Deep Learning in Computer Vision

Short introduction

Nathan Piasco

13/12/2017

Le Creusot

Introduction

Architectures

Architectures

Usual building functions

- 2D Convolution
- Non linear function (ReLU)
- Batch Normalization
- Pooling (mean/max)
- Fully connected layer (MLP)

Usual buildings "block"

- *Features extraction:* Conv + Batch Norm + Relu + Max pooling
- *Classification* FC + SoftMax

Famous nets: Alexnet

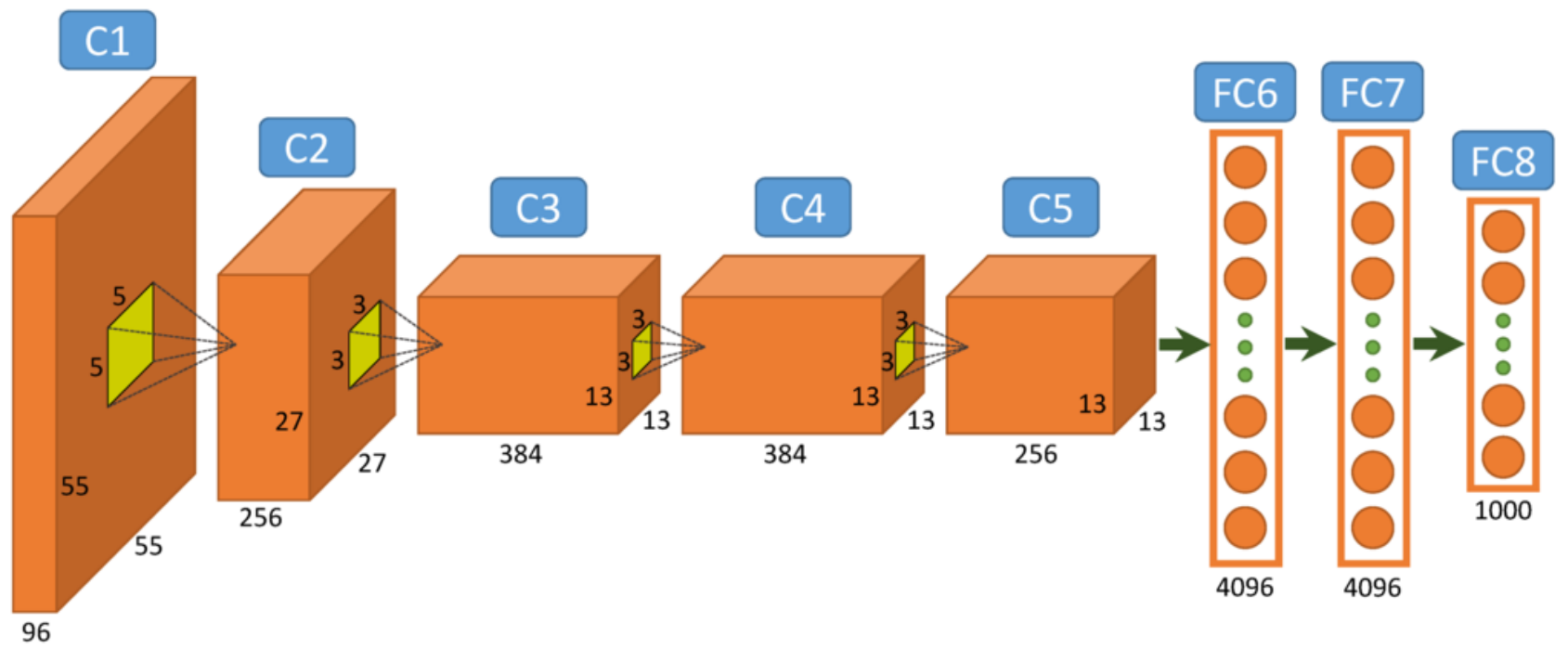


Figure 1: Alexnet

[?]

Famous nets: VGG

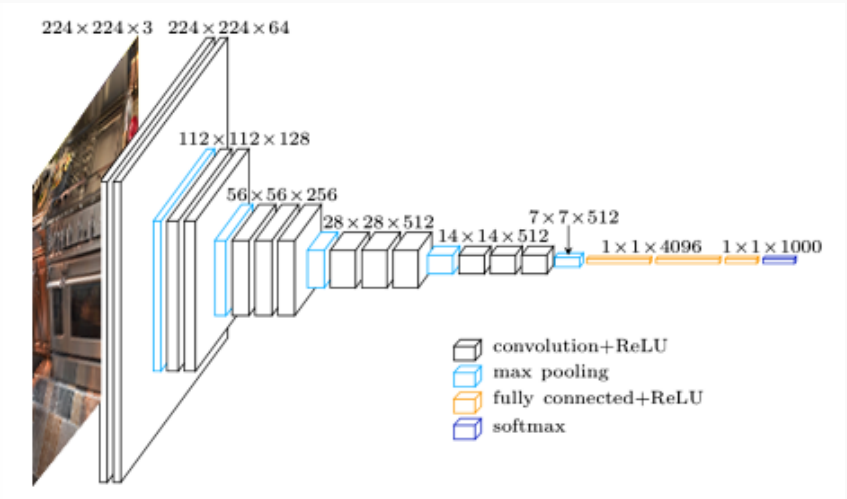


Figure 2: VGG16

16 vs 5 convolution for Alexnet

[?]

Famous nets: GoogLeNet

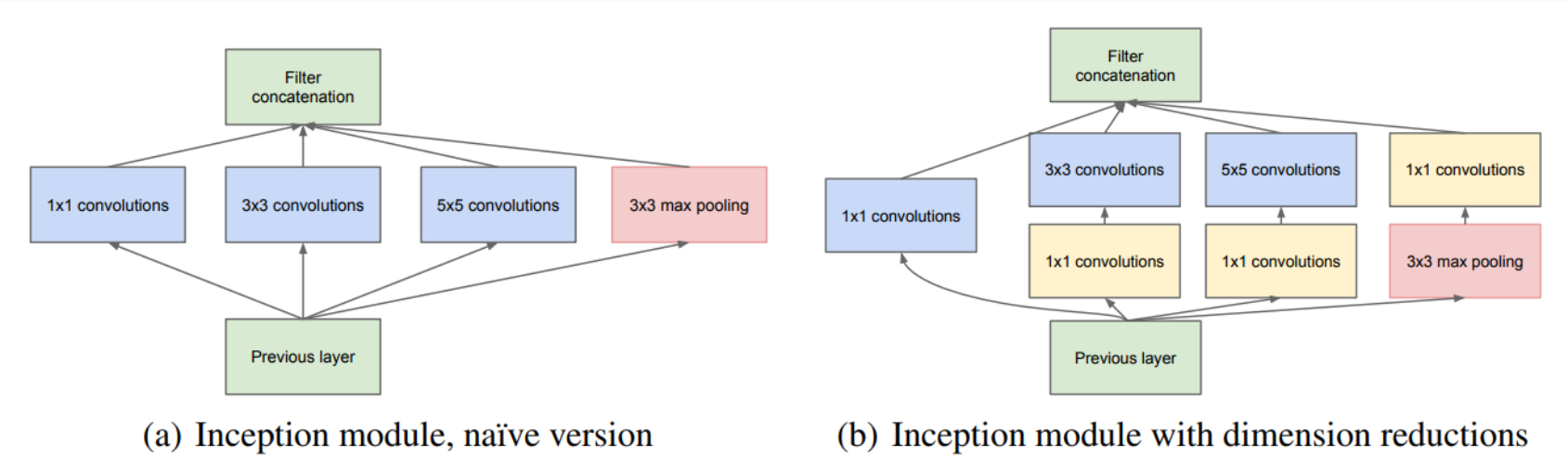


Figure 3: New inception module

Convolution on smaller input = put more convolutions

[?]

Famous nets: ResNet

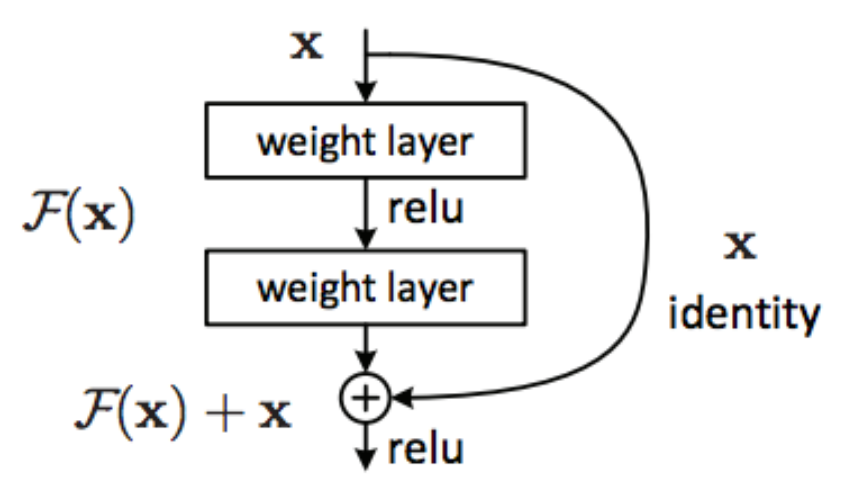


Figure 4: Residual block

Residual networks easier to optimize = put more convolutions (8xVGG)

[?]

Famous nets: DenseNet

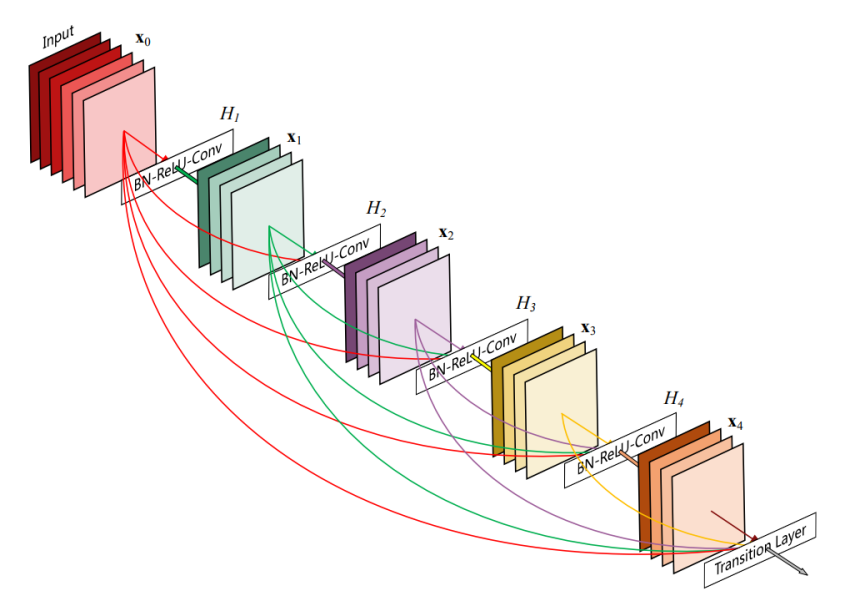


Figure 5: Dense block

Less parameters for similar results (CVPR2017)

[?]

Introduction

Recurrent Networks

Temporal informations in DNN

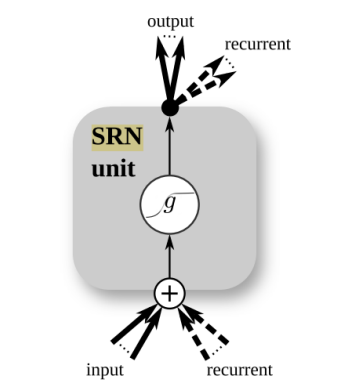


Figure 6: Simple Recurrent Network

Can be applied on sequential data: speech, video, graph...

More complex RN

- Long Short-Term Memory (LSTM)
- Gated Recurrent Unit (GRU)

Introduction

Applications

CNN Applications

In computer vision, deep learning have been first applied for image classification, with:

- CNN for features extraction
- MLP for classification

All the weights (CNN + MLP) are optimized within a common framework **end-to-end**.

DL are now used in others computer vision applications.

Application: Keypoints detection/description

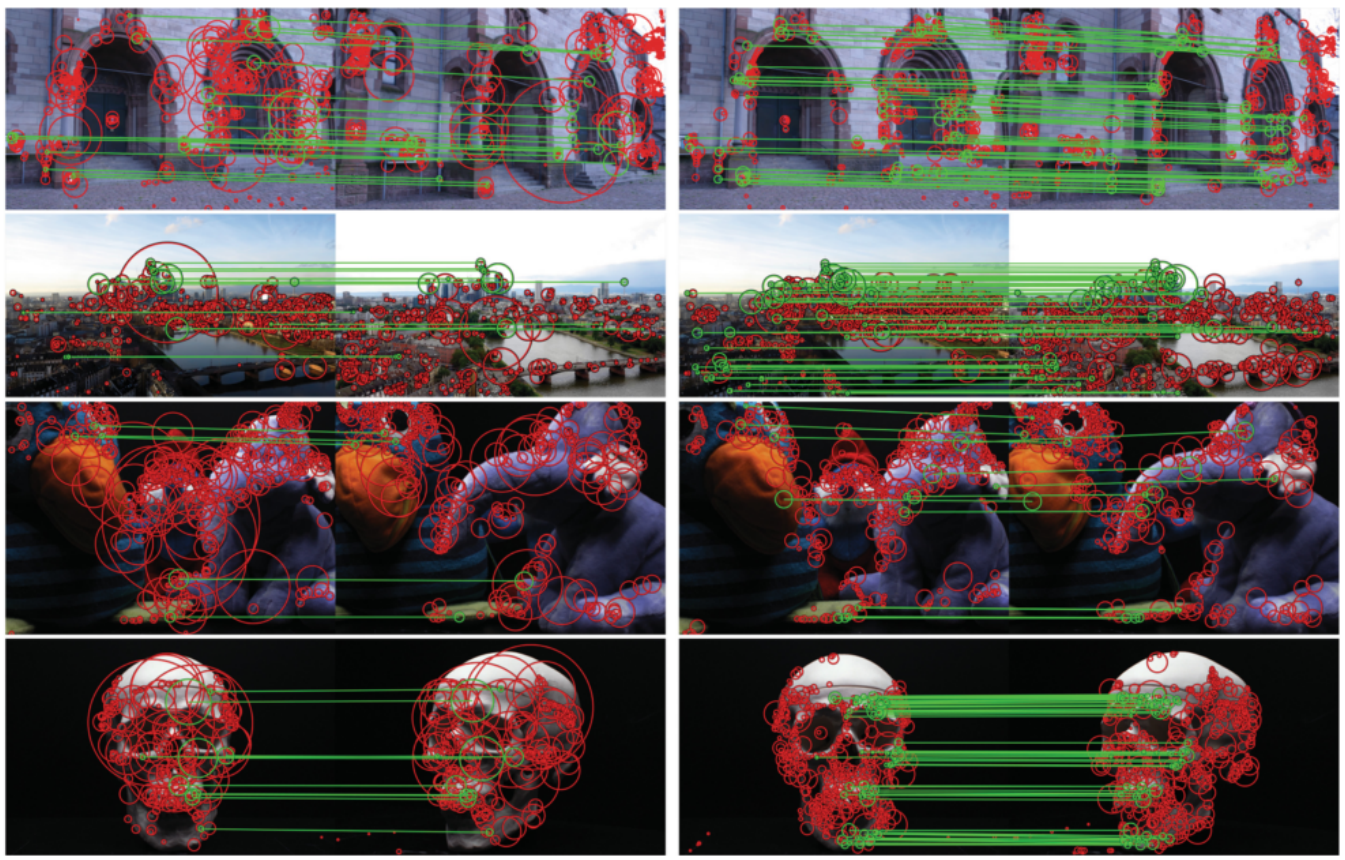


Figure 7: SIFT vs LIFT

[?]

Modality Transfer for VBL

Future research direction

Roadmap

pLaTINUM context

Find the nearest sphere in the database according to a input composed of heterogeneous data

We want to create an indirect method to localize a query within a set of geolocalized RGB-D spheres. The considered pipeline will be:

- Create a discriminative descriptor
- Compute the representation of the spheres with our descriptor
- Compute the similarity of the query (also described by our descriptor) with the spheres representation
- Retrieve the closest sphere

Roadmap

pLaTINUM context

Find the nearest sphere in the database according to a input composed of heterogeneous data

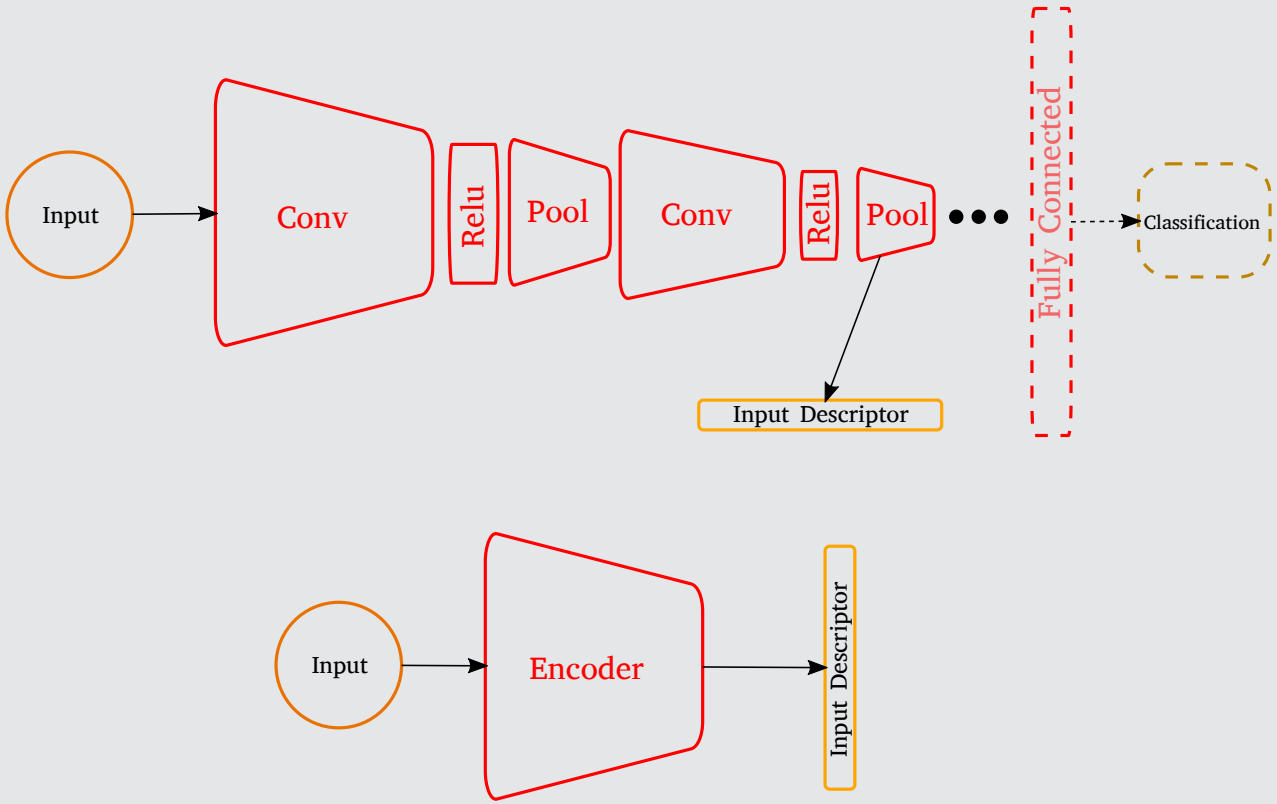
We want to create an indirect method to localize a query within a set of geolocalized RGB-D spheres. The considered pipeline will be:

- **Create a discriminative descriptor**
- Compute the representation of the spheres with our descriptor
- Compute the similarity of the query (also described by our descriptor) with the spheres representation
- Retrieve the closest sphere

Data representation

We first focus our work on creating a robust data representation. Research on state of the art shown that CNN are the best choice.

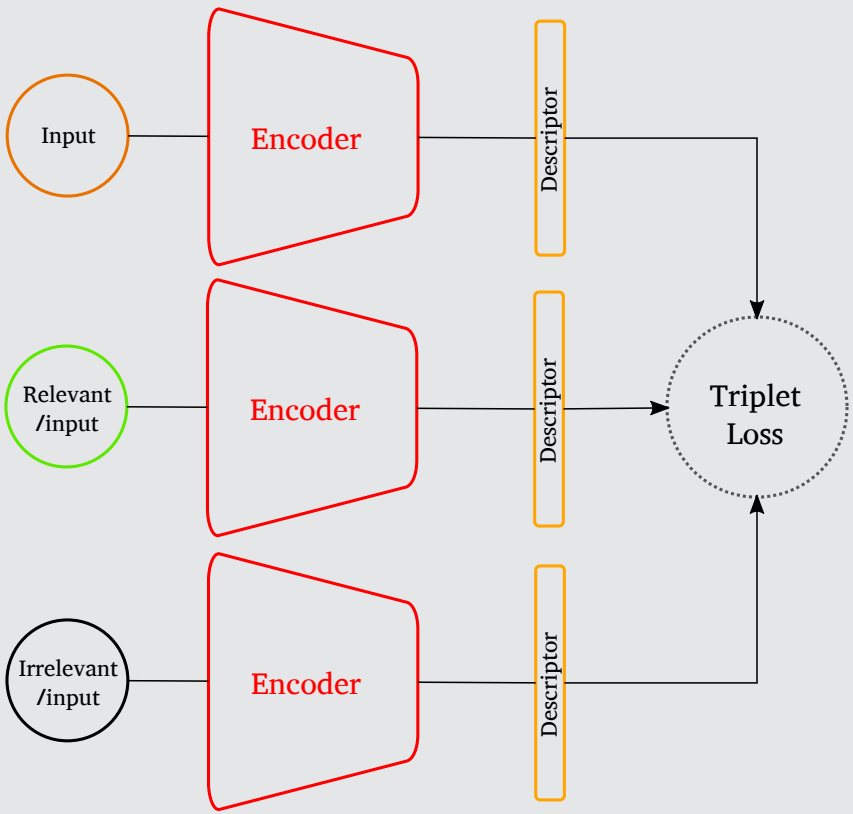
Encoder for data representation



Data representation

We begin with a pre-trained network and perform a fine tuning of its weights.

Encoder training for VBL task



Multiple modalities

How to use more than one modality at the time?

Ideal Desired System specification

Training data type	Testing data type
RGB + Depth + Semantic + Laser reflectance + ...	RGB +/or Depth +/or Semantic +/or Laser reflectance +/or ...

What are we first trying to do

Training data type	Testing data type
RGB + (Depth or Laser reflectance)	RGB

Modality Transfer for VBL

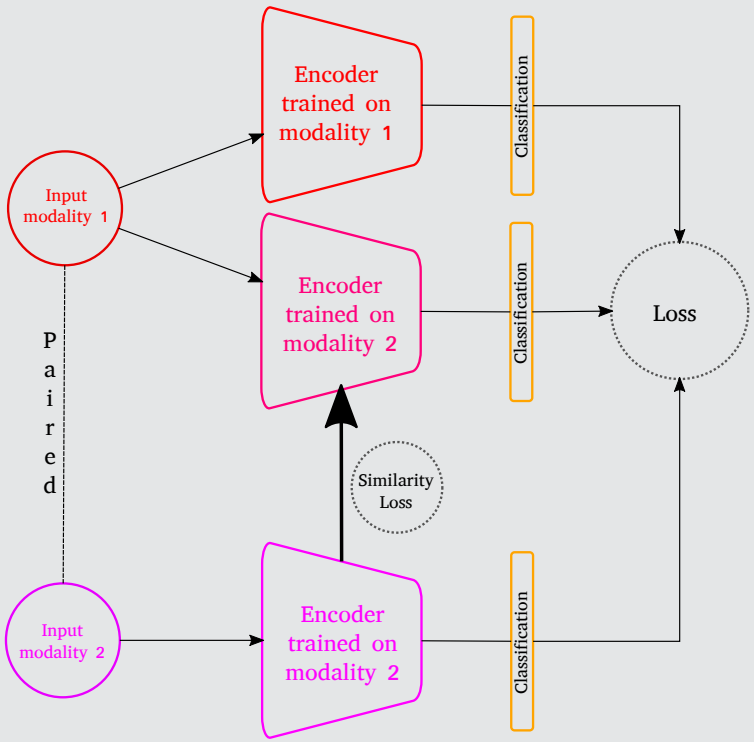
Modality transfer with CNN

Modality Hallucination Network

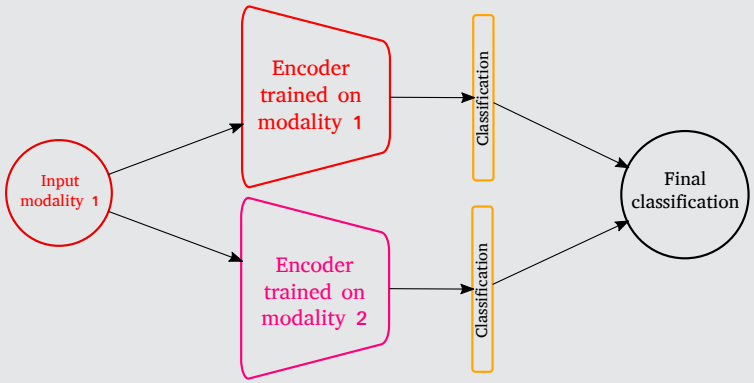
Original contribution

From [Hoffman et al., 2016], applied for the task of semantic interpretation with Fast-RCNN.

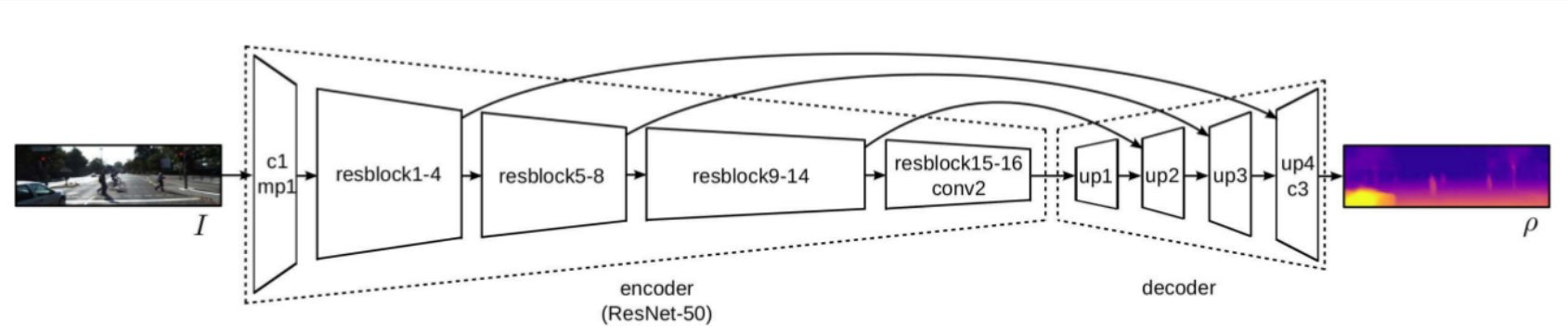
Modality hallucination training



The network at test time



Depth map inference



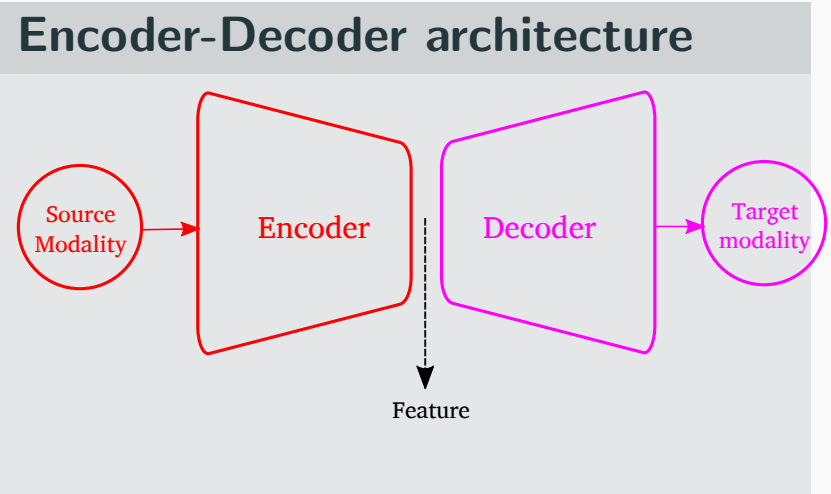
[Kuznietsov et al., 2017]

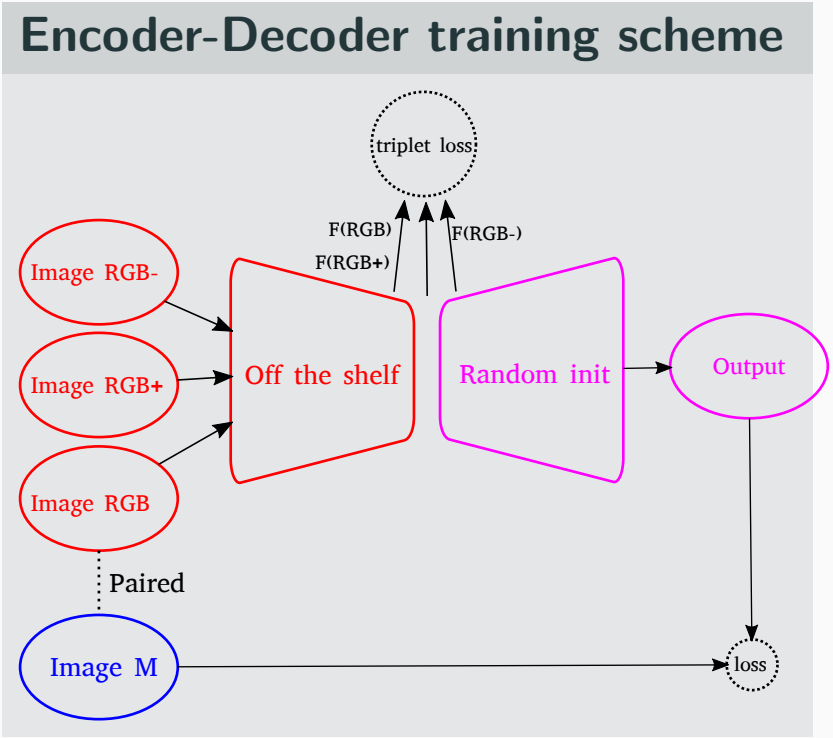
CNN can learn a model to transform a input from one modality to another (widely used to infer depth map from monocular images).

Intuition

As we already use a neural network to describe our data, we should use this kind of architecture to transfer the knowledge among modalities during training.

Proposed architecture





Decoder part is initialized with pre-trained weights and decoder network is randomly initialized.

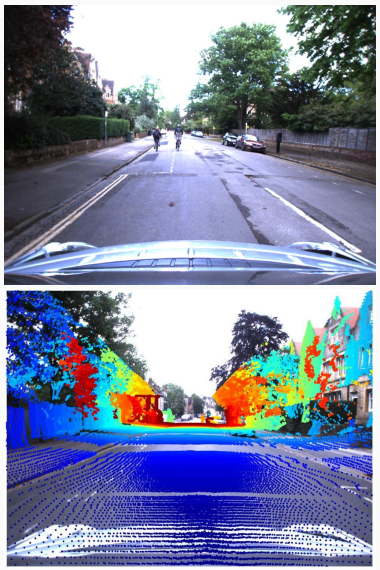
Modality Transfer for VBL

Preliminary Results

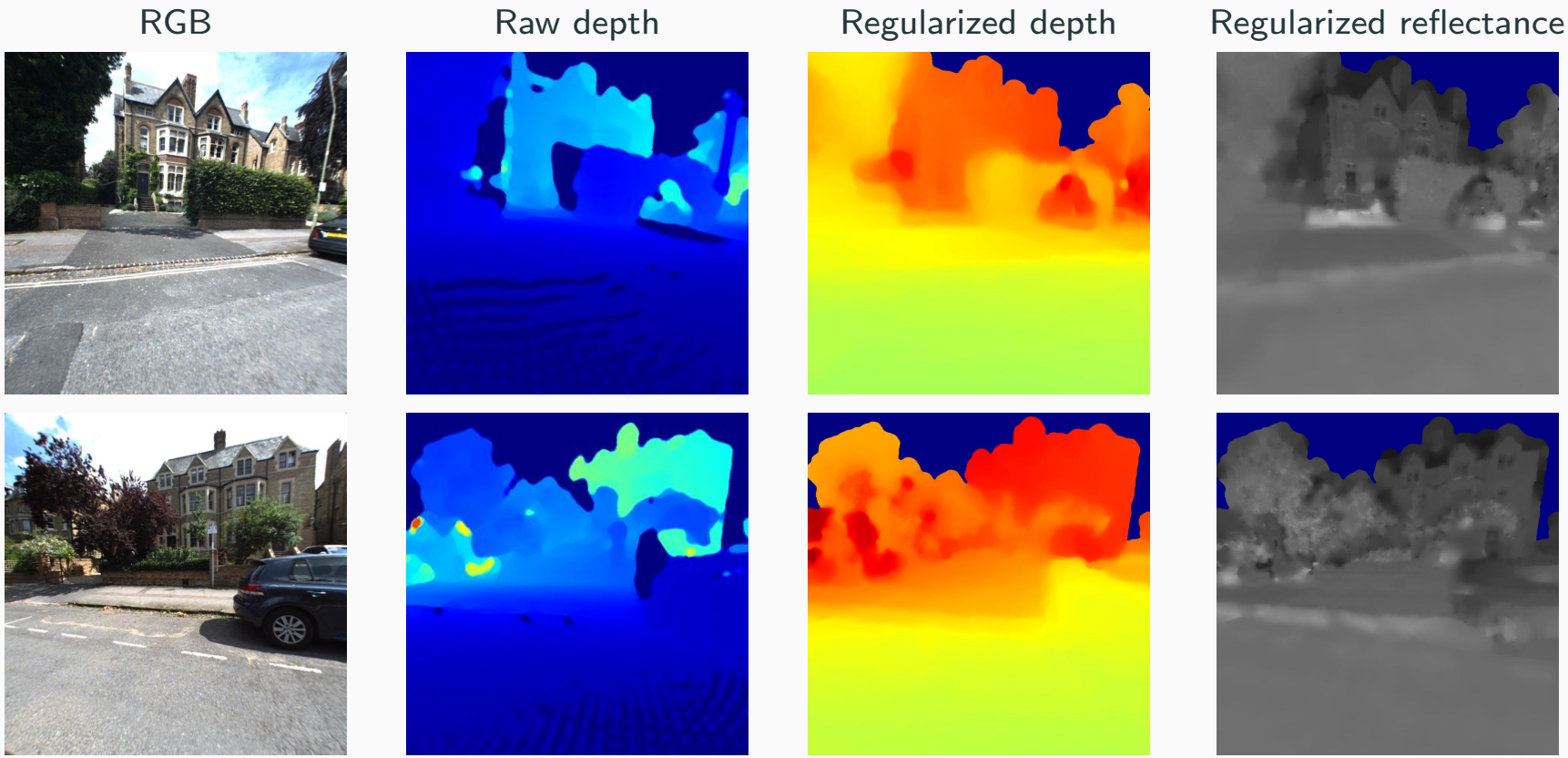
Datasets

We use Oxford RobotCar dataset [Maddern et al., 2016] as it includes:

- Time redundancy for each car trajectory
- 4 cameras on the car & 3 LIDARS (3 modalities: RGB, Depth & Reflectance)

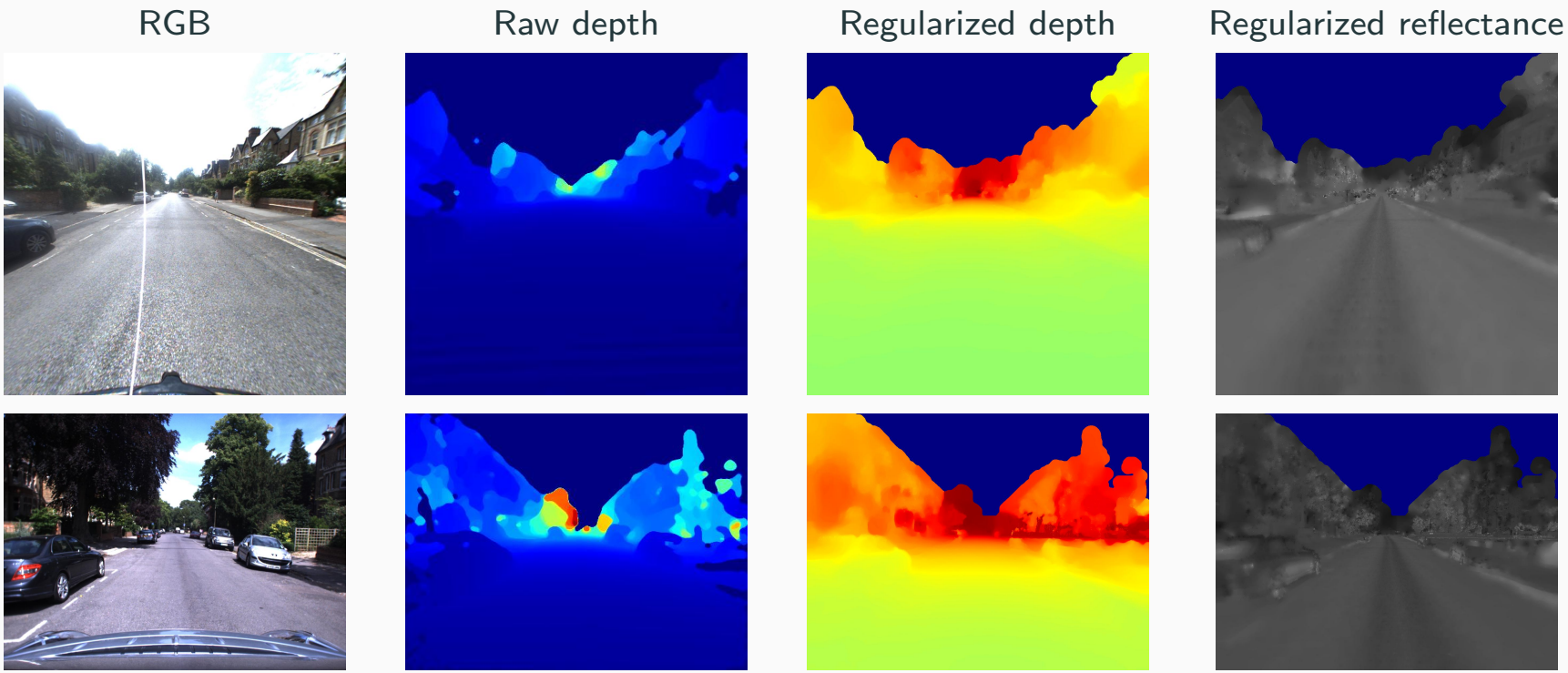


Data



Post-processing algorithm from [Bevilacqua et al., 2017].

Data



Data are not perfect...

Implementation details

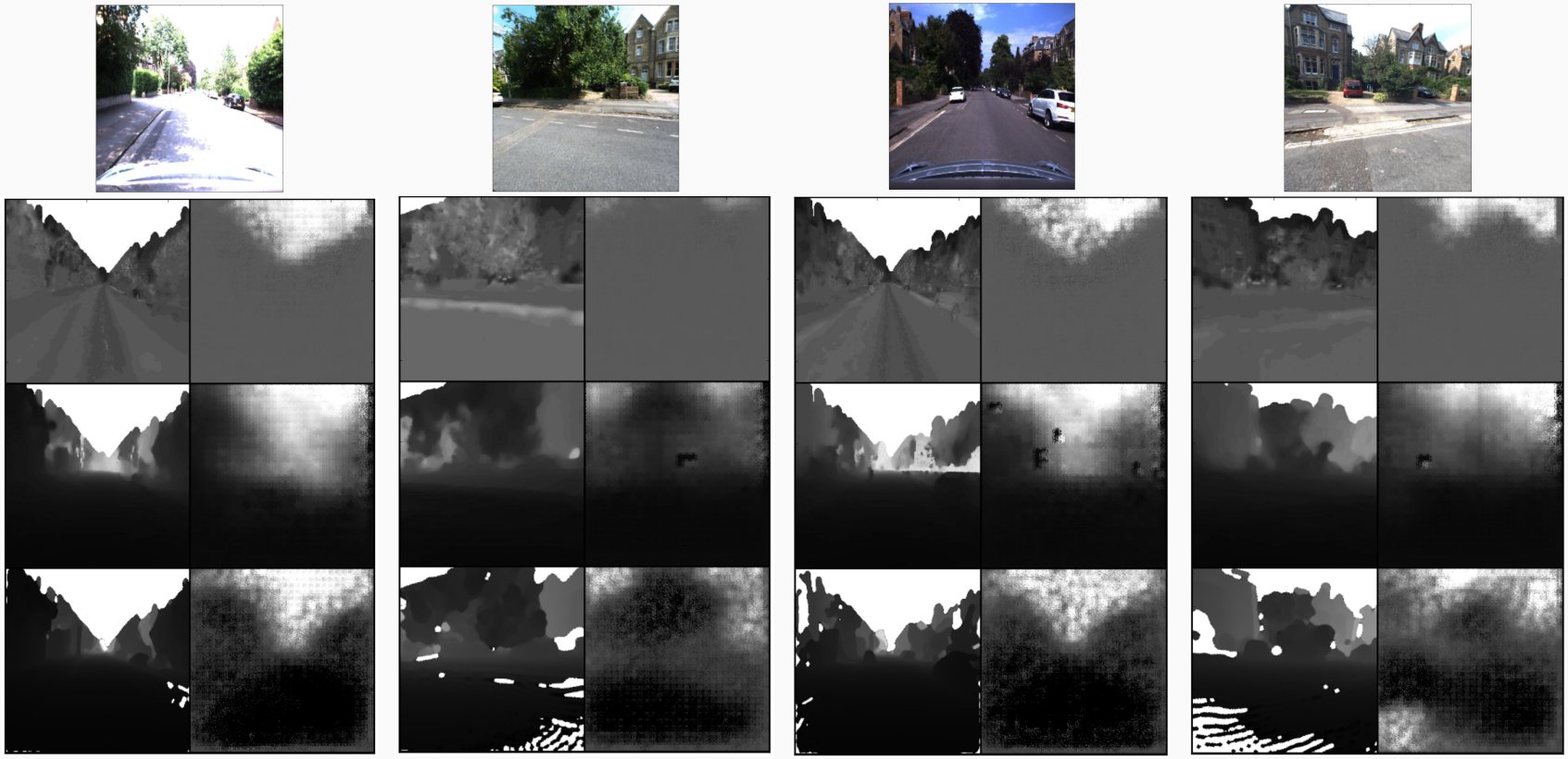
Deep Learning framework: Pytorch

Coder net architecture: Alexnet

Size of the training dataset: 200 triplets (200 * 3 images * 2 modalities)

Trainings and validation frames are from a different region of Oxford than testing images.

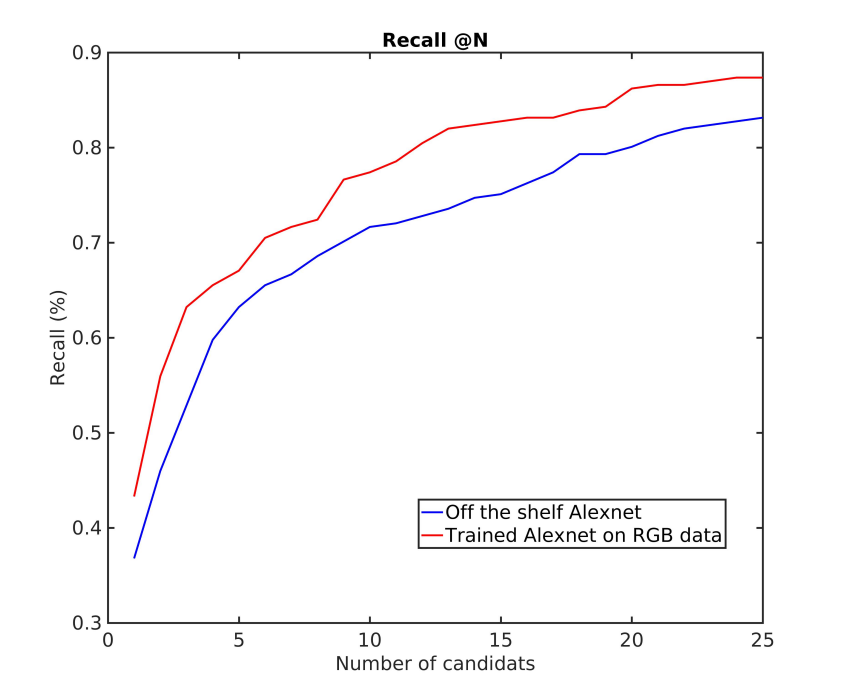
Does the transfer learning work?



Does it improve the localization result?

Evaluation

We state a query as registered if on of the top K retrieved images is at a distance inferior at 25 metres.



First results **are not** better with our Encoder-Decoder net than results obtained with single Encoder training.

Discussion time

References I


Arandjelović, R. and Zisserman, A. (2014).
DisLocation : Scalable descriptor.
In *Proceedings of the Asian Conference on Computer Vision (ACCV)*.


Bevilacqua, M., Aujol, J. F., Biasutti, P., Brédif, M., and Bugeau, A. (2017).
Joint inpainting of depth and reflectance with visibility estimation.
ISPRS Journal of Photogrammetry and Remote Sensing, 125:16–32.


Hoffman, J., Gupta, S., and Darrell, T. (2016).
Learning with Side Information through Modality Hallucination.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 826–834.

References II

Kuznietsov, Y., Stückler, J., and Leibe, B. (2017).
Semi-Supervised Deep Learning for Monocular Depth Map Prediction.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Maddern, W., Pascoe, G., Linegar, C., and Newman, P. (2016).
1 year, 1000 km: The Oxford RobotCar dataset.
The International Journal of Robotics Research (IJRR), page 0278364916679498.