Technical Showcase

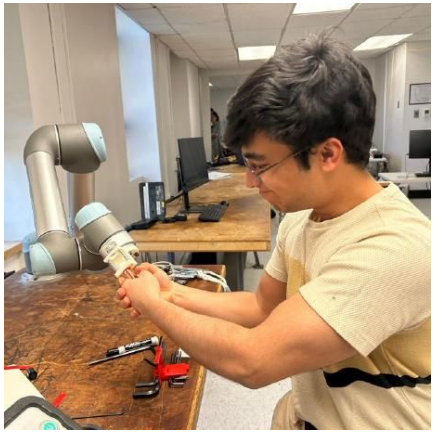
A short description of previous projects, experience and my skills

Ojas Taskar
MSE Robotics
Johns Hopkins

Introduction & About Me

Robotics Engineer interested in building **robust** and **deployable** perception systems.

- Currently on course to complete MSE, Robotics at Johns Hopkins with specialization in **perception & cognitive systems** in May 2026.
- Previously, completed Bachelor's in Electronics & Telecommunication from VJTI, Mumbai, specializing in **path planning for robotics.**
- Experienced working in **higher-level perception algorithms** as well as **lower-level hardware implementation**:
    - dataset curation, automated (robotic) data capture for training, 6D pose estimation or learning-based applications
    - model design, architectures, research-to-code implementation, training infrastructure
    - hardware selection, model profiling, compression, incorporating systems constraints into algorithm pipeline
    - optimization using CUDA kernels, Nsight Compute, TensorRT and ONNX
    - debugging failure cases, dataset augmentation, closed-loop learning, logging performance to cloud
    - git/version control, dockerization for easier deployment
- Most comfortable with Python (PyTorch), TensorFlow, C++ and also JAX.
- Research experience in 6D pose estimation, generative (diffusion) models, point cloud processing, image registration and robot learning.
- Coursework & Project experience in **3D Rendering, Novel View Synthesis, Object Detection, Semantic Segmentation**
- **My goal:** to build useful perception systems for real-world scenarios, bridge ideas from research into industrial use-cases, focus on simplistic solutions, built from first principles rather than overcomputing.



Previous Experience - JHU

Image Analysis and Communication Lab (IACL), JHU June 2025 – Aug 2025

Research Intern — Medical Image Synthesis

Focus: Synthesis of paramagnetic rim lesions in healthy-brain susceptibility-weighted MRI imaging.

- Worked on an approach to use diffusion models (conditional DDIMs) for synthesizing data, particularly lesion on brain MRI, since this data is not easily available publicly.
- Formulated training pipelines and tricks for faster inference of forward diffusion pass.
- Collaborated with large lab environment (25+ members), provided literature reviews, weekly presentations, codebase development with multiple users etc
- Selected from a batch of 85+ graduate students in ECE
- **Skills gained:**
    a. Dealing with inaccurate and incomplete volumetric data
    b. Utilising rendering tools such for 3D synthesis, helping data visualization
    c. Generative models in medical imaging

Vestibular & Ocular Motor Research Laboratory Sep 2024 – June 2025

Graduate Research Assistant

Focus: Multi-camera 6-DoF Head Pose Tracking for robot-assisted TMS procedures.

- Worked on robust pose estimation of a patient's head from multi-camera view to perform robotic procedures using fully automated robotic control.
- Developed iterative algorithms to solve depth ambiguity from monocular images taken from Intel RealSense v2 camera.
- Wrote accelerated pipelines for point-cloud registration, when point clouds have been derived using two different imaging methods.
- Co-authored a manuscript in IEEE VR 2026 on pose graph optimization for object tracking in scenes.
- **Skills gained:**
    a. Debugging failure cases in robotic systems, data logging
    b. Registration methods between pairs of 3D data
    c. capturing data using a robotic arm

Previous Experience - VJTI

KIREAP, Inc Jan 2024 – Jun 2024

Perception Research Intern

Focus: Vision pipelines for **end-to-end** autonomous drone landing.

- Worked on **end-to-end** vision pipelines for object detection onboard drones.
- Formulated mini-neural networks for specific tasks such as blur reversal, calibration error detection, AprilTag false positive detection etc
- Designed system with power, compute, hardware constraints
- Deployed onboard an actual drone, dealt with sim2real gap, inference acceleration
- **Skills gained:**
    a. drone (robot) localization using visual SLAM, sensor fusion, working with multi-modal sensors and data
    b. path planning, with real-time constraints
    c. testing algorithms in simulation

1 Martian Way Industries June 2023 - Aug 2023

Robotics Intern

- Worked on simplifying interfaces to a simulator platform (Microsoft AirSim).
- Deep learning projects such as crack detection for concrete blocks, **automated 3D structure from motion pipeline**, point cloud collection, Open3D visualization
- Reinforcement learning for path planning.
- **Skills gained:**
    a. Working with robotics software such as ROS, Gazebo, RViz
    b. 2D image to 3D structure methods

Project Demonstration

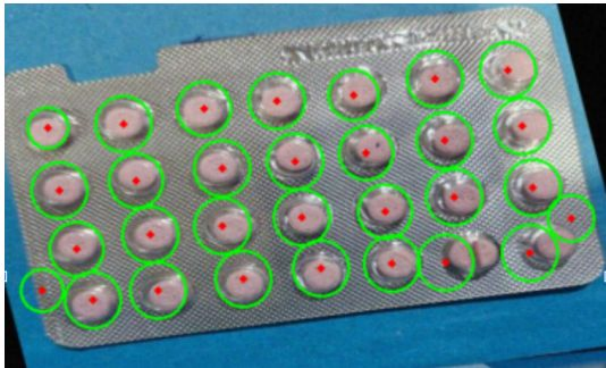
Contrastive Pre-training for Instance Classification

Problem:

1. **Automated defect detection** in medical tablet manufacturing.
2. Traditional supervised learning requires large labeled datasets.
3. In industrial settings, new defect classes appear **stochastically** in product lifecycle.
4. Continuous manual labeling becomes impractical and expensive.

Goal:

1. **Evaluate if self-supervision** can act as a possible solution.
2. Learn meaningful representations without any explicit labeling during **training.**
3. Backbone for downstream usage.
4. Flexible to any kind of downstream model i.e some **representation** learning.



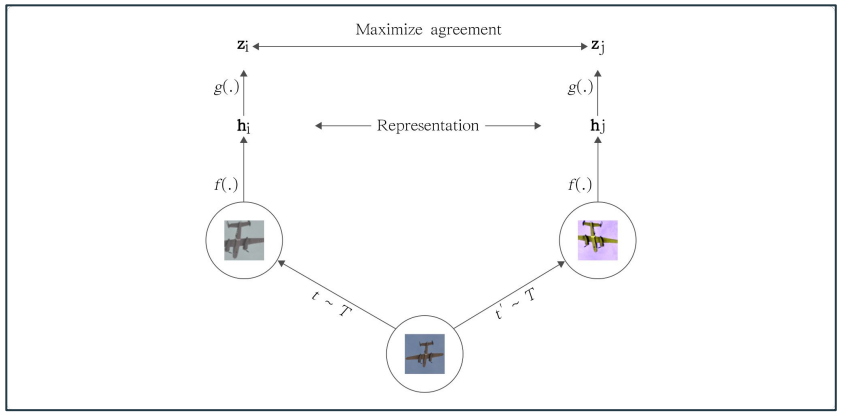
Contrastive Pre-training for Instance Classification

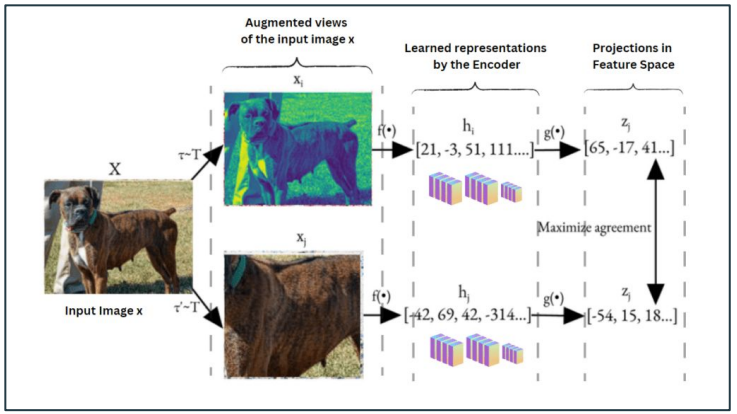
Approach:

1. Learn representation using pre-training.
2. During training, teach the pre-training model to identify dissimilarities.
3. Train some downstream classifier using a small portion of labelled data.

Algorithm Details:

1. Utilised a **contrastive method, SimCLR,** for pretraining (implemented from scratch)
2. Study methods to increase downstream classification accuracy without needing labels by adding **augmented data.**
3. Train large downstream models (Vision Transformer) using LoRA on backbone.





Contrastive Pre-training for Instance Classification

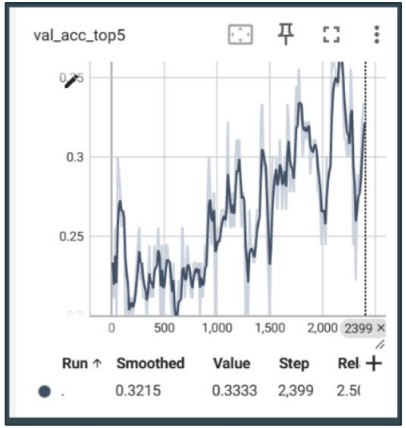
Results and what worked:

1. Achieved 42% top-5-validation loss using simple pre-training.
2. Data augmentation such as **gaussian crop** and **color jitter** was helpful in increasing accuracy without adding labels.
3. Modularized everything with PyTorch Lightning for plug-and-play.

```
Test accuracy for  10 images per label: 62.79%
Test accuracy for  20 images per label: 68.60%
Test accuracy for  50 images per label: 74.44%
```

What didn't work and why:

1. Still requires us to manually label some data for testing.
2. Very noisy validation loss curve with several hyperparameter tricks.



3D Reconstruction using Gaussian Splatting and NeRFs

Problem:

1. Investigate various common 3D rendering and novel view synthesis pipelines.
2. Evaluate based on fidelity of reconstruction, training time and rendering speed.
3. Look at acceleration of pipeline using CUDA/profiling.

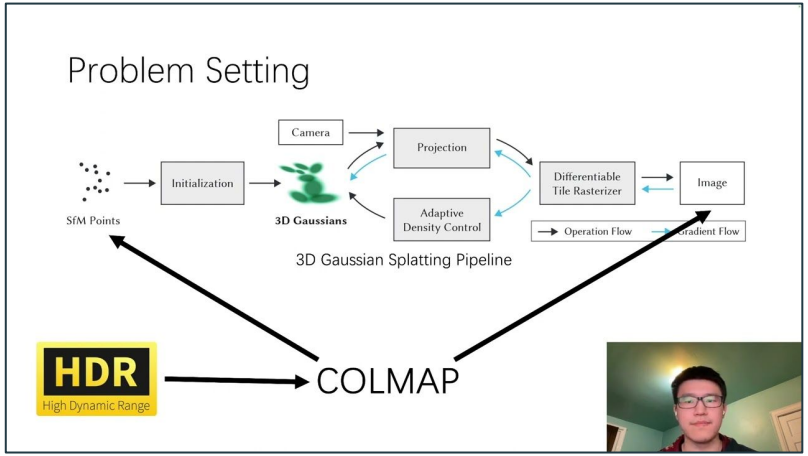
Goal:

1. Use **COLMAP and Nerfstudio** to train all our models.
2. Evaluate NeRF vs Gaussian Splatting.

3D Reconstruction using Gaussian Splatting and NeRFs

Approach:

1. Perform regular reconstruction approach: from data capture to 3D model.
2. Use COLMAP to process input image/video data, let it output **SfM** candidates for motion and structure.
3. Use these estimates as input to NerfStudio, using pre-trained and custom models.



Algorithm Details:

1. Utilised few algorithms from Nerfstudio: **Nerfacto, Instant-NGP, TensorRF, Splatfacto**
2. Experiment with reducing **image resolution** before training these models.

Model	Strength	Weakness	Use Case
Nerfacto	Balanced speed and quality	General-purpose, not specialized	Default for most use cases
Instant-NGP	Fastest training and rendering	May lose detail in complex scenes	Real-time applications
TensoRF	Memory-efficient, fast	Lower fidelity in some cases	Large scenes with memory constraints
Splatfacto	Real-time rendering, detailed	Newer, experimental	Dynamic scenes, AR/VR applications

Table I
COMPARISON OF THE VARIOUS NERFSTUDIO ALGORITHMS

3D Reconstruction using Gaussian Splatting and NeRFs

Results & Observations:

1. Nerfacto is slower to train and worse at rendering than all splatting models.
2. InstantNGP does real-time rendering, but is slower to train than regular splatting.
3. TensorRF is very powerful, with high quality renders, but is not real-time.

Optimization:

1. Using **Nsight Compute** and **PyTorch Profiler,** checked kernel execution time, SM occupancy and memory efficiency (global and bandwidth utilization).
2. NeRF has bottlenecks in ray sampling, evaluating MLP output and integrating color equation across every sampled point in space.
3. Splatting bottlenecks in sorting Gaussians by visibility, projecting Gaussians to 2D and rasterization.
4. Achieved **5.2x** speedup by batching ray sampling, increasing SM occupancy and fusing small kernels.

Figure 5. Chair visualized using Nerfacto

Figure 6. Speaker visualized using Splatfacto

Figure 7. Splatfacto on full resolution pepper box data

Figure 8. Nerfacto on full resolution pepper box data

Figure 9. Nerfacto on reduced data

Figure 10. Splatfacto on reduced data

Model	Dataset	Training Time (min)	Steps	Render Quality (/5)
Nerfacto	poster	23	120 K	3
Instant-NGP	poster	34	34 K	3
Instant-NGP (Bounded)	poster	38	20 K	4
Splatfacto	poster	17	10.1 M	4
TensoRF	blender	41	70 K	5

Table II
PRELIMINARY PERFORMANCE OF CHOSEN NERF ALGORITHMS


THANK YOU