

Methodology for the YOLOv8 Image Recognition Project

The goal was to use YOLOv8 to recognize two new images: a person and an object. The general steps consisted of cleaning the data, adding new images and labels to the data, splitting the data, training the model, and then predicting with the model.

For the cleaning step, there were two images and two labels in the 'coco128' folder that had a missing counterpart and therefore needed to be removed. Afterwards, we used our camera to take new images and draw boxes around them so as to generate text files with labels using the tool from <https://github.com/ivangrov/ModifiedOpenLabelling>. The new classes started from 80 and 81, where the former was a person (myself in this case) and the latter was an object (a cup of ramen in this case). Then, we split the data based on an 80/20 training-validation ratio using the provided code in 'yolov8_ee104_split_train_val_files.py'. It was important to keep the 'train2017' folder inside the 'ee104_val' folder in order for this to work properly.

Finally, the training and predicting were performed from these two commands:

```
yolo task=detect mode=train  
data=C:/Users/USERNAME/datasets/coco128/coco128_ee104.yaml model=yolov8n.pt  
epochs=100
```

```
yolo task=detect mode=predict  
model=C:/Users/USERNAME/ultralytics/runs/detect/train/weights/best.pt source=0 show=True
```

Documentation for dragons.py

This is a Python program that implements a two-player game where the players navigate through a dungeon to collect eggs. The dungeon contains three lairs with sleeping dragons guarding the eggs. The objective of the game is to collect a certain number of eggs without losing all lives or waking the dragons.

The code imports the necessary modules `math` and `random`. The constants are defined, which include the width and height of the game window, the center coordinates of the window, the color of the text, the number of eggs to collect, the starting positions of the two heroes, the attack distance, the dragon wake time, the egg hide time, and the move distance of the heroes. The global variables are also defined, which include the number of lives, eggs collected, and the game over status.

The program creates three dictionaries to represent the lairs in the dungeon. Each dictionary contains a sleeping dragon, a number of eggs to collect, and timers to control the dragon's sleep and egg hiding. The lairs are brought into the game, along with the two heroes.

The `draw()` function is defined to draw the background, lairs, and heroes, and display different messages depending on the game state. The `draw_lairs()` function is called within `draw()` to draw the lairs and hide the eggs depending on the timer conditions. The `draw_counters()`

function is also called to display the number of eggs collected and the number of lives remaining.

The `update()` function is defined to enable player movement and perform collision checks.

The `update_lairs()` function is also called to control the dragon's sleep and egg hiding. The `check_for_collisions()` function is called within `update()` to check for collisions between the heroes and the dragons, and between the heroes and the eggs.

Overall, this program creates a simple two-player game with a dungeon, dragons, and eggs.

The game mechanics involve navigating the heroes through the dungeon to collect eggs without waking the sleeping dragons or losing all lives.