



Analysis of visual receptive fields  
in deep neural networks

Khánh Nam NGUYỄN & David LAMBERT  
M1 BIM 2021-2022 - Second semester project report

Advisor: Denis Sheynikhovich, associate professor  
Aging in Vision and Action Lab, Institut de la Vision  
Faculté des Sciences et d'Ingénierie - Sorbonne Université

“Then she got into the lift, for the good reason that the door stood open; and was shot smoothly upwards. The very fabric of life now, she thought as she rose, is magic. In the eighteenth century, we knew how everything was done; but here I rise through the air; I listen to voices in America; I see men flying – but how it’s done I can’t even begin to wonder. So my belief in magic returns.”

V. Woolf, *Orlando: A Biography*. London, UK: Hogarth Press, 1928

/-/

“(..) In this world, with great power there must also come – great responsibility!”

S. Ditko and S. Lee, “Spider-Man #1”. In *Amazing Fantasy*, Vol. 1-15. New York, NY, USA: Marvel Comics, 1962.

# Contents

Introduction . . . . .	4
Methods . . . . .	5
1. A very brief and partial history of convolutional neural networks (1948-2012) . . . . .	5
2. Receptive field analysis in deep learning . . . . .	8
3. Implementation . . . . .	9
4. The Avatar experiment and AvatarNet . . . . .	10
Results . . . . .	12
1. AlexNet: top K images . . . . .	12
2. AlexNet: object detection and receptive field computation . . . . .	12
3. Testing feature detection hypotheses with manufactured images: a study in magenta . . . . .	14
4. Application of receptive field extraction to the analysis of the Avatar experiment . . . . .	14
Discussion . . . . .	15
Stimulus localization versus stimulus identification . . . . .	15
Discrepancy map computation time and stride adjustment . . . . .	16
AlexNet versus the state of the art . . . . .	17
Bibliography . . . . .	18
Acknowledgements . . . . .	20

## Introduction

Machine learning methods based on deep artificial neural networks, and their application to computer vision, have proven their efficiency and have found their place in research and industrial contexts, but suffer from the lack of explainability of the internal representations that these networks elaborate to reach their goal, which has earned them their reputation as “black box” algorithms.

Concern has steadily grown regarding this conceptual weakness, in a context where deep learning models are involved in increasingly varied applications, sometimes with potentially serious repercussions[1]. This concern has given rise to explainable artificial intelligence (explainable AI, or XAI) as an active field of research.

One way towards XAI would be to gain insight into how each unit of the network contributes to the global decision. Methods proposed by Zhou et al.[2] can be used to extract the visual receptive fields of a deep convolutional neural network’s units, by feeding the network locally disturbed images, and comparing the resulting activation with the baseline activation obtained from the original image.

A discrepancy map is then obtained, showing to which area of the original image a given unit reacts, thus extracting each unit’s receptive field. This method was successfully applied by Bonner & Epstein to analyze behavioral fMRI data in the context of spatial navigation[3].

The first phase of this project aims to reproduce some of the results obtained by Zhou et al. to analyze the inner workings of a neural network with an established degree of performance in image classification tasks.

These results will then be applied to a second network, closely related to the first, but trained on images from an experiment on human vision and spatial navigation by Becu et al.[4] of the Aging in Vision and Action team at the Institute of Vision, that seeks to establish a relationship between the age of a human subject and the visual cues they rely on while orienting themselves in an artificial environment.

Using a deep neural network to understand human behavior implies to first understand how the network solves the task, which means deep learning as a black box is not a suitable approach: we evaluate receptive field analysis as a solution to this explainability problem.

## Methods

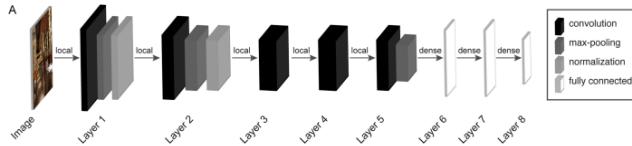
### 1. A very brief and partial history of convolutional neural networks (1948-2012)

The idea of using computers to simulate systems based on networks of neurons or neuron-inspired units is practically as old as modern computers themselves, with sources tracing it to Alan Turing's seminal paper from 1948, Intelligent Machinery[5].

The perceptron, invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt[6], is perhaps the best-known early example of an artificial neural network (NN) applied to computer vision (CV), but various reasons, including but not limited to less impressive results compared to that of rival schools of thought in the field that soon became known as artificial intelligence (AI), and insufficient computing resources, stalled this field of research until the 1980s[7].

Discoveries on the inner workings of animal visual cortices by Nobel Prize in Physiology or Medicine recipients Hubel and Wiesel in 1959[8] gave birth, approximately twenty years later, to the first convolutional neural network (CNN), where convolution filters inspired by the mechanisms of mammalian vision, which provide a way to compute the resemblance of an input with an expected pattern, were first introduced as neuron-like units. In these early attempts, the weights that connected these neurons into a network had to be adjusted manually, by trial and error, to obtain the expected result during the training phase, which hindered the efficiency of the method.

The field of supervised learning with deep CNNs made a significant jump forward in 1989, with the introduction of residual error backpropagation through the hidden layers of the deep LeNet network[9], which allowed LeCun and his colleagues to automate the weight computation, opening the way to significant advances in the field[10].



The AlexNet architecture (figure from Bonner and Epstein, 2018)

AlexNet, the deep CNN we will be considering in the rest of the present report, will turn ten years old in September 2022. It entered the ImageNet competition in 2012 and dramatically outperformed its top competitors, with a 10% lead on the second contestant[11]. A direct descendant of LeNet, this network architecture, pictured above, introduced several innovations such as depth in terms of its number of layers, repurposing graphics processing units (GPUs, essential in computing real-time images from video game engines) to dramatically

shorten computing time by parallelizing computations, introducing dropout as a regularization technique, and using a ReLU (rectified linear unit) instead of the hyperbolic tangent or sigmoid functions previously used as activation functions[12].

AlexNet classifies images: it takes an image as input, and returns as output a list of a thousand confidence scores corresponding to the classes from the hand-labeled ImageNet dataset, the top-1 score being the prediction it is most confident about. Its top-1 error rate is close to 43%: its most confident prediction is wrong 43 times out of 100.

More recent network architectures have long replaced AlexNet as the state of the art, but its level of performance, and the way it achieves it, which we will discuss in the following section, is still very significant in a number of contexts, which makes it an ideal model to probe.

In particular, the spontaneous emergence of object detectors during the training process is part of the reason why AlexNet represented such a leap forward: while learning to recognize, for example, bookshelves, the network essentially teaches itself to recognize the texture of wood, or a slim rectangle containing text as the spine of a book, without ever being explicitly told what wood is, what text is, what a book is, that it has a spine, or that book spines are likely to contain text.



Janine (right) and Roberto (left) taking their daily stroll through the Pierre & Marie Curie campus, with AlexNet’s top-5 predictions and confidence scores.

The above figure shows an input image, the associated predictions, and the relatively low confidence scores from a version of AlexNet trained on ImageNet:

- storks resemble the statues on the lawn, and the columns in the background are similar to their legs;
- a whippet is a species of dog with varying fur patterns, some of which can be mistaken for the pattern in Janine’s feathers;
- as for the lawn-mower, a prediction in which identifying lawn is involved, the bias of ImageNet-trained CNNs toward texture detection over shape detection has been established in the literature[13].



Top 5 predictions:  
 1: drake, confidence: 81.21%    2: goose, confidence: 14.47%    3: vulture, confidence: 3.40%  
 4: bustard, confidence: 0.17%    5: peacock, confidence: 0.16%

Cropped image with associated top-5 predictions and confidence scores.

The above figure pictures AlexNet’s predictions associated with a crop from the previous image. After elimination of conflicting stimuli from the input, AlexNet’s confidence in its top-1 prediction shoots up. “Drake” is a term that can be used for a dragon and a popular singer but also a male duck, the latter being more relevant to the problem at hand.

## 2. Receptive field analysis in deep learning

The core concept of the methods implemented here and based on Zhou et al.[2] is the discrepancy map, which allows us to identify a specific unit's receptive field (RF), thus localizing the area of a given input image that the target unit is most sensitive to.

First, the entire image bank, or dataset, is passed through the network, and the images resulting in the ten highest activation values of the target unit are identified by intercepting the output of the target unit.

Once these top ten images have been identified, a set of locally disturbed images is obtained from each of the ten images: an occlusion window, made of noisy pixels of a randomly chosen color, is passed over the image in a succession of steps, the number of which is determined by the size of the original image, the size of the occlusion window, and the stride, or characteristic dimension of the step, of the patch between each step of the process.

The figure below represents two examples out of the five hundred occluded images generated by applying an 11 by 11 pixel noisy patch with a stride of 11 pixels on a 256 x 240 pixels image. The noisy patch is the square in the upper left corner of the image and to the left of the cat's eye, respectively.



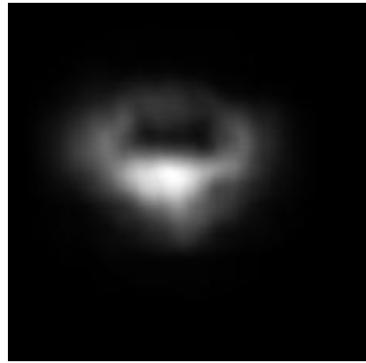
Two examples of locally disturbed images

This batch of occluded images is then passed through the network, and the resulting activation in the target unit is intercepted and compared with the baseline activation from the undisturbed original image. The location of the specific disturbance is then colored in greyscale against a dark background, with an intensity proportional to the impact of the disturbance on the unit's activation, measured by the difference between the baseline and the activation resulting from the disturbed image.



Top 10 images for a given unit with corresponding discrepancy maps (from Zhou et al., 2015)

The ten resulting discrepancy maps, one for each of the top ten images, are then calibrated (centered) and overlaid on top of each other, to obtain a best-effort estimation of the unit's receptive field.



Receptive field corresponding to the above figure (from Zhou et al., 2015)

### 3. Implementation

This project essentially consists in a Python implementation of the receptive field analysis methods from Zhou et al.[2], based only on the specific level of detail included in the paper, without using as reference any previous implementation of said methods or related ones. This implementation uses the following Python packages :

- PIL for image processing;
- the PyTorch `torch` and `torchvision` packages for deep learning;
- `sys`, `os`, and `datetime` for command-line argument input handling, file system manipulation, and execution time monitoring respectively;
- `numpy` for noise patch generation and some specific tensor manipulations;
- the `pyplot` package from the `matplotlib` library for occasional plots.

In particular, the AlexNet pretrained model from the PyTorch Hub and the accompanying AlexNet tutorial[14] allowed for the quick and fluid deployment of an operational experimental setup, without any prior background in neural networks, machine learning model deployment, or experience with PyTorch.

Additionally, the forward hooks [15] and buffers [16] implemented in the `torch`

package provided a straightforward way to intercept and affect the output of specific target units, which was instrumental in computing the discrepancy maps:

- forward hook registration attaches a user-defined function to a specific layer of the network, that is then triggered when an image is passed through the network ;
- buffer registration attaches a buffer to a specific layer, which allows the user to store any output from the forward hook inside the layer module, for later retrieval and processing.

Batch scripts were used in conjunction with Python scripts to automate most of the computation pipeline.

The results highlighted below feature selected images obtained by analysing two different networks:

- first, a version of AlexNet pretrained on the ImageNet dataset and made available by the PyTorch team through their Model Hub, a catalogue of off-the-shelf, ready-to-use model architectures and weight matrices (“AlexNet”), to validate the implementation;
- then, a version of AlexNet trained by the Aging in Vision and Action team on a dataset obtained from their Avatar experiment (“AvatarNet”), described below.

#### 4. The Avatar experiment and AvatarNet

The Avatar experiment led by the Aging in Vision and Action team at the Vision Institute[4] is aimed at studying visual cue processing during a spatial orientation task, and the prevalence of different types of cues depending on the age of the human subject under consideration. To study these decisional criteria in conditions that were as close as possible to natural conditions, Becu et al. used the Streetlab platform, a controlled experimental environment with textured walls evoking an urban setting.

This allowed them to closely monitor both the position and the eye movements of their subjects in real time using motion capture equipment and eye trackers, a setup which was then used to reconstruct images representing the point of view of the human subject at very frequent intervals during the orientation task.

Floor plan of the Streetlab experimental setup, flattened to show the textured panes used on the walls of the room.

Over the course of the experiment, two populations of subjects emerged: one that accomplished the task while relying predominantly on geometric cues (“geo”), and the other who predominantly used the landmarks represented on the wall textures, such as an ATM, a brightly-colored sign or a texture (“lmk”).

Sample image from the “geo” class showing the reconstructed environment

The images obtained from each of these two populations were then labeled using



the two classes “geo” and “lmk” which corresponded to the prevalent type of visual cue used by the subject they were associated with.

The data wrangling to obtain the Avatar dataset from the experimental data, and the training process for AvatarNet undertaken by Hachim Bani and Bilel Abderrahmane Benziane under supervision from Denis Sheynikhovich[17] preceded, and fall outside of the scope of, the present project, but can be roughly described as follows:

- duplicates or overly similar images were eliminated from the dataset using the cosine distance as a metric;
- an untrained version of the AlexNet architecture is trained on the Places dataset[18];
- the last convolutional layer of the network, and all its downstream layers, were then residually trained on the Avatar dataset, to classify the images in two classes, as described above.

Training AvatarNet took around two weeks. This training time was for the most part dedicated to training the network on the Places dataset, comprising 10 million images for a total of around 100 Gb. Training the end layers of the network on the 50 000 images, 7 Gb Avatar dataset took about an hour.

The team trained AvatarNet using a widely commercially available model of

desktop computer equipped with a quad-core CPU running at 3.6GHz, a dedicated GPU with 2 Gb of DDR3 VRAM, 32Gb of RAM, and an NVMe drive allowing for low latency during the frequent drive-RAM swaps while training on the Places dataset.

The validity of a deep CNN as a computational model for algorithmic processes in the visual cortex has previously been established by Bonner & Epstein[3], who used methods from Zhou et al.[2] applied to a network trained on the Places dataset[18] to gain insights into biological vision through *in silico* experiments.

The question considered in the second phase of the deep CNN analysis using the methods from Zhou et al. on AvatarNet is whether or not the network can be a viable candidate as an *in silico* model to reason about the experiment from Becu et al.

## Results

### 1. AlexNet: top K images

The following top 5 images were assembled using max activation as a criteria, using a subset of the Places dataset comprising around four thousand images:



Top five images from unit 6 of layer 4



Top five images from unit 122 of layer 4

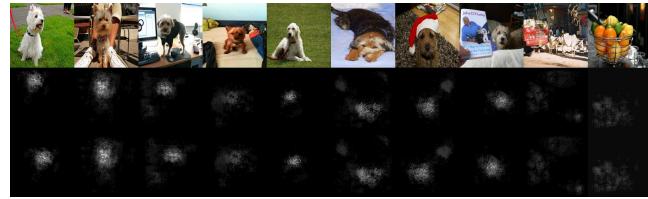
The following top 10 images were assembled using max activation as a criteria, using a richer dataset of around a hundred and twenty-three thousand images[19]:



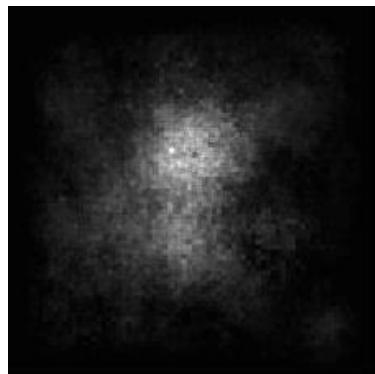
Top ten images from unit 9 of layer 11

### 2. AlexNet: object detection and receptive field computation

We applied the receptive field computation pipeline to unit 9 of layer 11.



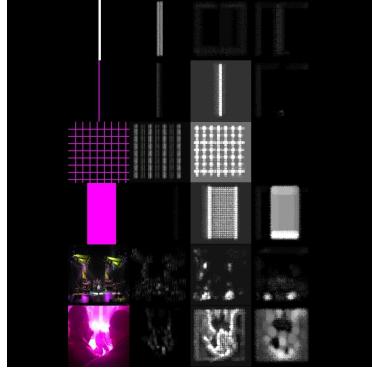
Top ten images for unit 9 of layer 11 (top) with associated discrepancy map (middle) and calibrated discrepancy maps (bottom)



Receptive field for unit 9 of layer 11 as obtained by summing the calibrated discrepancy maps as featured in the previous image

### 3. Testing feature detection hypotheses with manufactured images: a study in magenta

We used manufactured images to analyze if two units of layer 4, which reacted to similar images, were potentially redundant or actually coded for different stimuli.



Discrepancy maps for different input images (left) for unit 23 of layer 1, unit 6 of layer 4, and unit 122 of layer 4 (left to right). Lines 1 to 4 relate to manufactured images, lines 5 and 6 to images from the Places 256 dataset which show up in the top 5 for units 6 and 122 of layer 4.

### 4. Application of receptive field extraction to the analysis of the Avatar experiment

The following are samples from the set of discrepancy maps showing receptive fields from layer 11 of the features section of AvatarNet, the last post-convolutional ReLU layer.

The discrepancy maps being very dissimilar in size and shape for the same unit, summing them to compute a receptive field would not have allowed us to extract any information. Alternately, we chose to display an individual discrepancy map overlaid on top of the original image, to help discern the area that resulted in the target unit's activation.



Roster showing top 10 images for unit 4 of layer 11 (top) and their overlay with their discrepancy maps (bottom)



Roster showing top 10 images for unit 5 of layer 11 (top) and their overlay with their discrepancy maps (bottom)



Roster showing top 10 images for unit 30 of layer 11 (top) and their overlay with their discrepancy maps (bottom)

## Discussion

### Stimulus localization versus stimulus identification

While the receptive field of the unit chosen by Zhou et al.[2] to illustrate their method gives a clear image of the feature detected by the unit, we were not able to obtain results of a similar degree of precision by applying their exact method, for several reasons.

While this method might be relevant and effective for a circular stimulus, we believe the relevance of summing discrepancy maps is diminished in the following cases:

- more diffuse stimuli, such as color;
- stimuli with a higher degree of semantic complexity such as a dog's face identified by unit 9 of AlexNet's layer 11, or a landmark such as the ATM detected by unit 5 of layer 11 of AvatarNet;
- linear stimuli of differing directions such as the separation between wall and ceiling, picked out by AvatarNet's unit 30 from layer 11.

We posit that, in these cases, the receptive field obtained by summing the discrepancy maps, as in the case of unit 9 of AlexNet's layer 11 above, would not allow for feature identification.

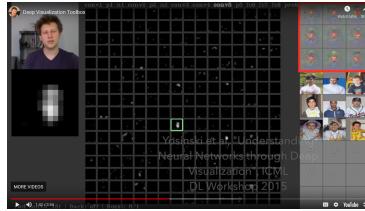
Additional *in silico* experiments, based on manufactured images, were required to precisely attribute feature detection to a specific unit, as evidenced by the case of units 6 and 122 of layer 4 of AlexNet, who encode magenta contrast (unit 6) versus magenta presence (unit 122).

Furthermore, by summing together these discrepancy maps, the main insight brought by the discrepancy maps, that is to say the location, coarse as it is, of the detected stimulus in the original image, is lost in the summing operation, whereas overlaying an individual discrepancy map with its original image at least allows us to visualize the activation peaks on top of the original image, thus giving a lead on which stimuli to test for or discriminate against in further investigation using manufactured images.

As such, the application of the receptive field computation pipeline from Zhou et al.[2] to AvatarNet doesn't allow us to meaningfully validate or invalidate the

relevance of the network as a model of human decisional criteria in the spatial navigation task from Becu et al.[4].

However, while it is true that localizing the stimulus source in the original image is insufficient, we suggest, to obtain a more precise diagnosis, to implement deconvolutional networks, a technique proposed in Zeiler and Fergus, 2014[20], that allows not only to locate the feature responsible for the unit activation, but also to visualize it in a much more straightforward manner. The figure below illustrates the technique as implemented by Yosinski et al. (2015)[21] in their Deepvis toolbox.



Screenshot from Yosinski et al., (2015)

#### **Discrepancy map computation time and stride adjustment**



Comparison between discrepancy maps using a coarse stride of 11 (top) and a finer stride of 3 (bottom) from the same original image

Logging the top ten images and obtaining the associated discrepancy maps of all 256 units of this layer took about three days of computation on a light notebook equipped with a quad core CPU running at 2.8 GHz, 16 Gb of RAM, and no dedicated GPU.

This computation time of three days already represents a compromise between a relatively reasonable computation time and the quality of the discrepancy maps:

as illustrated above, the discrepancy maps are coarser than those pictured above in the AlexNet analysis section.

The images from the Avatar dataset being larger than those of the dataset used to analyze AlexNet, using the same occlusion stride would have produced an exaggerated amount of occluded images, which is why we had to adjust the occlusion stride by a factor close to 3, thus cutting down computation time by a factor close to 9.

As a consequence, we conservatively estimate that obtaining all 2,560 discrepancy maps for AvatarNet with similar resolution to those obtained for AlexNet would have taken about four weeks of computation on the same hardware.

Using different hardware, or optimizing the code by, for example, passing images through the network in batches instead of one after the other, or using lighter data structures while sorting the images by maximum activation value in order to minimize computational overhead, are possibilities which could be explored in the future.

#### **AlexNet versus the state of the art**

While AlexNet's performance is far from that of today's state of the art networks based, for example, on the Transformers architecture also built on top of PyTorch, using AlexNet-based networks such as AvatarNet remains relevant, since their performance levels are sufficient, and considering the fact that an extensive toolbox has already been developed to analyse their behaviour.

However, one can imagine that at least some of the explainability methods discussed above would be easily transposable to more modern networks, thus keeping the door open to future exploratory work using these potentially faster and less computationally expensive networks.

## Bibliography

- [1] F. Urbina, F. Lentzos, C. Invernizzi et al. “Dual use of artificial-intelligence-powered drug discovery”. *Nature Machine Intelligence* no. 4, pp. 189–191, 2022. <https://doi.org/10.1038/s42256-022-00465-9>
- [2] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba. “Object detectors emerge in deep scene CNNs” In Proc. International Conference on Learning Representations (ICLR), 2015 <https://arxiv.org/pdf/1412.6856.pdf>
- [3] M.F. Bonner, R.A. Epstein. “Computational mechanisms underlying cortical responses to the affordance properties of visual scenes”. *PLoS Computation Biology* 14 (4), 2018.
- [4] M. Bécu, D. Sheynikhovich, G. Tatur, C. Agathos, L.L. Bologna, J.A. Sahel JA, A. Arleo. “Preference for geometric spatial cues during real-world navigation”. *Nature Human Behaviour*, 4(1), pp. 88-99, 2022. <https://pubmed.ncbi.nlm.nih.gov/31548677/>
- [5] A.M. Turing. “Machine Intelligence”, in *The Essential Turing: The ideas that gave birth to the computer age*, J.B. Copeland, Ed. Oxford: Oxford University Press, pp. 395-432
- [6] F. Rosenblatt. “The Perceptron - a perceiving and recognizing automaton” in *Report 85-460-1*. Ithaca, NY, USA: Cornell Aeronautical Laboratory, 1957
- [7] S. Russell, P. Norvig. “Artificial Intelligence: A Modern Approach”, 4th edition. London, UK: Pearson, 2020
- [8] D.H Hubel, T.N. Wiesel. “Receptive Fields of Single Neurones in the Cat’s Striate Cortex.” in *Journal of Physiology* 148 (3), Oct. 1959. doi: 10.1113/jphysiol.1959.sp006308
- [9] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel. “Backpropagation applied to handwritten zip code recognition”. In *Neural computation* 1(4), pp. 541-551. Boston, MA, USA: MIT Press, 1989
- [10] A. Karpathy, “Deep Neural Nets: 33 years ago and 33 years from now”, March 2022. [Online]  
Available: <http://karpathy.github.io/2022/03/14/lecun1989/> [Accessed May 22, 2022]
- [11] Large Scale Visual Recognition Challenge 2012 (ILSVRC2012), “ImageNet 2012 results”, September 2012. [Online]  
Available: <https://image-net.org/challenges/LSVRC/2012/results.html> [Accessed May 22, 2022]
- [12] A. Krizhevsky, I. Sutskever, G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, 2012. [Online]  
Available: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf> [Accessed May 22, 2022]

- [13] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, W. Brendel, “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness”, Nov. 29 2018 [Online]  
 Available: <https://arxiv.org/abs/1811.12231v2> [Accessed May 22, 2022]
- [14] PyTorch Team, “AlexNet” [Online]  
 Available: [https://pytorch.org/hub/pytorch\\_vision\\_alexnet/](https://pytorch.org/hub/pytorch_vision_alexnet/) [Accessed May 22, 2022]
- [15] PyTorch Team, “register\_forward\_hook” [Online]  
 Available: <https://pytorch.org/docs/stable/generated/torch.nn.Module.html> [Accessed May 22, 2022]
- [16] PyTorch Team, “register\_buffer” [Online]  
 Available: <https://pytorch.org/docs/stable/generated/torch.nn.Module.html> [Accessed May 22, 2022]
- [17] H. Bani, “Utilisation de l’apprentissage profond pour comprendre l’effet de l’âge sur l’utilisation des repères visuels pour la navigation humaine”, Ecole Centrale de Marseille, 2021
- [18] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva and A. Torralba, “Places: A 10 Million Image Database for Scene Recognition,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452-1464, 1 June 2018 doi: 10.1109/TPAMI.2017.2723009
- [19] COCO dataset, 2017 [Online]  
 Available: <https://cocodataset.org/#download> [Accessed March 31, 2022]
- [20] M. D. ZEILER, R. FERGUS, “Visualizing and Understanding Convolutional Networks”, in *Computer Vision - EECV 2014. Lecture Notes in Computer Science*, D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars, Eds. Vol. 8689, Springer, 2014 DOI: [https://doi.org/10.1007/978-3-319-10590-1\\_53](https://doi.org/10.1007/978-3-319-10590-1_53)
- [21] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, “Understanding Neural Networks Through Deep Visualization” [Online] Available: <https://yosinski.com/deepvis> [Accessed May 20, 2022]

## Acknowledgements

The authors of this report would like to thank:

- the LCQB pedagogical team, in particular Alessandra Carbone, Martin Weigt and Émilie Auger, for their support and attention;
- Denis Sheynikhovich and Bilel Abderrahmane Benziane, for the invaluable help they have provided throughout this first contact with the strange new world of scientific research;
- Janine and Roberto, for the use of their likeness.

David Lambert would additionally like to thank:

- his family, whether blood-related, extended, chosen, or any linear combination of the three;
  - Nada Mimouni and Cédric Du Mouza, associate professors, and Alexandre Lescaut, registrar, at Cnam Paris;
  - Christine Huynh, Migration project manager at Simplon.co;
  - the internet;
- ... for their unwavering support, attention, and care.