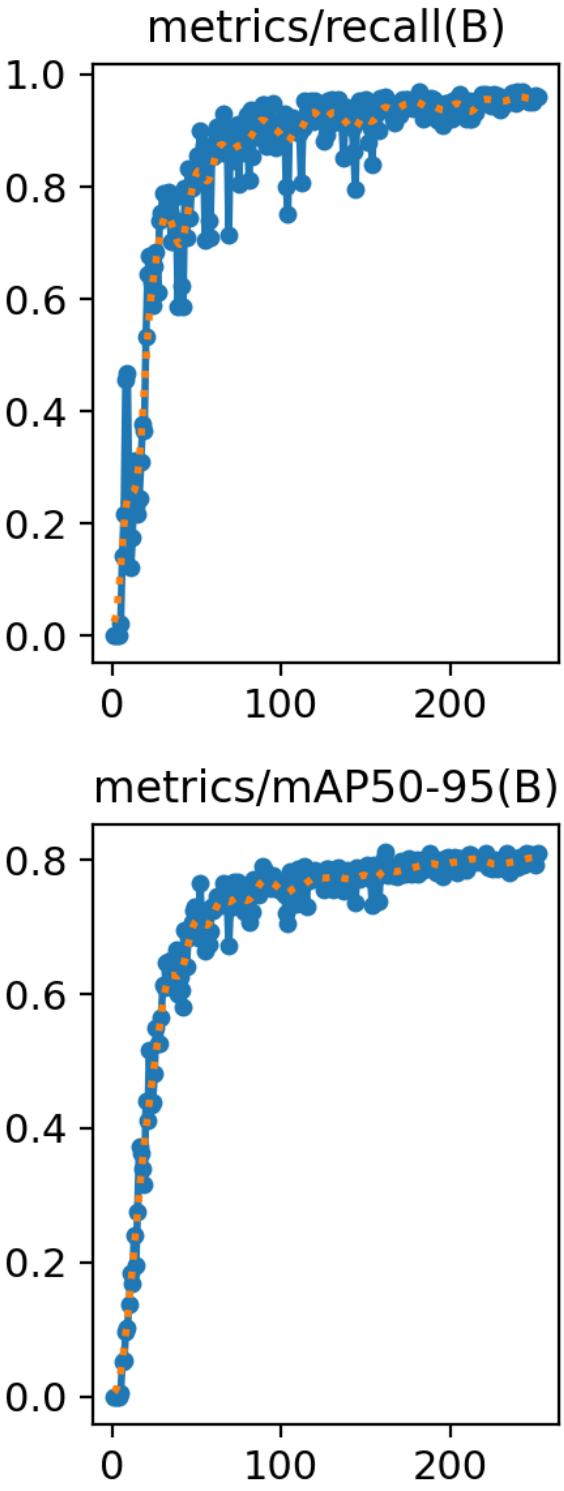


Fig. 4. Test and validation recall

Fig. 5. CoC base with marked buildings that was found by the model

Fig. 6. Another CoC base with marked buildings