Robotics Laboratory, University of Leeds, UK

Deep Learning Researcher Intern

UNIVERSITY OF LEEDS

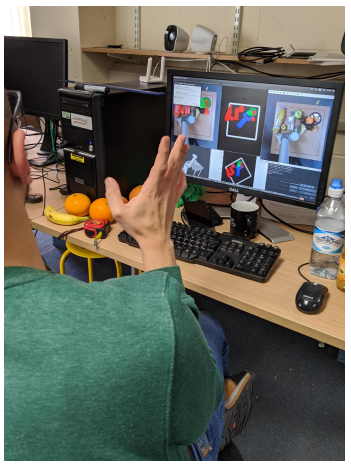
Journey:

Every year Government of Assam, India provides 2 overseas research scholarship to Computer Science students from the state of Assam. I received full undergraduate merit scholarship from Government of Assam, India in 2020 to research on "6D pose tracking of objects in occlusion or cluttered environments" in the Robotics Laboratory, University of Leeds, UK with Prof. Dr Mehmet Dogar. Dr. Dogar is an Associate Professor at the School of Computing, University of Leeds and previously, he was a postdoctoral researcher at CSAIL, MIT. Dr. Dogar received his PhD from the Robotics Institute at Carnegie Mellon University. Dr Dogar is also an EPSRC Fellow, a co-chair of the IEEE RAS Technical Committee on Mobile Manipulation and an Associate Editor for IEEE Robotics and Automation Letters. His lab holds expertise in robotics, manipulation planning, motion planning, grasping etc and it was a great experience for me. I worked with Sajid Iftarul Hussain (fellow cs scholar), Wissam Bejjani and Rafael Papallas in the research group. It was an amazing experience in the lab but I had to come back to India early in the last week of March 2020 [*covid 19 spreads]. I continued to work remotely for few weeks online from India. [certificate]

Photo: Me Felicitating Dr. Dogar with Assamese Gamosa.

The aim of our project is to set up a system to track the 6D (x, y, z, roll, pitch, YAW) of objects in a scene where there may be occlusions between the objects due to clutter. The data to be used will either be vision (RGB) data or both vision (RGB) and depth data (RGBD), which can possibly be extended within motion capture (opti-track marker data). Estimating the 6D pose of known objects is important for robots

to interact with the real world. The problem is challenging due to the variety of objects as well as the complexity of a scene caused by clutter and occlusions between objects.


Resources:

ROBOTIC MANIPULATION PLANNING

[Robotic Manipulation under clutter and uncertainty with and around people, Versatility in Robotic Manipulation: the Long Road to Everywhere, Models of Robotic Manipulation, Robotic Manipulation Explained, Robotic Manipulation @ Leeds, AR - Towards Robotic Manipulation @ CMU]

ROBOTIC MOTION PLANNING

[Sertac Karaman (MIT) on Motion Planning in a Complex World - MIT Self-Driving Cars, Deep Reinforcement Learning for Motion Planning, RRT, RRT* & Random Trees, Robust motion planning for walking robots and robotic birds, Springer - Handbook of Robotics]

Deep Learning for Robotics and GRASPING

[cmu robotics - Deep Learning for Robotics, Real-World Robot Learning (TensorFlow Dev Summit 2018), MIT Robot Seminar - Ken Goldberg - The New Wave in Robot Grasping, Robotic Grasping and Manipulation, Deep Learning & Robotics - Prof. Pieter Abbeel]

Papers on 6D pose estimation [Chronological order]:

[A Self-supervised Learning System for Object Detection using Physics Simulation and Multi-view Pose Estimation ✣ Improving 6D Pose Estimation of Objects in Clutter via Physics-aware Monte Carlo Tree Search ✣ SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again ✣ PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes ✣ Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects ✣ DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion ✣ Physics-based Scene-level Reasoning for Object Pose Estimation in Clutter ✣ Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation ✣ DPOD: 6D Pose Object Detector and Refiner ✣ Self-supervised 6D Object Pose Estimation for Robot Manipulation ✣ Motion-Nets: 6D Tracking of Unknown Objects in Unseen Environments using RGB ✣ DeepIM: Deep Iterative Matching for 6D Pose Estimation]

NOTES: Pieter Abbeel [Professor, UC Berkeley] [covariant.ai] | Many Exciting Challenges for AI in robotics:

> Few Shot Reinforcement Learning

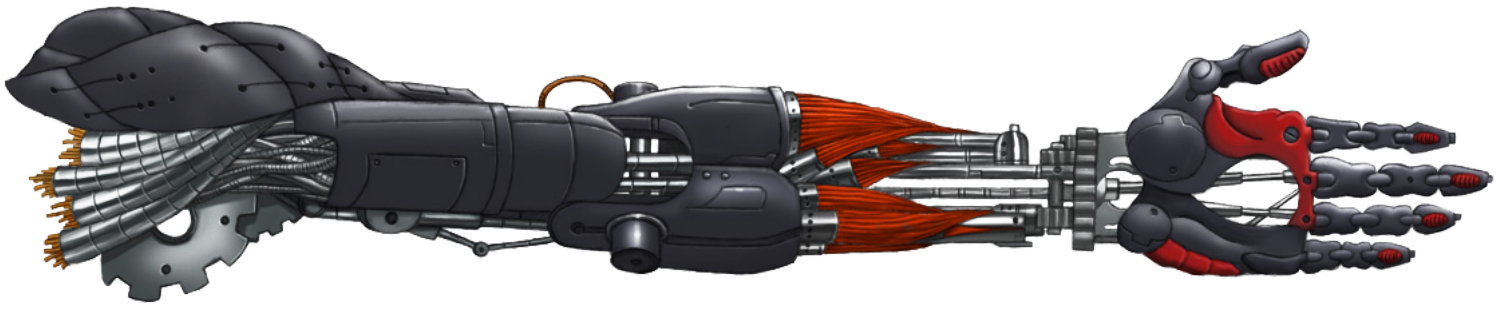
From past experiences, an agent becomes ready to learn more quickly from a few number of episodes in a new environment.

→ RL^2: Fast Reinforcement Learning via Slow Reinforcement Learning

→ Deep Reinforcement Learning for Tensegrity Robot Locomotion

> Leveraging Simulation!

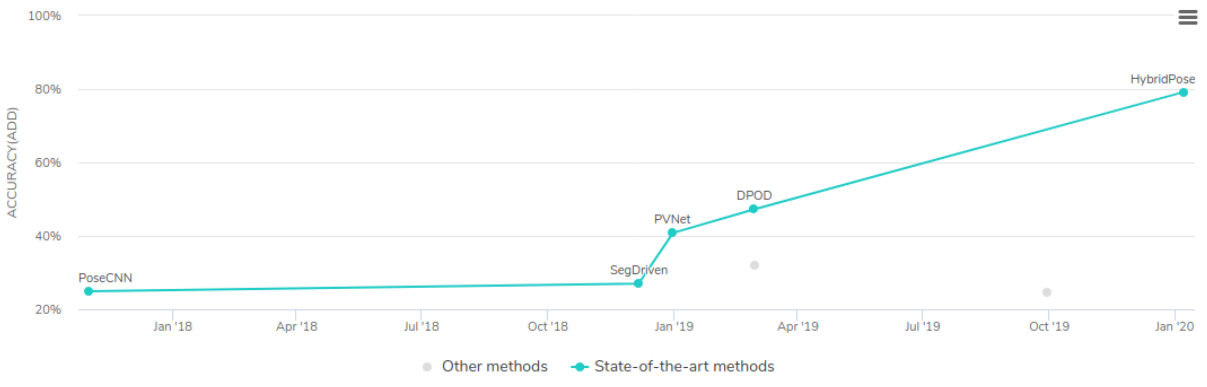
[**Carefully match the simulation to the world:** 3D Simulation for Robot Arm Control with Deep Q-Learning , ✣ Deep Learning a Grasp Function for Grasping under Gripper Pose Uncertainty , ✣ Dex-Net 3.0: Computing Robust Robot Vacuum Suction Grasp Targets in Point Clouds using a New Analytic Model and Deep Learning , ✣ Whole-body Model-Predictive Control applied to the HRP-2 Humanoid **Augment Simulated data with real data:** Playing for Data: Ground Truth from Computer Games ✣ Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping **Domain Confusion / Adaptation:** ✣ Deep Domain Confusion: Maximizing for Domain Invariance ✣ Sim-to-Real Robot Learning from Pixels with Progressive Nets **Domain Randomization:** CAD2RL: Real Single-Image Flight without a Single Real Image ✣ Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World ✣ Sim-to-Real Transfer of Robotic Control with Dynamics Randomization ✣ Learning Dexterous In-Hand Manipulation [paper] - OPEN AI - Learning Dexterity [video / blog] : Sergey Levine [Professor, EECS, UC Berkeley], Peter NEURIPS [notes]

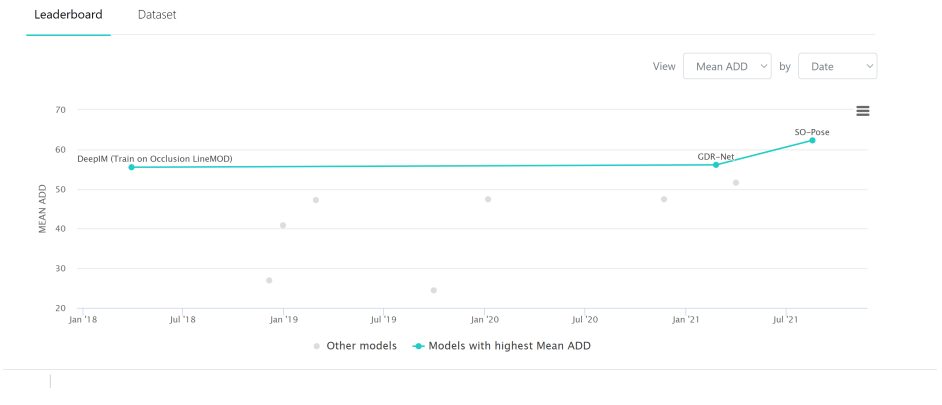


My Work:

○ Literature Survey & Analysis:

I researched on performance of existing works on occlusion and found out that 6D Pose Estimation is done using RGB or RGB-D on standard datasets such as OccludedLINEMOD. Here is the performance of RGB on OccluddedLINEMOD dataset:





6D Pose Estimation using RGB:

PoseCNN - /paper (CODE/github) holds a score of 24.9% accuracy in the OccludedLINEMOD dataset.

DPOD - /paper holds a score of 47.25% accuracy in the OccludedLINEMOD dataset.

HybridPose - /paper (CODE/github) holds an accuracy of 79.2% in the OccludedLINEMOD dataset.

DOPE - /paper (CODE/github) compares to PoseCNN in the YCB dataset and concludes to be better than PoseCNN. [video]

6D Pose Estimation using RGB-D:

PoseCNN - /paper (CODE/github (CODE/github)

DenseFusion - /paper (CODE/github (CODE/github)

DenseFusion achieves better results than PoseCNN [video]

• The Occlusion Dataset is a reannotation of the bench vise segment of the LINEMOD dataset. All objects are now annotated instead of simply the bench vise. I found these datasets interesting, keeping a bookmark RWTH Pose Occlusion & MIT-Hand-Occlusion. Main datasets : LINEMOD - /link, /link2; YCB DATASET - /link; OccludedLINEMOD dataset - /link2. There seems to be a new dashboard for 6D pose estimation now here [2021 update].

Quaternions : Quaternions and spatial rotation, Understanding Quaternions, 3blue1brown - Visualizing quaternions (4d numbers) with stereographic projection, Quaternions and 3d rotation, explained interactively.

PoseCNN is the method introduced in the paper PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes published by Yu Xiang, Tabber Schmidt, Venkatraman Narayanan and Dieter Fox in 2018. It's a Convolutional Neural Network for 6D object pose estimation. It estimates the 3D translation of an object by localizing its center in the image and predicting its distance from the camera. The 3D rotation of the object is estimated by regressing to a quaternion representation. They also introduced a novel loss function that enables PoseCNN to handle symmetric objects. They provided a video dataset for 6D pose estimation named YCB. It provides accurate 6D poses of 21 objects from the YCB dataset observed in 92 videos with 133,827 frames. The dataset used were the YCB video dataset and the OccludedLINEMOD dataset. PoseCNN holds a score of 24.9% accuracy in the OccludedLINEMOD dataset.

DOPE : Using synthetic data for training deep neural networks for robotic manipulation allows for unlimited amounts of pre-labeled training data. One of the problems of synthetic data has been to bridge the reality gap, so that networks trained on synthetic data operate correctly when exposed to real-world data. The paper Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects published by Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox and Stan Birchfield in 2018 explore the reality-gap in the context of 6-DoF (degrees of freedom) pose estimation of known objects from a single RGB image. The problem can be successfully spanned by a simple combination of domain randomized and photorealistic data. Using synthetic data generated in this manner, they introduced a one-shot deep neural network. This network also generalizes better to novel environments including extreme lightning conditions. Using this network, they demonstrated a real-time system estimating object poses with sufficient accuracy for real-world semantic grasping of known household objects in clutter by a real robot. They focused on rigid, known objects for which a prior training time to learn the appearance and shape of the object is allowed. They inferred the 3D pose of such objects, in clutter, from a single RGB image in real time for the purpose of enabling the robot to manipulate such objects. This network used a simple deep network architecture, trained entirely on simulated data, to infer the 2D image coordinated of projected 3D bounding boxes, followed by perspective-n-point. This system was called DOPE (Deep Object Pose Estimation). DOPE, compared to PoseCNN in the YCB dataset, is better.

Dense Fusion : A key technical challenge in performing 6D object pose estimation from RGB-D image is to fully leverage the two complementary data sources. DenseFusion is a generic framework for estimating 6D pose of a set of known objects from RGB-D images. It was introduced in the paper DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion published by Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martin-Martin, Cewu Lu, Li Fei-Fei, Silvio Savarese in 2019.
DenseFusion has a heterogeneous architecture that processes the two data sources individually and uses a novel dense fusion network to extract pixel-wise dense feature embedding, from which the pose is estimated. The datasets tested are YCB-Video and LineMOD. The advent of cheap RGB-D sensors has enabled methods that infer poses of low-textured objects even in poorly-lighted environments more accurately than RGB-only methods. Requires elaborate post-hoc refinement such as highly customized Iterative Closest Point (ICP) procedure in PoseCNN and a multi-view hypothesis verification scheme in MCN. These refinements cannot be optimized jointly for a

final objective.

The approach in DenseFusion is to embed and fuse RGB values and point clouds at per-pixel level. This per-pixel fusion scheme enables our model to explicitly reason about the local appearance and geometry information,which is essential to handle heavy occlusion.

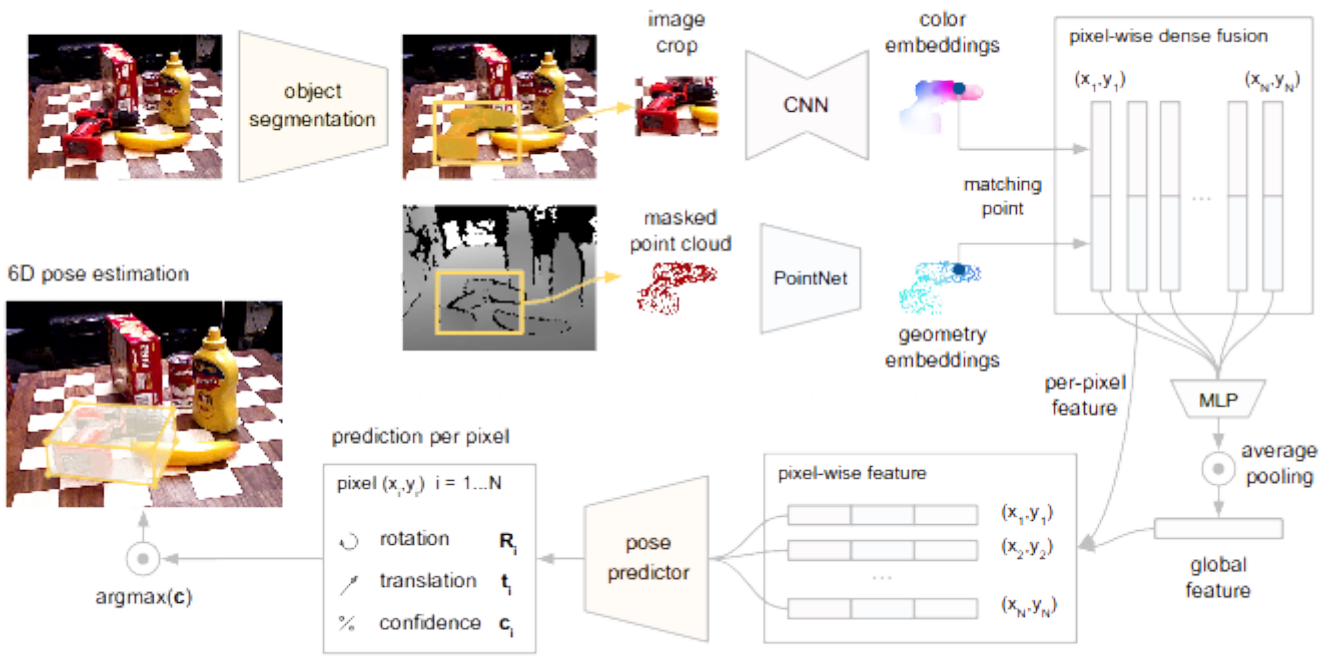
It is an iterative method which performs pose refinement within the end-to-end learning framework. DenseFusion outperforms the state-of-the-art PoseCNN after ICP refinement by 3.5% in pose ac-curacy while being 200x faster in inference time.

Contribution of DenseFusion:

♔ First, it is a principled way to combine color and depth information from the RGB-D input. DenseFusion augments the information of each 3D point with 2D information from an embedding space learned for the task and uses this new color-depth space to estimate the 6D pose.

♔ Second, DenseFusion integrates an iterative refinement procedure within the neural network architecture, removing the dependency of previous methods of a post-processing ICP step.

DenseFusion is most related to PointFusion, in which geometric and appearance information are fused in a heterogeneous architecture. Also outperforms PointFusion's naive fusion by concatenation method. The architecture of DenseFusion is:



DPOD : is a novel deep learning method for 3D object detection and 6D pose estimation from RGB images. It was introduced in the paper DPOD: 6D Pose Object Detector and Refiner published by Sergey Zakharov, Ivan Shugurov and Slobodan Ilie in 2019. DPOD estimated multi-class 2D-3D correspondence maps between an input image and available 3D models. Given the correspondences, a 6-DoF pose is computed via PnP and RANSAC. An additional RGB pose refinement of the initial pose estimates is performed using a custom deep learning-based refinement scheme. They demonstrated by comparing to a vast number of related works that a large number of correspondences is beneficial for obtaining high-quality 6D poses both before and after refinement. They performed evaluation on both real and synthetic data. DPOD holds a score of 47.25% accuracy in the OccludedLINEMOD dataset. It achieves a psoe accuracy that surpasses all other related deep learning-based pose refinement approaches, while having a simpler and more lightweight backbone architecture.

HybridPose is the most recent work on this problem. It was introduced in the paper HybridPose: 6D Object Pose Estimation under Hybrid Representations by Chen Song, Jiary Song, Qixing Huang in 2020. HybridPose utilizes a hybrid intermediate representation to express different geometric information in the input image, including key points, edge vectors, and symmetry correspondences. Hybrid representation allows pose regression to exploit more and diverse features when one type of predicted representation is inaccurate. Compared to other state-of-the-art pose estimation approaches, HybridPose is comparable in running time and is significantly more accurate. On Occlusion Linemod dataset, HybridPose achieved a prediction speed of 30 fps with a mean accuracy of 79.2%, representing a 67.4% improvement from the previous state-of-the-art pose estimation approach.

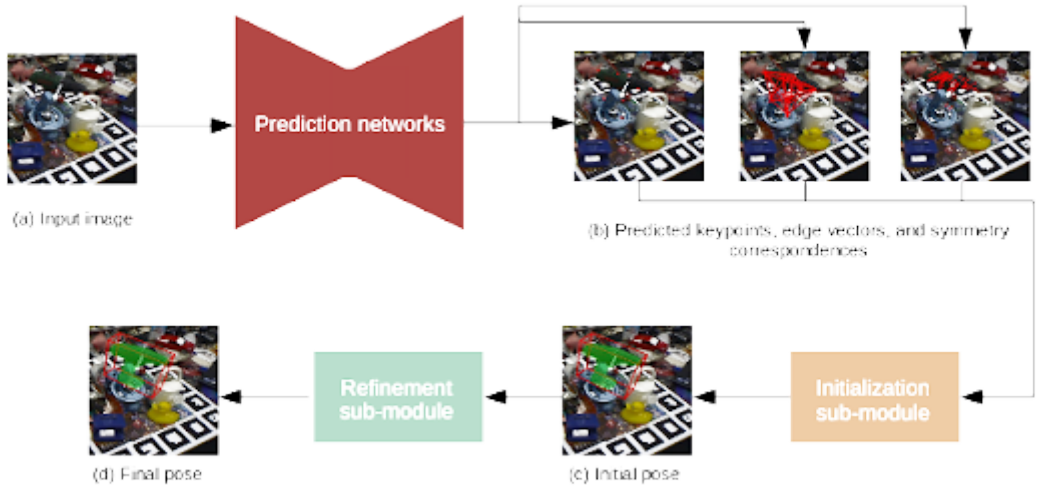
HybridPose is a novel 6D pose estimation approach that leverages multiple intermediate representations to express the geometric information in the input image. In addition to keypoints, HybridPose integrates a prediction network that outputs edge vectors between adjacent keypoints. It utilizes predicted dese pixel-wise correspondences that reflect the underlying symmetric relations between pixels. Compared to a unitary representation, this hybrid representation has many benefits:

HybridPose integrates more signals in the input image: edge vectors encode spatial relations among object parts, and symmetry correspondences incorporate interior details. HybridPose offers more constraints than using keypoints alone for pose regression, enabling accurate pose prediction even if a significant fraction of predicted elements are outliers (because of occlusion). Symmetry correspondences stabilize the rotation component of pose prediction, especially along the normal direction of the reflection plane.

HybridPose is evaluated on two popular benchmark datasets, Linemod and Occlusion Linemod. On Occlusion Linemod, HybridPose achieves an accuracy of 79.2%, which represents a 67.4% improvement from DPOD, the current state-of-the-art method on this benchmark dataset.

This approach is efficient and runs at 30 frames per second on a commodity workstation. Compared to approaches which utilize

sophisticated network architecture to predict one single intermediate representation, HybridPose achieves considerably better performance by using a relatively simple network to predict hybrid representations.



HybridPose consists of intermediate representation prediction networks and a pose regression module.The prediction networks take an image as input, and output predicted keypoints, edge vectors, and symmetry correspondences. The pose regression module consists of an initialization sub-module and a refinement sub-module. The initialization sub-module solves a linear system with predicted intermediate representations to obtain an initial pose. The refinement sub-module utilizes GM robust norm and optimizes to obtain the final pose prediction.

○ Experimentation:

We built our system setup and code environment to train and test the datasets for getting hands on with the computer vision problem. We needed a platform/environment that is portable, and not limited by the local system's hardware and software limitations. Container comes into the picture as a solution to this challenge.

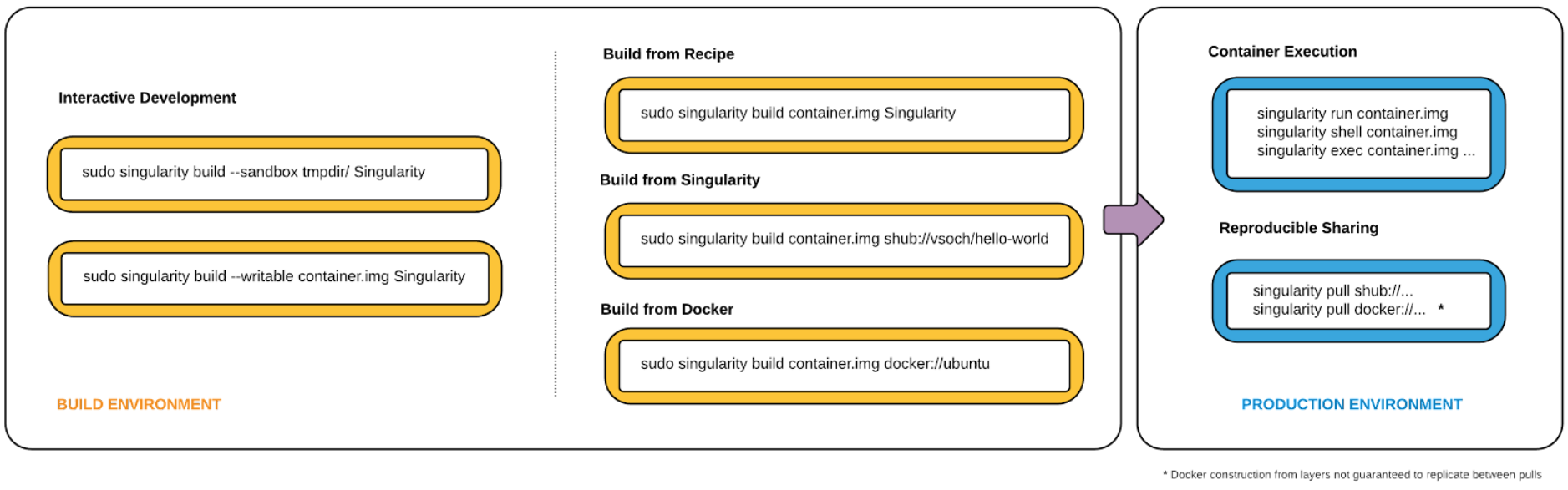
Singularity Container :

Singularity enables users to have full control of their environment. Singularity containers can be used to package entire scientific workflows, software and libraries, and even data. This means that we don't have to ask our cluster admin to install anything for us - we can put it in a Singularity container and run.

If we already use a Docker, Singularity can import our Docker images without having Docker installed or being a superuser. We can share your code by putting it in a Singularity container and your collaborator won't have to go through the pain of installing missing dependencies. We can 'swap out' the OS on your host for a different one within a Singularity container. As the user, We are in control of the extent to which your container interacts with its host. There can be seamless integration, or little to no communication at all.

Singularity: contact author - @bauerm97 on github - paper, code, video, article | Nvidia driver, NV110 Family (Maxwell) , NV117 - Product Type - Quadro ; Product Series - Quadro Series ; Product - Quadro K2200, Compute Capability - 3.0 [here]
Why? Escape "Dependency Hell" Why Singularity ? - Elaborate

Singularity is specifically built for High Performance Computing (runs without root privileges, seamless integration into existing infrastructure) Integrates perfectly into "Slurm" and no scheduling issues. Any container can be run by any user. Single .img file contains everything necessary.



I set up my Singularity container from scratch with a post script provided by our co-guide Rafael Papallas and began working on training the deep learning models Hybridpose and DenseFusion. Training of both DenseFusion and Hybridpose was done in a singularity

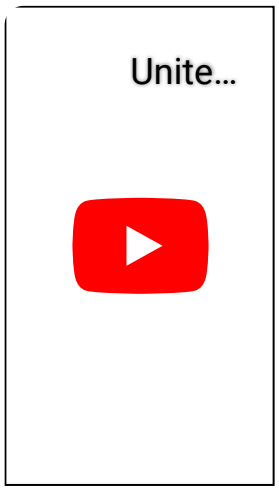
container with Ubuntu 16 LTS and Python 3. I further started to tweak the algorithm and tested with my own subset algorithms to understand 6d pose estimation. Finally I had my 6d pose estimation algorithms running in a singularity container.

Tech Stack: Python, PyTorch, TensorFlow, Docker, Singularity, Git.

YCB_Video_Dataset: Training and Testing sets follow PoseCNN. The training set includes 80 training videos 0000-0047 & 0060-0091 (chosen by 7 frame as a gap in our training) and synthetic data 000000-079999. The testing set includes 2949 keyframes from 10 testing videos 0048-0059. [link]

LINEMOD [L]: preprocessed linemod dataset [link]

Final Results:



Experimental Results

Research : I set up the pipeline of 6d-pose estimation research and did extensive analysis of the state of the art algortihms with their performance and architecture both on RGB and RGB-D data.

My UK Travel VLOG

Memories