PIEs: Pose Invariant Embeddings

Chih-Hui (John) Ho
Advisor: Professor Nuno Vasconcelos

Outline

- Introduction
- Motivation
- Proposed architecture
- Experiment
- Conclusion

Pose Invariant Embeddings

Introduction

- Human can tell what the object is regardless of its viewpoint or pose

Warplane

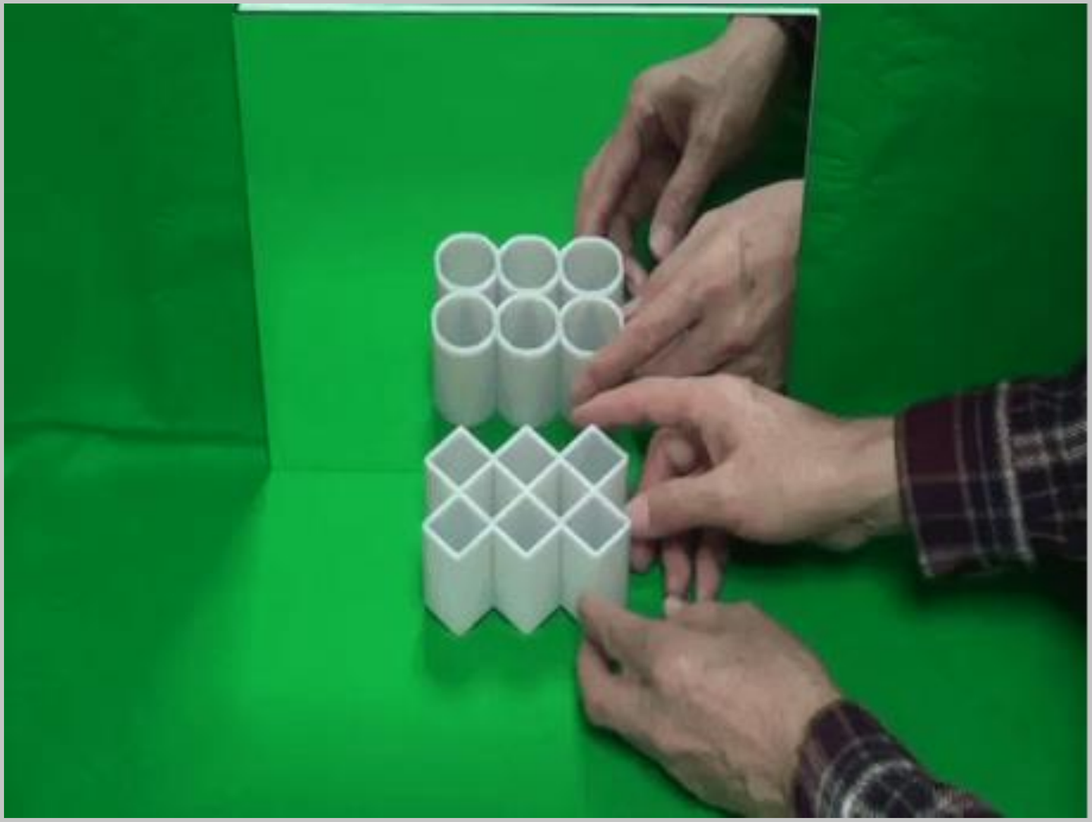
Introduction

- Human can tell what the object is regardless of its viewpoint or pose

Warplane

Introduction

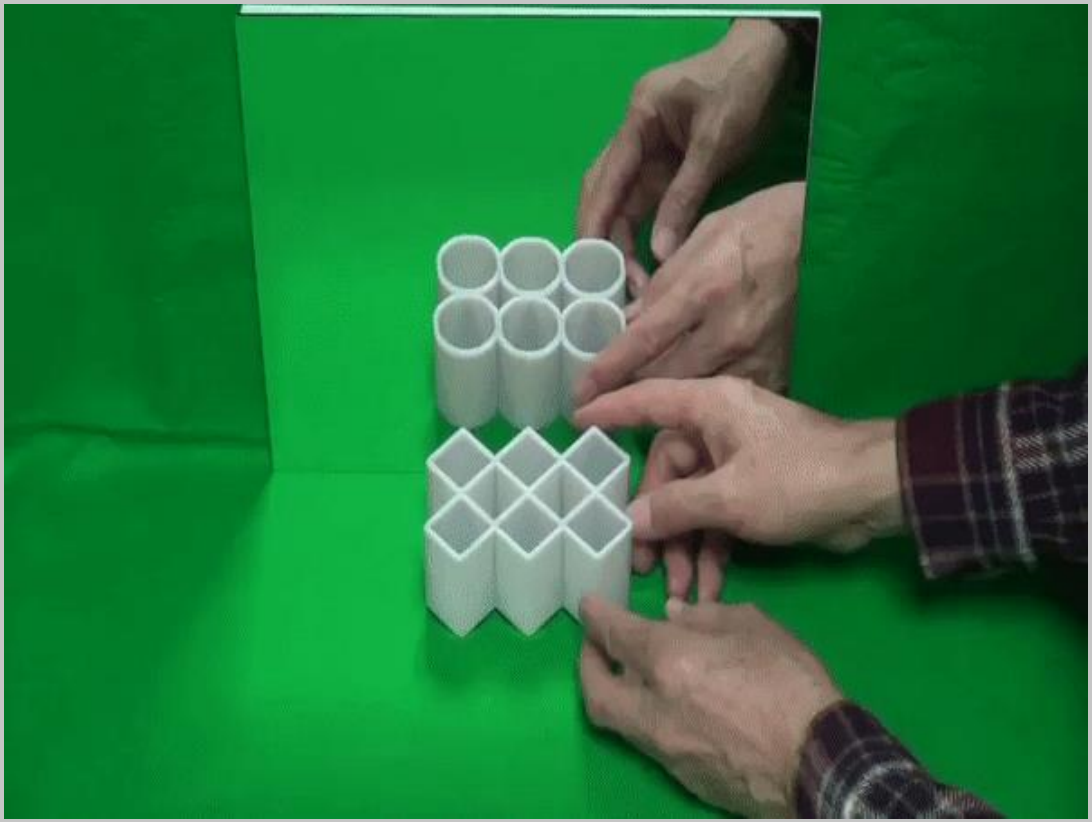
- Human can tell what the object is regardless of its viewpoint or pose
- Pose illusion for human



[Kokichi Sugihara: “Ambiguous Cylinder Illusion”]

Introduction

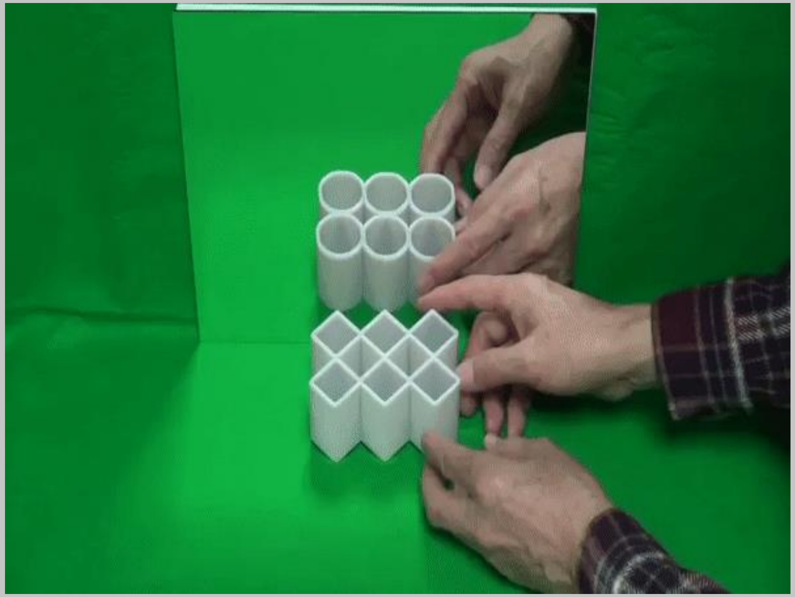
- Human can tell what the object is regardless of its viewpoint or pose
- Pose illusion for human



[Kokichi Sugihara: "Ambiguous Cylinder Illusion"]

Introduction

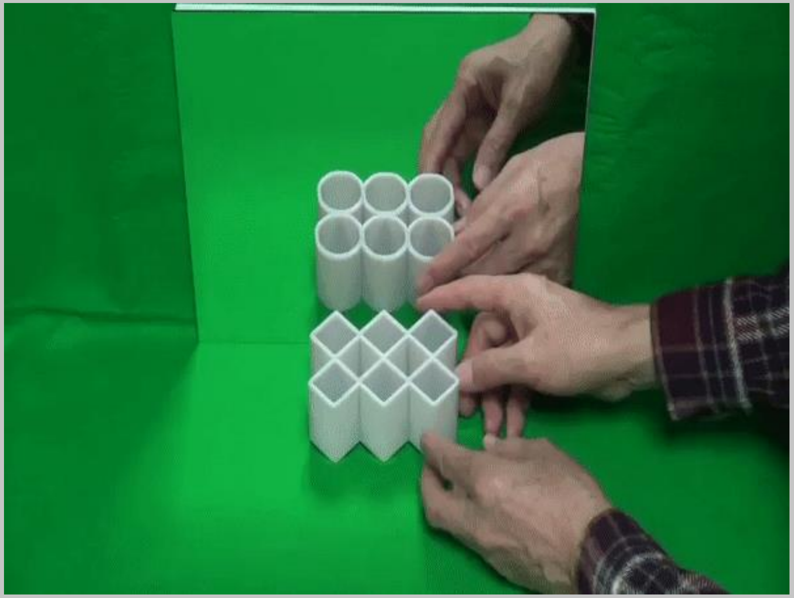
- Human can tell what the object is regardless of its viewpoint or pose
- Pose illusion for human
- Pose invariant recognition is a difficult task even for human on some cases



[Kokichi Sugihara: “Ambiguous Cylinder Illusion”]

Introduction

- What about classifier?
  - Learn features/embeddings invariant to pose transformations



[Kokichi Sugihara: “Ambiguous Cylinder Illusion”]

Pose Invariant **Embeddings**

Introduction

- Convolutional neural networks (CNN) has a huge impact on computer vision applications

Introduction

- Convolutional neural networks (CNN) has a huge impact on computer vision applications
- Some of the main tasks are
  - Classification
  - Retrieval

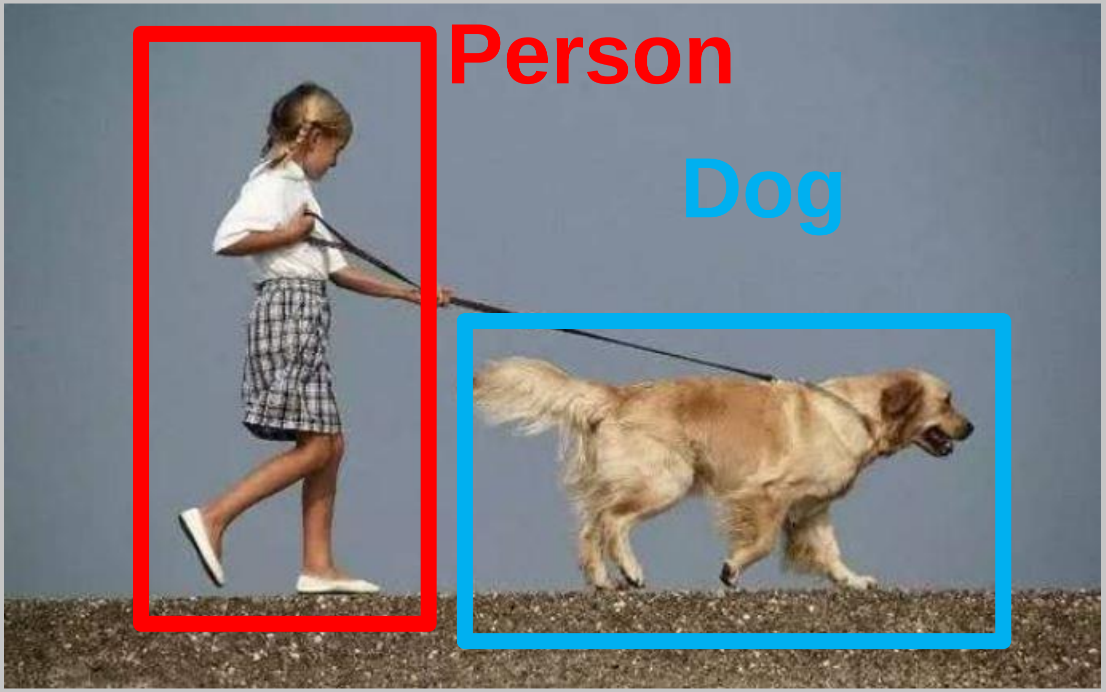
Introduction

- Convolutional neural networks (CNN) has a huge impact on computer vision applications
- Some of the main tasks are
  - Classification
  - Retrieval

Dog

Introduction

- Convolutional neural networks (CNN) has a huge impact on computer vision applications
- Some of the main tasks are
  - Classification
  - Retrieval

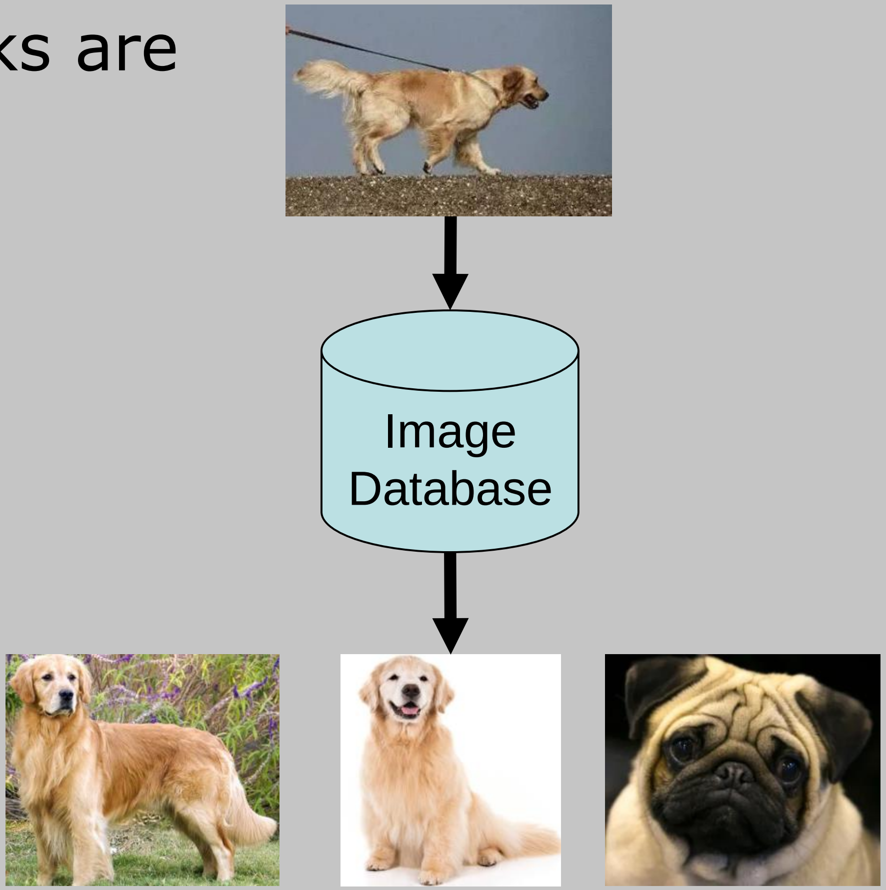


Introduction

- Convolutional neural networks (CNN) has a huge impact on computer vision applications
- Some of the main tasks are
  - Classification
  - Retrieval

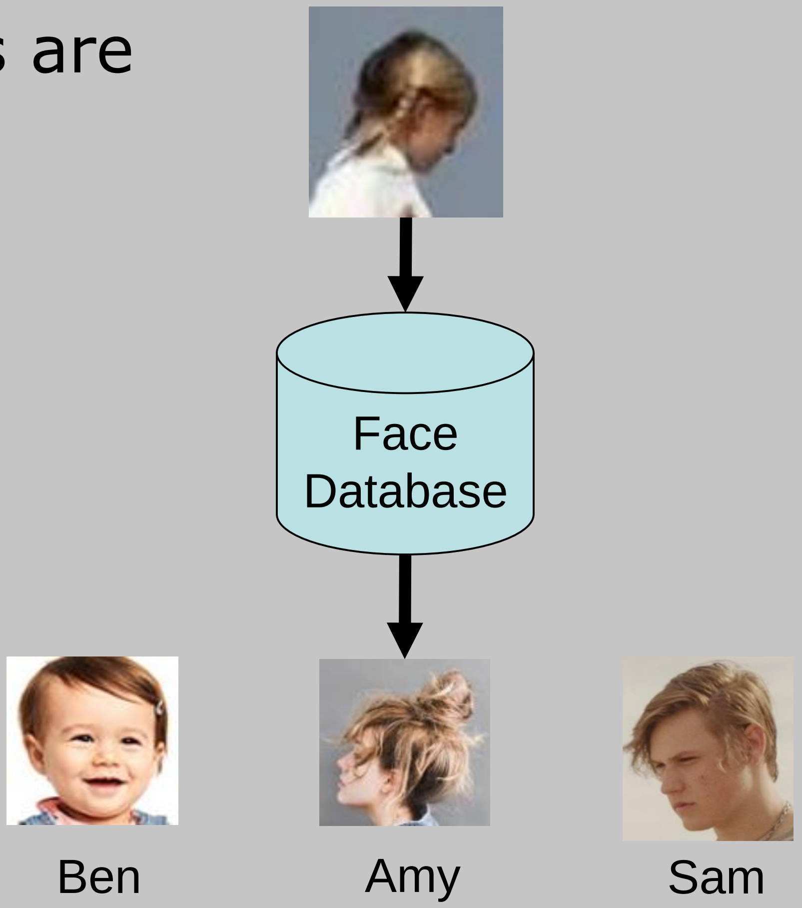
Introduction

- Convolutional neural networks (CNN) has a huge impact on computer vision applications
- Some of the main tasks are
  - Classification
  - Retrieval



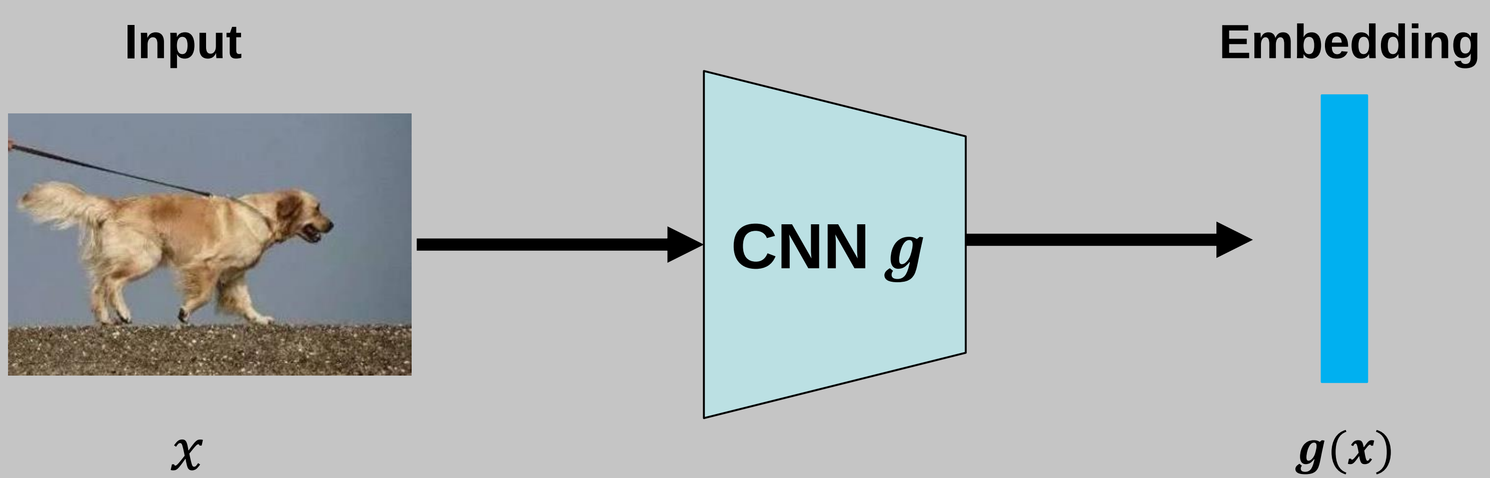
Introduction

- Convolutional neural networks (CNN) has a huge impact on computer vision applications
- Some of the main tasks are
  - Classification
  - Retrieval



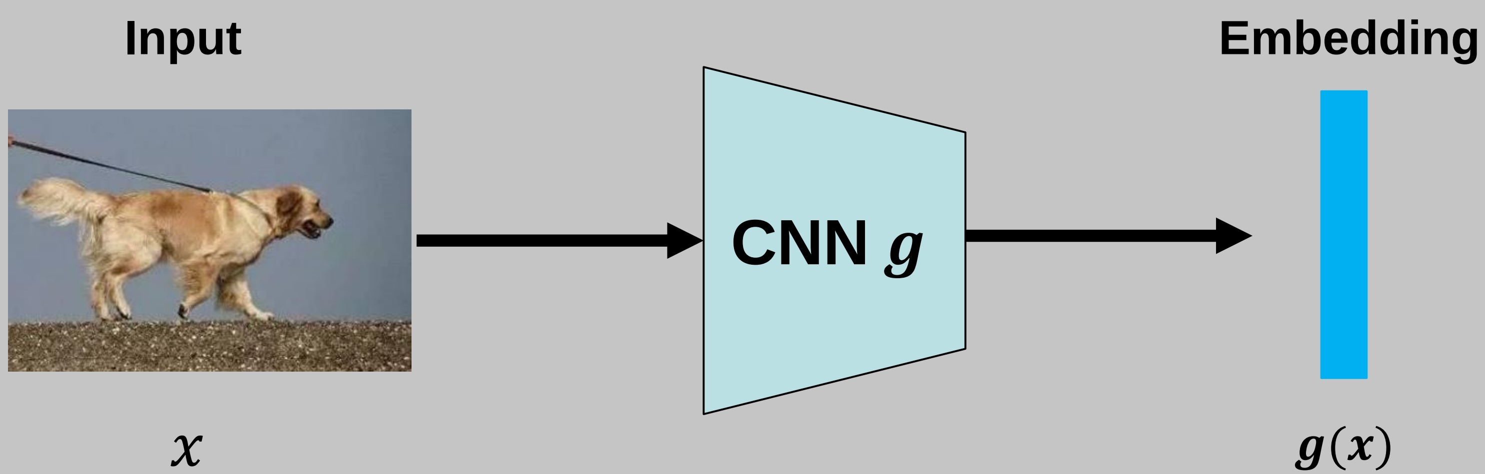
Introduction

- Classification and retrieval are related
  - Learn an embedding $g(x)$ from the input x using CNN g



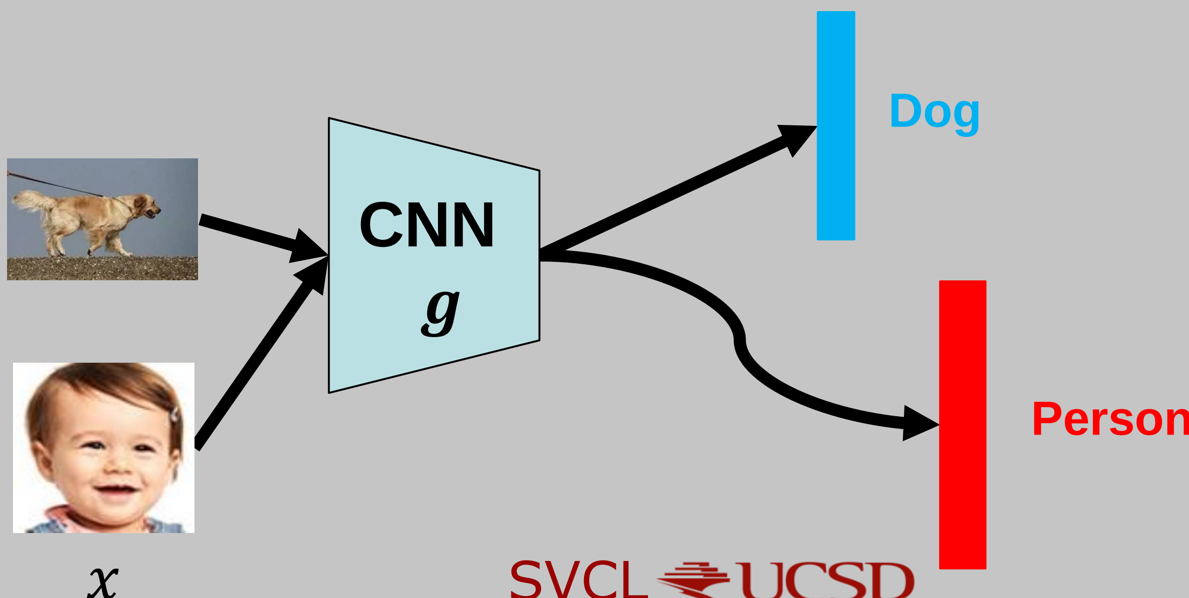
Introduction

- Classification and retrieval are related
  - Learn an embedding $g(x)$ from the input x using CNN g
- But different in terms of
  - their goals
  - their training approaches



Introduction

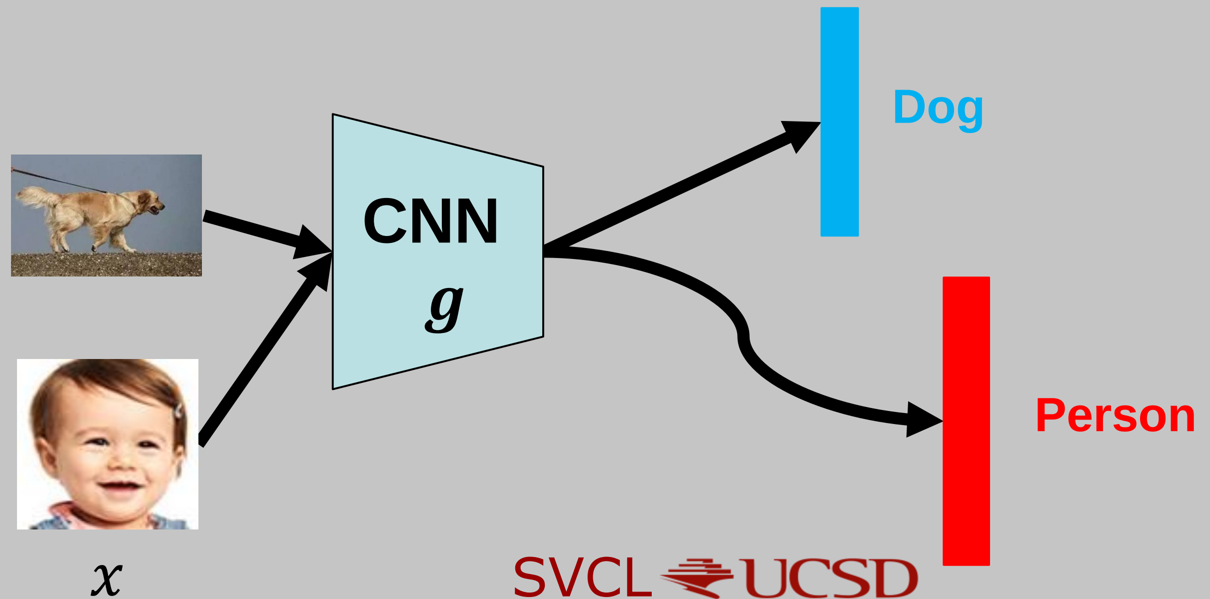
- Classification:
  - Learn discriminant embedding using feature extractor g



20

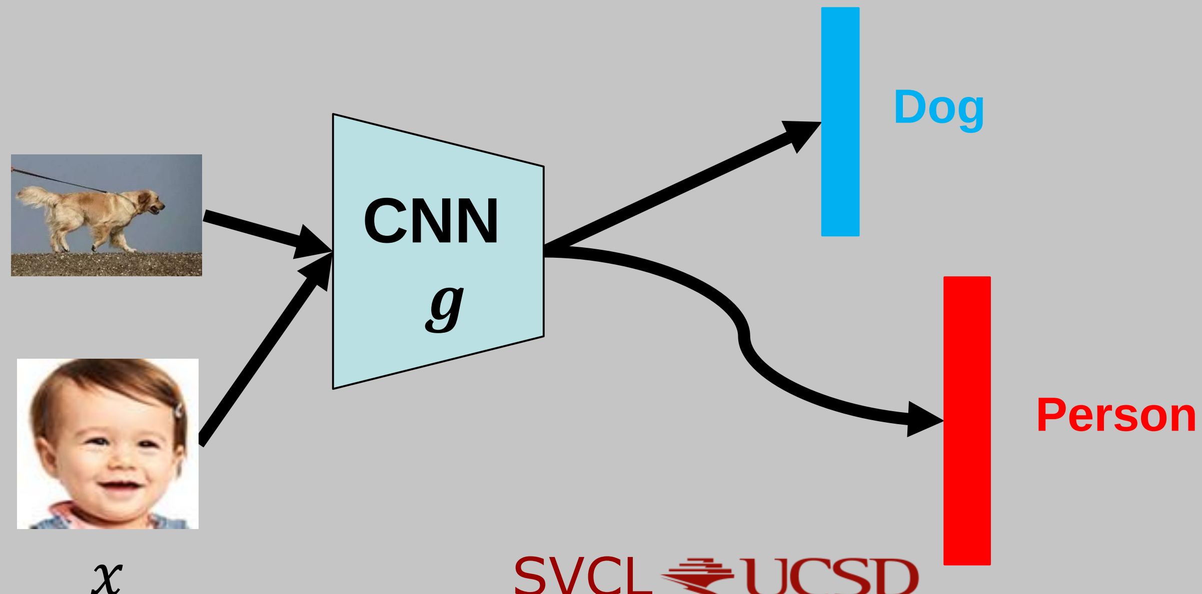
Introduction

- Classification:
  - Learn discriminant embedding using feature extractor g
  - Additional softmax layer W is trained on top of g



Introduction

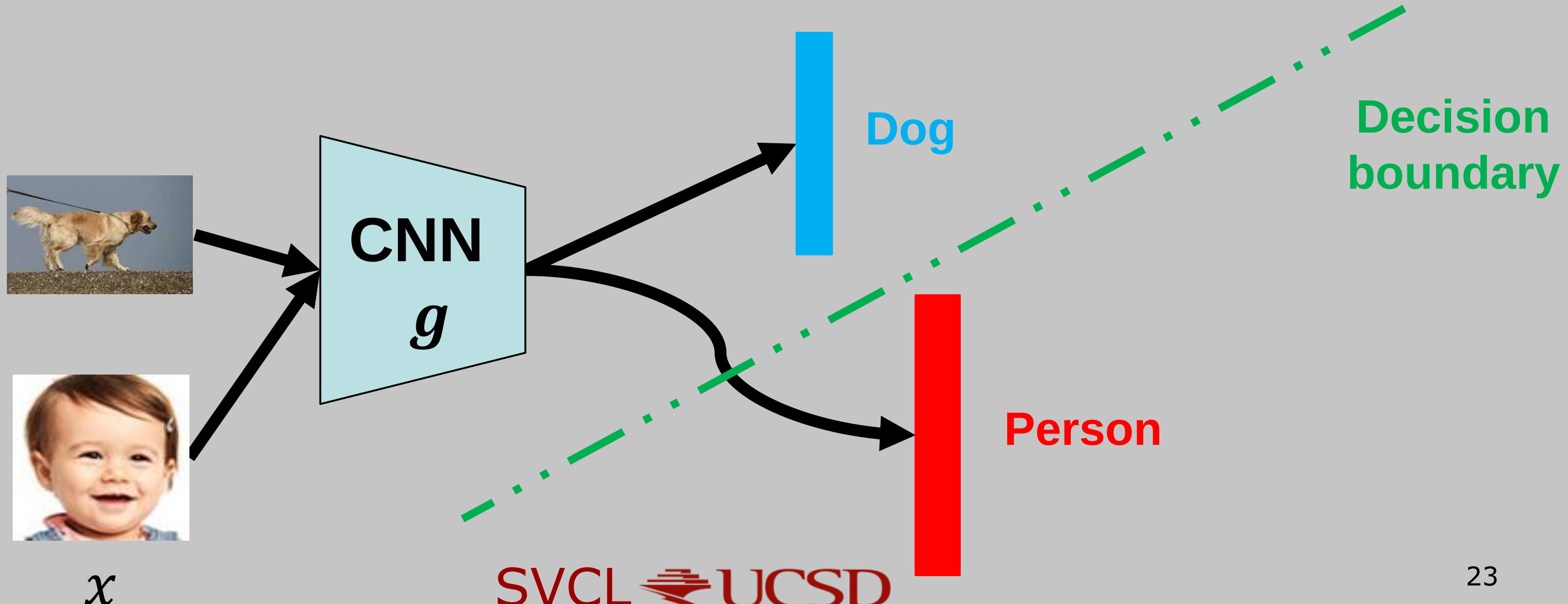
- Classification:
  - Learn discriminant embedding using feature extractor g
  - Additional softmax layer W is trained on top of g
  - Posterior probability $P_{Y|X}(y|x) = \frac{e^{w_y^T g(x)}}{\sum_{k=1}^{C} e^{w_k^T g(x)}}$



22

Introduction

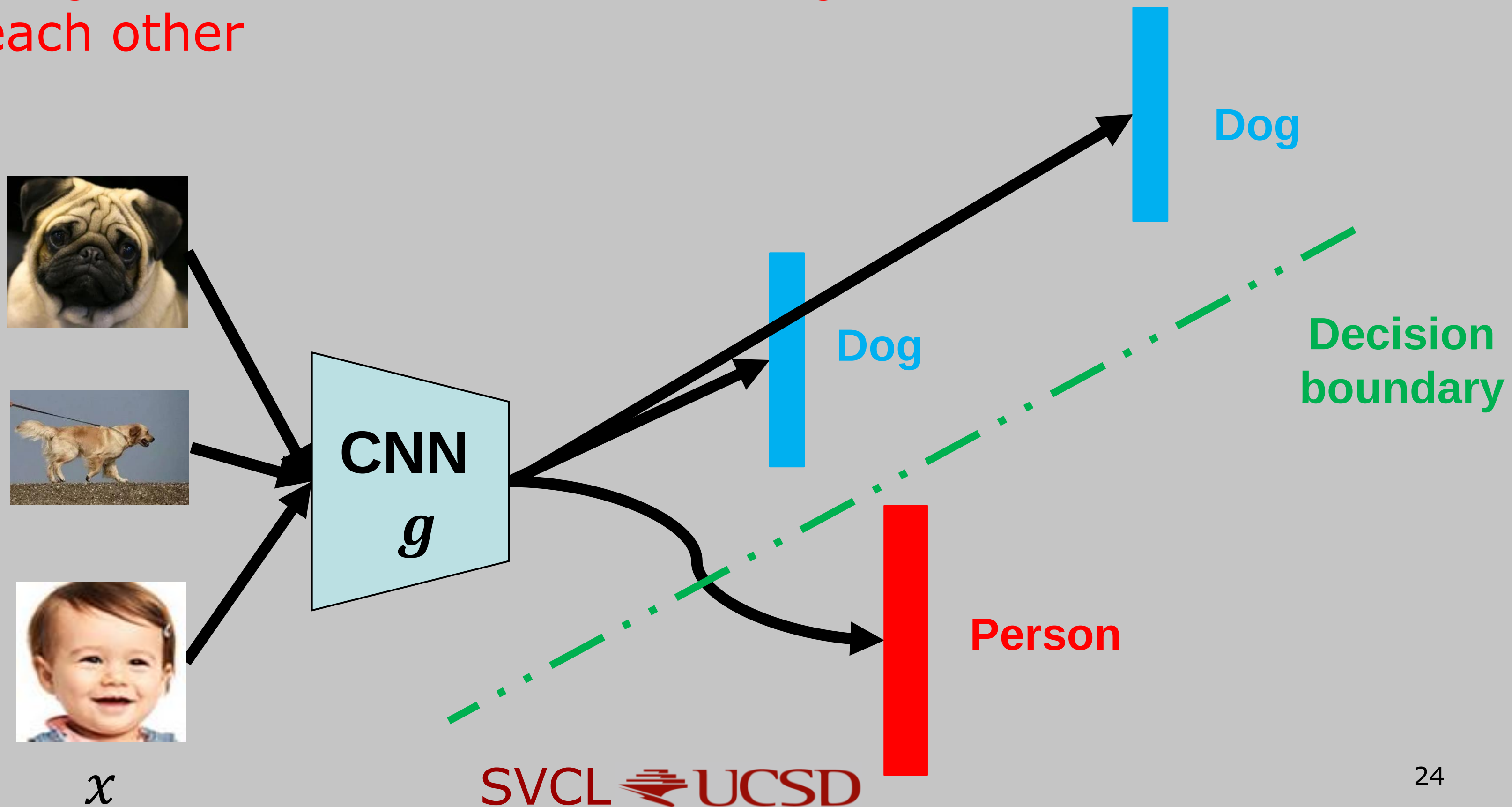
- Classification:
  - The learned embeddings from different classes are across decision boundary



SVCL UCSD

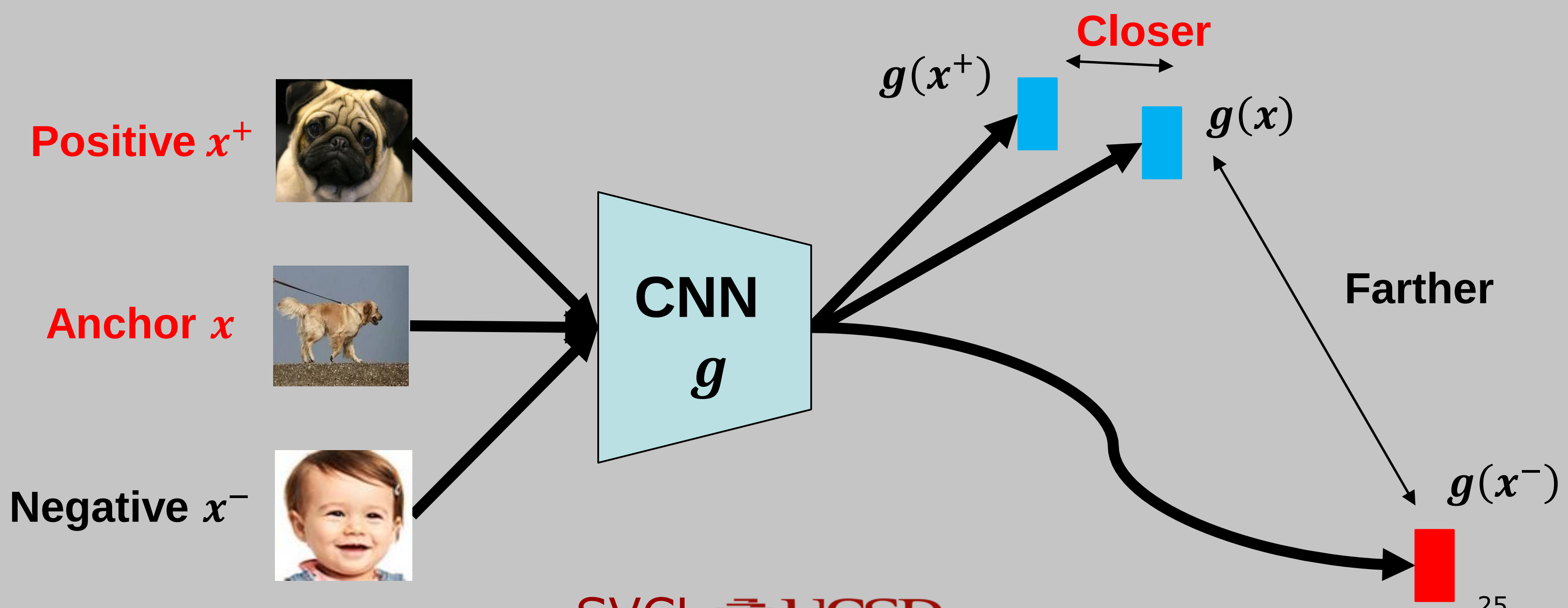
Introduction

- Classification:
  - The learned embeddings from different classes are across decision boundary
  - No guarantee that features belong same class are close to each other



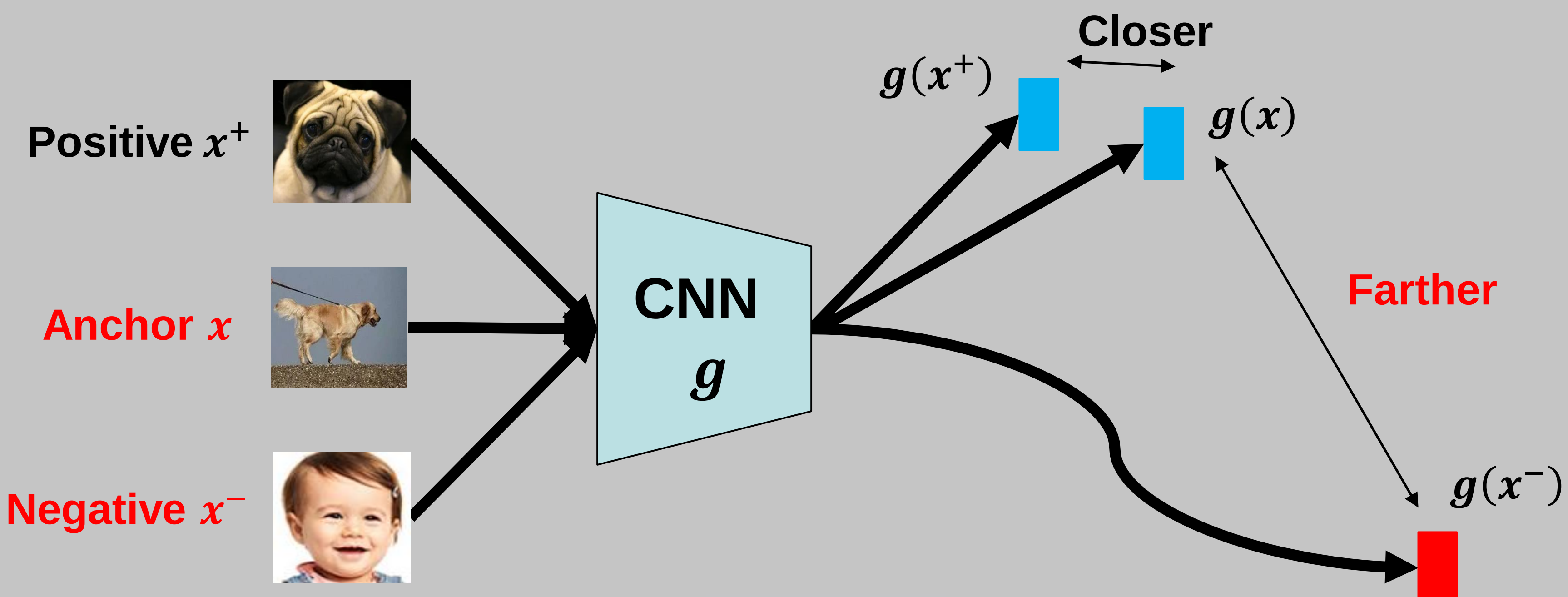
Introduction

- Metric learning for retrieval task:
  - Inputs from same class have closer distance



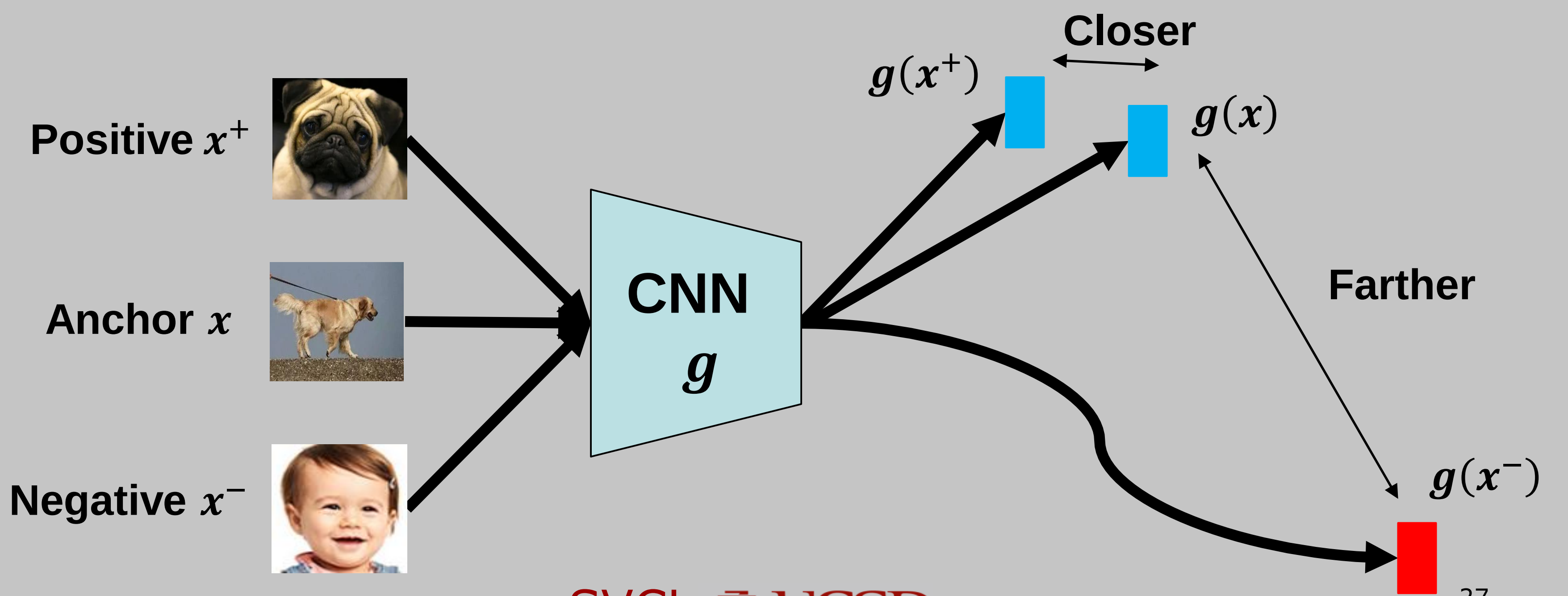
Introduction

- Metric learning for retrieval task:
  - Inputs from same class have closer distance
  - Inputs from different classes have farther distance



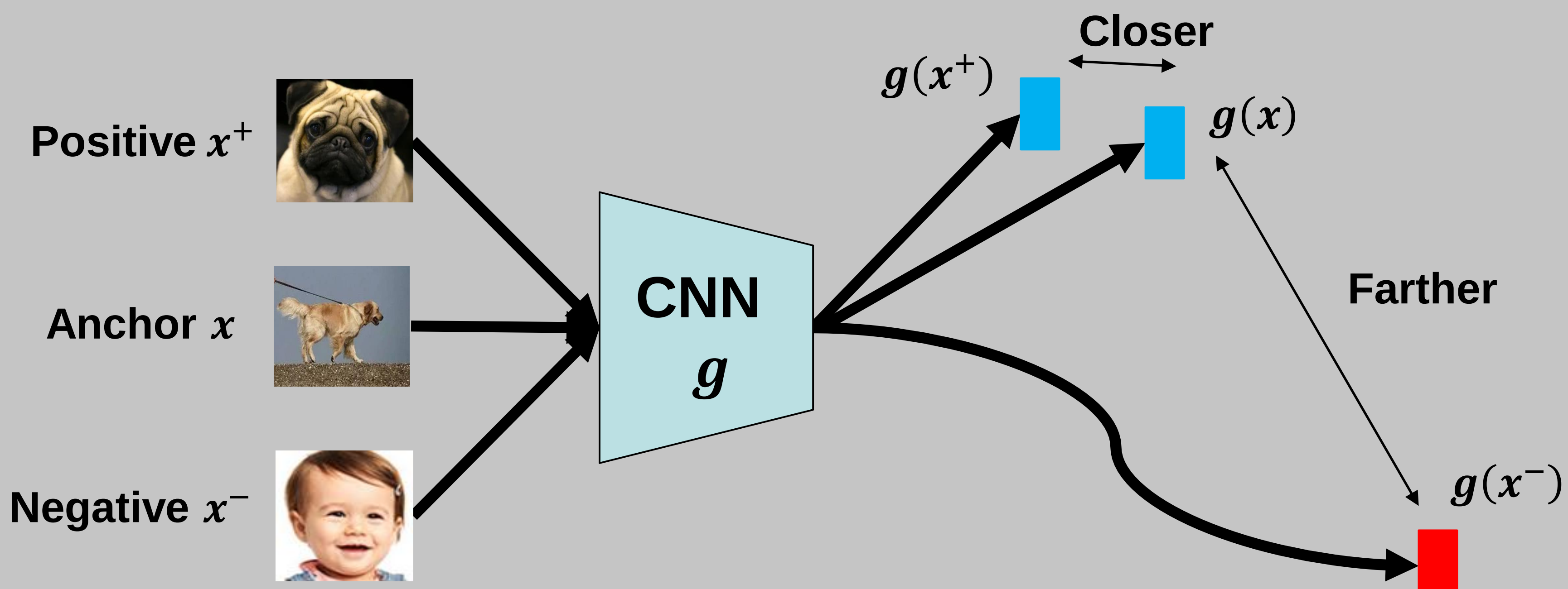
Introduction

- Metric learning for retrieval task:
  - Inputs from same class have closer distance
  - Inputs from different classes have farther distance
  - Train triplets (Positive, Anchor, Negative) with triplet loss



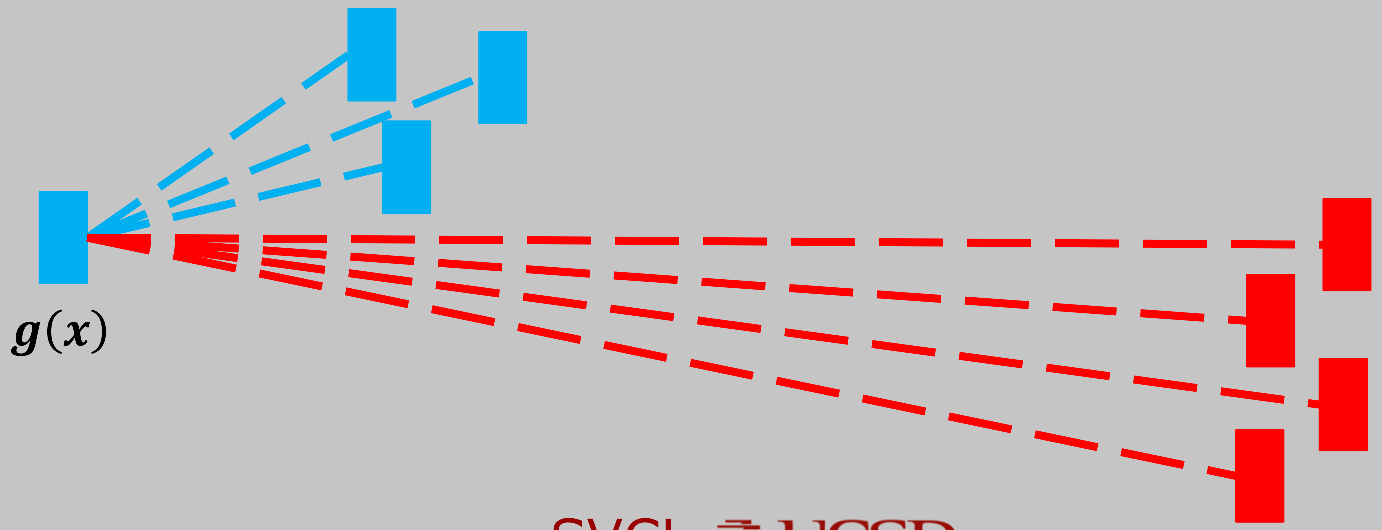
Introduction

- Metric learning for retrieval task:
  - Define $d(x, y)$ as the distance of 2 features x and y
  - Margin loss $\emptyset(\mathrm{v}) = \max(0, \mathrm{m} - \mathrm{v})$ with some margin m
  - Triplet loss $L(x, x^+, x^-) = \emptyset\Big(d\big(g(x), g(x^-)\big) - d\big(g(x), g(x^+)\big)\Big)$



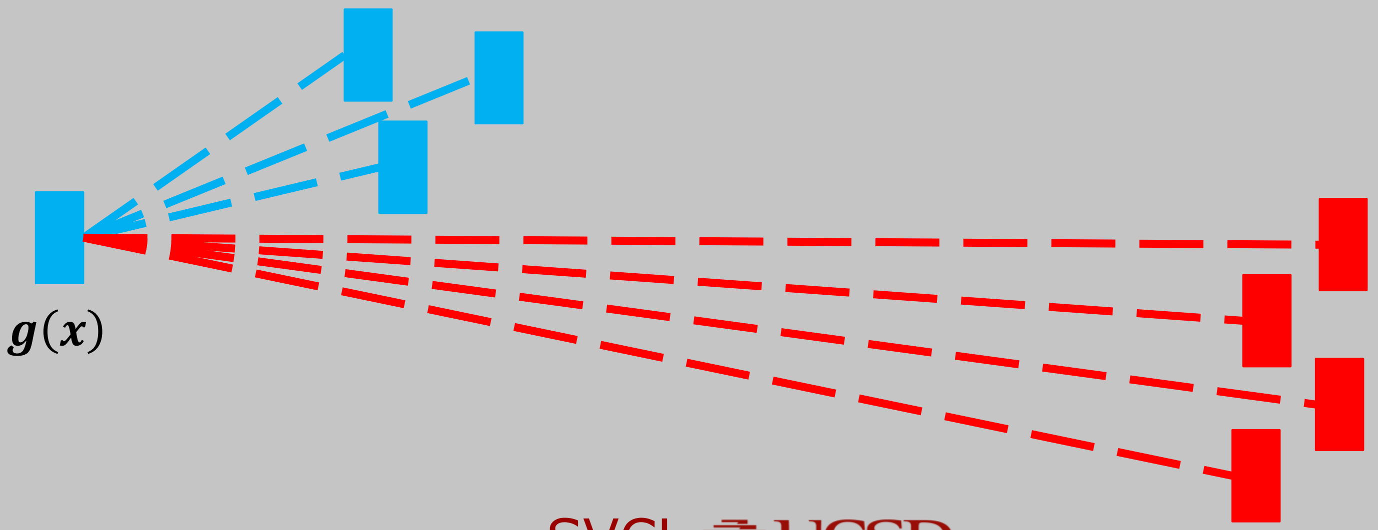
Introduction

- Metric learning for retrieval task :
  - If there are n images in the dataset → $O(n^3)$ triplets



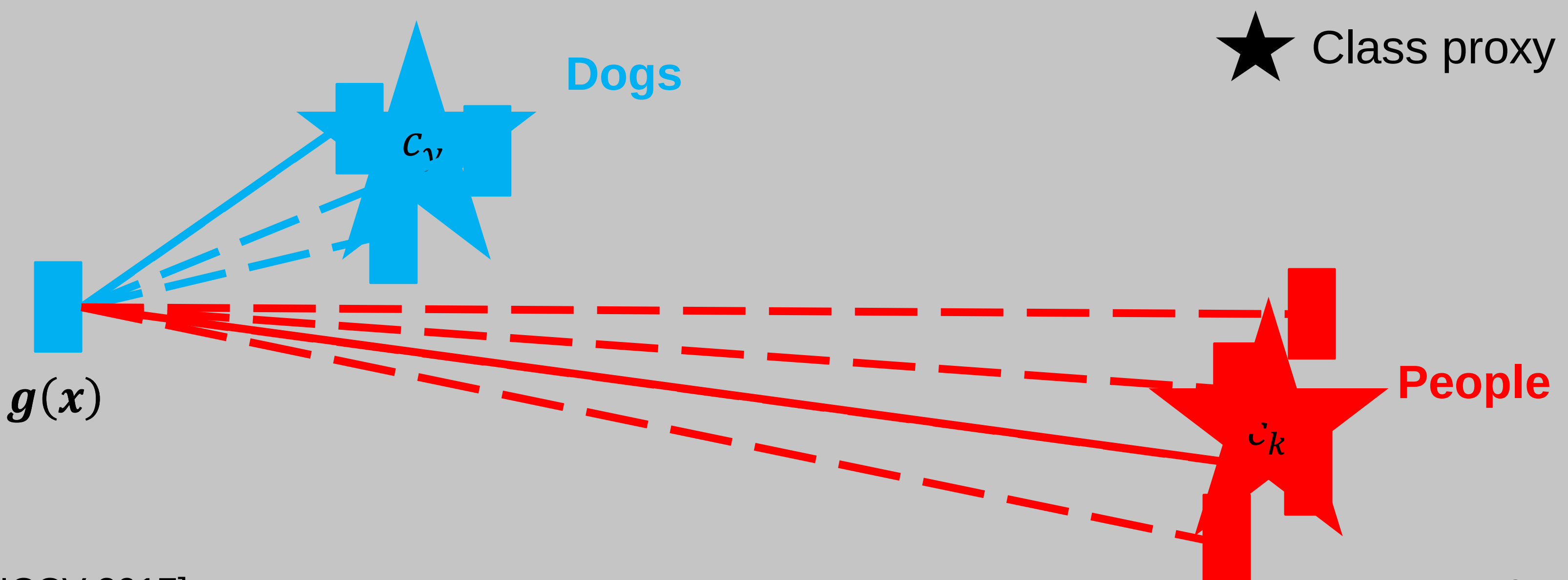
Introduction

- Metric learning for retrieval task :
  - If there are n images in the dataset → $O(n^3)$ triplets
  - Metric learning becomes a difficult problem as it is hard to converge



Introduction

- Metric learning for retrieval task :
  - Yair et al. proposed to introduce a proxy for each class
    - Proxy serves as a concise representation of a class
      - Star c_y represents the dog class



[Yair et al, ICCV 2017]

Introduction

- Metric learning for retrieval task :
  - Yair et al. proposed to introduce a proxy for each class
    - Proxy serves as a concise representation of a class
      - Star c_y represents the dog class
    - No more triplets during training
    - Faster convergence

[Yair et al, ICCV 2017]

Introduction

- Metric learning for retrieval task :
  - Minimize proxy loss $L(x, \boldsymbol{C}) = \frac{e^{-d(g(x), c_y)}}{\sum_{k \neq y} e^{-d(g(x), c_k)}}$

[Yair et al, ICCV 2017]

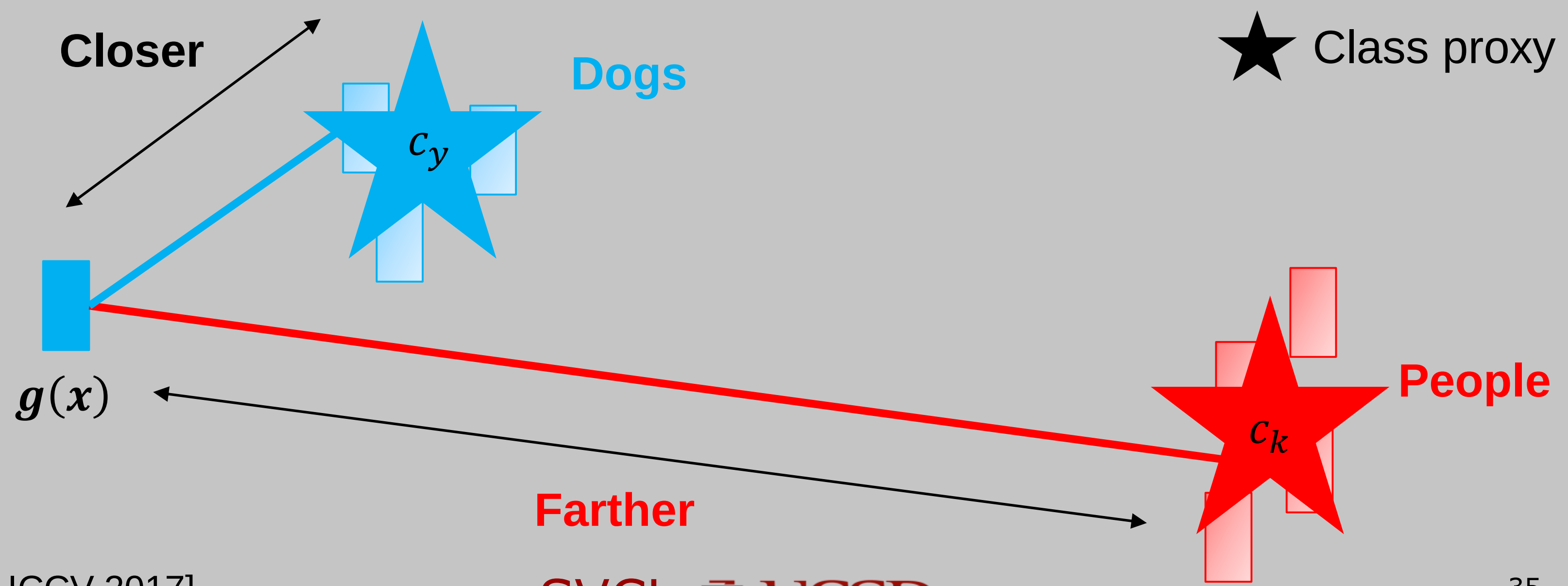
Introduction

- Metric learning for retrieval task :
  - Minimize proxy loss $L(x, \boldsymbol{C}) = \frac{e^{-d(g(x), c_y)}}{\sum_{k \neq y} e^{-d(g(x), c_k)}}$
    - Minimize distance of feature $g(x)$ to its associated class proxy c_y

[Yair et al, ICCV 2017]

Introduction

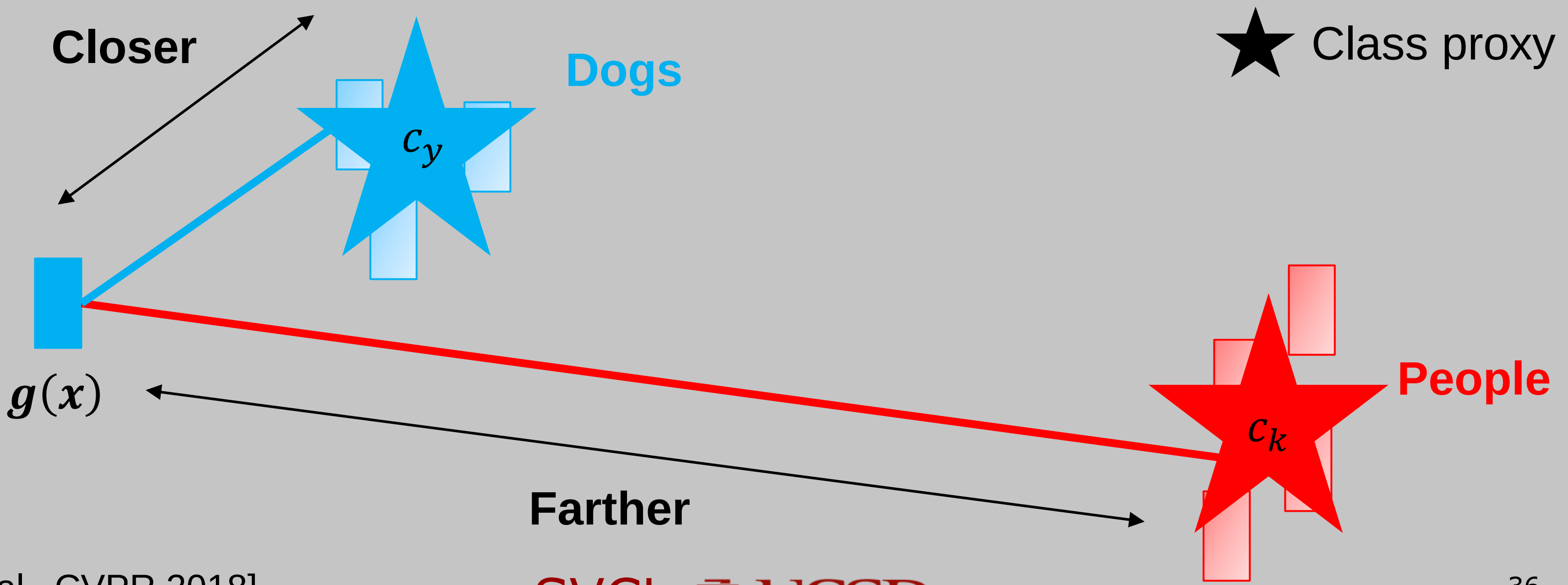
- Metric learning for retrieval task :
  - Minimize proxy loss $L(x, \boldsymbol{C}) = \frac{e^{-d(g(x), c_y)}}{\sum_{k \neq y} e^{-d(g(x), c_k)}}$
    - Minimize distance of feature $g(x)$ to its associated class proxy c_y
    - Maximize distance of feature $g(x)$ to other class proxies c_k



[Yair et al, ICCV 2017]

Introduction

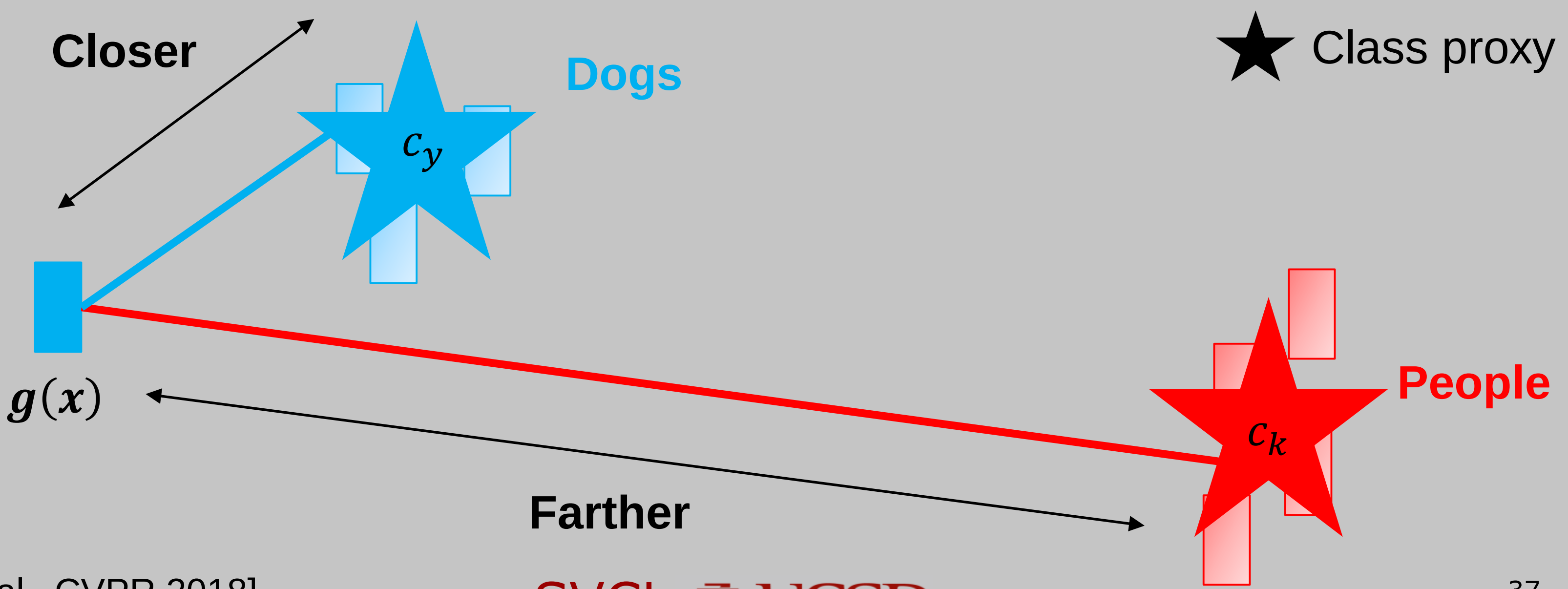
- Metric learning for retrieval task :
  - Xinwei et al. proposed triplet center loss by replacing triplets in triplet loss with proxies



[Xinwei et al., CVPR 2018]

Introduction

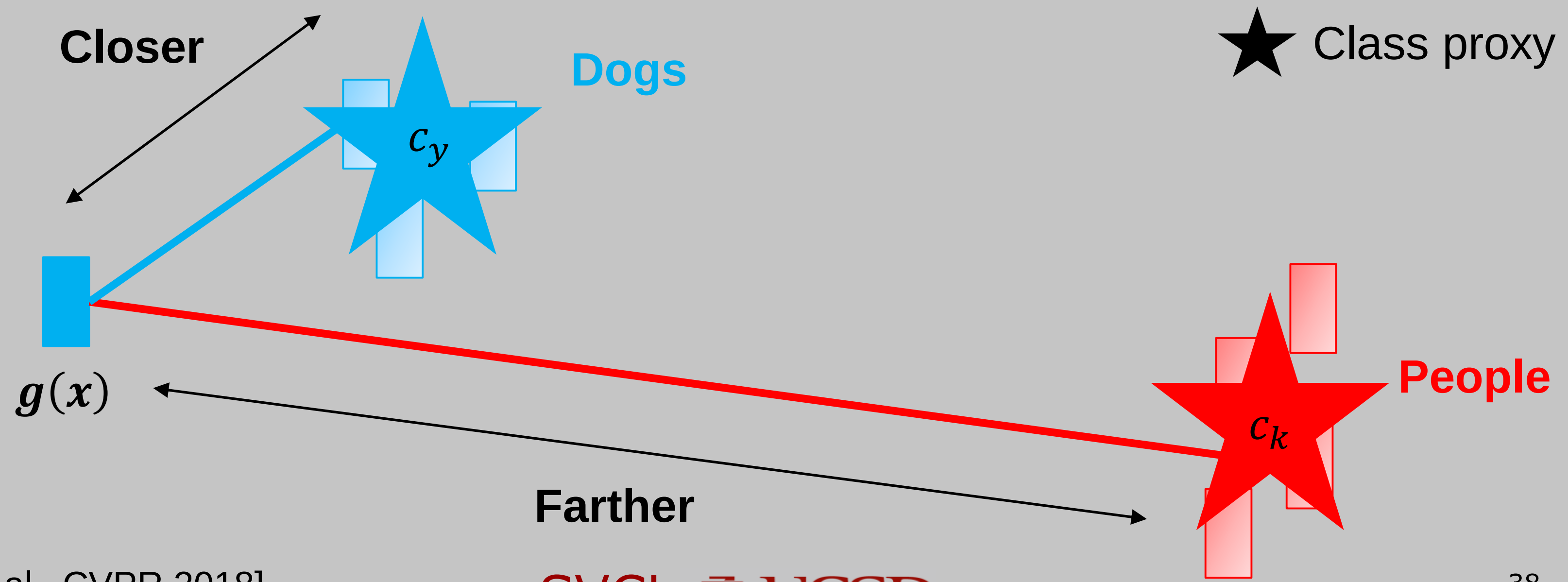
- Metric learning for retrieval task :
  - Margin loss $\emptyset(\mathrm{v}) = \max(0, \mathrm{m} - \mathrm{v})$ with some margin m
  - Triplet loss $L(x, x^+, x^-) = \emptyset\left(d\big(g(x), g(x^-)\big) - d\big(g(x), g(x^+)\big)\right)$



[Xinwei et al., CVPR 2018]

Introduction

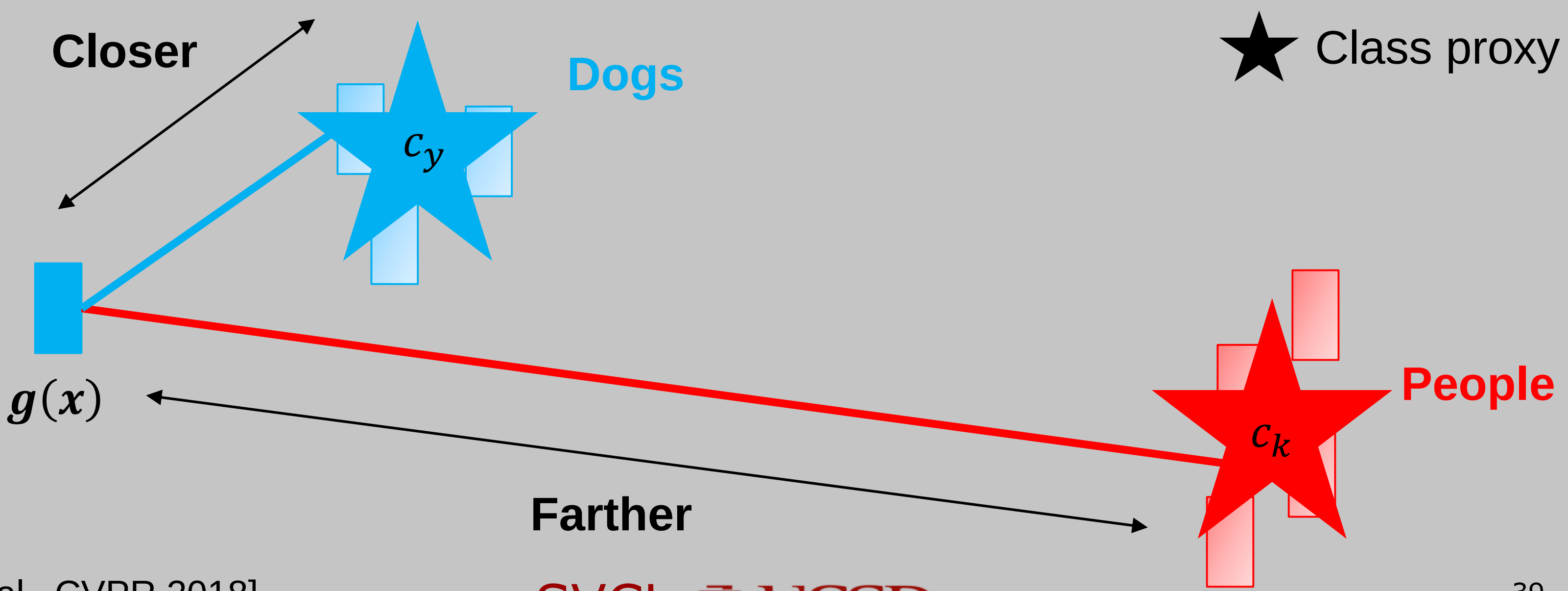
- Metric learning for retrieval task :
  - Margin loss $\emptyset(\mathrm{v}) = \max(0, \mathrm{m} - \mathrm{v})$ with some margin m
  - ~~Triplet loss~~ $\cancel{L(x, x^+, x^-) = \emptyset\Big(d\big(g(x), g(x^-)\big) - d\big(g(x), g(x^+)\big)\Big)}$
  - Triplet center loss $L(x, \boldsymbol{C}) = \emptyset\left(\min_{k \neq y} d(g(x), c_k) - d\big(g(x), c_y\big)\right)$



[Xinwei et al., CVPR 2018]

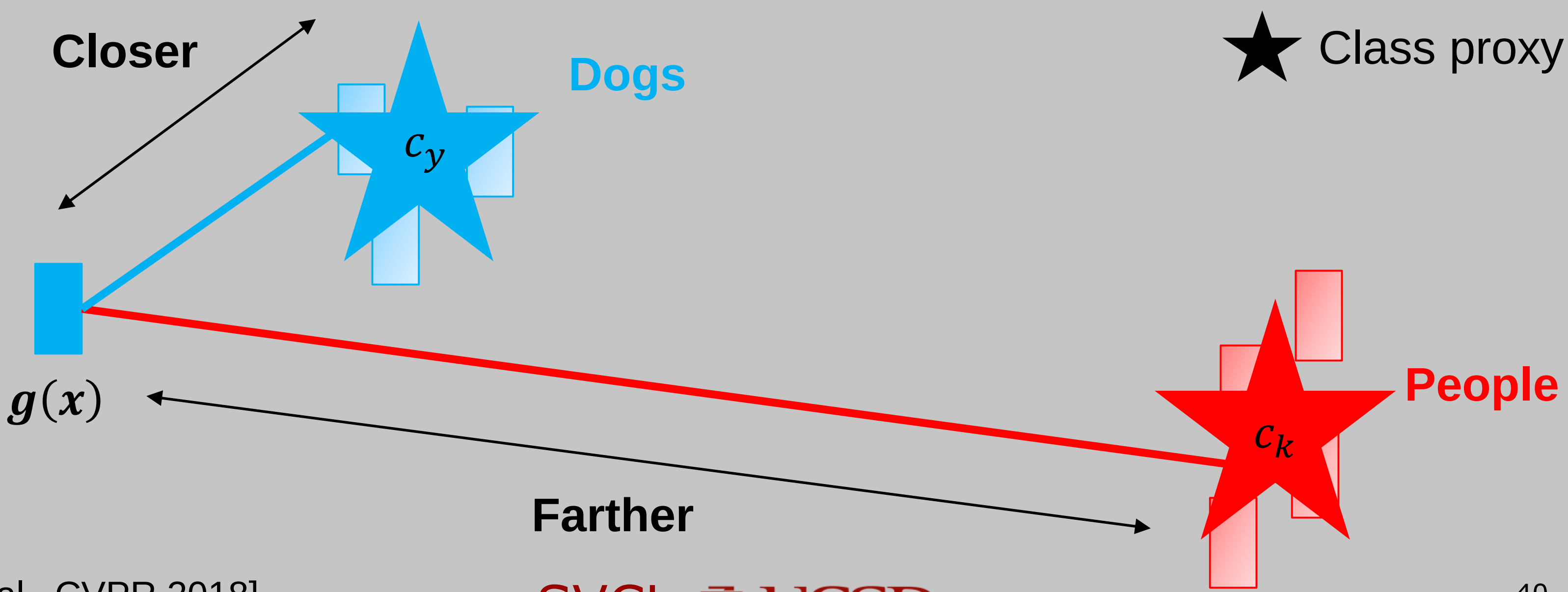
Introduction

- Metric learning for retrieval task :
  - Margin loss $\emptyset(\mathrm{v}) = \max(0, \mathrm{m} - \mathrm{v})$ with some margin m
  - ~~Triplet loss~~ $\cancel{L(x, x^+, x^-) = \emptyset\left(d\left(g(x), \boxed{g(x^-)}\right) - d\left(g(x), g(x^+)\right)\right)}$
  - Triplet center loss $L(x, \boldsymbol{C}) = \emptyset\left(\min_{k \neq y} d(g(x), \boxed{c_k}) - d\left(g(x), c_y\right)\right)$



Introduction

- Metric learning for retrieval task :
  - Margin loss $\emptyset(\mathrm{v}) = \max(0, \mathrm{m} - \mathrm{v})$ with some margin m
  - ~~Triplet loss $L(x, x^+, x^-) = \emptyset\Big(d\big(g(x), g(x^-)\big) - d\big(g(x), \boxed{g(x^+)}\big)\Big)$~~
  - Triplet center loss $L(x, \boldsymbol{C}) = \emptyset\left(\min_{k \neq y} d(g(x), c_k) - d\big(g(x), \boxed{c_y}\big)\right)$



Introduction

- Generating an embedding for different tasks is challenging

Introduction

- Generating an embedding for different tasks is challenging
- Transformations make it more complicated

Introduction

- Generating an embedding for different tasks is challenging
- Transformations make it more complicated
  - Lighting

Introduction

- Generating an embedding for different task is challenging
- Transformations make it more complicated
  - Lighting
  - Viewpoint

Introduction

- Generating an embedding for different task is challenging
- Transformations make it more complicated
  - Lighting
  - Viewpoint
  - Depth

Introduction

- ImageNet pretrained classifier on a warplane
  - Unstable classification output
  - Not robust to transformations

[Deng et al., CVPR 2009]

Introduction

- ImageNet pretrained classifier on a warplane
  - Unstable classification output
  - Not robust to transformations
- ImageNet
  - Lots of images per class
  - No dense viewpoints in dataset

[Deng et al., CVPR 2009]

Introduction

- ImageNet pretrained classifier on a warplane
  - Unstable classification output
  - Not robust to transformations
- ImageNet
  - Lots of images per class
  - No dense viewpoints in dataset
- Difficult to collect multiview data in the real world

[Deng et al., CVPR 2009]

Introduction

- Objects can be imaged from any viewpoint in synthetic graphic world

Introduction

- Objects can be imaged from any viewpoint in synthetic graphic world
- Synthetic dataset
  - ModelNet
  - ShapeNet

[Wu et al., CVPR 2015]
[Angel et al., ICCV 2017]

Introduction

- Synthetic data allows the study of 3D representation

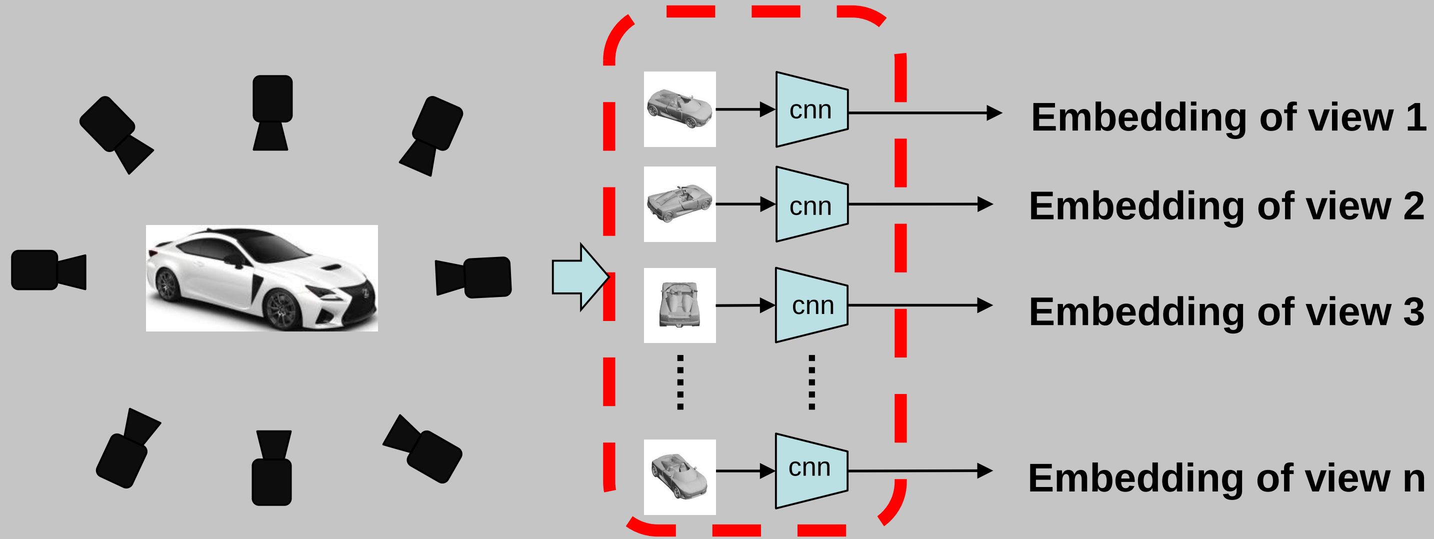
Introduction

- Synthetic data allows the study of 3D representation
- Hang et al. proposed multiview CNN (MVCNN)

[Hang et al., ICCV 2015]

Introduction

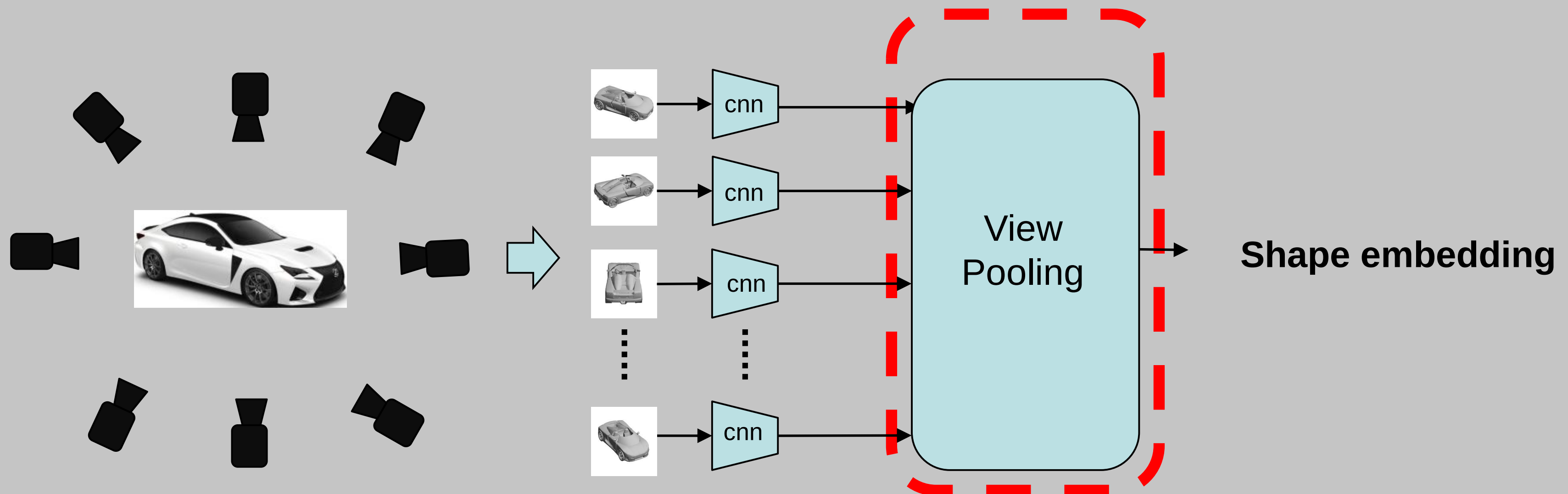
- Synthetic data allows the study of 3D representation
- Hang et al. proposed multiview CNN (MVCNN)
  - Extract embedding of each view with CNN



[Hang et al., ICCV 2015]

Introduction

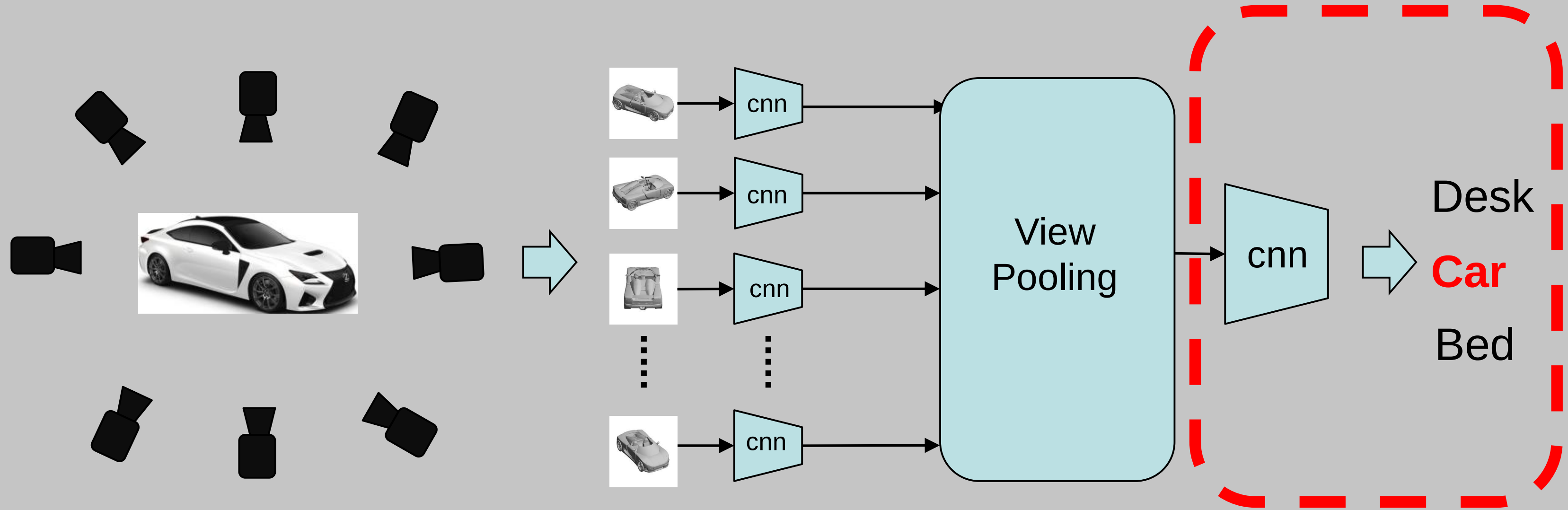
- Synthetic data allows the study of 3D representation
- Hang et al. proposed multiview CNN (MVCNN)
  - Extract embedding of each view with CNN
  - Aggregate multiple embeddings from different views to obtain shape embedding



[Hang et al., ICCV 2015]

Introduction

- Synthetic data allows the study of 3D representation
- Hang et al. proposed multiview CNN (MVCNN)
  - Extract embedding of each view with CNN
  - Aggregate multiple embeddings from different views to obtain shape embedding
  - Perform classification and retrieval tasks with the shape embedding



[Hang et al., ICCV 2015]

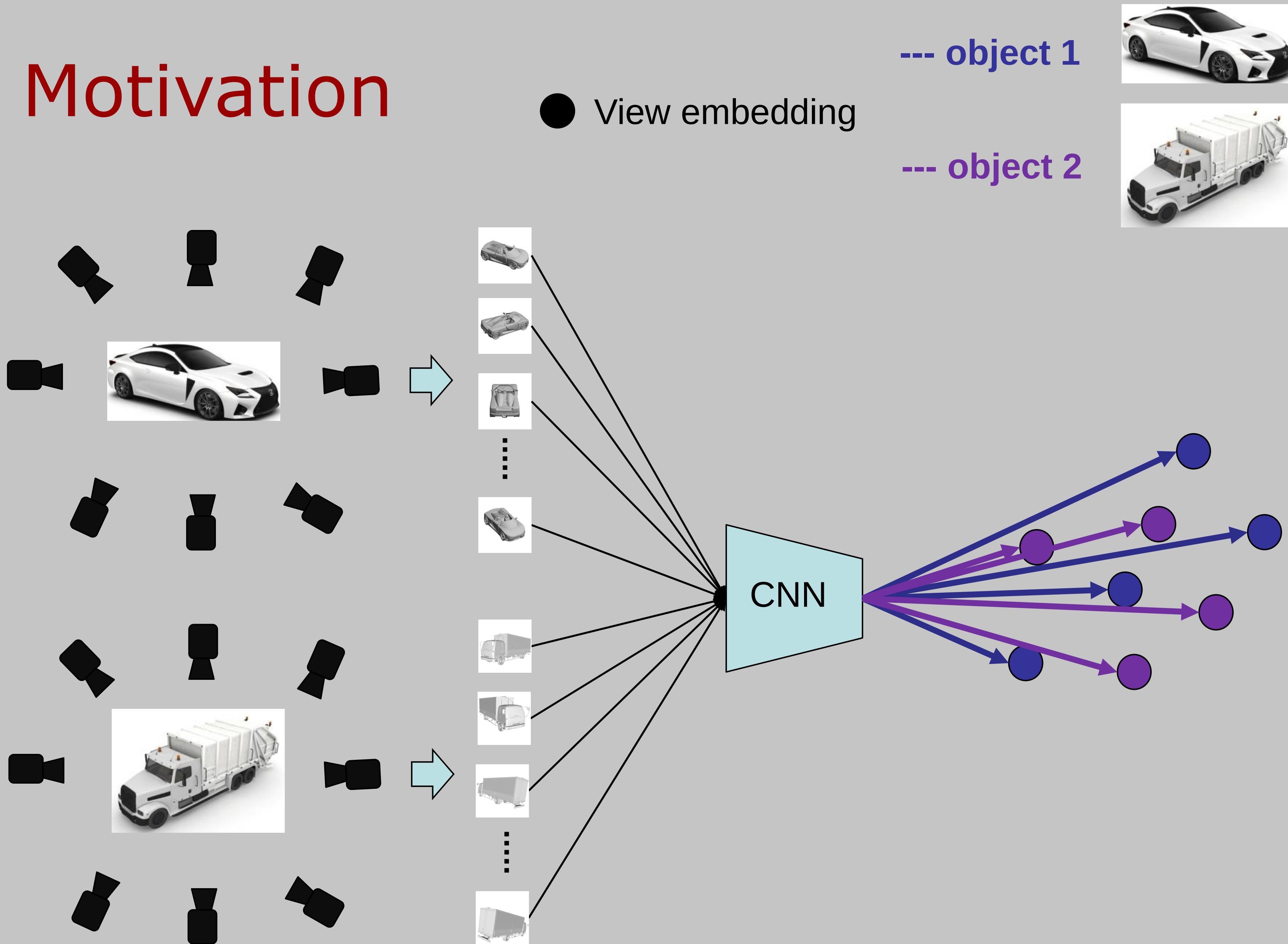
Motivation

- MVCNN performs better then simply averaging multiple predictions of CNN

Motivation

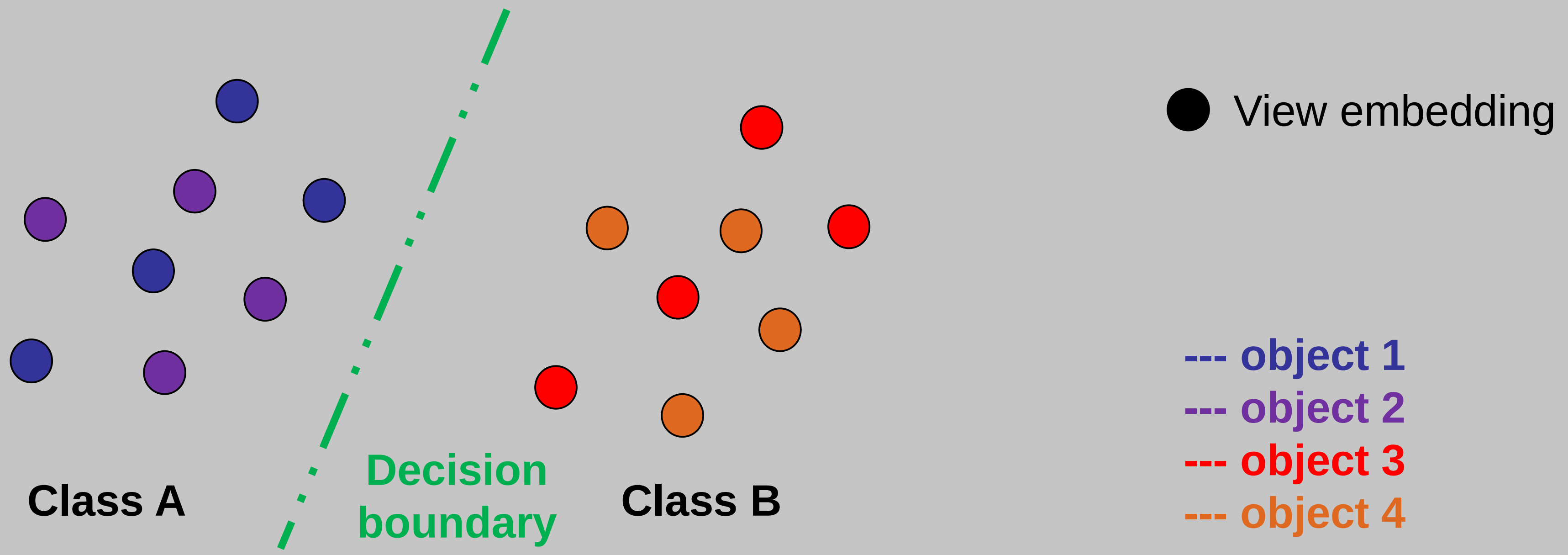
- MVCNN performs better then simply averaging multiple predictions of CNN
- Single view representation (e.g CNN)
  - Better on single view tasks using view embedding
  - No information about relationship between view embeddings from same object

Motivation



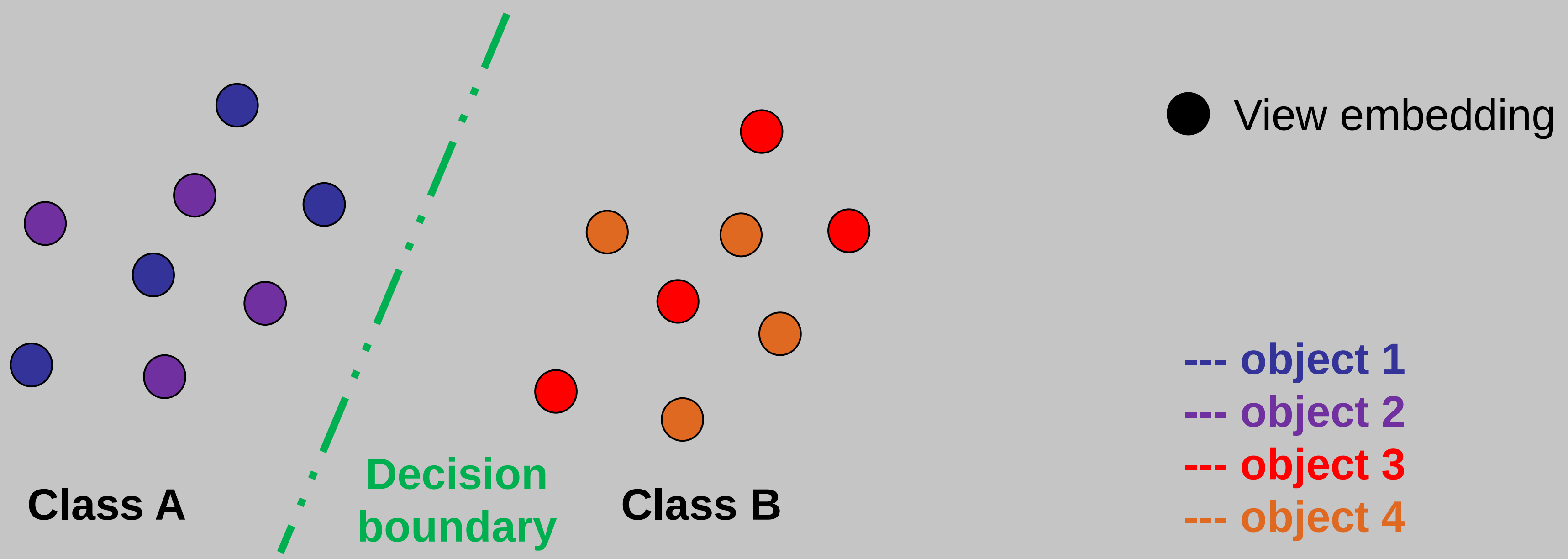
Motivation

- Single view representation (e.g CNN)
  - Configuration of view embeddings for 4 objects in 2 classes



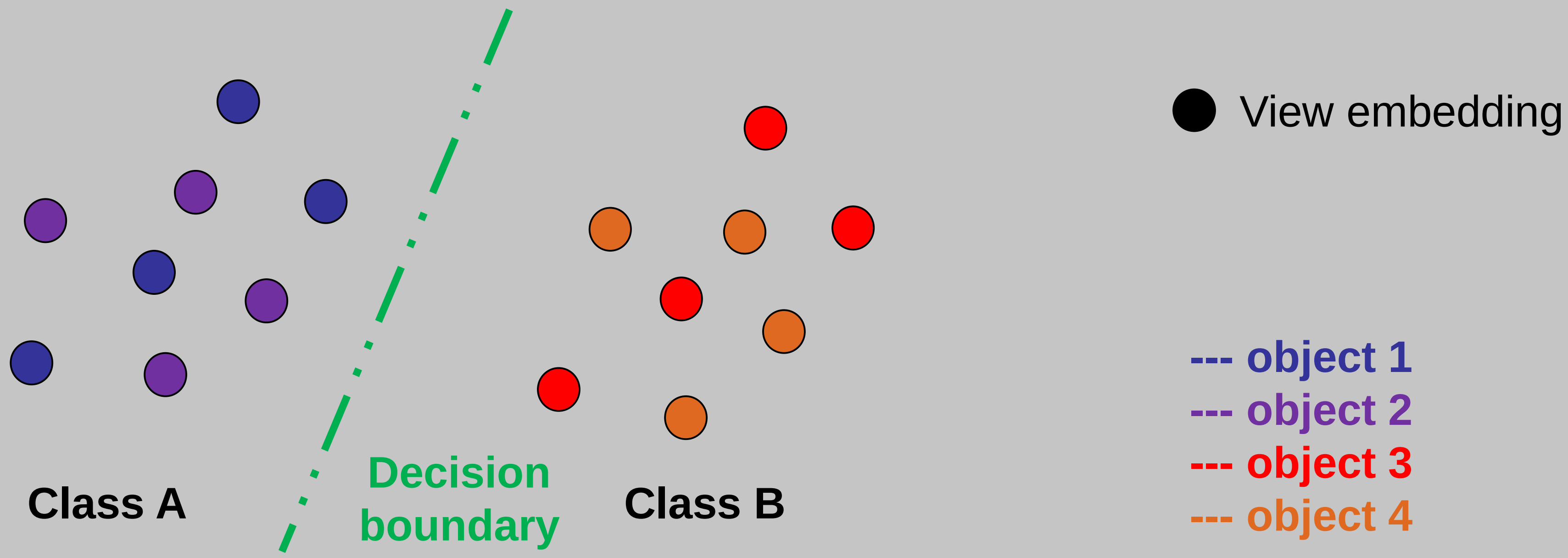
Motivation

- Single view representation (e.g CNN)
  - Configuration of view embeddings for 4 objects in 2 classes
  - Embeddings of images from different objects but same class can interleave with each other



Motivation

- Single view representation (e.g CNN)
  - Configuration of view embeddings for 4 objects in 2 classes
  - Embeddings of images from different objects but same class can interleave with each other
  - Not a good embedding for tasks such as retrieving other views from same object



Motivation

- Multiview representation (e.g MVCNN)
  - Multiview representation is better on multiview tasks using shape embedding

Motivation

- Multiview representation (e.g MVCNN)
  - Multiview representation is better on multiview tasks using shape embedding
  - Shape embedding is an invariant representation of an object

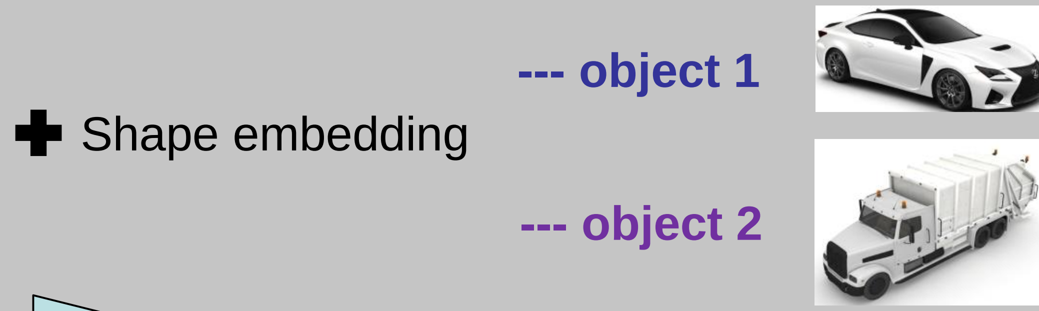
Motivation

- Multiview representation (e.g MVCNN)
  - Multiview representation is better on multiview tasks using shape embedding
  - Shape embedding is an invariant representation of an object
  - But worse on single view task

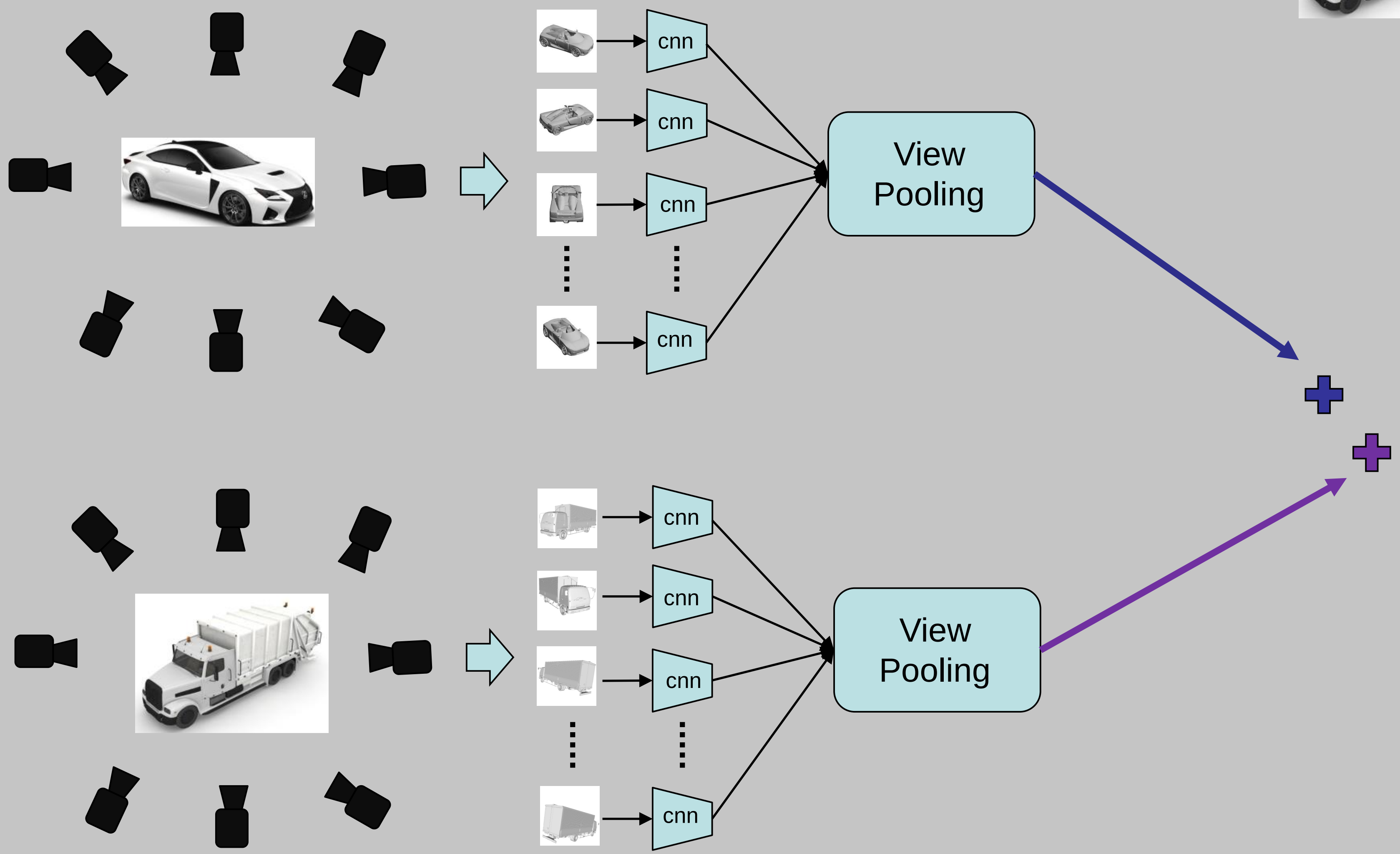
Motivation

- Multiview representation (e.g MVCNN)
  - Multiview representation is better on multiview tasks using shape embedding
  - Shape embedding is an invariant representation of an object
  - But worse on single view task
  - Multiview representation has no constraint between view embeddings of same object

Motivation



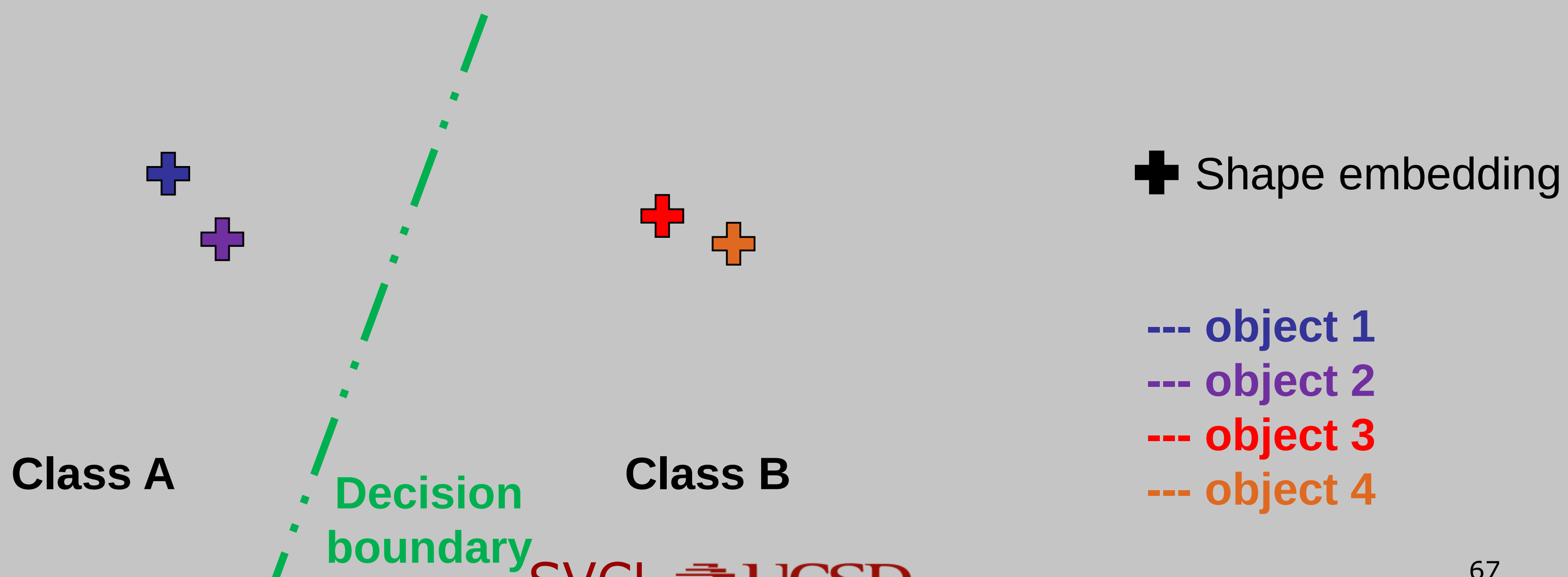
Shape embedding
--- object 1
--- object 2



cnn
cnn
cnn
cnn
View Pooling
cnn
cnn
cnn
cnn
View Pooling

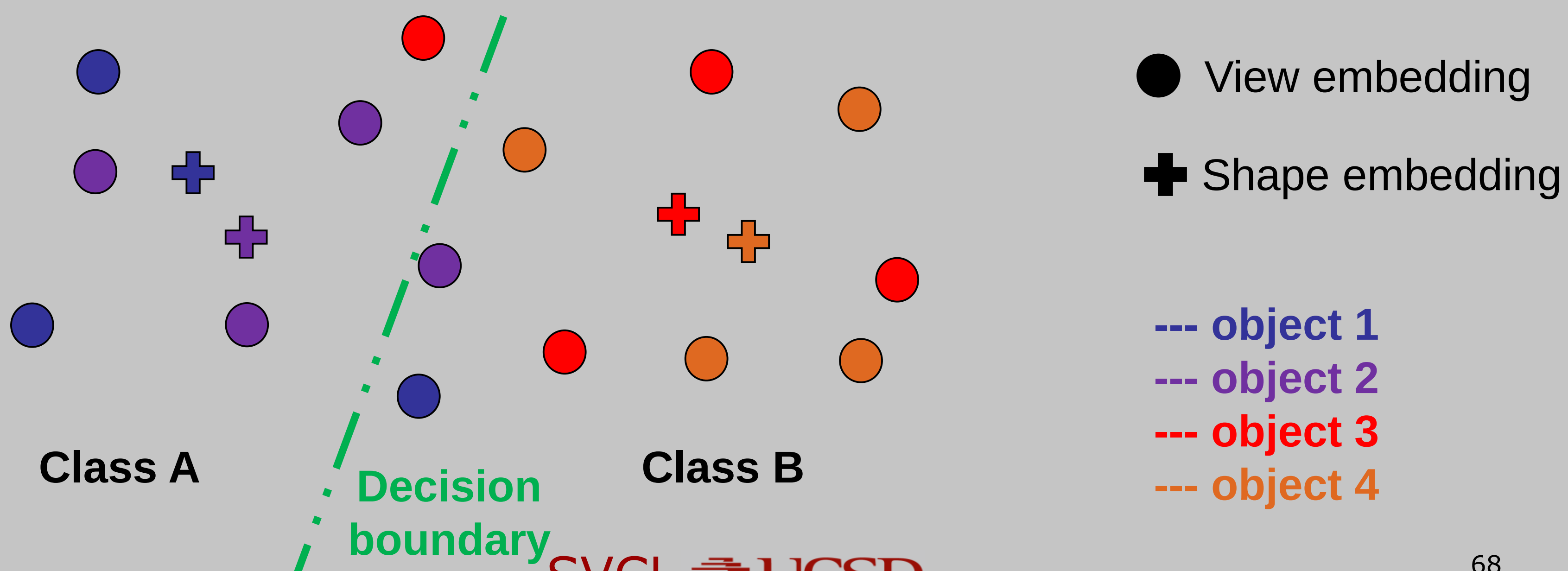
Motivation

- Multiview representation (e.g MVCNN)
  - Multiview representation has no constraint between view embeddings of same object
  - Configuration of shape embeddings for 4 objects in 2 classes
    - All shape embeddings are within decision boundary



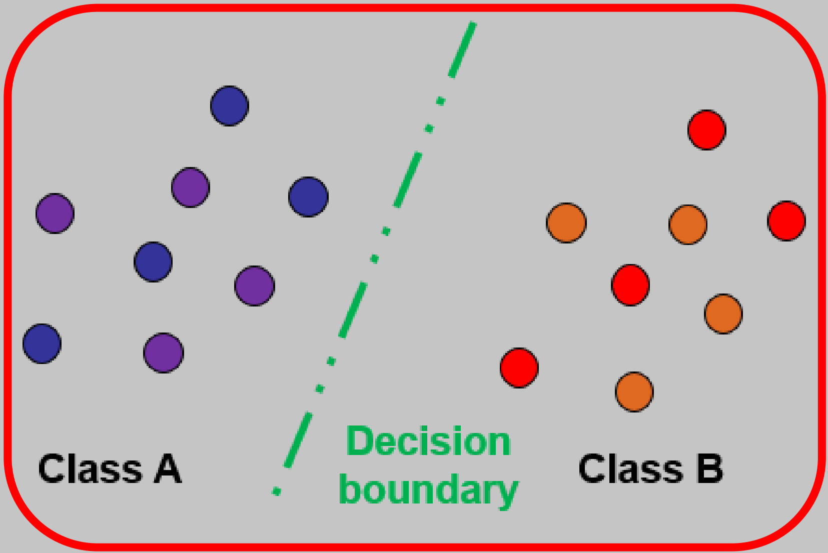
Motivation

- Multiview representation (e.g MVCNN)
  - Multiview representation has no constraint between view embeddings of same object
  - Configuration of shape embeddings for 4 objects in 2 classes
    - All shape embeddings are within decision boundary
  - No guarantee that view embedding will be inside the decision boundary

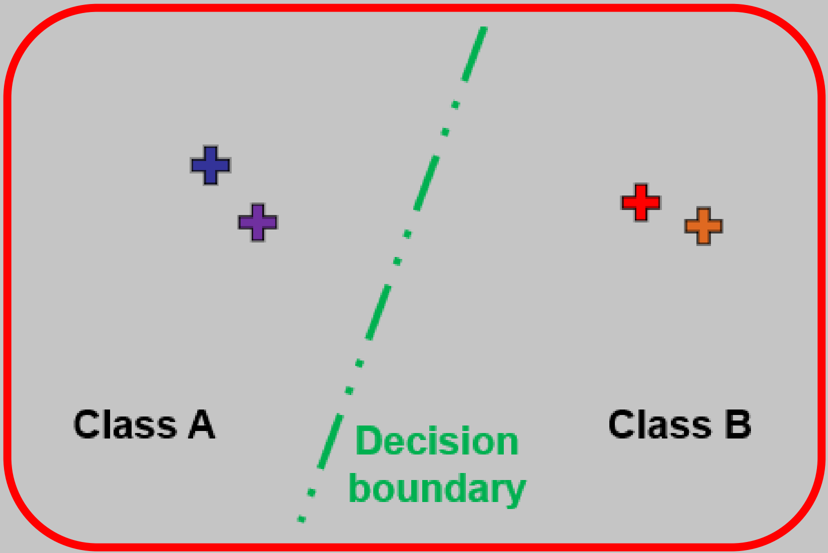


Motivation

- Both single view and multiview representation have its drawback



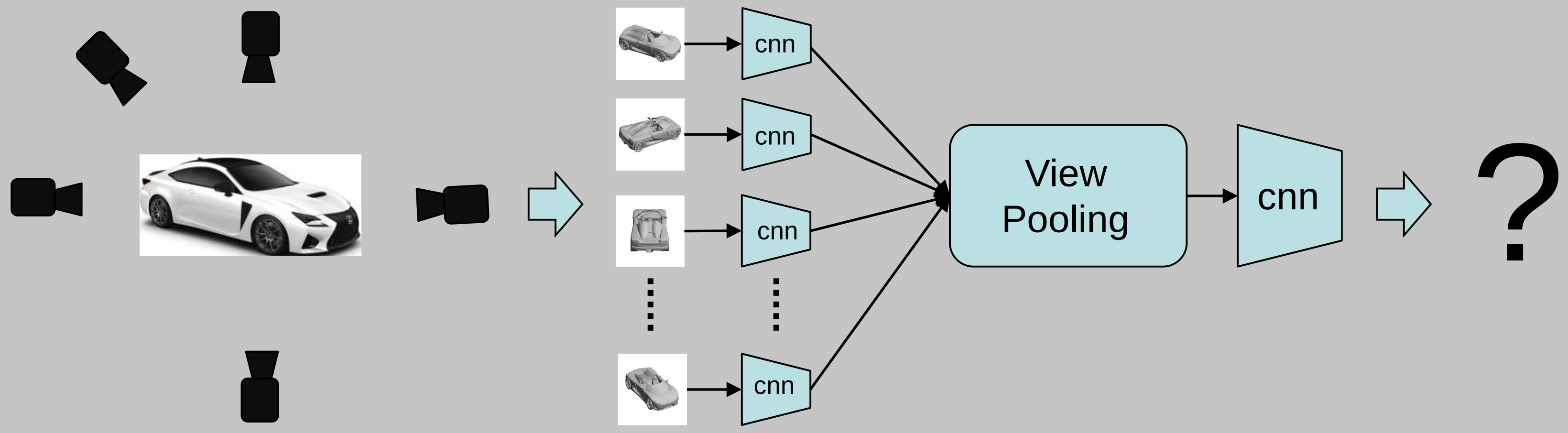
Single view representation



Multiview representation

Motivation

- Both single view and multiview representation have its drawback
- What if only partial views are given?

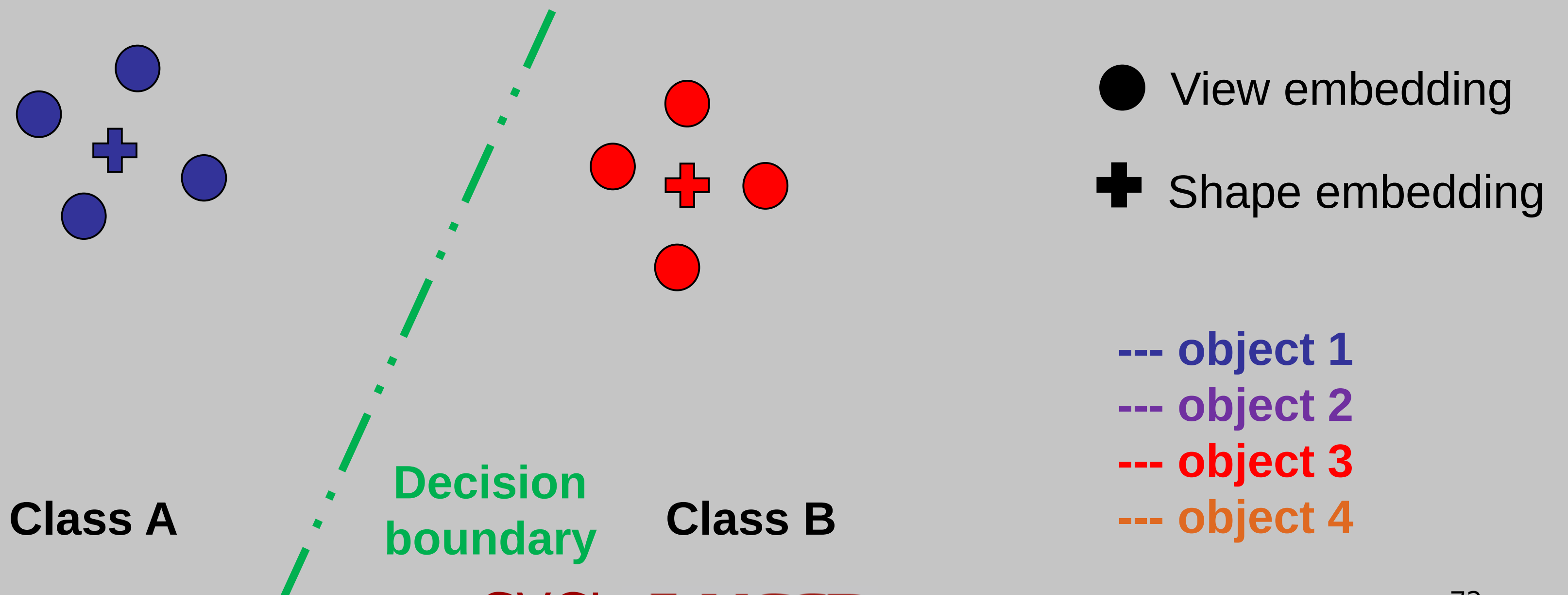


Proposed architecture

- Pose invariant embedding (PIE) is proposed

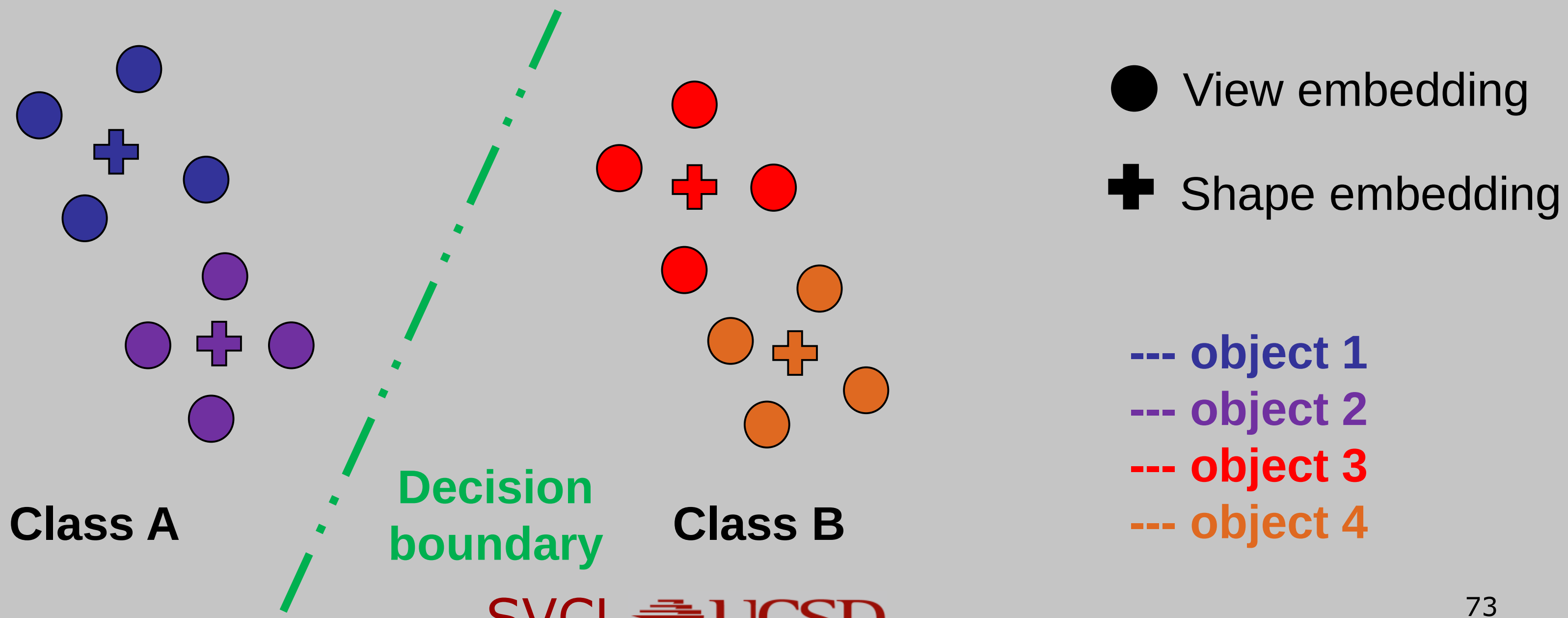
Proposed architecture

- Pose invariant embedding (PIE) is proposed
  - Different views from same object close to each other



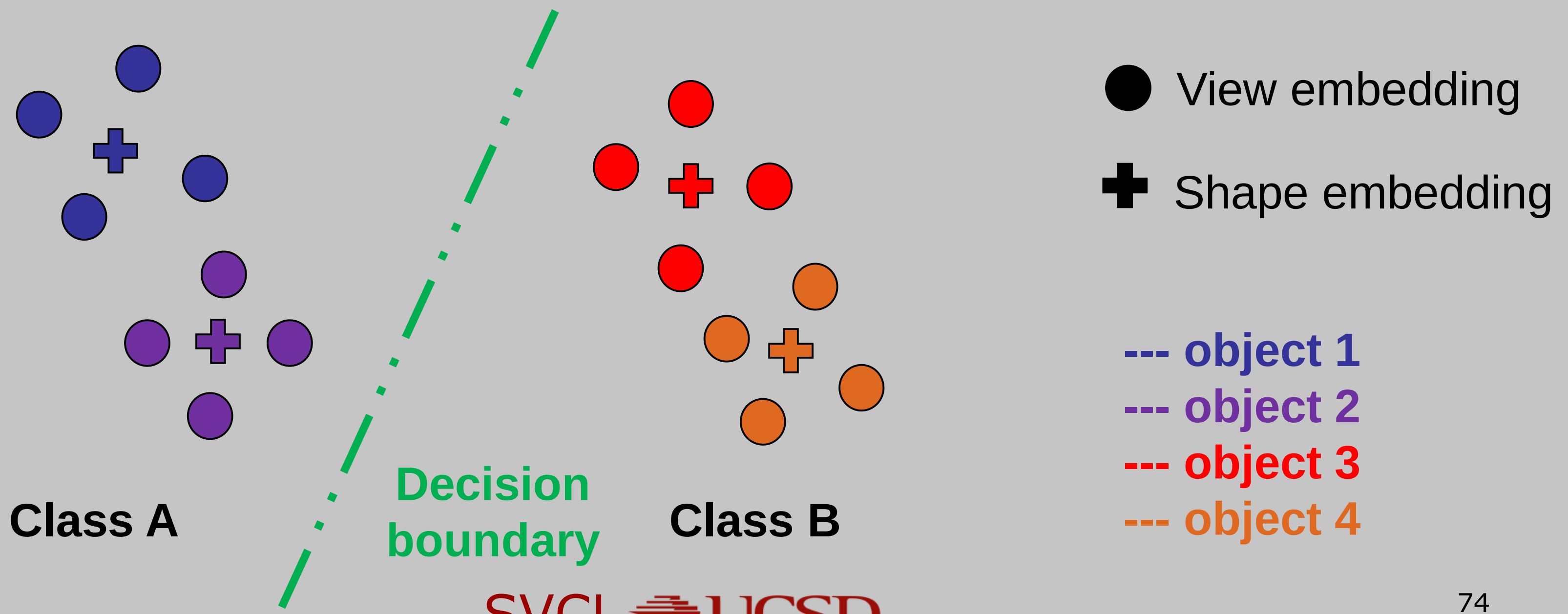
Proposed architecture

- Pose invariant embedding (PIE) is proposed
  - Different views from same object close to each other
  - Different objects from same class close to each other



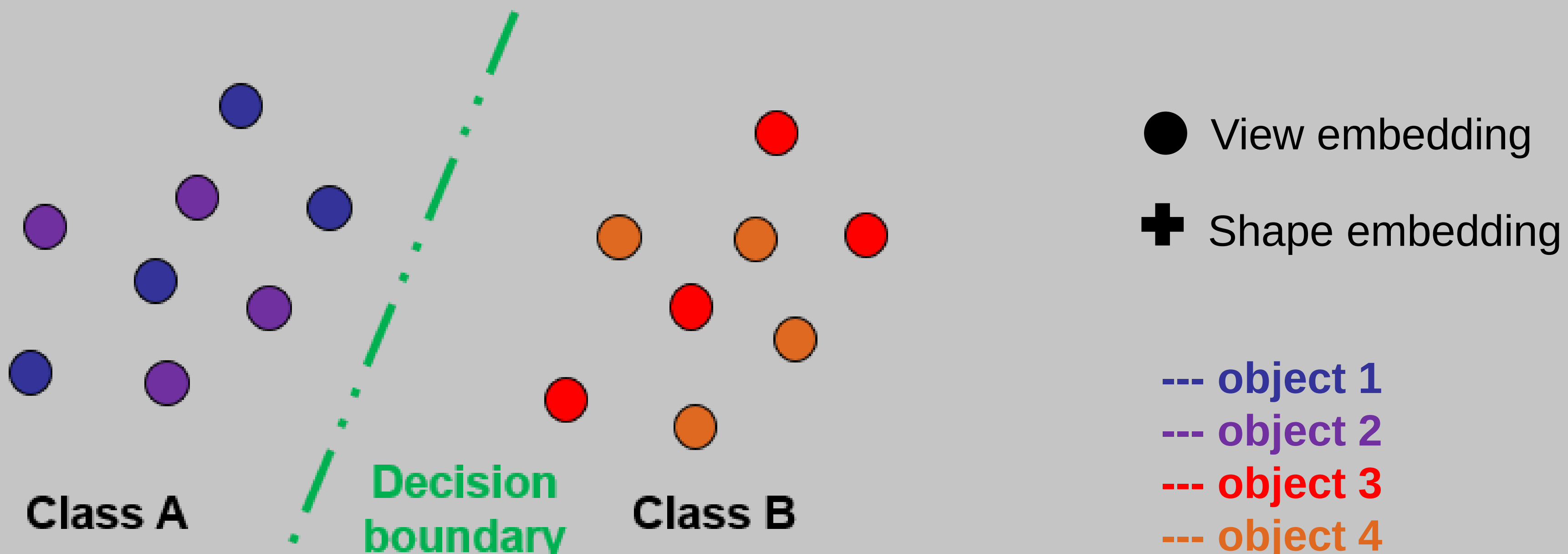
Proposed architecture

- Pose invariant embedding (PIE) is proposed
  - Different views from same object close to each other
  - Different objects from same class close to each other
- More robust to both multiview and single view inference



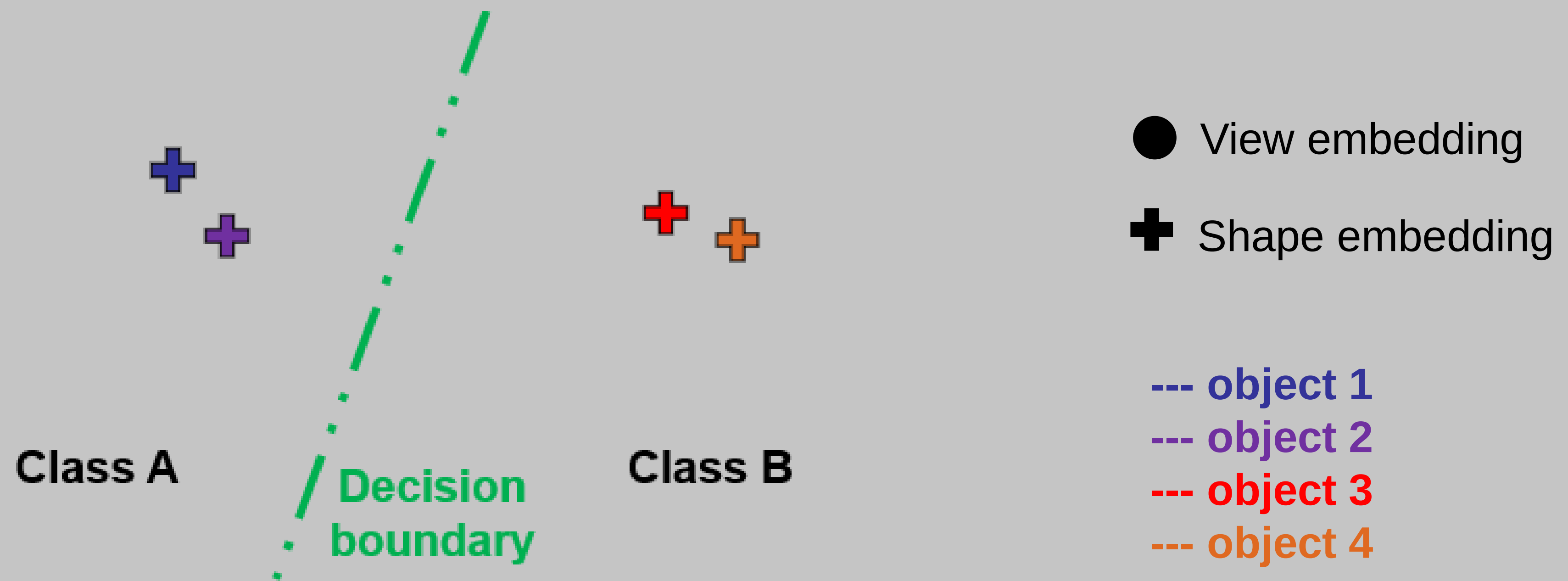
Proposed architecture

- Define y as class label, v as view and s as shape
- Probabilistic formulation
  - Single View: $P_{Y|V}(y|v)$



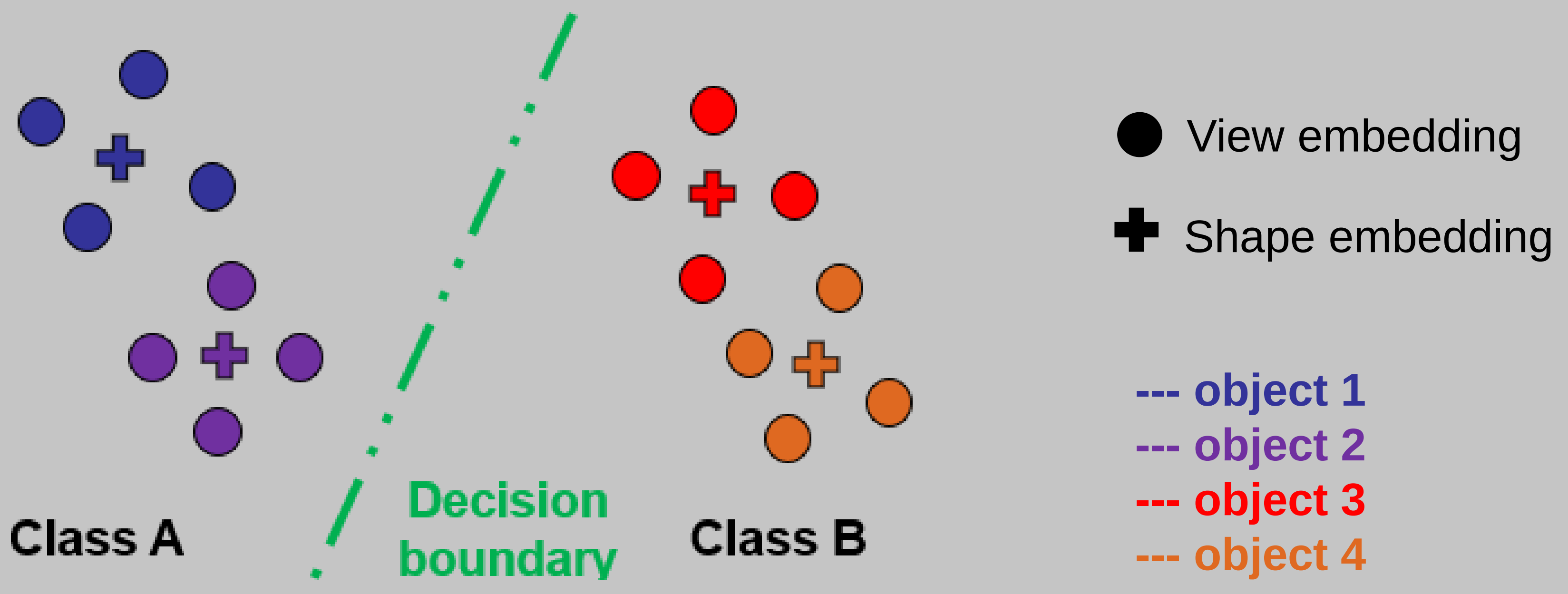
Proposed architecture

- Define y as class label, v as view and s as shape
- Probabilistic formulation
  - Single View: $\mathrm{P}_{Y|V}(y|v)$
  - Multiview: $\mathrm{P}_{Y|S}(y|s)$



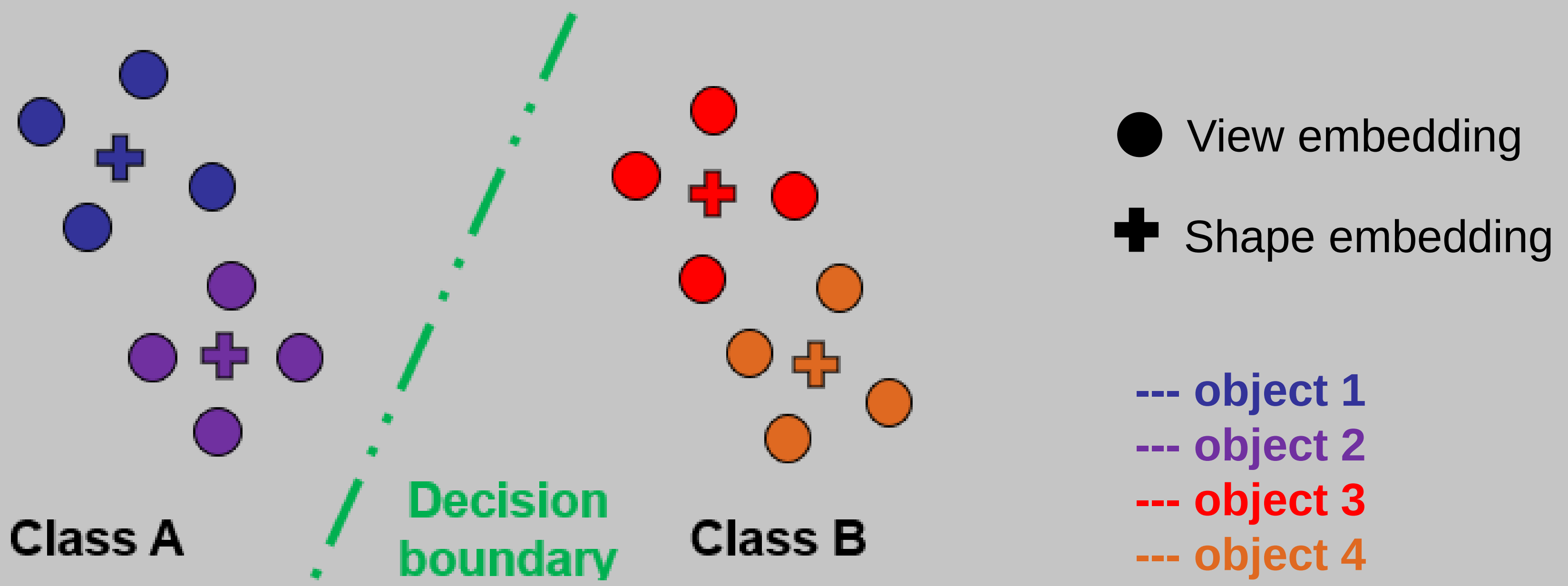
Proposed architecture

- Define y as class label, v as view and s as shape
- Probabilistic formulation
  - Single View: $\mathrm{P}_{Y|V}(y|v)$
  - Multiview: $\mathrm{P}_{Y|S}(y|s)$
  - PIE: $\mathrm{P}_{Y|V}(y|v) = \sum_s P_{Y|S,V}(y|s,v)P_{S|V}(s|v)$



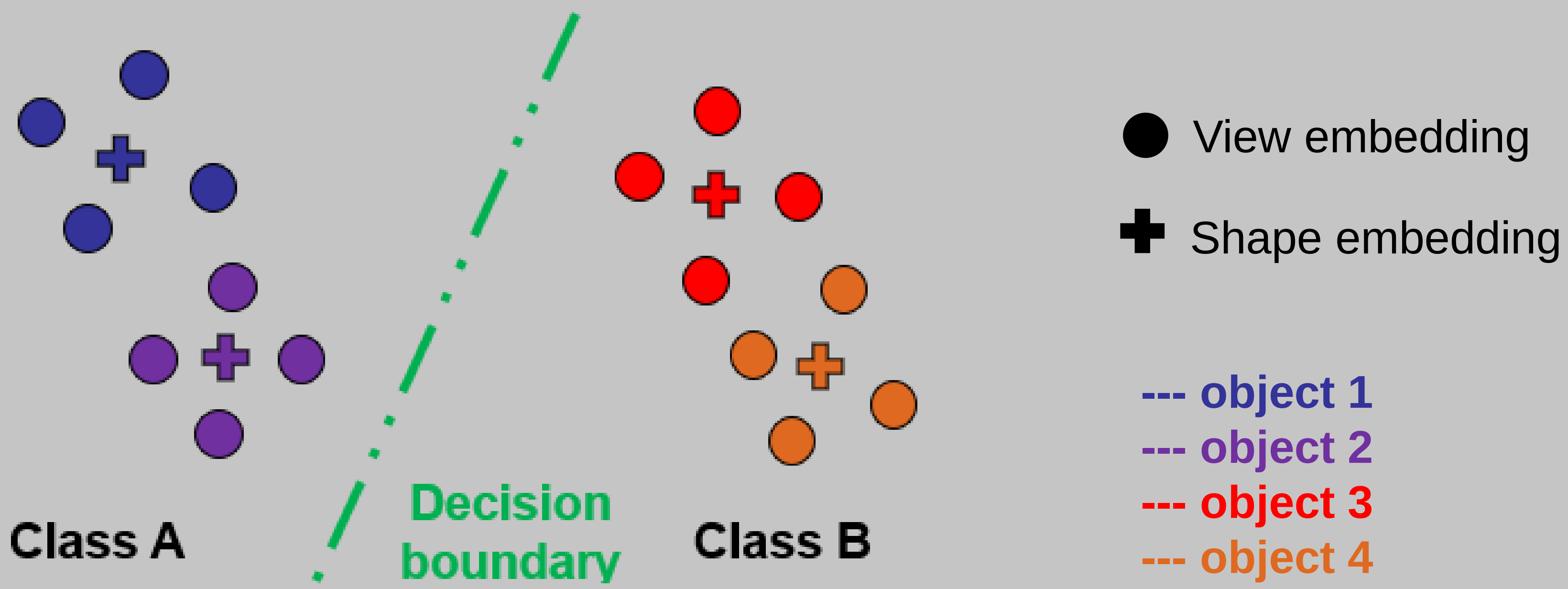
Proposed architecture

- Shape embedding is an invariant representation of an object



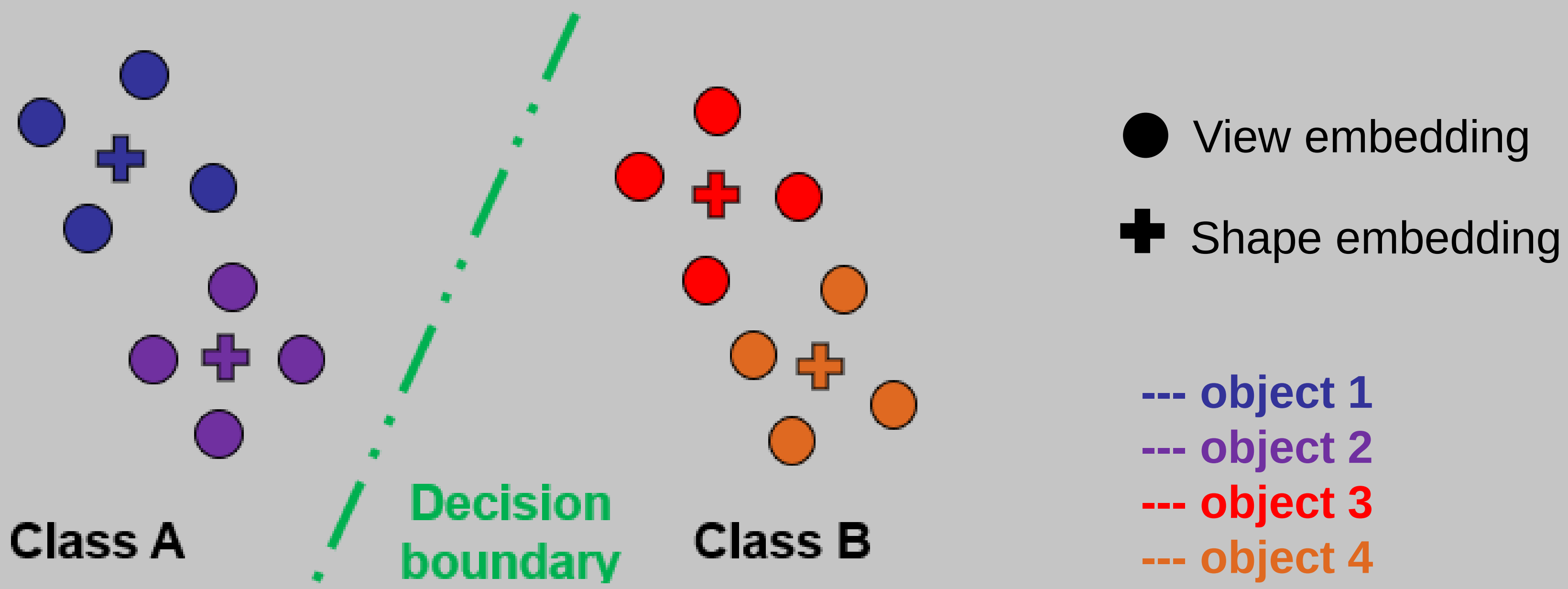
Proposed architecture

- Shape embedding is an invariant representation of an object
- Given the object is known, class is independent of view



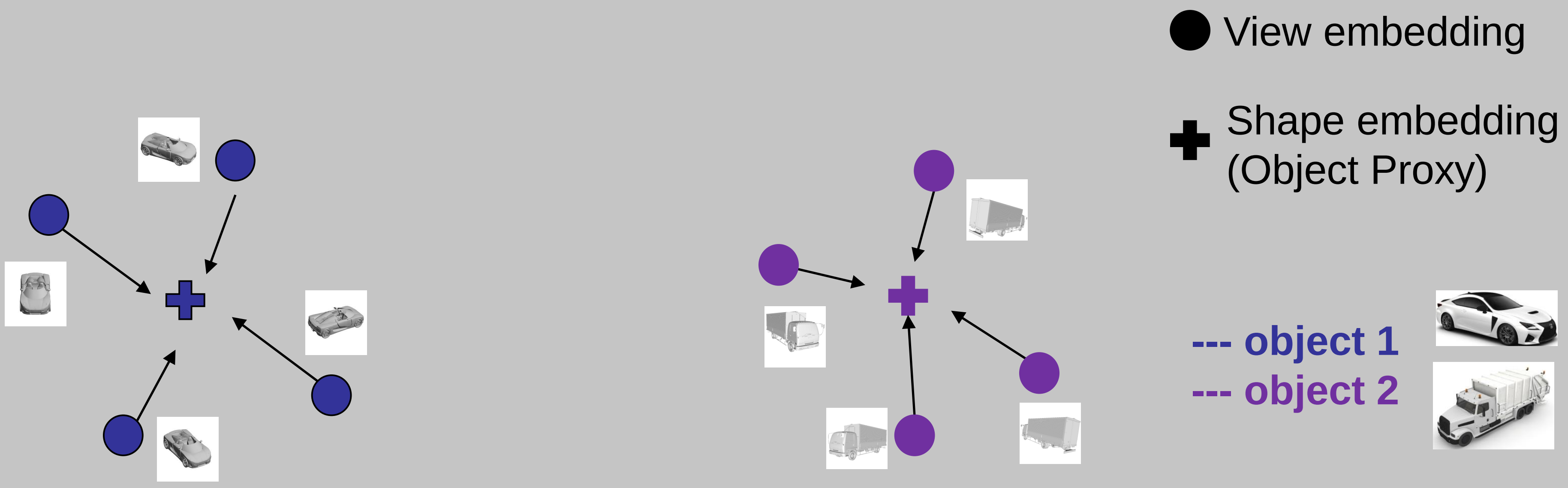
Proposed architecture

- Shape embedding is an invariant representation of an object
- Given the object is known, class is independent of view
  - PIE: $P_{Y|V}(y|v) = \sum_s P_{Y|S,V}(y|s,v) P_{S|V}(s|v) = \sum_s P_{Y|S}(y|s) P_{S|V}(s|v)$



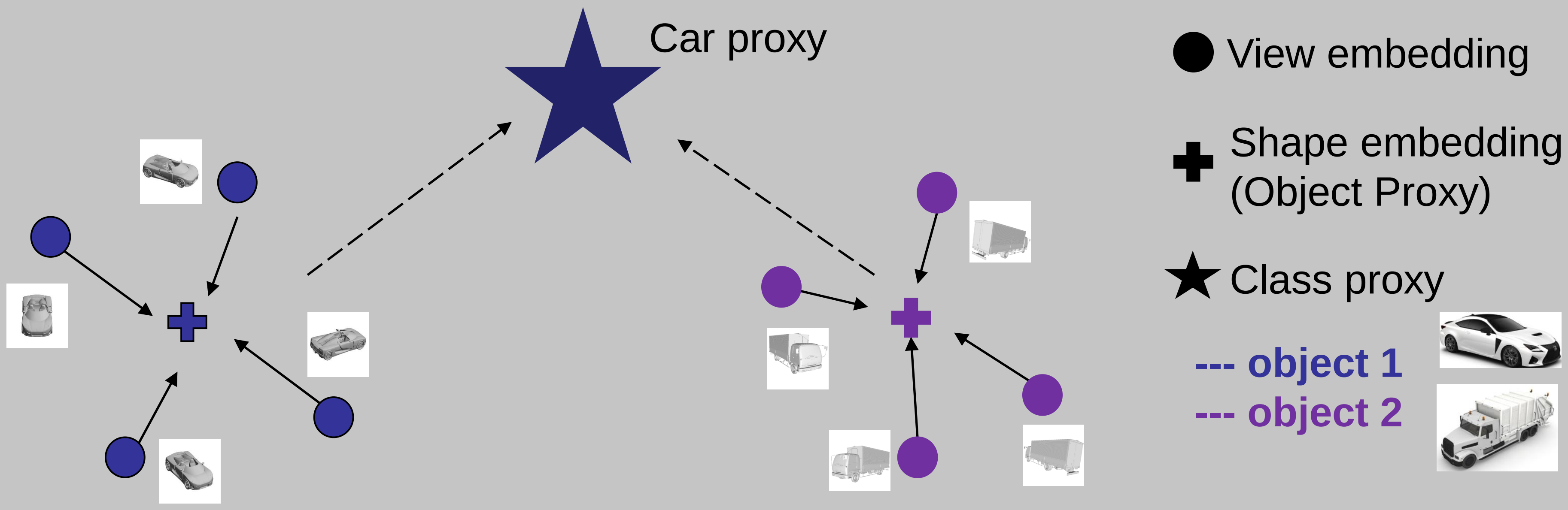
Proposed architecture

- Hierarchical models $P_{Y|V}(y|v) = \sum_s P_{Y|S}(y|s) P_{S|V}(s|v)$
  - View to object model
    - Shape embedding is used for object proxy
    - Make view embedding close to the associated object proxy



Proposed architecture

- Hierarchical models $P_{Y|V}(y|v) = \sum_s P_{Y|S}(y|s) P_{S|V}(s|v)$
  - View to object model
    - Shape embedding is used for object proxy
    - Make view embedding close to the associated object proxy
  - Object to class model
    - Make object proxy close to the associated class proxy



Proposed architecture

- Define y as class, v as view, s as shape and c_y as class proxy
- Pose invariant distance
  - $d^{inv}(v, s, c_y) = \alpha * d(v, s) + \beta * d(s, c_y)$

Proposed architecture

- Define y as class, v as view, s as shape and c_y as class proxy
- Pose invariant distance
  - $d^{inv}(v, s, c_y) = \alpha * d(v, s) + \beta * d(s, c_y)$
- Take proxy based method for example
  - Single view representation
    - $Loss = \frac{\exp(-d(v, c_y))}{\sum_{i \neq y} \exp(-d(v, c_i))}$

Proposed architecture

- Define y as class, v as view, s as shape and c_y as class proxy
- Pose invariant distance
  - $d^{inv}(v, s, c_y) = \alpha * d(v, s) + \beta * d(s, c_y)$
- Take proxy based method for example
  - Single view representation
    - $Loss = \frac{\exp(-d(v,c_y))}{\sum_{i \neq y} \exp(-d(v,c_i))}$
  - Multiview representation
    - $Loss = \frac{\exp(-d(s,c_y))}{\sum_{i \neq y} \exp(-d(s,c_i))}$

Proposed architecture

- Define y as class, v as view, s as shape and c_y as class proxy
- Pose invariant distance
  - $d^{inv}(v, s, c_y) = \alpha * d(v, s) + \beta * d(s, c_y)$
- Take proxy based method for example
  - Single view representation
    - $Loss = \frac{\exp(-d(v,c_y))}{\sum_{i \neq y} \exp(-d(v,c_i))}$
  - Multiview representation
    - $Loss = \frac{\exp(-d(s,c_y))}{\sum_{i \neq y} \exp(-d(s,c_i))}$
  - PIE
    - $Loss = \frac{\exp(-d^{inv}(v,s,c_y))}{\sum_{i \neq y} \exp(-d^{inv}(v,s,c_i))}$

Proposed architecture

- Define y as class, v as view, s as shape and c_y as class proxy
- Pose invariant distance
  - $d^{inv}(v, s, c_y) = \alpha * d(v, s) + \beta * d(s, c_y)$
- Take proxy based method for example
  - Single view representation
    - $Loss = \frac{\exp(-d(v, c_y))}{\sum_{i \neq y} \exp(-d(v, c_i))}$
  - Multiview representation
    - $Loss = \frac{\exp(-d(s, c_y))}{\sum_{i \neq y} \exp(-d(s, c_i))}$
  - PIE
    - $Loss = \frac{\exp(-d^{inv}(v, s, c_y))}{\sum_{i \neq y} \exp(-d^{inv}(v, s, c_i))}$

α=0, β=1

Proposed architecture

- The proposed idea can be incorporated with different training approaches
  - Proxy

	Representation		
	Single view	**Multiview**	**PIE**
Proxy	**Existed**	**Missing**	**Proposing**

Proposed architecture

- The proposed idea can be incorporated with different training approaches
  - Proxy
  - CNN

	Representation		
	Single view	Multiview	PIE
Proxy	Existed	Missing	Proposing
CNN	Existed	Existed	Proposing

Proposed architecture

- The proposed idea can be incorporated with different training approaches
  - Proxy
  - CNN
  - Triplet Center

	Representation		
	Single view	Multiview	PIE
Proxy	Existed	Missing	Proposing
CNN	Existed	Existed	Proposing
Triplet Center	Missing	Existed	Proposing

Proposed architecture

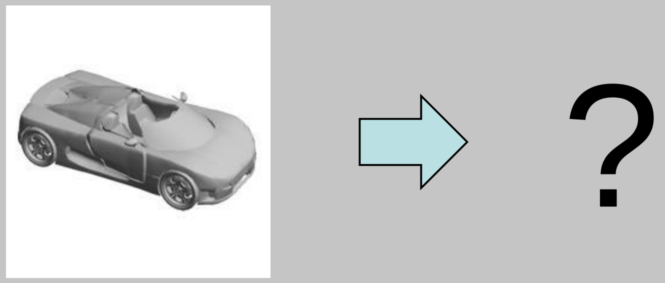
- The proposed idea can be incorporated with different training approaches
  - Proxy
  - CNN
  - Triplet Center
- Taxonomy of embedding
  - Some missing approaches in the literature are found

	Representation		
	Single view	**Multiview**	**PIE**
Proxy	Existed	Missing	Proposing
CNN	Existed	Existed	Proposing
Triplet Center	Missing	Existed	Proposing

Taxonomy of embedding

Experiment

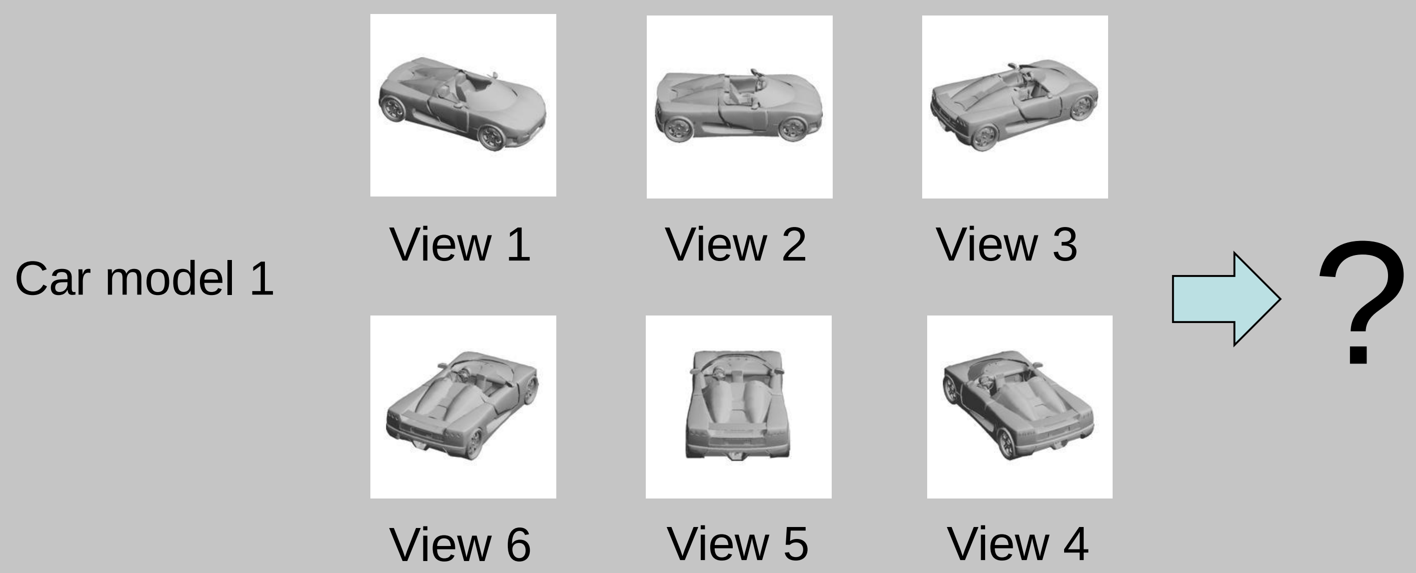
- 5 different tasks are evaluated
  - Classification:
    - Single view classification



View 1 of car model 1

Experiment

- 5 different tasks are evaluated
  - Classification:
    - Single view classification
    - Multiview classification



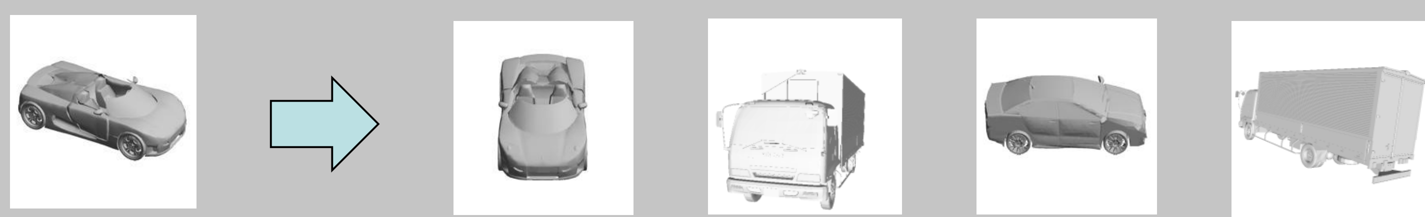
Experiment

- 5 different tasks are evaluated
  - Classification:
    - Single view classification
    - Multiview classification
  - Retrieval:
    - Single view object retrieval

View 1 of car model 1 | View 2 | View 3 | View 4 | View 5 | View 6

Experiment

- 5 different tasks are evaluated
  - Classification:
    - Single view classification
    - Multiview classification
  - Retrieval:
    - Single view object retrieval
    - Single view class retrieval

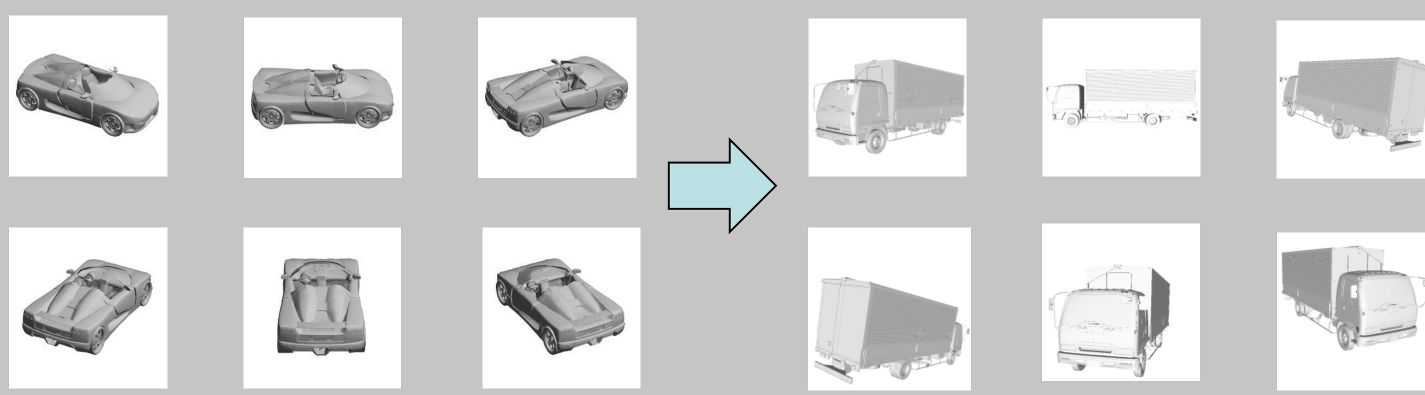


View 1 of car model 1

Other views of various cars

Experiment

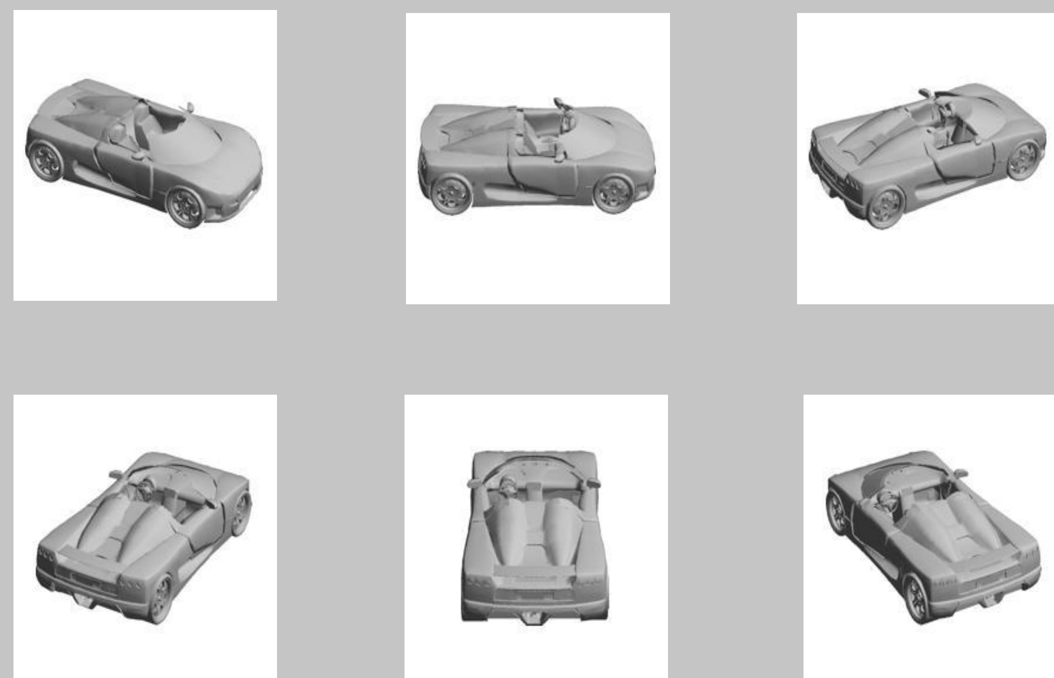
- 5 different tasks are evaluated
  - Classification:
    - Single view classification
    - Multiview classification
  - Retrieval:
    - Single view object retrieval
    - Single view class retrieval
    - Multiview class retrieval



Car model 1 Car model 2

Experiment

- 3 different datasets are evaluated
  - ModelNet



[Wu et al., CVPR 2015]

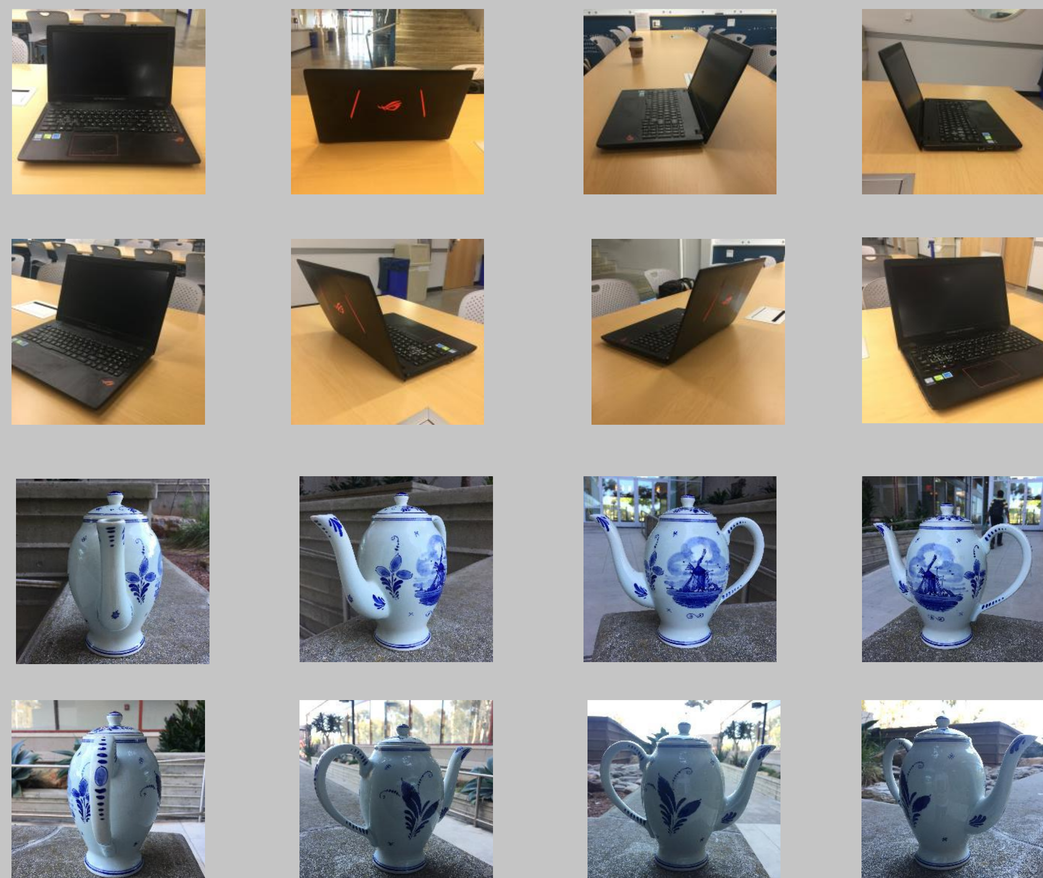
Experiment

- 3 different datasets are evaluated
  - ModelNet
  - MIRO

[Asako et al., CVPR 2018]

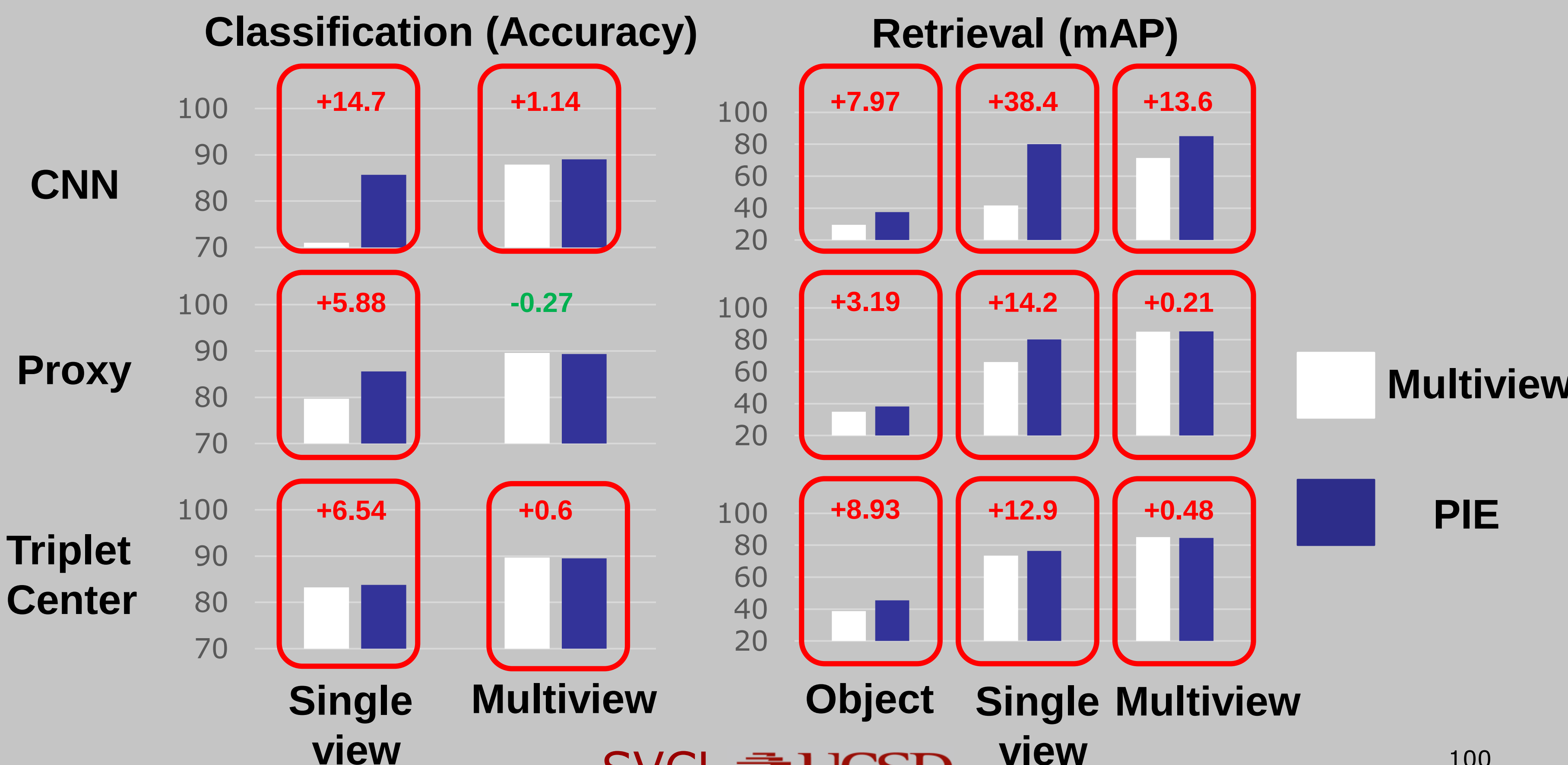
Experiment

- 3 different datasets are evaluated
  - ModelNet
  - MIRO
  - ObjectPI
    - 500 objects



Experiment

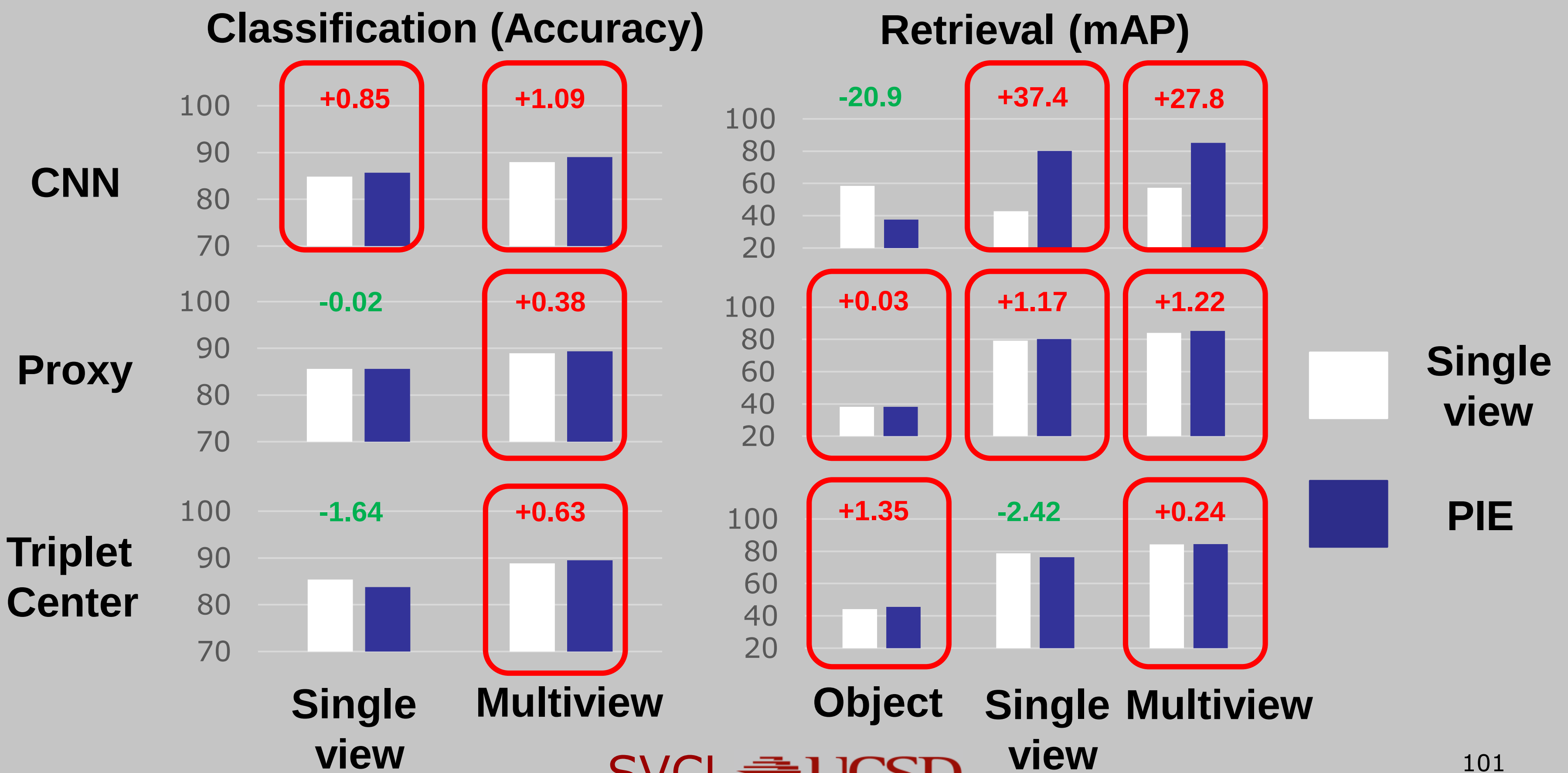
- PIEs wins multiview representation on 14 of the 15 results (5 tasks x 3 approaches) on ModelNet



SVCL UCSD

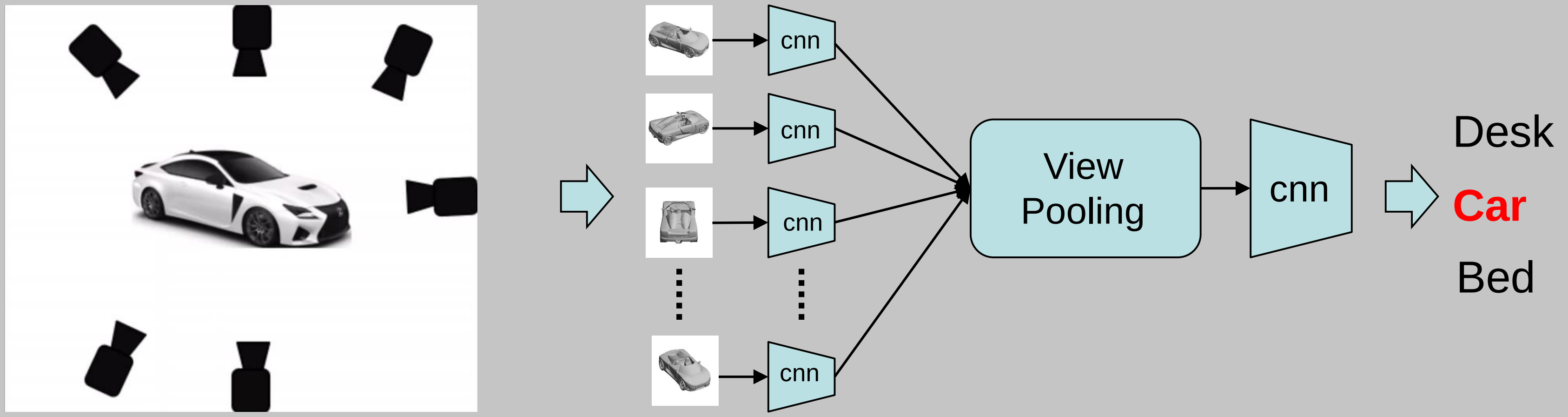
Experiment

- PIEs wins single view representation on 11 of the 15 results (5 tasks x 3 approaches) on ModelNet



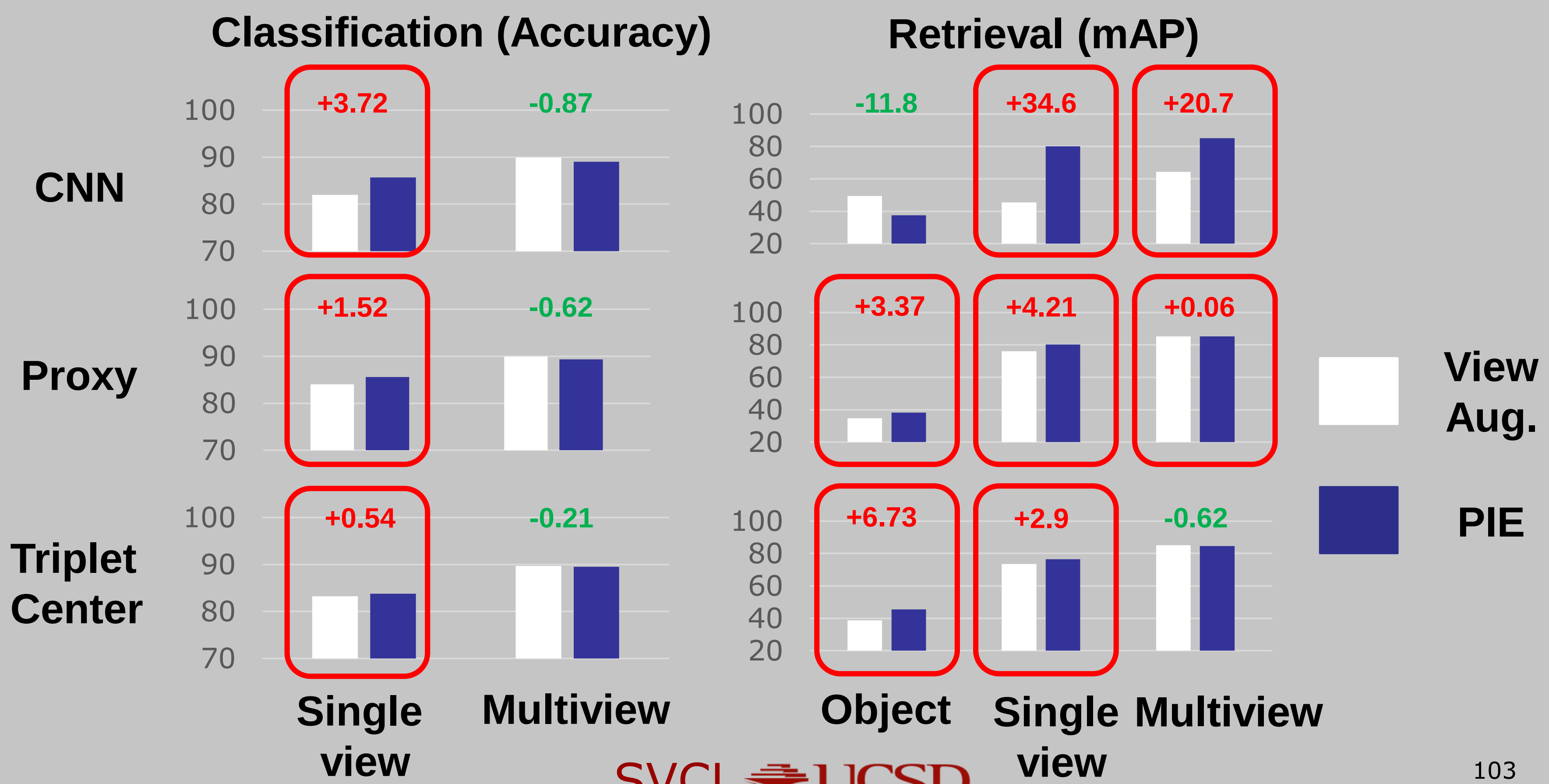
Experiment

- Training with view augmentation
  - Different number of views are provided to classifier



Experiment

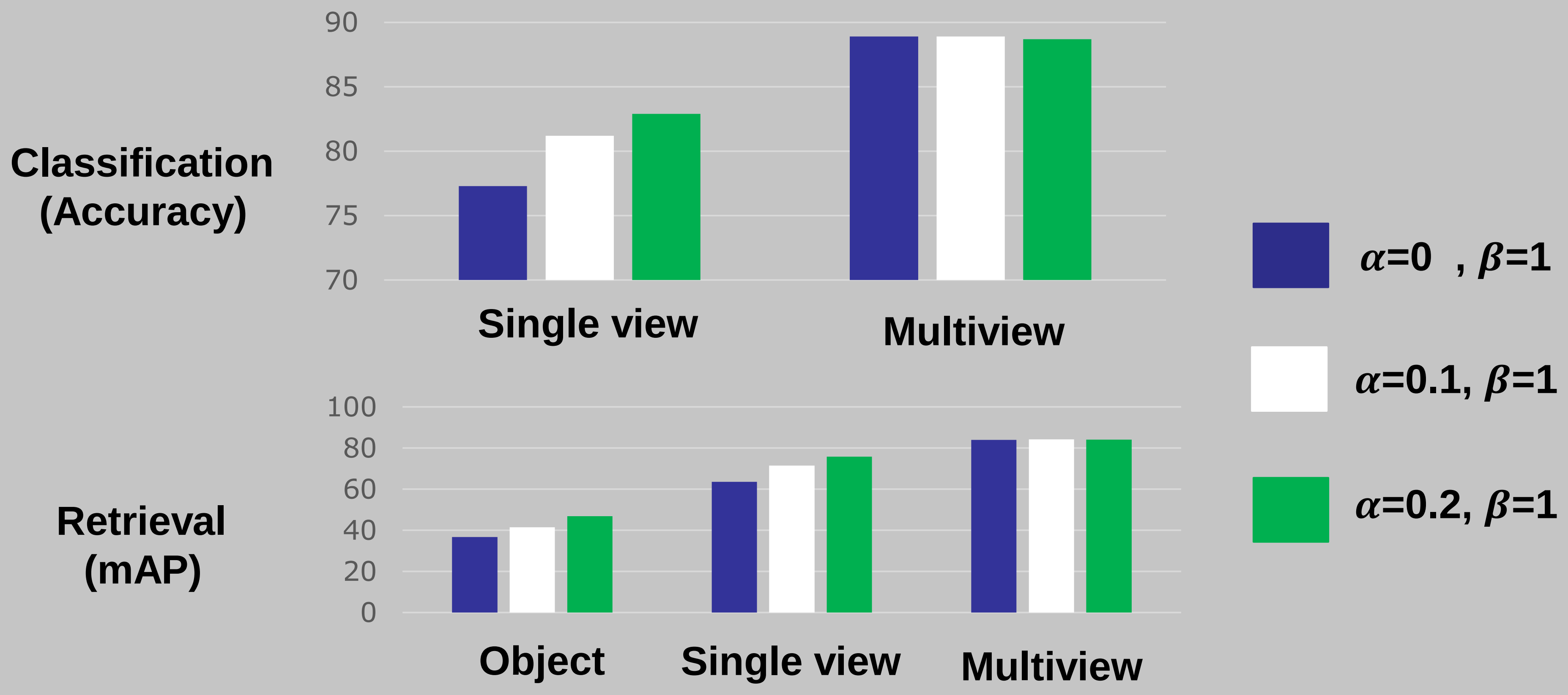
- PIEs wins view augmentation training on 10 of the 15 results (5 tasks x 3 approaches) on ModelNet



Experiment

$$d^{inv}(v, s, c_y) = \alpha * d(v, s) + \beta * d(s, c_y)$$

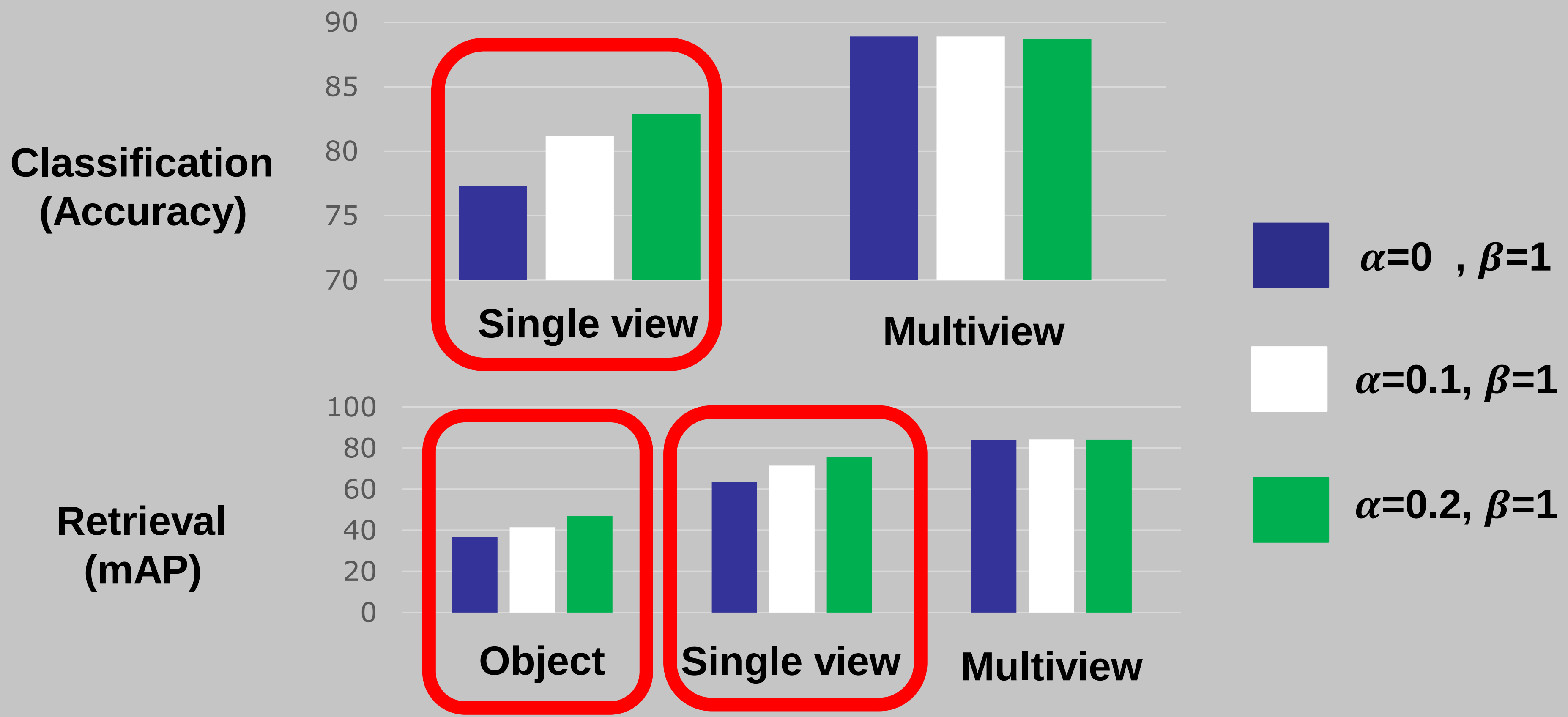
- Ablation study of pose invariant distance
  - As $\boldsymbol{\alpha}$ increase, results of single view tasks become better
  - As $\boldsymbol{\alpha}$ increase, results of multiview tasks become worse



Experiment

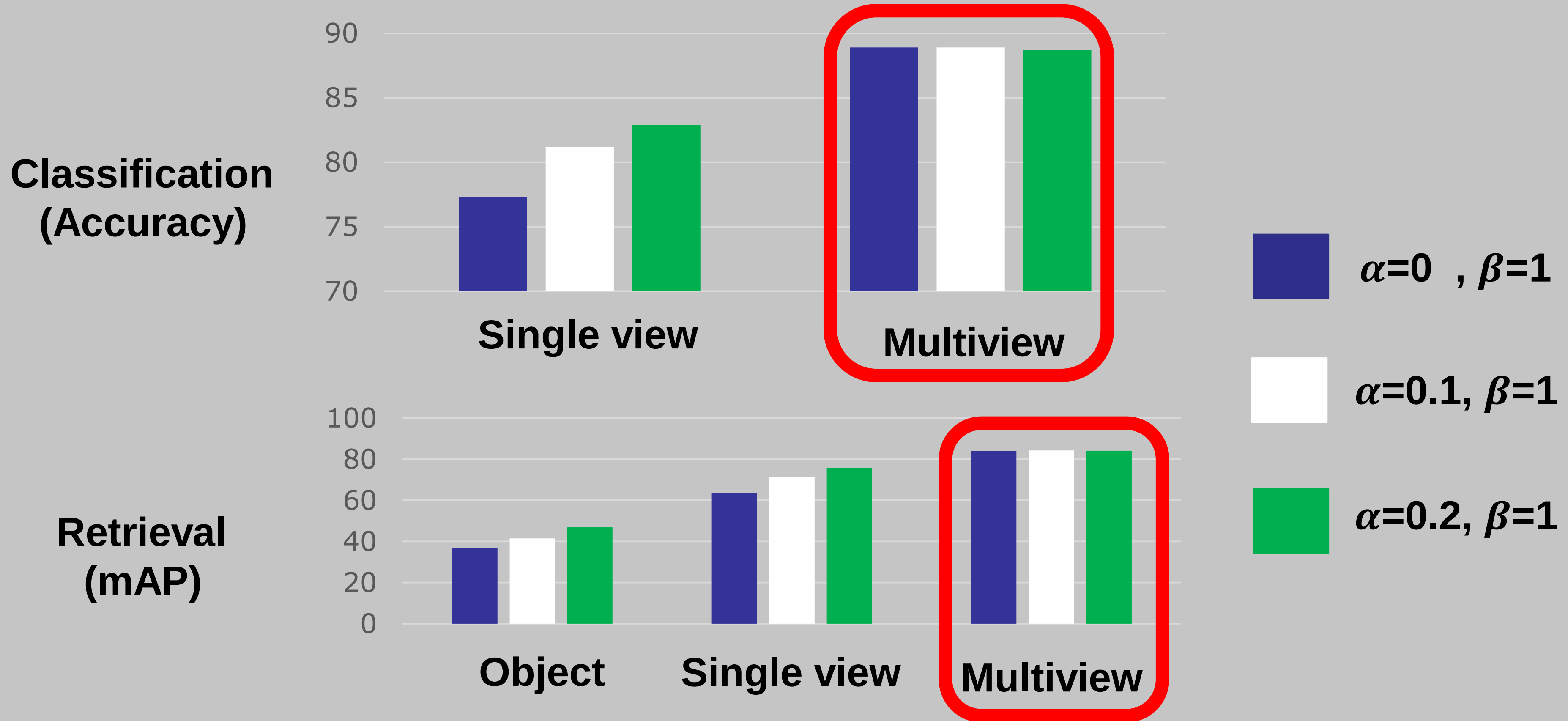
$$d^{inv}(v,s,c_y) = \alpha * d(v,s) + \beta * d(s,c_y)$$

- Ablation study of pose invariant distance
  - As α increase, results of single view tasks become better
  - As α increase, results of multiview tasks become worse



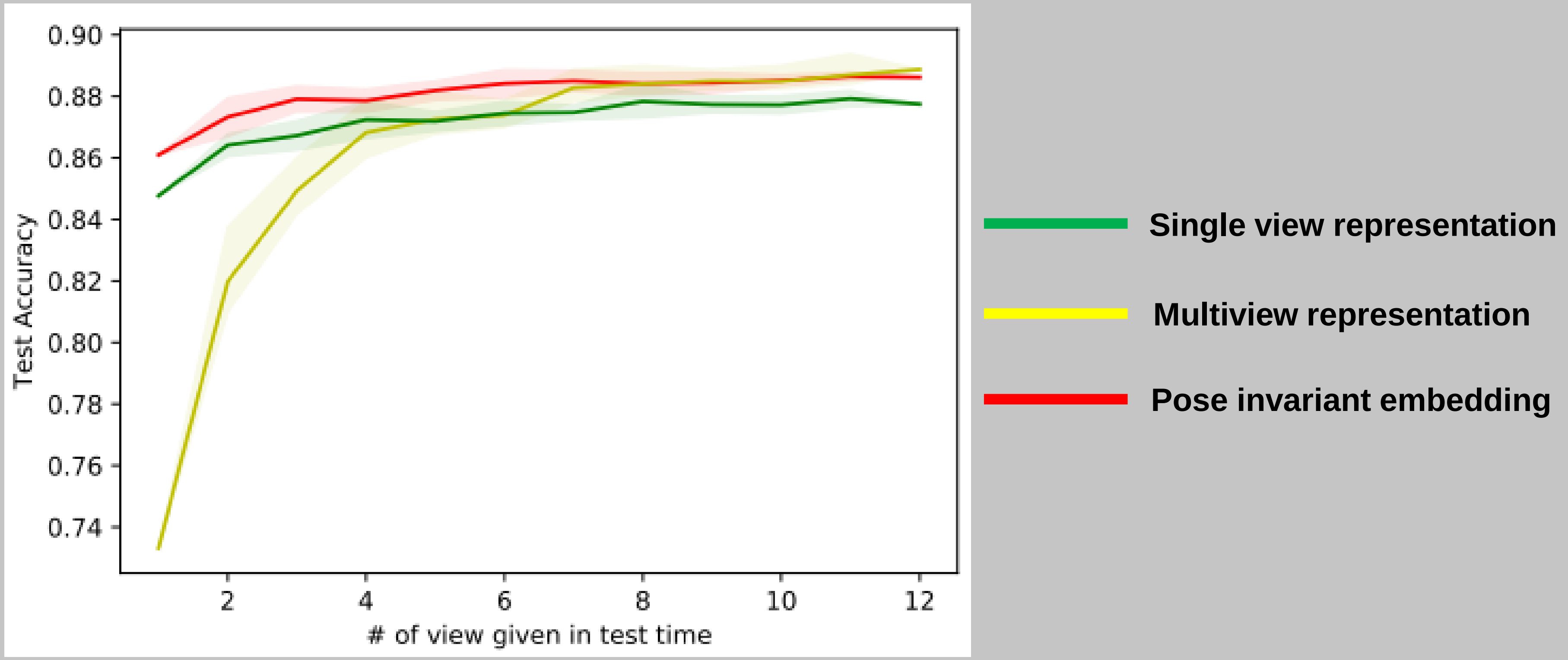
Experiment $d^{inv}(v, s, c_y) = \alpha * d(v, s) + \beta * d(s, c_y)$

- Ablation study of pose invariant distance
  - As α increase, results of single view tasks become better
  - As α increase, results of multiview tasks become worse



Experiment

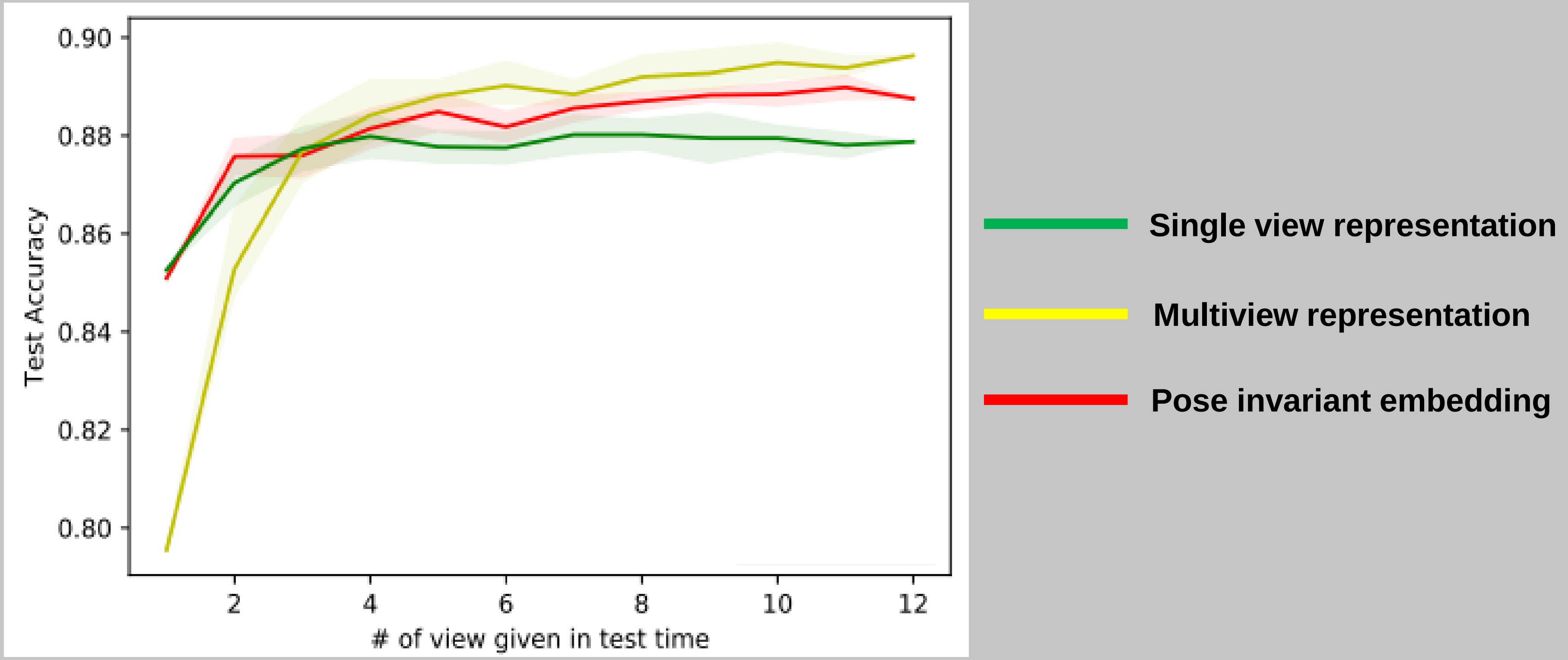
- Classification accuracy to number of views provided during inference time
  - PIE is more robust to the number of views provided



CNN based methods

Experiment

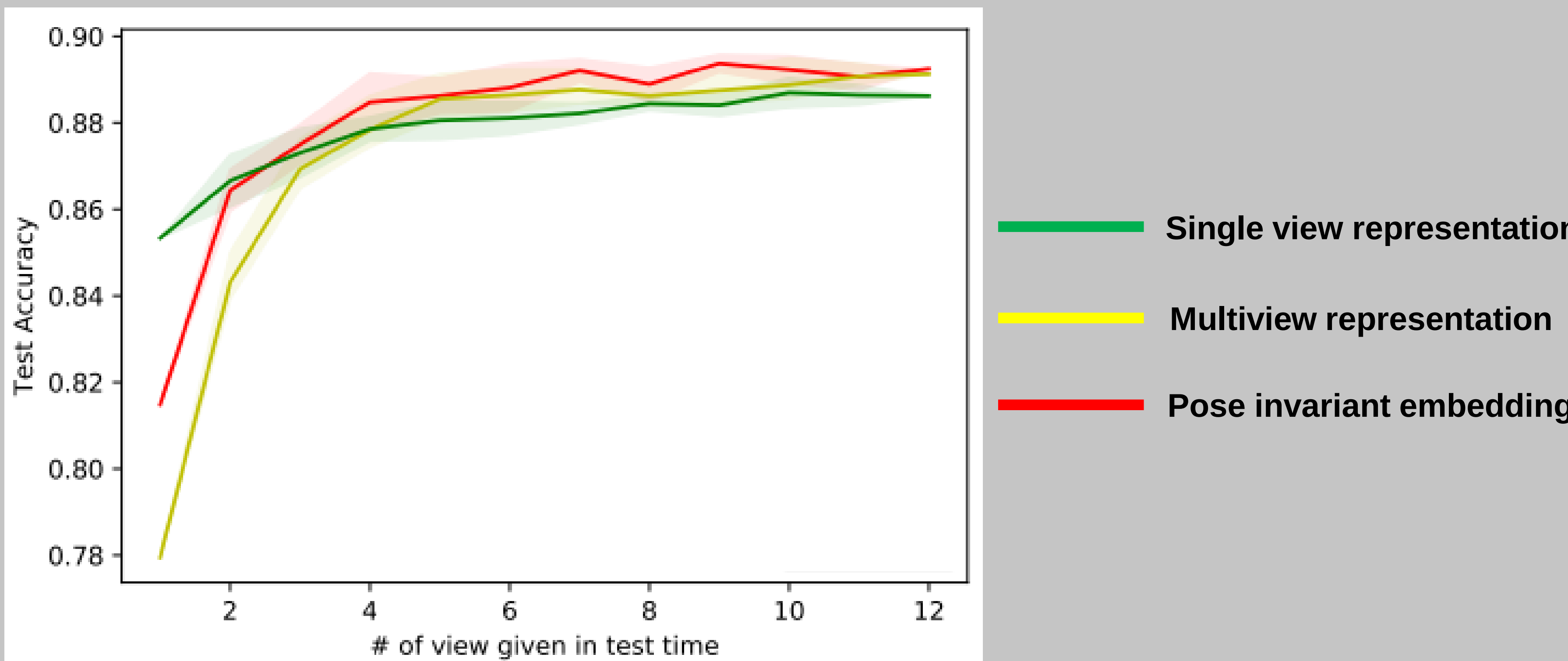
- Classification accuracy to number of views provided during inference time
  - PIE is more robust to the number of views provided



Proxy based methods

Experiment

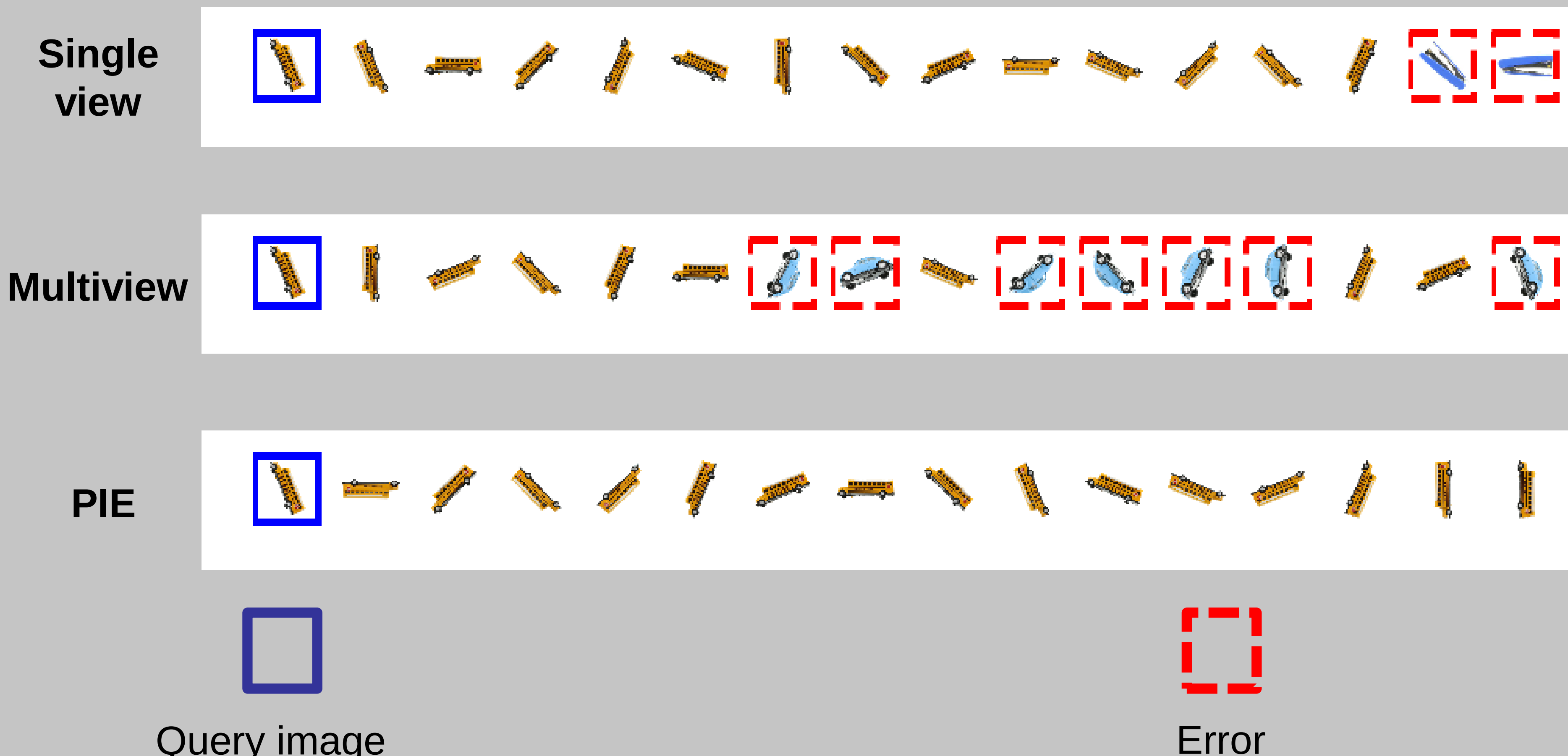
- Classification accuracy to number of views provided during inference time
  - PIE is more robust to the number of views provided



Triplet center based methods

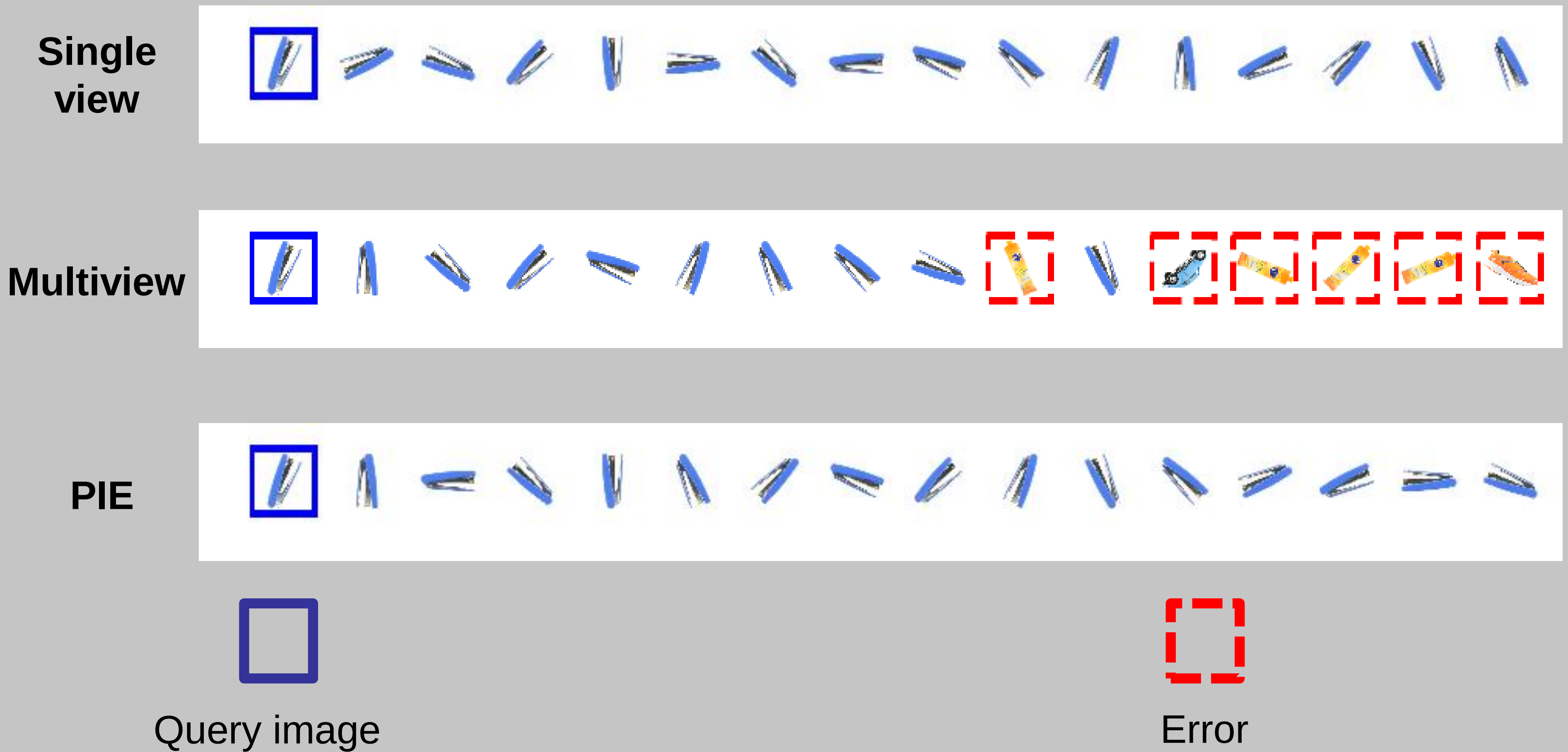
Experiment

- Retrieval results using CNN based embeddings on MIRO dataset



Experiment

- Retrieval results using CNN based embeddings on MIRO dataset



Conclusion

- Propose a taxonomy of embeddings that connects different metric learning approaches

Conclusion

- Propose a taxonomy of embeddings that connects different metric learning approaches
- Introduce pose invariant embedding (PIE) that can be applied to existing approaches

Conclusion

- Propose a taxonomy of embeddings that connects different metric learning approaches
- Introduce pose invariant embedding (PIE) that can be applied to existing approaches
- PIE is a hierarchical model
  - View to object
  - Object to class

Conclusion

- Propose a taxonomy of embeddings that connects different metric learning approaches
- Introduce pose invariant embedding (PIE) that can be applied to existing approaches
- PIE is a hierarchical model
  - View to object
  - Object to class
- Demonstrate the robustness of PIEs on
  - Classification and retrieval tasks
  - Single view and multiview inference

Conclusion

- Propose a taxonomy of embeddings that connects different metric learning approaches
- Introduce pose invariant embedding (PIE) that can be applied to existing approaches
- PIE is a hierarchical model
  - View to object
  - Object to class
- Demonstrate the robustness of PIEs on
  - Classification and retrieval tasks
  - Single view and multiview inference
- Propose a multiview dataset with real objects imaged under complex backgrounds

Publication

- **PIEs: Pose Invariant Embeddings**
  - **Chih-Hui Ho**, Pedro Morgado , Amir Persekian, Nuno Vasconcelos In, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, June 2019
- **Catastrophic Child 's Play: Easy to Perform, Hard to Defend Adversarial Attacks**
  - **Chih-Hui Ho***, Brandon Leung*, Erik Sandstrom, Yen Chang, Nuno Vasconcelos In, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, June 2019