On-Device Machine Learning Project Proposal

Group: YU; Group members: Yuqing Qin , Yukun Xia

Motivation

Learning-based methods for feature detectors and descriptors have the potential to have better performance than the classical methods. On certain tasks, eg. homography estimation, a deep learning model, SuperPoint, outperforms SIFT, ORB, and LIFT. However, learning-based methods are typically more computationally expensive. On high-end GPUs (i.e. Titan X), real-time operation is possible (i.e. SuperPoint: ~70 FPS), but these GPUs are hardly accessible to mobile applications, eg. AR glasses. To the best of our knowledge, it has not been carefully explored whether lower computing edge devices, such as Jetson Nano, are capable of efficiently running these deep learning detectors and descriptors.

Hypotheses (key ideas)

By deploying the optimization techniques (i.e. quantization and distillation), the SuperPoint network would be able to work on the low compute mobile device(i.e. Jetson Nano) without harming the performance(eg. mean Average Accuracy, and Pose Error) too much on benchmarks (eg. Image Matching Challenge Dataset, and KITTI).

Approach

- **Dataset**: KITTI and Image Matching Challenge Dataset
- **Model**: SuperPoint
- **Optimization**: quantization, distillation (i.e. backbone), distillation + quantization
- **Evaluation (Speed test)**: Calculate FPS for baseline and each optimized model
- **Evaluation (Performance test)**: Benchmark the mean Average Accuracy between two frames and long term pose estimation accuracy

Related work and baselines

Related work

- **Model:** SuperPoint: Self-Supervised Interest Point Detection and Description
- **Benchmark:** Image Matching Across Wide Baselines: From Paper to Practice
- **Benchmark:** Vision meets Robotics: The KITTI Dataset
- UnsuperPoint: End-to-end Unsupervised Interest Point Detector and Descriptor
- SuperGlue: Learning Feature Matching with Graph Neural Networks

Baseline

- Pre-trained SuperPoint running on Jetson Nano, (metrics: inference time, accuracy)
- ORB and SIFT running on Jetson Nano, (metrics: inference time, accuracy)

I/O

- **Inputs**: Images (either monocular or stereo)
- **Outputs**: Keypoint locations and corresponding descriptors
- **Existing tool for I/O**: OpenCV's image reading function for real images, or ROS bag replay for dataset

- **Hardware**:
    - Monocular camera or stereo camera
    - Justification: With a monocular camera only, the translation part of the estimated pose can only be a direction without scale

Training Devices

Cloud servers would be helpful to our training, and we estimate to use $200 GCP credit.

Potential challenges

- **Challenge 1**: No available stereo camera
    - Potential Solutions:
        - Only test stereo cases on datasets
        - Only verify and demonstrate the rotation estimation in real world
- **Challenge 2**: Out-of-memory when running SuperPoint
    - Potential Solutions:
        - Execute descriptor calculation in CPU after a sparse set of feature points is detected
        - Replace the descriptor part of the neural network with classic descriptors
- **Challenge 3**: The given pretrained SuperPoint is not generalizable
    - Potential Solution:
        - Try "Retraining SuperPoint Network" on more datasets

Potential extensions to the project

- Consider quantization-aware training.
- Use the camera on the Jetson, and perform the real-world test
- Try optimizing UnSuperPoint, and compare it with SuperPoint
- Try optimizing the SuperGlue network and use it as the matching method

Potential ethical implications of the project

Our project does not imply any potential ethical issue.

Timeline and milestones

Week 5	Baseline Evaluation (Model preparation)
Week 6	Baseline Evaluation (Benchmark)
Week 7 - 8	Baseline Performance Evaluation
Week 9	Optimization (quantization)
Week 10 - Week 13	Optimization (distillation)
Week 14 - Week 15	Presentation and Report Preparation