# Bridging the Gap between Deep Learning and Neuroscience: An Investigation of the Biological Plausibility of Deep Neural Networks for the Task of Visual Object Recognition

Fiammetta Strazzera Perniciani [1]
Supervisors: Leonardo Franco[2], Paul Tiesinga [1]

*[1]Radbouboud University Nijmegen, Donders Institute for Brain Cognition and Behaviour, The Netherlands*
*[2]Universidad de Málaga, Spain*

**In recent years, Deep Learning has achieved superhuman abilities in many tasks such as visual object recognition. Nevertheless, the brain outperforms Deep Networks in its ability to generalize to distorted images. Yet, the exact mechanisms used to achieve this invariance are still not completely understood. The interplay between neuroscience and Deep Learning could both advance the knowledge on the processes that occur in the brain and help the development of more efficient artificial networks. The aim of the present paper is to study the link between the brain and artificial neural models by comparing the behavior of a Convolutional Neural Network to our knowledge of the processing of visual information in the human cortex. The network's recognition ability under invariance conditions was tested when presenting input images that were different from the images employed for the training of the network. The test images were modified either with geometric deformations, by varying the rotation, position and size of the objects within the image, or by compromising the extent of visual information transmitted from the input when changing the quality, contrast and amount of noise. The results are compared to neural data obtained from behavioral and neuroimaging studies in which the subject's response time, accuracy and neural activations were recorded following the presentation of images with the various types of deformations. Furthermore, the fundamental characteristics of the architecture of the network and the backpropagation algorithm used for the training process are discussed in comparison to the structure of the visual stream and to the synaptic update processes that are thought to be employed by the brain for learning. Our investigation highlights that a great issue with current Deep Neural Networks is the limited performance under image distortions as compared to humans' invariant recognition ability. Furthermore, the present study underlines the differences in the implementation of the learning algorithm in computational models and in the brain as a starting point to improve Deep Learning towards more efficient and more biologically plausible networks.**

*Keywords: Convolutional Neural Network, Deep Learning, neuroscience, visual object recognition*

Corresponding author: Fiammetta Strazzera Perniciani; E-mail: fiammetta.strazzera@gmail.com

Building a representation of visual information is one of the most crucial functions of the visual system. Recognizing or classifying objects is a particularly complex task, since an object can appear in the visual field over various viewing conditions: this is referred to as the invariance problem. Transformations that preserve the identity of an object include changes in position, size, illumination and rotation of the object along with its background. By comparing the behaviour of object recognition algorithms under these invariance conditions to our knowledge of the processing of visual information in the human cortex, we aim at studying the link between the brain and artificial neural networks.

Given its great success over the last years, in the present study, the task of object recognition will be tackled using Deep Learning. Deep Learning, a research area in the field of Machine Learning, is the latest development of artificial neural network models that comprise several hidden layers. It has become the new gold standard among different applications in artificial intelligence. This is supported by its superhuman abilities in several tasks such as pattern recognition, game playing, medical diagnosis and social network filtering. This new technology is inspired by several features of the mammalian brain, without being constrained by any biological limitations.

Initially influenced by neuroscience, Deep Learning algorithms have strongly developed over the past years, making it possible to train artificial neural networks with several layers to complete various tasks efficiently. Nevertheless, those algorithms have now little explicit resemblance to the processes occurring in the mammalian brain. Yet, we claim that the interplay between neuroscience and Deep Learning can advance the study of learning processes in the brain. Neuroscience can help Machine Learning to develop the best strategies, optimizing functions and architectures. Moreover, it can formulate constraints on the implementation of learning algorithms so to properly match the real neural processes. Likewise, Deep Learning provides a tool with which hypotheses from neuroscience can be tested empirically.

The pioneer studies on visual processing in the brain were carried out by Hubel and Wiesel and represent the starting point for the development of Deep Learning algorithms (Hubel &Wiesel, 1959; Hubel & Wiesel, 1962). By performing various experiments investigating the visual system in the cat's brain, they showed that some regions in the visual cortex are sensitive to specific areas of the visual field, called receptive fields, or specific orientations or shapes. Specifically, the authors identified two types of cells in the brain: simple cells and complex cells. The former are neurons in the cortex that respond exclusively to one position or orientation while being silent to stimuli outside their focal area. The latter are units which fire in the presence of specific movements of the object in the visual field. Object recognition might be performed in the brain by integrating information of both types of cells (Schiller, Finlay & Volman,1976). In line with this work, several neural network architectures were proposed.

For instance, Convolutional Neural Networks, that perform nearly as robustly as our brains under several transformations of the objects in the visual field. In these networks each region has its visual receptive field and responds to specific features (Lecun, Bottou, Bengio & Haffner, 1998; Lecun, Haffner, Bottou & Bengio, 1999; Krizhevs, Suskever & Hinton, 2012; Szegedy et al., 2015). Additionally, other networks were designed in which layers that resemble the functioning of simple and complex cells are alternated (Serr, Oliva & Poggio, 2007; Riesenhuber & Poggio, 1999). Amongst the most popular Deep Neural Networks are LeNet-5 (Lecun et al., 1998), a five layers neural network usually applied to the task of recognizing handwritten numbers, HMAX (Serre et al., 2007), a biologically inspired hierarchical neural network, AlexNet (Krizhevsky et al., 2012), an extension of LeNet, GoogLeNet (Szegedy et al., 2015), a 22 layers deep network, and the VGG-16 and Very Deep networks comprising 16 and 19 layers respectively (Simonyan and Zisserman, 2014). For a complete overview of the fundamental innovations and techniques that led to the great performance of neural networks please refer to Nielsen (2018), and Yamins and DiCarlo (2016).

Image classification occurs instantaneously in the brain, whereas it is a challenging task for an artificial neural network. Building an artificial neural network that performs object recognition as accurately and efficiently as our own visual system might be achieved by mapping the spatial organization of the brain areas and portions of the cortex involved in this process. The ability to recognize objects relies on largely feedforward computations that flow throughout the visual ventral stream of the mammalian brain. The transmission of visual information starts in the retina, continues in the lateral geniculate nucleus of the thalamus (LGN) and then through the primary visual cortex V1, secondary visual cortex V2, visual cortex V4 to the inferior temporal cortex (IT) (Trappenberg, 2002). Each cortical area responds to

specific features of an image and unravels different types of information (Blumberg & Kreiman, 2010). It is likely that the IT is the portion of the visual stream that is mainly responsible for object recognition (DiCarlo, Zoccolan & Rust, 2012).

The algorithm used in the brain to solve object recognition is still not completely understood. Empirical findings in neuroscience, concerning the organization and structure of the visual ventral stream, can help to define the hypothesis space and orient the implementation of a possible algorithm. For instance, clues can be taken by studying the activity of neurons in the ventral visual stream, their firing rate, their sparseness and their tolerance, that is the ability to preserve preference for a limited range of object variables. Given its success in the last decades, several attempts have been made to integrate Deep Learning results and neuroscience data (Kheradpishe, Ghodrati, Ganjtabesh & Masquelier, 2016b; Baldi & Sadowski, 2014; Baldi & Sadowski, 2016; Dodge & Karam, 2016; Dodge & Karam, 2017; Geirhos et al., 2017). Several studies compared neural data obtained using functional magnetic resonance imaging (fMRI), electroencephalography (EEG) or magnetoencephalography (MEG) to activation of units in artificial neural networks (Kheradpisheh, Ghodrati, Ganjtabesh & Masquelier, 2016a; Güçlü & van Gerven, 2015).

It was shown that, as a neural network is trained to recognize objects, a hierarchical structure, in which increasingly complex features are processed, naturally emerges along its layers (Cichy, Khosla, Pantazis, Torralba & Oliva, 2016; Güçlü & van Gerven, 2015). This increasing complexity is comparable to the processing of visual information in the brain. Specifically, the last layer of a neural network is particularly predictive of IT neurons' responses and the previous layer is predictive of the responses of neurons in the V4 cortex (Yamins et al., 2014; Cadieu et al., 2014). In contrast, the biological plausibility of the training procedures applied in Deep Learning is still questioned. As a matter of fact, it is unlikely for neurons to perform backpropagation, the most common algorithm used to train neural networks (Rumelhart, Hinton & Williams, 1988). Nevertheless, it could be possible for the brain to approximate this training algorithm. Additionally, the implementation of its optimization and activation functions is largely consistent with the observations and hypotheses regarding the functioning of our brain (Marblestone, Wayne & Kording, 2016).

The aim of the present project is to carry out a detailed analysis of the aspects involved in the functioning of Deep Learning algorithms for object recognition. Specifically, we aim at analyzing whether these aspects have a neural correlate in the mammalian brain and can represent effective simplifications of the processes occurring in biological systems or whether they are completely artificial tools. Firstly, the paper will analyze the behavior of an artificial neural network when modifying the characteristics of the representation in the input images and compare it to neural data. This consists in studying the accuracy in recognizing objects and the activation of neural units when varying the rotation, position or size of the objects as well as changing the quality, the contrast, or adding noise to the input images. Secondly, the characteristics of artificial networks will be discussed in terms of their biological plausibility based on neuroscientific data. These characteristics include the architecture and connectivity, the neural activation functions, the training process, the use of the backpropagation algorithm and the dropout scheme to prevent overfitting.

## Methods

The task of image classification consists in taking an input image and giving the class that it belongs to among a fixed set of categories representing different objects or scenes. In order for the network to learn the correct classification, error signals are used to update the parameters of the network proportionally to the derivative of the classification error. In the present paper, image classification will be investigated using Deep Neural Networks (DNNs), that are characterized by several hidden layers (Goodfellow, Bengio & Courville, 2016). Our DNN was implemented in Keras (Chollet et al., 2015), an efficient and flexible application program interface (API). As a Deep Learning framework, we used Google's Tensorflow (Abadi et al., 2015), an open source library written in Python and used frequently in Machine Learning (Rampasek & Goldenberg, 2016).

Three fundamental factors shape DNNs and determine the correlation between representations in DNNs and cortical visual representations: the architecture, the task and the training procedure (Cichy et al., 2016).

### Deep Neural Network Architecture

It was proven that a network with a single hidden layer, given it has enough units, can approximate any function and operation of a Deep Network (Cybenko, 1989). It is true, however, that the

number of units needed in order to learn decreases exponentially with the depth of the network (Cohen, Sharir & Shashua, 2016; Liang & Srikant, 2016). Moreover, DNNs can represent a large number of possible configurations in the input space with very rich descriptions and are crucial in order to solve the complex problems required for artificial intelligence (Hastad, 1986; Bengio & Delalleau, 2011). With distributed representations, Deep Networks have the advantage of learning the input with a number of parameters that scales linearly and not exponentially with the dimensionality of the feature space, as opposed to non-parametric approaches (Hinton, 2014). Nevertheless, training neural networks with many layers is computationally expensive and frequently has the disadvantage of overfitting the data (Hastad & Goldmann, 1991; Bengio & Lecun, 2007). Overfitting occurs as the network has more parameters than training data and overlearns the input images, losing the ability to generalize. Therefore, in order to avoid these problems, we used a deep architecture with only five layers, resembling the networks proposed by Yamins et al. (2014) and Serre et al. (2007).

The type of neural network we chose for image classification is Convolutional. Convolutional Neural Networks (CNNs) are a type of feed-forward neural network, that is, a network in which the information flows in forward direction from one input layer to one output layer and which have no cycles. CNNs consist of a series of convolutional layers, followed by fully connected (dense) layers, in which the units are connected to all units in the previous layer with a linear operation and by an output layer in which each unit represents a different target class.

## Convolutional layers

Convolutional layers take the input and convolve it with a weight matrix, called filter or kernel. Convolution consists in sliding the filter through the input units and, for each slide, multiplying it element-wise to the corresponding portion of the input, adding up the result to form one unit of the output. The size of the kernel is smaller than the size of the input, therefore, each unit has a local receptive field (Lawrence, Giles & Tsoi, 1997). Each convolutional layer can have more than one filter, leading each to a different output, called a feature map. With the use of convolution, for each feature map the same weight matrix is shared throughout all the input units. Therefore, the weights of a convolutional layer are denominated shared weights. Thus, the same feature of the input object, such as orientation or shape, is detected in a feature map regardless of its position in the visual field: in this sense each filter learns to recognize a specific characteristic of the input.

## Max-pooling layers

Each convolutional layer is typically followed by a max-pooling or downsampling, operation, which reduces the size of each feature map by extracting subregions of the input layer with the maximum value (Zhou & Chellappa, 1988). Specifically, the max-pooling operation divides its input into disjoint regions of a given size and takes the maximum over all the values in each region. Therefore, only the locations that show the maximum correlation with each feature are kept, creating a new, smaller layer, whereas the other values in the region are discarded. Max-pooling preserves features specificity and helps increase robustness to clutter by discarding objects that cause low responses. This reduces the number of parameters of the network and thus the computational cost and processing time (Boureau, Ponce & Lecun, 2010). By reducing the number of parameters of the network, max-pooling additionally helps to prevent overfitting.

## Regularization techniques

In order to reduce overfitting, regularization techniques were applied to the architecture of network, such as dropout and L2-regularization. Dropout is a technique that allows the network to avoid learning the training data too specifically and being unable to classify new images (Hahnloser, Bengio, Frasconi & Schmidhuber, 2000). With dropout, noise is injected into the network in order to increase robustness over variations of the input images (Baldi & Sadowski, 2014). Dropout consists in randomly dropping units during the training of artificial neural networks, preventing each unit to rely excessively on the output of a specific input neuron (Hinton, Krizhevsky, Suskever & Salakhutdinov, 2012). When applying a dropout of probability (or level) p to a layer in the network, in each training iteration every unit in the network layer is deleted with probability p. The remaining weights are then trained according to the chosen training algorithm (Srivastava, Hinton, Krizhevsky, Sutskever & Salakhutdinov, 2014). In order to choose the level of dropout in each layer, a grid search was done by varying the dropout level in the convolutional layers and in the fully connected layer in the set of ten uniformly distributed values between 0 and 0.9. Dropout indeed increased the performance

of the network, confirming the findings of Paine, Khorrami, Han and Huang (2014). The best performing model which was selected for the experimental manipulations had a dropout of 0.1 after the convolutional layers and of 0.5 following the dense layer and reached 88.38% validation accuracy.

In addition to dropout, L2-regularization was included in the learning algorithm, as illustrated in the training procedure section below. Contrary to the findings by Loshchilov and Hutter (2017), this technique was found to be more effective than weight decay as a method to penalize excessively high connections between neurons and therefore reduce overfitting (M Zur, Jiang, Pesce & Drukker, 2009).

## Activation functions

Activation functions are applied to the output of each layer, adding a non-linearity that is necessary in order for the network to perform complex tasks (Jarrett, Kavukcuoglu, Ranzato & LeCun, 2009). The sigmoidal activation function is commonly used in Deep Learning, since it introduces non-linearities in the model. However, a known issue with this function is the vanishing gradient problem (Hochreiter, 1991; Hochreiter et al., 2001). The sigmoid approaches a constant value when moving away from the y-axis and consequently its derivative assumes infinitely small values. Therefore, the error signals needed for learning tend to vanish. In order to avoid this problem, we utilized the rectifier function following all the layers, with the exception of the softmax function that was applied to the output layer, in accordance with the work of Güçlü & van Gerven (2015).

The rectifier applies the function R(zi) = max(0,zi) to the output zi of neuron i in the layer, eliminating the neurons with negative outputs, thus giving rise to a sparse representation. Biologically inspired (Hahnloser, Seung & Slotine, 2003), it is the most frequently employed function in Deep Neural Networks (Ramachandran, Zoph and Le, 2017) because of its efficiency (Glorot, Bordes & Bengio, 2011; Nair & Hinton, 2010). A unit to which this activation function is applied is called Rectified Linear Unit (ReLU) (Nair & Hinton, 2010). The softmax activation function is defined as $S(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$, where $z_i$ represents the output of the neuron $i$ of a given layer. This function is usually used in the final layer of a network used for classification (Bishop, 2006) due to its normalizing effect on its output, preventing it from becoming too large.

## Batch normalization

The values of the input pixels as well as the activations of the units in each layer can have very distinct values throughout the layer, differing by several orders of magnitude. Those values can be adjusted by normalizing the training data and the activations of the layers, a technique called batch normalization. Constraining the units to have the same mean and variance reduces the covariance shift, that measures the amount of variation between activations in one layer (Ioffe & Szegedy, 2015). Batch normalization limits the amount to which updating the parameters in the earlier layers can affect the distribution of values of the following layers. This stabilizes the network, that becomes robust to changes in the input distribution. Therefore, each layer learns more independently, and this speeds up learning and gives the network the ability to generalize (Ma & Klabjan, 2017). Additionally, since the mean and variance for the normalization

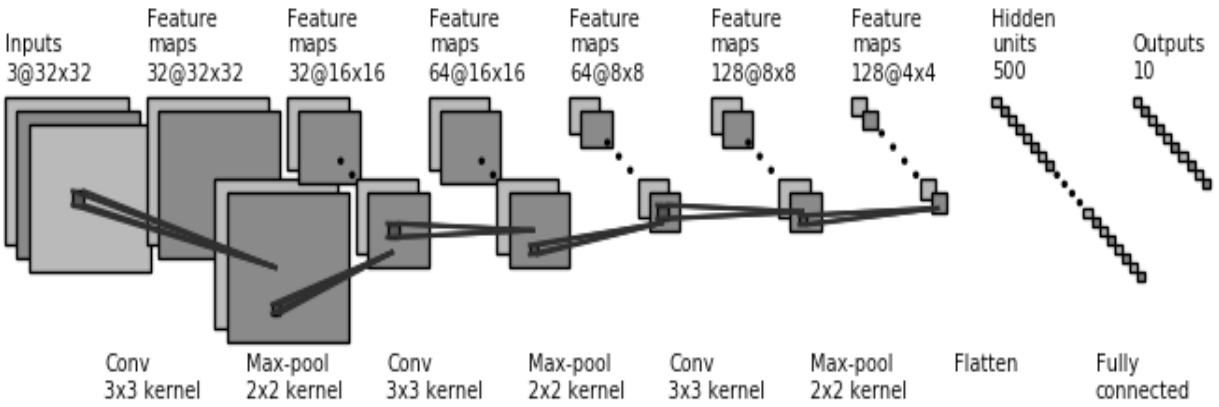


**Figure 1.** Neural network architecture. The first three layers consist of a convolution, with 3x3 kernels and 32, 64 and 128 filters respectively, and a max-pooling operation of size 2x2. Each plane is a feature map. The last convolutional module is followed by a fully connected layer and the output layer. Source: own elaboration.

are computed on batches of data rather than on the whole dataset, this adds noise to learning and therefore has a slight regularization effect that helps preventing overfitting. Batch normalization was added after each activation function, as it was shown to perform better if added after rather than before the layer of non-linearity (Mishkin & Matas, 2015).

As illustrated in Figure 1, the first 3 layers of the network are convolutional, followed by one fully connected layer and the output. Each convolutional layer consists of the convolution, a ReLu activation function, batch normalization, a max-pooling operation of size 2x2 and a dropout layer of probability 0.1. The weight matrices used for convolution have a 3x3 kernel and the filters are 32, 64 and 128 respectively. The fully connected layer is followed by a ReLu activation function and a dropout of probability 0.5. The softmax activation function is applied to the output layer.

## Task

Training on real world objects is critical for the correspondence between layers of the CNNs and cortical visual pathways, as shown by Cichy et al. (2016). The neural network described in the present paper was trained to recognize images in the CIFAR-10 dataset (Krizhevsky, 2009), which consists of 60000 RBG images of size 32x32 representing items from 10 categories. As shown in Figure 2, the categories are: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck.

## Training procedure


**Figure 2.** Example of images from the CIFAR-10 dataset. For each of the ten categories, 10 random images are shown. Source: https://www.cs.toronto.edu/~kriz/cifar.html

The images in the dataset are divided into training set, validation set and test set, in a 10:1:1 ratio. The first set is used to train the network and the remaining two for testing. The training set contains examples of inputs with their associated outputs. Learning is supervised in the sense that the network learns by comparing its prediction to a given target output (supervision). Neural network learning aims at reducing the prediction error, that is, the difference of activation between the actual and the desired output. This is achieved by propagating information from the output layer back to the input layer and updating the layer weights accordingly.

Training the neural network consists in finding the parameters θ of a neural network that significantly reduce a cost function that measures to which degree the predicted output differs from the target output. This cost function includes the loss of the network, that is a measure of the classification error over all the training set, as well as additional regularization terms. The backpropagation algorithm (backwards propagation of error) (Werbos & J. Paul John, 1974; Rumelhart et al., 1988) finds a local optimum of the function that the network is trying to learn by updating its parameters and going towards the direction of lower error.

The per-example loss function is given by:

$$L(x, \hat{y}, \theta) = -\hat{y} log\big(y(x, \theta)\big) \quad (1)$$

where $y(x,\theta)$ is the predicted output vector when the input is $x$, representing the probabilities of $x$ being in each of the classes, and $\hat{y}$ is the target output vector. In order to penalize network weights with high magnitudes, the regularization term

$$\frac{\lambda}{2}\|\theta\|^2 \quad (2)$$

is added to the loss function, where λ is a given penalization factor. Thus, when a batch B of N example images $x_1, \ldots, x_N$ with target outputs $\hat{y}_1, \ldots, \hat{y}_N$ is presented to the network, the total cost is:

$$J(B, \theta) = \frac{-1}{N}\sum_{n=1}^{N} \hat{y}_n \, log\big(y(x_n, \theta)\big) + \frac{\lambda}{2}\|\theta\|^2 \quad (3)$$

In our network, we choose N = 64 and λ = 10−6. In order to minimize the cost function J, an adaptive learning rate optimization algorithm, Adam, whose name derives from adaptive moment estimation, was used (Kingma & Ba, 2015). Adam chooses a separate learning rate for each parameter of the objective function, speeding learning when a different learning rate is needed for each parameter, and uses momentum: for each timestep, a fraction of

the previous update is added to the current update, moving faster to the direction of the minimum and decreasing the oscillations around it. The use of adaptive learning rates combined with momentum makes the algorithm efficient and fairly robust to the choice of hyperparameters (Reddi, Kale & Kumar, 2018).

## Data augmentation

When trained with "small" datasets such as CIFAR-10, which have less images than the total number of parameters of the network, often the models tend to overfit the data (Perez & Wang, 2017). In addition to adding dropout to the architecture of the network and weight regularization to the learning algorithm, another technique used to prevent overfitting is data augmentation (Simard et al., 2003; C. Wong et al., 2016; Cagli et al., 2017). Data augmentation has been proven particularly effective for image classification (Perez & Wang, 2017). This strategy consists of increasing the amount of training samples by applying a transformation, such as reflection, rotation, shear and shift, to the training images. For every epoch a new transformation is applied to every input image. Thus, distinct images are presented to the network each time. An example of such a transformation is illustrated in Fig.3, where an original image from the CIFAR-10 dataset is rotated by 15◦.

In this study, the training dataset was augmented by rotating the initial samples of a random angle between −15◦ and 15◦, translating them horizontally and vertically by 10% of their total width and height, and reflecting them across the vertical axis. Data augmentation significantly improved the performance of the neural network, confirming the results presented by Paine et al. (2014).

Indeed, the final accuracy of the network increased by around 7.3% when applying data augmentation compared to training with the original dataset, as illustrated in Figure 4.

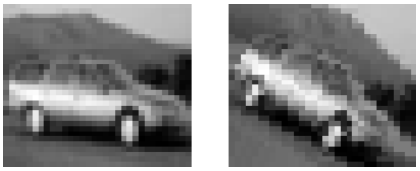

**Figure 3.** Example of the data augmentation process applied to one original image from the CIFAR-10 dataset. Left: original sample image. Right: new image created from the original through a 15◦ rotation.

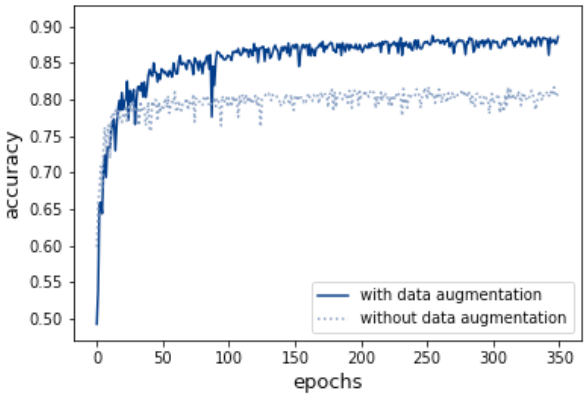


**Figure 4.** Performance of the Convolutional Neural Network model when training for 350 epochs. The black and gray lines show the validation accuracy over epochs when training with and without data augmentation respectively.

## Experiments

Experiments regarding the performance of humans in tasks in which images are modified with various transformations trace back to Koffka (1935) and Walsh and Kulikowski (1988). The visual system is particularly robust to deformations of the objects in the visual field (Rolls, 1992; Rolls & Deco, 2002) and recent computational models have shown similar behaviors (Huiping, Bingfang & Jinlong, 2003; Dodge & Karam, 2016; Kheradpisheh et al., 2016b), although in limited extent (Ghodrati, Farzmadhi, Rajaei, Ebrahimpour & Khaligh-Razavi, 2014; Pinto, Barhomi, Cox & DiCarlo 2011; Pinto, Cox & DiCarlo, 2008). The results section discusses whether the same holds in the chosen network, illustrating the results of a series of experiments that test the behavior of the model in order to compare it to behavioral and neurological data of the human visual stream.

The first set of experiments tested the view invariance of the network to geometric transformations of the input images. Motivated by the considerable translation invariance found in the inferior temporal visual cortex (Rolls & Deco, 2002), in Experiment 1, we translated the input images vertically and horizontally. The use of convolution strides in the first layers of the network suggests that it would show robustness to translation of the objects in the input images. In order to compare the experiment to previous studies that use different datasets, we first reduced the size of the images so that the objects would be mostly contained in the DNN's visual field when their position was varied and pasted them on backgrounds created with an inpainting technique (Telea, 2004). Additionally,

we aimed to investigate the rotation invariance of the model. Since the network was trained with data augmentation techniques, we expected it to show view independence when the rotation angle was within the range of ±15◦ used in the training phase. Moreover, according to Roll's hypothesis (Rolls, 1992), invariant representations can be created by associating different learned views and, therefore, training on every rotation is not necessary in order to build view invariant representations (Booth & Rolls, 1998). In Experiment 2, we tested this hypothesis by studying whether this rotation invariance would be present also when the rotation was greater than ±15◦. Finally, in Experiment 3, we studied the performance of the network when varying the scale of the objects contained in each test image. This was achieved by reducing the size of the image within the visual field, from a 32x32 to a 20x20 size and adding a background using an inpainting technique (Telea, 2004). We expected the network to show invariance to object scaling when the quality of the images was not excessively compromised.

In order to be able to compare the results of our study to previous literature, in the first set of experiments, we used an experimental setting similar to Kheradpisheh et al. (2016b). We considered three types of variation (rotation in plane, translation over horizontal and vertical axis, and scaling) of different levels of difficulty, from no-variation to high variation. For each of these conditions (variation type and difficulty level), we created a database by randomly selecting 300 training images and 150 test images for each of the object categories. Then, we applied the corresponding variation to each database and fed our pre-trained model with the varied images. Finally, for each condition the network was evaluated on the corresponding test images.

Experiments 4 to 6 investigated the performance of the network when tested under various deformations of the input images. Following Huiping et al. (2003) and Dodge & Karam (2016), we hypothesized that the network would show robustness to moderate deformations, but that the accuracy would drop to 0 after a certain threshold. In Experiment 4, we decreased the quality, that is, the resolution, of the images by progressively reducing the number of pixels in the input images. In Experiment 5, the contrast of the test images was changed from 0 (grey image) to 1 (original image) in steps of 0.05. Additionally, in Experiment 6, noise was injected to the network by randomly selecting an increasing number of pixels in each image and changing their value with a random value between 0 and 1 taken from a uniform distribution. The number of corrupted pixels varies between 0 and 800, being 32*32*3=3072 the size of the input, in steps of 25. Since there are three color channels, this implies that the maximum percentage of noise injected corrupted at most 75% of the input pixels.

Lastly, Experiment 7 consisted in studying the activations of the neurons to new stimuli when varying the amount of dropout in the network. Before starting the training procedure, the dropout was varied between 0 and 0.9 in steps of 0.1 in the first dense layer. We hypothesized that the sparsity of the neuronal activations to novel images would increase as a function of the level of dropout, as found by Baldi & Sadowski (2014).

# Results

## Geometric invariances

### Biological background for invariant object recognition

The ability to recognize objects under different viewing conditions is characteristic to the brain (DiCarlo et al., 2012). Although IT neurons show some tolerance to object deformation, individual neurons need not be invariant: in the visual stream, there are neurons which are view-independent and neurons whose response depends on the orientation of the object in a given image (Dicarlo & Cox, 2007). It is hypothesized that invariance is obtained by the hierarchical combination of these neurons, in which invariant features are progressively extracted (Rolls & Deco, 2002; Tanaka, 1996). In this framework, cells at higher layers pool input from lower layer cells, becoming more tolerant to changes (Riesenhuber & Poggio, 1999). Selectivity and invariance of object representations indeed increase along the visual stream (Franco, Rolls, Aggelopoulos & Jerez, 2007; Rust & DiCarlo, 2010). In a recent study (Cichy, Khosla, Pantazis, Torralba & Oliva, 2017), a marker of neural processing of spatial information was found in MEG data and compared to the development of spatial layout descriptions in computational models. Analogously to the visual stream, a gradual emergence of invariant representation was found to appear hierarchically in the neural network layers (Cichy et al., 2017).

A possible explanation for human view invariance, proposed by Biederman (1987), is that the brain represents objects by dividing their parts into 3-dimensional view-independent geometric primitives called geons that have clearly distinguishable properties in respect to symmetry,

roundness and size. Recognition of an object would occur by computing the geon descriptions of its parts and comparing them to the stored descriptions. This theory of recognition by shape makes recognition under disrupted viewing conditions easier. In support of this theory are the facts that elements that are essential to the perception of geons, such as borders, were proven to be highly relevant for object recognition in humans and that no visual priming effect was found when using distinct sets of geons between trials (Biederman, 2000). Nevertheless, the set of qualitative shape properties chosen by Biederman is arbitrary and there is no evidence for a structural description of geons in the brain. Furthermore, the theory fails to differentiate between distinct objects of the same type (Dickinson, 1999).

Another account for visual object recognition proposes that the brain stores representations as collections of different views of the object, and that recognition occurs through interpolation between those views and depends on the distance to the closest viewpoints (Spetch & Friedman, 2003). Alternatively, it was proposed that the brain may incorporate both approaches by relying on structural descriptions of the parts of the objects as well as on viewpoint-specific features (Tarr & Bülthoff, 1998).

### Experiment 1: Translation invariance

Following Kheradpisheh et al. (2016b), for the translation experiment, we selected four levels of variation defined by the percentage of translation of the images in the horizontal and vertical axes. The images were translated of a random number of pixels between ±1% of the total image size in the no-variation condition, and between ±20%, ±40%, ±60% in the conditions of variation levels 1, 2 and 3 respectively. We hypothesized that the convolution strides in the first layers of the network would create robustness to translations, and that the performance of the network would decrease when the object would start falling out of the receptive visual field. Since the objects depicted in the CIFAR-10 dataset occupy a large portion of the image, we had to decrease their size, thus reducing their quality, in order to vary their position without excessively losing visual information. This led to a lower general accuracy.

Taken together, as illustrated in Figure 5b, our network shows invariance when the position of the object within the visual field is varied in the first three conditions. Indeed, when the images are translated of up to 40% of their size, the performance of the network is stable and does not decrease significantly, similarly to what has been reported in Kheradpisheh et al. (2016b). Yet, in the highest variation condition the accuracy drops considerably, as opposed to the aforementioned study. This is likely due to the fact that, with high amounts of variation, the objects in our database fall out of the receptive field.

In order to compare our results to neurophysiological data, we computed the total drop in accuracy from the no-variation to the maximum variation condition. The performance drop is of approximately 6.3% in the present experiment, whereas it was reported to be around 3% for humans (Kheradpisheh et al., 2016b). However, it is to keep in mind that only four categories were used in said study.

In general, translation invariance is the most robust type of image variation for DNNs and humans when using uniform or natural backgrounds (Kheradpisheh et al., 2016b). For human subjects, this could follow from the fact that the brain represents objects in a rectangular coordinate system, making translations easy for the brain to overcome (Hinton, 2014b). In opposition, when using natural scenes in which more than one object was present, this translation invariance was shown to decrease in humans (Rolls & Deco, 2002).

### Experiment 2: Rotation transformation

For the rotation experiment, we defined the levels of variation by the range of the angle of rotation of the images. The images were rotated by a random angle between ±1◦ in the no-variation condition, and between ±15◦, ±30◦, ±60◦, ±90◦ in the conditions of variation levels 1, 2, 3 and 4 respectively. We hypothesized that the accuracy of the network would not decrease when the rotation angle was within the range of ±15◦, the angle of rotation for data augmentation used in the training phase.

Figure 5a illustrates the performance under the various rotation conditions. The network shows moderate robustness under the various levels of variation, especially in the first level. As a matter of fact, when rotating the images of an angle within the data augmentation angle of ±15◦, the performance decreases of less than 0.1% with respect to the no-variation condition. The recognition accuracy does not immediately drop when increasing the rotation angle outside the training range and the decrease follows a trend that is comparable to that depicted in the study by Kheradpisheh et al. (2016b). However, the performance decreases significantly in the last two levels of variation and is considerably lower than

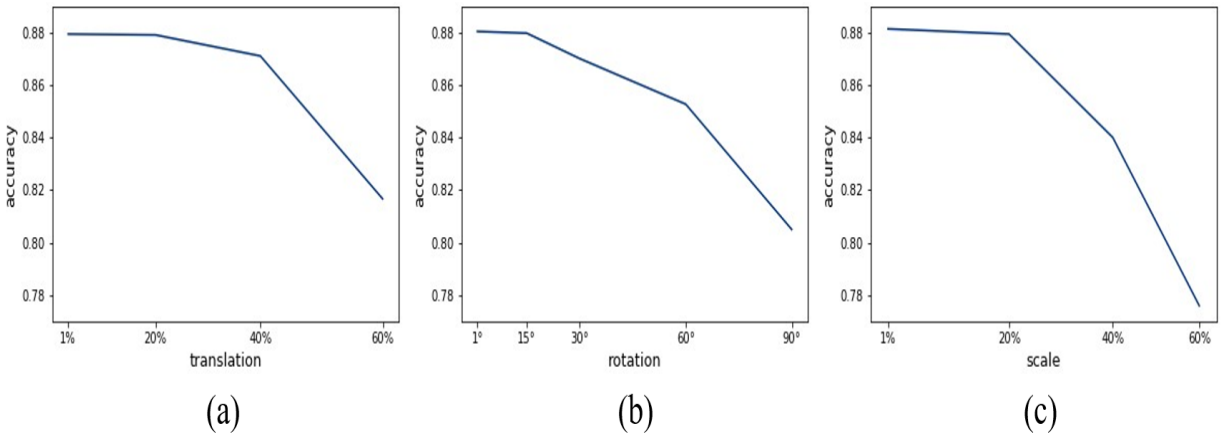


**Figure 5.** Empirical distribution of recognition accuracy over the various types of image deformations. The accuracy is averaged over all the images in the test set. (a) Performance of the network over image translation. The x-axis indicates the percentage of translation in the horizontal and vertical axes. (b) Performance of the network over image rotation. The x-axis indicates the angle of rotation, in degrees. (c) Performance of the network over object scaling. The x-axis indicates the size of the rescaled object with respect to the original.

that reported for the AlexNet (Krizhevsky et al., 2012) and the Very Deep (Simonyan & Zisserman, 2014) networks. The total drop in accuracy from the no-variation to the high variation condition is around 7.5% in our experiment. Nevertheless, it is to consider that in the aforementioned study only four categories were used. Moreover, this gap could be explained by the difference in the datasets used, and in the networks considered that are much deeper than our model.

In contrast with the hypotheses by Rolls (1992), the difficulties in recognizing rotated objects hold true when considering human subjects, either in terms of their response time (Murray, Jolicoeur, McMullen & Ingleton, 1993) or of their performance (Spetch & Friedman, 2003), which decreases of around 5% (Kheradpisheh et al., 2016b), for orientations to which the subjects were not trained on. This could suggest that recognition of a rotated object in the human visual system occurs through linear interpolation of two-dimensional learned views rather than by building a three-dimensional model (Bülthoff & Edelman, 1992). As a matter of fact, the first approach would explain the increase in recognition time and performance error proportionally to the amount of rotation of the object. On the contrary, the response time for recognition of rotated objects was found to diminish with practice. This suggests a shift from a mental rotation approach to a more orientation invariant approach, that could make use of geons, in which the object features are learned independently of their orientation, (Murray et al., 1993), or, more directly, suggests that the increase in the number of views with practice would lead to more uniform responses (Bülthoff & Edelman, 1993). The number of required views depends on the object and could be compared to the number of samples needed by a neural network in order to be able to generalize (Murray et al., 1993).

### Experiment 3: Scaling invariance

The size of the objects contained in each test image was progressively reduced in order to test the scaling invariance of the network. The four levels of variation were defined by the size of the new image with respect to the original image. The size of the images was reduced to a random quantity within 1% of the original image size in the no- and within 20%, 40%, 60% in the conditions of variation levels 1, 2 and 3 respectively. We expected the network to show scale invariance to a certain degree, but to drop significantly in the highest variation level, as reported in Kheradpisheh et al. (2016b). Indeed, as illustrated in Figure 5c, the shape of the curve resembles that of the accuracy of the AlexNet (Krizhevsky et al., 2012) and the Very Deep (Simonyan & Zisserman, 2014) models tested in the paper. As hypothesized, the network shows robustness to image scaling when the quality of the images is not excessively compromised.

Regarding a comparison to biological data, the total drop in accuracy from the first to the last condition is of nearly 10% for human subjects (Kheradpisheh et al., 2016b) and similar (around 10.5%) for our model. It was observed that human performance in size invariant tasks significantly improves with practice, but the improvement is specific to each object and does not transfer to novel objects (Furmanski & Engel, 2000).

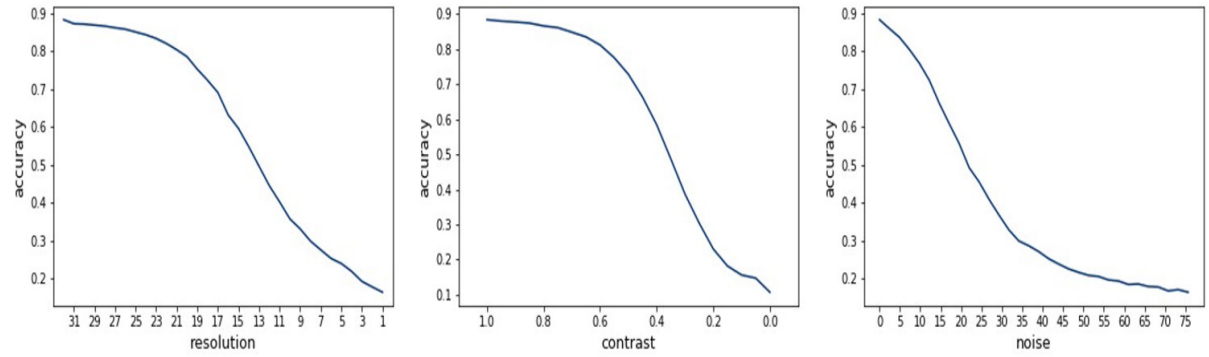


**Figure 6.** Empirical distribution of recognition accuracy over the various types of image distortions. The accuracy is averaged over all the images in the test set. (a) Performance of the network over image resolution. The x-axis indicates the length of the side of the input test images, in pixels. (b) Performance of the network as a function of contrast. The x-axis indicates the contrast of the modified image with respect to the original image. (c) Performance of the network as a function of noise. The x-axis indicates the percentage of noise that is randomly added to the input pixels.

This finding corroborates the hypothesis that recognition occurs in object specific mechanisms in late areas of the visual stream and is consistent both with the geons theory and with image-based models (Furmanski & Engel, 2000; Tarr & Bülthoff, 1998).

In general, when testing our model, the accuracies were higher for the translation and the rotation variations compared to the scaling condition. The same was observed for other DCNNs and for human subjects, both for their accuracy and their recognition time (Kheradpisheh et al., 2016b). This suggests that translations and rotations are easier to tolerate and need less processing time than scaling variations.

## Quality, contrast and noise

### Experiment 4: Decreasing the quality of the input

We decreased the resolution of the images by progressively reducing the number of pixels in the input images used for testing, resulting in blurred images. The performance of the network as a function of image resolution is illustrated in Figure 6a. The pattern is similar to the accuracy of the networks tested in Figure 2 of Dodge and Karam (2016): for the first levels of blurring the accuracy does not diminish significantly, however, the network is sensitive to high blurring levels and the performance gradually decreases to chance level when the resolution is reduced to one pixel. The significant reduction in accuracy could be due to the fact that the reduction of quality also removes textures in the input images, which may be a crucial feature used by neural networks for model recognition (Dodge & Karam, 2016).

### Experiment 5: Modifying the contrast of the images

Similarly, the contrast of the test images was gradually decreased, from a factor of 1 (original image) to a factor of 0 (grey image) in steps of 0.05, obtaining the performance depicted in Figure 6b. The recognition accuracy over contrast shows a greater robustness with respect to the other deformations, confirming the results obtained in Dodge & Karam (2016) and in Geirhos et al. (2017). As a matter of fact, the accuracies in the contrast experiment of the latter study range from approximately 91 − 94% for VGG-16 (Simonyan & Zisserman, 2014), GoogLeNet (Szegedy et al., 2015) and human subjects and 84% for the AlexNet (Krizhevsky et al., 2012) model when the contrast factor is 1 to chance level for the contrast factor of 0.1. Likewise, the performance of our model starts from the original accuracy of approximately 88.4%, it drops when the contrast factor decreases to less than 0.4 until reaching chance level for a contrast factor of 0.1. A similar performance is achieved by human observers in this task (Geirhos et al., 2017).

In humans, the contrast gain control system evolved as a sophisticated contrast normalization technique and is responsible for the robustness to contrast variations (Geisler & Albrecht, 1995). In order to achieve a greater contrast invariance, images could be normalized in the first layers of the

network, the training set could be augmented with low contrast images or a mechanism similar to the contrast gain control present in humans could be included in the architecture of the network (Geirhos et al., 2017).

### Experiment 6: Adding noise to the input images

Noise was added to the test images in various percentages by replacing a random set of the pixels with values drawn from a uniform distribution. The percentage of noise varies between 0% and 75%. When adding noise to the input images, the accuracy rapidly decreases following the trend in Figure 6c, approaching chance level when more than 25% of the input pixels are replaced. This rapid drop in the network performance is in accordance with the studies presented in Dodge and Karam (2016) and Geirhos et al. (2017): after the first 10% of noise is added, the accuracy drops of 12% in our model, whereas it drops of approximately 47% in VGG-16 (Simonyan & Zisserman, 2014) and GoogLeNet (Szegedy et al., 2015) and of 50% in AlexNet (Krizhevsky et al., 2012). In contrast, the drop-in accuracy for human subjects was of only 5%. Handling noise is very challenging for artificial models and drastic differences were found between DNNs and humans in this task, with human subjects outperforming artificial models (Dodge & Karam, 2017). A possible explanation could be that, since the noise was picked from a uniform distribution, it has a high frequency, thus even small changes in the input and in the first layers of the network propagate considerably in higher layers, significantly modifying the output of the model (Dodge & Karam, 2016).

In conclusion, the performance under blur and noise is reduced independently of the artificial model taken into consideration, suggesting that this depends on the architecture and training of the networks (Dodge & Karam, 2016). Therefore, the obvious solution of modifying the model accordingly or training the model on blurred or noisy images arises naturally, although this could consequently compromise the performance of the network with high quality images.

## Dropout and Sparsity

### Experiment 7: Correlating dropout and sparse representations

Evidence of sparse representations in early visual areas (Simoncelli, 2005; Lennie, 2003; Berry, Warland & Meister, 1997; Reinagel, 2001) and the efficiency of sparse networks (Olshausen & Field, 1996; Olshausen & Field, 2004) motivate the study of the emergence of sparsity in trained networks. Therefore, a possible correlation between dropout and sparsity was tested by varying the amount of dropout in the network. We trained the ten networks resulting from increasing the dropout level in the first dense layer from 0 to 0.9 in steps of 0.1 and hence analyzed the empirical distribution of neuronal responses of the resulting networks to the images in the test set.

In accordance with Baldi and Sadowski (2014), we found that high levels of dropout contribute to sparse representations: the activations of neurons in the first dense layer of the network are significantly closer to 0 in Figure 7b compared to Figure 7a. In particular, Figure 5, representing the mean activations of each layer of the network when presenting the 5000 test images, can be compared to Figure 11.1 in Baldi and Sadowski (2014): there is a clear prevalence of neurons with activations that are close to 0. This correlation derives from the tendency of dropout of preventing each neuron to rely excessively on other units (Hinton et al., 2012), which is achieved by minimizing the variance across neuronal activations. Therefore, sparse representations are favored.

## Discussion

In this paper, we compared the processing of information in the visual system to the behavior of a Convolutional Neural Network by analyzing the aspects involved in the implementation of the computational model and in the functioning of the backpropagation algorithm for object recognition. The network's response was tested when presenting input images that were different from the images employed for the training of the network. The test images were modified either with geometric deformations, by varying the rotation, position and size of the objects within the image, or by compromising the extent of visual information transmitted from the input when changing the quality, contrast and amount of noise. The results were compared to neural data obtained from behavioral and neuroimaging studies in which the subjects' response times, accuracies and neural activations were recorded following the presentation of images with the various types of deformations. Furthermore, the fundamental characteristics of the architecture of the network as well as the training process were discussed in comparison to the structure of the visual stream and to the synaptic

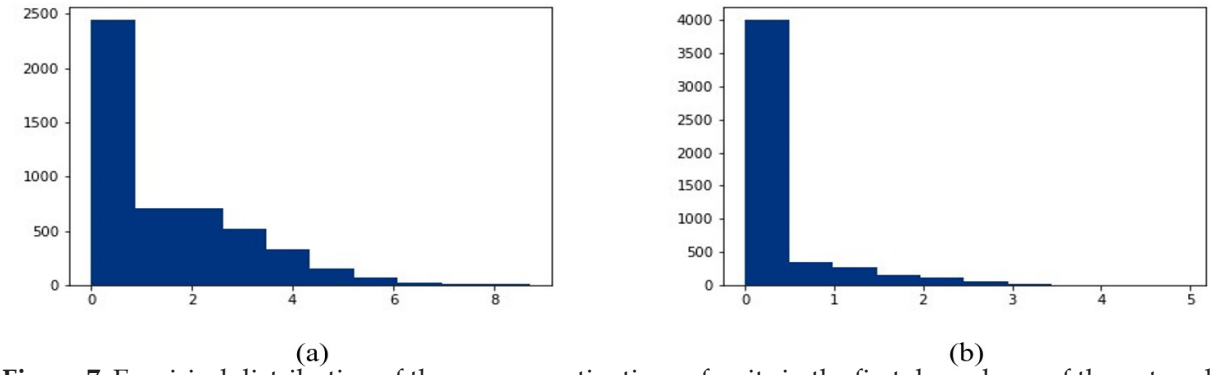


**Figure 7.** Empirical distribution of the average activations of units in the first dense layer of the network when presenting the 5000 images in the test set, either when using no dropout (a) or a 0.7 dropout level (b). The x-axis indicates the amount of activation; the y-axis represents the number of units corresponding to each amount of activation.

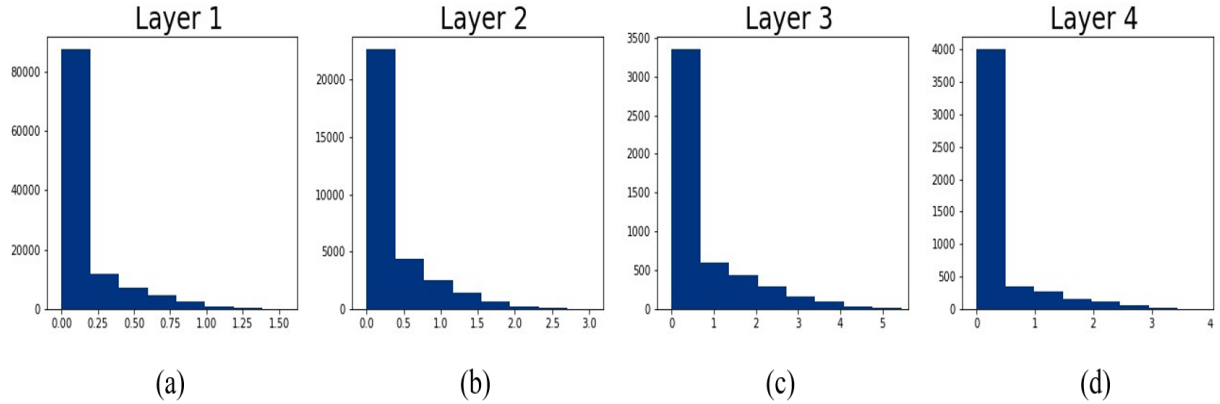


**Figure 8.** Empirical distribution of the average activations of units in each layer of the network when presenting the 5000 images in the test set. The x-axis indicates the amount of activation; the y-axis represents the number of units corresponding to each amount of activation. (a),(b),(c) Average neuronal activations in the three convolutional layers. (b) Average neuronal activation in the first dense layer. The prevalence of small activations is due to the tendency of dropout of favoring sparse representations.

update processes that are thought to be employed by the brain for learning.

In summary, our results indicate that our computational model is invariant to different kinds of deformations in limited amounts. In many of our experiments and in previous research (Dodge & Karam, 2016, Dodge & Karam, 2017; Geirhos et al., 2017; Kheradpisheh et al., 2016a,b), when increasing the level of image deformation, the accuracy of the networks decreases more rapidly than with human subjects. It was demonstrated that the brain responds differently to distinct kinds of image deformations: for instance, size invariance signals appear earlier than position invariance signals (Isik, Meyers, Leibo & Poggio, 2014) and rotation invariance signals (Dill & Edelman, 1997) in the brain, suggesting that some, but not all, mechanisms for invariant object recognition could be built-in (Nishimura, Scherf, Zachariou, Tarr & Behrmann, 2014). Characterizing the amount of image transformation independently of which type of variation is applied and, therefore, correlating each task to the others is not straightforward. Nevertheless, our results are comparable to the conclusions of other studies, that found both DNNs and the human brain to be more robust to rotation and translation than to scaling of the test images, in which the most amount of visual information is lost (Kheradpisheh et al., 2016b). Indeed, a considerable correlation between computational models and the human brains in terms of categorization accuracy was found (Kheradpisheh et al., 2016b). This suggests that the tasks that have the greatest computational complexity likewise represent the most challenging image variations for humans.

On the one hand, the present results illustrate that the performance of the network rapidly decreases when lowering the quality of the input images, adding noise or modifying the contrast, yet many of the images that are misclassified by the CNN are still recognizable by humans (Geirhos et al., 2017). Moreover, it was found that artificial networks can be easily mislead with low noise percentages (Goodfellow, Shlens & Szegedy, 2014) or fooled into

falsely recognizing objects in images of pure noise (Nguyen, Yosinski & Clune, 2014), even though these conditions are carefully chosen and unlikely to occur. On the other hand, there exist examples of the opposite situation, in which images that have very poor resolution, or a significant amount of noise were successfully classified by artificial networks and not by humans (Wright, Yang, Ganesh, Sastry & Ma, 2009). The present study, along with previous results on different DNNs (Ghodrati et al., 2014; Pinto et al., 2011; Dodge & Karam, 2016, Dodge & Karam, 2017; Geirhos et al., 2017), suggest that, even though neural networks have reached human classification abilities on known benchmarks, there is still a gap between Convolutional Neural Networks and the human visual system when the images are distorted (Dodge & Karam, 2017). This gap could in part be explained by the greater exposition of humans to image transformations compared to DNNs through experience and evolution. Nevertheless, it is true that humans overcome DNNs in their ability to generalize to unseen distortions (Geirhos et al., 2017). These dissimilarities give insights into the aspects that need to be improved in order to bridge the gap between neuroscience and Deep Learning and suggest a starting point for future research, in which for instance the training data could be augmented with distorted images.

It could be argued that the chosen model and training dataset are rather simple compared to the complexity of the human visual system architecture and of the real world tasks: deeper networks have proven to reach higher performance accuracies in different recognition tasks (Krizhevsky et al., 2012; Szegedy et al., 2015; Simonyan & Zisserman, 2014; Yamins & DiCarlo, 2016). Nevertheless, this straightforward design allows for tight experimental control without excessively affecting the performance. This permitted to focus on investigating and replicating the architecture and the behavior of the brain's visual system rather than optimizing its performance on a specific task. As a matter of fact, the chosen network is inspired by neural processes and the single components are, where possible, subject to biological constraints as opposed to various more complex artificial models. Several preliminary experiments were performed in order to select the number of layers, the connectivity between neurons, the dropout level and the activation functions, based on the network's performance as well as the artificial units' activations in comparison with biological data.

This procedure would have been excessively expensive in terms of computational cost if considering deeper architectures. Using different combinations of neural network architectures and datasets while considering the tradeoff between experimental control and model complexity is a crucial next step for future research in order to achieve more accurate and biologically plausible results.

Moreover, it would have been interesting to analyze further aspects of the network's response other than the performance accuracy, as, for instance, the neural unit activation in comparison to brain data obtained in fMRI studies (Kheradpisheh et al., 2016a; Güçlü & van Gerven, 2015). Additionally, it is crucial to test the robustness of artificial networks by constructing alternative experimental designs, with different recognition tests or types of images, i.e. synthetic images (Pinto et al., 2008).

In the following, additional similarities between DNNs and the human visual system along with aspects that question the biological plausibility of the current implementation of neural networks, from the learning procedure to the architecture, will be discussed. For instance, neurophysiological aspects such as spike timing dependent plasticity, dendritic computation, local excitatory-inhibitory networks may explain how gradient descent methods could be implemented in the brain (Marblestone et al., 2016).

An important assumption that makes the comparison between artificial and real networks possible is that the brain has developed cost functions, shaped by evolution, and is able to optimize them in order to adjust the connections across neurons and achieve its goals (Marblestone et al., 2016). Similarly, learning in computational models is based on the optimization of cost functions using backpropagation. The backpropagation algorithm is extremely powerful and is therefore commonly used in neural networks, although it has been widely believed to be biologically implausible (Crick, 1989; Stork, 1989), for various reasons.

To begin with, it requires labelled data for learning, even though almost all real data are unlabeled. Human brains have considerably more degrees of freedom, that is, connections, than seconds of life and consequently than the amount of labelled data they could possibly receive (Hinton, 2014). It is therefore impossible to learn weights for all the synapses in the brain, even though it is unlikely that all the connections need to be used (Hinton, 2016). Moreover, DNNs could employ unsupervised learning, in which learning can occur through unlabeled data and combine it with backpropagation only for fine-tuning the weights or for transfer learning. This technique consists in

exploiting previously learned representations to transfer a prior on the distribution of the input in order to learn new data more easily (Hinton, 2016). Transfer learning could simulate human's ability to learn a new task with few examples, as opposed to the thousands of examples required by current state of the art DNNs. However, for the purposes of the present study, unsupervised learning was not used since it is not necessary for the CIFAR-10 dataset (Paine et al., 2014).

Furthermore, while the standard artificial neurons do not encode precise timing, it is thought that synaptic weights in the brain change with Spike-Timing-Dependent-Plasticity (STDP): potentiate when a presynaptic spike is rapidly followed by a postsynaptic spike and depress when the opposite situation occurs (Gerstner, Kempter, van Hemmen & Wagner, 1996; Markram & Sakmann, 1995). The magnitude of the weight change decreases as the presynaptic and postsynaptic spikes separate. STDP can be seen as a spike-based formulation of the Hebbian postulate, stating that synapses are strengthened if a presynaptic neuron fires slightly before the postsynaptic one (Hebb, 1949). This theory is extended with the concept of synaptic weakening, in which synapses are weakened if the presynaptic cell is consistently not co-active with the postsynaptic neuron (Stent, 1973). However, it was proven that in the visual and motor systems information is mostly carried by the average firing rate of neurons, that is more compatible to learning update rules in artificial models, rather than by the spike timing (Baldi & Sadowski, 2014).

Moreover, artificial units need to send two different kinds of signals: the forward signal, representing its activity and used to generate a prediction, and the backward signal, that is, the derivative of the cost function used for the learning update. On the contrary, there is evidence that feedforward and feedback connections in the brain are implemented in distinct paths and real neurons only use one kind of signal, encoded with spikes (Douglas et al., 1989).

Additionally, DNNs are mainly feedforward networks and the feedback mechanisms occur only during learning, whereas there is evidence for continuous feedback processes in the brain (Bullier, McCourt & Henry, 1988; Felleman & Van, 1991; De Pasquale & Sherman, 2011; Mignard & Malpeli, 1991). The feedback and feedforward computations are implemented in two distinct phases and, since the feedback step needs to follow the feedforward step, a synchronization mechanism is needed (Bengio, Lee, Bornschein, Mesnard & Lin, 2015). This synchronization is not necessary in the case of recurrent neural networks, which are more compatible with neurophysiological processes under this point of view (Simard, Ottaway & Ballard, 1988). The need for two distinct phases and a separate network for the feedback of error is eliminated by associating each neuron with a mirror neuron that imitates the feedforward path in order to cancel the top-down component (Guerguiev, Lillicrap & Richards, 2017). This allows for a network that continuously generates predictions and feedback at the same time. Yet, there is no known biochemical mechanism that could duplicate the weight of a synapse between two cells (Baldi & Sadowski, 2016).

Furthermore, backpropagation assigns blame on a neuronal basis, depending on how each neuron contributed to the error, therefore feedback paths need exact knowledge of the downstream synapses (Bengio et al., 2015). Otherwise stated, in order to compute the global cost function, each neuron would need to know the output of every other neuron, whereas evidence of local learning rules has been found in some regions of the brain (Rolls & Deco, 2002). A global cost function requires the unlikely condition of the weights matrix to be symmetric (Grossberg, 1987), although the use of random weights has proven to work well in practice and gives a good approximation of backpropagation (Lillicrap, Cowden, Tweed & Akerman, 2014; Lillicrap, Cowden, Tweed & Akerman, 2016) when the synaptic signs do not change between feedback and feedforward connections (Liao, Leibo & Poggiom, 2015).

These discrepancies are solved if the error derivatives needed for backpropagation are encoded in the temporal change of the neuronal firing rates (Hinton & McClelland, 1988). This allows the output of a neuron to represent an error derivative at the same time, as it is also indicating the presence or absence of a feature (Whittington & Bogacz, 2015). Consequently, in the learning rule the weight update is proportional both to the presynaptic activity and to the rate of change of the postsynaptic activity, analogously to the STDP learning update (Bi & Poo, 1998). In this framework, STDP could be identified as a form of stochastic gradient descent (Hinton, 2016; Bengio et al., 2015). This learning rule can approximate the differential anti-Hebbian plasticity in which synapses updates depend on the product of the presynaptic firing rate and the time derivative of postsynaptic firing rate (Xie & Seung, 2000).

Another issue with backpropagation is that it requires for each connection to communicate with both positive and negative derivatives. In

contrast, according to Dale's Law (Strata & Harvey, 1999), real synapses do not change sign. However, employing neurons that are either entirely inhibitory or excitatory is unlikely to limit the functions that can be learned (Tripp & Eliasmith, 2016; Parisien, Anderson & Eliasmith, 2008).

Moreover, artificial neurons in the backpropagation algorithm can assume values in a continuous range. On the contrary, real neurons transmit information through binary spikes (Bengio et al., 2015). Nonetheless, backpropagation is very robust to noise, thus the network units could be rounded to 0 or 1, a technique similar to dropout, without compromising the model performance (Hinton, 2016). Finally, backpropagation involves purely linear computations, whereas dendrites can alternate linear and non-linear calculations. A learning rule similar to the one proposed by Hinton and McClelland (1988) solves this problem by including a non-linear term derived from the probability of firing in the weight update (Bengio et al., 2015).

Markedly, the architecture of DNNs is particularly effective in object recognition and resembles the architecture of cortical visual pathways (LeCun et al., 1999). When an object appears in the visual field, the information flows from the retina, through the LGN, to the primary and secondary visual cortex, then to V4 and finally to the inferotemporal cortex (Trappenberg, 2002). The information flows from the retina to the IT in 100 ms and then starts to flow backwards in order to update the synaptic connections. However, if the gaze is interrupted, the feedforward activations detected within the first 100 ms after presentation of the input are similar to the activations of units in Convolutional Neural Networks (Goodfellow et al., 2016).

Similar to the processing of visual information in the brain, a hierarchical structure naturally emerges along the layers of a Deep Neural Network, that process increasingly complex features (Cichy et al., 2016; Güçlü & van Gerven, 2015; Ba & Caruana, 2014), as occurs in successive regions of the visual stream. For instance, the primary visual cortex has a two-dimensional structure that reflects the images encoding the visual information that hit the retina. It is formed of simple and complex cells, which respond to specific shapes or movements respectively (Goodfellow et al., 2016). From these simple aspects, increasingly complex features are represented in successive brain areas, until the encoding of high-level characteristics in the inferotemporal cortex. Analogously, networks with multiple layers will automatically learn to recognize simple features, as edges and color, in the first layers and increasingly complex features, from shapes to higher-order characteristics like faces, in successive layers (Cichy et al., 2016; Güçlü & van Gerven, 2015; Ba & Caruana, 2014). Therefore, all intermediate features with different levels of complexity between the raw data and the final representation of an object can be represented in distinct layers, supporting the network's ability to generalize.

Specifically, the first layers of a DNN exhibit properties similar to early visual areas (Cichy et al., 2016). As a matter of fact, the receptive fields in the visual cortex can be accurately modelled by Gabor filters, in which the weights follow a Gabor function (Marĉelja, 1980; Jones & Palmer, 1987). A new architecture that incorporates Gabor filters into convolutional DNNs has recently been proposed, performing similarly to many known CNNs on the popular benchmarks such as MNIST, CIFAR-10, CIFAR-100 and ImageNet (Luan et al., 2017). By any means, the first layers of a neural network were proven to naturally converge to Gabor filters even when not explicitly programmed to do so (Bengio et al., 2015). Additionally, recent studies found that the penultimate and ultimate layers of a neural network are particularly predictive of V4 and IT neurons' responses respectively (Yamins et al., 2014; Cadieu et al., 2014), specifically when the network is trained with supervised methods (Khaligh-Razavi & Kriegeskorte, 2014). Accordingly, based on similar computational models (Cichy et al., 2016; Güçlü & van Gerven, 2015; Yamins & DiCarlo, 2016), we identify the three convolutional layers in our network with the primary visual cortex (V1), secondary visual cortex (V2), and the visual area V4, respectively, and the first dense layer with the inferior temporal cortex (IT).

In addition, the convolution operation in CNNs is inspired by biological processes in the visual cortex. Some regions in the visual cortex are sensitive to particular areas of the visual field and to specific features of the object such as orientation, shape, and movement in space. (Hubel & Wiesel, 1959; Hibel & Wiesel 1962). Similarly, in convolutional layers each neuronal unit is connected only to a subset of units in the previous layer and each filter is sensitive to a specific shape or feature. Moreover, biological circuits were proven to be able to perform the convolution operation (Cichy et al., 2016), that is thought to occur in simple cells (Serre et al., 2007).

In contrast, the biological plausibility of shared weights, that is, using the same weight matrix for all the input layer, has been questioned, since the brain uses local fields. However, a similar technique to

weight sharing can be approximated (Hinton, 2016): if two regions encoding low-level features in an early layer are close enough, then they share some high-level features in a higher layer, that gives top down supervision for both lower layer features. Thus, learning features in a low-level region helps creating higher level representation from which other low-level regions can extract information. By knowing the region's input as well as the desired high-level features, learning can be considerably accelerated. Therefore, it is possible to transfer information across regions in a layer without transporting weights.

Likewise, the use of the max-pooling operation is open to criticism: Hinton (2014) observed that, although pooling gives a small amount of translation invariance at each layer, it ignores the relations between the parts of each image and loses information that is currently not relevant but could be useful for future tasks. This suggests that more levels of structure are needed in order to properly disentangle the data. Nevertheless, it was proven that biological circuits are able to perform the max-pooling operation (Cichy et al., 2016). This idea has been supported by studies with both intracellular (Lampl, Ferster, Poggio & Riesenhuber, 2004) and extracellular (Gawne & Martin, 2002) recordings. In the simple and complex cells paradigm, the latter are thought to be responsible for the pooling operation (Serre et al., 2007): the size of the receptive fields indeed decreases from the simple to the complex stage.

Regarding the use of dropout in neural networks, its biological counterpart may be the neuronal refractory period, occurring after an action potential, in which the neuron is incapable of firing. Furthermore, by linking a dropout of probability p to a neuron that spikes with probability $1 - p$, Hinton (2016) demonstrated that randomly dropping units in neural networks is similar to the random noise inherent to the spiking rate of biological neurons that follows a Poisson process.

Moreover, the data augmentation technique used to increase the number of training samples in DNNs is biologically plausible. As a matter of fact, it might mimic the learning of invariant object representations in the brain, that occurs through a varied dataset consisting of distinct views of the objects under different viewing conditions. Furthermore, the variability in the input data could also derive from eye movements, such as drifts or saccades. It was shown that in models trained with high levels of data augmentation, the last layers exhibit greater similarities to the responses of IT in humans compared to networks trained without this technique (Hernández-García, Mehrer, Kriegeskorte, König & Kietzmann, 2018).

An additional technique used to train CNNs that biological circuits are able to approximate is batch normalization (Cichy et al., 2016). Indeed, homeostatic plasticity mechanisms in the brain operate a sort of synaptic scaling that minimizes the current into a neuron and is comparable to the application of batch normalization (Turrigiano & Nelson, 2004; Stellwagen & Malenka, 2006; Turrigiano, 2008). However, the normalization statistics change for every timestep and are computed having complete knowledge of the output of all neurons in each layer, which would be impossible for real neurons. Nevertheless, a more biologically plausible technique for normalization was proposed by Liao et al. (2016), that learns running estimates of the mean and variance only in local regions and is computationally efficient.

A last feature regarding the implementation of CNNs that can be compared to mechanisms in the human brain concerns activation functions. The biological counterpart of activation functions is the action potential firing, that determines the firing of a neuron as a function of its input (L. Hodgkin & F. Huxley, 1990). The rectifier function that was used in our network is biologically inspired and compatible with our current knowledge of the functioning of real neurons (Hahnloser et al., 2003).

## Conclusion

In conclusion, we aimed to achieve a more complete overview of the differences and similarities between artificial networks and the brain. The question was tackled by comparing the behavior of Deep Neural Networks, inspired by neuroscience, and the human visual system. Deep architectures have the ability of representing properties of increasing complexity and abstraction in distinct layers and are therefore very expressive. As a matter of fact, DNNs have achieved superhuman abilities in many object classification tasks. Yet, there is still a significant gap between Deep Networks and the brain in terms of invariant recognition ability, that may be due in part to some limitations of current computational models. In artificial networks, the lack of the extensive feedback information that is provided to the visual system and used to continuously update and refine visual representations could explain their lower recognition accuracies under image variations. Moreover, CNNs are purely visual, whereas, in the brain, visual information is integrated with input

from many other senses, which is likely to improve its internal object representations, giving it an advantage over artificial networks. Looking at the insights from neuroscience and focusing on the issues discussed in the present paper, Deep Learning can be improved even further towards more efficient and more biologically plausible networks.

## References


A. Olshausen, B., & Field, D. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature. 381, 607–609.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., S. Corrado, G., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Ba, J. L. & Caruana, R. (2014). Do deep nets really need to be deep? Advances in Neural Information Processing Systems, 27, 2654–2662.

Baldi, P. & Sadowski, P. (2014). The dropout learning algorithm. Artificial Intelligence, 210, 78–122.

Baldi, P. & Sadowski, P. (2016). A theory of local learning, the learning channel, and the optimality of backpropagation. Neural Networks, 83, 51–74.

Bengio, Y. & Delalleau, O. (2011). On the expressive power of deep architectures. Springer-Verlag: Berlin, Heidelberg.

Bengio, Y. & Lecun, Y. (2007). Scaling learning algorithms towards AI. MIT Press.

Bengio, Y., Lee, D.-H., Bornschein, J., Mesnard, T., & Lin, Z. (2015). Towards biologically plausible deep learning.

Berry, M. J., Warland, D. K., & Meister, M. (1997). The structure and precision of retinal spike trains. Proceedings of the National Academy of Sciences, 94(10), 5411–5416.

Bi, G.-q. and Poo, M.-m. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. Journal of neuroscience, 18(24), 10464–10472.

Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. Psychological review, 94(2), 115–147.

Biederman, I. (2000). Recognizing depth-rotated objects: A review of recent research and theory. Spatial vision, 13(2), 241–253.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning (Information Science and Statistics). Berlin, Heidelberg: Springer-Verlag.

Blumberg, J. & Kreiman, G. (2010). How cortical neurons help us see: visual recognition in the human brain. The Journal of Clinical Investigation, 120(9), 3054–3063.

Booth, M. C. & Rolls, E. T. (1998). View-invariant representations of familiar objects by neurons in the inferior temporal visual cortex. Cerebral cortex, 8(6), 510–523.

Boureau, Y.-L., Ponce, J., & Lecun, Y. (2010). A theoretical analysis of feature pooling in visual recognition. 27th International Conference on Machine Leaning, Haifa, Israel.

Bullier, J., McCourt, M., & Henry, G. (1988). Physiological studies on the feedback connection to the striate cortex from cortical areas 18 and 19 of the cat. Experimental Brain Research, 70(1), 90–98.

Bülthoff, H. H. and Edelman, S. (1992). Psychophysical support for a two-dimensional view interpolation theory of object recognition. Proceedings of the National Academy of Sciences of the United States of America, 89(1):60–64.

Bülthoff, H. H. & Edelman, S. (1993). Evaluating object recognition theories by computer graphics psychophysics. In, 139–164.

C. Wong, S., Gatt, A., Stamatescu, V., & D. McDonnell, M. (2016). Understanding data augmentation for classification: when to warp?

Cadieu, C. F., Hong, H., Yamins, D. L. K., Pinto, N., Ardila, D., Solomon, E. A., Majaj, N. J., & DiCarlo, J. J. (2014). Deep neural networks rival the representation of primate IT cortex for core visual object recognition. PLOS Computational Biology, 10(12), 1–18.

Cagli, E., Dumas, C., and Prouff, E. (2017). Convolutional neural networks with data augmentation against jitter-based countermeasures – profiling attacks without pre-processing.

Chollet, F. et al. (2015). Keras. https://keras.io.

Cichy, R. M., Khosla, A., Pantazis, D., & Oliva, A. (2017). Dynamics of scene representations in the human brain revealed by magnetoencephalography and deep neural networks. NeuroImage, 153, 346–358.

Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A., & Oliva, A. (2016). Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. Scientific Reports, 6.

Cohen, N., Sharir, O., & Shashua, A. (2016). On the expressive power of deep learning: A tensor analysis. JMLR: Workshop and Conference Proceedings. 49, 1-31

Crick, F. (1989). The recent excitement about neural networks. Nature. 337(6203), 129–132.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems. 2, 303-314.

De Pasquale, R. & Sherman, S. M. (2011). Synaptic properties of corticocortical connections between the primary and secondary visual cortical areas in the mouse. Journal of Neuroscience, 31(46), 16494–16506.

Dicarlo, J. & Cox, D. (2007). Untangling invariant object recognition. Trends in Cognitive Neuroscience, 11(8), 333–341.

DiCarlo, J. J., Zoccolan, D., & Rust, N. C. (2012). How does the brain solve visual object recognition? Neuron, 73(3), 415–434.

Dickinson, S. J. (1999). Object representation and recognition. Cognitive science.

Dill, M. & Edelman, S. (1997). Translation invariance in object recognition, and its relation to other visual transformations.

Dodge, S. F. & Karam, L. J. (2016). Understanding how image quality affects deep neural networks. 2016 Eight International Conference on Quality of Multimedia Experience (QoMEX). Portugal: Lisbon.

Dodge, S. F. and Karam, L. J. (2017). A study and comparison of human and deep learning recognition performance under visual distortions. 26th International Conference on Computer Communication and Networks (ICCCN). Canada: Vancouver.

Douglas, R. J., Martin, K. A., & Whitteridge, D. (1989). A canonical microcircuit for neocortex. Neural Computation. 1(4), 480–488.

Felleman, D. J., & Van, D. E. (1991). Distributed hierarchical processing in the primate cerebral cortex. Cerebral cortex. 1(1), 1–47.

Franco, L., Rolls, E. T., Aggelopoulos, N. C., & Jerez, J. M. (2007). Neuronal selectivity, population sparseness, and ergodicity in the inferior temporal visual cortex. Biological Cybernetics. 96, 547–560.

Furmanski, C. S., & Engel, S. A. (2000). Perceptual learning in object recognition: Object specificity and size invariance. Vision research, 40(5), 473–484.

Gawne, T. J. & Martin, J. M. (2002). Responses of primate visual cortical v4 neurons to simultaneously presented stimuli. Journal of Neurophysiology, 88(3),1128–1135.

Geirhos, R., Janssen, D. H. J., Schütt, H. H., Rauber, J., Bethge, M., & Wichmann, F. A. (2017). Comparing deep neural networks against humans: object recognition when the signal gets weaker. Computer Vision and Pattern Recognition.

Geisler, W., & Albrecht, D. (1995). Bayesian analysis of identification performance in monkey visual cortex: nonlinear mechanisms and stimulus certainty. Vision research, 35(19), 2723–2730.

Gerstner, W., Kempter, R., van Hemmen, J. L., & Wagner, H. (1996). A neuronal learning rule for sub-millisecond temporal coding. Nature, 383(6595), 76–78.

Ghodrati, M., Farzmahdi, A., Rajaei, K., Ebrahimpour, R., and Khaligh-Razavi, S.-M. (2014). Feedforward object-vision models only tolerate image variations compared to human. Frontiers in computational neuroscience, 8:74.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, USA: Fort Lauderdale.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. USA: MIT Press.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. CoRR.

Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. Cognitive Science, 11(1), 23–63.

Güçlü, U. & van Gerven, M. A. J. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. Journal of Neuroscience, 35, 10005–10014.

Guerguiev, J., Lillicrap, T. P., & Richards, B. A. (2017). Towards deep learning with segregated dendrites. ELife, 6.

Hahnloser, R., Sarpeshkar, R., A. Mahowald, M., Douglas, R., & Sebastian Seung, H. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. ICLR, 405, 947–951.

Hahnloser, R. H. R., Seung, H. S., & Slotine, J.-J. (2003). Permitted and forbidden sets in symmetric threshold-linear networks. Neural Computation, 15(3), 621–638.

Hastad, J. (1986). Almost optimal lower bounds for small depth circuits. Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing, 86, 6–20, USA: New York.

Håstad, J. & Goldmann, M. (1991). On the power of small-depth threshold circuits. Computational Complexity, 1(2), 113–129.

Hebb, D. O. (1949). The organization of behavior: A neuropsychological theory. New York: Wiley.

Hernández-García, A., Mehrer, J., Kriegeskorte, N., König, P., & Kietzmann, T. C. (2018). Deep neural networks trained with heavier data augmentation learn features closer to representations in hit. 2018 Conference on Cognitive Computational Neuroscience.

Hinton, G. (2014) Can the brain do back-propagation?

Hinton, G. (2016) What is wrong with convolutional neural nets?

Hinton, G. E. & McClelland, J. L. (1988). Learning representations by recirculation. Neural Information Processing Systems, 358–366.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. Computer Vision and Neural Computing.

Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen.

Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Hubel, D. H. & Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. Journal of Physiology, 148, 574–591.

Hubel, D. H. & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. Journal of Physiology, 160, 106–154.

Huiping, H., Bingfang, W., & Jinlong, F. (2003). Analysis

to the relationship of classification accuracy segmentation scale image resolution.

Ioffe, S. & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. Computer Vision and Neural Computing.

Isik, L., Meyers, E. M., Leibo, J. Z., & Poggio, T. (2014). The dynamics of invariant object recognition in the human visual system. Journal of neurophysiology, 111(1), 91–102.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., & LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? 2009 IEEE 12th International Conference on Computer Vision, 2146– 2153.

Jones, J. P. & Palmer, L. A. (1987). An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. Journal of Neurophysiology, 58(6), 1233–1258.

Khaligh-Razavi, S.-M. & Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain it cortical representation. PLoS Computational Biology, 10(11).

Kheradpisheh, S. R., Ghodrati, M., Ganjtabesh, M., & Masquelier, T. (2016a). Deep networks can resemble human feed-forward vision in invariant object recognition. Scientific Reports, 6(32672).

Kheradpisheh, S. R., Ghodrati, M., Ganjtabesh, M., & Masquelier, T. (2016b). Humans and deep networks largely agree on which kinds of variation make object recognition harder. Frontiers in Computational Neuroscience. 92(10).

Kingma, D. P. & Ba, J. L. (2015). Adam: A method for stochastic optimization. ICLR 2015, 2654–2662.

Koffka, K. (1935). Principles of Gestalt Psychology. New York: Harcourt, Brace.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097–1105.

L. Hodgkin, A. & F. Huxley, A. (1990). A quantitative description of membrane current and its application to conduction and excitation in nerve. Bulletin of Mathematical Biology, 52, 25–71.

Lampl, I., Ferster, D., Poggio, T., & Riesenhuber, M. (2004). Intracellular measurements of spatial integration and the max operation in complex cells of the cat primary visual cortex. Journal of Neurophysiology, 92(5), 2704–2713.

Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face recognition: a convolutional neural network approach. IEEE Transactions on Neural Networks, 8(1), 98–113.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999). Object recognition with gradient-based learning. Shape, Contour and Grouping in Computer Vision.

Lennie, P. (2003). The cost of cortical computation. Current Biology, 13, 493–497.

Liang, S. & Srikant, R. (2016). Why deep neural networks for function approximation?

Liao, Q., Kawaguchi, K., & Poggio, T. A. (2016). Streaming normalization: Towards simpler and more biologically-plausible normalizations for online and recurrent learning. Computer Vision and Neural Computing.

Liao, Q., Leibo, J. Z., & Poggio, T. A. (2015). How important is weight symmetry in backpropagation? Computer Vision and Neural Computing..

Lillicrap, T., Cownden, D., Tweed, D., & J. Akerman, C. (2014). Random feedback weights support learning in deep neural networks.

Lillicrap, T. P., Cownden, D., Tweed, D. B., & Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. Nature communications. 7(13276).

Loshchilov, I. & Hutter, F. (2017). Fixing weight decay regularization in adam. Computer Vision and Neural Computing.

Luan, S., Zhang, B., Chen, C., Cao, X., Han, J., & Liu, J. (2017). Gabor convolutional networks. Computer Vision and Neural Computing.

M Zur, R., Jiang, Y., Pesce, L., & Drukker, K. (2009). Noise injection for training artificial neural networks: A comparison with weight decay and early stopping. Medical Physics. 36(10), 4810–4818.

Ma, Y. & Klabjan, D. (2017). Convergence analysis of batch normalization for deep neural nets.

Marblestone, A. H., Wayne, G., & Kording, K. P. (2016). Toward an integration of deep learning and neuroscience. Frontiers in Computational Neuroscience, 10.

Marčelja, S. (1980). Mathematical description of the responses of simple cortical cells. Journal of Optical Society of America, 70(11), 1297–1300.

Markram, H.& Sakmann, B. (1995). Action potentials propagating back into dendrites triggers changes in efficacy. Society for Neuroscience.

Mignard, M. & Malpeli, J. G. (1991). Paths of information flow through visual cortex. Science, 251(4998), 1249–1251.

Mishkin, D. & Matas, J. (2015). All you need is a good init. Computer Vision and Neural Computing.

Murray, J. E., Jolicoeur, P., McMullen, P. A., & Ingleton, M. (1993). Orientation-invariant transfer of training in the identification of rotated natural objects. Memory & Cognition, 21(5), 604–610.

Nair, V. & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. ICML'10, 807–814, USA.

Nguyen, A. M., Yosinski, J., & Clune, J. (2014). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. Computer Vision and Neural Computing.

Nielsen, M. A. (2018). Neural networks and deep learning.
Nishimura, M., Scherf, K., Zachariou, V., Tarr, M., & Behrmann, M. (2014). Size precedes view: Developmental emergence of invariant object representations in lateral occipital complex. Journal of Cognitive Neuroscience. 27, 1–18.
Olshausen, B. A. & Field, D. J. (2004). Sparse coding of sensory inputs. Current Opinion in Neurobiology, 14(4), 481–487.
Paine, T. L., Khorrami, P., Han, W., & Huang, T. S. (2014). An analysis of unsupervised pre-training in light of recent advances. Computer Vision and Neural Computing.
Parisien, C., H Anderson, C., & Eliasmith, C. (2008). Solving the problem of negative synaptic weights in cortical models. Neural Computation. 20(6), 1473–1494.
Perez, L. & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning.
Pinto, N., Barhomi, Y., Cox, D. D., & DiCarlo, J. J. (2011). Comparing state-of-the-art visual features on invariant object recognition tasks. Applications of computer vision (WACV), 2011 IEEE workshop on, 463–470.
Pinto, N., Cox, D. D., & DiCarlo, J. J. (2008). Why is real-world visual object recognition hard? PLoS Computational Biology, 4(1), 27.
Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. Computer Vision and Neural Computing.
Rampasek, L. & Goldenberg, A. (2016). Tensorflow: Biology's gateway to deep learning? Cell Systems, 2(1), 12–14.
Reddi, S. J., Kale, S., & Kumar, S. (2018). On the convergence of adam and beyond. International Conference on Learning Representations.
Reinagel, P. (2001). How do visual neurons respond in the real world? Current Opinion in Neurobiology, 11(4), 437–442.
Riesenhuber, M. & Poggio, T. (1999). Hierarchical models of object recognition in cortex. Nature Neuroscience, 2, 1019–1025.
Rolls, E. (1992). Neurophysiological mechanisms underlying face processing within and beyond the temporal cortical visual areas. Philosophical Transactions of the Royal Society of London Series B-Biological Sciences. 335(1273), 11–21.
Rolls, E. T. & Deco, G. (2002). Computational Neuroscience of Vision. Oxford University Press.
Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Neurocomputing: foundations of research. chapter learning representations by back-propagating errors. 696–699.
Rust, N. C. & DiCarlo, J. J. (2010). Selectivity and tolerance ("invariance") both increase as visual information propagates from cortical area v4 to it. The Journal of neuroscience, 30(39), 12978–12995.
S Stent, G. (1973). Stent gs. a physiological mechanism for hebb's postulate of learning. Proceedings of the National Academy of Science USA, 70, 997-1001.
Schiller, P. H., Finlay, B. L., & Volman, S. F. (1976). Quantitative studies of single-cell properties in monkey striate cortex. i. spatiotemporal organization of receptive fields. Journal of neurophysiology, 39(6), 1288–1319.
Serre, T., Oliva, A., & Poggio, T. (2007). A feedforward architecture accounts for rapid categorization. Proceedings of the National Academy of Science. U.S.A., 104(15), 6424–6429.
Simard, P., Ottaway, M., & Ballard, D. (1988). Analysis of recurrent backpropagation.
Simard, P. Y., Steinkraus, D., & Platt, J. (2003). Best practices for convolutional neural networks applied to visual document analysis. Institute of Electrical and Electronics Engineers, Inc.
Simoncelli, E. (2005). Statistical modeling of photographic images.
Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.
Spetch, M. L. & Friedman, A. (2003). Recognizing rotated views of objects: Interpolation versus generalization by humans and pigeons. Psychonomic Bulletin & Review, 10(1), 135–140.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929–1958.
Stellwagen, D. & Malenka, R. C. (2006). Synaptic scaling mediated by glial tnf-α. Nature. 440(7087), 1054.
Stork (1989). Is backpropagation biologically plausible? International 1989 Joint Conference on Neural Networks, 241–246.
Strata, P. & Harvey, R. (1999). Dale's principle. 50, 349–350.
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. Computer Vision and Pattern Recognition (CVPR).
Tanaka, K. (1996). Inferotemporal cortex and object vision. 19, 109–139.
Tarr, M. J. & Bülthoff, H. H. (1998). Image-based object recognition in man, monkey and machine. Cognition, 67(1), 1–20.
Telea, A. (2004). An image inpainting technique based on the fast marching method. Journal of Graphics Tools, 9(1), 23–34.
Trappenberg, T. (2002). Fundamentals of Computational Neuroscience. Oxford University Press.
Tripp, B. & Eliasmith, C. (2016). Function approximation in inhibitory networks. Neural Networks, 77, 95– 106.
Turrigiano, G. G. (2008). The self-tuning neuron: synaptic scaling of excitatory synapses. Cell, 135(3), 422– 435.
Turrigiano, G. G. & Nelson, S. B. (2004). Homeostatic plasticity in the developing nervous system. Nature Reviews Neuroscience, 5(2), 97.
Walsh, V. & Kulikowski, J. (1988). Perceptual Constancy: Why Things Look as They Do. Cambridge University

Press.
Werbos, P. & J. Paul John, P. (1974). Beyond regression: new tools for prediction and analysis in the behavioral sciences.
Whittington, J. & Bogacz, R. (2015). An approximation of the error back-propagation algorithm in a predictive coding network with local hebbian synaptic plasticity. Cold Spring Harbor Laboratory.
Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2009). Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(2), 210–227.
Xie, X. & Seung, H. S. (2000). Spike-based learning rules and stabilization of persistent neural activity. Advances in Neural Information Processing Systems. 12, 199–208.
Yamins, D. L. K. & DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. Nature Neuroscience, 19, 356–365.
Yamins, D. L. K., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., & DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proceedings of the National Academy of Sciense U.S.A., 111(23), 8619–8624.
Zhou, Y. T. & Chellappa, R. (1988). Computation of optical flow using a neural network. IEEE 1988 International Conference on Neural Networks, 71–78.