LGN-CNN: a biologically inspired CNN architecture

Federico Bertoni, Giovanna Citti, Alessandro Sarti

To cite this version:

Federico Bertoni, Giovanna Citti, Alessandro Sarti. LGN-CNN: a biologically inspired CNN architecture. Movement Disorders, 2021, 10.1016/j.neunet.2021.09.024 . hal-03416354

LGN-CNN: a biologically inspired CNN architecture

Federico Bertoni · Giovanna Citti · Alessandro Sarti

Abstract

In this paper we introduce a biologically inspired Convolutional Neural Network (CNN) architecture called LGN-CNN that has a first convolutional layer composed by a single filter that mimics the role of the Lateral Geniculate Nucleus (LGN). The first layer of the neural network shows a rotational symmetric pattern justified by the structure of the net itself that turns up to be an approximation of a Laplacian of Gaussian (LoG). The latter function is in turn a good approximation of the receptive field profiles (RFPs) of the cells in the LGN. The analogy with the visual system is established, emerging directly from the architecture of the neural network. A proof of rotation invariance of the first layer is given on a fixed LGN-CNN architecture and the computational results are shown. Thus, contrast invariance capability of the LGN-CNN is investigated and a comparison between the Retinex effects of the first layer of LGN-CNN and the Retinex effects of a LoG is provided on different images. A statistical study is done on the filters of the second convolutional layer with respect to biological data. In conclusion, the model we have introduced approximates well the RFPs of both LGN and V1 attaining similar behavior as regards long range connections of LGN cells that show Retinex effects.

Keywords CNN · LGN · Visual system · Retinex theory · Minimal functional symmetry properties

---

*This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 754362.
**This project has received funding from GHAIA - Marie Skłodowska-Curie grant agreement No 777822.

Federico Bertoni*
Sorbonne Université, Paris, France
E-mail: federico.bertoni4@unibo.it

Giovanna Citti**
Dipartimento di Matematica, Università di Bologna, Italy.
E-mail: giovanna.citti@unibo.it

Alessandro Sarti**
CAMS, CNRS - EHESS, Paris, France.
E-mail: alessandro.sarti@ehess.fr

1 Introduction

1.1 Architecture of the visual system and CNNs

The visual system is composed by many cortices that elaborate the visual signal received from the retina via the optical nerve. Each cortex receives information from other cortices, processes it through horizontal connectivity, forward it to higher areas and send feedback to previous ones. The structure is very complex and not totally ordered as physiologically described for example in (Hubel 1988). Geometrical model of the first visual cortex, we refer to (Ferraro and Caelli 1994), (Petitot 2009), (Citti and Sarti 2006).

The first neural nets have been inspired by a simplification of this structure, and present a hierarchical structure, where each layer receives input from the previous one and provides output to the next one. Despite this simplification, they reached optimal performances in processes typical of the natural visual system, as for example object-detection (Redmon et al. 2016), (Ren et al. 2015) or image classification (He et al. 2016), (Simonyan and Zisserman 2014).

More recently relations between CNNs and human visual system have been widely studied, with the ultimate scope of making the CNN even more efficient in specific tasks. A model of the first cortical layers described as layers of a CNN has been studied in (Serre et al. 2007). In (Yamins and DiCarlo 2016) and in (Yamins et al. 2013) the authors were able to study higher areas by focusing on the encoding and decoding ability of the visual system. Recurrent Neural networks have been introduced to implement the horizontal connectivity (as for example in (Sherstinsky 2018)), or feedback terms (for example in (Liang and Hu 2015)). A modification of these nets, more geometric and more similar to the structure of the brain, have been recently proposed in (Montobbio et al. 2019).

It is well known that both V1 RFPs and the first convolutional layer of a CNN are mainly composed by Gabor filters. We refer to (Daugman 1985), (Jones and Palmer 1987), (Lee 1996), (Petitot 2009) for the visual system and to (Yamins and DiCarlo 2016), (Yamins et al. 2013), (Girosi et al. 1995) for properties of CNNs.

Biological based models of V1 in terms of Gabor filters have been made in (Zhang et al. 2019) and (Serre et al. 2007) and the statistic of the RFPs of a macaque's V1 was studied in (Ringach 2002), but a comparison between these results and the statistics of learned filters is still missing.

1.2 Invariance properties in CNNs

Gabor invariance properties are mainly invariance with respect to translation and rotations. CNNs are translation equivariant since they are defined in terms of convolutional kernels (see (Cohen and Welling 2016), (Cohen et al. 2018)). Rotation invariance properties can be imposed either obtaining the whole bank of filters from a learned one (Marcos et al. 2016) and (Wu et al. 2015), or rotating any test image (Fasel and Gatica-Perez 2006), (Dieleman et al. 2015), (Dieleman et al. 2016). A different kind of pooling or kernel procedure are used in (Laptev et al. 2016) and (Gens and Domingos 2014), while (Barnard and Casasent

1991) studied invariances with respect to other feature spaces.

1.3 LGN, Retinex and contrast perception

In the human visual system, the process in V1 operated by Gabor filters, is preceded by a preprocessing operated by radially symmetric families of cells, both in the retina and in the LGN (see (Hubel 1988)). The RFPs of cells can be approximated by a LoG which is rotational symmetric (for a review see for example (Petitot 2009)). It does not seem that an analogous layer is present in classical CNN, but it is known that it is crucial for human contrast perception. It has been investigated in Retinex theory formulated in 1964 by E. H. Land in (Land 1964) and further developed by E. H. Land and J. McCann in (Land and McCann 1971). Several developments are due to (Brainard and Wandell 1986), (Provenzi et al. 2005) and (Lei et al. 2007) among others. An analysis of retinal and cortical components of the Retinex theory has been performed by several research teams in the past years (for a review see e.g. (Yeonan-Kim and Bertalmío 2017), (Valberg and Seim 2013)) and it has been shown that the Magnocellular cells are the ones involved in contrast perception (see e.g. (Enroth-Cugell and Robson 1966), (Solomon et al. 2006)). Variational approaches have been proposed by (Kimmel et al. 2003), (Morel et al. 2010) and (Limare et al. 2011). A geometrical model which makes a first step in relating the architecture of the visual system and invariance of RFPs has been presented in (Citti and Sarti 2015). The action of radially symmetric RFP is interpreted as a LoG, while the horizontal connectivity is modeled as an inverse of the Laplacian operator, and allows to recover the given image up to a contrast shift.

1.4 Our contribution

In the present paper, we introduce a new neural architecture inspired by the structure of visual cortices and study the properties of the filters of the first and second layers. The first layer contains a single filter and models the LGN. We show that it has the same radially symmetric shape, as LGN receptive profiles and is able to reproduce a Retinex effect. Then we show that the statistics of filters of the second layer much better fits the observed distributions, comparing with the esperimental results of (Ringach 2002). The paper is organized as following.

In Section 2 we recall the structure of LGN and V1 and the RFPs of their cells. We described an interpretation of the Retinex model given by Morel (Morel et al. 2010) and a statistical study on the RFPs of the distributions of V1 cells in a macaque by Ringach (Ringach 2002).

In Section 3 we introduce our LGN-CNN architecture and describe it in detail. The new architecture is implemented in Section 4. It provides classification performances comparable to the classical one, but it enforces the development of rotation symmetric filter during the training phase. As a result, the filter is a good approximation of RFP of LGN.

In Section 5 we test our filter on contrast perception phenomena. Indeed, we use the learned kernel to repeat and justify the Retinex model. We test the model on a number of classical Retinex illusion, comparing with the results of (Morel et al. 2010).

Thus, in Section 4.3 we show that the filters in the second layer of the net mimics the shape of Gabor filters of V1 RFPs. We have already recalled that this happens also in standard CNN. We compare the statistics of the filters with the results obtained on the RFPs of a macaque's V1 from the work of Ringach (Ringach 2002). By construction both in our net and in the visual system the second layer do not exhibit LoG, which are collected in the first layer. Also, other statistical properties of the filters are present in our LGN-CNN.

The proof of the symmetry of the filter of the first layer is collected in appendix A. It is provided in the simplified case of a neural network composed by only a layer composed by a single filter.

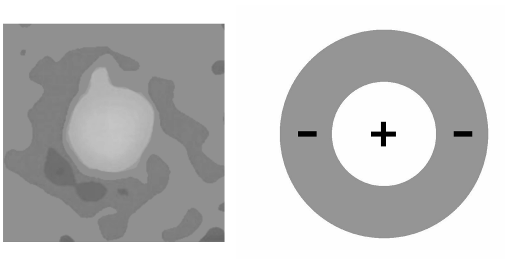


Fig. 1: On the left: RFP of an LGN cell where the excitatory area is in white and the inhibitory one is in gray. On the right: Its approximation by a LoG. *From*: (DeAngelis et al. 1995).

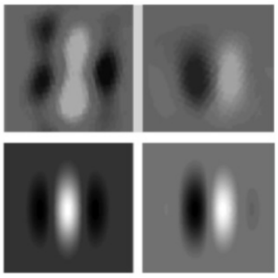


Fig. 2: First row: RFPs of two simple cells of V1 where the excitatory area is in white and the inhibitory one is in black. Second row: their approximations by Gabor functions. *From*: (Sarti and Citti 2011).

2 The visual system

The visual system is one of the most studied part of the brain. We describe here the aspects important for our study and refer the reader for a more general description to (Sundsten and Nolte 2001), (Jessell et al. 2000).

The retina is a light-sensitive layer of tissue which receives the visual stimulus and translates it into electrical impulses. These impulses first reach the LGN whose cells preprocess the visual stimulus. Then the impulse is processed by the cells of V1, whose output is taken in input to all the other layers of the visual system.

We are mainly interested in the cells of the LGN and in the simple cells of V1. Each cell receives the electrical impulse from a portion of the retina Ω called receptive field (RF). The RF of each cell is divided in excitatory and inhibitory areas which are activated by the light and that can be modeled as a function $\Psi : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ called receptive field profile (RFP). Thus, if the excitatory areas are activated the firing rate of the cell increases whereas it decreases in case of inhibitory areas activation. Figure 1 shows the RFP of an LGN cell that can be modeled by a LoG.

2.1 LGN RFP: neural interpretation of Retinex model

The Retinex algorithm, introduced in (Land and McCann 1971), mimics the contrast invariant process performed by our visual system and associates to an image $I : \Xi \subset \mathbb{R}^2 \to \mathbb{R}$ the perceived image $\widetilde{I}$.

In (Morel et al. 2010) and (Limare et al. 2011) the authors have formalized the Retinex theory as the solution of the following discrete Poisson PDE

$$-\Delta_d \widetilde{I} = M(I) \tag{1}$$

where Δ_d is the classical discrete Laplacian and M is a modified version of the discrete Laplacian.

In (Citti and Sarti 2015) a neural interpretation of the model has been introduced. The RFP of the LGN cell which takes in input the visual signal acting by convolution on it, modeled as a modified discrete LoG, $M(G_\sigma) \approx \Delta G_\sigma$, where G_σ is a Gaussian bell Hence the action on an input image is the following:

$$Out_{LGN}(I) = \Delta(G_\sigma * I) \approx \Delta I$$

The horizontal connectivity in this layer is radially symmetric and modeled as the fundamental solution $\log(\sqrt{x^2+y^2})$ whose associated operator is the inverse of the Laplacian Δ^{-1} and allows to recover the function $\widetilde{I}$:

$$\widetilde{I} = \Delta^{-1} * Out_{LGN}(I).$$

As a result

$$\Delta \widetilde{I} = Out_{LGN}(I) \approx \Delta I,$$

is the Retinex equation. In general, $\widetilde{I}$ will not coincide with I, but will differ by a harmonic function.

Our aim is to replace the action of the RFP with the filter learned by the LGN-CNN. If it is a good approximation of the associated ΔG_σ, then its inverse will allow to recover the perceived image $\widetilde{I}$ in problems of contrast perception. In Section 5 we will describe in detail the process.

2.2 Statistics of V1 RFPs

As first discovered by Daugmann, RFPs of the primary cortex V1 can be approximated by Gabor functions defined as follows:

$$\begin{aligned} h(x', y') =& A e^{(-(x'/\sqrt{2}\sigma_x)^2 - (y'/\sqrt{2}\sigma_y)^2)} \\ & \cos(2\pi f x' + \phi) \end{aligned} \tag{2}$$

where (x', y') is translated and rotated from the original coordinate system (x_0, y_0)

$$\begin{aligned} x' =& (x - x_0)\cos\theta + (y - y_0)\sin\theta \\ y' =& -(x - x_0)\sin\theta + (y - y_0)\cos\theta. \end{aligned}$$

as shown in Figure 2.

Recently, Ringach in (Ringach 2002) has proved that RFPs are not uniformly distributed with respect to all the Gabor parameters, but they have a very particular statistic. Ringach defines two coefficients n_x and n_y which estimate the elongation in x and y directions respectively

$$(n_x, n_y) = (\sigma_x \cdot f, \sigma_y \cdot f).$$

In particular, if $f = 0$ the function h in (2) simplifies to a Gaussian since the cosine becomes a constant. Otherwise it is elongated:

- Fitting a Gabor function defined in equation (2) to the RFPs;
- Comparing the results on $(n_x, n_y) = (\sigma_x \cdot f, \sigma_y \cdot f)$ plane.

Figure 10c shows the statistical distribution of RFPs of V1 cells in monkeys in (n_x, n_y) plane obtained by Ringach in (Ringach 2002). In (Barbieri et al. 2014) the authors have studied the same statistical distribution with respect to the Uncertainty Principle associated to the task of detection of position and orientation.

3 Introducing LGN-CNN architecture

In this section we introduce one of the main novelty of this paper, a CNN architecture inspired by the structure of the visual system and, in particular, takes into account LGN cells.

The retinal action in a CNN has been implemented in (Lindsey et al. 2019), where the authors have proposed a bottleneck model for the retinal output. In our model we propose a single filter layer at the beginning of the CNN that should mimic the action of the LGN. As we have already discussed in Section 2 the RFP of an LGN cell can be modeled by a LoG that acts directly on the visual stimulus. Since the LGN preprocesses the visual stimulus before it reaches V1, we should add a first layer at the beginning of the CNN that reproduces the role of the LGN.

In particular, if we consider a classical CNN we can add before the other convolutional layers, a layer ℓ^0 composed by only one filter Ψ^0 of size $s^0 \times s^0$ and a ReLU function. Note that after the first layer ℓ^0 we will not apply any pooling. In this way taking a classical CNN and adding ℓ^0 will not modify the structure of the neural network and the number of parameters will only increase by $s^0 \times s^0$. Furthermore, Ψ^0 will prefilter the input image without modifying its dimensions; this behavior mimics the behavior of the LGN which let the neural network to be closer to the visual system structure. Figure 3 shows a scheme of the first steps of the visual pathway (i.e., LGN and V1) in parallel with the first two layers ℓ^0 and ℓ^1 of the LGN-CNN architecture.

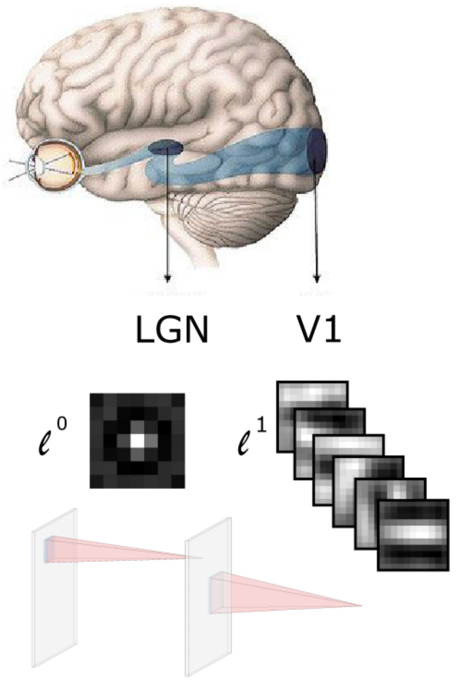


Fig. 3: Scheme of the LGN and V1 in parallel with the first two layers ℓ^0 and ℓ^1 of the LGN-CNN architecture.

The theoretical idea behind this structure can be found in a simple result on rotational symmetric convex functionals. In particular, we recall that a rotational symmetric convex functional F has a unique minimum ω. Since F is rotational symmetric, $F(\omega \circ g) = F(\omega)$ for a rotation g. Thus, since the minimum is unique, $\omega = \omega \circ g$, implying the rotational symmetry of ω. There are several results on symmetries of minimum for functionals as for example in (Lopes 1996), (Gidas et al. 1981). Our aim is to extend these results in the case of CNNs in particular on our architecture that we name as Lateral Geniculate Nucleus Convolutional Neural Network (LGN-CNN).

We will also show that the Gabor-like filters in the second convolution layer, reprojected in the (n_x, n_y) plane introduced by Ringach and recalled above, satisfy the same properties of elongation which characterizes the RFPs of simple cells in V1. This analysis should enforce the link between our architecture and the visual system structure, at least as regards simple cells in V1.

4 Applications of LGN-CNN

4.1 Settings

In this Section we describe the settings for testing our architecture. We use MatLab2019b for academic use.

We train our LGN-CNN architecture on a dataset of natural images called STL-10 (see (Coates et al. 2011)) that contains 5000 training images divided in 10 different classes. We have modified the training set in the following way:

- Changing the images from RGB color to grayscale color using the built-in function *rgb2gray* of MatLab;
- Applying a circular mask to each image, leaving unchanged the image in the circle and putting the external value to zero;
- Rotating each image by 5 random angles augmenting the dataset to 25000 images; thanks to the previous step no boundary effects are present;
- Cropping the 64×64 centered square that does not contain the black boundaries;
- Subtraction of the mean value in order to have zero mean input images.

Thus, after these steps we have obtained a rotation invariant training set composed by 25000 64×64 images. We have applied the same steps to the test set but we have rotated each image to just one random angle. Since the images are 64×64 we have decided to use quite large filters in the first and second layer (7×7 and 11×11 respectively) in order to obtain more information about their shapes.

Figure 4 shows the architecture of our CNN. Let us note that between each convolutional layer and its ReLU function there is a batch normalization layer b with the same size of the number of filters of the corresponding convolutional layer. The input is an image of size $64 \times 64 \times 1$. Then there is the first convolutional layer ℓ^0 composed by only Ψ^0 of size 7×7 followed by a ReLU R. After ℓ^0 the batch normalization layer performs a normalization similar to the one that the retina performs as described in (Carandini and Heeger 2012). Indeed, one of the main difference is the subtraction of the mean value μ performed by the batch normalization layer defined as follows

$$\widehat{x_i} = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}},$$

where x_i is the element to normalize, μ is the mean value of the batch, σ is the standard deviation of the batch and ϵ is a small value that prevents bad normalizations in case of really small standard deviations. However, since the input images have zero mean then the convolution with Ψ^0 have still zero mean. Thus, the batch normalization layer between ℓ^0 and ℓ^1 has similar characteristics as the biological one. On the other hand,

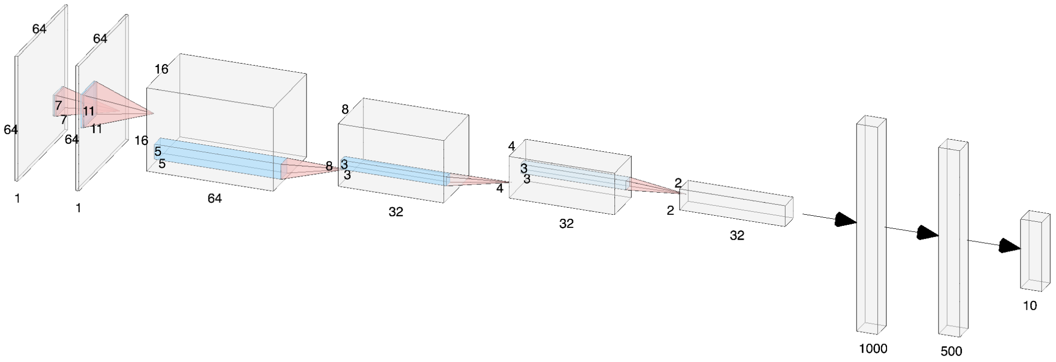


Fig. 4: Architecture of LGN-CNN.

the two approaches differ since in the batch normalization layer the statistical parameters (μ, σ) are calculated channel-wise over all input of the batch, while in the normalization described in (Carandini and Heeger 2012) the σ value is calculated from a single input instance over a restricted spatial neighborhood. However, we expect that the final result will not differ heavily applying the batch normalization layer.

Then, the second layer ℓ^1 composed by 64 filters of size 11 × 11 receives as input a matrix of the same size of the image. Note that the stride is 2 and that the spatial dimensions half. After that we apply a ReLU and a max POOLING p_m^2 with squares of size 2 × 2. The third convolutional layer ℓ^2 is composed by 32 filters of size 5 × 5 × 64 and it is followed by a ReLU and a max POOLING. Then, we apply a convolutional layer ℓ^3 composed by 32 filters of size $3 \times 3 \times 32$ followed by a ReLU and a max POOLING. The last convolutional layer ℓ^4 has the same filters as ℓ^3 followed by a ReLU and a max POOLING. Eventually three fully-connected (FC) layers of size 1000, 500 and 10 respectively are applied giving as output a vector of length 10. Finally, we apply a softmax σ, $\sigma(x)_z = \frac{e^{x_z}}{\sum_k e^{x_k}}$ where x is the output of FC^3, in order to obtain a probability distribution over the 10 classes. The functional that models this neural network is the following

$$\begin{aligned} F(I) :=&(\sigma \circ FC^3 \circ FC^2 \circ FC^1 \circ \\ &p_m^2 \circ R \circ b \circ \ell^4 \circ p_m^2 \circ R \circ \\ &b \circ \ell^3 \circ p_m^2 \circ R \circ b \circ \ell^2 \circ \\ &p_m^2 \circ R \circ b \circ \ell^1 \circ R \circ b \circ \ell^0)(I) \end{aligned} \tag{3}$$

A cross-entropy loss for softmax function defined as in equation (4) is applied to the functional (3) where $\widetilde{z}$ is the label selected by the neural network and $y(I)$ is the true label.

$$\begin{aligned} L(F(I), y(I)) =& \log(\sum_z e^{(F_z(I) - F_{\widetilde{z}}(I))}) + \\ &F_{\widetilde{z}}(I) - F_{y(I)} \end{aligned} \tag{4}$$

We have trained the neural network for 30 epochs with an initial learning rate of 0.01, a learning rate drop factor of 0.97 and a piecewise learning rate schedule with a learning rate drop period of 1. The mini batch size is 128 with an every-epoch shuffle, the L2 regularization term is 0.02 and the momentum is 0.9.

In Table 1 there are summarized the mean performances over 10 different trainings of

LGN + 4CL	LGN + 8CL	LGN + 12CL
70.41%	72.73 %	73.61 %

Table 1: Mean performances of several architectures over 10 different trainings with LGN layer plus 4, 8 and 12 other convolutional layers (CL).

three CNN architectures with the LGN layer ℓ^0 plus several convolutional layers (4, 8 and 12 respectively). As expected, the performance increases when the architecture is deeper. We stress out that the results obtained on the LGN layer and the first convolutional layer are not affected from the rest of the neural network. Indeed, it is possible to add more layers and build more deeper architectures to obtain better results on the classification task without losing the properties of the first layers of the LGN-CNN. Since our focus was to analyze the structures that arises in the first two layers, we have not further investigated the performance of the CNN.

4.2 The first layer of LGN-CNN

After the training phase we analyze the neural network focusing on the first layer in this Section. Figure 5 shows the filter Ψ^0 and 5b shows its approximation with minus the LoG.The two have a high correlation of 95.21% computed using the built-in function of MatLab *corr2*.

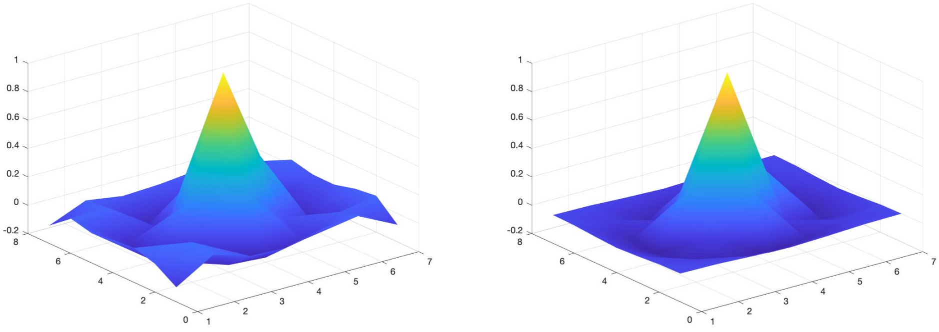
Moreover, Figure 6a shows the 2D image of Ψ^0 obtained after the training phase. Then, in Figures 6b and 6c we plot a $2D$ approximation of Ψ^0 as a 280 × 280 filter and minus the LoG in which the rotational symmetric pattern is clearer.

We would like to point out that in the LGN there exists both cells with on-center/off-surround as well as off-center/on-surround RFPs. In order to model these kinds of cells we have modified the current architecture by adding a second filter in the layer ℓ^0. Since the STL10 dataset contains natural images with many contours we should expect to have oriented filters. This is not the case and Figure 7 shows the on-center/off-surround and off-center/on-surround obtained after the training phase. For simplicity, we have considered the model with a single filter.

In order to quantify the rotational properties of the filter, we compute the correlation between the filter obtained in the first layer of the CNN and a new one, obtained via a rotation invariance symmetrization. We will test the behavior of the first layer of LGN-CNN for different values of the L2 regularization term and adding more convolutional layers. In particular, we added to the previous architecture described in Section 4.1 two convolutional layers composed by 32 filters of size 3 × 3 × 32 (each one followed by a batch normalization layer b and ReLU R) after the third layer ℓ^3. We have added other two convolutional layers with the same characteristics after the layer ℓ^4 .

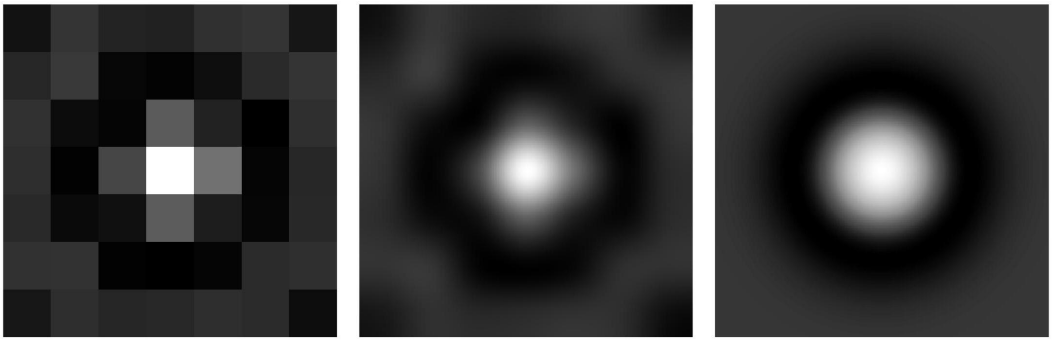
The normalized rotational invariant filter is obtained by the following procedure:

- Using the function *imresize* with scale 3 and bilinear method to enlarge the filter;
- Rotating the filter with *imrotate* by 360 discrete angles between 1 and 360 degrees and summing them up;
- Applying again the function *imresize* with scale 1/3 and nearest method to recover a filter with the same size of Ψ^0;



(a) *Filter Ψ^0 of first layer of LGN-CNN.* (b) *Minus the Laplacian of Gaussian (LoG).*

Fig. 5: Comparison between the filter Ψ^0 and minus the LoG. The two have a high correlation of 95.21% computed using the built-in function of MatLab *corr2*.



(a) *Filter Ψ^0 of first layer of LGN-CNN.* (b) *Approximation of the filter Ψ^0 of first layer of LGN-CNN.* (c) *Minus the Laplacian of Gaussian (LoG).*

Fig. 6: The first figure shows the 7 × 7 filter Ψ^0 of the neural network. To better visualize the filter Ψ^0, we provide an approximating 280 × 280 filter and minus the LoG.

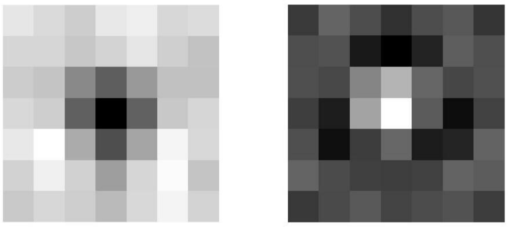


Fig. 7: On-center/off-surround and off-center/on-surround filters of ℓ^0 with 2 filters.

- Normalizing the filter by subtracting the mean and dividing it by L^2 norm.

We call the filter obtained in this way Ψ^0_S and we estimate the correlation between Ψ^0 and Ψ^0_S using the Matlab function *corr2*.

In Table 2 we have reported the correlations with different values of L2 regularization term and for two LGN-CNN architectures, the one we have introduced in Section 4.1 which is indicated by 'LGN + 4 layers' and the deeper one described in this Section indicated by 'LGN + 8 layers'. All the other training parameters are provided in Section 4.1. As we can see for 'LGN + 4 layers' ar-

L2 term	LGN + 4 layers	LGN + 8 layers
0.01	88.3 %	85.40 %
0.02	97.15 %	90.79 %
0.03	93.06 %	91.41 %
0.04	95.61%	92.58 %
0.045	93.55 %	94.79 %
0.05	95.44 %	94.11 %

Table 2: Correlation between Ψ^0 and Ψ^0_S varying the L2 regularization term and the number of layers of the LGN-CNN architecture. Best rotational symmetric filters are selected in cyan for both architectures.

	No DA	Mild DA	Hard DA
LoG corr	93.77 %	93.68 %	95.21 %
Ψ^0_S corr	93.92 %	94.16 %	97.15 %

Table 3: Correlation between Ψ^0 and its LoG approximation and between Ψ^0 and Ψ^0_S varying the data augmentation (DA) applied to the dataset.

chitecture the L2 regularization term that let Ψ^0 to be the closest rotational symmetric filter is 0.02. This is the filter we have shown in Figures 5 and 6 and we have used through the paper. The Table 2 shows that the rotational symmetry of Ψ^0 is stable with respect to variations of L2 regularization term and adding convolutional layers.

Furthermore, we have studied the properties of Ψ^0 varying the data augmentation (DA) applied to the dataset. Table 3 shows the correlation of Ψ^0 with the LoG and the correlation of Ψ^0 with Ψ^0_S with three different DA applied to the dataset. In the first one we have not applied any DA (No DA); in the second one we have randomly rotated the images of an angle of 0, $\frac{\pi}{2}$, π, $\frac{3\pi}{2}$ radiant (Mild DA); in the third one we have applied the DA described in Section 4.1 (Hard DA). From Table 3 it emerges that the introduction of rotation invariances in the dataset by rotating the images lightly affects the correlation with the LoG but gives stability to Ψ^0 allowing it to be more rotational symmetric.

In conclusion, thanks to the analysis performed on the properties of Ψ^0, we can argue that the structure of the architecture itself influences the shape of the filters and that the introduction of ℓ^0 with a single filter Ψ^0 is a good model of the LGN.

4.3 The second layer of LGN-CNN

To enforce the link between our architecture and the structure of the visual system, we have studied the filters in the second layer comparing them with some real data obtained on monkey in (Ringach 2002). Therefore, we have trained two different CNNs, an LGN-CNN defined by the functional (5)

$$\begin{aligned} F(I) :=& (\sigma \circ FC^1 \circ R \circ \ell^3 \circ p_a^4 \circ R \circ \ell^2 \\ & \circ p_m^4 \circ R \circ \ell^1 \circ R \circ \ell^0)(I) \end{aligned} \tag{5}$$

and a classical CNN defined by the functional (6) in which we have eliminated the first convolutional layer ℓ^0 and its following ReLU R, characteristic of our architecture.

$$\begin{aligned} F(I) :=& (\sigma \circ FC^1 \circ R \circ \ell^3 \circ p_a^4 \circ R \circ \ell^2 \\ & \circ p_m^4 \circ R \circ \ell^1)(I) \end{aligned} \tag{6}$$

Let us note that in both architectures ℓ^1 contains filters with Gabor shapes after training. This is a well-known result on the filters of the first convolutional layer of CNNs as for example in (Serre et al. 2007), (Yamins and DiCarlo 2016); however, the introduction of a first layer composed by a single filter does not change this behavior. Indeed, we have studied the statistical distribution

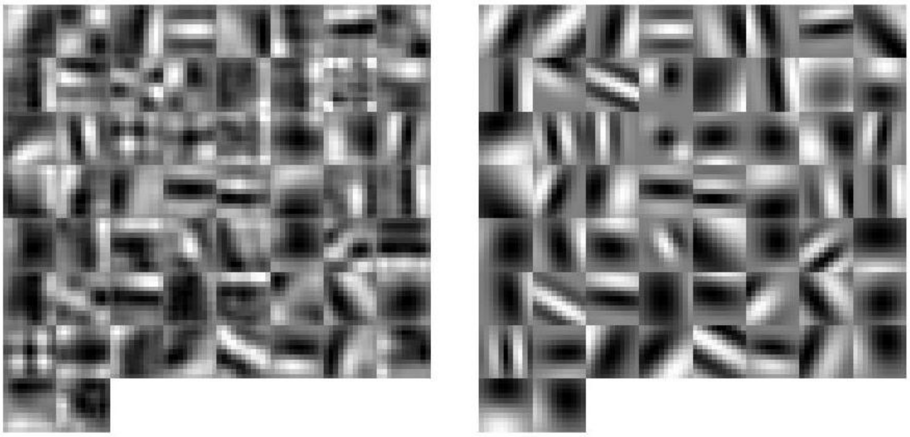


Fig. 8: On the left: filters from LGN-CNN. On the right: their approximation with the function (2).

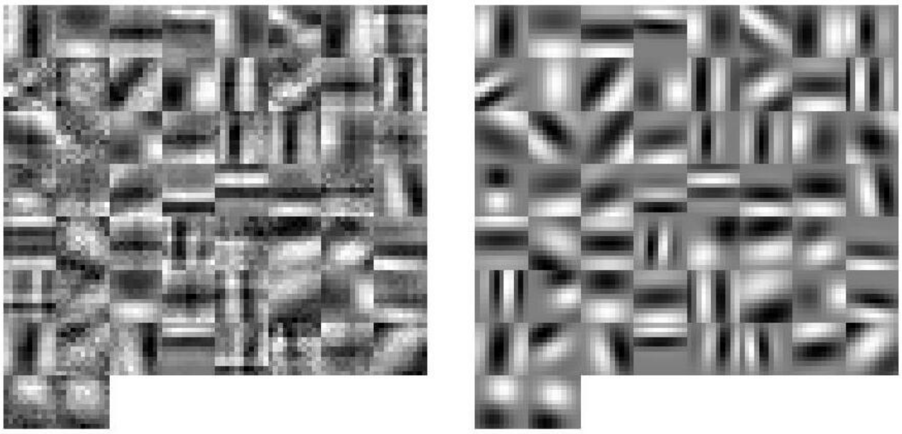


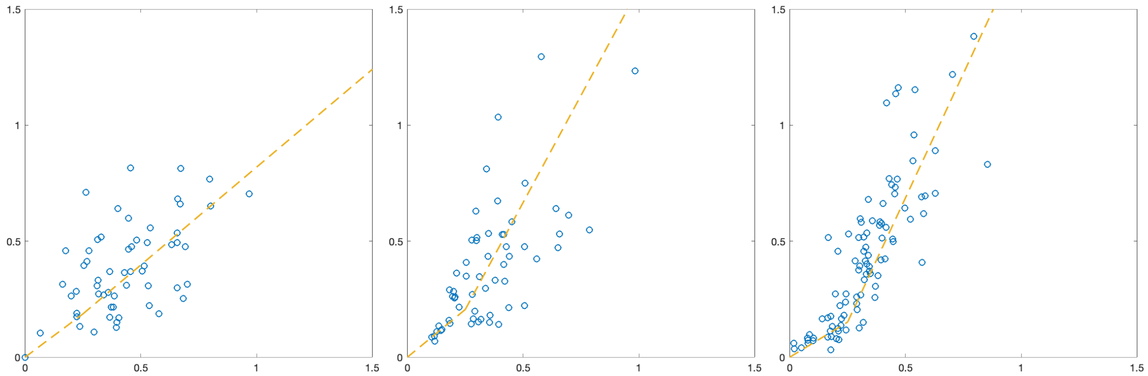
Fig. 9: On the left: filters from classical CNN. On the right: their approximation with the function (2).

of these banks of filters confronting the results with the real data of Ringach.

In the case of LGN-CNN we have not approximated the filters in ℓ^1 directly but the filters obtained by the convolution with Ψ^0.

We have approximated the filters in the banks using the function (2); Figure 8 shows some of the filters of LGN-CNN and their approximation and the same occurs in Figure 9 in the case of classical CNN. Let us note that the mean correlation estimated with the built-in MatLab function *corr2* increases from classical CNN to LGN-CNN from just 71.62% to 93.50%. This suggests that introducing the layer ℓ^0 with a single filter better regularize the filters in the following convolutional layer ℓ^1.

We follow the same step as Ringach in (Ringach 2002) by plotting in the (n_x, n_y) plane. In order to compare the plots, we looked for the distribution that best fits the



(a) *Distribution of filters in ℓ^1 of classical CNN defined by functional (6).*

(b) *Distribution of filters in ℓ^1 of LGN-CNN defined by functional (5).*

(c) *Distribution of RFPs of simple cells from (Ringach 2002).*

Fig. 10: Comparison between the statistical distribution on (n_x, n_y) plane of filters of a classical CNN, of our architecture and of RFPs of real data.

neural data. In particular, it approximates the points closer to the origin with a line $y = \alpha x$ and then it approximates the rest of the points with a line starting from the end of the previous one.

Figure 10 shows the three plots. Let us note that introducing ℓ^0 modifies the elongation of Gabor filters in ℓ^1. In particular, in classical CNN the filters are often more elongated in the x direction as we can see from the slope of the interpolating line in Figure 10a. In Figure 10b we can see that the slope changes greatly and that the filters become much more elongated in the y direction. This behavior is the same in the case of RFPs (Figure 10c) in which the distribution has a similar slope of LGN-CNN. This enforces more the link of LGN-CNN with the structure of the visual system motivating us to pursue in this direction.

5 Retinex algorithm via learned kernels

In this section we test the rotational symmetric filter on Retinex effect and contrast-based illusions. We described in Section 2 how the model of (Morel et al. 2010) have been neurally interpreted in (Citti and Sarti 2015) and applied to LoG. Our approach aims to find the reconstructed image $\widetilde{I}$ for a general operator

$$M : \Xi \subset \mathbb{R}^2 \to \mathbb{R}, \tag{7}$$

solving the following problem

$$M * \widetilde{I} = M * I. \tag{8}$$

by finding the inverse operator of M

$$\widetilde{M} : \Xi \subset \mathbb{R}^2 \to \mathbb{R}. \tag{9}$$

By definition the inverse $\widetilde{M}$ of the operator M satisfies

$$M * \widetilde{M} = \delta. \tag{10}$$

Applying the steepest descent method, we obtain an iterative process, which can be formally expressed as

$$\widetilde{M}_{t+1} = \widetilde{M}_t + dt \cdot (M * \widetilde{M}_t - \delta). \quad (11)$$

The algorithm will stop at time T when $\frac{||\widetilde{M}_{T+1} - \widetilde{M}_T||_{L^1}}{dt} < \epsilon$, for a fixed error $\epsilon > 0$. Thus, $||M * \widetilde{M}_T - \delta||_{L^1} < \epsilon$ and indeed $\widetilde{M}_T$ would be a good approximation of $\widetilde{M}$. Finally, we can compare the image I with the reconstructed one $\widetilde{I}$ and see if any Retinex effects occur. Let us note that one difference between our approach and the one proposed by Morel is that he imposed the Neumann boundary conditions to the Poisson equation whereas, in our model, the Neumann boundary conditions imposed to the filter are inherited from the inverse operator itself. Furthermore, we have faced the problem for a Laplacian operator to see if the results are similar to the Morel ones. Indeed, in our Retinex algorithm we convolve a given visual stimulus I with a fundamental solution M and we reconstruct the perceived image $\widetilde{I}$ using the inverse operator $\widetilde{M}$.

5.1 Study of the inverse operators

Firstly, we show the inverse operators obtained through the algorithm described by equation (11). We start from the classical discrete Laplacian operator and then we will move to convolutional operators, in particular a discrete LoG and the filter Ψ^0 of an LGN-CNN. In order to compare the inverse operators, we will show the 2D plots obtained by selecting a slice of the 3D inverse operator itself.

We start by comparing the inverse of the discrete Laplacian with respect to its exact inverse operator given by the function $\log(\sqrt{x^2+y^2})$. Figure 11b shows the approximation of the inverse of the Laplacian and 11a shows the exact inverse in the interval $[-5, 5]$. We can note that the approximation is close to the exact one.

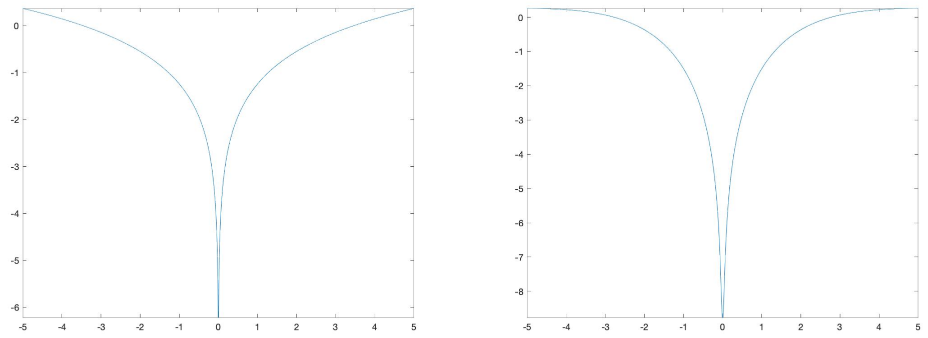
In the second row of Figure 11 we have compared the inverse operators of a LoG and Ψ^0 where Figure 11c shows the LoG inverse operator and Figure 11d shows the Ψ^0 inverse operator. Since their inverse operators are close we expect similar Retinex effects in the next Section.

5.2 Application of the algorithm

In this Section, we aim to test our algorithm and our different operators to see if Retinex effects occur. Let us note that Retinex is an algorithm that mimics our color perception and does not try to improve the image quality. Indeed, the grayscale values should modify towards our color perception intensity values.

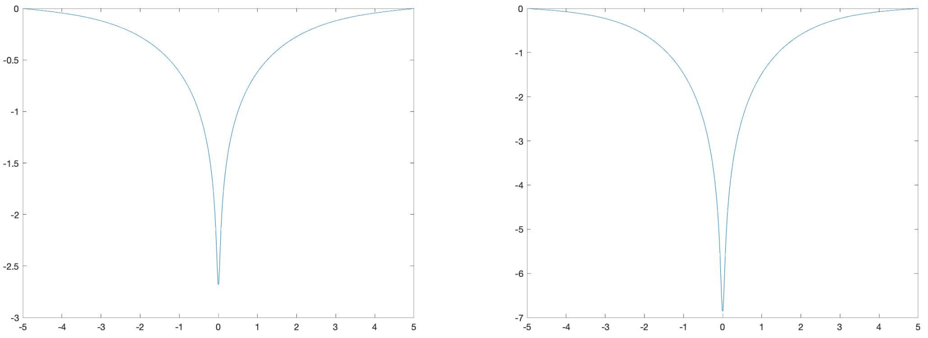
5.2.1 Circles on a gradient background

We start with a simple grayscale image (see Figure 12a) in order to see how the different operators work. We are using the same image as in (Morel et al. 2010) and (Limare et al. 2011) in which they obtain remarkable Retinex effects. It has a background that shifts from white (value 1) to black (value -1) maintaining the gradient constant from left to right. There are also two gray dots (same value 0), the left one with a brighter background, the right one with a darker one. Because of the different backgrounds, our visual system perceives the two



(a) *Exact inverse of a Laplacian* $\log(\sqrt{x^2+y^2})$ *in the interval* $[-5,5]$.

(b) *Inverse of discrete Laplacian.*



(c) *Inverse of LoG.*

(d) *Inverse of the first layer of an LGN-CNN.*

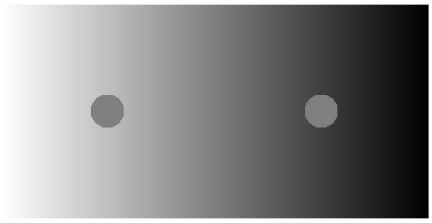
Fig. 11: Comparison between inverse operators. First row: inverse of the discrete Laplacian and its exact inverse operator. Second row: inverse operators of a discrete LoG and the first filter Ψ^0 of an LGN-CNN.

dots differently: the left one is perceived as darker with respect to its true color; the right one is perceived as brighter.

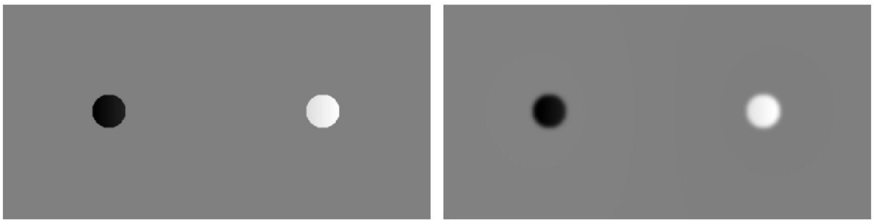
We first consider the discrete Laplacian operator since we expect to obtain the same Retinex effects as in (Morel et al. 2010). Figure 12c shows the result obtained with our method. It is clearly the same result of the experiments performed by Morel and the values obtained in the position of the dots are close to -0.9 and 0.9 (indeed really close to completely black and white dots) whereas the entire background is gray with 0 values.

Thus, we have considered the exact inverse of the Laplacian operator $\log(\sqrt{x^2+y^2})$. From Figure 12b it is clear that the Retinex effects occur to the gray dots. Furthermore we can note that the contours of the two dots are more defined w.r.t. the discrete Laplacian. Also in this case the values in the position of the dots are close to -0.9 and 0.9.

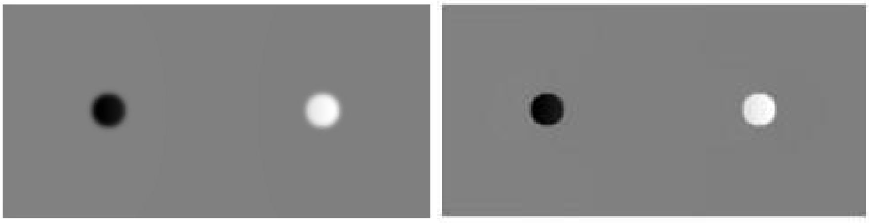
Then, we have performed our experiment with the inverse of a LoG. We can note from Figure 12d that the Retinex effects occur also in this case (with values close to -0.9 and 0.9), similarly to the Retinex effects of the



(a) *Starting image with two gray dots on a gradient background.*



(b) *Retinex effects of exact inverse of Laplacian.* (c) *Retinex effects of discrete Laplacian.*



(d) *Retinex effects of LoG.* (e) *Retinex effects of Ψ^0 of LGN-CNN.*

Fig. 12: Retinex effects of some inverse operators on starting image 12a.

discrete Laplacian. Also in this case the contours of the dots are not completely clear.

Finally, we have tested the inverse of the convolutional operator Ψ^0 of an LGN-CNN introduced in Section 4. In Figure 12e we can see that this operator shows Retinex effects where the values of the dots are close to -0.9 and 0.9. In this case the contours of the dots are even clearer than the ones obtained with the LoG and the discrete Laplacian.

To summarize, we have shown that our method reproduces the same results of (Morel et al. 2010) in the case of the Laplacian operator. Furthermore, we have tested it on other operators obtaining remarkable results. It is particularly interested the case of the convolutional operator Ψ^0 since it is able to show Retinex effects even if it is a learned filter with no a-priori structure, enforcing the link between the LGN-CNN architecture and the visual system.

5.2.2 Adelson's checker

We have also tested our algorithm on the Adelson's checker shadow illusion as in (Morel et al. 2010) and (Limare et al. 2011). Since we are more interested in the Retinex effects of LoG and Ψ^0 operators we have analyzed their abilities on the grayscale image (see Figure 13a). It shows a checkerboard with light gray and dark gray square with a cylinder on it that shadows a part of the squares. In particular, the square labeled 'A' and the square labeled 'B' have the same grayscale value (in our case, since -1 is black and 1 is white, they have -0,4953 value). The illusion is built in such a way that, even if they have the same value, they are perceived in a completely different way. Indeed square 'A', which is outside the shadow and surrounded by light gray squares, is perceived as a dark gray square. On the other hand, square 'B', which is inside the shadow and is surrounded by dark gray squares, is perceived as lighter.

We expect that the two operators should reproduce the same behavior of our perception, in particular square 'A' should have a smaller value whereas square 'B' should have a bigger value. Figure 13c shows the Retinex effects obtained with the LoG operator. In particular, the value of square 'A' changes to -0,6553 whereas the value of square 'B' changes to 0,02351. Thus, we have studied the behavior of Ψ^0 operator whose results are shown in Figure 13e. Even in this case Retinex effects occur where the value of square 'A' changes to -0,6035 and the value of 'B' changes to 0,2348.

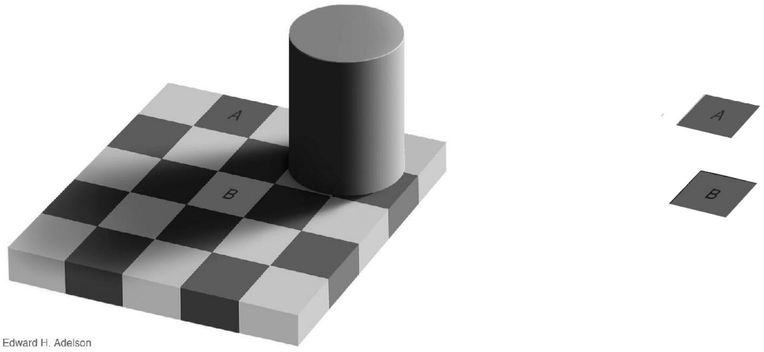
Figures 13d, 13d and 13f highlight the two squares 'A' and 'B' in the starting image and in the recovered images using the LoG and Ψ^0. In this way it is clearer that the Retinex effects occur in both cases.

To summarize, we have shown that the filter Ψ^0 considered as a convolution operator shows Retinex effects really closed to Retinex effects of LoG. This enforces again the link between the structure of our architecture and the structure of LGN.

5.3 Information transmission efficiency

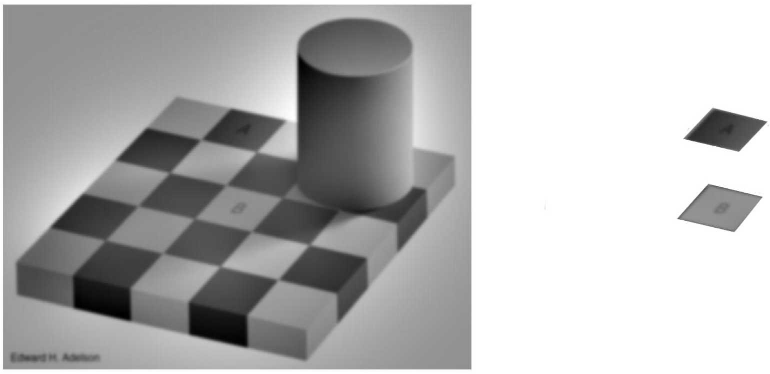
We are now interested to see if the layer ℓ^0 have similar properties as regards the information transmission efficiency of the LGN. Indeed, it is well established (see e.g. (Reinagel and Reid 2000), (Zaghloul et al. 2003), (Uglesich et al. 2009), (Im and Fried 2015), (Pregowska et al. 2019)) that the average firing rate of the retinal neurons that drive information to the LGN is much bigger than that of the LGN. In particular, LGN is able to delete spikes preserving the more informative ones leading to a loss of information.

Thus, we have studied the information loss on the 8000 images of STL10 test set by convolving each image with Ψ^0 and computing the entropy from the histogram of gray scale values via the built-in MatLab function *entropy*. It turns out that on average the entropy decreases from 7.04 to 5.97 with a loss of 15.27 % of the information. Thus, we have reconstructed the images using the Retinex algorithm described in Section 5. The average entropy increases to 6.92 leading to a loss of 1.83% of the information w.r.t. the original dataset. This suggests that almost the entire information contained in the visual stimulus can be reconstructed via some feedback or horizontal connections, where the reconstructed stimulus becomes invariant w.r.t. lightness constancy. Figure 14 shows an example of the convolution and the reconstruction performed on an image of the dataset with the corresponding gray values histogram with respect to the entropy is calculated.



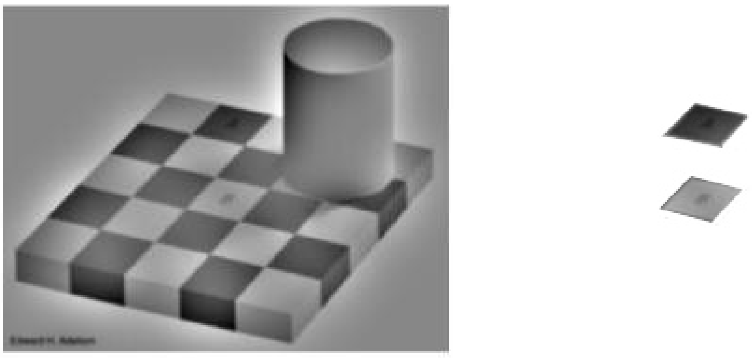
(a) *Grayscale Adelson's checker shadow illusion.*

(b) *Grayscale Adelson's checker shadow illusion: the two squares.*



(c) *Retinex effects of LoG.*

(d) *Retinex effects of LoG: the two squares.*



(e) *Retinex effects of Ψ^0 of LGN-CNN.*

(f) *Retinex effects of Ψ^0 of LGN-CNN: the two squares.*

Fig. 13: Comparison between the Retinex effects of some operators on grayscale Adelson's checker shadow illusion.

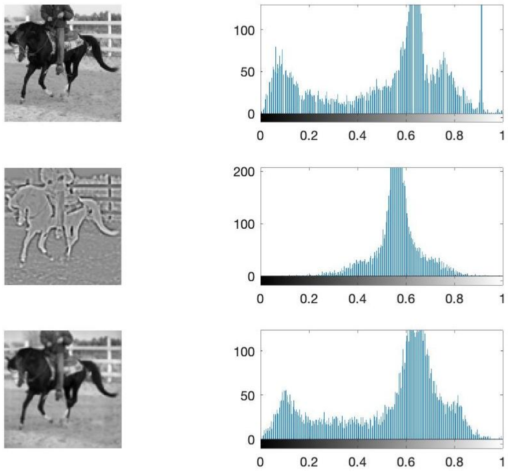


Fig. 14: In order on the left: a grayscale image I, the convolved image $\widetilde{I}$ and the reconstructed image $\widetilde{I}$ via eq. (8). On the right: the corresponding histograms of the grayscale values.

6 Conclusions

The study of the role of the LGN in the visual system and the rotation invariance properties of the RFPs of its cells has leaded our research to the introduction of a CNN architecture that mimics this structure. In particular, we have added to a CNN a first convolutional layer composed by a single filter which attains a rotational symmetric pattern. The filter Ψ^0 has inherited this property from the modified architecture of the neural network.

We have also shown that it is not only rotational symmetric but it also obtains a LoG shape that approximates the RFPs of the LGN cells. In order to study these similarities, we have shown that the Retinex effects of the LoG and Ψ^0 are really closed to each other. These behaviors enforce the link between the visual system structure and the architecture of CNNs.

Furthermore, we have analyzed the statistical distribution of the filters of the second convolutional layer that attain a Gabor shape even with the introduction of the first layer. We have shown that the statistical distribution becomes closer to the real data of RFPs of simple cells in V1 from (Ringach 2002) enriching the connections with the neural structure.

Then, we have faced the theoretical problem regarding the rotation symmetry of the first convolutional layer. We have studied the solution of a convex functional composed by a convolution and a ReLU. Thanks to uniqueness of such functional, we have shown that the solution Ψ^0 has to be rotational invariant. Then, we have built an architecture composed by a single filter Ψ^0 and a ReLU in which Ψ^0 has attained a rotational invariant pattern close to a Gaussian.

In the future we will face the theoretical problem regarding the rotation symmetry of the first convolutional layer for a general LGN-CNN. Furthermore, we will analyze the modifications that in an LGN-CNN occur to the bank of filters of other convolutional layers of deeper architecture, comparing them with neural data. We will also introduce an autoencoder associated to this architecture, which can reconstruct perceived images with the Retinex effect.

A Appendix

A.1 Rotation symmetry of Ψ^0

In this section we define the setting in which we study the rotation symmetry of Ψ^0.

Let us consider the architecture of an LGN-CNN in which we can split the first convolutional layer composed by only one filter from the rest of the neural network which will be fixed. Thus, this first layer can be approximated by a function $\Psi^0 : \mathbb{R}^2 \to \mathbb{R}$, assuming $\Psi^0 \in L^1_{loc}(\mathbb{R}^2)$. A general image can be defined as a function $I : \mathbb{R}^2 \to \mathbb{R}$ where we assume $I \in L^1_{loc}(\mathbb{R}^2)$. We can consider a subset $\Gamma \subset L^1_{loc}(\mathbb{R}^2)$ of all the images where for each image I is defined a labelling $y : \Gamma \to \mathbb{R}$, where $y(I)$ is the corresponding label to image I.

We require some rotational invariant properties on this set Γ. In particular, if we consider a rotation R_θ of an angle θ on R^2 plane around its center then the composition $I_\theta = R_\theta(I) = I(R_{-\theta}(x))$ is still an image and we can also assume that $I_\theta \in \Gamma$ (i.e., that the subset Γ is close under rotation). Furthermore, the rotated image should maintain the same label, i.e., $y(I_\theta) = y(I)$. Since the images we will consider in this problem are grayscale ones (thus with values between 0 and 1) we can also assume that the images are normalized with $||I||_{L^1_{loc}} \leq 1$. Summarizing all these properties we can assume that Γ is a compact set on L^1_{loc}, $\Gamma = \{I \in L^1_{loc}; ||I||_{L^1_{loc}} \leq 1\}$

Thus, the rest of the neural network will be defined by a nonlinear functional

$$\begin{aligned} &C : L^1_{loc}(\mathbb{R}^2) \to L^1_{loc}(\mathbb{R}^2) \\ &C(f(x)) = \max(0, f(x)) \end{aligned} \tag{12}$$

which is one of the more frequently nonlinear function used in CNN, called ReLU. And then we can define

$$\begin{aligned} &F : L^1_{loc}(\mathbb{R}^2) \times L^1_{loc}(\mathbb{R}^2) \to \mathbb{R} \\ &F(I, \Psi^0) := \int_{\mathbb{R}^2} C\left((I * \Psi^0)(z)\right) dz. \end{aligned} \tag{13}$$

Then $F(I, \Psi^0)$ is the label that our architecture associates to the image I and should eventually approximates $y(I)$.

Thus, our aim is to find a function Ψ^0 in such a way that F approximates well the known functional $y(I)$. In particular we would like to minimize the following functional

$$\min_{\Psi^0 \in L^1_{loc}(\mathbb{R}^2)} \int_\Gamma |F(I, \Psi^0) - y(I)| d\mu(I) \tag{14}$$

where the integral done over the set Γ is a Bochner's integral (see e.g., Section 5 of chapter V of (Yosida 1995) and (Mikusinski 1978)).

Our aim is to find a function Ψ^0 that attains the minimum of (14) where F is defined in (13) and C is defined in (12). We would like to find out if there exist some rotational invariant properties on the function Ψ^0. Let us note that the functional defined in (14) is convex thanks to the convexity of the function C defined in (12) and continuous. Then the existence and uniqueness of a solution is guaranteed (see e.g., Section 1.4 of (Brezis 2011)).

Remark 1 *Let us consider two function $f, g \in L^1(\mathbb{R}^2)$ and a rotation R_θ of an angle θ. Then*

$$f * R_\theta(g)(x) = (R_{-\theta}(f) * g)(R_{-\theta}(x))$$

Proof

$$\begin{aligned} f * R_\theta(g)(x) &= \int_{\mathbb{R}^2} f(x-y) R_\theta(g(y)) dy \\ &= \int_{\mathbb{R}^2} f(x-y) g(R_{-\theta}(y)) dy = \end{aligned}$$

then we substitute $y' = R_{-\theta}(y)$ whose Jacobian has determinant equal to 1

$$\begin{aligned} &= \int_{\mathbb{R}^2} f(x - R_\theta(y')) g(y') dy' \\ &= \int_{\mathbb{R}^2} f(R_\theta(R_{-\theta}(x)) - R_\theta(y')) g(y') dy' \\ &= \int_{\mathbb{R}^2} f(R_\theta(R_{-\theta}(x) - y')) g(y') dy' \\ &= \int_{\mathbb{R}^2} R_{-\theta}(f(R_{-\theta}(x) - y')) g(y') dy' = \end{aligned}$$

then we substitute $y'' = R_{-\theta}(x) - y'$ whose Jacobian has determinant equal to 1

$$\begin{aligned} &= \int_{\mathbb{R}^2} R_{-\theta}(f(y'')) g(R_{-\theta}(x) - y'') dy'' \\ &= (R_{-\theta}(f) * g)(R_{-\theta}(x)) \end{aligned}$$

and this conclude the proof.

Now we can demonstrate the rotational invariance of Ψ^0.

Theorem A1 *Let Ψ^0 be a solution to the problem (14) where F is defined in (13) and C is defined in (12). Then Ψ^0 is rotational invariant.*

Proof Let us consider the rotated solution $R_\theta(\Psi^0)$ of an angle $\theta \in [0, 2\pi]$.

$$\int_\Gamma | \int_{\mathbb{R}^2} C\left((I * R_\theta(\Psi^0))(z)\right) dz - y(I) | d\mu(I) =$$

and because of remark 1

$$= \int_\Gamma | \int_{\mathbb{R}^2} C\left((R_{-\theta}(I) * \Psi^0)(R_{-\theta}(z))\right) dz - y(I) | d\mu(I)$$

Since for a general $f \in L^1(\mathbb{R}^2)$ it holds

$$\int_{\mathbb{R}^2} f(R_\theta(x))dx = \int_{\mathbb{R}^2} f(x)dx,$$

then

$$= \int_\Gamma | \int_{\mathbb{R}^2} C\left((R_{-\theta}(I) * \Psi^0)(z)\right) dz - y(I)| d\mu(I) =$$

Finally, since Γ is close under rotation and $y(I) = y(R_{-\theta}(I))$ by hypothesis

$$= \int_\Gamma | \int_{\mathbb{R}^2} C\left(((I) * \Psi^0)(z)\right) dz - y(I)| d\mu(I)$$

Thus, $R_\theta(\Psi^0)$ attains the same value of Ψ^0 for every choice of θ. But thanks to the uniqueness of the solution of this problem because of the compactness of Γ and convexity of the functional (14), $\Psi^0 = R_\theta(\Psi^0)$ $\forall \theta \in [0, 2\pi]$ and this concludes the proof.

Remark 2 *Let us note that the first part of the proof of theorem A1 is valid for a general LGN-CNN architecture. In particular, if we have a solution* Ψ^0 *to the minimization problem (14), then every rotation* $R_\theta(\Psi^0)$ *of an angle* $\theta \in [0, 2\pi]$ *is still a solution. The uniqueness of solution in theorem A1 guarantees that* Ψ^0 *is rotational invariant whereas for a general LGN-CNN this does not hold.*

A.2 Testing the theorem on the same architecture

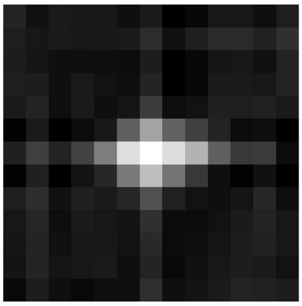
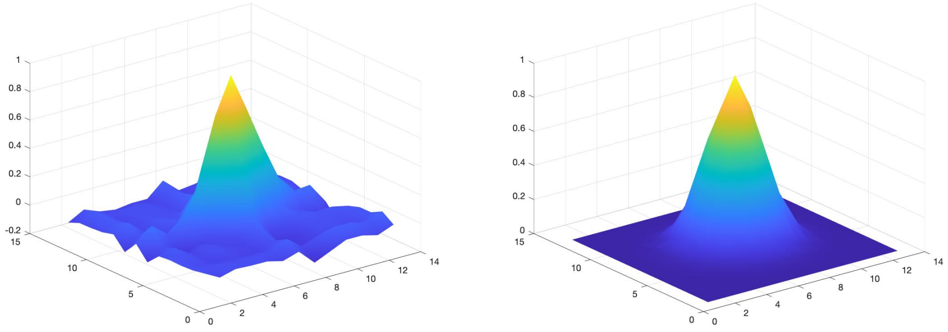


Fig. 15: Filter Ψ^0 of LGN-CNN obtains after training.

In this Section we face the same problem of the proof in Section A to see if ℓ^0 becomes rotational invariant. Since we are going to use a simple architecture composed by a single convolutional layer ℓ^0 with a single filter Ψ^0 and a ReLU we do not expect to obtain a LoG shape filter as in Section 4. Indeed, in the previous case the architecture of LGN-CNN has other convolutional layers whose aim was to further analyze the image and in particular the contours of the objects. For this reason we expected in that case that the LGN-CNN behaved similarly to the LGN and the V1, i.e., Ψ^0 had a LoG shape. On the other hand, in the test we are performing now we can only expect a rotational invariant filter as stated in theorem A1.

To perform this test we have built a new dataset of images starting from the dataset MNIST (a set of digits images, see (LeCun and Cortes 2010)) and the dataset Fashion-MNIST (a set of cloths images, see (Xiao et al. 2017)). They are two similar datasets, composed by grayscale images of size 28×28. The aim of our LGN-CNN is to classify the input as a digit or a cloth, indeed if the image belongs to MNIST or Fashion-MNIST dataset. The new training set has been built by taking the first half of MNIST dataset (30000 images) and the first half of Fashion-MNIST dataset (30000 images), for a total of 60000 images. We have followed the same steps for the test set (for a total of 10000 images) and we have randomly sorted the training and test sets. Then each image has been labeled by 0 if it is a digit and by 1 if it is a cloth.

We have built a really simple LGN-CNN architecture that contains only a first layer with a single filter Ψ^0 of size 13×13 followed by a ReLU and by a fully connected layer. We have trained this neural network for a total of 25 epochs obtaining an accuracy of 98.55 % on the classification task. Figure 15 shows Ψ^0 of this LGN-CNN architecture. We can observe that it has a rotational invariant shape as we expected from theorem A1. Then we have tried to approximate the filter Ψ^0 with a Gaussian with the following formula $G(x, y) = \alpha e^{\frac{-x^2-y^2}{2\sigma^2}}$. Figure 16 shows Ψ^0 and its approximation by a Gaussian. The rotation invariance of Ψ^0 is now enforced thanks to the approximation obtained with a rotational invariant function as the Gaussian.



(a) *Filter Ψ^0 of first layer of LGN-CNN.* (b) *Gaussian function.*

Fig. 16: Comparison between the filter Ψ^0 and a Gaussian. We can see that Ψ^0 is really close to a discrete approximation of a Gaussian fitted to the data.

References

D. Barbieri, G. Citti, A. Sarti, (2014). How uncertainty bounds the shape index of simple cells, *The Journal of Mathematical Neuroscience*, Vol. 4, N.5.

E. Barnard , D. Casasent, (1991). Invariance and neural nets. *IEEE Trans Neural Netw.*;2(5):498-508. doi:10.1109/72.134287

D. H. Brainard and B. A. Wandell, (1986). Analysis of the retinex theory of color vision, *J. Opt. Soc. Amer.* A, vol. 3, no. 10, p. 1651, Oct.

H. Brezis, (2011). Functional Analysis, Sobolev Spaces and Partial Differential Equations. *Springer-Verlag* New York.

M. Carandini, D. Heeger (2012). Normalization as a canonical neural computation. Nat Rev Neurosci 13, 51-62. https://doi.org/10.1038/nrn3136

G. Citti, A. Sarti, (2006). A Cortical Based Model of Perceptual Completion in the Roto-Translation Space, *Journal of Mathematical Imaging and Vision*, vol.24, n.3, pag.307-326.

G. Citti, A. Sarti, (2015). A gauge field model of modal completion. *J. Math. Imaging Vision* 52, no. 2, 267-284.

A. Coates, H. Lee, A. Y. Ng, (2011). An Analysis of Single Layer Networks in Unsupervised Feature Learning. *AISTATS.*

T. Cohen, M. Geiger and M. Weiler, (2018). A General Theory of Equivariant CNNs on Homogeneous Spaces. arXiv 1811.02017.

T. Cohen and M. Welling, (2016). Group Equivariant Convolutional Networks. arXiv 1602.07576.

J. G. Daugman, (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters, *J. Opt. Soc. Am.* A2, 1160-1169.

G. C. DeAngelis, I. Ozhawa, R. D. Freeman, (1995). Receptive-field dynamics in the central visual pathways, *Trends in Neuroscience* 18, 10 451-458.

S. Dieleman, J. De Fauw, K. Kavukcuoglu, (2016). Exploiting cyclic symmetry in convolutional neural networks. arXiv preprint arXiv:1602.02660.

S. Dieleman, K. W. Willett, J. Dambre, (2015). Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, vol. 450, no. 2, pp. 1441- 1459.

C. Enroth-Cugell, J. G. Robson, (1966). The contrast sensitivity of retinal ganglion cells of the cat. *The Journal of physiology*, 187(3), 517-552. https://doi.org/10.1113/jphysiol.1966.sp008107

B. Fasel, D. Gatica-Perez, (2006). Rotation-invariant neoperceptron. In Proc. International Conference on Pattern Recognition (ICPR), vol. 3. IEEE, pp. 336-339.

R. Gens, P. M. Domingos, (2014). Deep symmetry networks. *Advances in Neural Information Processing Systems*, pp. 2537-2545.

B. Gidas, W. N. Ni, L. Nirenberg, (1981). Symmetry of positive solutions of nonlinear elliptic equations in RN. *Math. Anal. Appl.*, Part I (L. Nachbin, Ed.), Academic Press, San Diego.

F. Girosi, M. Jones, T. Poggio, (1995). Regularization Theory and Neural Networks Architectures. *Neural Computation*, vol. 7, pp. 219-269.

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

M. Ferraro and TM. Caelli, (1994). Lie transformation groups, integral transforms, and invariant pattern recognition. *Spatial Vision* 8 , 33-44.

D. H. Hubel, (1988). Eye, brain, and vision. New York : *Scientific American Library* : Distributed by W.H. Freeman.

M. Im, SI. Fried (2015). Indirect activation elicits strong correlations between light and electrical responses in on but not off retinal ganglion cells. J Physiol 593(16):3577-3596.

T. M. Jessell, E. R. Kandel, J. H. Schwartz, (2000). Central visual pathways. *Principles of neural science.* New York: McGraw-Hill. pp. 533-540. ISBN 978-0-8385-7701-1. OCLC 42073108.

J. P. Jones, L. A. Palmer, (1987). An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *J. Neurophysiol.* 58, 1233-1258.

R. Kimmel, M. Elad, D. Shaked, R. Keshet, I. Sobel, (2003). A Variational Framework for Retinex. *International Journal of Computer Vision.* 52. 7-23. 10.1023/A:1022314423998.

E. H. Land, (1964). The Retinex. *American Scientist.* 52(2): 247-64.

E. H. Land and J. McCann, (1971). Lightness and retinex theory, *J. Opt. Soc. Amer.*, vol. 61, no. 1, pp. 1-11, Jan.

D. Laptev, N. Savinov, J. M. Buhmann, M. Pollefeys, (2016). TI-pooling: transformation-invariant pooling for feature learning in convolutional neural networks. arXiv preprint arXiv:1604.06318.

Y. LeCun, C. Cortes, (2010). MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/

T. S. Lee, (1996). Image Representation Using 2D Gabor Wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 18, No. 10.

L. Lei, Y. Zhou, and J. Li, (2007). An investigation of retinex algorithms for image enhancement, *J. Electron.*, vol. 24, no. 5, pp. 696-700, Sep.

M. Liang and X. Hu, (2015). Recurrent convolutional neural network for object recognition, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 3367-3375, doi: 10.1109/CVPR.2015.7298958.

N. Limare, A. B. Petro, C. Sbert and J. M. Morel, (2011). Retinex Poisson equation: a model for color perception. *Image Processing On Line.*

J. Lindsey, S. A. Ocko, S. Ganguli, S. Deny, (2019). A unified theory of early visual representations from retinal to cortex through anatomically constrained deep CNNs. *Proc. of 7th ICLR 19, New Orleans.*

O. Lopes, (1996). Radial symmetry of minimizers for some translation and rotation invariant functionals. *Journal of differential equations* 124, 378388.

D. Marcos, M. Volpi, D. Tuia, (2016). Learning rotation invariant convolutional filters for texture classification. *CoRR*, http://arxiv.org/abs/1604.06720.

J. Mikusinski, (1978). The Bochner Integral. In: The Bochner Integral. *Mathematische Reihe*, vol 55. Birkhauser, Basel.

N. Montobbio, L. Bonnasse-Gahot, G. Citti, A. Sarti, (2019). KerCNNs: biologically inspired lateral connections for classification of corrupted images. Preprint on ArXiv.

J. M. Morel, A. B. Petro and C. Sbert, (2010). A PDE formalization of Retinex theory. *IEEE transactions on image processing*, Vol. 19, NO. 11, November.

J. Petitot, (2009). Neurogéométrie de la vision. *Les édition du École Polytechnique.*

A. Pregowska, A. Casti, E. Kaplan, E. Wajnryb, J. Szczepanski (2019). Information processing in the LGN: a comparison of neural codes and cell types. Biological Cybernetics 113:453.

E. Provenzi, L. D. Carli, A. Rizzi, and D. Marini, (2005). Mathematical definition and analysis of the retinex algorithm, *J. Opt. Soc. Amer.* A, vol. 22, pp. 2613-2621.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, (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.

P. Reinagel , RC. Reid (2000). Temporal coding of visual information in the thalamus. J Neurosci 20(14):5392-5400

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

D.L. Ringach, (2002). Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. *J. Neurophysiol.*; 88(1):455-63.

A. Sarti, G. Citti, (2011). On the origin and nature of neurogeometry. *La Nuova Critica.*

T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, T. Poggio, (2007). Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 411-426.

A. Sherstinsky, (2018). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *CoRR.*

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

S. G. Solomon, B. B. Lee, H. Sun, (2006). Suppressive surrounds and contrast gain in magnocellular-pathway retinal ganglion cells of macaque, *J. Neurosci.* 26(34), 8715-8726.

J. W. Sundsten, J. Nolte, (2001). The human brain: an introduction to its functional anatomy. *St. Louis: Mosby.* pp. 410-447. ISBN 978-0-323-01320-8. OCLC 47892833.

R. Uglesich, A. Casti, F. Hayot, E. Kaplan (2009). Stimulus size dependence of information transfer from retina to thalamus. Front Syst Neurosci 3(10):1-9.

A. Valberg, T. Seim, (2013). Neurophysiological correlates of color vision: a model. *Psychology & Neuroscience*, 6(2), 213-218. https://doi.org/10.3922/j.psns.2013.2.09

F. Wu, P. Hu, D. Kong, (2015). Flip-rotate-pooling convolution and split dropout on convolution neural networks for image classification. arXiv preprint arXiv:1507.08754.

H. Xiao, K. Rasul, R. Vollgraf, (2017). Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. ArXiv.

D. Yamins, J. DiCarlo, (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, Vol 19, 356 EP.

D. Yamins, H. Hong, C. Cadieu, J. Dicarlo, (2013). Hierarchical modular optimization of convolutional networks achieves representations similar to macaque it and human ventral stream. *Adv. Neural Inf. Process. Syst.* 26, 3093-3101.

J. Yeonan-Kim, M. Bertalmío, (2017). Analysis of retinal and cortical components of Retinex algorithms. *Journal of Electronic Imaging*, vol. 26. doi:10.1117/1.JEI.26.3.031208.

K. Yosida, (1995). Functional Analysis. *Springer-Verlag* Berlin Heidelberg.

KA. Zaghloul, K. Boahen, JB. Demb (2003). Different circuits for on and off retinal ganglion cells cause different contrast sensitivities. J Neurosci 23(7):2645-2654

Y. Zhang, TS. Lee, M. Li, F. Liu, S. Tang, (2019). Convolutional neural network models of V1 responses to complex patterns. *J Comput Neurosci.* 46(1):33-54. doi:10.1007/s10827-018-0687-7