A Deep Convolutional Neural Network for Location Recognition and Geometry Based Information

Francesco Bidoia[1] Matthia Sabatelli[1,2] Amirhossein Shantia[1] Marco A. Wiering[1] Lambert Schomaker[1]

[1]Institute of Artificial Intelligence and Cognitive Engineering, University of Groningen

[2] Montefiore Institute, Department of Electrical Engineering and Computer Science, Université de Liège, Belgium

Introduction

- Autonomous Navigation Systems (ANS) have recently gained a lot of attention from various research domains, nevertheless in order to work well, most of them are based on fairly expensive sensor technologies.
- As a solution it is possible to build the ANS based only on visual input (i.e. camera) which, in combination with the odometry of the system, allow the system to map an environment and navigate.
- In this work we show that standard Computer Vision and Deep Learning algorithms fail in understanding the visual information that is received as input from a camera. This is given due to their lack in preserving the **Geometrical Properties** of the images. The consequence of this lack of understanding turns into an ANS which is unable to **localize** itself.
- To solve this problem we present a novel **Deep Convolutional Neural Network** [2] inspired by the famous *Inception Module* [3] which is tested on 2 Datasets that we have created: the first one is used for *Scene Recogntion* while the latter one for *Location Recognition*.

Datasets Creation

The Datasets consist in a set of recordings of an environment filled with specifically designed different **Visual Markers** (VM) as presented in Figure 1.

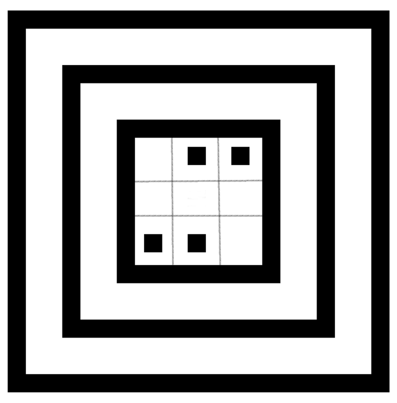

Figure 1: *Example of a Visual Marker used for the creation of the Datasets*

We assign to each VM, a specific location in the environment which the robot will have to recognize.
We create the Datasets as follows:

- Scene Recognition Dataset
  - 8144 different pictures
  - 10 classes that correspond to 10 different VMs
  - Data-Augmentation up to $32,5640$ samples

  80% is used as Training set, while 10% is used as Validation and Testing sets respectively.
- Location Recognition Dataset
  - 1 Extra class (φ) is added to the Dataset
  - φ consisting of all the images presented in the previous Dataset but flipped, as shown in Figure 2

  We split the Dataset into 50% for Training and 50% for Testing.

Figure 2: *Two flipped pictures representing the same location. The images have been used as part of the Location Recognition Dataset*

Histogram of SIFT Features + Artificial Neural Network

- Proposed by [1] the Scale Invariant Feature Transform (SIFT) in combination with the Bag of Visual Word (BOW) was the state of the art technique in Computer Vision before the advent of CNNs
- Not suitable for our Dataset which contains pictures with few unique features that do not help the algorithm during the generalization phase
- Instead we identify the features via the use of a **Dense Grid of Key Points** over the images as shown in Figure 3

Figure 3: *Visualization of the Key Points on an Image present in the Datasets*

- We obtain:
  - 1 key point every 8 pixels
  - Total of 5440 key points per picture
  - 128 features long vector per key point
- We use the BOW clustering algorithm to reduce the amount of features which are then given to an Artificial Neural Network for the final classification

Inception V3 Convolutional Neural Network

This Artificial Neural Network consists in:

- 33 layers deep neural network
- Replaces 5×5 convolutions with two consecutive 3×3 ones as shown in Figure 4
- Similarly a 3×3 convolution is replaced by 3×1 and 1×3 convolutions
- We take the original architecture publicly available on `GitHub` [1] and train only a final fully connected layer of 250 hidden units
- ≈ 1.3 million of parameters have to be trained
- Due to the extensive use of **Pooling** the geometrical properties of the inputs get changed misleading the **Localization** phase

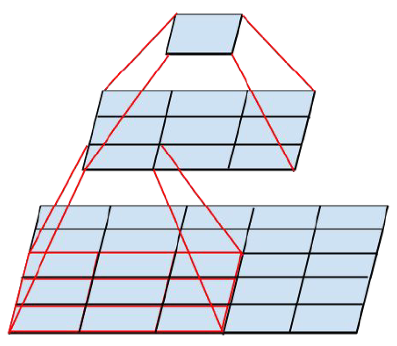

Figure 4: *Example of how two 3×3 convolutions can be used instead of a 5×5 one*

Novel Deep Convolutional Neural Network Architecture

We propose the following neural architecture.

Motivation

- Use the optimization techniques of the Inception $V3$ architecture presented in Figure 4
- **Avoid Pooling** in order to preserve the geometrical information of the input

Neural Architecture

- Input Images as a $300 \times 300 \times 3$ tensor
- 7×7 convolution with 48 channels and strides $= 3$
- 9×9 convolution with 32 channels and strides $= 2$
- 11×11 convolution with 24 channels and strides $= 1$
- 22×22 convolution with 16 channels and strides $= 1$
- **Inception Module** that uses a 1×1 followed by a 3×3 convolution in parallel with a 1×1 and 5×5 one
- Final fully connected layer of 200 Hidden Units
- *Rectified Linear Unit* (ReLU) $f(x) = max(0, x)$ as activation function
- Dropout rate of 0.1

Results

Scene Dataset Accuracies

- We report in Table 1 the accuracies that have been obtained by the different Computer Vision algorithms. τ refers to the amount of computation time in hours.

Table 1: *The accuracies on the Scene Dataset*

Algorithm	Validation-Set	Testing-Set	τ
BOW	89.7%	88.5%	72
Inception $V3$	100%	100%	2.3
DCNN	100%	98.7%	1.5

- Despite being the architecture performing best we see that the Inception $V3$ does not generalize when tested on the Location Dataset.
- We report in Table 2 the accuracies of the neural architectures. We introduce the class **Tricked** in which we report if the ANN labels a flipped image as an original one.

Location Dataset Accuracies

Table 2: *The accuracies on the Location Dataset*

Algorithm	Correct	Tricked	Missed
Inception $V3$	0%	97.43%	2.57%
DCNN	99.7%	0%	0.3%

- We see that our novel DCNN is able to classify correctly the images in the Location Dataset, meaning that the geometrical properties of the input are maintained.
- Figure 5 shows the robustness of the system that makes use of our neural architecture in a simulated environment.

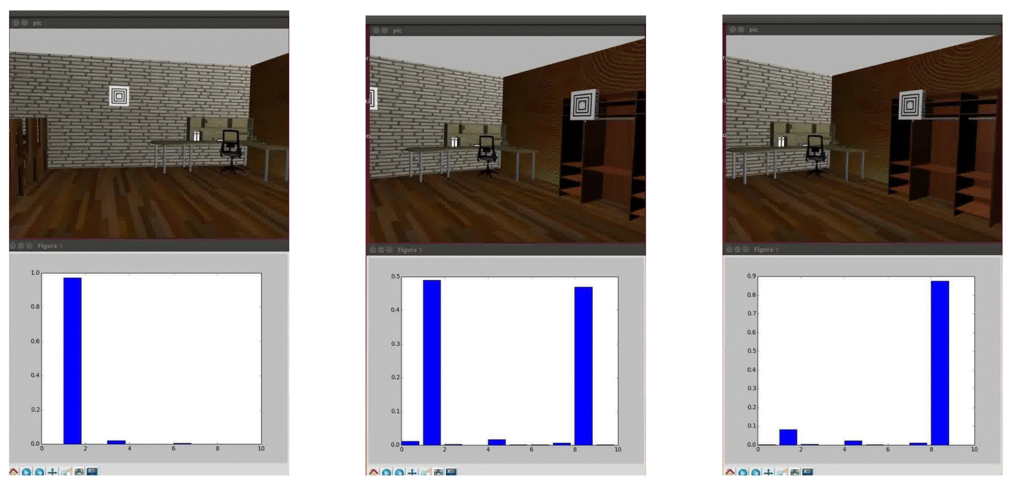

Figure 5: *The softmax output of the DCNN on two flipped pictures representing the same location. We observe how the ANN is able to distinguish the two locations*

Conclusion

- In this work we show how it is possible to build Deep Convolutional Neural Networks without having to rely on dimensionality reduction techniques such as **Pooling**.
- We show how this can be particularly useful in the context of Autonomous Navigation Systems that are based on visual input.

References

[1] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[2] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[3] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[1]`https://github.com/tensorflow/models/tree/master/inception`