A system for Furniture Analysis and Home Scene Recognition

Computer Vision and Cognitive System

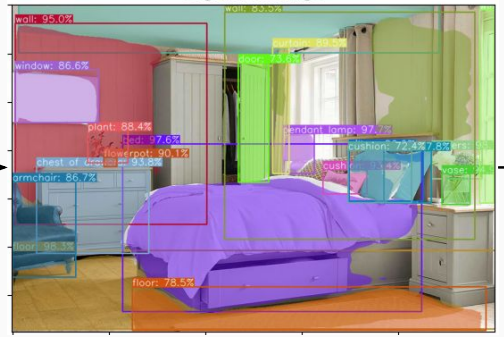
Cartella Giuseppe
Marchesini Kevin
Sarto Sara

Academic Year 2020/2021

PIPELINE OVERVIEW

1

INSTANCE SEGMENTATION
MASK R-CNN



2

3.1

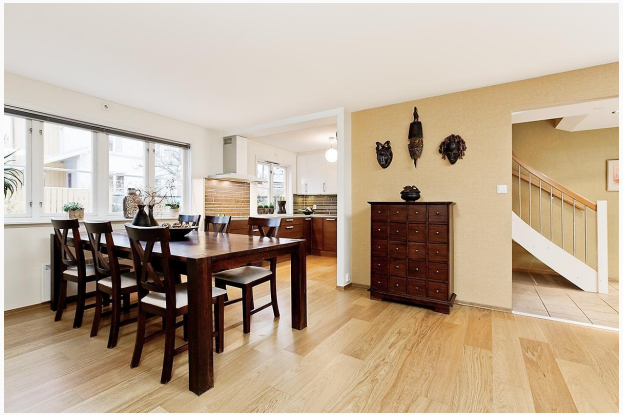
FURNITURE RETRIEVAL
DIFFERENT METHODS

3.2

ROOM CLASSIFICATION
DIFFERENT METHODS

Datasets - ADE20K

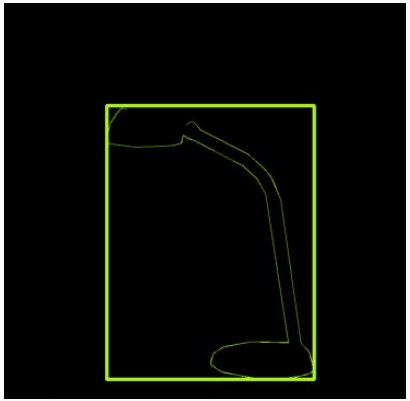
- Fully annotated dataset: bounding boxes and masks for each object in the image
- Filtered to obtain 10 different room categories for a total of 5297 examples
- Used for Mask-RCNN training



Datasets - Retrieval

- Google images merged with a dataset from Kaggle
- Focus on 6 object classes.
- 1938 total images.
- Custom annotation with indications about bounding box and label

Retrieval - Bounding box



1 Input Image → 2 Canny algorithm → 3 Polygon with max area

1 Input Image → 2 Mask R-CNN pre-trained

Retrieval Dataset - GRABCUT RESULTS

Example of Grabcut result after having obtained the bounding box from Mask R-CNN. The class “sofa” is a COCO category.

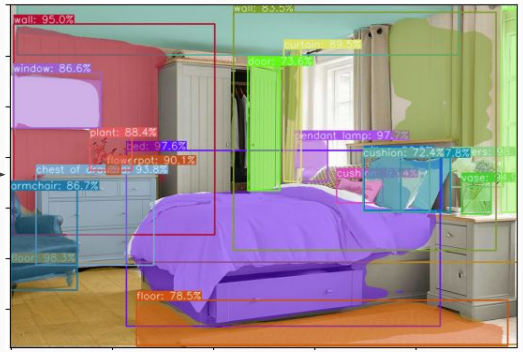
Example of Grabcut result after having obtained the bounding box from Canny algorithm. The class “armchair” is not a COCO category.

1- INSTANCE SEGMENTATION

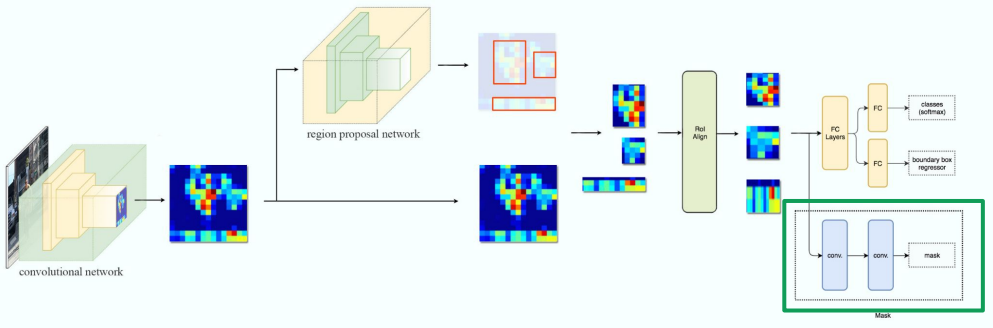
1

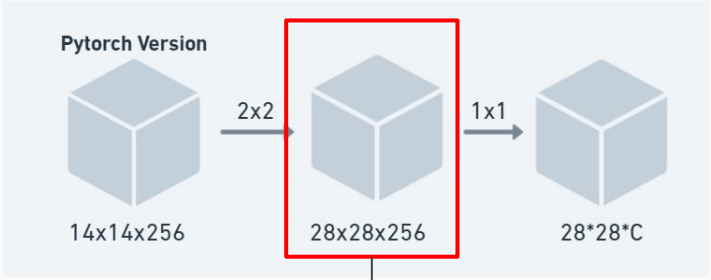
INSTANCE SEGMENTATION
MASK R-CNN

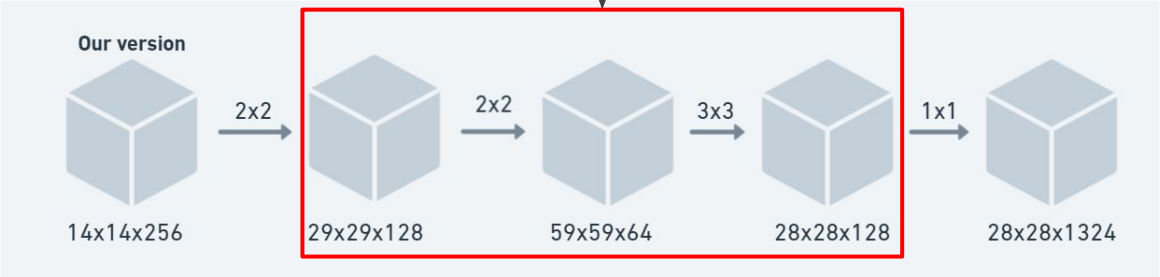
2



Instance Segmentation - Our changes





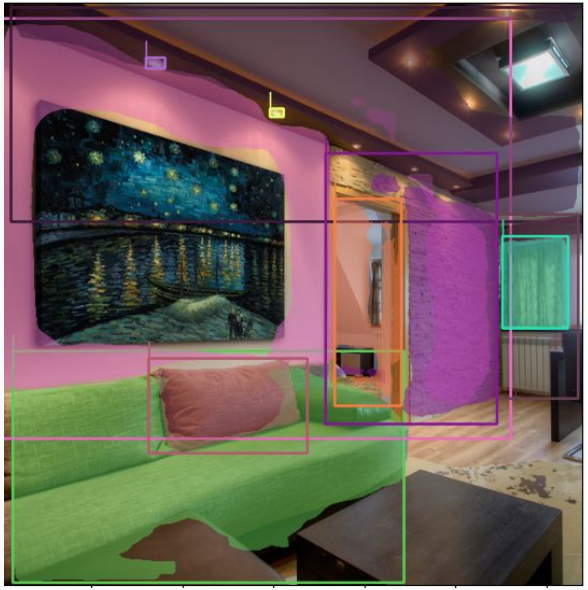


Transpose convolutions

Convolutions

Instance Segmentation - Results

Classical Mask R-CNN

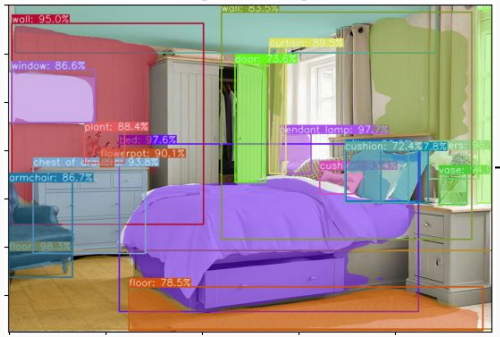


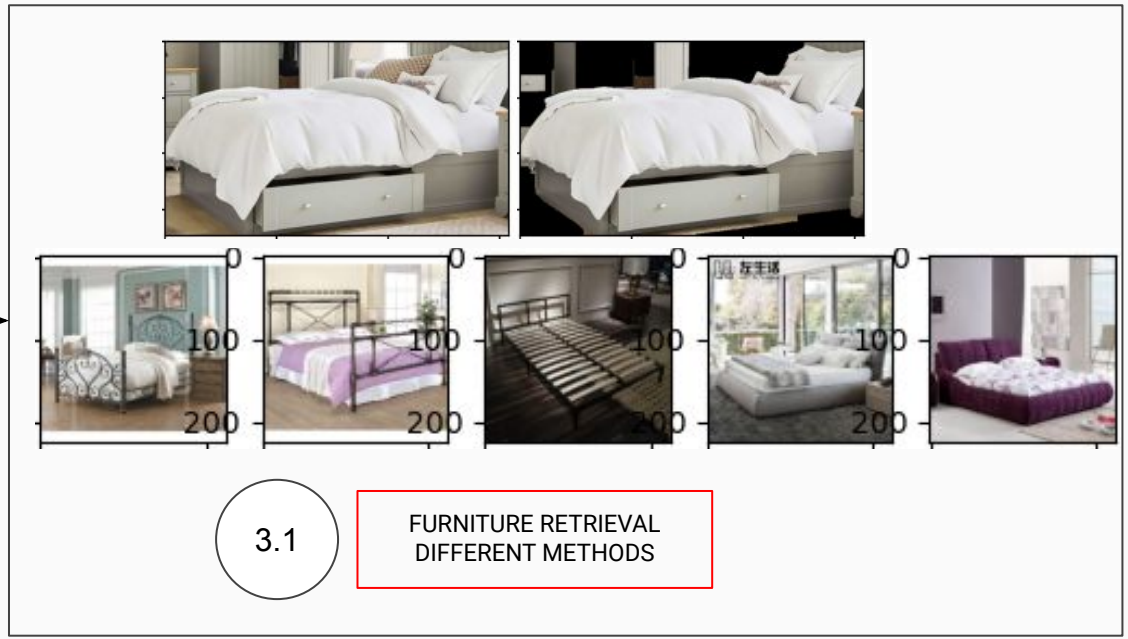
Modified Mask R-CNN

	Classical Mask R-CNN	Modified Mask R-CNN
AP @ IoU=0.5	57.5%	61%

2- FURNITURE RETRIEVAL

2





Furniture Retrieval - Approaches

Compared 3 different methods.
The idea is to compute the feature vector of an input image, compare it with the ones of the handmade dataset of single objects and retrieve the most similar ones.

Query image (bounding box returned by the network) processing:

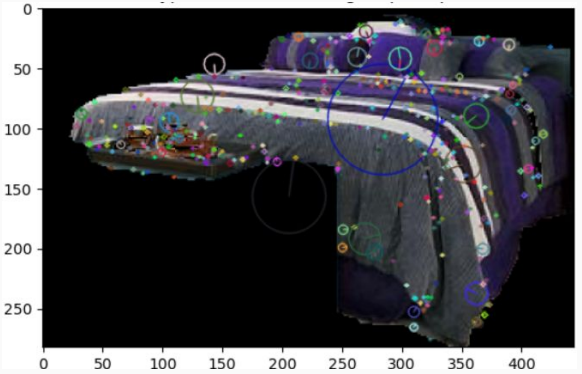
- bilateral filter
- Grabcut

Methods:

- SIFT
- CONVOLUTIONAL AUTOENCODER
- DHASH

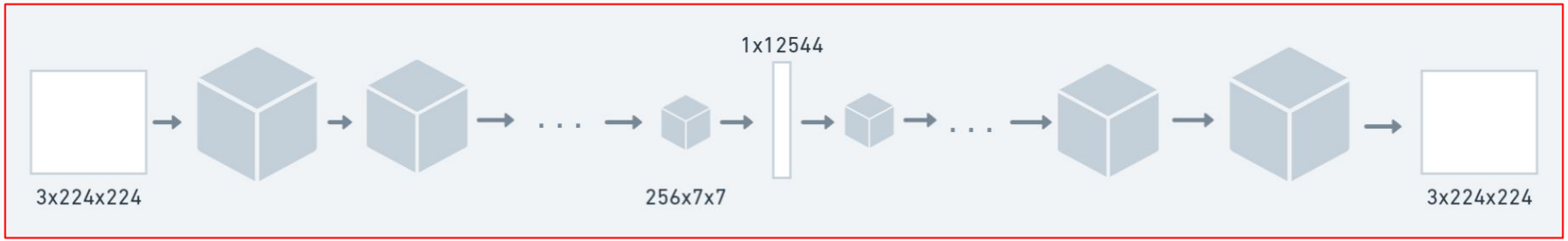
Furniture Retrieval - SIFT

- Pre-computed all key-points and descriptors of the objects in our retrieval dataset and saved in memory
- Compute key-points and descriptor of the bounding box of the input image
- Brute Force Matcher + KNNMatch



Furniture Retrieval - CONVOLUTIONAL AUTOENCODER

- Unsupervised method, trained for 50 epoches.
- 5 convolutional layers and 5 transpose convolutional layers.
- Two main components:
  - *encoder*: series of convolutional and max pooling layers → it returns the latent vector
  - *decoder*: series of transpose convolutional layers → it tries to reconstruct the input image
- Comparison between test image latent vector and all the dataset image latent vectors using the KNN algorithm. The five most similar objects are returned.

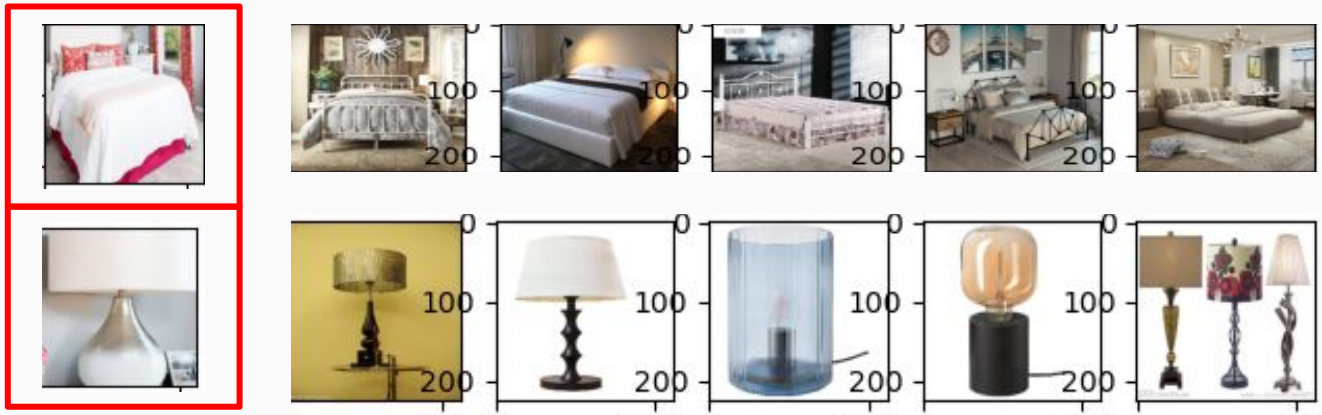


Furniture Retrieval - CONVOLUTIONAL AUTOENCODER

Some examples of retrieved objects

Furniture Retrieval - DHASH

- Very fast algorithm
- Creates an hash of the query.
- Similarity is based on Hamming Distance.
- Mainly used when looking for the same exact image, but in our case performed quite well even for finding similar objects.



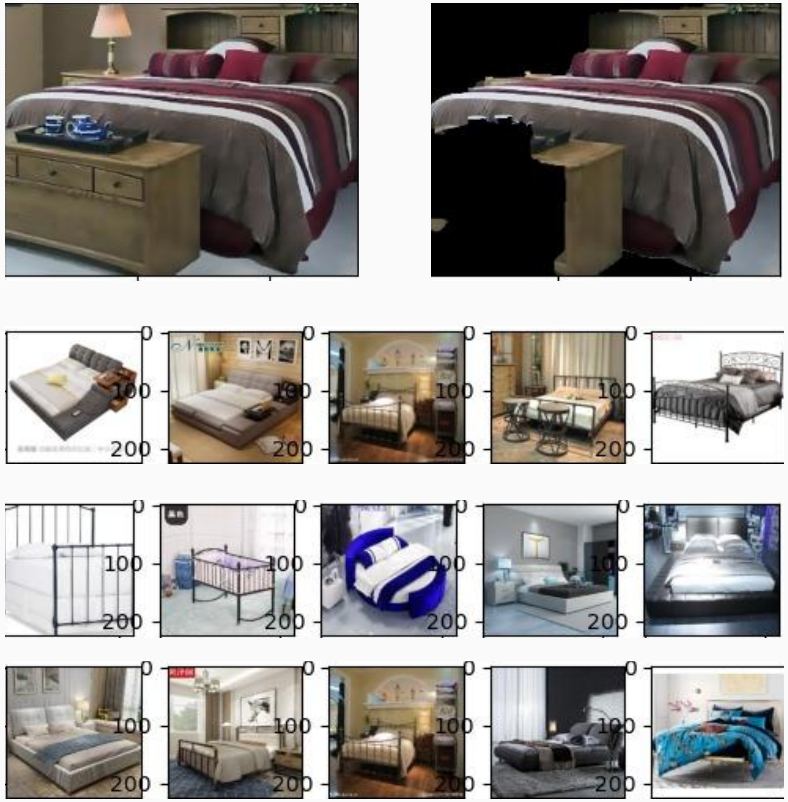
Some examples of retrieved objects

Furniture Retrieval - COMPARISON and EVALUATIONS

Remarks:

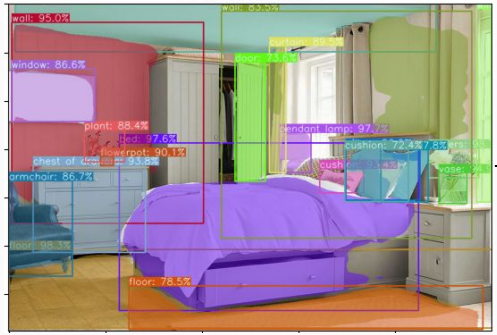
- With SIFT and Dhash, similarity comparison has been computed between objects of the same class. On the other side, autoencoder retrieves objects from the entire dataset.

Retrieval Method	MAP on Default Mask R-CNN	MAP on modified Mask R-CNN
AUTOENCODER	54.4%	**66.3%**
SIFT	57%	57.8%
DHASH	54.9%	55.2%



3- ROOM CLASSIFICATION

2



3.2

ROOM CLASSIFICATION
DIFFERENT METHODS

Room classification

Considering all training images of ADE20K we created a binary matrix indicating which objects are present in the room and we used it to train our two models.

When an input image is given to the pipeline, thanks to our network, we know the objects present in the room. Therefore, we construct a binary feature vector and we feed it to our classification models.

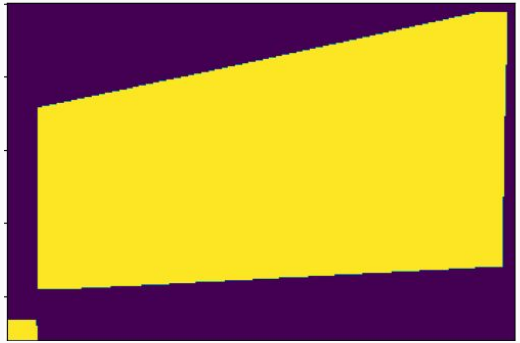
	Random Forest	Fully connected
101 objects	**96.2%**	92.2%
all objects	**98.1%**	96.8%

Despite the differences between the two different methods, what actually increased our accuracy was making the dataset balanced in terms of number of objects in each class.

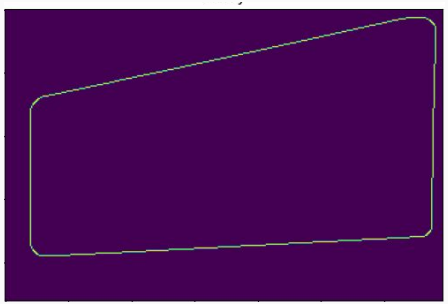
RECTIFICATION

Input Img

1 Mean Shift



2 Thresholding

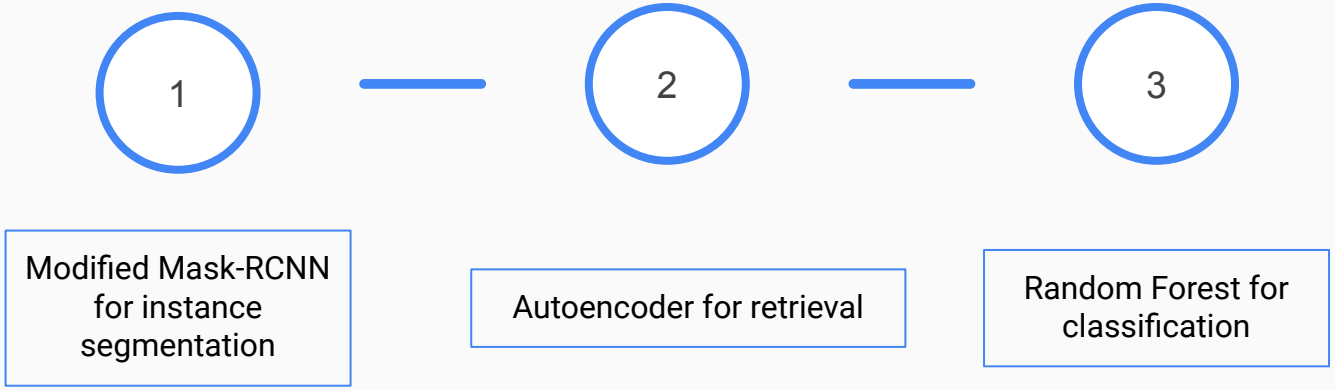


3 Canny

4 Contours

5 Result

Conclusions - Best pipeline



Thanks for the attention!