6D Pose Estimation via Keypoint Heatmap Regression with RGB-D Residual Neural Network

Official repository for the Machine learning and Deep learning project "6D Pose Estimation via Keypoint Heatmap Regression with RGB-D Residual Neural Network".

- The entire pipeline (train, val, test) is organized into `.ipynb` notebooks located within the corresponding folder, executed in sequential order. A detailed description specifying which notebook corresponds to each phase is provided in the *Notebooks* subsection, while all necessary instructions for executing the code are provided directly within each notebook.
- Relative report is *

Repository structure

Data

The *data* folder contains following subfolders:

- *cropped_resized_data/* - Contains RGB images cropped by YOLO bounding boxes and resized to 256×256 for heatmap-based keypoint training.
- *cropped_resized_depth_data/* - Contains corresponding depth maps cropped and resized in the same way for the depth extension model.
- *full_data/* - Holds the complete set of RGB and depth images split into train/test folders, along with *gt.json* supplying 6D ground-truth poses and bounding boxes.
- *point_sampling_data/* - Includes *3D_50_keypoints_fps/cps.json* (FPS/CPS-sampled 3D points), *2D_50_keypoints_labels_fps/cps.json* (projected 2D points), and the *heatmaps_sigma_2* folder with Gaussian heatmaps used for training.
- *predicted_key_points/* - Stores JSON files of 2D keypoint coordinates predicted by different model variants. These keypoints are used as input to a PnP solver to compute the final 6D object pose.
- *raw/* - Contains the original LineMOD dataset organized by object ID, with subfolders *rgb/, depth/, mask/* and metadata files (*gt.yml, info.yml, train.txt, test.txt*) for 6D pose annotation.
- *yolo_data/* - Provides *train/* and *val/* splits of images, label files, and a *data.yml* configuration for training the YOLO object detection.

Models

The *models* folder contains all trained model checkpoints and it consists of the following subdirectories:

- *resnet/* - Stores checkpoint files for various ResNet-based keypoint heatmap models (baseline and depth-extended experiments)
- *yolov10m/* - Contains all artifacts from training and evaluating the YOLOv10-medium detector, including checkpoints, bounding-box predictions, and performance plots.
- *yolov10m.pt* - A standalone PyTorch checkpoint of the pretrained YOLOv10-medium model

Notebooks

The *notebooks* directory contains Google Colab notebooks that implement the end-to-end workflows for data preparation, model training, evaluation, and test.

- *ph1_01_data_prep.ipynb* - This notebook executes the essential data preparation steps required for the project
- *ph2_01_data_prep.ipynb* - This notebook prepares the dataset for YOLO, which requires a specific structure and format.
- *ph2_02_object_detection.ipynb* - In this notebook, object detection is performed on RGB images using YOLO.
- *ph3_01_data_prep.ipynb* - This notebook parses YOLO detections, crops RGB and depth patches to 256×256, and prepares them for heatmap-based keypoint regression.
- *ph3_02_point_sampling.ipynb* - This notebook applies curvature-based (CPS) and farthest-point (FPS) sampling to CAD point clouds and exports them for the next phase of the project.
- *ph3_03_dimension_projection.ipynb* - The notebook samples keypoints from the 3D point cloud, projects them into 2D image coordinates and converts them into Gaussian heatmaps.
- *ph3_04_baseline_model_definition.ipynb* - In this notebook, we train a convolutional neural network using a heatmap-based approach to accurately predict position of object keypoints from RGB images
- *ph3_05_depth_extension_with_general_training_experiments.ipynb* - In this notebook, we extend our heatmap-based keypoint regression to leverage both RGB and depth inputs, and we carry out general training experiments to compare different activation functions and learning-rate schedulers.
- *ph4_01_pnp_add.ipynb* - In this notebook, the PnP algorithm was implemented together with the RANSAC method to robustly estimate the 6D pose of the object based on known 2D-3D correspondences.
- *end-to-end/* - This directory contains the *ph5_01_end_to_end.ipynb* notebook, which unifies all pipeline stages into a single cohesive workflow, and *modules/* folder supplying all necessary scripts and utilities to execute the full system and presents results obtained on the test dataset for all models.