Aditya Wagh

Website: https://adityamwagh.me

Email: aditya.wagh@nyu.edu
Phone: (929) 424 1931

Education

- **New York University** — **New York City, NY**
  Master of Science in Mechatronics and Robotics; GPA: 3.667/4 — Sep 2021 – May 2023
- **Birla Institute of Technology and Science, Pilani** — **Pilani, India**
  Bachelor of Engineering in Electronics and Instrumentation — Aug 2015 – May 2019

Experience

- **Central Electronics Engineering Research Institute** — **Pilani, India**
  Deep Learning Intern — Jul 2018 - Dec 2018
  - **Data Annotation**: Contributed to the development of a software pipeline for pixel wise annotation of a novel data set consisting of 6000+ Infrared and RGB aerial images of power cables.
  - **Mask RCNN**: Fine-tuned a pretrained Mask RCNN model for panoptic segmentation of power cables on this new dataset and achieved a validation accuracy of approximately 70%

Technical Skills

- **Languages**: Python, C++, CUDA, Bash, MATLAB, HTML, CSS
- **Frameworks**: PyTorch, Keras, TensorFlow, OpenCV, Open3D
- **Tools & Platforms**: VSCode, Vim, Git, GitHub, HPC Clusters
- **Operating Systems**: Linux, MacOS, Windows

Relevant Coursework

- **Robotics**: Foundations of Robotics, Robot Perception, Robot Localization and Navigation
- **Machine Learning**: High Performance Machine Learning, Deep Learning, Mathematics for ML, Introduction to Tensorflow Neural Networks & Deep Learning, CNNs in Tensorflow,

Projects

- **Post-Earthquake Damage Assessment using Fully Convolutional Networks** — Tensorflow, Keras | GitHub
  - Designed multi-task fully convolutional networks for semantic segmentation of building components and their damage state.
  - Implemented batch normalization layers to enable faster convergence and better generalization over real data since the data used for the project was synthetically generated using physics based graphical models.
  - Achieved a mAP of 83% over 5 component classes and mAP of 70% for 5 damage state classes.
- **Dimensionality Reduction using Convolutional Autoencoders** — PyTorch | GitHub
  - Developed a deep convolutional autoencoder to reduce the dimensions of 28x28 sized Fashion MNIST images.
  - Designed an encoder-decoder network with latent space dimension of 64 to guarantee proper reconstruction of input.
  - Implemented batch normalization layers, learning rate decay and exponential scaling for faster convergence.
  - Further reduced dimensions of the latent space to 2 dimensions using t-SNE to visualise the reduced dimensions.
- **Visual Place Recognition using Bag of Visual Words** — Tensorflow, Keras | GitHub
  - Computed SIFT features for each image in database and queries using OpenCV's built-in SIFT feature extractor.
  - Employed the k-means clustering algorithm to compute 800 cluster centroids to be used as visual words to generate a frequency histogram of features in each image.
  - Computed histograms of visual words for all the query images and database images using OpenCV's histogram generator and extracted similar images from the database by using the k-nearest neighbours algorithm on the generated histograms.
- **Two-View Geometry based Relative Pose Estimation** — OpenCV | GitHub
  - Calibrated a camera using a calibration rig and removed radial distortion from the input images using the obtained camera matrix and distortion coeffecients.
  - Computed the fundamental matrix using the normalized 8 point algorithm and obtained the essential matrix using the fundamental matrix and camera matrix..
  - Decomposed the essential matrix to obtain the orientation and translation vectors between the images.
- **Marker based Augmented Reality** — OpenCV | GitHub
  - Obtained interest points to compute the epipolar geometry by detecting the corners of an AprilTag fiducial marker.
  - Solved a PnP problem to compute 3D to 2D correspondence between the marker corners and face of a cube in 3D space.
  - Projected 8 corners of the cube on the image and constructed a cube in 2D by joining the points.
- **3D Plane fitting in Point Cloud Data** — Open3D, Plotly | GitHub
  - Implemented the RANSAC algorithm to remove outlier data points which do not lie on a plane in the 3D point cloud data.
  - Randomly selected 3 points in data and computed plane parameters using parametric equation of a plane.
  - Computed the best plane parameters by minimizing perpendicular distance of each point in data from the plane.