DeepRob

Seminar 1
3D Perception: RGB-D Architectures
University of Michigan and University of Minnesota

This Week: 3D Perception

- Seminar 1: RGB-D Architectures
  1. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes, Xiang et al., 2018
  2. A Unified Framework for Multi-View Multi-Class Object Pose Estimation, Li et al., 2018
  3. PVN3D: A Deep Point-Wise 3D Keypoints Voting Network for 6DoF Pose Estimation, He et al., 2020
  4. Learning RGB-D Feature Embeddings for Unseen Object Instance Segmentation, Li et al., 2021
- Seminar 2: Point Cloud Processing
  1. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, Qi et al., 2017
  2. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, Qi et al., 2017
  3. PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation, Xu et al., 2018
  4. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion, Wang et al., 2019

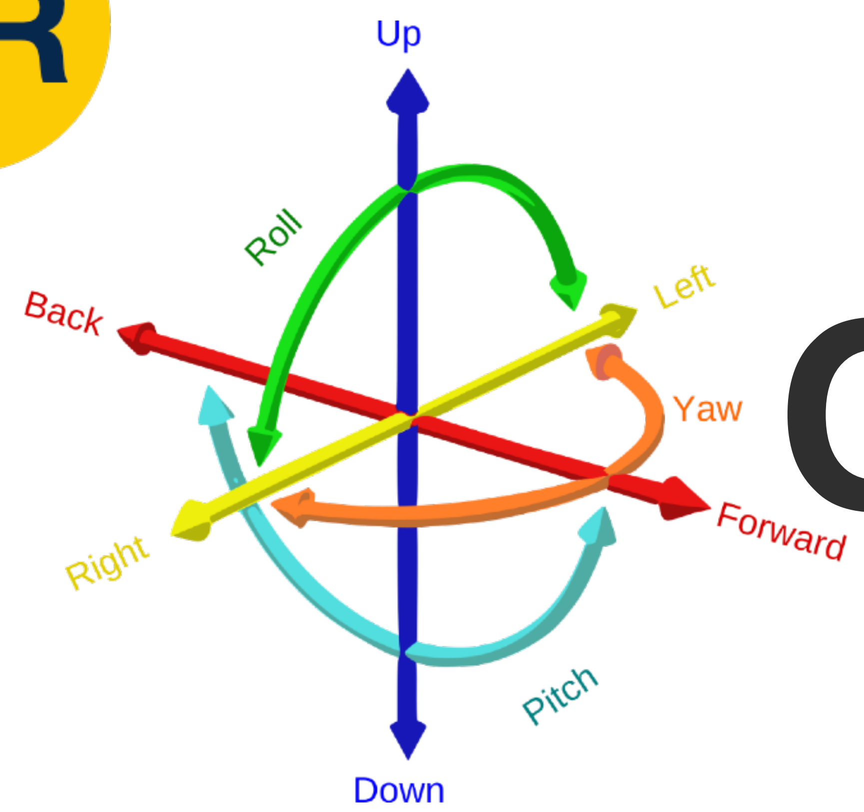
Today: RGB-D Architectures

- Seminar 1: RGB-D Architectures
  1. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes, Xiang et al., 2018
  2. A Unified Framework for Multi-View Multi-Class Object Pose Estimation, Li et al., 2018
  3. PVN3D: A Deep Point-Wise 3D Keypoints Voting Network for 6DoF Pose Estimation, He et al., 2020
  4. Learning RGB-D Feature Embeddings for Unseen Object Instance Segmentation, Li et al., 2021
- Seminar 2: Point Cloud Processing
  1. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, Qi et al., 2017
  2. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, Qi et al., 2017
  3. PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation, Xu et al., 2018
  4. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion, Wang et al., 2019

A Unified Framework for Multi-View Multi-Class Object Pose Estimation

By: Chi Li, Jin Bai, Gregory D. Hager

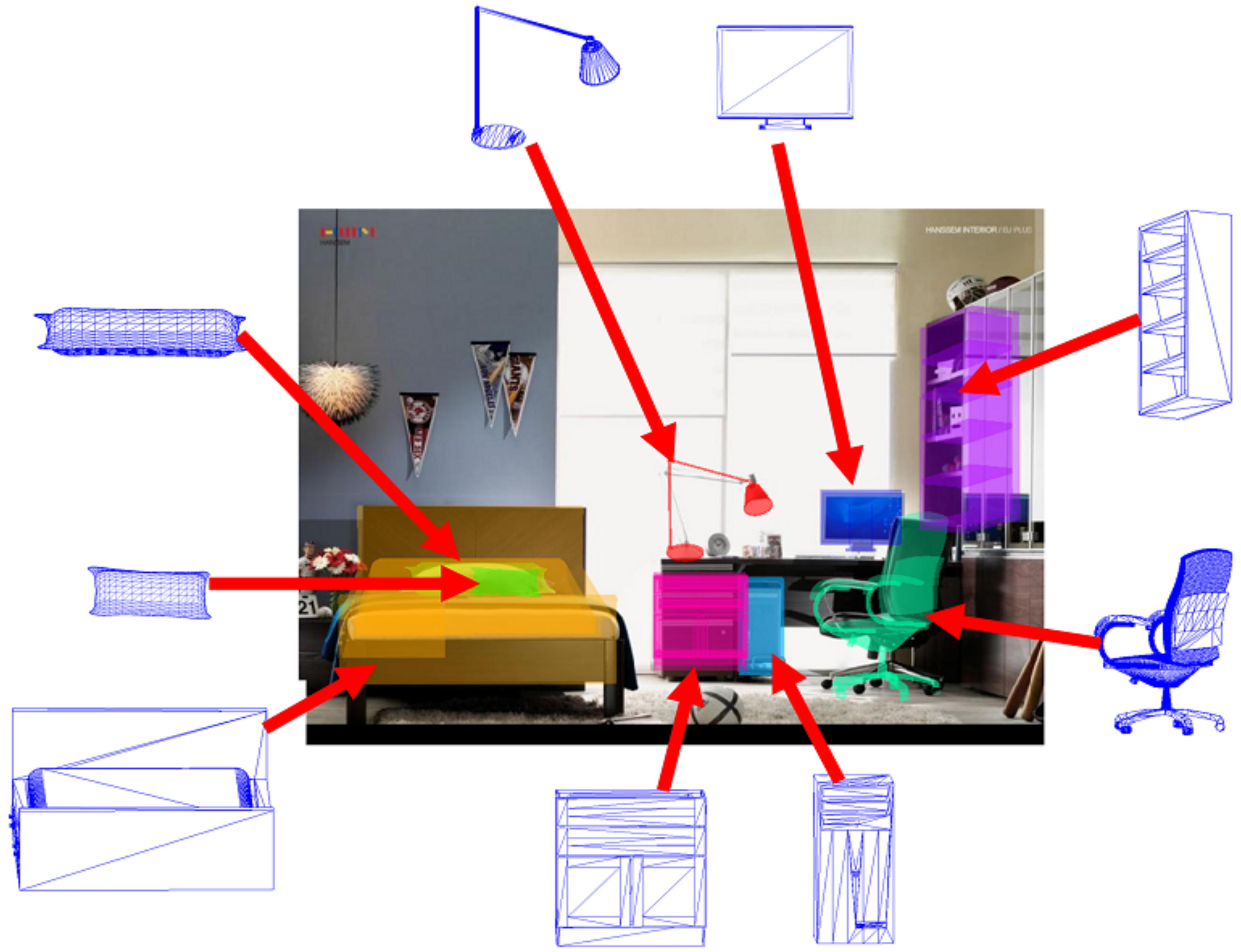
Presented by: Nibarkavi Naresh



Objective

"Accurately infer six Degree-of-Freedom (6-DoF) pose for a large number of object classes from single or multiple views"

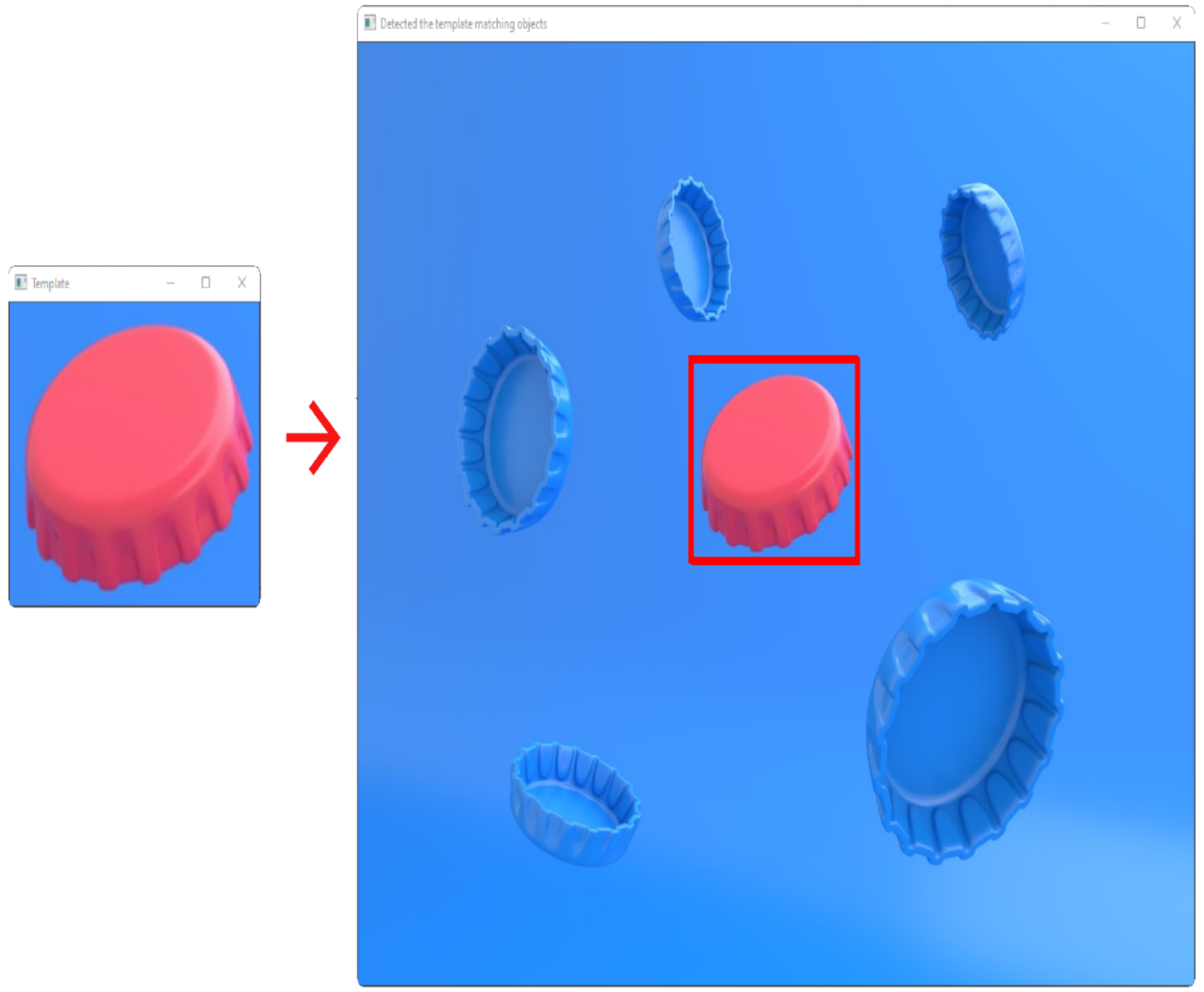
JHUScene-50 and YCB-Video for pose estimation



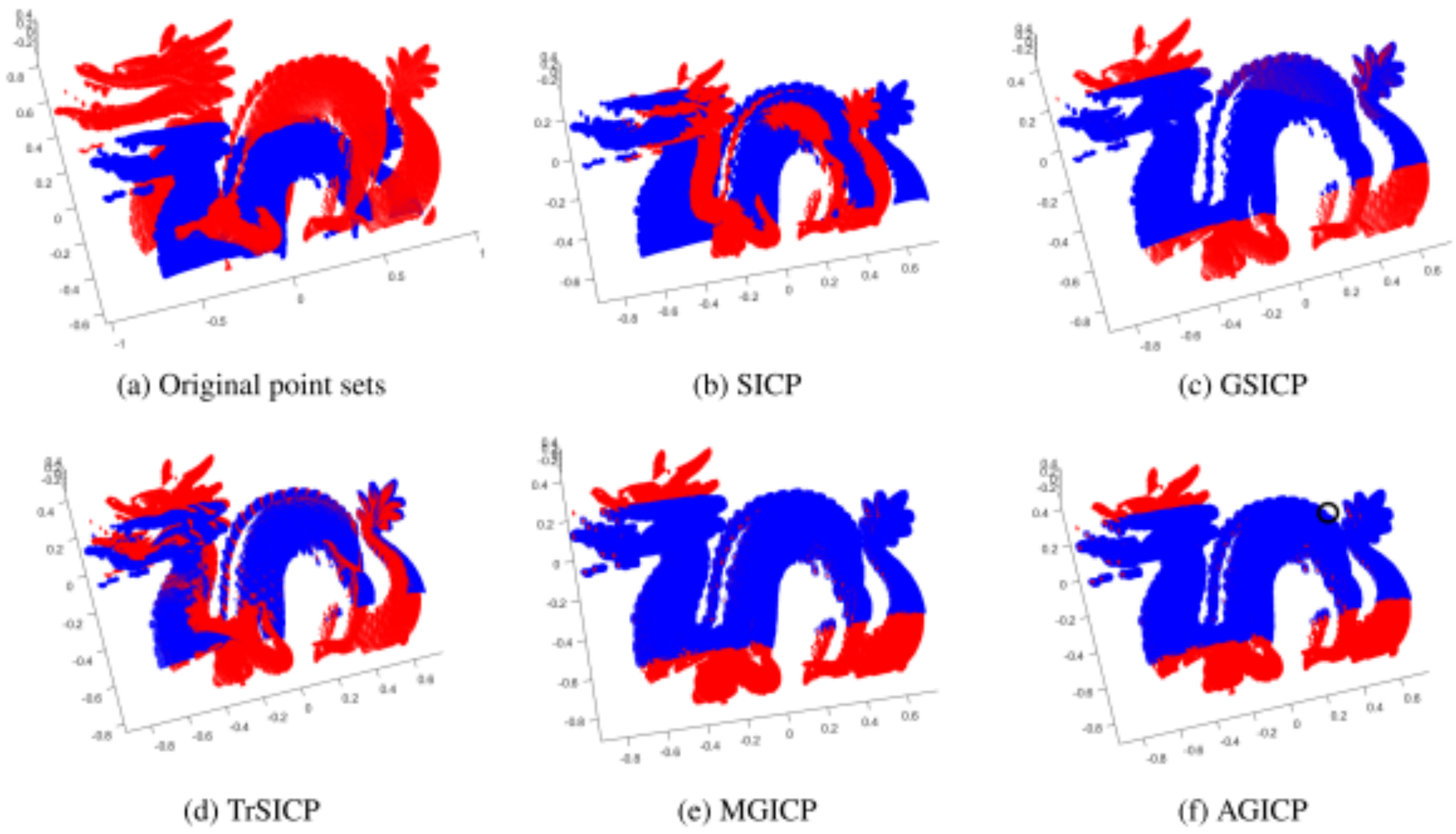
ObjectNet-3D for viewpoint estimation

Related Work

Template Matching



Bottom-up approaches



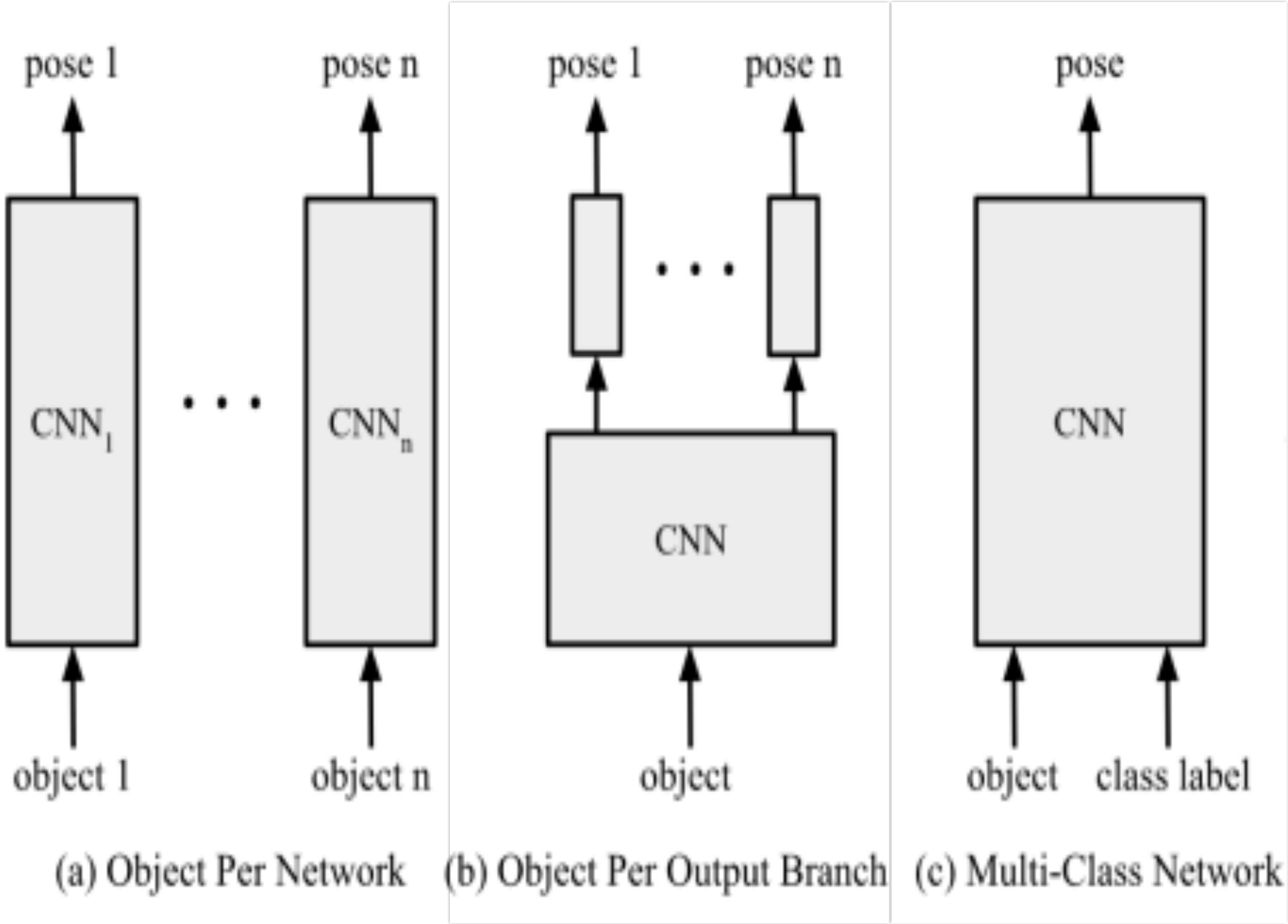
Coarse-to-fine ICP (Iterative Closest Point)

DOI:[10.1109/ACCESS.2020.2976132](https://doi.org/10.1109/ACCESS.2020.2976132)

Learning end-to-end pose machines

Microsoft's Human Pose Estimation

Single-view object pose estimation learning architectures

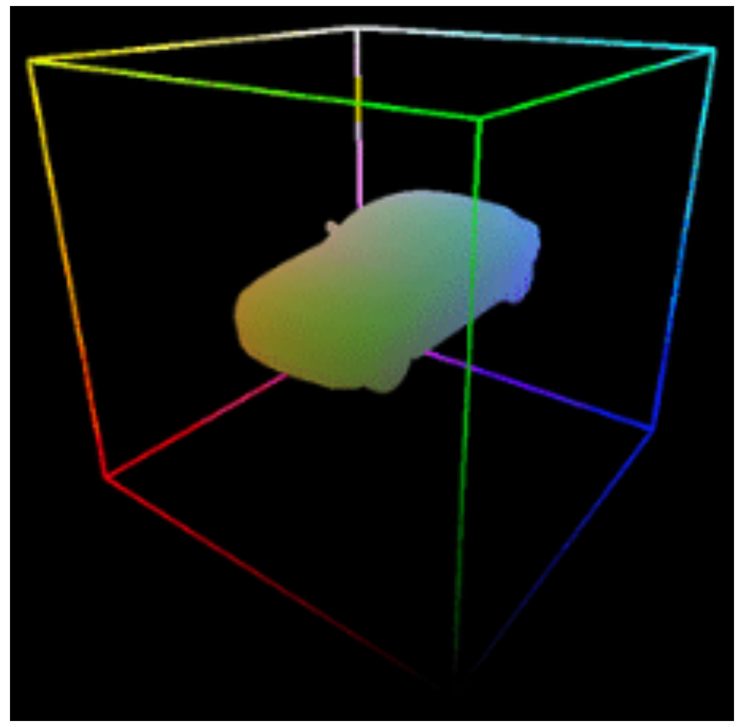


Naïve approaches **Author's networks**

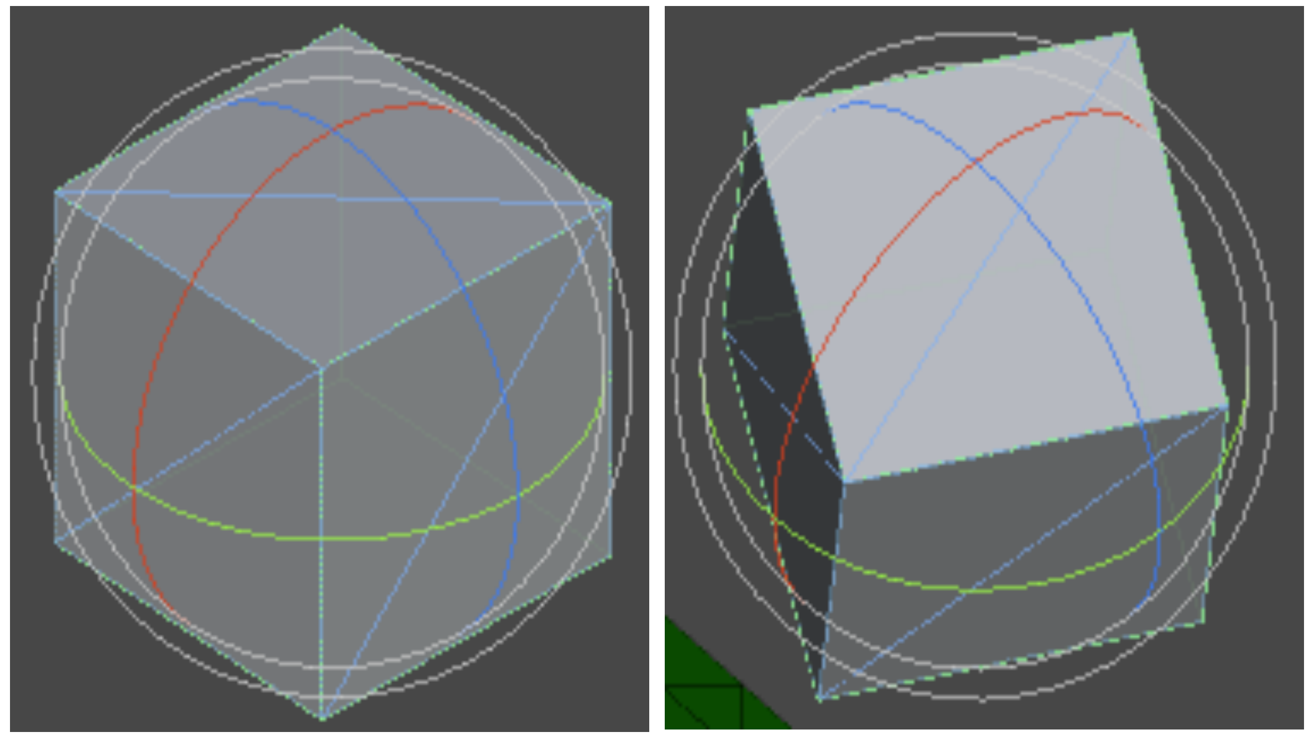
Multi-class network architecture for the single view - Input

Input 1 – RGB image with RoI

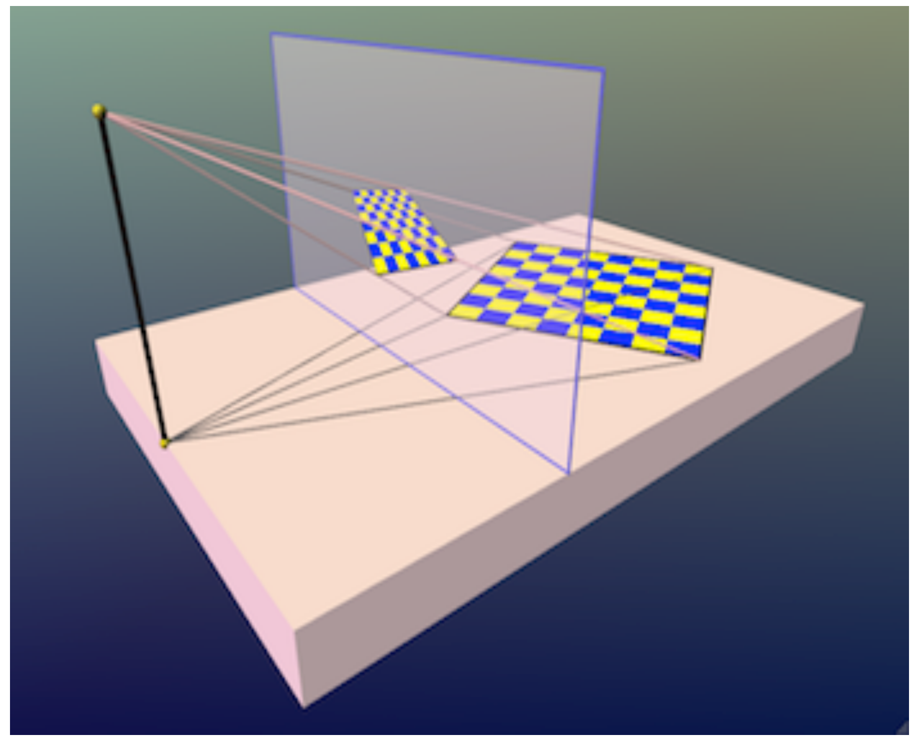
Input 2 – XYZ map with normalised 3D coordinates



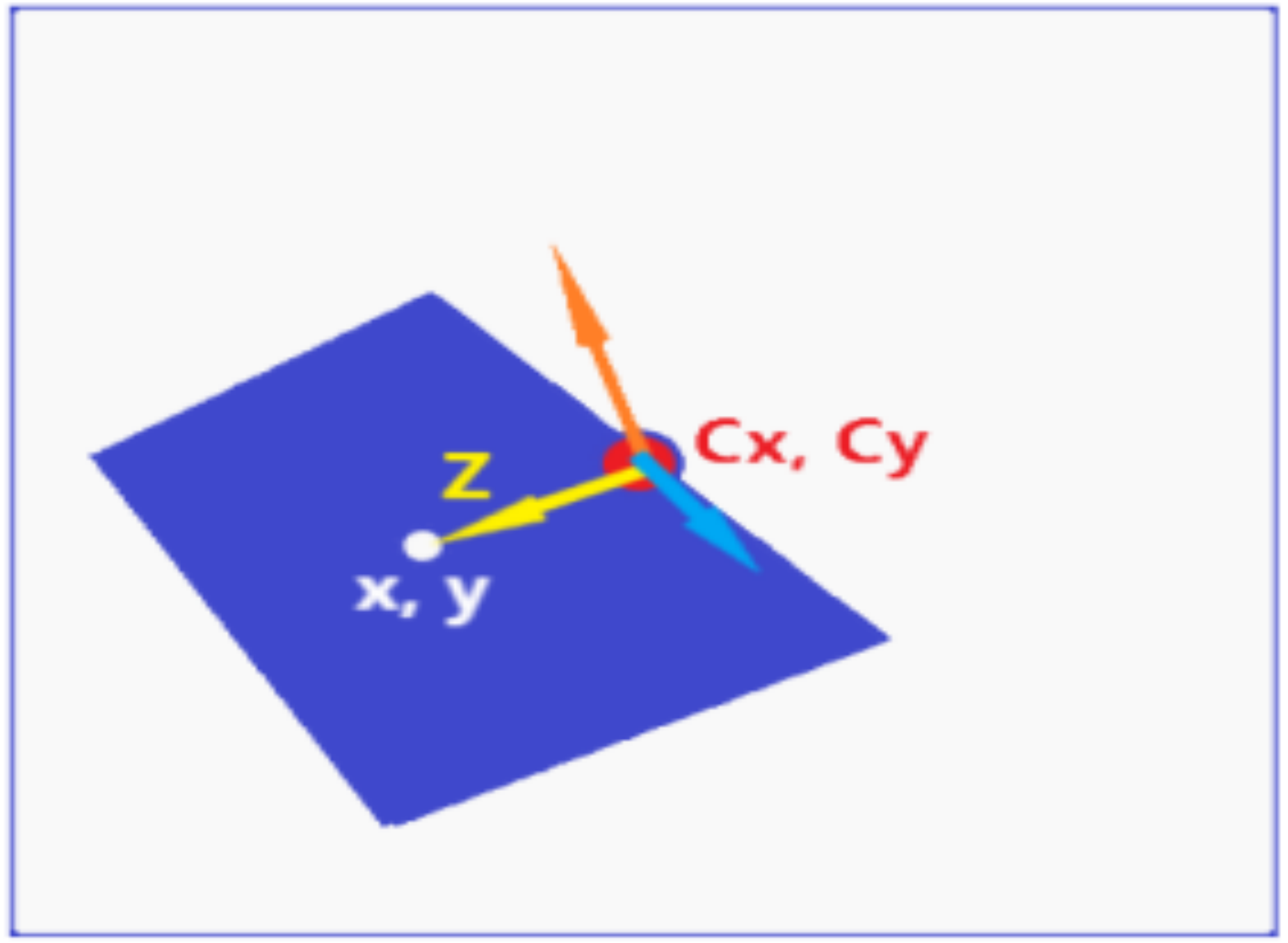
Challenge – Different annotations



Region of Interest (RoI)



Rectified annotation



Solution for orientation

$$\boldsymbol{v} = [(x - c_x)/f_x, (y - c_y)/f_y, 1]$$

v – 3D orientation towards the center of RoI

(x, y) – the center of RoI

(c_x, c_y) – the center of the 2D camera

f_x, f_y – focal lengths of X and Y

Solution for XYZ axes

aligning the Z axis $[0, 0, 1]$ to $\boldsymbol{v}$

Z axes - $[X_v, Y_v, Z_v]$

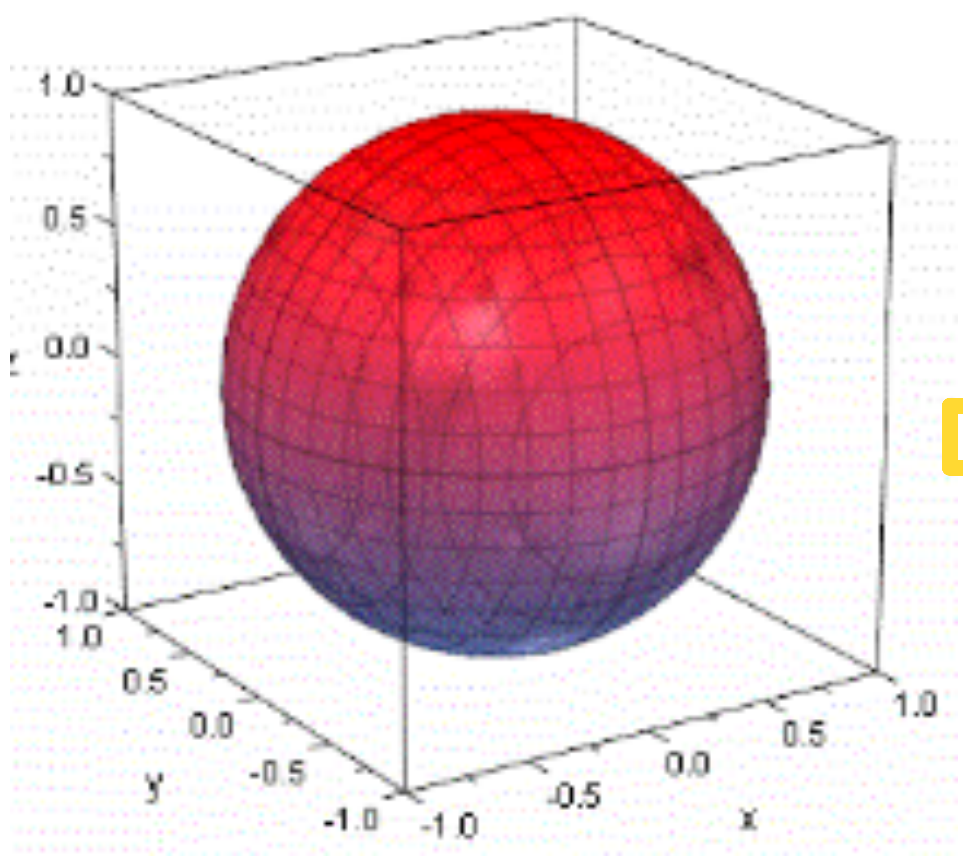
$X_v = [0, 1, 0] \times Z_v$,

$Y_v = Z_v \times X_v$,

$Z_v = v / ||v||_2$

Multi-class network architecture for the single view - Output

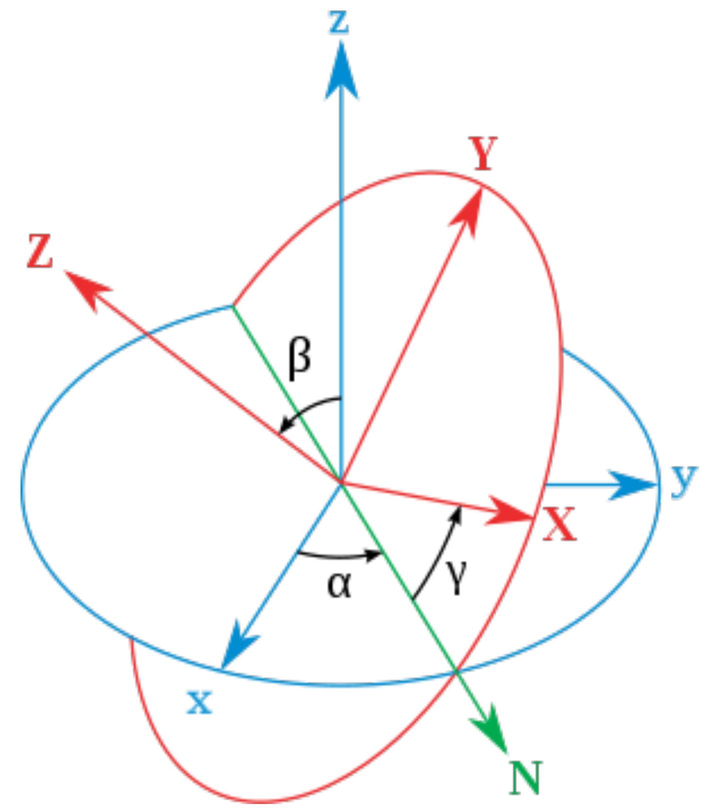
SO(3)



Sampling uniform rotations

Discretization

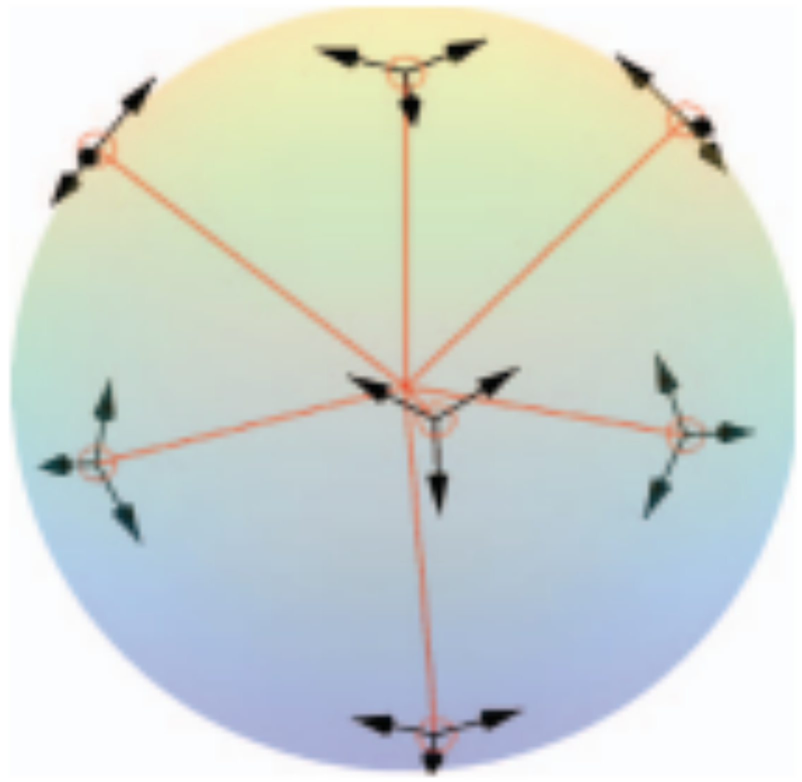
Euler angles



Non-uniform tessellation

Uniform tessellation

Discretized Euler bins



Scoring scheme for bin-delta pair

Any rotation is represented as a bin-delta pai

$$b_i^R = \begin{cases} \theta_1 & : i \in NN_1(R) \\ \theta_2 & : i \in NN_k(R) \setminus NN_1(R) \\ 0 & \end{cases}, \quad (b^R, d^R)$$

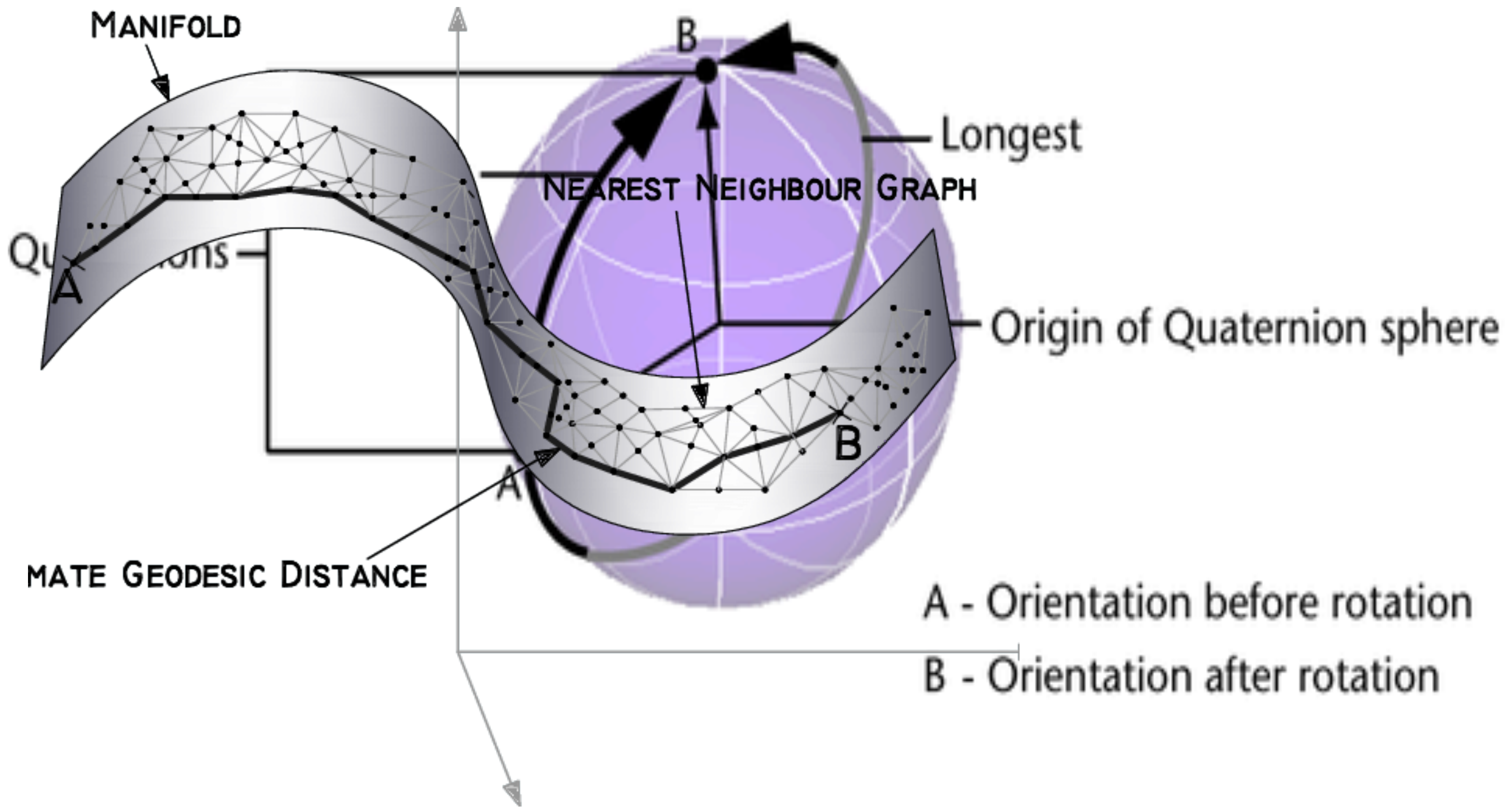
b^R_i - confidence of R belonging to bin i

$$d_i^R = \begin{cases} R \cdot R_i^T & : i \in NN_k(R) \\ 0 & : \text{Otherwise} \end{cases}$$

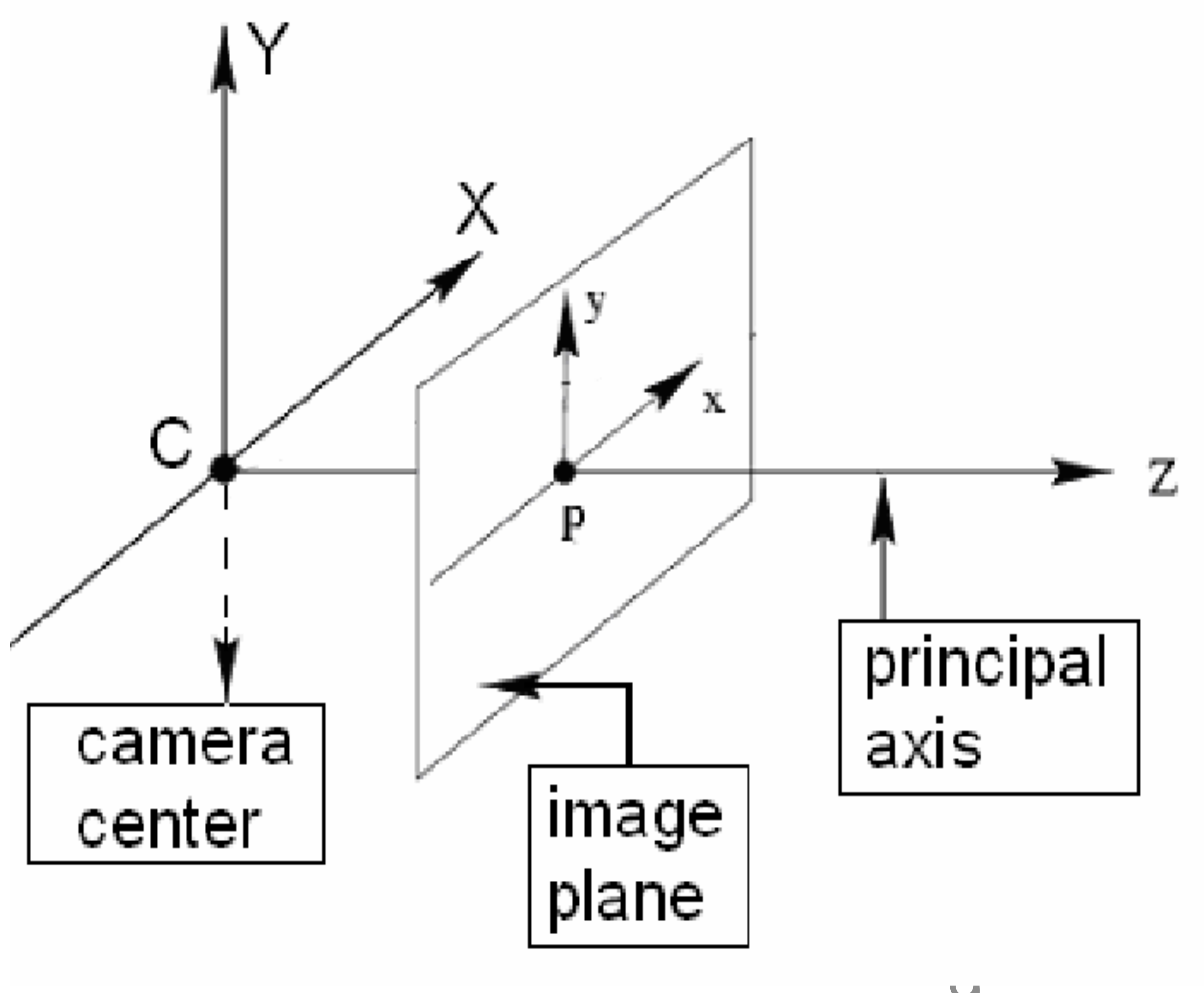
d^R_i - deviation from the centre $\hat{R}_i$

Geodesic distance w.r.t. Manifold

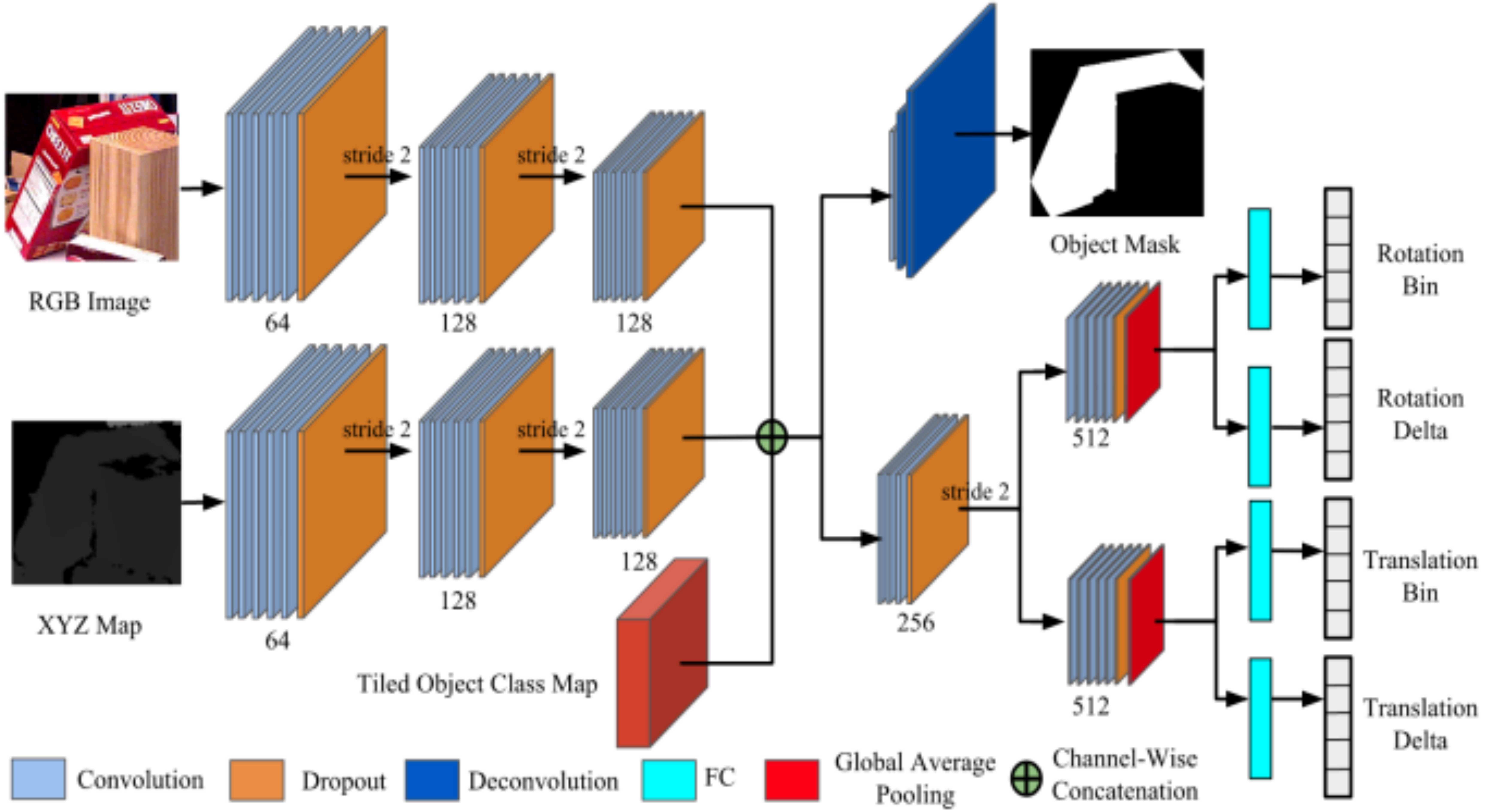
Scoring scheme for bin-delta pair



Translation 3D vector



Multi-class network architecture for the single view

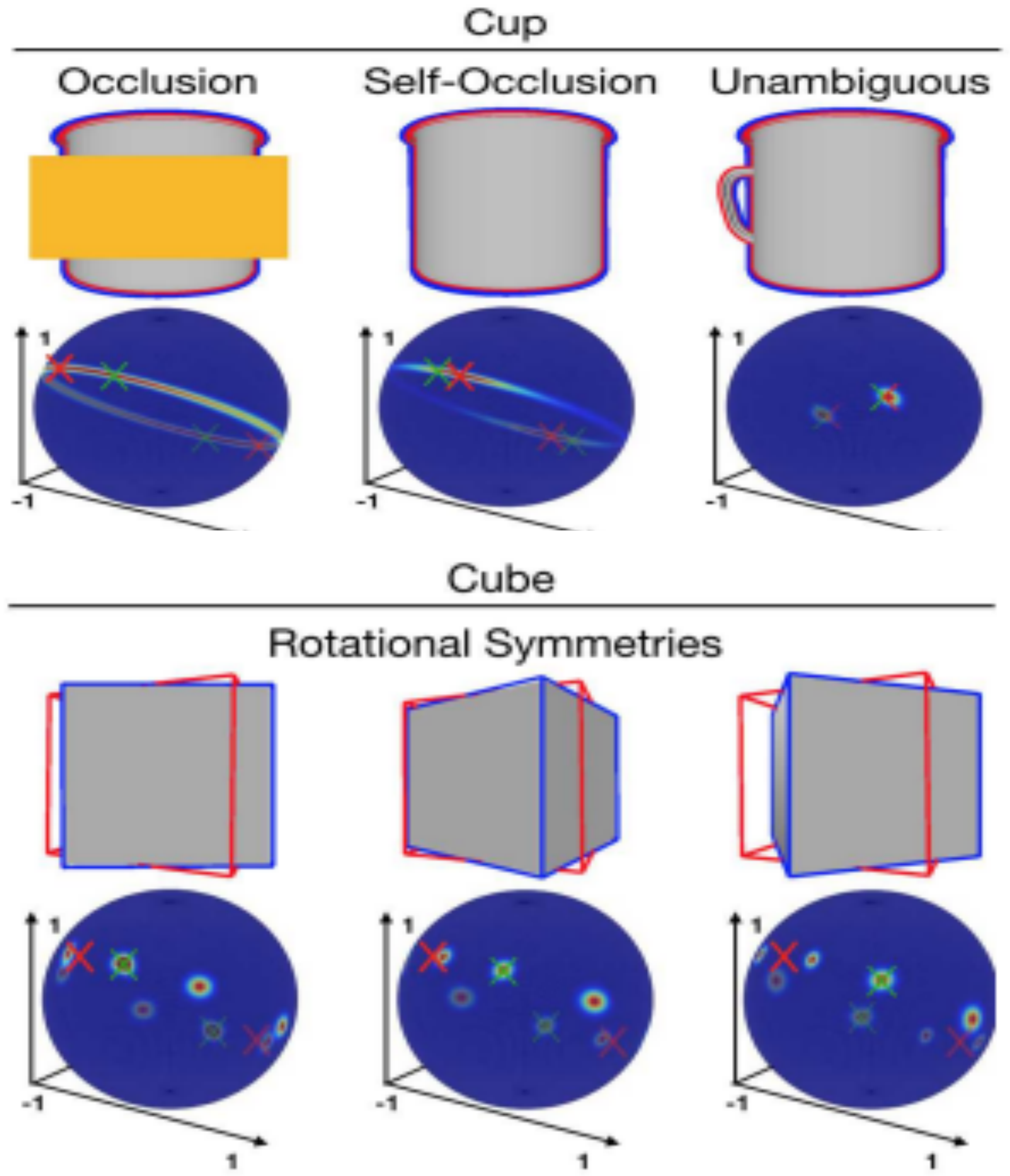


$$\mathcal{L} = l_{seg} + l_{R_b}(\widetilde{\boldsymbol{b}^R}, \boldsymbol{b}^R) + l_{R_d}(\widetilde{\boldsymbol{d}^R}, \boldsymbol{d}^R) + \sum_{i \in \{X,Y,Z\}} \left(l_{T_b}(\widetilde{\boldsymbol{b}^{T_i}}, \boldsymbol{b}^{T_i}) + l_{T_d}(\widetilde{\boldsymbol{d}^{T_i}}, \boldsymbol{d}^{T_i}) \right)$$

DR

Multi-view object pose estimation learning architecture

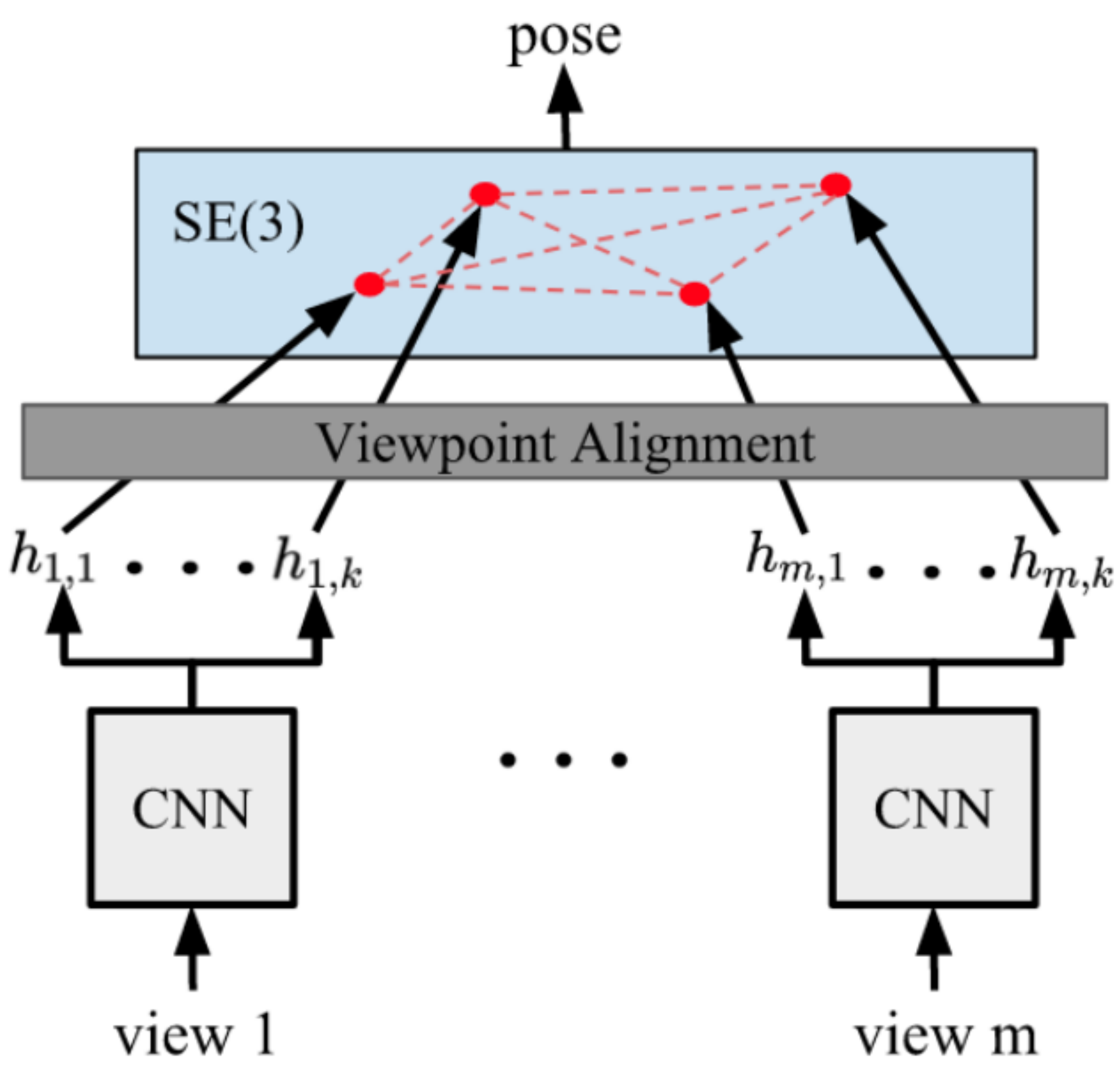
Limitations



https://arxiv.org/abs/1812.00287

Challenges
Ambiguities due to occlusion and object symmetry

Solution
Additional views of the same instance

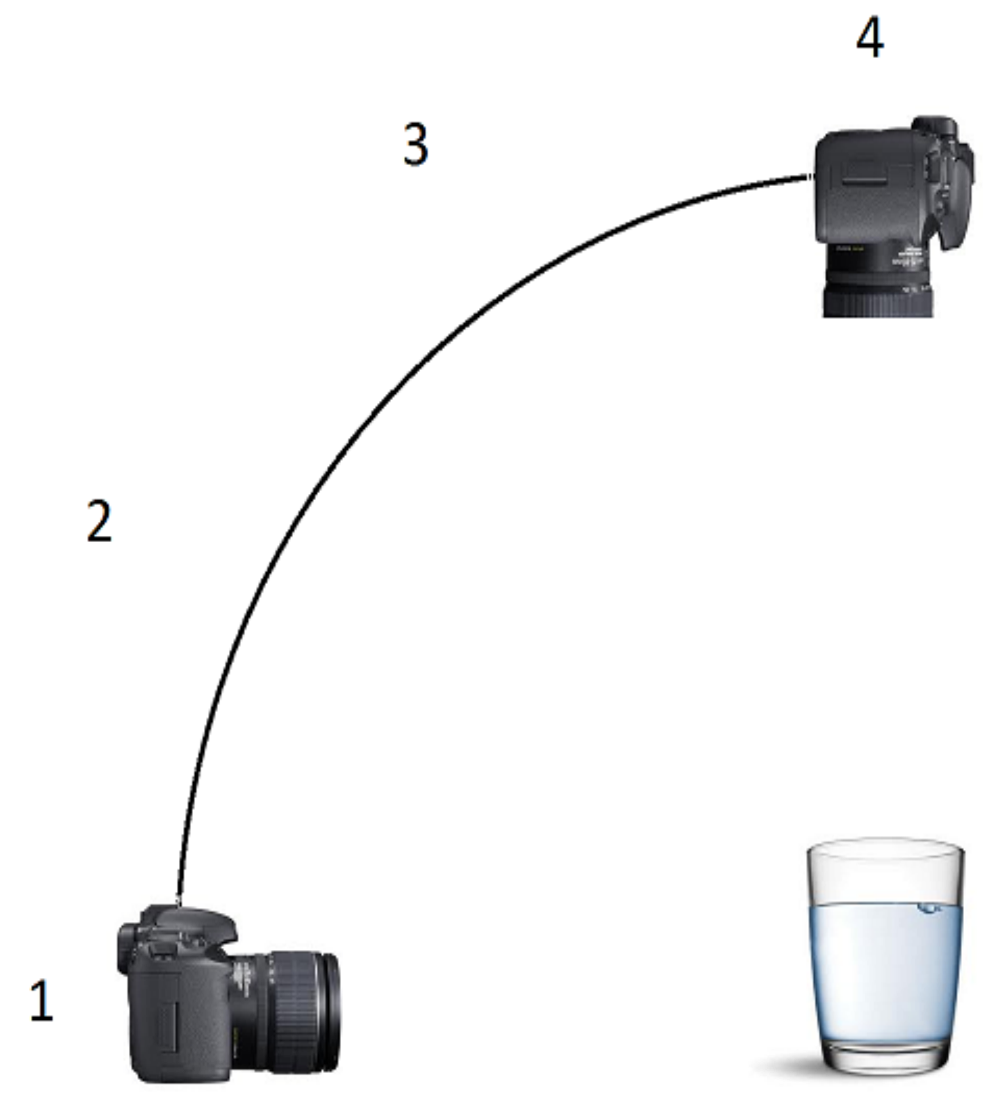


(d) Multi-View Multi-Class Framework

Figure (d) illustrates a multi-view, multi-class pose estimation framework where $h_{m,k,}$ the k^{th} pose hypothesis on view *m*. The network selects pose hypotheses, computed from the single-view multi-class network, based on a distance metric robust to object symmetry

Multi-view object pose estimation – Hypothesis voting

Multiple views of an object



$$\mathcal{H} = \{h_{1,1}, \cdots, h_{i,j}, \cdots, h_{n,K}\}$$

H – hypothesis from n views,

$h_{i,j}$ - pose hypothesis *j* in view *i* with respect to the camera coordinate of view 1

$$h_1 = (R_1, T_1) \text{ and } h_2 = (R_2, T_2)$$

Discrepancy between the hypothesis

$$D(h_1, h_2) = \frac{1}{m} \sum_{x_1 \in \mathcal{M}} \min_{x_2 \in \mathcal{M}} \|(R_1 x_1 + T_1) - (R_2 x_2 + T_2)\|_2$$

Voting score

$$V(h_{i,j}) = \sum_{h_{p,q} \in \mathcal{H} \setminus h_{i,j}} \max\left(\sigma - D(h_{i,j}, h_{p,q}), 0\right)$$

Decouple translation and rotation

$$\widetilde{D}(h_1, h_2) = \|T_1 - T_2\|_2 + \frac{1}{m} \sum_{x_1 \in \mathcal{M}} \min_{x_2 \in \mathcal{M}} \|R_1 x_1 - R_2 x_2\|_2$$

Pre-computed pairwise distances

$$\frac{1}{m} \sum_{x_1 \in \mathcal{M}} \min_{x_2 \in \mathcal{M}} \|R_1 x_1 - R_2 x_2\|_2 \approx \frac{1}{m} \sum_{x_1 \in \mathcal{M}} \min_{x_2 \in \mathcal{M}} \|\hat{R}_{N_1(R_1)} x_1 - \hat{R}_{N_1(R_2)} x_2\|_2$$

Evaluation metric & Ablative study

Pose estimation: ADD-S / reprojection error / mPCK (mean Per-Class Precision at K)

$$D(h_1, h_2) = \frac{1}{m} \sum_{x_1 \in \mathcal{M}} \min_{x_2 \in \mathcal{M}} \|(R_1 x_1 + T_1) - (R_2 x_2 + T_2)\|_2$$

Viewpoint estimation: AVP (Average Viewpoint Precision) & AOS (Average Orientation Similarity)

$$AOS = \frac{1}{11} \sum_{r \in \{0, 0.1, .., 1\}} \max_{\tilde{r}: \tilde{r} \geq r} s(\tilde{r})$$

$$r = \frac{TP}{TP+FN}$$

The orientation similarity *s* ∊ [0, 1] at recall *r* is a normalized ([0..1]) variant of the cosine similarity defined as

$$s(r) = \frac{1}{|\mathcal{D}(r)|} \sum_{i \in \mathcal{D}(r)} \frac{1 + \cos \Delta_\theta^{(i)}}{2} \delta_i$$

Ablative study

Method	RGB			RGB-D	
	YCB-Video	JHU	ObjectNet-3D	YCB-Video	JHU
plain	61.0	25.0	51.7 / 38.3	61.8	19.6
BD + Seg	66.2	26.3	50.3* / 41.3*	89.5	70.0
BD + TC	68.5	29.3	**56.0 / 50.0**	90.1	76.4
Sep-Branch + Seg + BD	73.8	31.6	52.5* / 42.9*	90.2	77.7
Sep-Net + Seg + BD	62.1	28.7	NA	87.1	66.9
MCN (Seg + TC + BD)	**80.2**	**33.9**	NA	**90.8**	**78.9**

BD – Bin & Delta representation

Seg – Deep supervision of object segmentation

TC – Tiled Class map

Sep-Branch – Separate output Branch for each object

Sep-Net – Separate Network for each object

Results

MCN on YCB-Video

MCN on JHUScene-50

Conclusion

- A multi-class CNN architecture for accurate pose estimation with three novel features:
  - a) a single pose prediction branch that is coupled with a discriminative pose representation in SE(3) and is shared by multiple classes
  - b) a method to embed object class labels into the learning process by concatenating a tiled class map with convolutional layers
  - c) deep supervision with an object mask which improves the generalization from synthetic data to real images
- A multi-view fusion framework that reduces single-view ambiguity based on a voting scheme
- An efficient implementation is proposed to enable fast hypothesis selection during inference

PVN3D

A Deep Point-wise 3D Keypoints Voting Network for 6DoF Pose Estimation

By: Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, Jian Sun

Presented by: Wai-Ting (Bruce) Li

The Authors

- Yisheng He
  - 4th year PhD at Hong Kong University of Science and Technology
  - Advised by: Prof. Qifeng Chen, Prof. Long Quan, and Dr. Jian Sun
- Wei Sun
  - Affiliated with Megvii Inc.
- Etc.

Background

- The **6DoF pose estimation problem** is to estimate the 3D rigid body transformation from object coordinate system to camera coordinate system
- The problem is challenging due to variations of lighting conditions, sensor noise, occlusion of scenes, etc.

Benefits if we solve this problem

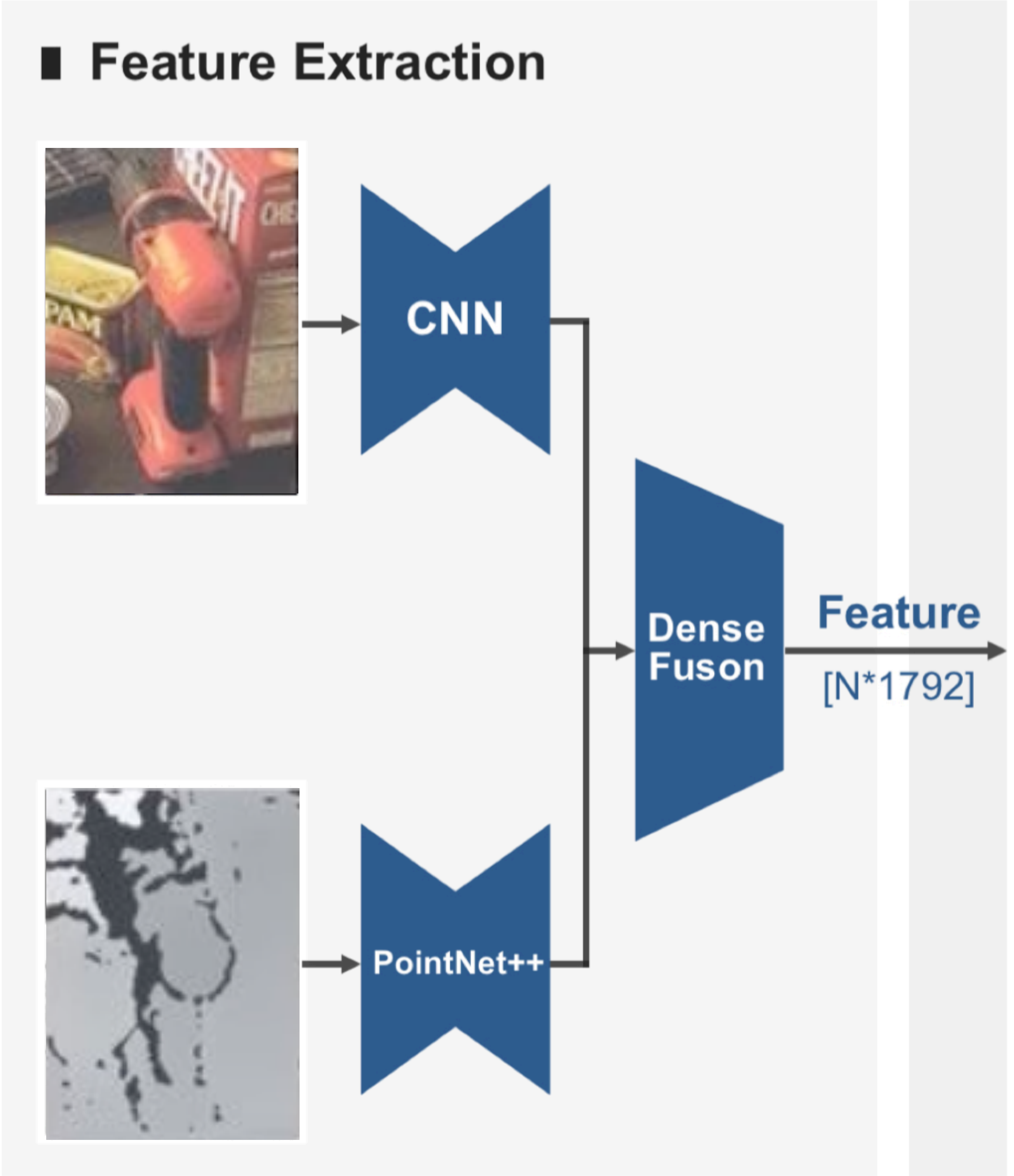
Knowing the precise pose of an object will be useful to the following tasks:

- Object recognition and tracking
- Robot manipulation
- Autonomous navigation
- Augmented reality

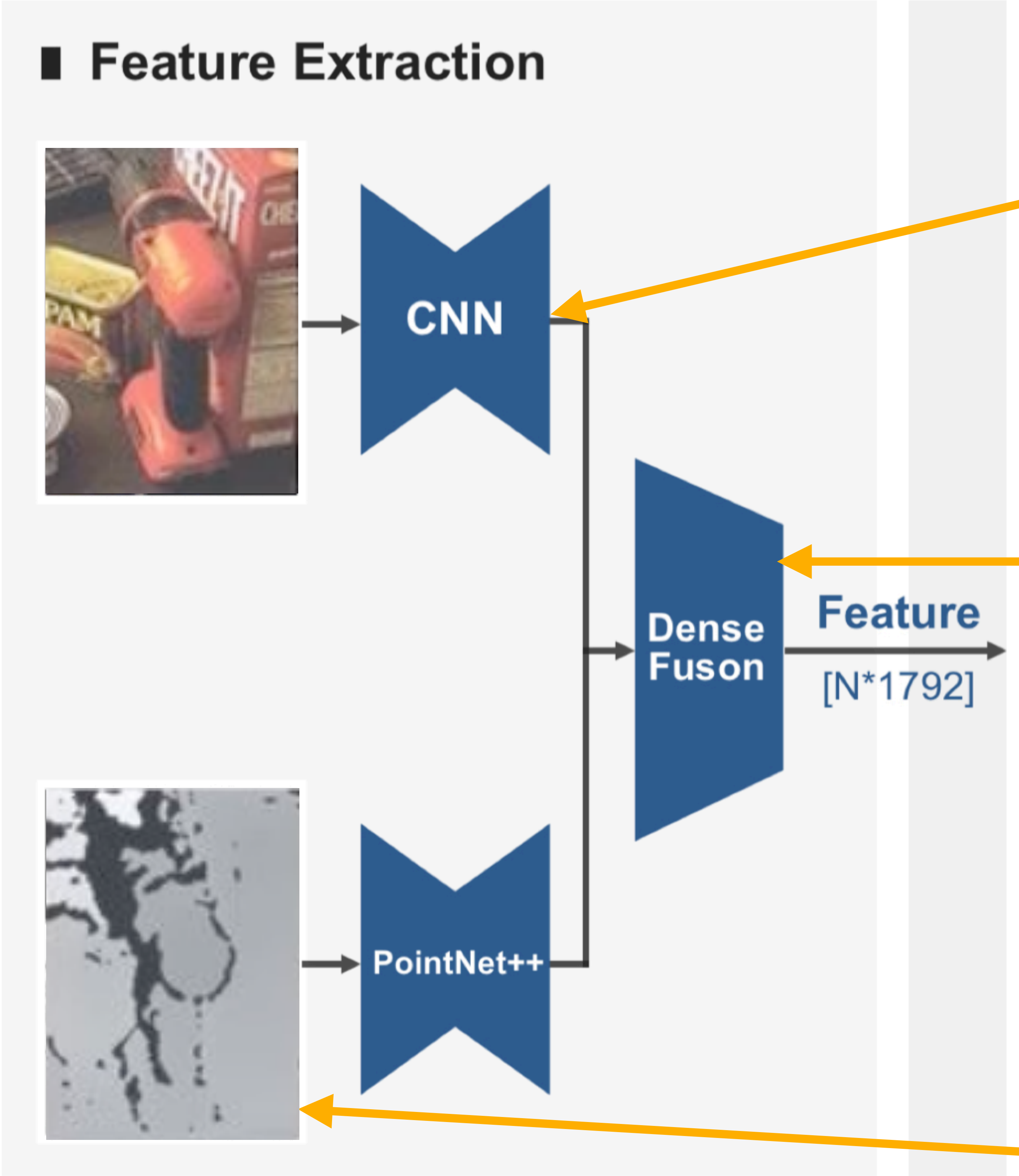
Contributions

1. A deep 3D keypoints Hough voting network with instance & semantic segmentation for 6DoF pose estimation of a single RGBD image,
2. State-of-the-art 6DoF pose estimation performance on YCB-Video and LineMOD datasets,
3. Comprehensive analysis and comparison among 3D keypoint-based, directly regression, and dense correspondence methods.

Approach - Single Instance



Approach - Single Instance

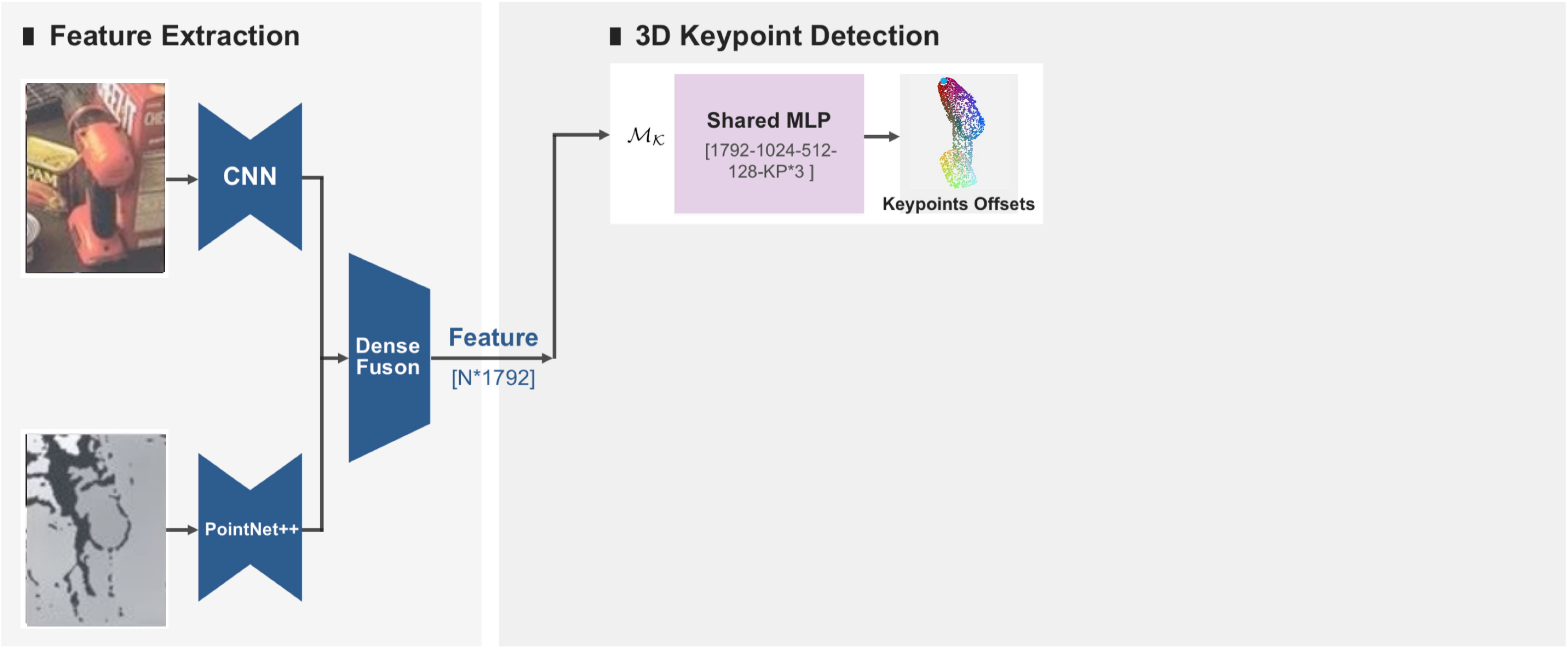


Pretrained ResNet34 on ImageNet

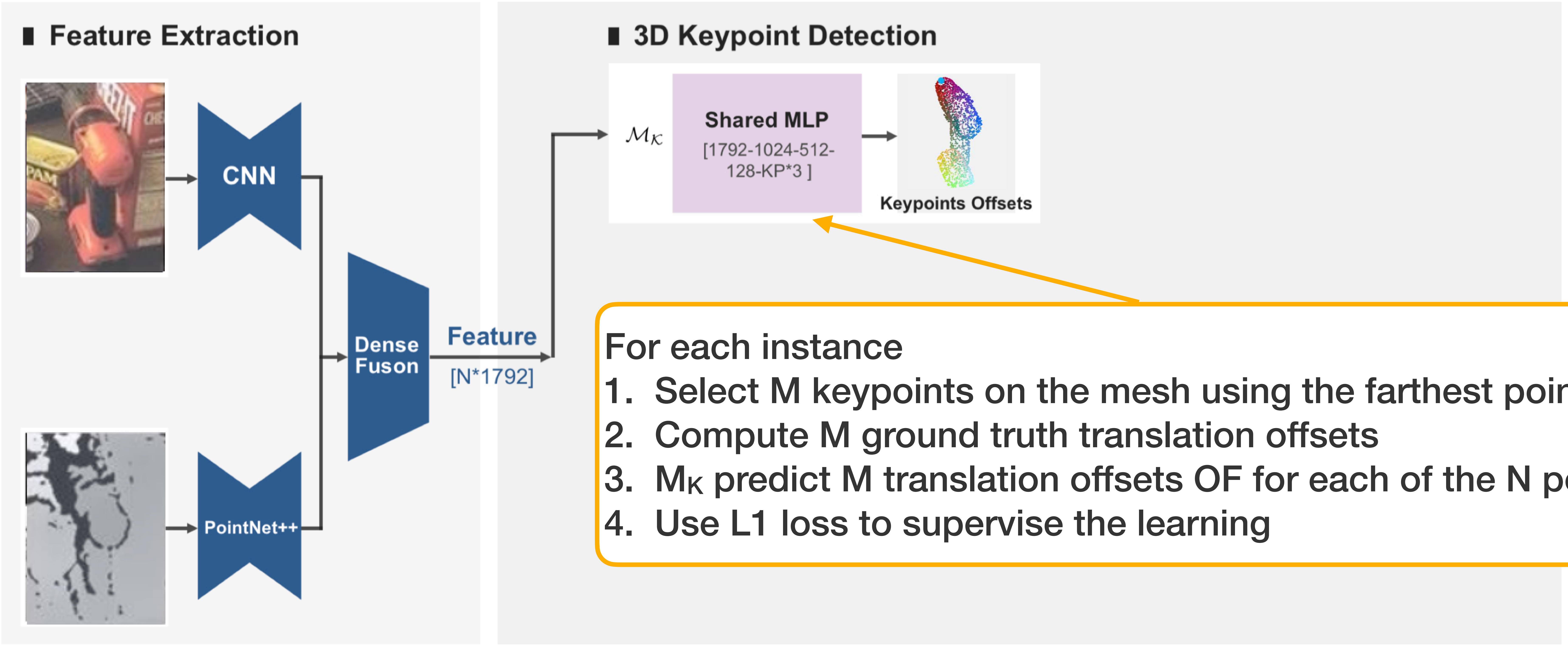
DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion (coming up in the next lecture)

Project each pixel in the depth image onto a 3D space using camera calibration parameters to form a point cloud

Approach - Single Instance



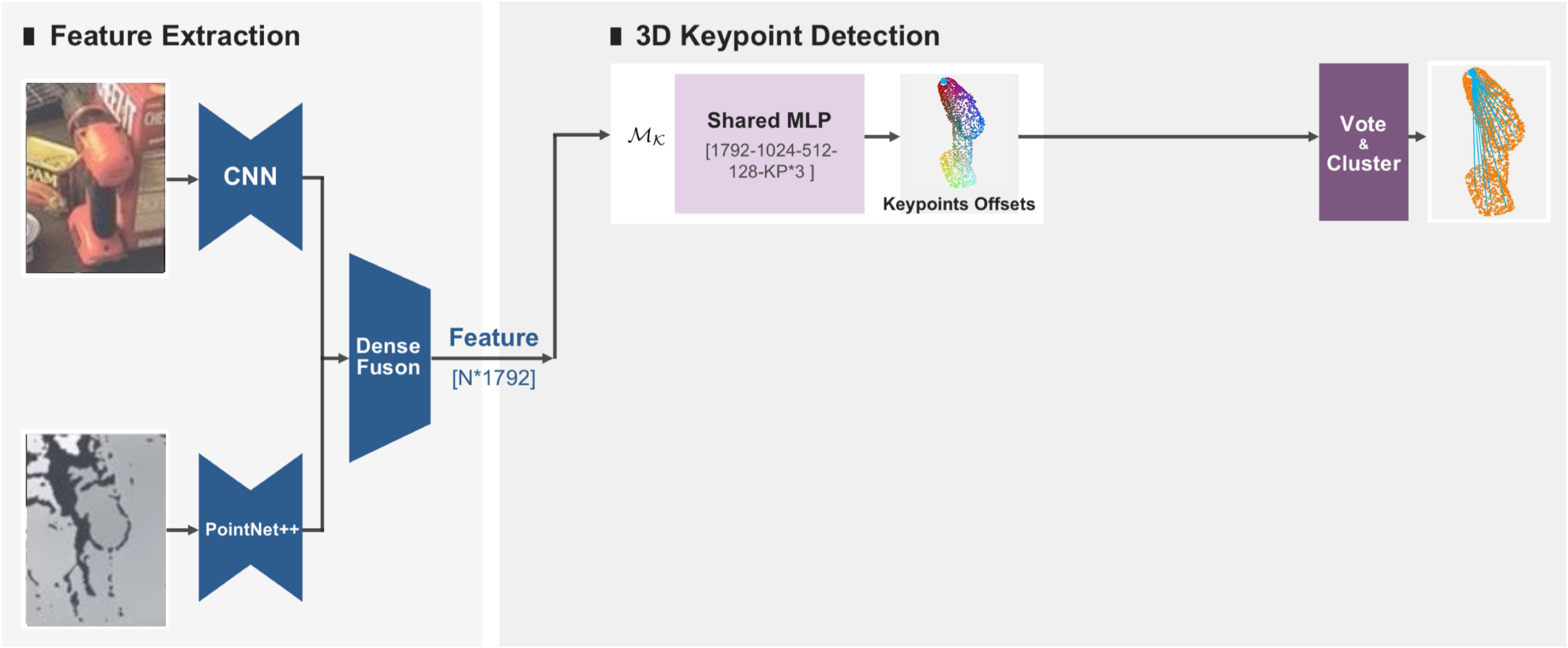
Approach - Single Instance



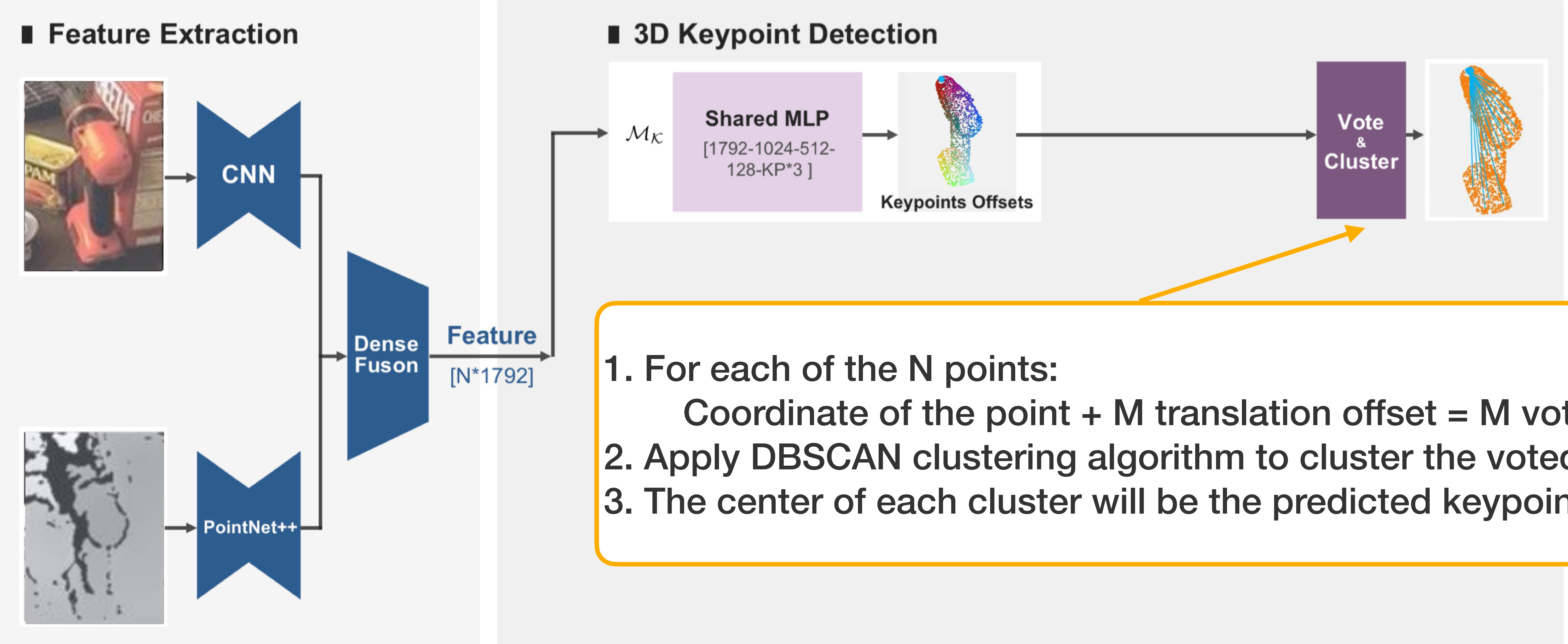
For each instance

1. Select M keypoints on the mesh using the farthest point sampling (FPS) algorithm
2. Compute M ground truth translation offsets
3. M_K predict M translation offsets OF for each of the N points
4. Use L1 loss to supervise the learning

Approach - Single Instance

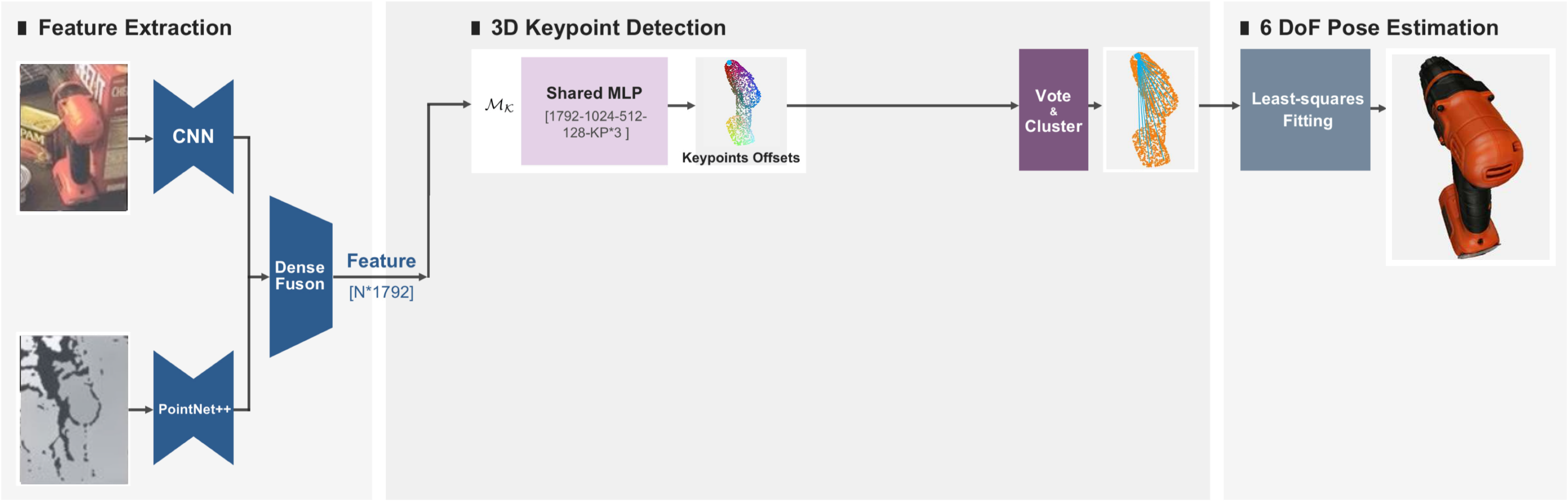


Approach - Single Instance

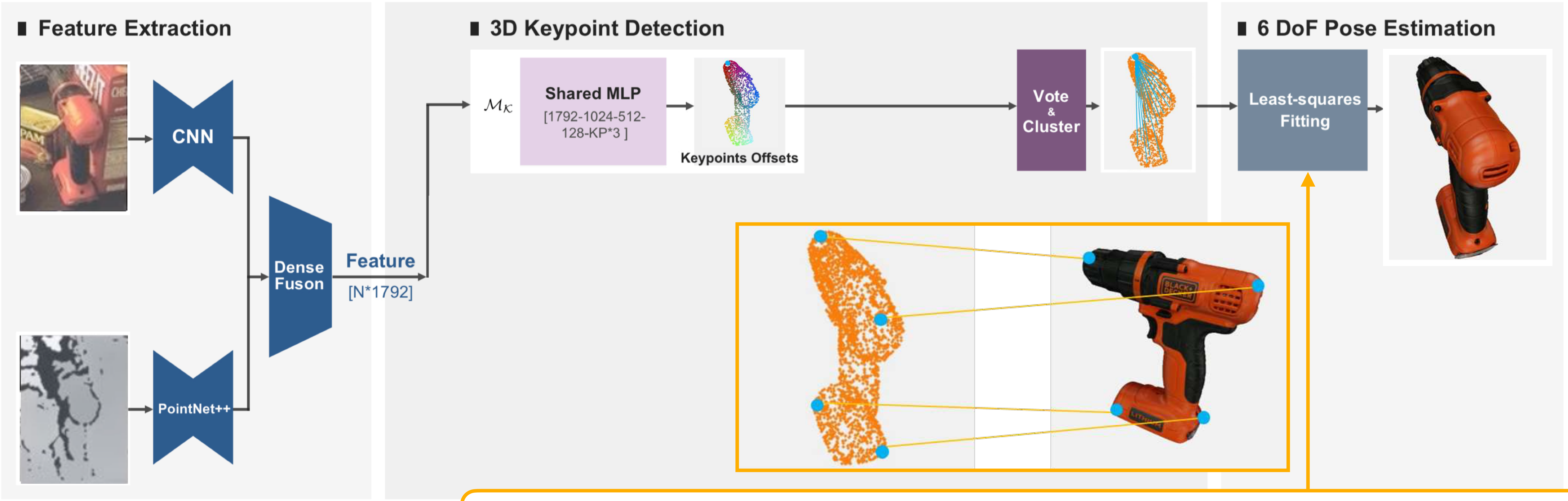


1. For each of the N points:
   Coordinate of the point + M translation offset = M voted keypoints
2. Apply DBSCAN clustering algorithm to cluster the voted keypoints
3. The center of each cluster will be the predicted keypoints

Approach - Single Instance

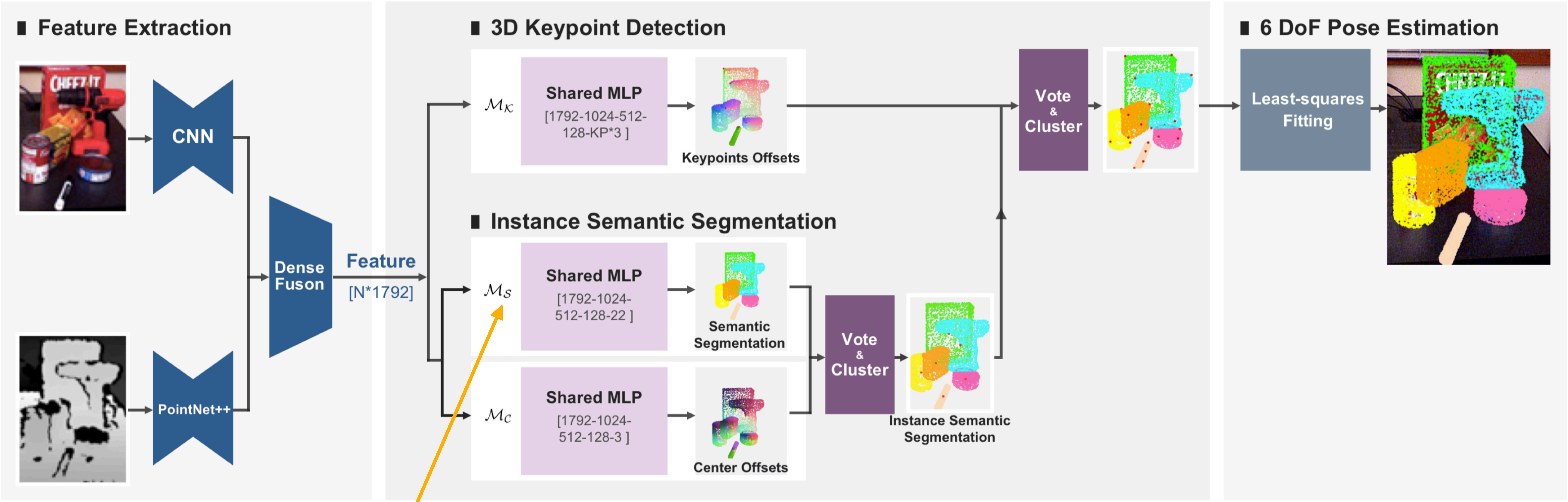


Approach - Single Instance



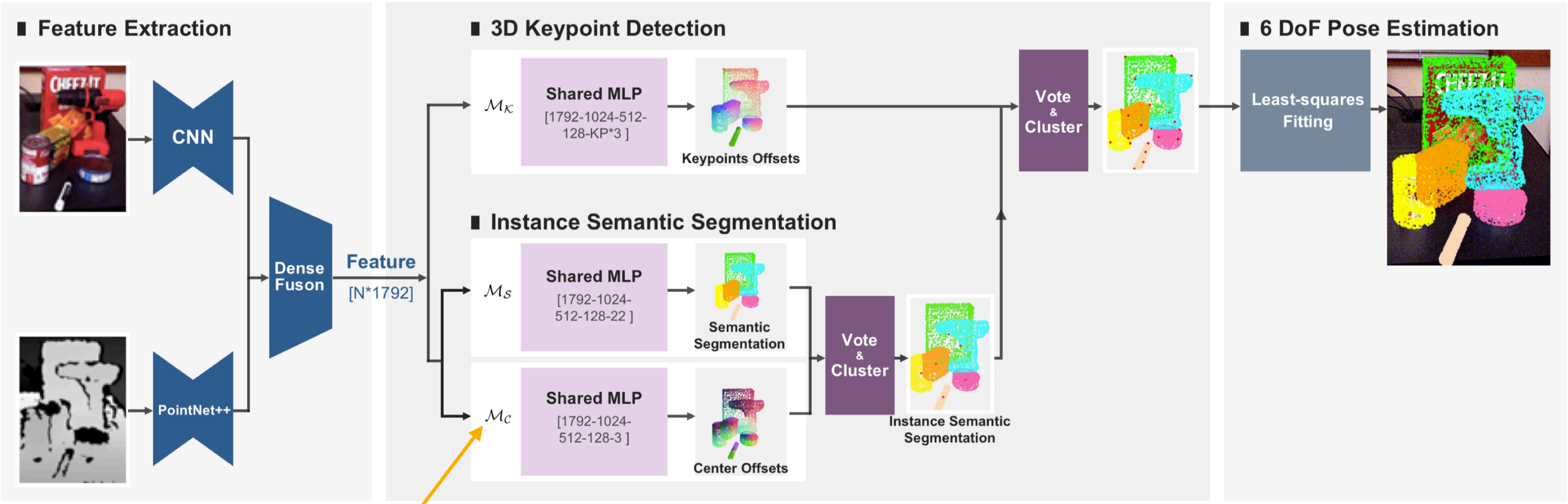
Learn the 3D rotation matrix, R, and the translation vector, t

Approach - Multiple Instances



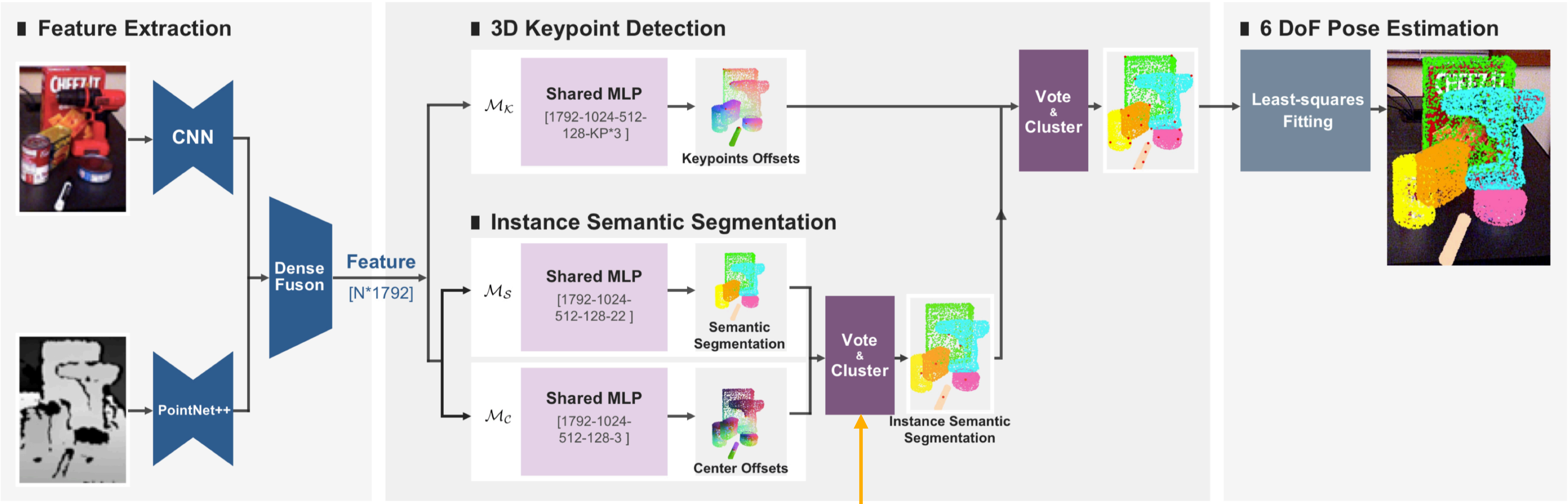
M_S predicts the per-point semantic labels

Approach - Multiple Instances



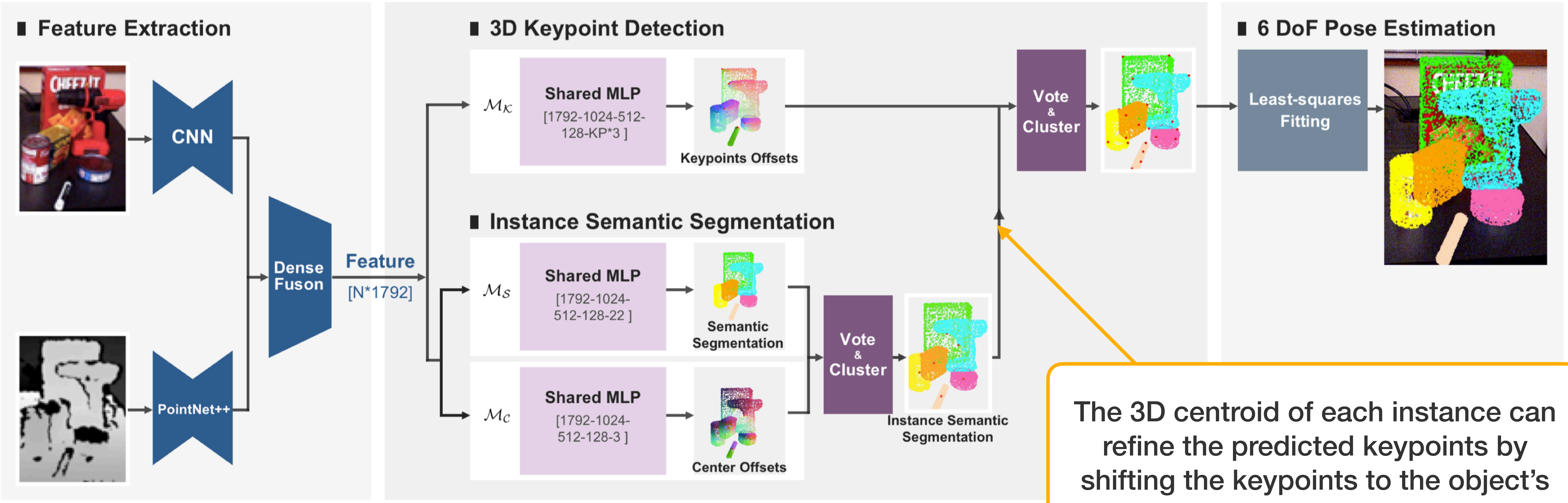
M_C predicts the Euclidean translation offset to the center of the objects each points belongs to and help to distinguish different instance

Approach - Multiple Instances



Together, M_S + M_C can predict 3D instance segmentation

Approach - Multiple Instances



The 3D centroid of each instance can refine the predicted keypoints by shifting the keypoints to the object's true center of mass.

Evaluation Metrics

- **ADD**: the average distance between object vertexes transformed by the 6D pose and the ground truth pose
- **ADD-S**: a metric for symmetric objects where the distances are computed based on the closest point
- **ADD-S AUC**: the area under the accuracy-threshold curve, which is obtained by varying the distance threshold in evaluation
- **ADD(S) AUC**: similar to ADD-S AUC but calculate ADD for non-symmetric objects and ADD-S for symmetric objects

Quantitative Results

	Without Iterative Refinement						With Iterative Refinement					
	PoseCNN[52]		DF(per-pixel)[50]		PVN3D		PoseCNN+ICP[52]		DF(iterative)[50]		PVN3D+ICP	
	ADDS	ADD(S)	ADDS	ADD(S)	ADDS	ADD(S)	ADDS	ADD(S)	ADDS	ADD(S)	ADDS	ADD(S)
002_master_chef_can	83.9	50.2	95.3	70.7	96.0	**80.5**	95.8	68.1	**96.4**	73.2	95.2	79.3
003_cracker_box	76.9	53.1	92.5	86.9	**96.1**	**94.8**	92.7	83.4	95.8	94.1	94.4	91.5
004_sugar_box	84.2	68.4	95.1	90.8	97.4	96.3	**98.2**	**97.1**	97.6	96.5	97.9	96.9
005_tomato_soup_can	81.0	66.2	93.8	84.7	**96.2**	88.5	94.5	81.8	94.5	85.5	95.9	**89.0**
006_mustard_bottle	90.4	81.0	95.8	90.9	97.5	96.2	**98.6**	**98.0**	97.3	94.7	98.3	97.9
007_tuna_fish_can	88.0	70.7	95.7	79.6	96.0	89.3	**97.1**	83.9	**97.1**	81.9	96.7	**90.7**
008_pudding_box	79.1	62.7	94.3	89.3	97.1	95.7	97.9	96.6	96.0	93.3	**98.2**	**97.1**
009_gelatin_box	87.2	75.2	97.2	95.8	97.7	96.1	98.8	98.1	98.0	96.7	**98.8**	**98.3**
010_potted_meat_can	78.5	59.5	89.3	79.6	93.3	**88.6**	92.7	83.5	90.7	83.6	**93.8**	87.9
011_banana	86.0	72.3	90.0	76.7	96.6	93.7	97.1	91.9	96.2	83.3	**98.2**	**96.0**
019_pitcher_base	77.0	53.3	93.6	87.1	97.4	96.5	**97.8**	96.9	97.5	96.9	97.6	**96.9**
021_bleach_cleanser	71.6	50.3	94.4	87.5	96.0	93.2	96.9	92.5	95.9	89.9	**97.2**	**95.9**
024_bowl	69.6	69.6	86.0	86.0	90.2	90.2	81.0	81.0	89.5	89.5	**92.8**	**92.8**
025_mug	78.2	58.5	95.3	83.8	97.6	95.4	94.9	81.1	96.7	88.9	**97.7**	**96.0**
035_power_drill	72.7	55.3	92.1	83.7	96.7	95.1	**98.2**	**97.7**	96.0	92.7	97.1	95.7
036_wood_block	64.3	64.3	89.5	89.5	90.4	90.4	87.6	87.6	**92.8**	**92.8**	91.1	91.1
037_scissors	56.9	35.8	90.1	77.4	**96.7**	**92.7**	91.7	78.4	92.0	77.9	95.0	87.2
040_large_marker	71.7	58.3	95.1	89.1	96.7	91.8	97.2	85.3	97.6	**93.0**	**98.1**	91.6
051_large_clamp	50.2	50.2	71.5	71.5	93.6	93.6	75.2	75.2	72.5	72.5	**95.6**	**95.6**
052_extra_large_clamp	44.1	44.1	70.2	70.2	88.4	88.4	64.4	64.4	69.9	69.9	**90.5**	**90.5**
061_foam_brick	88.0	88.0	92.2	92.2	96.8	96.8	97.2	97.2	92.0	92.0	**98.2**	**98.2**
ALL	75.8	59.9	91.2	82.9	95.5	91.8	93.0	85.4	93.2	86.1	**96.1**	**92.3**

Table 1. Quantitative evaluation of 6D Pose (ADD-S AUC [52], ADD(S) AUC [19]) on the YCB-Video Dataset. Symmetric objects' names are in bold.

Quantitative Results

	Without Iterative Refinement						With Iterative Refinement					
	PoseCNN[52]		DF(per-pixel)[50]		PVN3D		PoseCNN+ICP[52]		DF(iterative)[50]		PVN3D+ICP	
	ADDS	ADD(S)	ADDS	ADD(S)	ADDS	ADD(S)	ADDS	ADD(S)	ADDS	ADD(S)	ADDS	ADD(S)
002_master_chef_can	83.9	50.2	95.3	70.7	96.0	**80.5**	95.8	68.1	96.4	73.2	95.2	79.3
003_cracker_box	76.9	53.1	92.5	86.9	96.1	**94.8**	92.7	83.4	95.8	94.1	94.4	91.5
004_sugar_box	84.2	68.4	95.1	90.8	97.4	96.3	98.2	**97.1**	97.6	96.5	97.9	96.9
005_tomato_soup_can	81.0	66.2	93.8	84.7	96.2	88.5	94.5	81.8	94.5	85.5	95.9	**89.0**
006_mustard_bottle	90.4	81.0	95.8	90.9	97.5	96.2	98.6	**98.0**	97.3	94.7	98.3	97.9
007_tuna_fish_can	88.0	70.7	95.7	79.6	96.0	89.3	97.1	83.9	97.1	81.9	96.7	**90.7**
008_pudding_box	79.1	62.7	94.3	89.3	97.1	95.7	97.9	96.6	96.0	93.3	98.2	**97.1**
009_gelatin_box	87.2	75.2	97.2	95.8	97.7	96.1	98.8	98.1	98.0	96.7	98.8	**98.3**
010_potted_meat_can	78.5	59.5	89.3	79.6	93.3	**88.6**	92.7	83.5	90.7	83.6	93.8	87.9
011_banana	86.0	72.3	90.0	76.7	96.6	93.7	97.1	91.9	96.2	83.3	98.2	**96.0**
019_pitcher_base	77.0	53.3	93.6	87.1	97.4	96.5	97.8	96.9	97.5	96.9	97.6	**96.9**
021_bleach_cleanser	71.6	50.3	94.4	87.5	96.0	93.2	96.9	92.5	95.9	89.9	97.2	**95.9**
024_bowl	69.6	69.6	86.0	86.0	90.2	90.2	81.0	81.0	89.5	89.5	92.8	**92.8**
025_mug	78.2	58.5	95.3	83.8	97.6	95.4	94.9	81.1	96.7	88.9	97.7	**96.0**
035_power_drill	72.7	55.3	92.1	83.7	96.7	95.1	98.2	**97.7**	96.0	92.7	97.1	95.7
036_wood_block	64.3	64.3	89.5	89.5	90.4	90.4	87.6	87.6	92.8	**92.8**	91.1	91.1
037_scissors	56.9	35.8	90.1	77.4	96.7	**92.7**	91.7	78.4	92.0	77.9	95.0	87.2
040_large_marker	71.7	58.3	95.1	89.1	96.7	91.8	97.2	85.3	97.6	**93.0**	98.1	91.6
051_large_clamp	50.2	50.2	71.5	71.5	93.6	93.6	75.2	75.2	72.5	72.5	95.6	**95.6**
052_extra_large_clamp	44.1	44.1	70.2	70.2	88.4	88.4	64.4	64.4	69.9	69.9	90.5	**90.5**
061_foam_brick	88.0	88.0	92.2	92.2	96.8	96.8	97.2	97.2	92.0	92.0	98.2	**98.2**
ALL	75.8	59.9	91.2	82.9	95.5	91.8	93.0	85.4	93.2	86.1	96.1	**92.3**

Table 1. Quantitative evaluation of 6D Pose (ADD-S AUC [52], ADD(S) AUC [19]) on the YCB-Video Dataset. Symmetric objects' names are in bold.

PVN3D achieves best ADD-S in 14/21 classes including the average overall on the YCB-Video dataset

Quantitative Results

	Without Iterative Refinement						With Iterative Refinement					
	PoseCNN[52]		DF(per-pixel)[50]		PVN3D		PoseCNN+ICP[52]		DF(iterative)[50]		PVN3D+ICP	
	ADDS	ADD(S)	ADDS	ADD(S)	ADDS	ADD(S)	ADDS	ADD(S)	ADDS	ADD(S)	ADDS	ADD(S)
002_master_chef_can	83.9	50.2	95.3	70.7	96.0	**80.5**	95.8	68.1	**96.4**	73.2	95.2	79.3
003_cracker_box	76.9	53.1	92.5	86.9	**96.1**	**94.8**	92.7	83.4	95.8	94.1	94.4	91.5
004_sugar_box	84.2	68.4	95.1	90.8	97.4	96.3	**98.2**	**97.1**	97.6	96.5	97.9	96.9
005_tomato_soup_can	81.0	66.2	93.8	84.7	**96.2**	88.5	94.5	81.8	94.5	85.5	95.9	**89.0**
006_mustard_bottle	90.4	81.0	95.8	90.9	97.5	96.2	**98.6**	**98.0**	97.3	94.7	98.3	97.9
007_tuna_fish_can	88.0	70.7	95.7	79.6	96.0	89.3	**97.1**	83.9	**97.1**	81.9	96.7	**90.7**
008_pudding_box	79.1	62.7	94.3	89.3	97.1	95.7	97.9	96.6	96.0	93.3	**98.2**	**97.1**
009_gelatin_box	87.2	75.2	97.2	95.8	97.7	96.1	98.8	98.1	98.0	96.7	**98.8**	**98.3**
010_potted_meat_can	78.5	59.5	89.3	79.6	93.3	**88.6**	92.7	83.5	90.7	83.6	**93.8**	87.9
011_banana	86.0	72.3	90.0	76.7	96.6	93.7	97.1	91.9	96.2	83.3	**98.2**	**96.0**
019_pitcher_base	77.0	53.3	93.6	87.1	97.4	96.5	**97.8**	96.9	97.5	96.9	97.6	**96.9**
021_bleach_cleanser	71.6	50.3	94.4	87.5	96.0	93.2	96.9	92.5	95.9	89.9	**97.2**	**95.9**
024_bowl	69.6	69.6	86.0	86.0	90.2	90.2	81.0	81.0	89.5	89.5	**92.8**	**92.8**
025_mug	78.2	58.5	95.3	83.8	97.6	95.4	94.9	81.1	96.7	88.9	**97.7**	**96.0**
035_power_drill	72.7	55.3	92.1	83.7	96.7	95.1	**98.2**	**97.7**	96.0	92.7	97.1	95.7
036_wood_block	64.3	64.3	89.5	89.5	90.4	90.4	87.6	87.6	**92.8**	**92.8**	91.1	91.1
037_scissors	56.9	35.8	90.1	77.4	**96.7**	**92.7**	91.7	78.4	92.0	77.9	95.0	87.2
040_large_marker	71.7	58.3	95.1	89.1	96.7	91.8	97.2	85.3	97.6	**93.0**	**98.1**	91.6
051_large_clamp	50.2	50.2	71.5	71.5	93.6	93.6	75.2	75.2	72.5	72.5	**95.6**	**95.6**
052_extra_large_clamp	44.1	44.1	70.2	70.2	88.4	88.4	64.4	64.4	69.9	69.9	**90.5**	**90.5**
061_foam_brick	88.0	88.0	92.2	92.2	96.8	96.8	97.2	97.2	92.0	92.0	**98.2**	**98.2**
ALL	75.8	59.9	91.2	82.9	95.5	91.8	93.0	85.4	93.2	86.1	**96.1**	**92.3**

Table 1. Quantitative evaluation of 6D Pose (ADD-S AUC [52], ADD(S) AUC [19]) on the YCB-Video Dataset. Symmetric objects' names are in bold.

PVN3D achieves best ADD(S) in 16/21 classes including the average overall on the YCB-Video dataset

Quantitative Results

	RGB			RGBD					
	PoseCNN DeepIM [26, 52]	PVNet [37]	CDPN [27]	Implicit ICP[45]	SSD-6D ICP[22]	Point-Fusion[50]	DF(per-pixel)[50]	DF(ite-rative)[50]	PVN3D
ape	77.0	43.6	64.4	20.6	65.0	70.4	79.5	92.3	**97.3**
benchvise	97.5	**99.9**	97.8	64.3	80.0	80.7	84.2	93.2	99.7
camera	93.5	86.9	91.7	63.2	78.0	60.8	76.5	94.4	**99.6**
can	96.5	95.5	95.9	76.1	86.0	61.1	86.6	93.1	**99.5**
cat	82.1	79.3	83.8	72.0	70.0	79.1	88.8	96.5	**99.8**
driller	95.0	96.4	96.2	41.6	73.0	47.3	77.7	87.0	**99.3**
duck	77.7	52.6	66.8	32.4	66.0	63.0	76.3	92.3	**98.2**
eggbox	97.1	99.2	99.7	98.6	**100.0**	99.9	99.9	99.8	99.8
glue	99.4	95.7	99.6	96.4	**100.0**	99.3	99.4	**100.0**	**100.0**
holepuncher	52.8	82.0	85.8	49.9	49.0	71.8	79.0	92.1	**99.9**
iron	98.3	98.9	97.9	63.1	78.0	83.2	92.1	97.0	**99.7**
lamp	97.5	99.3	97.9	91.7	73.0	62.3	92.3	95.3	**99.8**
phone	87.7	92.4	90.8	71.0	79.0	78.8	88.0	92.8	**99.5**
ALL	88.6	86.3	89.9	64.7	79.0	73.7	86.2	94.3	**99.4**

Table 3. Quantitative evaluation of 6D Pose on ADD(S) [19] metric on the LineMOD dataset. Objects with bold name are symmetric.

Quantitative Results

	RGB			RGBD					
	PoseCNN DeepIM [26, 52]	PVNet [37]	CDPN [27]	Implicit ICP[45]	SSD-6D ICP[22]	Point-Fusion[50]	DF(per-pixel)[50]	DF(iterative)[50]	PVN3D
ape	77.0	43.6	64.4	20.6	65.0	70.4	79.5	92.3	**97.3**
benchvise	97.5	**99.9**	97.8	64.3	80.0	80.7	84.2	93.2	99.7
camera	93.5	86.9	91.7	63.2	78.0	60.8	76.5	94.4	**99.6**
can	96.5	95.5	95.9	76.1	86.0	61.1	86.6	93.1	**99.5**
cat	82.1	79.3	83.8	72.0	70.0	79.1	88.8	96.5	**99.8**
driller	95.0	96.4	96.2	41.6	73.0	47.3	77.7	87.0	**99.3**
duck	77.7	52.6	66.8	32.4	66.0	63.0	76.3	92.3	**98.2**
eggbox	97.1	99.2	99.7	98.6	**100.0**	99.9	99.9	99.8	99.8
glue	99.4	95.7	99.6	96.4	**100.0**	99.3	99.4	**100.0**	**100.0**
holepuncher	52.8	82.0	85.8	49.9	49.0	71.8	79.0	92.1	**99.9**
iron	98.3	98.9	97.9	63.1	78.0	83.2	92.1	97.0	**99.7**
lamp	97.5	99.3	97.9	91.7	73.0	62.3	92.3	95.3	**99.8**
phone	87.7	92.4	90.8	71.0	79.0	78.8	88.0	92.8	**99.5**
ALL	88.6	86.3	89.9	64.7	79.0	73.7	86.2	94.3	**99.4**

Table 3. Quantitative evaluation of 6D Pose on ADD(S) [19] metric on the LineMOD dataset. Objects with bold name are symmetric.

PVN3D achieves best ADD(S) in 11/13 classes including the average overall on the LineMOD dataset

Qualitative Results

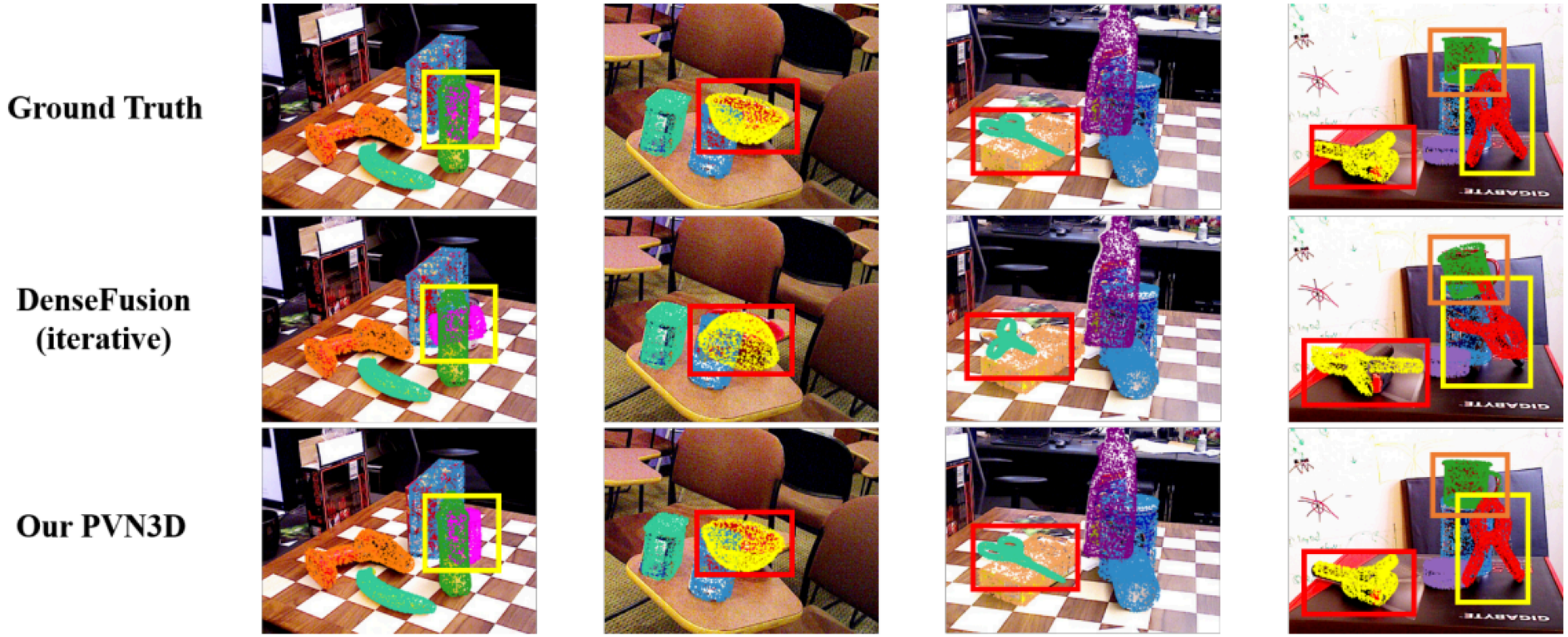
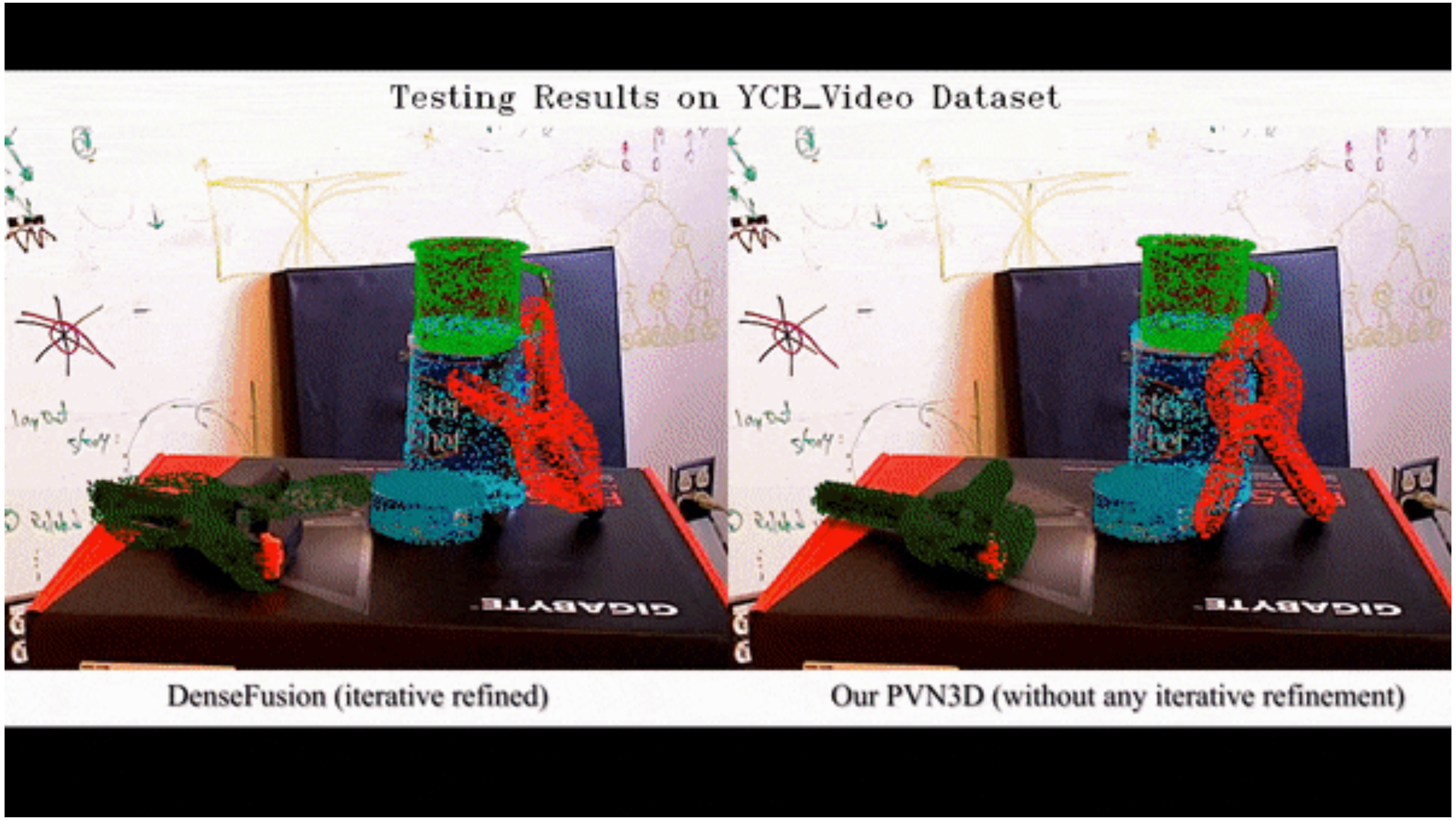


Figure 3. **Qualitative results on the YCB-Video dataset.** Points on different meshes in the same scene are in different colors. They are projected back to the image after being transformed by the predicted pose. We compare our PVN3D **without any iterative refinement procedure** to DenseFusion with iterative refinement (2 iterations). Our model distinguishes the challenging large clamp and extra-large clamp and estimates their poses well. Our model is also robust in heavily occluded scenes.

Qualitative Results



Conclusions

- 3D keypoints voting neural network with instance segmentation that outperforms several previous approaches
- The authors concluded that 3D keypoint-based approach is a promising direction to address the 6DoF pose estimation problem.
- A precise estimation of pose can be useful in object recognition and tracking, robot manipulation, autonomous navigation, and augmented reality

Limitations and Directions for Future Work

- Limitations
  - Tested only on a limited set of object categories
  - The proposed architecture is computationally expensive which limits its real-time application on, i.e., edge devices
- Future directions
  - Combine with other sensor data
  - More efficient models for feature extraction

DR

Learning RGB-D Feature Embeddings for Unseen Object Instance Segmentation

By: Yu Xiang, Christopher Xie, Arsalan Mousavian, Dieter Fox

Presented by: Andrew Scheffer, Ashwin Saxena

DeepRob

Learning RGB-D Embeddings for Unseen Object Instance Segmentation

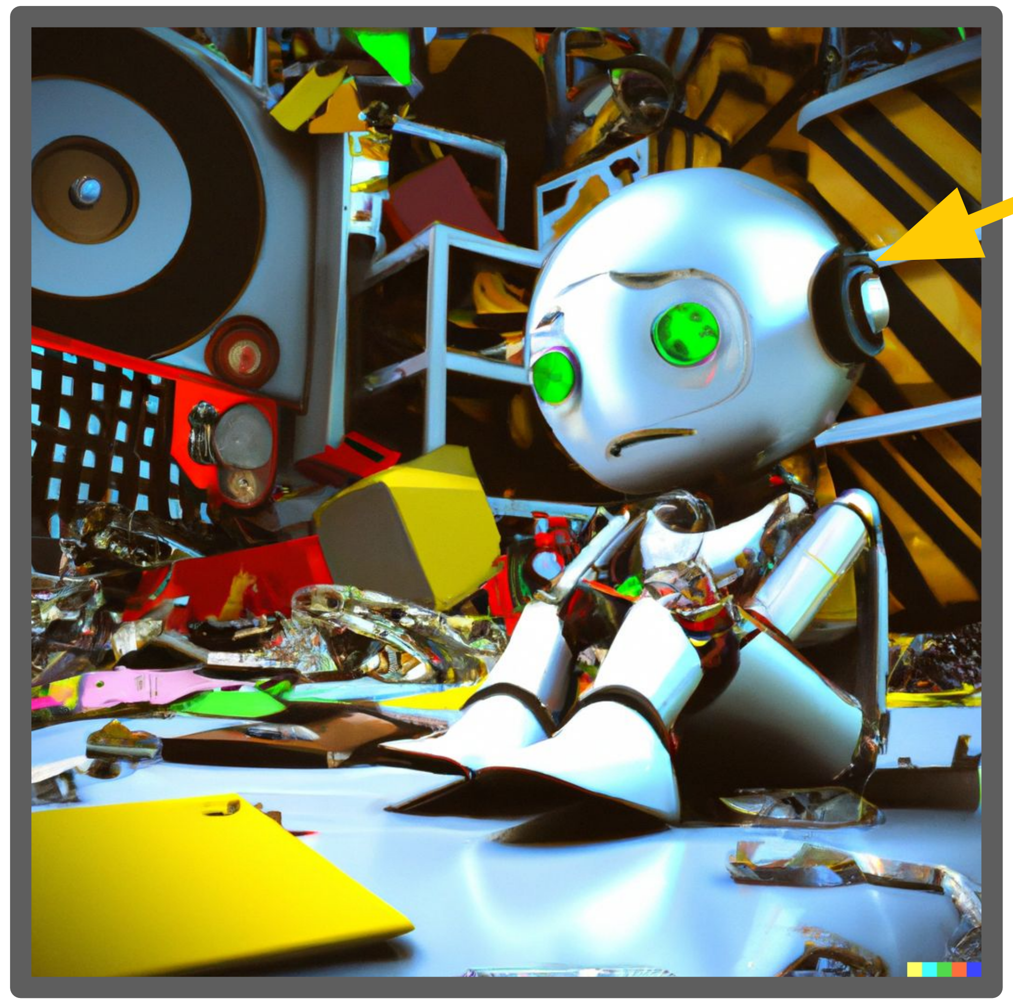
University of Michigan

Andrew Scheffer, Ashwin Saxena

Unseen Object Instance Segmentation (UOIS)

Segmenting **unseen objects** in cluttered scenes is an important task in robotic perception.

Useful in environments where the **types of objects are potentially unknown** (i.e. kitchens, machine shops, etc).



DeepSob

Related Work

Unseen Object Instance Segmentation

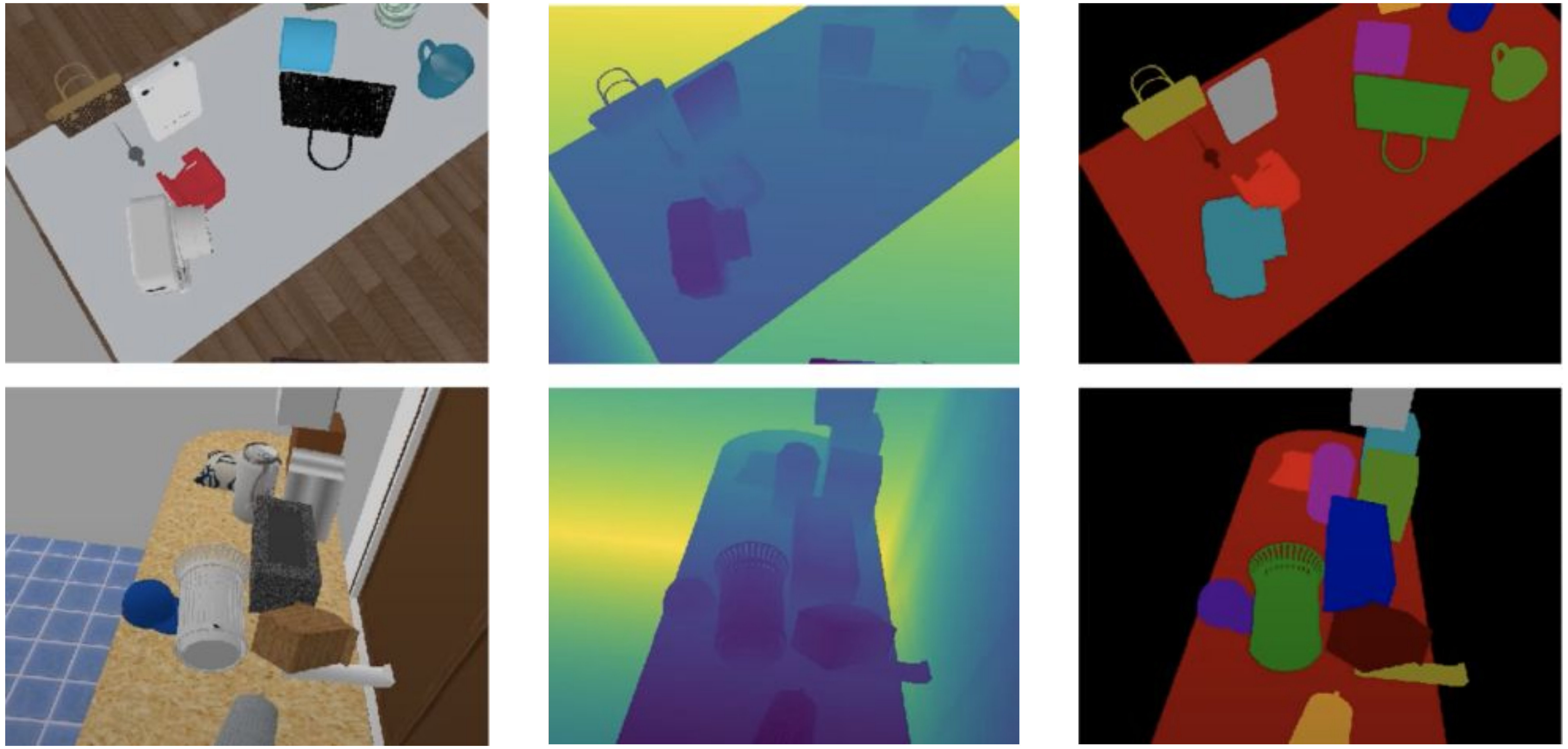
- Older methods based on edges, contours
- Over-segmentation problem
- sim-to-real gap with synthetic data

Deep Metric Learning

- Metric learning to learn feature representation
- Older methods used real images

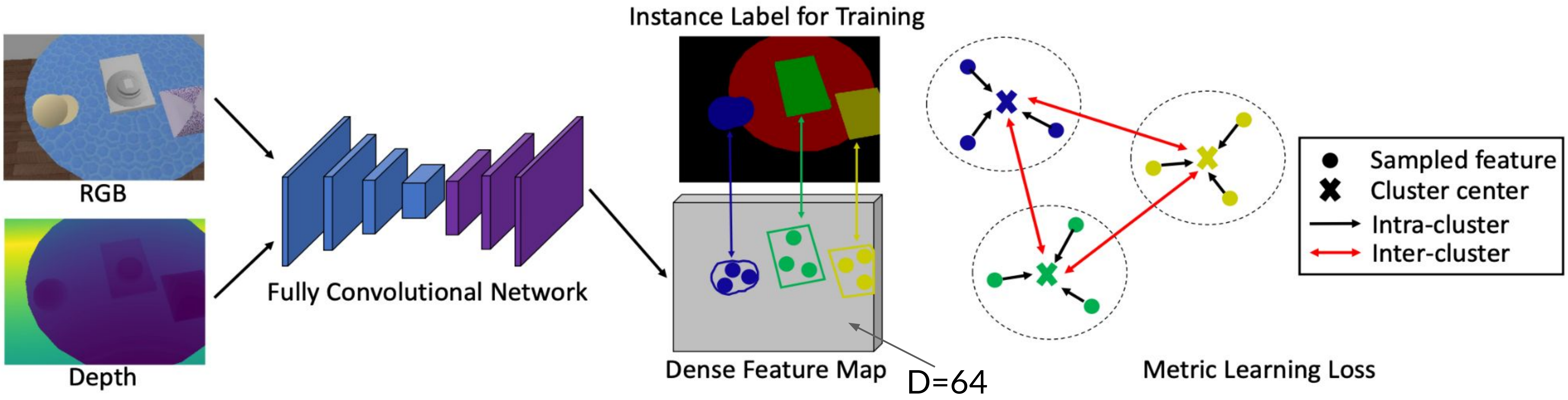
RGB-D Synthetic Data

Use synthetic dataset containing 40,000 scenes & 7 RGB-D images per scene



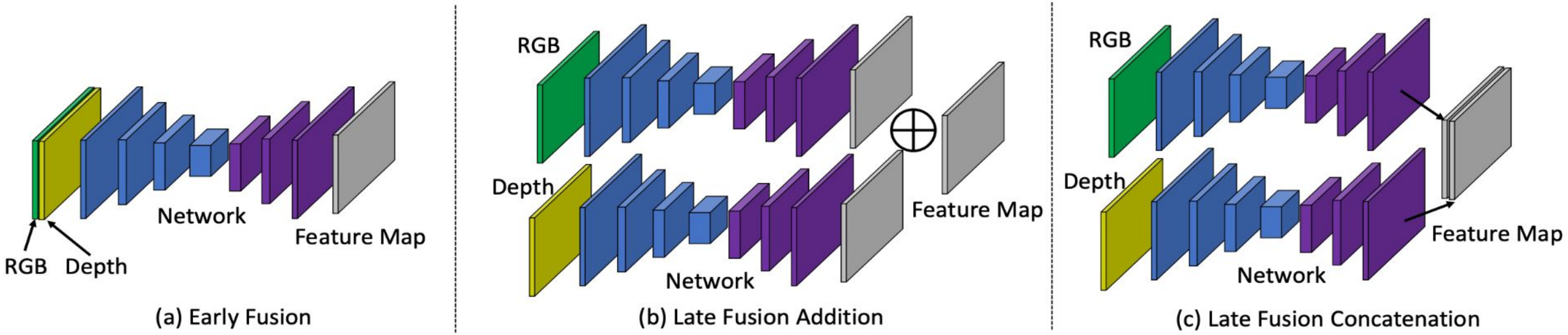
RGB Depth Instance Label

Learning Feature Embeddings



High-Level Pipeline Diagram

Learning Feature Embeddings



Three different ways of fusing RGB and depth data to compute embeddings

Metric Learning Loss Function

For each object in the image, N pixels are sampled to compute the loss (N = 1000).

$$\ell_{\text{intra}} = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{N} \frac{\mathbb{1}\left\{d(\mu^k, \mathbf{x}_i^k) - \alpha \geq 0\right\} \, d^2(\mu^k, \mathbf{x}_i^k)}{\sum_{i=1}^{N} \mathbb{1}\left\{d(\mu^k, \mathbf{x}_i^k) - \alpha \geq 0\right\}},$$

$$\ell_{\text{inter}} = \frac{2}{K(K-1)} \sum_{k<k'} \left[\delta - d(\mu^k, \mu^{k'})\right]_+^2$$

ℓ_{intra} – Intra-Cluster Loss Function

Pushes feature embeddings of pixels on the same object close to the cluster center H

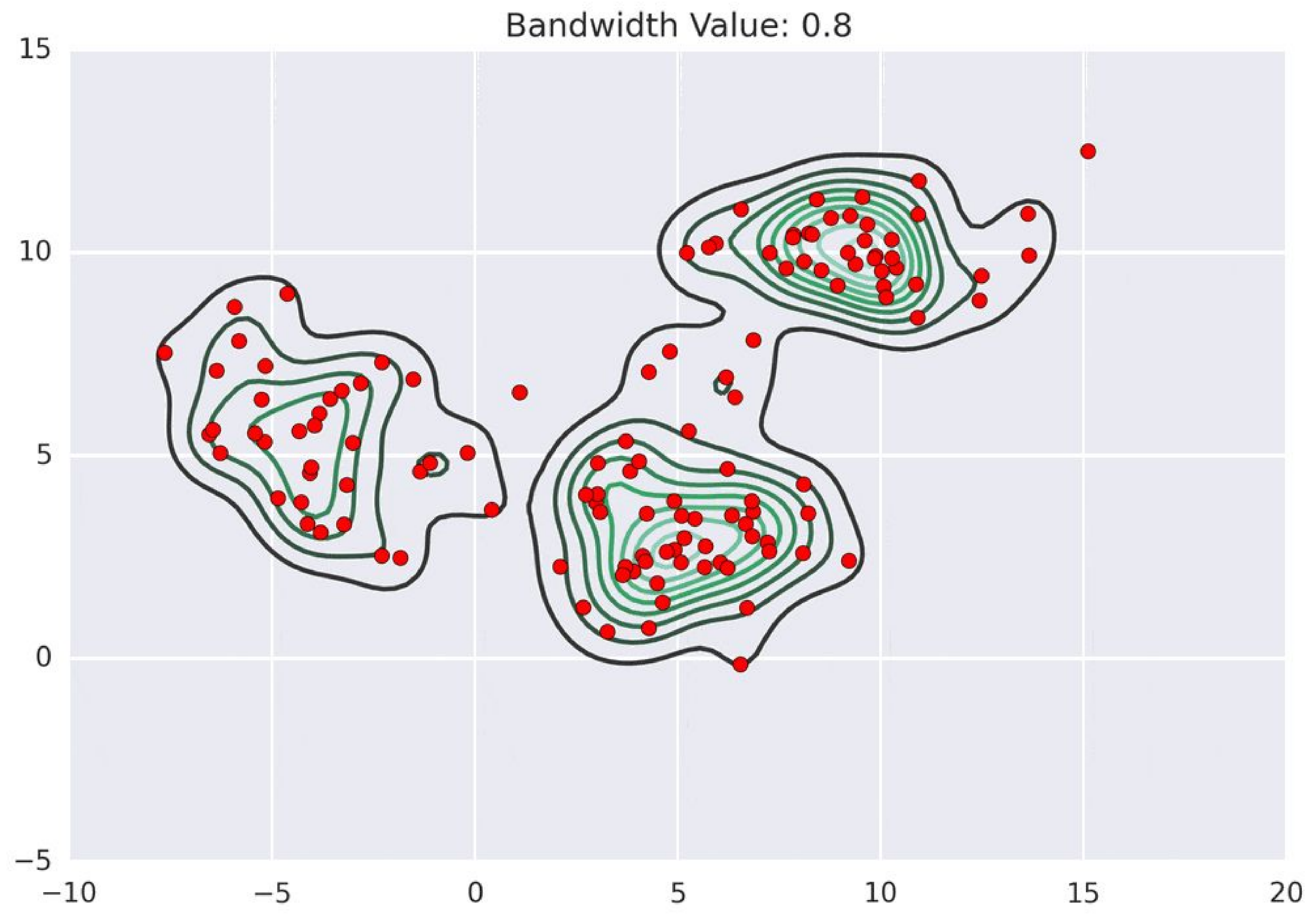
ℓ_{inter} – Inter-Cluster Loss Function

Pushes the different cluster centers away from each other in embedding space

$$\mathcal{L} = \lambda_{intra}\ell_{intra} + \lambda_{inter}\ell_{inter}$$

Separating Instances - Mean Shift Clustering

- Mean shift algorithm to cluster pixels
  - seeks local maxima of the distribution
- Mean shift exploits the density of the points to generate a reasonable number of clusters.



2 Stage Clustering Process

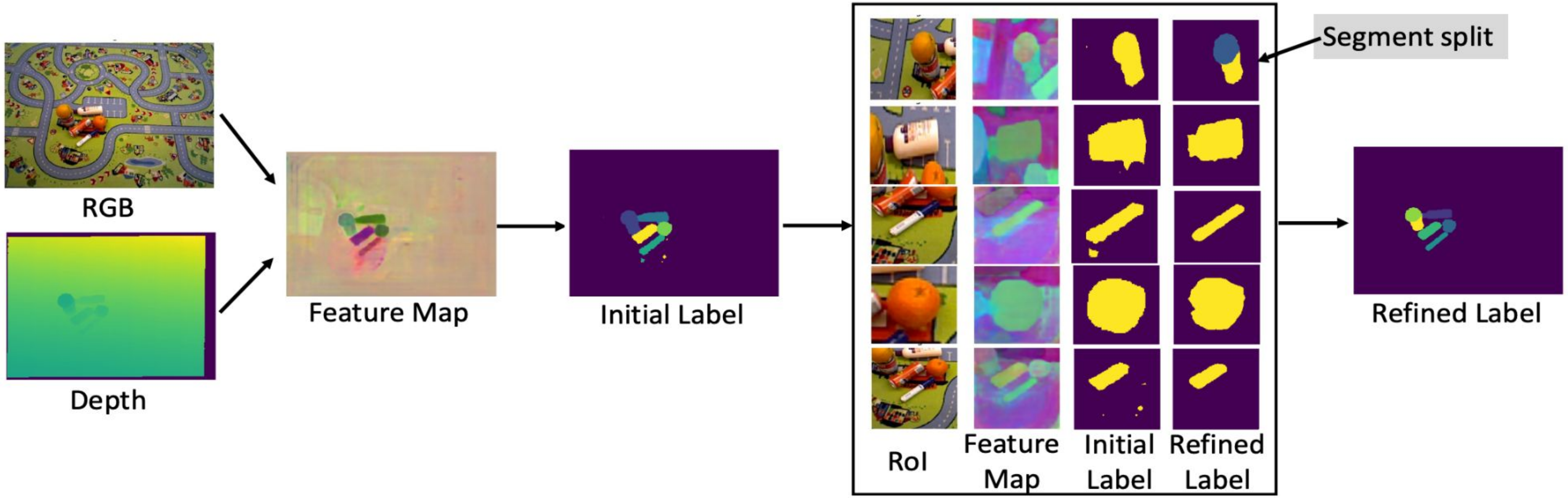
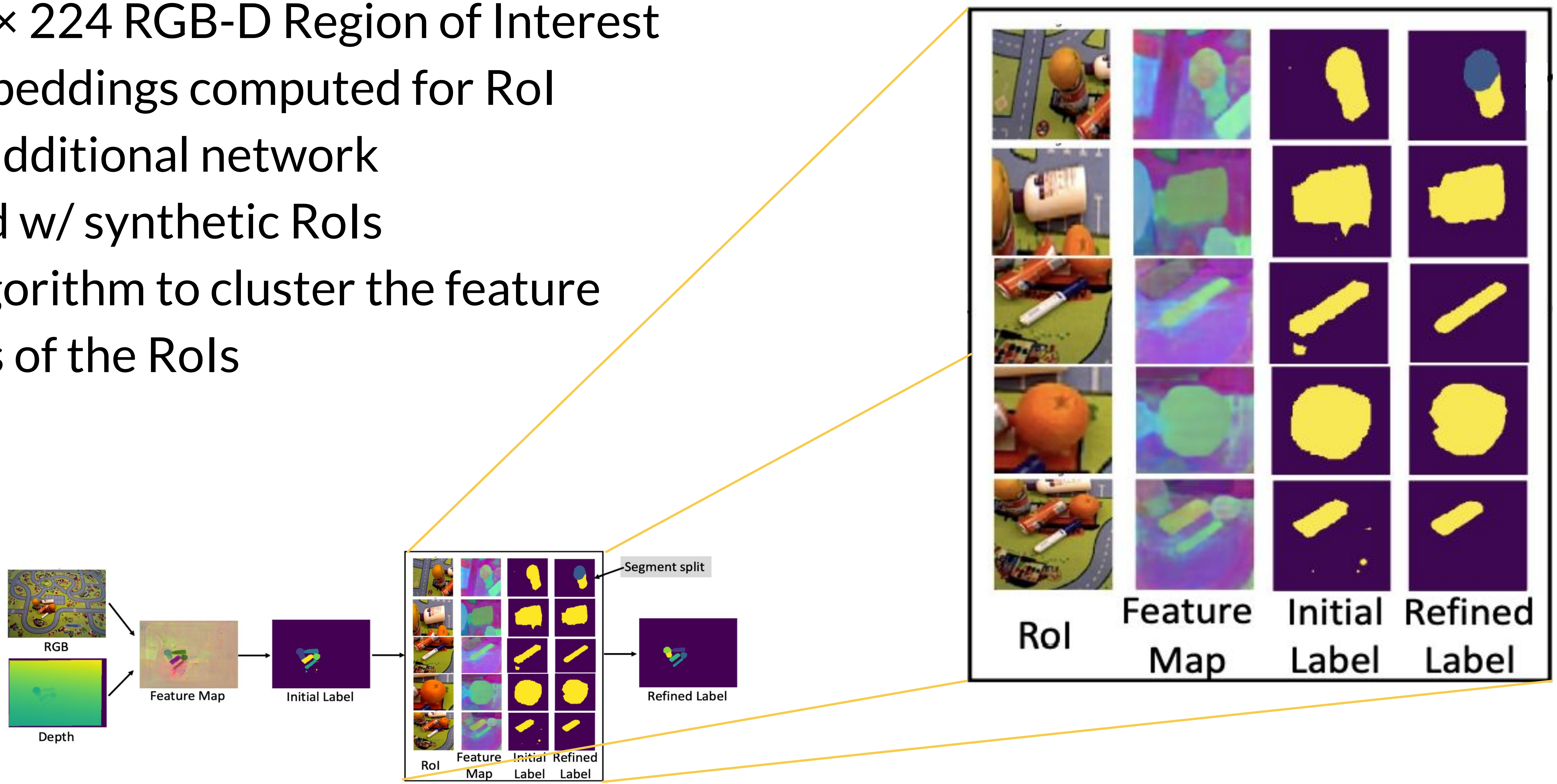


Figure 4: The two-stage clustering process in our method. The first stage clusters feature embeddings of all the image pixels. The second stage refines the segment for each RoI by clustering feature embeddings of the RoI.

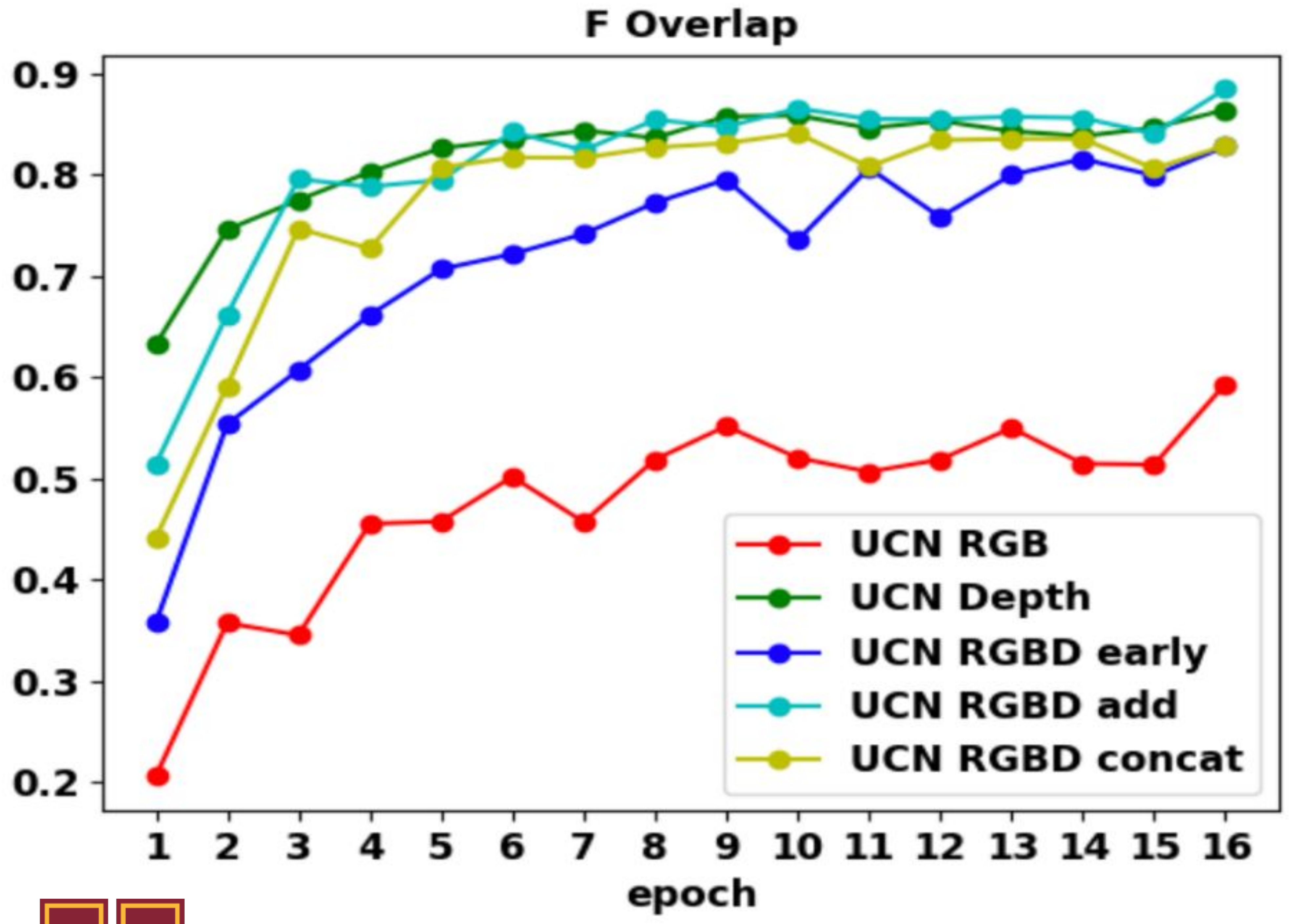
Zoom-In Cluster Refinement

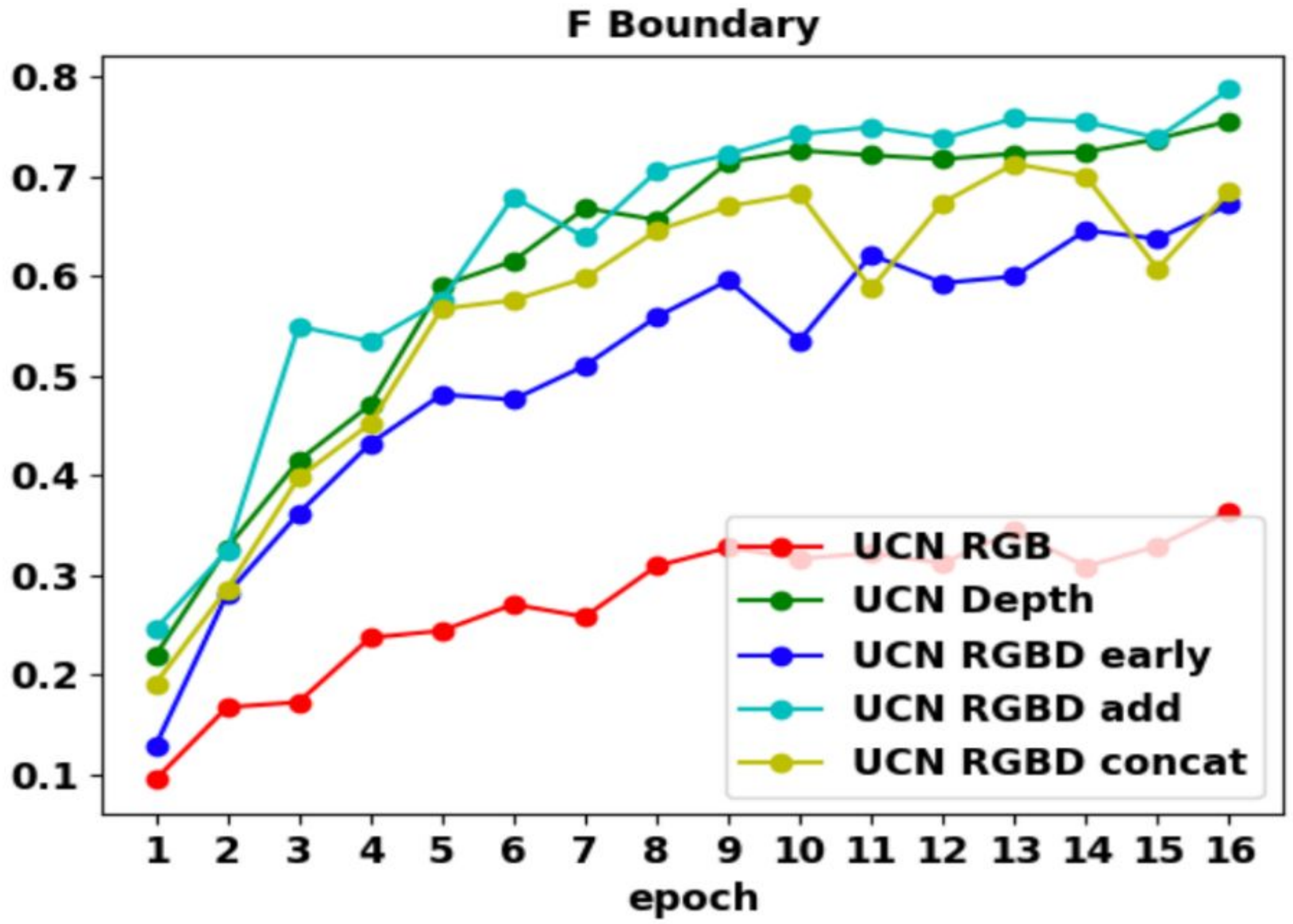
- Crop a 224 × 224 RGB-D Region of Interest
- Feature embeddings computed for RoI
  - using additional network
  - trained w/ synthetic RoIs
- vMF-MS algorithm to cluster the feature embeddings of the RoIs



Evaluation and Key Results

Ablation studies show that adding the feature vectors of two separate models (RGB & Depth) is most effective





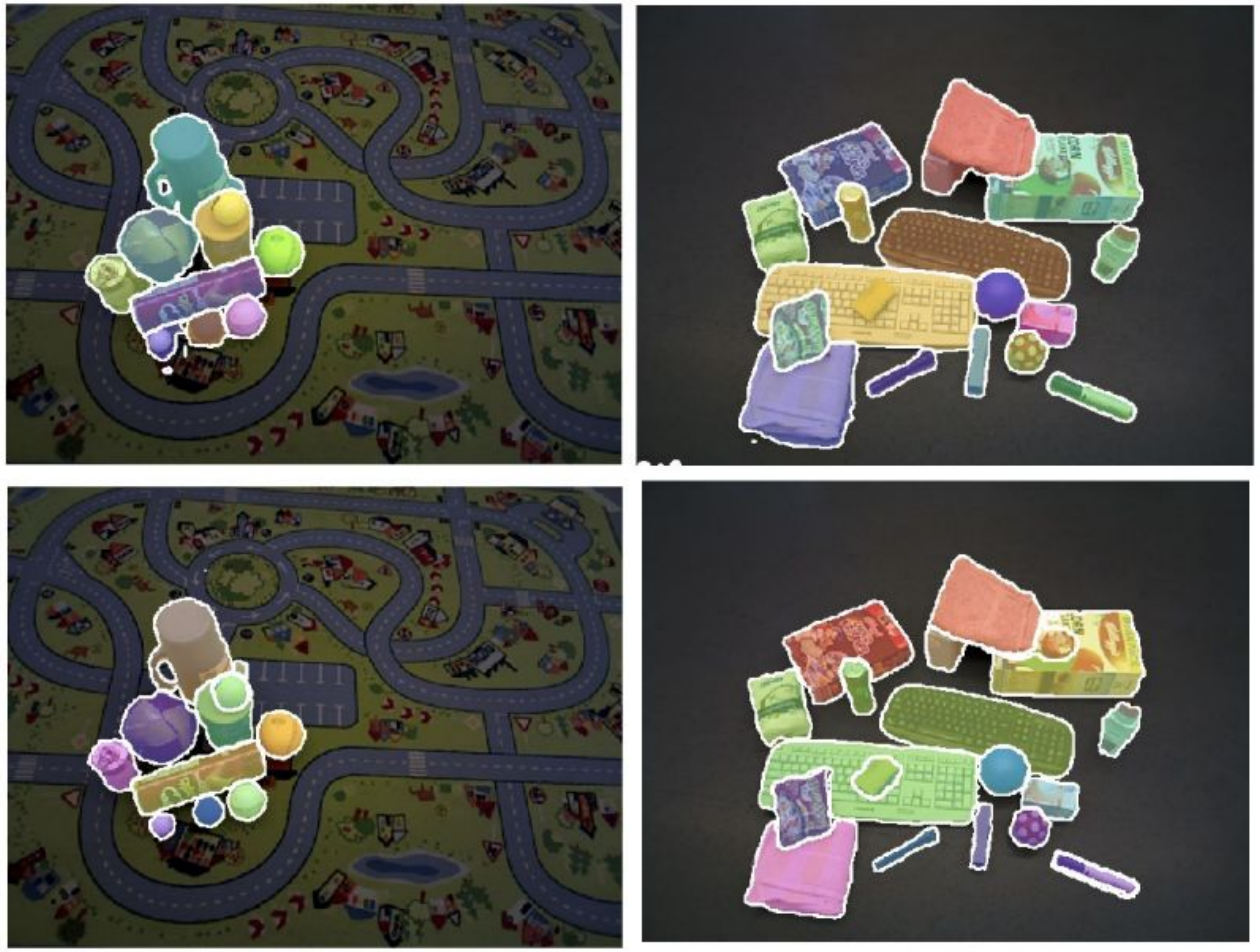
Evaluation and Key Results

Zoom-in refinement for 2-stage cluster algorithm **significantly** improves F-score.

	Overlap			Boundary		
	Precision	Recall	F-Score	Precision	Recall	F-score
RGBD add	86	92.3	88.5	80.4	78.3	78.8
RGBD add + Zoom-in	**91.6**	**92.5**	**91.6**	**86.5**	**97.1**	**86.1**

Initial Label

Refined Label



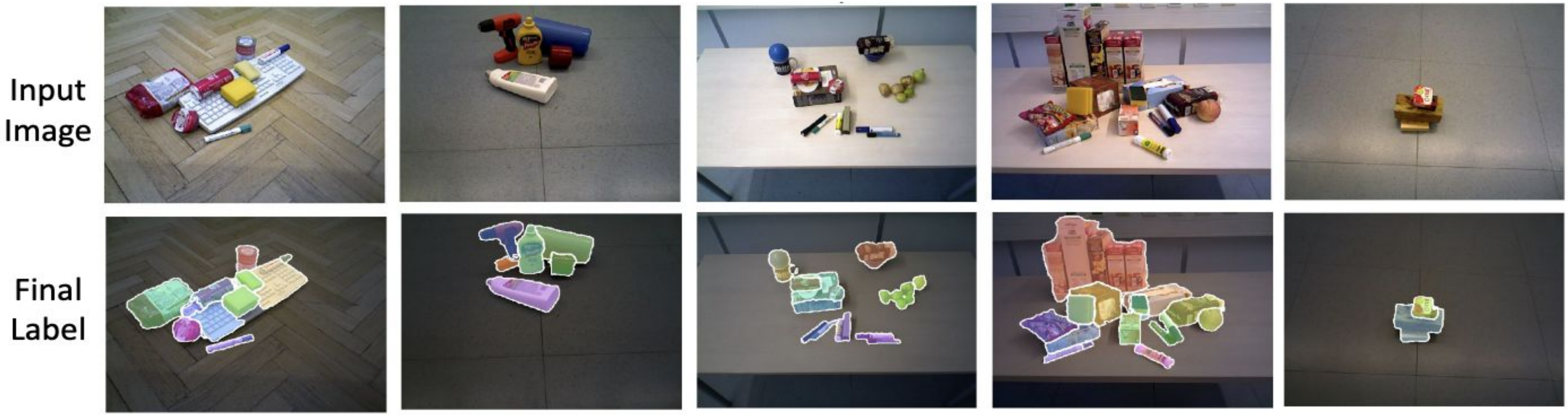
Evaluation and Key Results

	Overlap			Boundary		
	Precision	Recall	F-Score	Precision	Recall	F-score
MRCNN Depth	85.3	85.6	84.7	**83.2**	76.6	78.8
UOIS-Net-2D	**88.3**	78.9	81.7	82	65.9	71.4
UOIS-Net-3D	86.5	86.6	86.4	80	73.4	76.2
UCN (Ours)	87.4	**88.7**	**87.8**	82.2	**83.3**	**82.3**

UCN when compared with other SOTA neural networks.
The F score is significantly higher in both cases

Need for Future Work

The proposed method still suffers when instances of unseen objects are grouped closely together



Examples of segmentation failure

Questions?

Andrew Scheffer, Ashwin Saxena

Next Time: Point Cloud Processing

- Seminar 1: RGB-D Architectures
  1. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes, Xiang et al., 2018
  2. A Unified Framework for Multi-View Multi-Class Object Pose Estimation, Li et al., 2018
  3. PVN3D: A Deep Point-Wise 3D Keypoints Voting Network for 6DoF Pose Estimation, He et al., 2020
  4. Learning RGB-D Feature Embeddings for Unseen Object Instance Segmentation, Li et al., 2021

- Seminar 2: Point Cloud Processing
  1. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, Qi et al., 2017
  2. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, Qi et al., 2017
  3. PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation, Xu et al., 2018
  4. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion, Wang et al., 2019

DeepRob

Seminar 1
3D Perception: RGB-D Architectures
University of Michigan and University of Minnesota