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

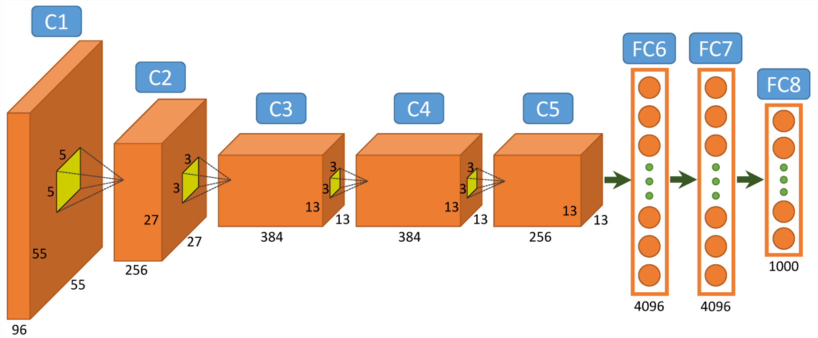


Figure 1: Alexnet

[Krizhevsky et al., 2012]

Famous nets: VGG

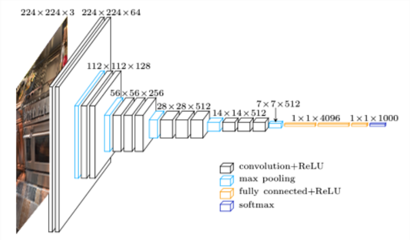


Figure 2: VGG16

16 vs 5 convolution for Alexnet

[Simonyan and Zisserman, 2014]

Famous nets: GoogLeNet

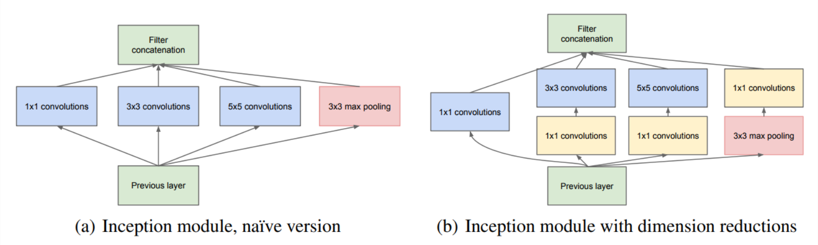


Figure 3: New inception module

Convolution on smaller input = put more convolutions

[Szegedy et al., 2015]

Famous nets: ResNet

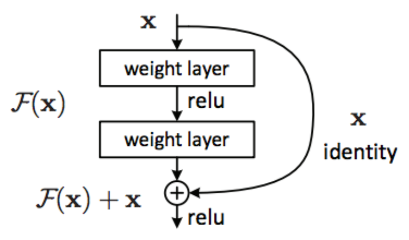


Figure 4: Residual block

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

[He et al., 2016]

Famous nets: DenseNet

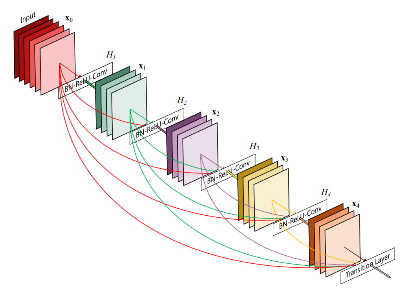


Figure 5: Dense block

Less parameters for similar results (CVPR2017)

[Huang et al., 2016]

Introduction

Recurrent Networks

Temporal informations in DNN

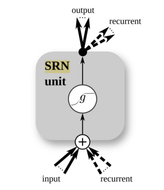


Figure 6: Simple Recurrent Network

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

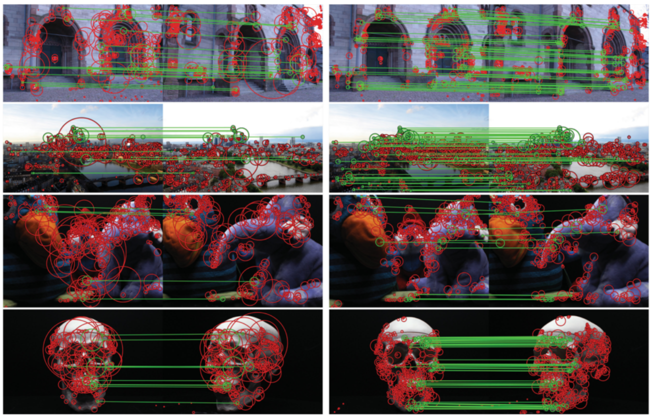


Figure 7: SIFT vs LIFT

[Yi et al., 2016]

Application: Region Proposal Network

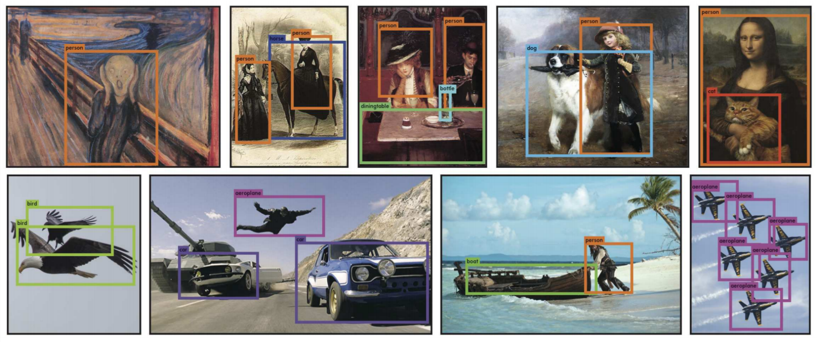


Figure 8: RPN network

Multi-task learning network (Region proposal & classification) Best Paper CVPR2017 [Redmon and Farhadi, 2016]

[Redmon et al., 2016, Girshick, 2015, Ren et al., 2015]

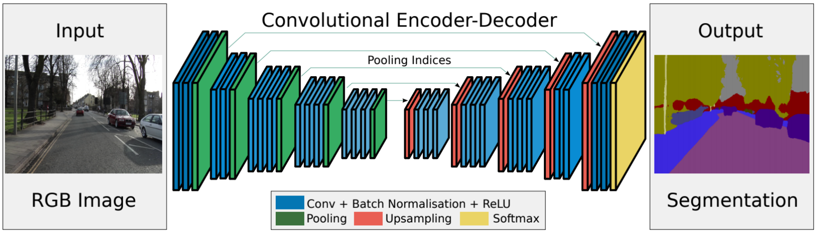


Figure 9: Encoder-Decoder architecture for semantic segmentation

Introducing max-unpooling to obtain an output as wide as the input.

[Badrinarayanan et al., 2015]

Application: Instance segmentation

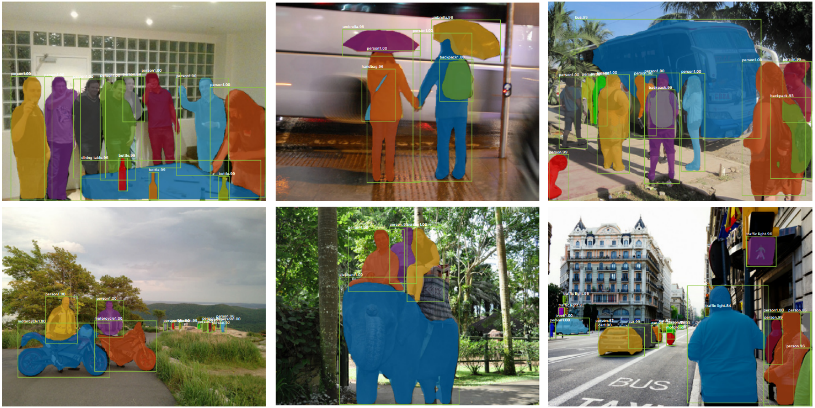


Figure 10: Instance and pixel-level segmentation

Best Paper ICCV2017 [He et al., 2017]

Application: Depth from stereo

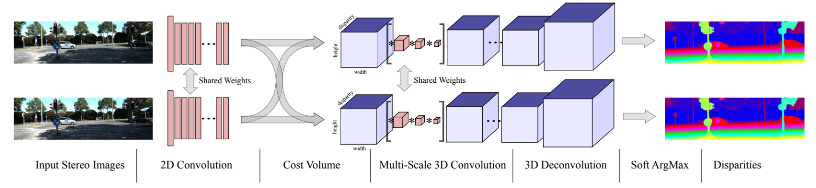


Figure 11: Disparity inference from stereo pairs

Double-inputs network

[Kendall et al., 2017]

Application: Depth from mono

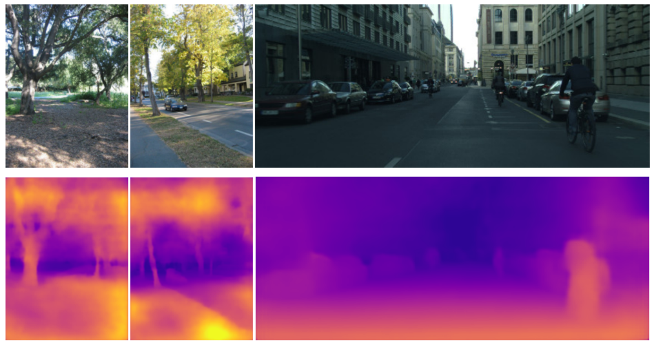


Figure 12: Disparity inference from mono image

Used to improve monocular SLAM [Tateno et al., 2017]

[Kuznietsov et al., 2017]

Application: Visual Odometry

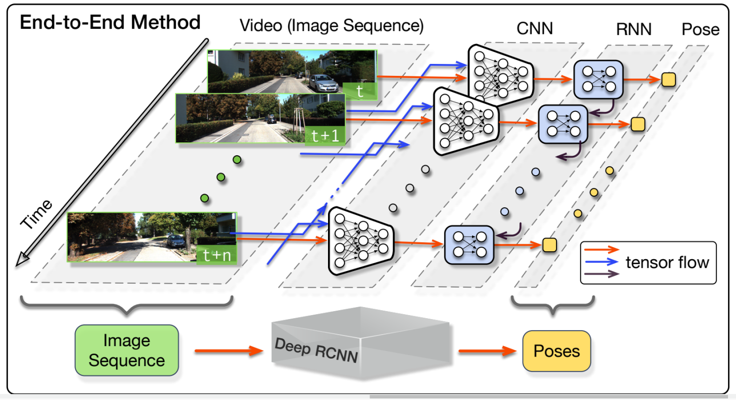


Figure 13: Motion estimation from image sequence

[Wang et al., 2017]

More Applications

And much more:

- Video analysis
- Scene understanding
- Face recognition
- Pose tracking
- Crowd analysis
- Suspicious behaviour detection
- Data generation
- ...

CNN as global Descriptors

Visual Based Localization

Content based image retrieval

Roadmap

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

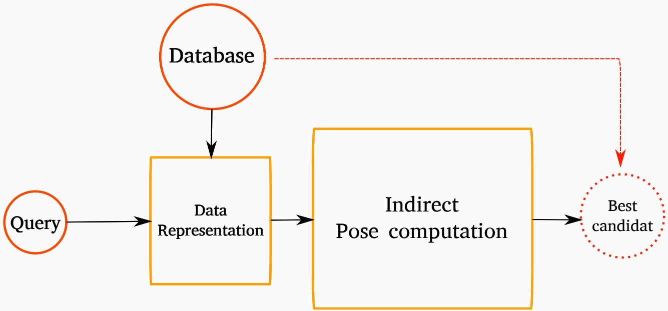
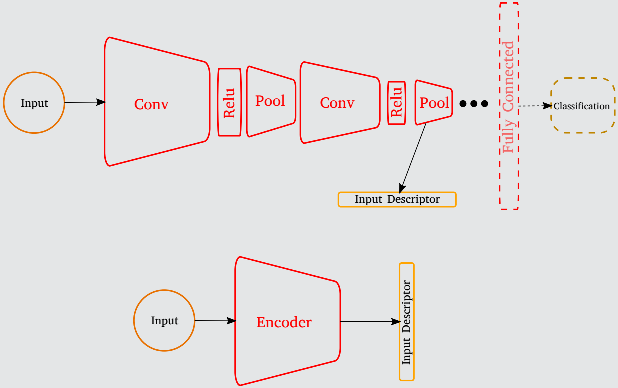


Figure 14: Visual Based Localization (VBL) pipeline

We first focus our work on creating a robust data representation. Research on state of the art shown that CNN are the best choice [Arandjelović et al., 2017].

Encoder for data representation



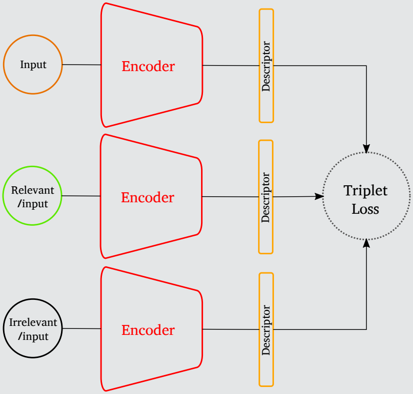
CNN as global Descriptors

Training a global feature extractor

CNN training

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

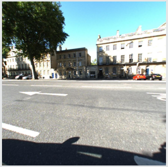
Encoder training for VBL task



CNN training

Training data

Triplet := (Query Image, Positive example, Negative example)




Cost function

During the training, we minimize the following loss:

$$TripletLoss = max(0, \lambda + \|F(I) - F(I^{+})\| - \|F(I) - F(I^{-})\|) \quad (1)$$

Where:

- $F(I)$ the global descriptor of image I computed by the CNN
- λ design a constant margin

Multiple modalities

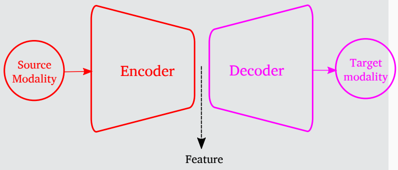
How to use more than one modality at the time?

What are we first trying to do

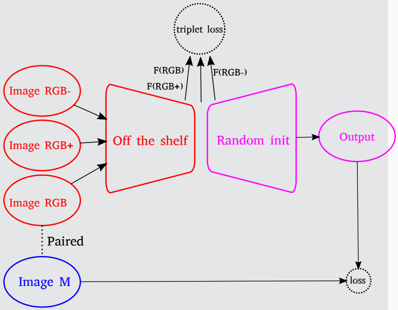
Training data type	Testing data type
RGB + (Depth or Laser reflectance)	RGB

We suppose that complex data (image + lidar related modality) can be acquired **offline**, but all the modality could not be available during test time. The idea is to guide the CNN during the training with **multiple modalities** to improve the description of input of **single modality**.

Encoder-Decoder architecture



Encoder-Decoder training scheme

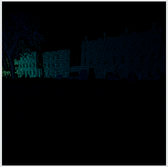


Decoder part is initialized with **pre-trained weights** and decoder network is randomly initialized.

Training

Training data

Triplet := (Query Image + Modality, Positive example, Negative example)

Cost function

Loss become **multi-task**:

$$L = TripletLoss + \alpha \sum_{i,j}^{h,w} |p(i,j) - gt(i,j)| \quad (2)$$

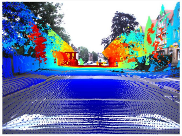
Where:

- p denote CNN inferred modality and gt ground through modality
- α design loss weighting factor

Datasets

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

- Time redundancy for each car trajectory and GPS tags associated to images: **automatic triplets creation!**
- 4 cameras on the car & 3 LIDARS (3 modalities: RGB, Depth & Reflectance)


Important parameters for training

Vocabulary

- **Forward pass:** operation to obtain input representation/classification
- **Batch** several training example forward passed to a CNN before gradient computation and back-propagation
- **Number of Epoch** correspond to the number of times all the batch data have been passed through the CNN
- **Learning rate, Momentum, Weight decay, etc.** Meta-parameters of the optimizer used during the training

Hard and time consuming to fined all the best parameters...

CNN as global Descriptors

Conclusion and advices

Implementation details

Deep Learning framework: `Pytorch`. Easy to use (Python!), fast to learn (1h tutorial), a lot of already available architectures

Net architecture: Alexnet. Begin with a little network (fast to train) to determinate the meta-parameters, then move to a deeper net (VGG).

Size of the training dataset: 400 triplets (400 * 3 images * 2 modalities). Small dataset automatically created. I use **pre-trained network**!

Trends: Robotics

Using deep learning as tools within classical framework:

- Improving monocular SLAM
- Global descriptor to find loop closure

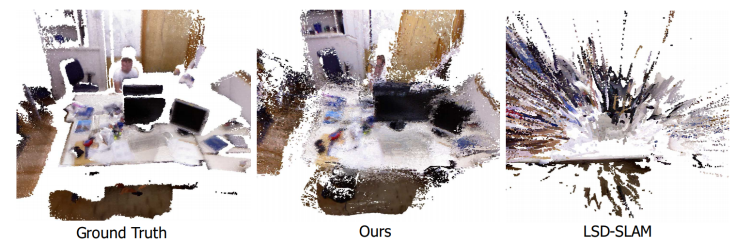


Figure 15: Pure rotational camera motion SLAM problem solved, CNN-SLAM [Qi et al., 2016]

Semi-supervised/unsupervised architecture:

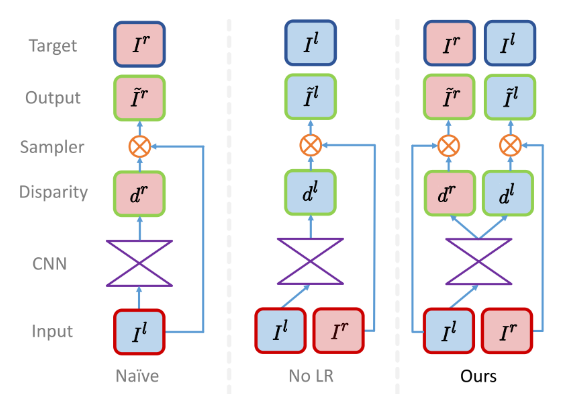


Figure 16: Disparity from mono using stereo pairs as training data [Godard et al., 2016]

Trends: Computer Vision

Multi-task learning:

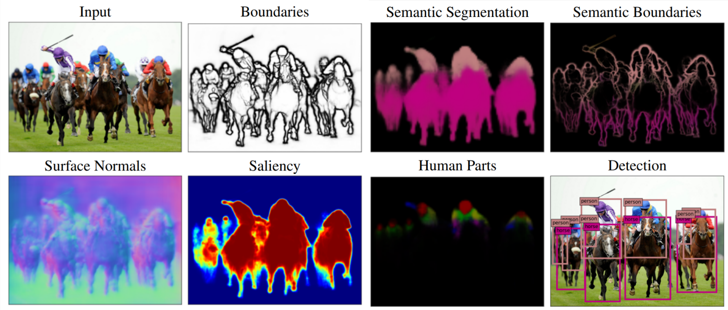


Figure 17: Adding sub-goal increase accuracy over all tasks [Kokkinos, 2016]

No longer limited to RGB modality:

- Deep Learning on Points Cloud, Graph
- Modality fusion (RGB + Depth map) [Hazirbas et al., 2016]
- Multi-spectral images (remote sensing field)

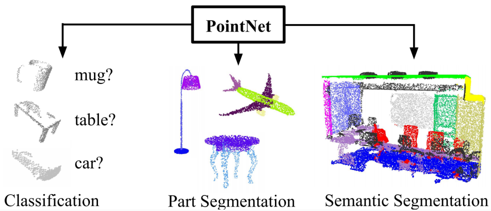


Figure 18: PointNet [Qi et al., 2016]

Discussion time

References I

Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J. (2017).
NetVLAD: CNN architecture for weakly supervised place recognition.
IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), pages 5297–5307.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015).
Segnet: A deep convolutional encoder-decoder architecture for image segmentation.
arXiv preprint arXiv:1511.00561.

Girshick, R. (2015).
Fast r-cnn.
In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

References II

Godard, C., Mac Aodha, O., and Brostow, G. J. (2016).
Unsupervised monocular depth estimation with left-right consistency.
arXiv preprint arXiv:1609.03677.

Hazirbas, C., Ma, L., Domokos, C., and Cremers, D. (2016).
Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture.
In *Asian Conference on Computer Vision*, pages 213–228. Springer.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017).
Mask r-cnn.
arXiv preprint arXiv:1703.06870.

He, K., Zhang, X., Ren, S., and Sun, J. (2016).
Deep residual learning for image recognition.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

References III

Huang, G., Liu, Z., Weinberger, K. Q., and van der Maaten, L. (2016).
Densely connected convolutional networks.
arXiv preprint arXiv:1608.06993.

Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., and Bry, A. (2017).
End-to-end learning of geometry and context for deep stereo regression.
arXiv preprint arXiv:1703.04309.

Kokkinos, I. (2016).
Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory.
arXiv preprint arXiv:1609.02132.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012).
Imagenet classification with deep convolutional neural networks.
In *Advances in neural information processing systems*, pages 1097–1105.

References IV

Kuznietsov, Y., Stückler, J., and Leibe, B. (2017).
Semi-supervised deep learning for monocular depth map prediction.
arXiv preprint arXiv:1702.02706.

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.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2016).
Pointnet: Deep learning on point sets for 3d classification and segmentation.
arXiv preprint arXiv:1612.00593.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016).
You only look once: Unified, real-time object detection.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788.

References V

Redmon, J. and Farhadi, A. (2016).
Yolo9000: better, faster, stronger.
arXiv preprint arXiv:1612.08242.

Ren, S., He, K., Girshick, R., and Sun, J. (2015).
Faster r-cnn: Towards real-time object detection with region proposal networks.
In *Advances in neural information processing systems*, pages 91–99.

Simonyan, K. and Zisserman, A. (2014).
Very deep convolutional networks for large-scale image recognition.
arXiv preprint arXiv:1409.1556.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015).
Going deeper with convolutions.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

References VI

Tateno, K., Tombari, F., Laina, I., and Navab, N. (2017).
Cnn-slam: Real-time dense monocular slam with learned depth prediction.
arXiv preprint arXiv:1704.03489.

Wang, S., Clark, R., Wen, H., Trigoni, N., Clark, R., Wang, S., Wen, H., Trigoni, N., Markham, A., Markham, A., et al. (2017).
Deepvo: towards end-to-end visual odometry with deep recurrent convolutional neural networks.
In *International Conference on Robotics and Automation*.

Yi, K. M., Trulls, E., Lepetit, V., and Fua, P. (2016).
Lift: Learned invariant feature transform.
In *European Conference on Computer Vision*, pages 467–483. Springer.