Object Re-Identification

Aymen Bernoussi

Othmane Farah

Hatim Meskine

Anass Ouali Alami

Saad Mdaa

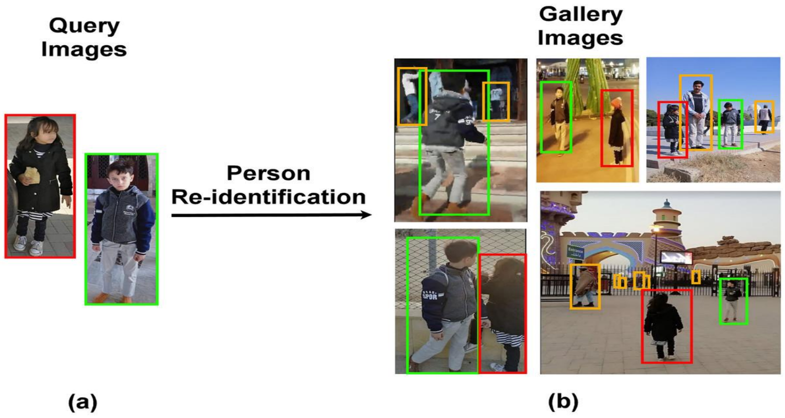
Nimble One

INP ENSEEIHT

Contents

I. Introduction

- Person ReId : associating images of the same person taken from different cameras or from the same camera in different occasions.



- Object ReId : consists of determining the exact instance of an object from an an initial set of informations (images)

Re-Identification : challenges

- Variations in visual appearance caused by different viewpoint from cameras
- Significant changes in human pose across time and space
- Different individuals with similar appearances
- Indoor environments: scenes are cluttered with many objects

Objective of the project

- The objective of our project is to re-identify several instances of objects from several views in an indoor environment.
- Nimble one is currently building Aru, a new robotic assistant for homes and industries.
- Object Reid is one of the tasks assigned to it

Aru : the robotic assistant designed by NimbleOne

https://nimbleone.io/

Literature review

Feature extraction

- Computer vision approach :
  - ORB
  - SIFT
  - SURF
- Deep Learning approach :
  - Convolutional neural networks

List of datasets for Re-Identification:

- **3Rscan : 1482** 3D reconstructions of **478** indoor environments with annotated object boxes.

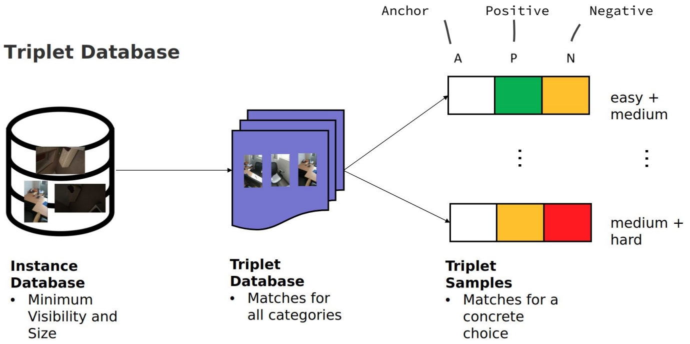
- ScanNet : RGB-D video dataset containing 2.5 million views in more than 1500 scans, annotated with 3D camera poses, surface reconstructions, and instance-level semantic segmentations.

- Market 1501 : dataset for person re-identification. It contains 1501 identities which are captured by six different cameras.

2. 3RScan dataset

- Large scale, real-world dataset : **1482** 3D reconstructions of **478** indoor environments with annotated object boxes.

- Triplet Dataset Toolkit with **Pytorch**



https://github.com/WaldJohannaU/3RScan

Generating instances

The following tools were used from the 3RScan repositories described before:

- **rio_renderer**: rendering all artifacts (bounding-box file; rendered rgb, label, instance and depth image; occlusion scores for each object) for each frame in a scan
- **FrameFilter**: generate a file 2Dinstances.txt. This file is a list of all object instances from all frames in all scans that fulfill a minimum amount of filtering options.

Load instances

- utils.py : file containing all functions to load instances.
- For each instance : return a python dictionary
    - image : tensor
    - bounding box : dictionary
    - label : name of the object
    - instance_id : id of the object
    - reference : reference of the room
    - scan": directory of the instance
    - frame_nr: the number of the frame

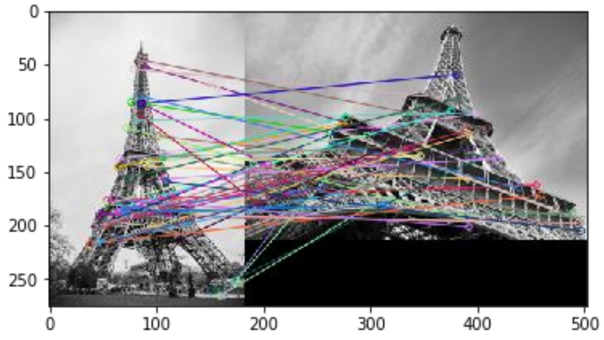
Load triplets

- triplet_utils.py : load triplets based on instances
- for each instance return a dictionary:
    - anchor : a dictionary of our image
    - pos : a positive image (corresponding to the same object in the same room)
    - neg : a list of negatives instances (we chose a list of size 1 to make the training easier)
- We can also filter the triplets based on a visibility score

Features matching with SIFT (Scale-invariant feature transform)

- **SIFT** (1999) is a feature detection algorithm to detect and describe local features of image.
    - invariant to rotation, affine transformations and intensity.

- Computationally demanding.
- Not effective for low powered devices.
- Not suitable for real time applications.
- not free software.

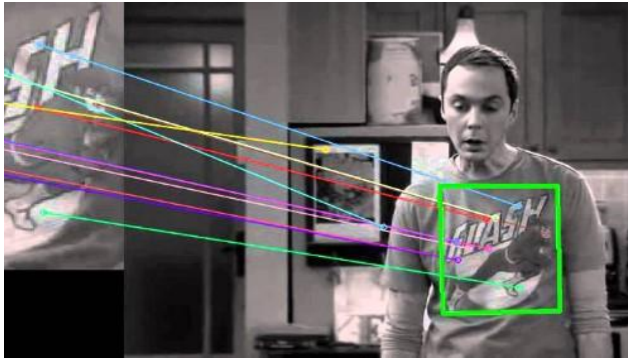


https://cdn.analyticsvidhya.com/wp-content/uploads/2019/09/index_61.png

Features matching with SURF (Speeded Up Robust Features)

- **SURF** (2006) is a feature detection algorithm inspired by SIFT.
  - based on 2D Haar wavelet response sums and makes efficient use of integral images.

- several times faster than SIFT.
- less accurate than SIFT.
- not free software.

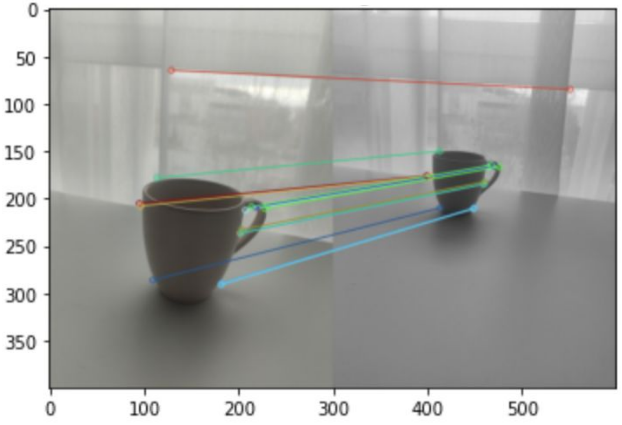


https://miro.medium.com/max/700/0*5tH4g-DWevzcs_8Y.jpg

Features matching with ORB (Oriented FAST and Rotated BRIEF)

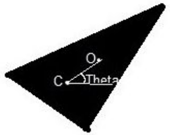
- **ORB** (2011) an efficient and viable alternative to SIFT and SURF.
  - conceived mainly because SIFT and SURF are patented algorithms.

- Detects features as well as SIFT (and better than SURF)
- Nearly two orders of magnitude faster than both.
- free software.

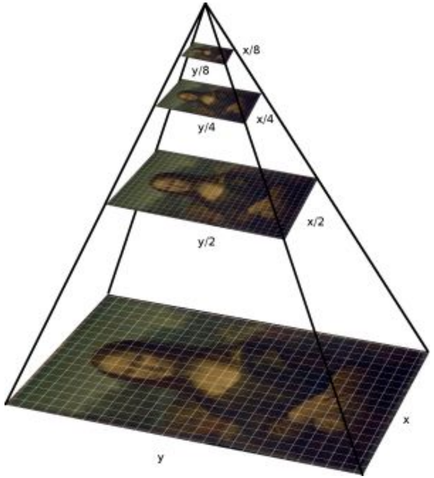


Features matching with ORB (Oriented FAST and Rotated BRIEF)

- ORB is the combination of **FAST** : a keypoint detector and **BRIEF** descriptor with some modifications to improve the performance.
- **FAST** is not rotation independent ⟶ **oFAST**



- **FAST** is not scale invariant ⟶ Multi-scale Image Pyramid



FAST : Features from Accelerated and Segments Test
BRIEF : Binary Robust Independent Elementary Features

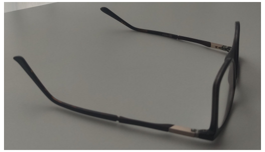
Features matching with ORB (Oriented FAST and Rotated BRIEF)

- First Test
  - **5** objects
  - Different angles
  - Uniform background.



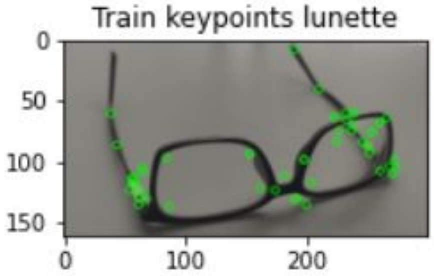


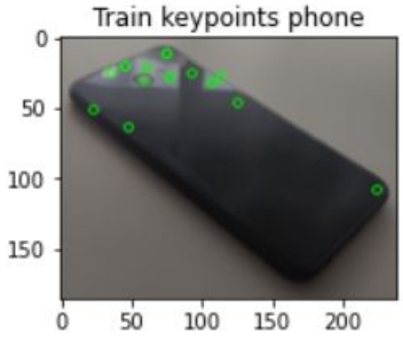


Features matching with ORB (Oriented FAST and Rotated BRIEF)

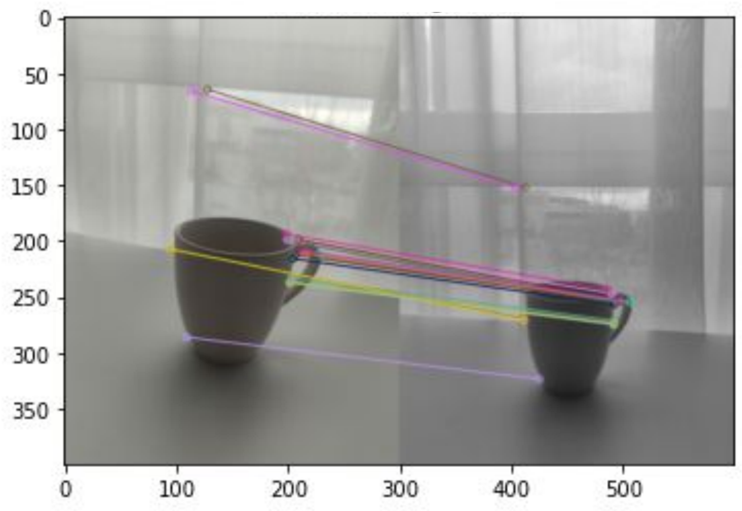
- Results & Remarks
  - Feature detection and matching is much better for objects with well-defined corners.

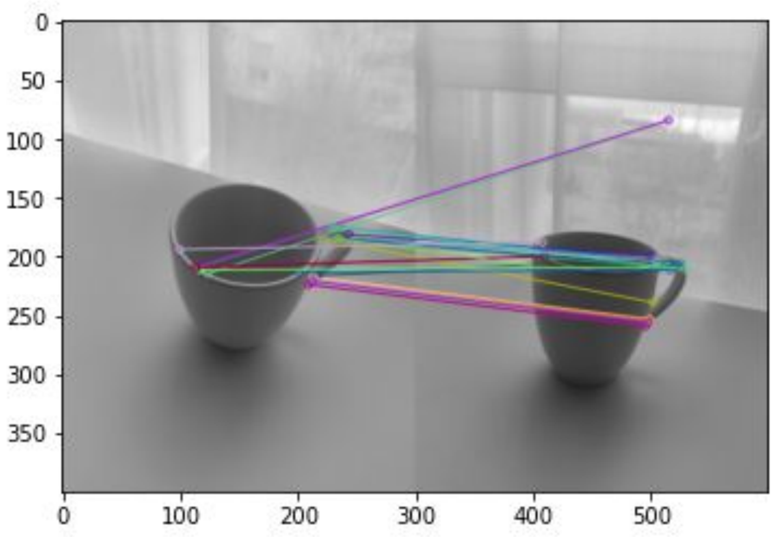




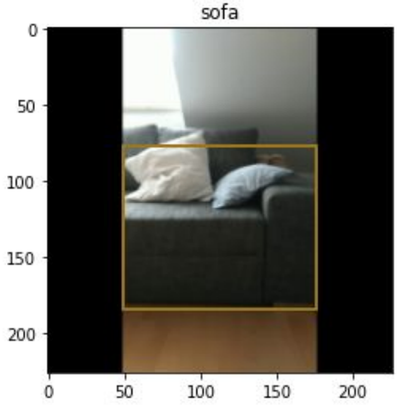
Features matching with ORB (Oriented FAST and Rotated BRIEF)

- Results & Remarks
  - The background interference.





ORB Results : 3RScan



- Best results without cropping (Fixed objects).

Cropped dataset

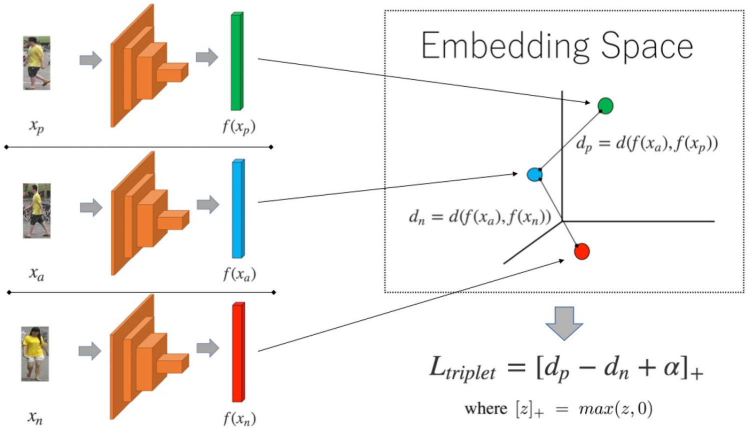
Rank	1	5	20
Object re-id (10% Training Data)	49.80%	69.49%	84.70%
Class re-id (10% Training Data)	51.76%	76.14%	88.92%
Reference re-id (10% Training Data)	71.84%	89.05%	99.22%

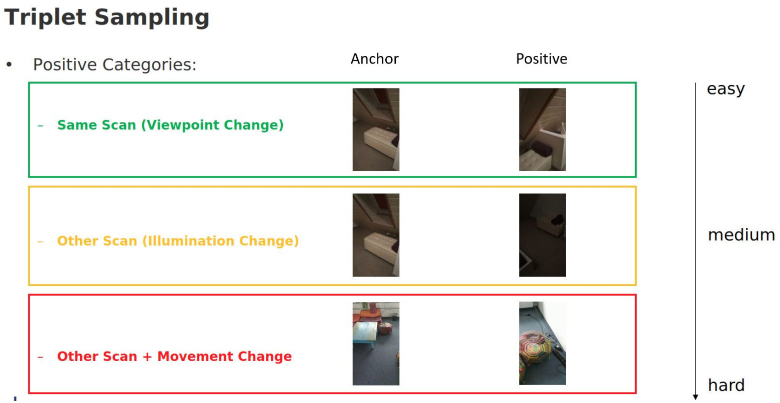
Uncropped dataset

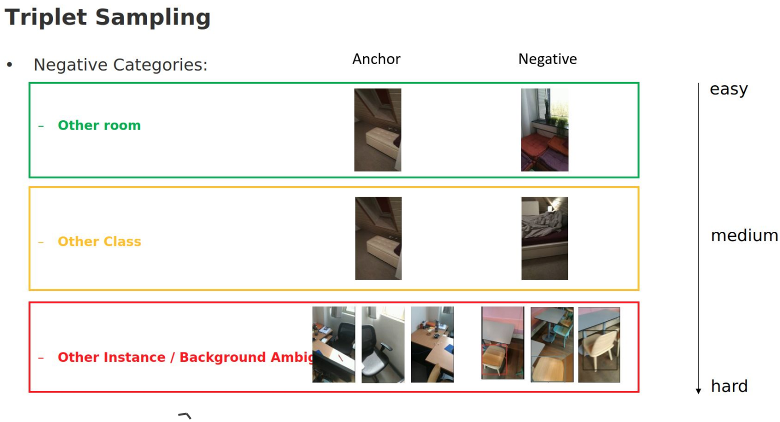
Rank	1	5	20
Object re-id (10% Training Data)	63.14%	75.39%	87.54%
Class re-id (10% Training Data)	64.41%	81.86%	93.43%
Reference re-id (10% Training Data)	76.37%	90.39%	99.02%

Deep Learning approach:

- We work with triplet based convolutional networks
- A triplet is composed of an anchor image of the object of interest, a positive image and a negative one
- We can control the triplet sampling procedure



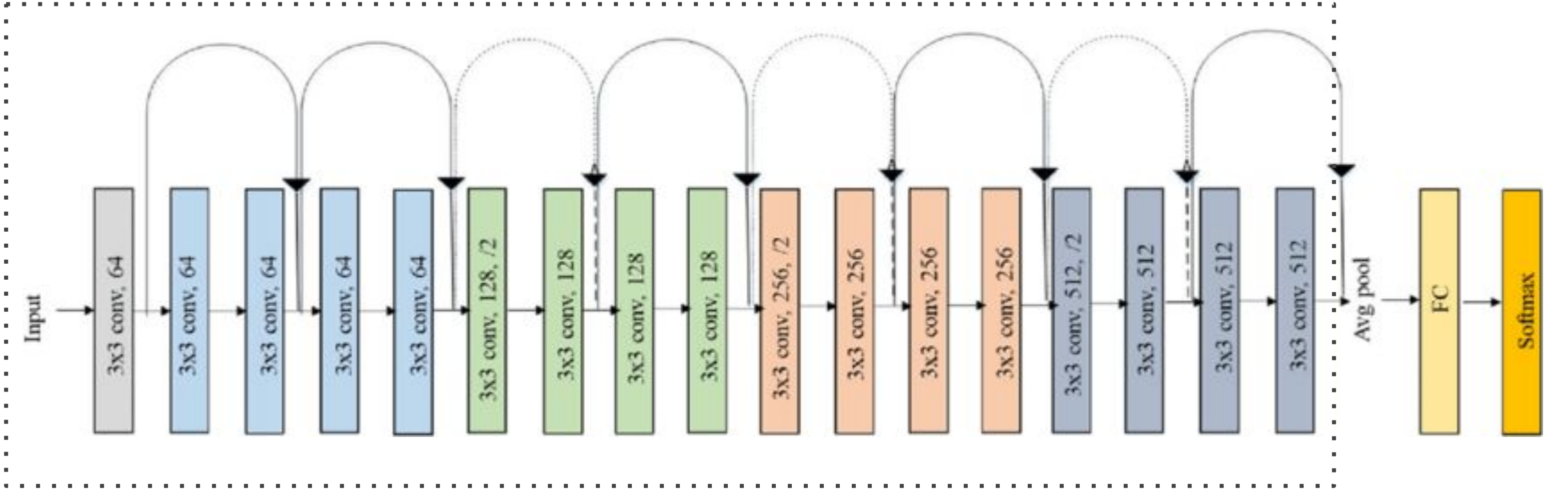




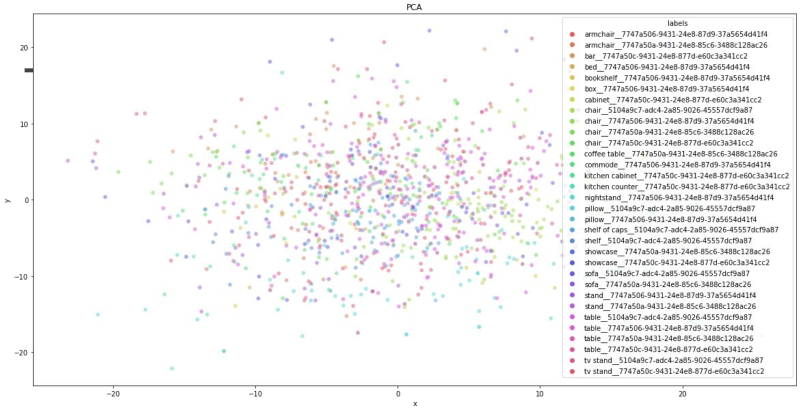
Deep Learning approach : first model

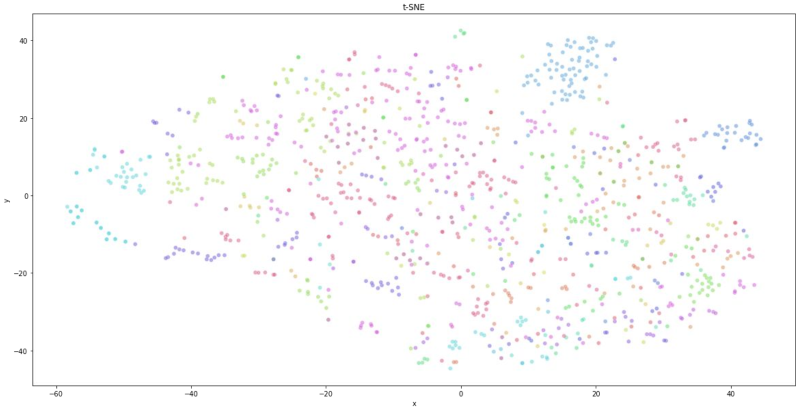
Model architecture:

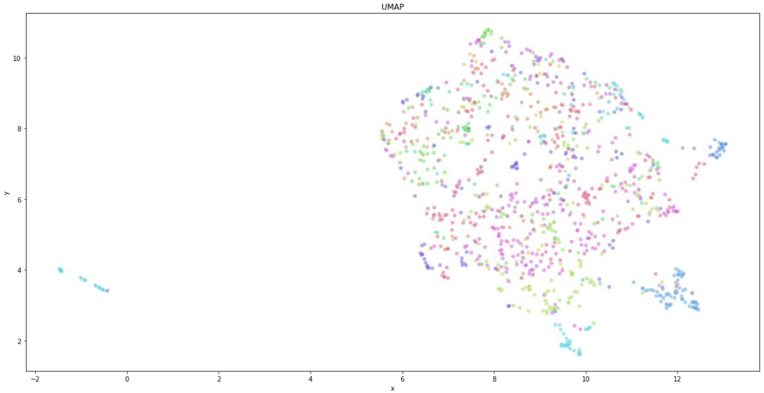
- Uses the first layers of a pretrained ResNet-18
- 3 extra conv layers to train with our triplets



Deep Learning approach : results







Deep Learning approach : results

k	1	2	5	10	20	50
Rank k	44.19%	55.61%	69.19%	76.18%	84.15%	94.78%

k	1	2	5	10	20	50
mAP	44.19%	49.90%	50.68%	47.18%	41.88%	34.47%

- Performs worse than ORB (49.80% for cropped images and 63.14% for uncropped ones)!
  - Model not complex enough - Similar results even with a ResNet-50 based architecture
  - Not enough triplets to train the additional layers

Deep Learning approach : another model

- Use with the pretrained layers of the ResNet-18 only
- Work with bounding box images
- Additional layers for resizing the bounding boxes and computing the max value per activation map in the spatial domain

k	1	2	5	10	20	50
Rank k	54.53%	62.79%	73.72%	82.18%	87.79%	93.99%

k	1	2	5	10	20	50
mAP	54.52%	58.66%	59.09%	55.04%	48.47%	39.54%

Deep Learning approach: using pretrained ResNets

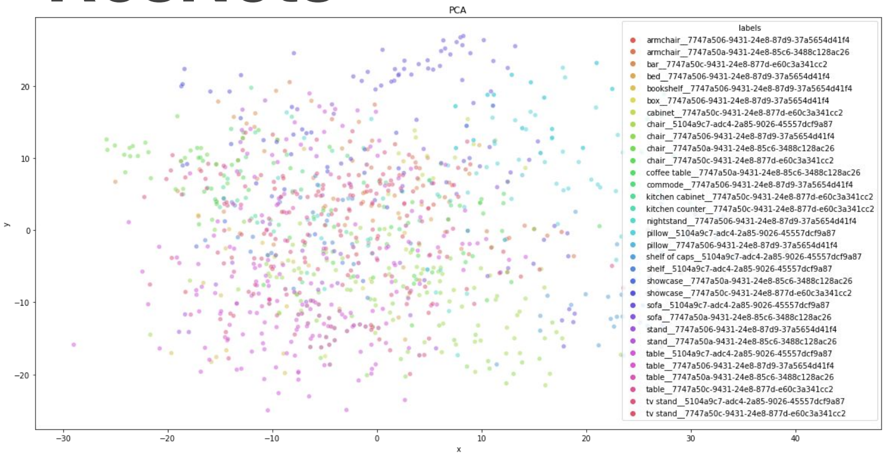
Cropped dataset

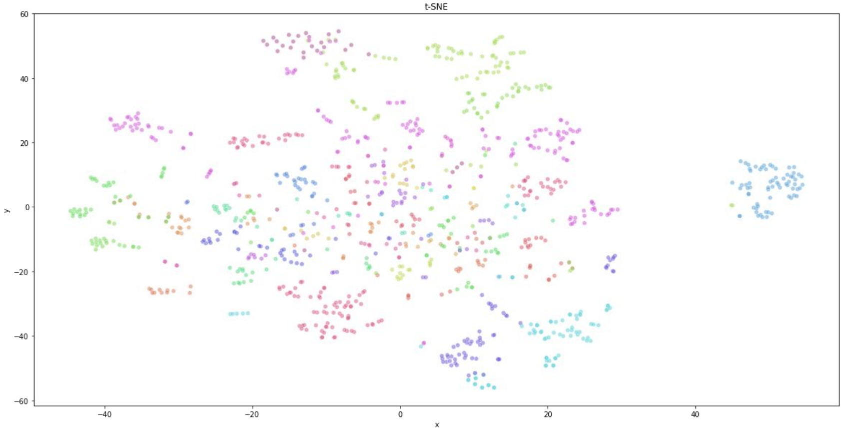
Models	ResNet-18	ResNet-34	ResNet-50	WideResNet-50
Rank 1	58.07%	64.37%	64.47%	62.30%

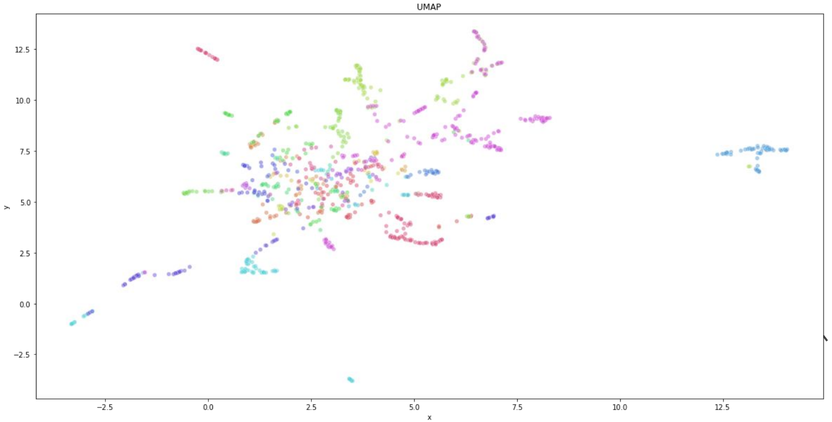
Uncropped dataset

Models	ResNet-18	ResNet-34	ResNet-50	WideResNet-50
Rank 1	67.12%	65.64%	66.14%	64.04%

Deep Learning approach: using pretrained ResNets







Deep Learning VS ORB

- Deep Learning models better than ORB
  - Overall better results
  - Easier to extract features from the images
- Some problems that we still have with Deep Learning models
  - Still needs to be tested for moving objects
  - We can clearly see that models that were trained on big datasets have a better performance than the same models even with some added layers.

ReRanking

- Compute the KNN with distance that is known.
- Compute the KNN with a new distance.

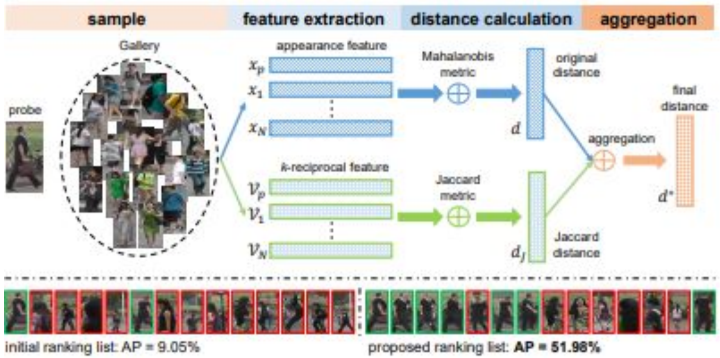
$$d_J(p, g_i) = 1 - \frac{card(R^*(p,k) \cap R^*(g_i,k))}{card(R^*(p,k) \cup R^*(g_i,k))}$$

with R*(p,k) the k-reciprocal neighbors of p.

- Give a final list that takes into account the two distances.

$$d^*(p, g_i) = (1-\lambda)d_J(p, g_i) + \lambda d(p, g_i)$$

with λ the penalty factor.



ReRanking Results : 3RScan

Uncropped dataset

Rank	1	5	20
Object re-id (10% Training Data)	63.14%	75.39%	87.54%
Class re-id (10% Training Data)	64.41%	81.86%	93.43%
Reference re-id (10% Training Data)	76.37%	90.39%	99.02%

Uncropped dataset (with re-ranking)

Rank	1	5	20
Object re-id (10% Training Data)	60.10%	75.78%	87.43%
Class re-id (10% Training Data)	61.47%	81.96%	93.52%
Reference re-id (10% Training Data)	75.19%	91.27%	99.06%

Conclusion

- In general, deep learning methods are better (we always have features for our images) with a score as least as equal to ORB
- Paths to explore :
    - Training on more data (only 150 directories in our work)
    - Using ReRanking for deep learning (new distance for proximity between images)
    - Using some machine learning algorithms (other than KNN : XGBoost,...) for ReID