Machine learning improvements for robotic applications in industrial context

case study of autonomous sorting

Doctor of Philosophy thesis defense

presented by **Joris Guérin**

on December 10 2018

Jury

M. Olivier PIETQUIN, Professeur des universités, Google Brain Paris	Rapporteur
M. Jean-Pierre GAZEAU, Ingénieur de recherche, Université de Poitiers	Rapporteur
M. Lorenzo NATALE, Maître de conférences, Instituto Italiano di Tecnologia	Examinateur
M. Ivan LAPTEV, Directeur de recherche, INRIA Paris	Examinateur
M. Byron BOOTS, Maître de conférences, Georgia Institute of Technology	Examinateur
M. Olivier GIBARU, Professeur des universités, Arts et Métiers ParisTech	Examinateur
M. Stéphane THIERY, Maître de conférences, Arts et Métiers ParisTech	Invité
M. Éric NYIRI, Maître de conférences, Arts et Métiers ParisTech	Invité

Table of content

Outline

Robots in industry

Current use

- Repeatable
- Precise
- Fast

Limitations

- Not adaptive
- Confined environment
- Large production batches

New context

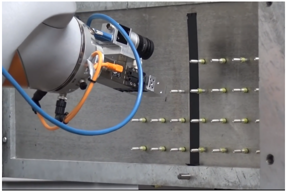
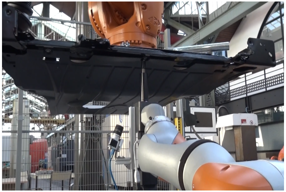
- Industry 4.0
- Mass customization
- Human-robot collaboration

(Lasi et al., 2014)

New goals

Robotic applications more flexible, more robust, easier to program

Tasks


Manufacturing, assembly, metrology, **Sorting**, ...

Technological bricks

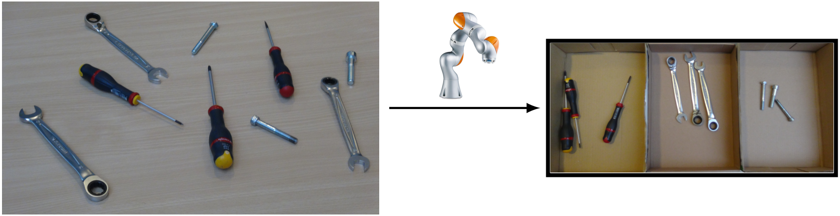
Scene understanding Object understanding Object localization

Grasping Trajectory generation Path planning Metrology

...

Unsupervised Robotic Sorting

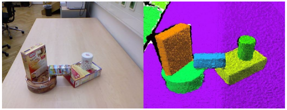
Robotic sorting



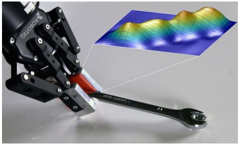
Improve flexibility

Required technical bricks

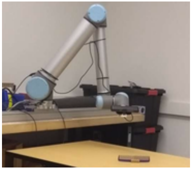
Scene segmentation
(Shi et al., 2016)



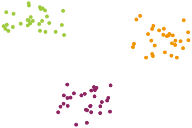
Grasping
(Bohg et al., 2014)



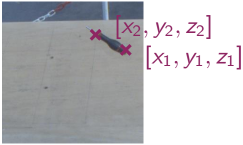
Data acquisition



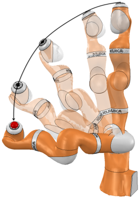
Data clustering



Object localization



Trajectory generation



Decision making pipeline

Gap-ratio Weigthed K-means

- Color and shape features
- Robust to lighting condition

➔ More expressive representation: **Images**

Proposed pipeline

Physical objects → **Image acquisition** → Digital images → **Image Clustering** → Classes

Image acquisition

- Multi-view sorting
- Optimal view selection

Image Clustering

- Feature extraction
- Deep ensemble clustering

Outline

1. Introduction
2. **Image clustering**
3. Image acquisition
4. Model independent trajectory learning
5. Conclusion

What is Image Clustering (IC)?

Other uses:

- Searching web image databases *(Avrithis et al., 2015)*,
- Medical image classification *(Wang et al., 2017)*,
- Video storyline reconstruction *(Kim et al., 2014)*, ...

Current approach

Images $\longrightarrow$ [Pretrained Deep CNN Feature Extractor] $\longrightarrow$ Features $\longrightarrow$ [Clustering Algorithm] $\longrightarrow$ Classes

Never studied

- Cross validation impossible
- Satisfying results
- Trained on the same dataset

Concentrate most research

- DEC,
- IDEC,
- JULE, ...

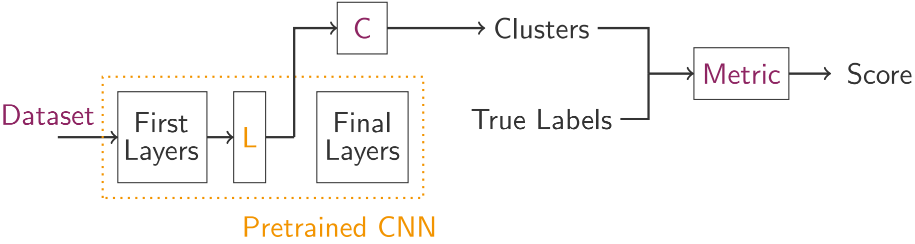
Many pretrained CNN available

-

Does it have an impact?

(Liu et al., 2016), (Wang et al., 2017), (Gong et al., 2015), (Hu et al., 2017)

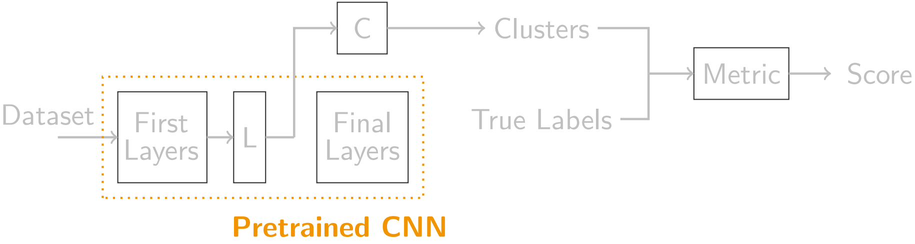
Experiments design



Questions

- Choice of CNN architecture?
- Choice of cutting layer?
- Relation to other design choices?

Experiments design



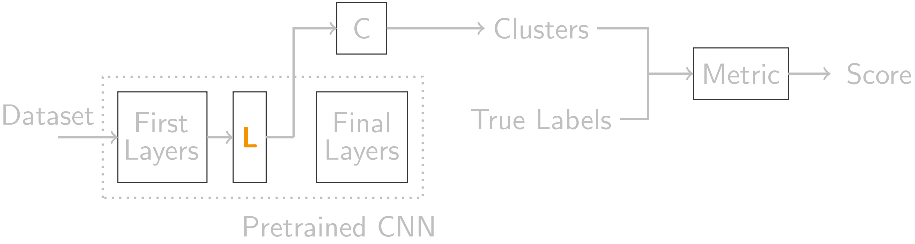
5 architectures

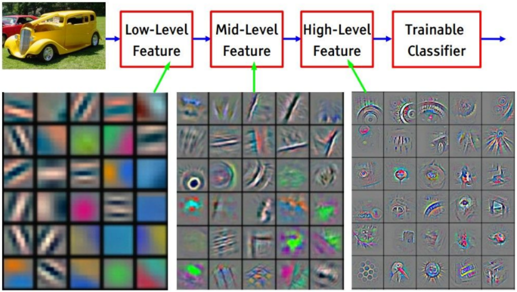
VGG16 VGG19 InceptionV3

Xception ResNet50

(Simonyan and Zisserman, 2014), (He et al., 2016), (Szegedy et al., 2016), (Chollet, 2016)

Experiments design

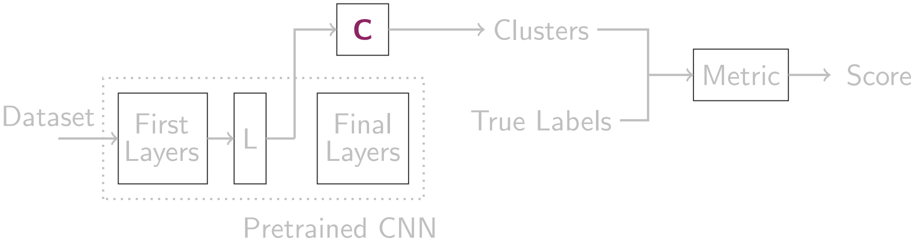




L1: End of conv block
L2: 2^{nd} layer before softmax
L3: Last layer before softmax

(Zeiler and Fergus, 2014)

Experiments design

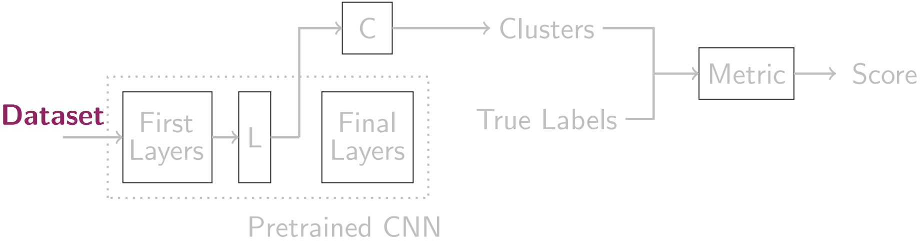


Standard algorithms

Centroid based	K-means
Connectivity based	Agglomerative

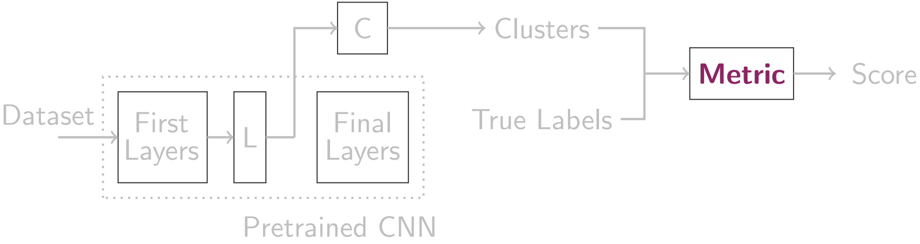
(Xu and Wunsch, 2005), (Arthur and Vassilvitskii, 2007), (Murtagh, 1983)

Experiments design



Task	Dataset	# images	# classes	Balanced
Natural object	VOC2007	2841	20	No
	COIL100	7200	100	Yes
Scene	Archi	4794	25	No
	MIT	15620	67	No
Fine-grained	Flowers	400	17	Yes
	Birds	2800	200	No
Face	UMist	564	20	Yes
	FEI	6033	200	Yes

Experiments design



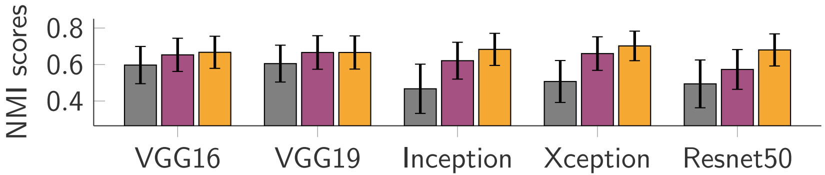
Supervised datasets → External validation metrics

Normalized Mutual Information $$NMI(Y, C) = \frac{2 \times I(Y,C)}{H(Y)+H(C)}$$

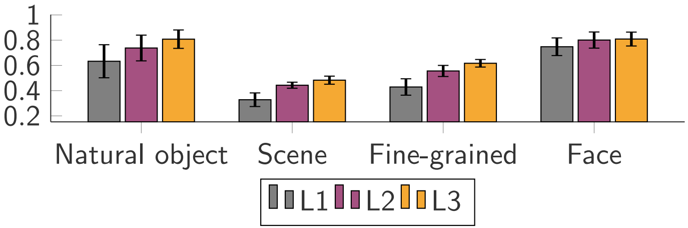
Cluster purity $$PUR(Y, C) = \frac{1}{N} \sum_{c \in C} \max_{y \in Y} | c \cap y |$$

Between 0 and 1 - Higher is better

Cutting layer's influence



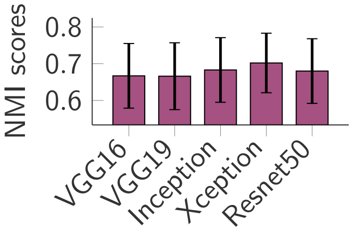
Layer-architecture interaction



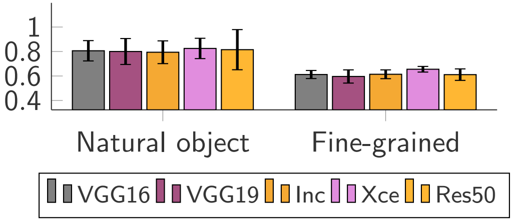
Layer-task interaction

(mean and std across other parameters)

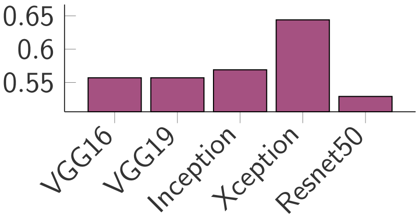
Architecture's influence



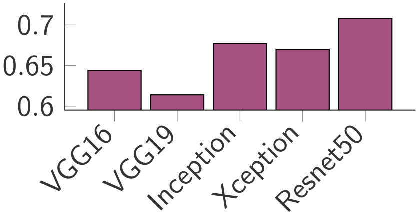
Overall results



Architecture-task interaction

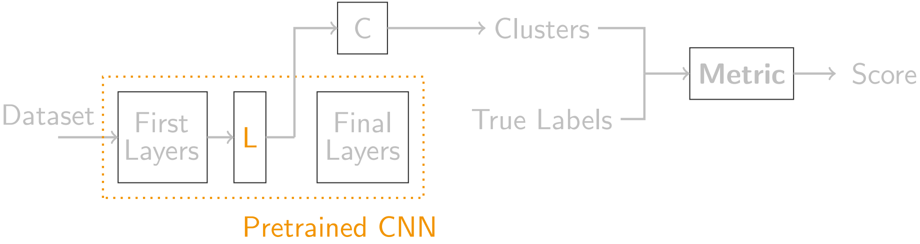


Birds - Agglomerative



Flowers - Agglomerative

Intermediate conclusion



Cutting layer choice

- Last layer before softmax
- For all datasets

CNN architecture choice

- No simple rules
- No cross validation

Could it be useful to combine them?

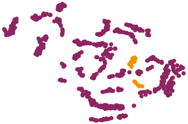
Complementarity of architectures? - Intuition

Pretrained on the same dataset
But
Different ways to solve a task

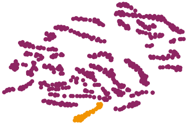
UMist face dataset

	NMI	PUR	FM	FM_{C_4}
InceptionResnet	**0.775**	**0.642**	**0.537**	0.442
VGG16	0.689	0.550	0.372	**0.653**
Densenet121	0.684	0.553	0.384	**0.700**

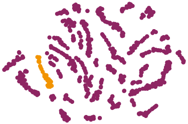
2d t-SNE visualization *(Maaten and Hinton, 2008)*



InceptionResnet



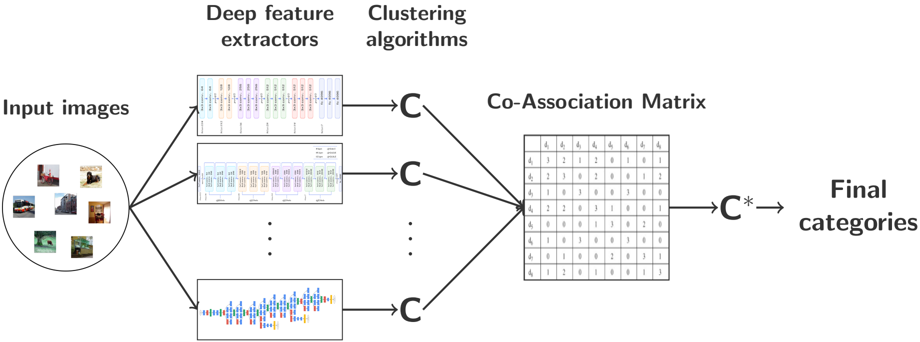
VGG16



Densenet121

First experiments

Ensemble method *(Vega-Pons and Ruiz-Shulcloper, 2011)*



Experiments

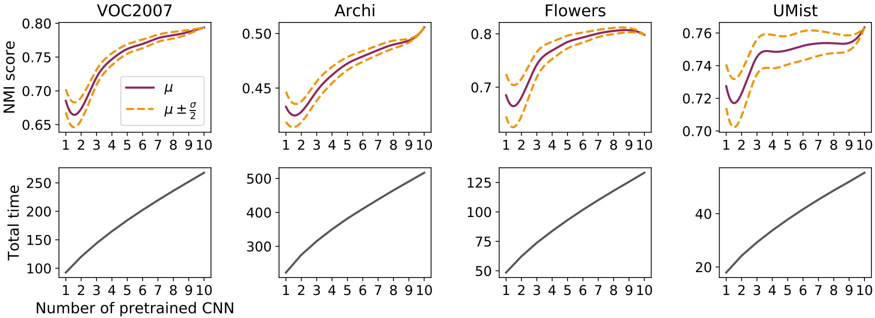
- 1 to 10 pretrained CNNs

  Densenet, Inception-resnet, NasNet

- 4 datasets from 4 tasks

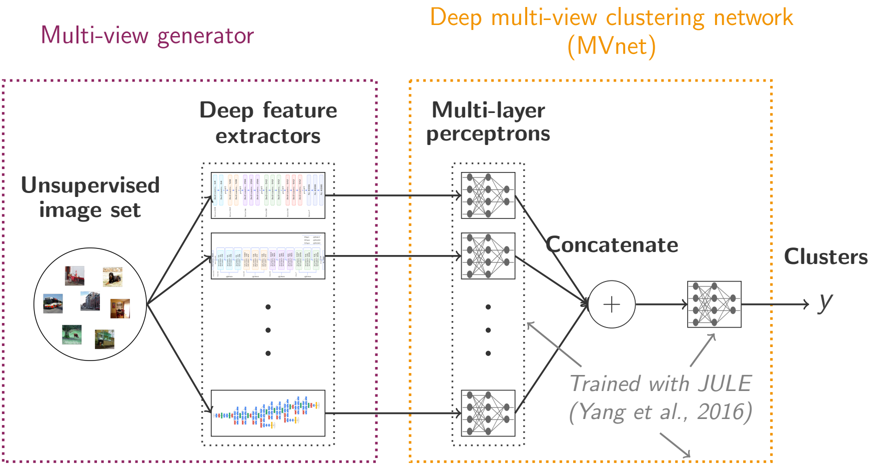
  VOC2007, Archi, Flowers, UMist

First results



Evolution of the NMI score and total time (in sec) for different numbers of pretrained CNN feature extractors.

Deep Multi-View Clustering



JULE

- Jointly learns **feature representation** and **cluster assignments**
- Adapted initialization for Multi-View data

Results

Evaluation: $\text{MIX}_\alpha = \alpha \text{ NMI} + (1 - \alpha) \text{ PUR}$

Average results across all 8 datasets

Method	$\text{MIX}_{0.5}$ score
Ours	**0.749**
Best Net + JULE	0.740
Worst Net + JULE	0.611
Leader Net + JULE	0.706
Best Net + Agg	0.712
MVEC + JULE	0.724
CC + JULE	0.703
MVEC + Agg	0.711

→ **State of the art results on most studied datasets**

Outline

1. Introduction

2. Image clustering

3. **Image acquisition**

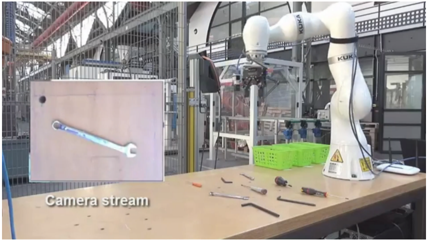
4. Model independent trajectory learning

5. Conclusion

Problem statement

Early implementation of URS

Problem statement



- Top-down perpendicular views
- Xception + Agglomerative

Lighting conditions
Scene layout
Objects poses

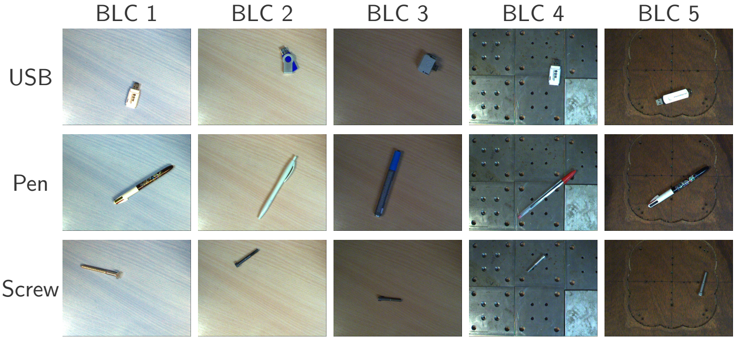
Unmastered task variations
Tunable action levers

Physical Objects → Image Acquisition → Image dataset → Image Clustering → Classes

Camera poses
Camera parameters

Robustness testing

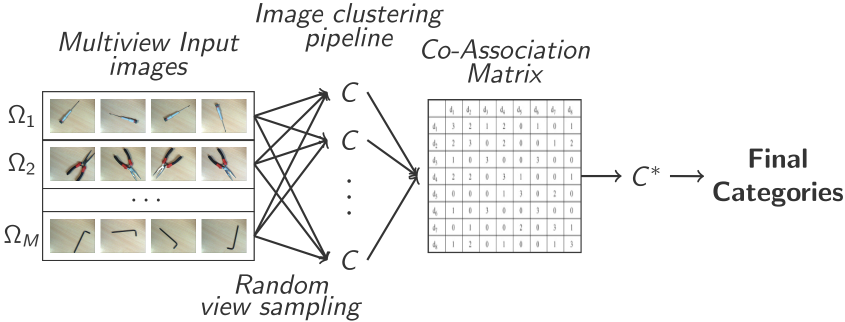
Robustness testing dataset



Artificially modified brightness

Multiple poses approach

Ensemble clustering pipeline

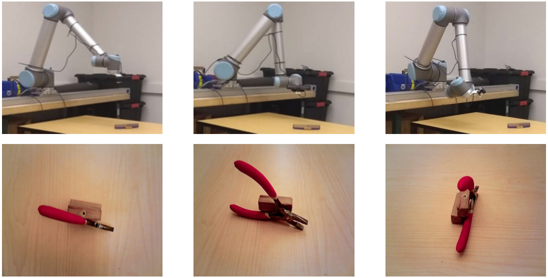


Results

		BLC1	BLC2 Dark+	Dark	Normal	Bright	Bright+	BLC3	BLC4	BLC5
NMI	MV	**0.95**	**0.91**	**1.00**	**1.00**	**0.96**	**0.84**	**0.95**	**0.84**	**0.95**
	SV	0.86	0.77	0.88	0.90	0.84	0.73	0.84	0.69	0.83

View selection problem

Importance of view selection:



Top view Good view Bad view

Objective:

Top perpendicular view

P_{cam}

Proposed camera pose

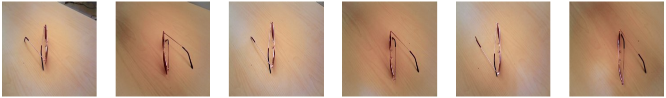
$\rightarrow$ $\boxed{\text{Model}}$ $\rightarrow$ $S(I)$

Semantic score

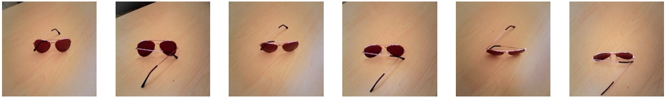
Building a large multi-view dataset

Example: 1 object in 1 pose

Top view



$(45°, 45°)$ $(45°, 225°)$ $(60°, 45°)$ $(60°, 225°)$ $(75°, 45°)$ $(75°, 225°)$



$(45°, 135°)$ $(45°, 315°)$ $(60°, 135°)$ $(60°, 315°)$ $(75°, 135°)$ $(75°, 315°)$

Views are parameterized by two angles θ and φ

Dataset statistics

# Classes	# Object/class (*total*)	# Poses/object (*total*)	# Views/pose (*total*)
29	4-6 (*144*)	3 (*432*)	17-22 (*9112*)

Fitting a "Clusterability score" to the images

Estimating the quality of an image for clustering

- Sample N clustering problem (3×10^7)
- For each clustering problem cp:
  - Compute the individual Fowlkes-Mallows index of each image:

(Fowlkes and Mallows, 1983)

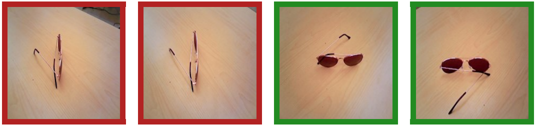
$$FMI^{i}_{cp} = \frac{TP_i}{\sqrt{(TP_i + FP_i)(TP_i + FN_i)}}$$

- Compute the Monte Carlo estimate of the clusterability index:

$$S(I) = \sum_{\mathrm{cp}} FMI^{I}_{\mathrm{cp}} / N^{I}_{\mathrm{cp}}$$

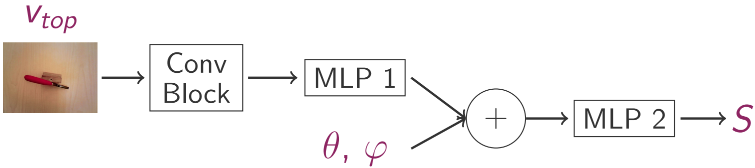
N^{I}_{cp}, *number of cp in which* I *is present*

Qualitative validation



Training a clusterability score predictor

Network architecture



Data splitting

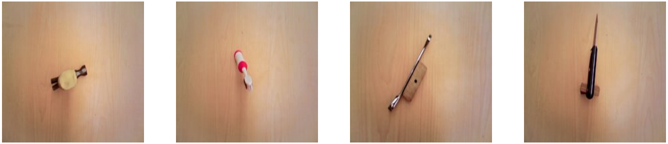
Clusterability index fitting	24 classes	
Neural network parameter selection	Training: 19	Testing: 5
Semantic View Predictor validation	5 classes	

Results

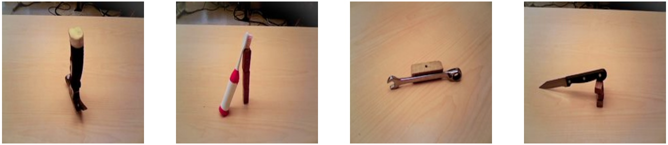
Quantitative results

		FM	NMI	PUR
	TOP	0.44	0.51	0.70
XCE_AGG	RAND	0.48	0.56	0.74
	SV-net	**0.55**	**0.63**	**0.78**

Qualitative evaluation



Example top views



Associated SV-net selections

Outline

Towards fully functional URS

Scene segmentation
(Shi et al., 2016)

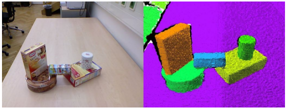


Image acquisition

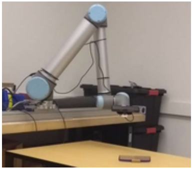
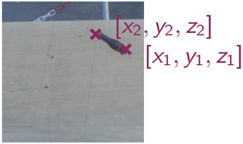
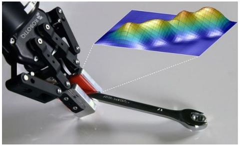


Image clustering

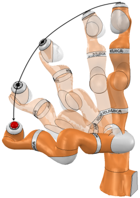
Object localization



Grasping
(Bohg et al., 2014)



Trajectory generation



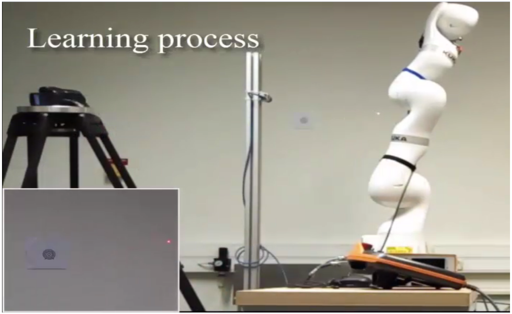
Model independent trajectory learning

Objectives

Build a trajectory learning framework which is

- Independent of the studied system
- Sample efficient

Practical example



- Angular position control
- Cartesian cost
- Independence of:
  - Robot geometry
  - Tool orientation
  - Robot location

Overview of the iLQR method

Trajectory definition

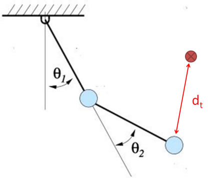
Initial state - - -> State ⟶ [Controller] ⟶ Action ⟶ [Dynamics] ⟶ State - - -> Final state

(State, Action) ⟶ Cost

x_t: State vector of the system

u_t: Control vector

l_t: Cost



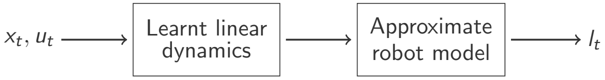
Optimization process

$$x_{t+1} = F_t(x_t, u_t) \quad \longleftarrow \quad 1^{st} \text{ order Taylor expansion}$$
$$l_t = L_t(x_t, u_t) \quad \longleftarrow \quad 2^{nd} \text{ order Taylor expansion}$$

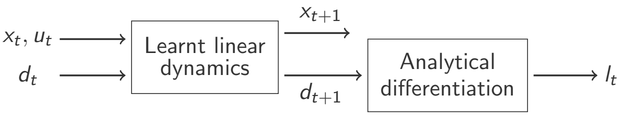
Use **dynamic programming** to optimize the controller to take actions that minimize the cost.

(Li et al., 2004)

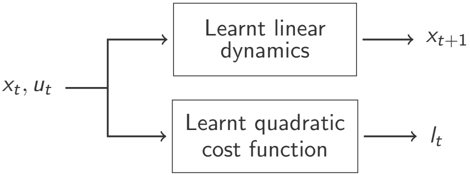
Compute Taylor expansion of the cost



(a) Using a model of the robot. *(Levine et al., 2014)*



(b) Including the distance d_t in the state representation. *(Levine et al., 2015)*



(c) Learning the quadratic approximation of the cost.

Practical example

Model independent trajectory learning - target reaching task

Outline

1. Introduction

2. Image clustering

3. Image acquisition

4. Model independent trajectory learning

5. **Conclusion**

Conclusion and open problems

Image Clustering

- Multiple pretrained CNNs improve results
- DMVC is state-of-the-art
  - ➔ Transfer to other tasks?
  - ➔ Study properties of training parameters?

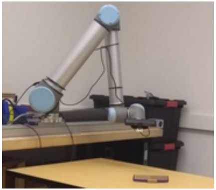
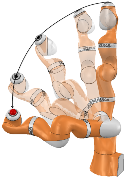


Image Acquisition

- Multiple views increase robustness
- Semantic view selection
  - ➔ Multiple view selection?



Trajectory learning

- Model independent method
  - ➔ Integrate in a global framework

Publications

Journal

- Guérin et al., "Unsupervised robotic sorting: Towards autonomous decision making robots", International Journal of Artificial Intelligence & Applications (IJAIA), March 2018

Conferences

- Guérin and Boots, "Improving Image Clustering with Multiple Pretrained CNN Feature Extractors", proceedings of BMVC 2018, Newcastle, UK. (29.9% acceptance)
- Guérin et al., "Semantically Meaningful View Selection", proceedings of IROS 2018, Madrid, Spain. (46.7% acceptance)
- Guérin et al., "CNN features are also great at unsupervised classification", proceedings of AIFU 2018, Melbourne, Australia.
- Guérin et al., "Automatic Construction of Real-World Datasets for 3D Object Localization using Two Cameras", proceedings of IECON 2018, Washington D.C., USA.
- Guérin et al., "Learning local trajectories for high precision robotic tasks: application to KUKA LBR iiwa Cartesian positioning", proceedings of IECON 2016, Florence, Italy
- Guérin et al., "Locally optimal control under unknown dynamics with learnt cost function: application to industrial robot positioning", Journal of Physics: Conference Series.
- Guérin et al., "Clustering for different scales of measurement: the gap-ratio weighted K-means algorithm", proceedings of AIAP 2017, Vienna, Austria

Machine learning improvements for robotic applications in industrial context

case study of autonomous sorting

Doctor of Philosophy thesis defense

presented by **Joris Guérin**

on December 10 2018

Jury

M. Olivier PIETQUIN, Professeur des universités, Google Brain Paris	Rapporteur
M. Jean-Pierre GAZEAU, Ingénieur de recherche, Université de Poitiers	Rapporteur
M. Lorenzo NATALE, Maître de conférences, Instituto Italiano di Tecnologia	Examinateur
M. Ivan LAPTEV, Directeur de recherche, INRIA Paris	Examinateur
M. Byron BOOTS, Maître de conférences, Georgia Institute of Technology	Examinateur
M. Olivier GIBARU, Professeur des universités, Arts et Métiers ParisTech	Examinateur
M. Stéphane THIERY, Maître de conférences, Arts et Métiers ParisTech	Invité
M. Éric NYIRI, Maître de conférences, Arts et Métiers ParisTech	Invité

t-Distributed Stochastic Neighbor Embedding

High dimensional space

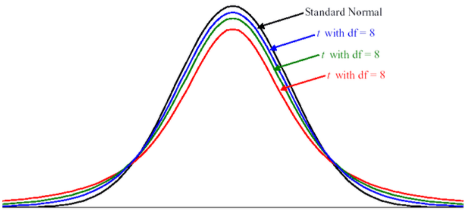
$$p_{ij} = \frac{\exp(-||x_i - x_j||^2 / 2\sigma_i^2)}{\sum_k \sum_{l \neq k} \exp -||x_k - x_l||^2 / 2\sigma_i^2}$$

Low dimensional space

$$q_{ij} = \frac{(1+||y_i - y_j||^2)^{-1}}{\sum_k \sum_{l \neq k} (1+||y_k - y_l||^2)^{-1}}$$

$$\underset{\text{points}}{\text{Minimize}} \quad KL(P||Q) = \sum_i \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Student's *t*-distribution

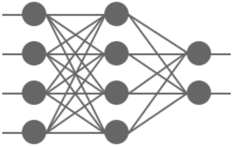


- Random initialization
- Gradient descent
- Preserves local structures
- Little dependant on tunable parameters

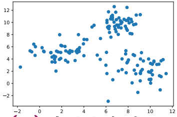
(Maaten and Hinton, 2008)

Joint Unsupervised Learning of Deep Representations and Image Clusters *(Yang et al., 2016)*

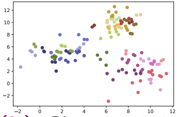
Unsupervised data in input space →



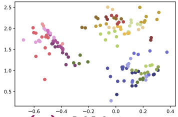
→ Unsupervised data in output space



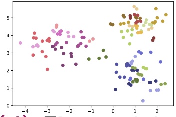
(a) Initial data ($N_c = 150$)



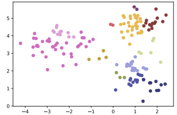
(b) Clusters init. ($N_c = 42$)



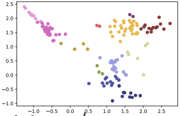
(c) NN init. ($N_c = 42$)



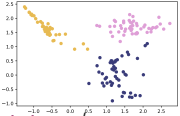
(d) First training ($N_c = 42$)



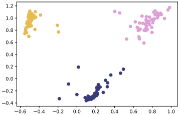
(e) First merging ($N_c = 9$)



(f) 2^{nd} training ($N_c = 9$)



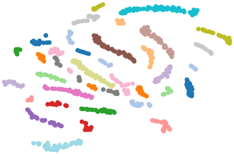
(g) 2^{nd} merging ($N_c = 3$)



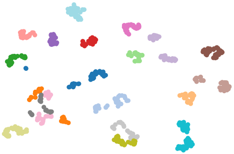
(h) Final training ($N_c = 3$)

New representation

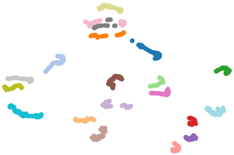
2d t-SNE visualization of the features extracted from the **UMist dataset** at different stages of the **DMVC framework**.



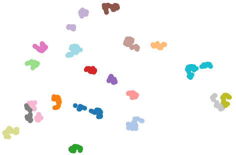
(a) Densenet169 features



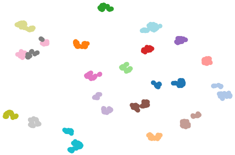
(b) D169 + JULE



(c) Concat

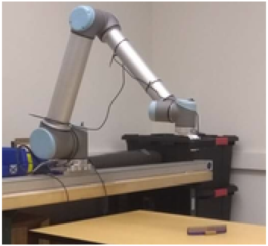


(d) $\text{MVnet}_{\text{fix}}$



(e) MVnet

View parameterization



Procedure:

- 3D camera
- Bounding box
- 75% of the image
- Parameterization