Gebze Institute of Technology
Department of Computer Engineering
BIL 565 / 463
Computer Vision
Spring 2024

Semester Project

Title: Learning Keypoint Detectors and Descriptors using OpenCV Pseudo-Labels
Due Date: June 9th 2025

Objective:

Students will design, train, and evaluate a deep learning model to detect and describe keypoints in images. The keypoint labels will be generated using classical OpenCV methods. The trained system will be evaluated on publicly available RGB-D datasets and stereo correspondence tasks.

Task Description:

1. **Dataset Collection**
   - Use RGB images from the RGB-D Object Dataset and the Middlebury stereo dataset.
   - Focus on high-resolution, well-lit scenes with diverse object structures.
2. **Pseudo-Label Generation**
   - Use OpenCV feature detectors (e.g., SIFT, ORB, FAST) to generate keypoint locations as training labels.
   - Generate keypoint descriptors with OpenCV as ground truth for descriptor learning.
3. **Network Design**
   - Design a CNN or ViT-based architecture that takes an image as input and produces:
     - A 2D keypoint probability heatmap
     - A 1D descriptor vector (e.g., 32-128 dimensions) per keypoint
   - There should be a method to select keypoints from the heatmap with some threshold values
4. **Training**
   - Train your model on RGB-D Object Dataset images using pseudo-labels.
   - Augment data with synthetic views or transformations.
   - Clearly state the loss functions and optimizer used.
5. **Evaluation**
   - Evaluate the model on the Middlebury stereo dataset by using:
     - Keypoint matching on rectified stereo pairs
     - Descriptor matching (e.g., using cosine or L2 distance)
   - Compare the correspondence results with classical OpenCV methods (SIFT, ORB, etc.)
   - Provide numerical evaluations such as repeatability, matching accuracy, precision-recall.
6. **Reporting**

- Submit a **detailed report** that includes:
    - Network architecture and training strategy
    - Visualizations of predicted keypoints and descriptors
    - Quantitative performance metrics
    - Discussion of results, including failure cases
- Projects **will not be graded** without a report.

Deliverables:

1. Source code (Python, PyTorch or TensorFlow recommended)
2. Trained model weights
3. A report (PDF, max 6 pages excluding references)
4. Short demo video or interactive notebook

Performance Metrics Suggestions:

Metric	Description
Repeatability	Ratio of correctly repeated keypoints in stereo pairs
Matching Accuracy	% of correct keypoint matches
Descriptor Quality	Descriptor retrieval precision and recall (at fixed threshold)
Inference Speed	Time per image at test time

Additional Notes:

- You may use pre-trained CNN backbones if clearly documented.
- AI tools like ChatGPT or Copilot are allowed; include an appendix section describing how you used them.
- Collaboration rules: Discuss ideas, but **code and report must be written individually.**