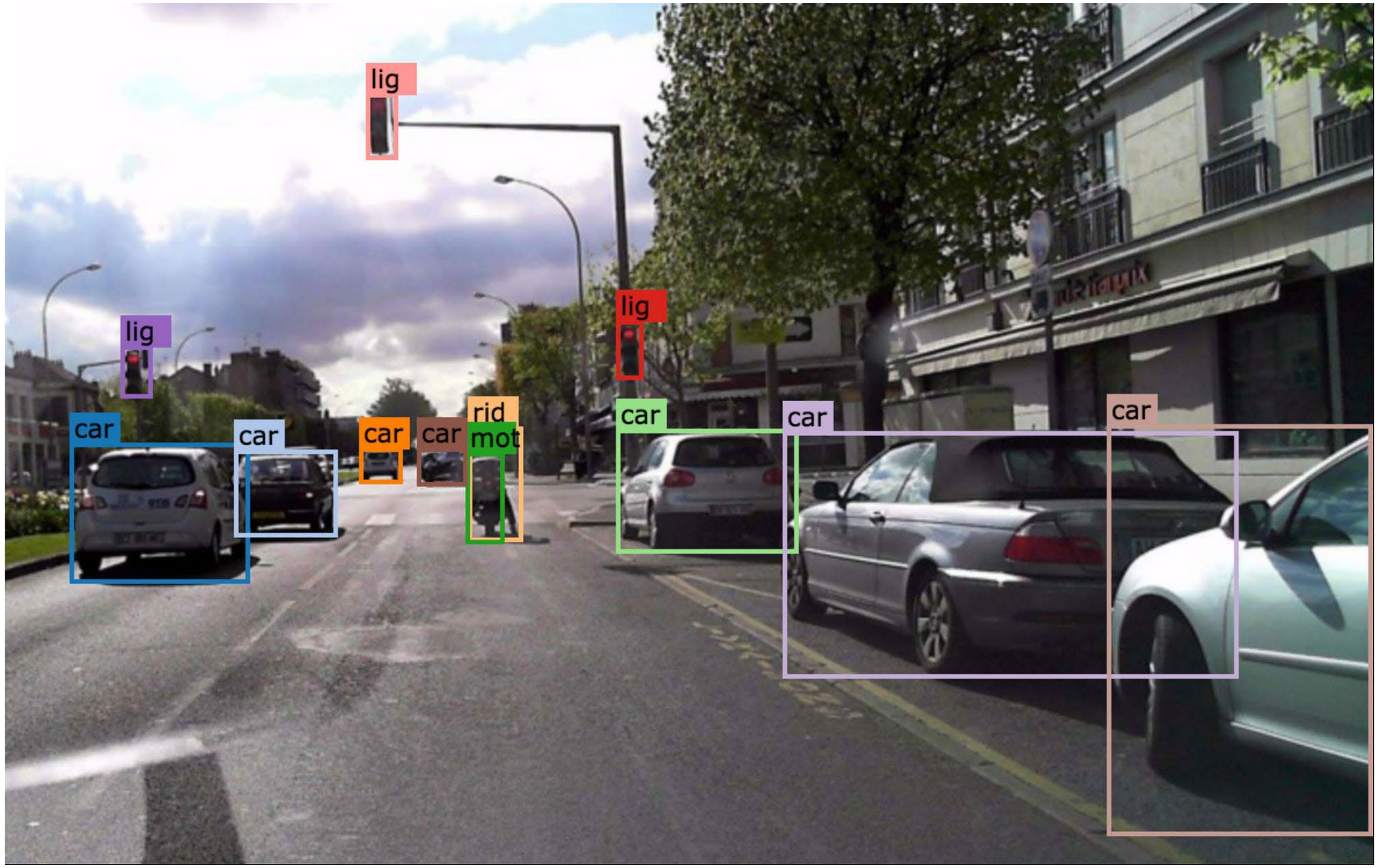
OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

YOLO Object Detection for Jetson Nano

EE5104 mini-project

Spring 2024

60% of final grade



Assignment Overview

- In this project, you will:
  1. Take a YOLO model train it on the BDD100k dataset.
  2. Comment on the model performance and then implement an optimized model that can be run on the Jetson Nano.
  3. Run the model in real-time on a camera stream.
  4. Implement a people counting algorithm as extra credit.

Instructions (1) – Building your model.

- Building your own YOLO model:
  - You are free to choose whatever YOLO model you want.
  - You may take existing repos and modify them for this application. However, you cannot use a "blackbox" implementation e.g. "from ultralytics import yolo"
    - No pre-built models or complete repos
  - You can follow tutorials or resources online – but this must be fully referenced in your report.
  - Feel free to use whichever Python libraries you wish.

Instructions (2) – Training your model.

- Training the model on the B1100k Dataset:
  - You can either train locally/Kaggle, and port the model onto the nano or train the model on the nano.
    - You must comment on this decision in your report.
  - The BDD100k dataset contains 10 categories for object detection, tasks. Train your implementation to detect these classes
  - The dataset can be downloaded from Kaggle or the BDD100k website.

Instructions (3) – Training your model cont'd.

- Comment on your model performance:
  - Training loss & accuracy, validation loss & accuracy.
  - Precision, recall, F1 score, and confusion matrix.
  - Hint: programs like weights and biases (www.wandb.com) might be useful
  - If needed, you can further improve model performance by methods described in assignment 1.
- Once you are happy with the model, you must optimise it so it can run on the Jetson Nano.
  - Hint: research Nvidia optimization tools
- Comment on the optimised model performance:
  - Model performance, accuracy, and model size.

Instructions (4) – Inference on Jetson Nano.

- □ If your model is trained locally, then you must transfer the model onto the Jetson Nano.
- □ The model must run in real-time on a camera stream.
- □ You can use a PI Cam or a webcam.
- □ The implementation on the Jetson Nano must:
  - Correctly detect and classify all 10 BDD100k classes
  - Run at 30 FPS.

Instructions (5) – Extra Credit.

- For extra credit you can implement a simple people counter.
- The implementation is up to your design. Your implementation must:
  - Correctly count the people in the room in real-time.
  - Increment and decrement the counter as people enter & leave the scene.
  - Have the count values displayed on the stream output.
- As a starting point a basic implementation consists of:
  - Object Detection -> Assign Unique ID to each object -> Tally each unique ID.
- Marks will be going for creative implementations and robustness.

Jetson Nano Kit

- Each student will be presented with a Jetson nano kit, containing:
  - 1 Jetson Nano 4gB board
  - 1 x Power Supply
  - 1 x Picam
  - 1 x HDMI cable
  - 1 x USB wifi adapter
  - 1 x 64gB microSD card

Jetpack OS

- The Jetson nano runs on a customized Ubuntu OS
- You will be provided with a baseline image for this project. This baseline image comes with the following:
  - A Python 3.6.9 virtual environment
  - OpenCV 4.5.3 (cuda enabled)
  - YoloV5 (revision 5.0)
  - TeamViewer (remote desktop application)
  - Example OpenCV and Yolo scripts
  - A python script to connect to eduroam
- It is highly recommended that you use the provided image for this project

Submission & Grading (1)

- This project is worth 60% of your overall grade.
  - 10% for the live demo.
    - Graded on visual quality of implementation. No bugs, glitches, etc.
    - Run-time optimization and efficiency.
    - Model performance, robustness and reliability.
  - 45% for the code and submitted report.
  - 5% for extra credit.

Submission & Grading (2)

- Submit your final working code as a .py file and a report summarizing your project in a single zip file.
- Format the submission as: firstname_surname.zip
- Ensure your code is working, well formatted and commented
- The report shall outline:
  - Details of your implementation, libraries used etc
  - Details of real-time optimization for the Jetson nano
  - Suggestions for future improvement
  - A bibliography (external code, libraries)

Support

- □ Dedicated sessions will be organized to help students with startup/board setup issues
- □ Use the Canvas EE5104 discussion board for help and support – do not send direct emails!!!!
- □ Students are encouraged to actively participate in the Canvas Discussion Boards. Post problems, share solutions, advice etc

Advice

- □ Use virtualenv or anaconda for development on the nano.
- □ Back up your code and SD card often.
- □ Collaborate with your peers & ask for help during support sessions and class.
- □ Start simple – implement a basic version that works, before attempting a more complex/better solution
  - Stick to the configuration provided
  - It’s easy to “break” your build environment on these devices

Good luck!