Senior Computer Vision Engineer - Technical Assignment: Car Part Segmentation

Thank you for your interest in the Senior Computer Vision Engineer role at AUTO1 Group. This technical assignment is designed to simulate a real-world problem our team tackles: segmenting key car parts on vehicles from our inspection devices.

The goal is to assess your practical deep learning skills, your ability to work within production constraints, and your software engineering practices. We expect this task to take approximately 3-6 hours. Please read the instructions carefully.

The Goal

Your primary goal is to develop and train an efficient semantic segmentation model to identify five specific car parts. You will then provide a command-line tool to run inference on a test set of images.

The Dataset

You are provided with a dataset containing the following folders:

- **images/**: 600 training images (`.jpg` format). These are anonymized images from our scans where the background has been removed.
- **masks/**: 600 corresponding ground truth segmentation masks (`.png` format).
- **test_images/**: 68 test images for which you will generate predictions. No ground truth is provided for this set.

The segmentation masks use specific integer values to represent each class:

- **0**: Background
- **32**: Front Door
- **64**: Rear Door
- **96**: Front Fender
- **128**: Rear Fender
- **160**: Door Handle

A Note on Confidentiality: The images provided in this dataset are the property of AUTO1 Group. They are for the sole purpose of completing this technical assignment and must not be shared, published, or distributed publicly on the internet (e.g., on GitHub, personal blogs) or by any other means.

Core Requirements

- **Model Development**: Develop and train a semantic segmentation model of your choice on the provided training data. You can use any existing pre-trained models for this task.
- **Technical Stack**: The entire project must be implemented using Python 3.11+ and PyTorch 2+.
- **Constraints**: Your solution must adhere to the following constraints:
    - **Model Size**: The final trained model file (e.g., `.pth`) must be under 180 MB.
    - **Inference Speed**: The average inference time per image should be under 1 second. We will use an NVIDIA RTX 5090 as a reference. If you use different hardware, please note it and report your own timing
- **Inference Script**: You must provide a command-line interface (CLI) script that takes a directory of input images and a directory for output masks, and runs inference using your trained model. The command should be structured as follows:

```
python3 inference.py --input path/to/test_images --output
path/to/predictions
```

- **Minimal Dependencies**: Please aim for a minimal set of external dependencies. Only add libraries that are essential for completing the core task. Every dependency should have a clear purpose.
- **Output Format**: The output prediction masks must be saved as single-channel `.png` files. The pixel values in the masks must correspond to the class labels defined above (0, 32, 64, etc.).

Evaluation Criteria

We will evaluate your submission based on the following criteria, in order of importance:

- **Correctness & Performance**: The model's segmentation performance (e.g., mIoU) on our hidden test set and its adherence to the size and speed constraints.
- **Methodology & Justification**: Your choice of model architecture, loss function, data augmentations, and other training parameters. A senior engineer should be able to justify their decisions. This is a key part of the evaluation.
- **Code Quality**: The clarity, structure, and maintainability of your code. We value clean, well-commented, and robust software that uses a minimal and justified set of dependencies.
- **Reproducibility**: The ease with which we can set up your environment (using a `requirements.txt` or `pyproject.toml` file) and run both your training and inference scripts.

Deadline

You are given **1 week** to complete the task.

Deliverables

Please package the following into a single **.zip** file for submission:

- **Source Code**: All Python scripts required to train your model and run inference.
- **Trained Model**: The final trained model file (.pth, .pt, or similar) that meets the size constraint.
- **Test Set Predictions**: A folder named `predictions/` containing the output masks generated by running your `inference.py` script on the `test_images/` directory.
- **Requirements File**: A `requirements.txt` file listing all necessary dependencies.
- **README.md**: A comprehensive `README.md` file that includes:
    - Instructions on how to set up the environment and install dependencies.
    - The exact command to run to train your model.
    - The exact command to run your `inference.py` script.
    - A section explaining your design choices.

We appreciate you taking the time to complete this assignment. It's an important step in our process, and we are excited to see your approach to solving the problem.

Questions

If any part of the task description is unclear, please don't hesitate to ask for clarification.

Good luck! We look forward to seeing your solution.