Object Recognition in Egocentric Videos for Assistance of Neuro-Prosthesis Wearer

Philippe PÉREZ DE SAN ROMAN, Jenny BENOIS-PINEU,
Daniel CATTAERT,Aymar DE RUGY, Florent PACLET
Funded by: PEPS Suivip CNRS-Idex

November 14, 2016

Summary

Introduction

Introduction 1

Motivations:

Amputees: Grasping actions, daily life activities

Neuro-prosthesis: Electromyogram control

Liberty degree: $\uparrow$ Amputation, $\uparrow$ Degree-of-freedoms,
$\downarrow$ Electromyogram control signal

What if we can see what the subject sees ?

Visually identify the object-of-interest.

$\rightarrow$ Scene camera

But where is it in the subjects view ?

When the subject intents to grasp an object, he looks at it.

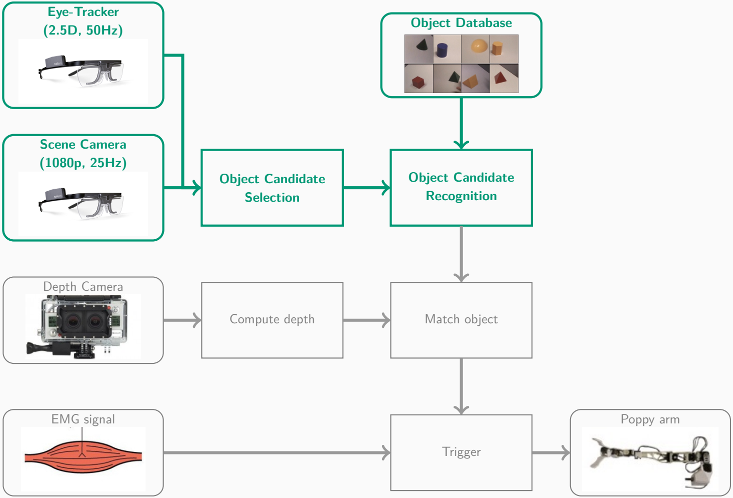
$\rightarrow$ Eye-tracker

Where is it exactly w.r.t. the prosthesis ?

$\rightarrow$ Depth camera at the prosthesis to locate it in 3D

Introduction 2

Framework of prosthesis control system:



State of the art

Psycho-visual Attention

Scene exploration times:

Scene discovery: Sparse eye motions, 240-300 ms

Fixation: Focus of the object-of-interest, 250 ms

Micro-saccades: Small oscillations on the object-of-interest, 6-300 ms[1]

Grasping: Motion initiation, 400-900 ms

Distractors Light, motion, other objects, 100-500 ms[2,3]

Our experiment confirms, even with a few subjects, these times.

[1] Susana Martinez-Conde et al. (2009). "Microsaccades: a neurophysiological analysis". In: *Trends in neurosciences* 32.9, pp. 463–475.

[2] Helene Devillez, Anne Guérin-Dugué, and Nathalie Guyader (2015). "How task difficulty influences eye movements when exploring natural scene images." In: *EUSIPCO*. IEEE, pp. 1536–1540.

[3] D. Villani et al. (2015). "Visual exploration patterns of human figures in action: an eye tracker study with art paintings". In: *Front Psychol* 6, p. 1636.

Object Recognition with Depp CNN

Convolutional Neural Network (CNN):

- Supervised machine learning algorithm
- Inspired by animal brain and visual cortex
- Image $I \rightarrow$ Probability $\widehat{P}(\ I \mid class\ \ C\ )$

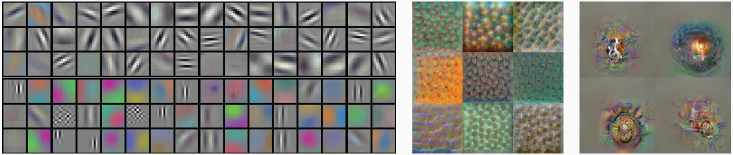
1 layer:

- Convolution, Non-linearity, Max Pooling $\rightarrow$ Feature extraction

More layers (going deeper):

- Abstract the image contents
- Edges $\rightarrow$ Textures $\rightarrow$ Object parts

Example of features computed with a CNN:



Object Recognition with Depp CNN 2

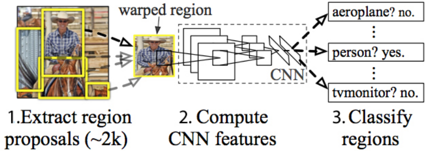
Common approach to explore an image:

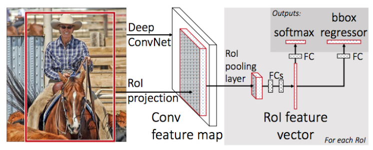
Full Search: To many proposals, very slow

R-CNN: 2000 object proposals, 42s

Fast R-CNN: 2000 object proposals, 320ms

Region based CNN[4,5] and Fast R-CNN[6]:





[4] J.R.R. Uijlings et al. (2013). "Selective Search for Object Recognition". In: *International Journal of Computer Vision* 104.2, pp. 154–171.

[5] R. B. Girshick et al. (2014). "Rich fea- ture hierarchies for accurate object detection and semantic segmentation". In: *CVPR*.

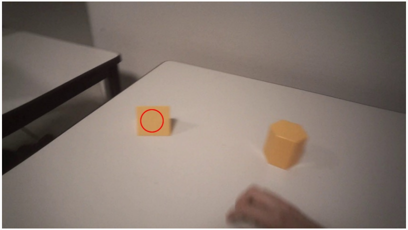
[6] Ross B. Girshick (2015). "Fast R-CNN". . In: *CoRR* abs/1504.08083.

Problem Formulation

Problem Formulation

Our problem: Recognize the object-of-interest in the egocentric videos of the scene camera, using by eye-tracking, in the time of an eye-fixation ($< 250ms$)

1. **Object Localization:**
   → It it focused by eye-tracking after the scene exploration has been completed
   → Fixation is maintained, except if a distractor appears

   → One, and only one, "object proposal"
2. **Object classification:**
   → Deep Convolutional Neural Network

LEGO Dataset

Experimental setup

4 Subjects were equipped with:

- The eye-tracker
- The scene camera

Scene setup:

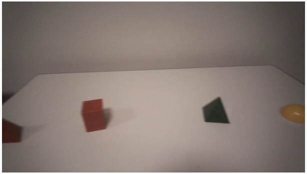
- white background
- white table
- 4 objects out of 8 (in line)

A subject wearing the eye-tracker:

Experiment:

1. The subject sits at the table, his eyes are closed.
2. We place the objects.
3. He is instructed to grasp one as we start the recordings
4. He opens his eyes, finds the object, and grasps it.
5. We stop the recordings.

The view from the scene camera:



Videos

Content:

Egocentric video, Eye-tracking, Annotation

Number of videos in the LEGO dataset:

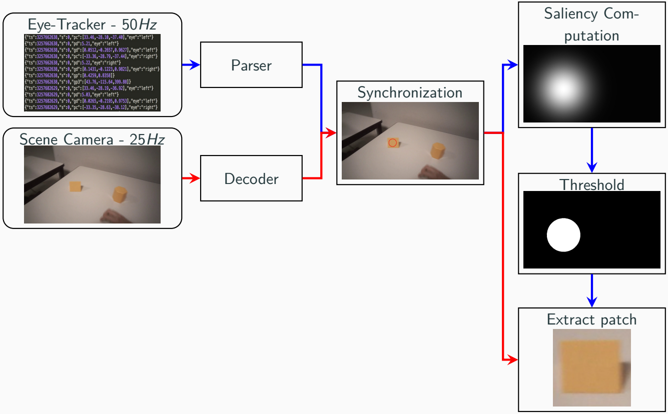
By categories for Training, Validation and Testing.

	Categories	Training	Validation	Testing	Total
	Background	90	31	0	121
	Cone	13	5	4	22
	Cylinder	4	2	1	7
	Hemisphere	8	3	3	14
	Hexagonal_Prism	10	4	3	17
	Rectangular_Prism	17	5	6	28
	Rectangular_Pyramid	10	4	3	17
	Triangular_Prism	17	5	4	26
	Triangular_Pyramid	11	3	4	18
	Total/BGD	**90**	**31**	**28**	**149**

Object Candidate Selection

Object Candidate Selection

Object Candidate Selection framework:



Synchronization

Problem:

Scene camera 25 Hz

Eye-tracker 50 Hz, missing values (eyes closed, blinking)

Solution:

Spline interpolation: At the time t of a frame
Using a time window of δt milliseconds

Saliency computation 1

Equation for the computation of the Wooding Map[7]:

Normalized Wooding Map:

$$W(I, f, x, y) = \frac{A}{\|W\|_\infty} \times \exp \frac{-(x - x_f)^2 - (y - y_f)^2}{2 \cdot \sigma(I, d_f)^2 + \epsilon}$$

Radius adapted by the distance of the fixation point:

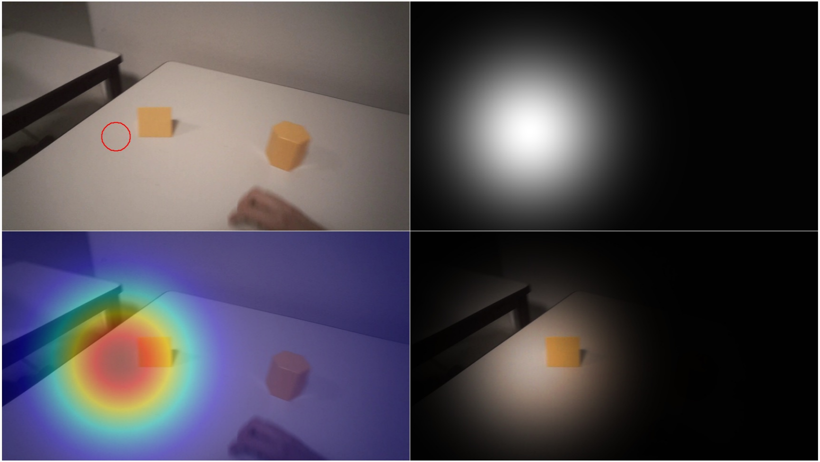
$$\sigma(I, d) = \frac{A}{d} \cdot \frac{width(I) tan(180\alpha\pi)}{2 tan(\beta\pi/180)}$$

$\alpha = 2^\circ$ focal vision radius, $\beta = 24^\circ$ camera opening angle, $A = 1600mm$ maximum distance.

[7] D.S. Wooding (2002). "Fixation Maps: Quantifying Eye-movement Traces". In: *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*. ETRA '02. New York, NY, USA: ACM, pp. 31–36. ISBN: 1-58113-467-3. DOI: 10.1145/507072.507078. URL: http://doi.acm.org/10.1145/507072.507078.

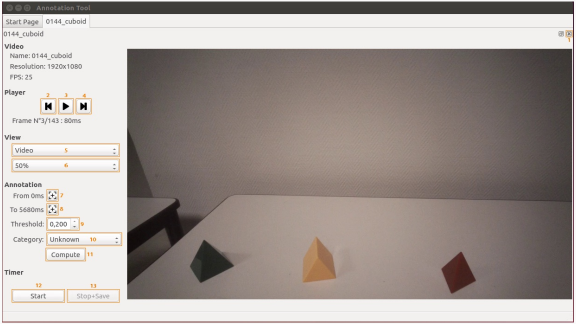
Saliency computation 2

Various visualization of the Wooding Map:



Videos Semi-Automatic Annotation

User interface of our annotation software:



Parameters: Time interval
threshold
category

Advantage: Fast

Disadvantages: Distractors induce the annotator into error

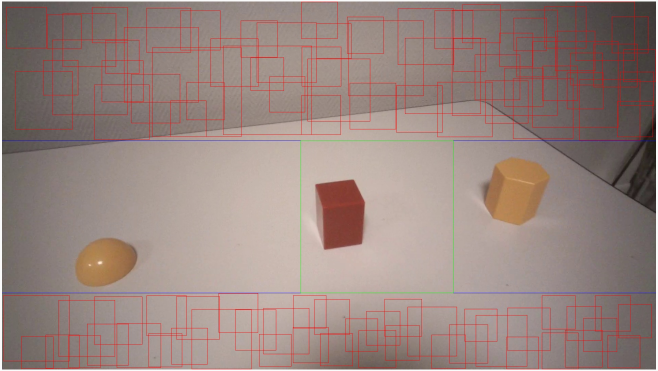
Patch Extraction and Augmentation 1

Patch Extraction example:

(1) Bounding box of the object of interest

(2) Exclusion rectangle

(3) Background patches candidates.

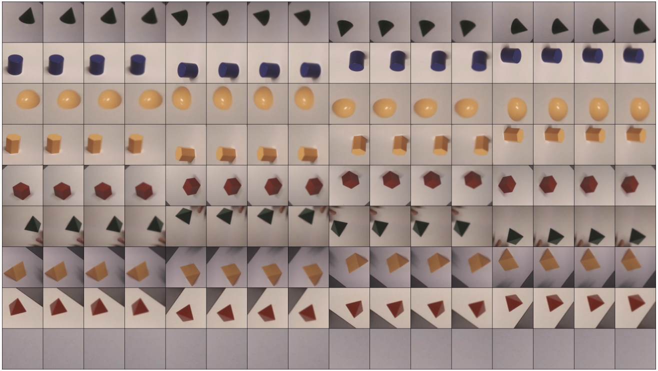


Patch Extraction and Augmentation 2

Augmentations: Label preserving transformations

Rotations none, 90°, 180°, 270°

Blur: none, 3x3, 5x5, 7x7



Patch Extraction and Augmentation 3

Number of image patches extracted:

By category for Training, Validation and Testing

	Categories	Training	Validation	Testing	Total
	Background	123 424	43 024	0	166 448
	Cone	17 824	6 352	420	24 596
	Cylinder	6 544	2 928	111	9 583
	Hemisphere	13 360	4 016	272	17 648
	Hexagonal_Prism	16 592	5 776	235	22 603
	Rectangular_Prism	24 032	6 816	620	31 468
	Rectangular_Pyramid	10 736	4 448	308	15 492
	Triangular_Prism	21 168	7 744	396	29 308
	Triangular_Pyramid	16 784	4 976	412	22 172
	Total	**250 464**	**86 080**	**2 774**	**339 318**

Similar number to ImageNet dataset[8]

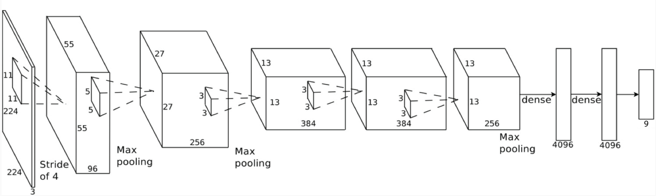
[8] O. Russakovsky et al. (2015). *ImageNet Large Scale Visual Recognition Challenge*.

Object Candidate Recognition with a Deep CNN

Network Architecture: ImageNet[9]

5 Convolutional layers: Compute image features.

3 Fully connected layers: Classify features in 9 categories



[9]A. Krizhevsky, I. Sutskever, and G.E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.* Pp. 1106–1114.

Stochastic Gradient Descent:

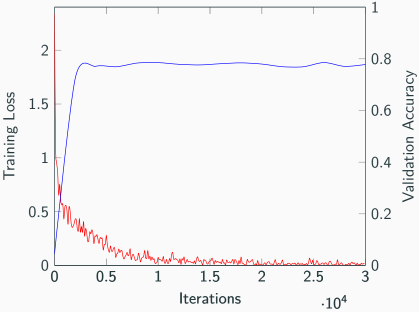
Weight update formula:

$$V_{t+1} = \mu V_t - \gamma \nabla L(D,\ W_t)$$
$$W_{t+1} = W_t + V_{t+1}$$

Learning rate $\gamma = 10^{-4}$ (fixed), momentum coefficient $\mu = 0.9$

Training progress:

1 Training loss over iterations, 2 validation accuracy over iteration



Temporal fusion

Temporal fusion: filter out distractors

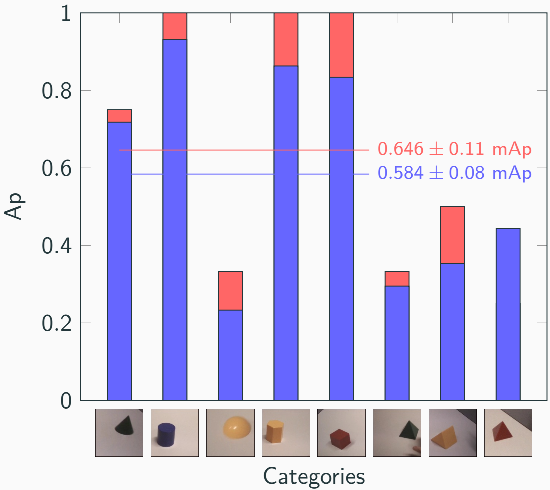
Mean fusion over the patches p_i extracted on the frames of the video V:

$$c(V) = \operatorname*{argmax}_c \{\sum_{i=1}^{N} s(p_i,\ c)\}$$

Results

Precision:

Average precision per category (Ap), mean average precision (mAp)
1 Without temporal fusion, 2 with temporal fusion.



Results

Timing of our system:

Compared to an eye-fixation time, and other methods:

	Our method	Eye fixation	Fast R-CNN	R-CNN
Saliency	5 ms			
Classify	8 ms			
Total	13 ms	250 ms	320 ms	42000 ms

Conclusion and Perspectives

Conclusion and Perspectives

We have developed a method that can:

- Select a single candidate object in the subject egocentric view, using eye-tracking.
- Recognize this objects between 8 categories with 65% mAp.
- All of it in 13ms, which is much faster that an eye fixation, and our video frame-rate.

Perspectives:

- Try other CNN.
- Adapt the method to more complex scenarios.
- Study the effect of noise in the training dataset
- Noise robust training method for Deep CNN

Acknowledgments

Many thanks to:

specially CNRS-Idex grant PEPS-Suivip and the INCIA

The students who helped

Souad Chaabouni, Marin Gutierrez, Cyril Hatchi, Antoine Pitaud, Mylene Tahar

Thank you for your attention

Any questions ?

Fully Connected layer (FC)

Fully connected neuron:

$$\widehat{Y}_n = b + \sum_{i=1}^{L} \sum_{j=1}^{H} \sum_{k=1}^{D} W_{i,j,k}^{(n)} \cdot X_{i,j,k}$$

Parameters to train: Independent between neurons

W**:** weights

b**:** bias

Design choices:

Number of output: Number of neurons in the layer
i.e. the top shape

Convolutional layer (Conv)

Convolutional neuron:

$$\widehat{Y}_{i,j,k} = b + \sum_{di=-k_1}^{+k_1} \sum_{dj=-k_2}^{+k_2} \sum_{k'=1}^{D} W^{(k)}_{di,dj,k'} \cdot X_{i+di,j+dj,k'}$$

$$(i,j) \in L \times H$$

Parameters to train: shared between neurons of the same filter depth k:

W**:** weight

b**:** bias

Design choices:

$k_1,\ k_2$**:** kernel size

Number of output: Number of neuron in the layer,
i.e. depth of the top shape

Stride: Step size between neurons on L and H

Pad: Expand input blob on L and H

g**:** Group input and output channels in g groups

Max pooling layer (Pool)

Max pooling layer:

$$\widehat{Y}_{i,j,k} = \max_{\substack{i' \in [i-k_1,\ i+k_1] \\ j' \in [j-k_2,\ j+k_2]}} \{X_{i',j',k}\}$$

Design choices:

- $k_1,\ k_2$: kernel size
- Stride: Step size between neurons on L and H (2)
- Pad: Expand input blob on L and H

Local response normalization (LRN)

Local response normalization:

$$\widehat{Y}_{i,j,k} = X_{i,j,k} \ / \ (\ b + \alpha \sum_{k'=k-n}^{k+n} (X_{i,j,k'})^2 \ )^{\beta}$$

Design choices:

- n: kernel size (5)

Hyper-Parameters:

- α: scaling parameter (10^{-4})
- β: exponent ($0,75$)
- b: (1)

Activation Functions

Threshold:

$$\widehat{Y}_{i,j,k} = \begin{cases} 1 \text{ if } X_{i,j,k} > \tau \\ 0 \text{ else} \end{cases}$$

Sigmoid

$$\widehat{Y}_{i,j,k} = \frac{1}{1 + e^{-X_{i,j,k}}}$$

The Rectified Linear Unit (ReLU)

$$\widehat{Y}_{i,j,k} = max(0,\ X_{i,j,k})$$

Soft Max

Soft max: Activations $\rightarrow$ probability distribution

$$\widehat{P}_i = e^{X_i} \; / \; \sum_{j=1}^{N} e^{X_j}$$

Loss function

The overall loss over a dataset D is:

$$L(W) = \frac{1}{|D|} \sum_{i=1}^{|D|} E(X_i, l_i) + \lambda r(W)$$

$r(W)$ is an L_2 regularization term

Loss function: for input image X_i with known label l_i:
(multinomial logistic loss)

$$E(X_i,\ l_i) = -\frac{1}{N} \sum_{j=1}^{N} log(\widehat{P}_i)\delta(\widehat{l}_i, l_i)$$

$$\delta(l, l_i) = \ 1 \text{ if } l = l_i,\ 0 \text{ else}$$

Hyper-Parameter:

- λ: Weight decay (0.0005)

ImageNet architecture details

Layer	Depth	Type	Name	Parameters	Top shape
23	8	Soft Max	prob		C
22	8	FC	ip8		C
21	7	Dropout	drop7	$ratio = 0,5$	4096
20	7	ReLU	relu7		4096
19	7	FC	ip7	$b = 1$	4096
18	6	Dropout	drop6	$ratio = 0,5$	4096
17	6	ReLU	relu6		4096
16	6	FC	ip6	$b = 1$	4096
15	5	Max pooling	pool5	$k = 3\text{x}3,\ s = 2$	6x6x256
14	5	ReLU	relu5		13x13x256
13	5	Convolution	conv5	$k = 3\text{x}3,\ nb = 256,\ pad = 1,\ b = 1$	13x13x256
12	4	ReLU	relu4		13x13x384
11	4	Convolution	conv4	$k = 3\text{x}3,\ nb = 384,\ pad = 1,\ b = 1$	13x13x384
10	3	ReLU	relu3		13x13x384
9	3	Convolution	conv3	$k = 3\text{x}3,\ nb = 384,\ pad = 1,\ b = 0$	13x13x384
8	2	LRN	norm2	$k = 5\text{x}5,\ \alpha = 10^{-4},\ \beta = 0,75$	13x13x256
7	2	Max pooling	pool2	$k = 3\text{x}3,\ s = 2$	13x13x256
6	2	ReLU	relu2		27x27x256
5	2	Convolution	conv2	$k = 5\text{x}5,\ nb = 256,\ pad = 2,\ b = 1$	27x27x256
4	1	LRN	norm1	$k = 5\text{x}5,\ \alpha = 10^{-4},\ \beta = 0,75$	27x27x96
3	1	Max pooling	pool1	$k = 3\text{x}3,\ s = 2$	27x27x96
2	1	ReLU	relu1		55x55x96
1	1	Convolution	conv1	$k = 11\text{x}11,\ nb = 96,\ s = 4,\ b = 0$	55x55x96
0	0	Data	data		227x227x3