

Project Report №3

Team: GroShi

Members:

Arsen	Mutalapov	a.mutalapov@innopolis.university
Gleb	Kirillov	g.kirillov@innopolis.university
Ruslan	Kudinov	r.kudinov@innopolis.university

Project topic: Grocery object detection.

Github: https://github.com/system205/PMLDL_project

GitHub with deployment:

<https://github.com/system205/GroceryObjectDetectionTelegramBot>

Current status

1. We **deployed** the YOLOv8 small model locally in Docker with GPU. For that, we used the [Roboflow Inference Server](#).
2. We **saved** the trained **weights** on GitHub in the “train” folder.
3. We created an [InnoGrocery Telegram Bot](#) that **accepts users’ photos** and says what products our model recognizes on them. Also, the time of inference and request processing is measured and output.
4. We prepared **a single command** (docker-compose up) to run the bot with the server to make inferences locally.
5. We **collected more images**. We identified that our previously trained model performed on items wrapped with **cellophane** poorly.
6. We **explored** our dataset more. We analyzed the **area of masks and bounding boxes** of our marked objects.
7. According to the graph in the appendix, we understood the final production-ready **confidence level** for our model: 0.715
8. Visualized training with **TensorBoard**

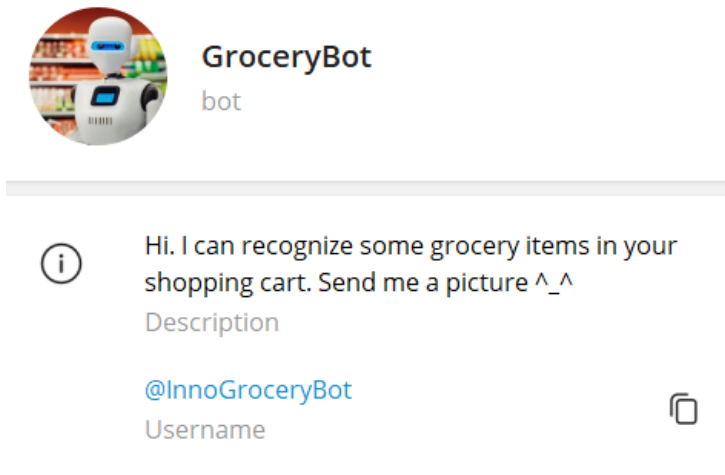
Future work

As we have already shown that YOLOv5 is worse than YOLOv8 we won’t compare them more. Instead, we will try to train and deploy larger versions of YOLOv8. So far, we have a small one that has

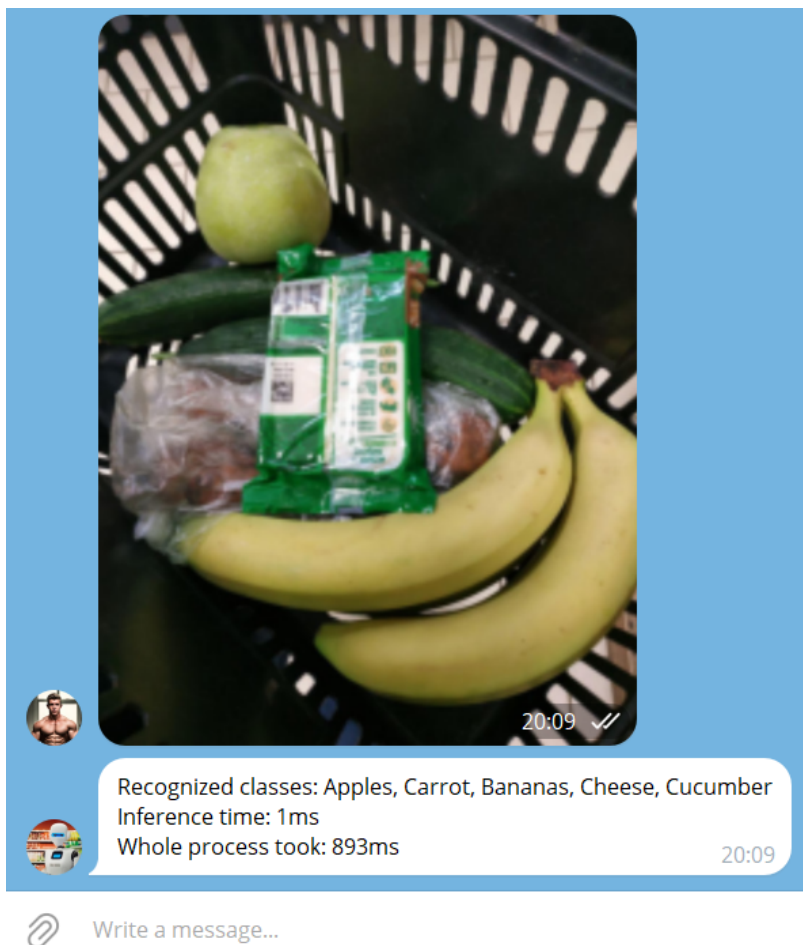
11.2M parameters and 28.6B FLOPs. We will test medium and large versions that have twice and four times more parameters. Finally, we write a bit more in README files.

Appendix

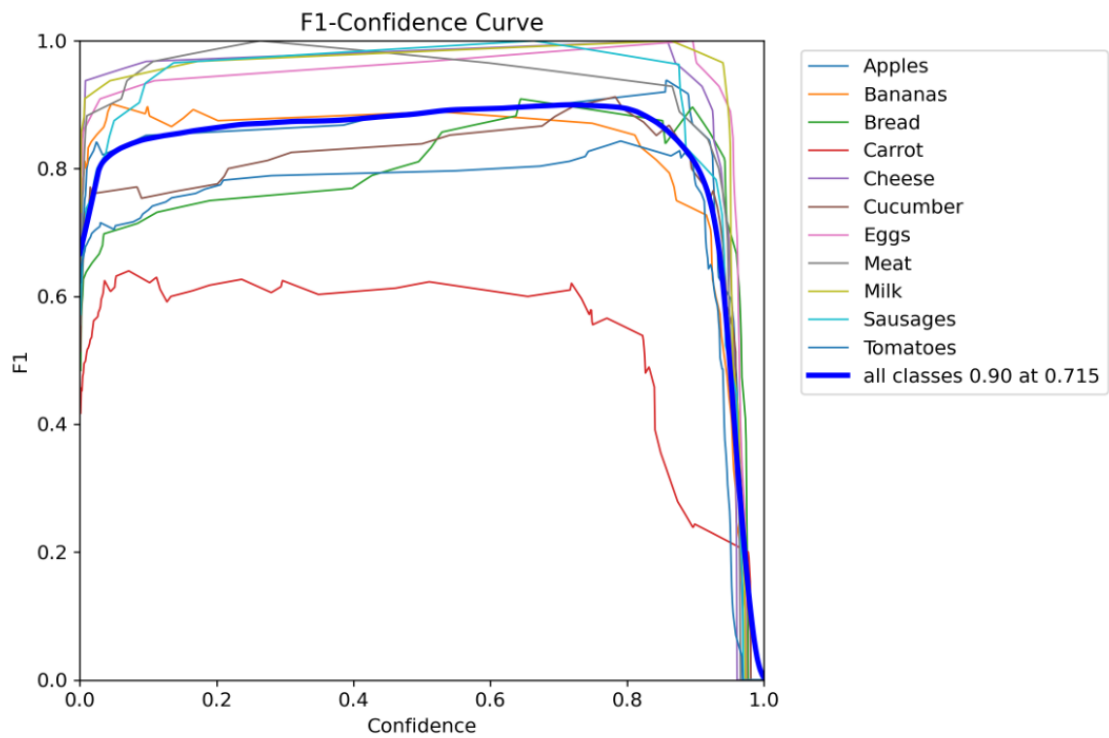
Telegram bot



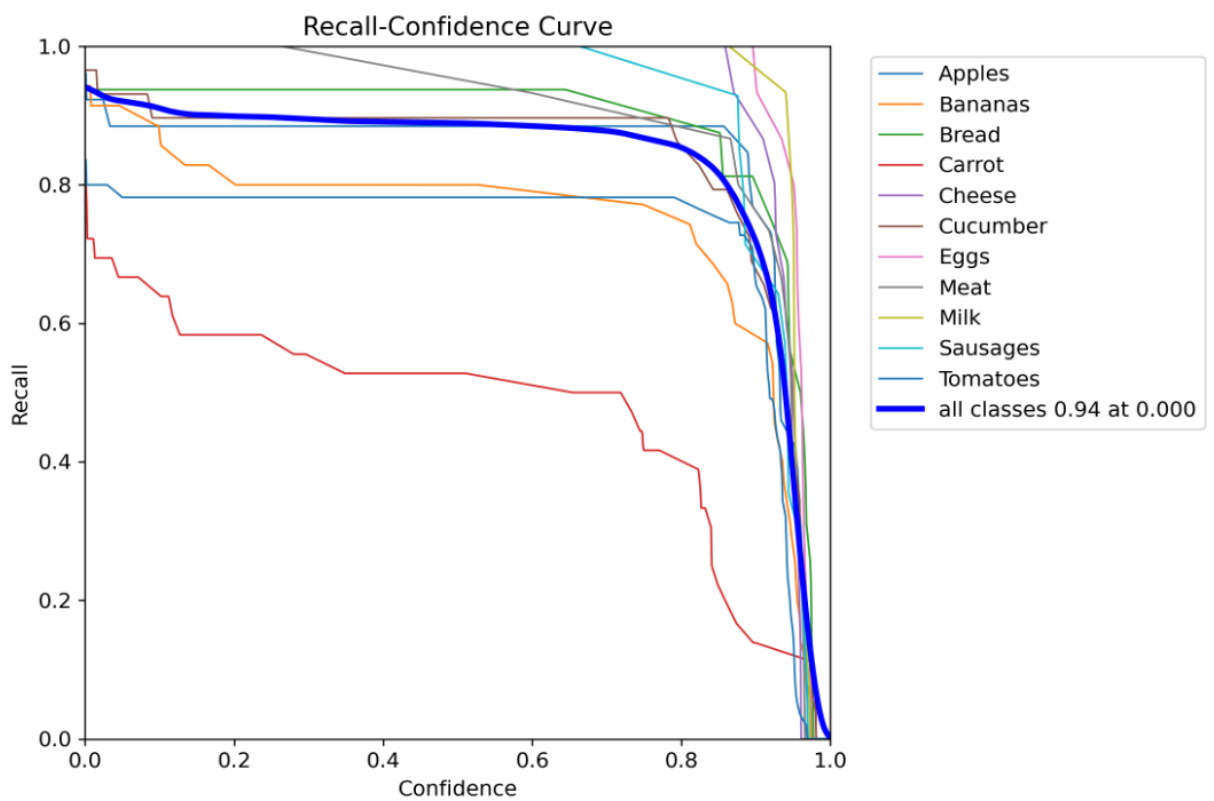
Example of interaction



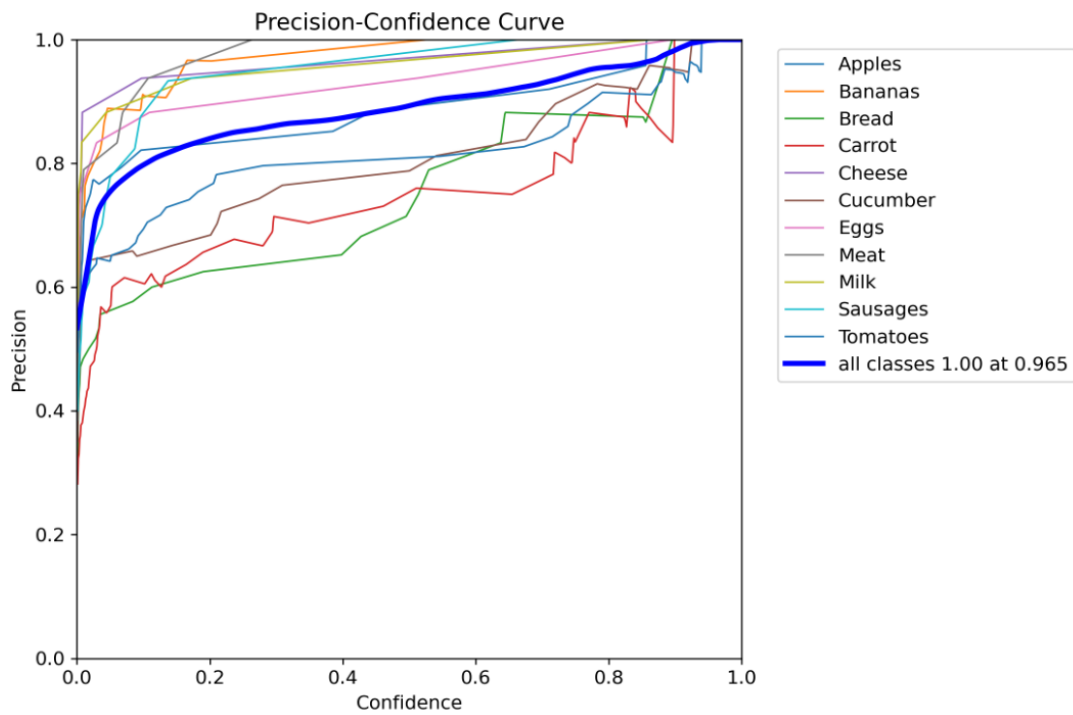
Graph with confidence to F1



Graph with confidence to recall



Graph with confidence to precision



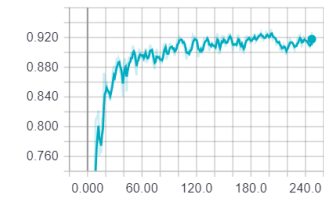
New inference example on an expanded dataset



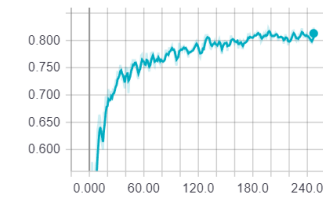
TensorBoard training:

metrics

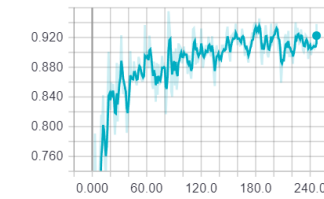
metrics/mAP50(B)



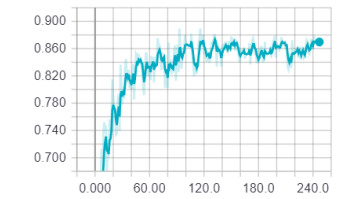
metrics/mAP50-95(B)



metrics/precision(B)

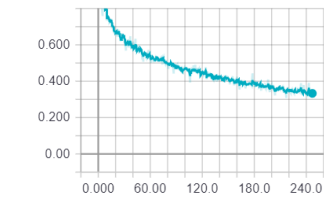


metrics/recall(B)

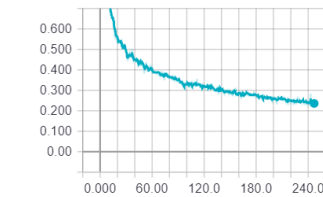


train

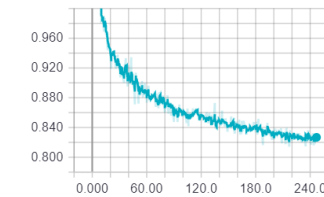
train/box_loss



train/cls_loss

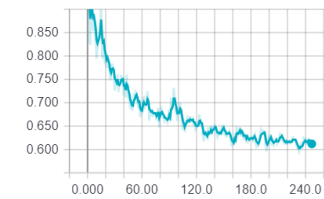


train/dfl_loss

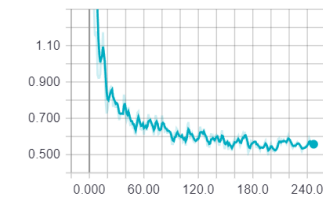


val

val/box_loss



val/cls_loss



val/dfl_loss

