

Color spaces emerging from deep convolutional networks

Ivet Rafegas

Computer Vision Center

C. Sc. Dpt. UAB. Bellaterra (Barcelona)

Maria Vanrell

Computer Vision Center

C. Sc. Dpt. UAB. Bellaterra (Barcelona)

Abstract

Defining color spaces that provide a good encoding of spatio-chromatic properties of color surfaces is an open problem in color science [8, 22]. Related to this, in computer vision the fusion of color with local image features has been studied and evaluated [16]. In human vision research, the cells which are selective to specific color hues along the visual pathway are also a focus of attention [7, 14]. In line with these research aims in this paper we study how color is encoded in a deep Convolutional Neural Network (CNN) that has been trained on more than one million natural images for object recognition. These convolutional nets achieve impressive performance in computer vision, and rival the representations in human brain. In this paper we explore how color is represented in a CNN architecture that can give some intuition about efficient spatio-chromatic representations. In convolutional layers the activation of a neuron is related to a spatial filter, that combines spatio-chromatic representations. We use an inverted version of it to explore the properties. Using a series of unsupervised methods we classify different type of neurons depending on the color axes they define and we propose an index of color-selectivity of a neuron. We estimate the main color axes that emerge from this trained net and we prove that color-selectivity of neurons decreases from early to deeper layers.

1. Introduction

Although color is by definition a property of a point of a surface, in most visual tasks it requires to be described as a non-isolated point. It usually appears influenced by the shape and types of materials of the surface, the lighting effects of the surround and the observer conditions. All of these, obligates to describe the spatio-chromatic properties of a surface as a whole. The representation of spatio-chromatic properties can be studied from different points of view.

In color science, color appearance models [8] have defined spaces and methods to describe spatial effects in color

perception. In human vision, one focus of attention to study the spatio-chromatic representations in the visual pathway has been measuring color selectivity of specific cells, concluding with the importance of single and double-opponent cells [14, 7]. In computer vision, where the goal is to build good computational models to perform visual tasks, such as object recognition, the evaluation of different methods to fuse color and local image features has attracted most of the interest [16].

Currently, in computer vision image feature selection and classification is mainly driven by the impressive results provided by deep convolutional networks. These networks are built with different architectures and are trained to perform different visual tasks. In general, they are based on hierarchical feedforward architectures combining different levels of convolutional and pooling layers. After being trained with large image datasets they provide excellent local color-feature selectors to encode invariant representations of complex objects at the top of the net.

Although these successful architectures are designed to solve engineering problems, they show some biological inspiration which can be proved in different aspects of these architectures: (a) a deep hierarchy similar to the different stages of the ventral stream of the human visual system, (b) layers based on a bank of convolution operations encoding the translation-invariant spatial properties of specific features across the visual field, and (c) the max pooling and subsampling steps that insert some local tolerance and introduce scale invariance along the hierarchy. These properties already appeared in previous bio-inspired models like HMAX [13]. Based on these ideas, Kriegeskorte in [10] has recently proposed deep neural networks as a framework to model biological vision and brain information processing.

Considering the amazing performance of human brain in object recognition, that is achieved with invariance to lighting, specularities or any surrounding influence that varies the color, in this paper we use a trained CNN to understand how color is represented in this architecture. Exploring the properties of how color is encoded in a CNN can allow a double outcome, on one side to get new inspiration about how spatio-chromatic representations can be improved and

on the other side a better understanding on how CNN is encoding visual information, that is a central topic in computer vision.

We perform two main analysis in the paper: (1) we build a decoded version of the spatial filter associated to a neuron, and we use it to classify the neurons in terms of their color representation, (2) we analyze the images which provoke maximum activations of a given neuron and we propose an index to evaluate their color selectivity property. This work is a first approach towards the understanding of color in CNN, and multiple research lines are opened from the conclusions we present after the analysis.

The paper is organized as follows, in the next section we define a CNN and afterwards we explain how we perform the filter projection, this is a generic estimation of the properties of the filter in the image space. Subsequently, we use a series of unsupervised methods to classify different type of neurons depending on the color axes they define. In the discussion section we estimate the main color axes that emerge from this trained net and we propose an index for color-selectivity of a neuron.

2. Inversion of the neuron activation

A Convolutional Neural Network is defined as several stacked layers operating on their inputs to produce a representation change, thus each layer yields a new level in the encoding process. Layers parameters are learned using the backpropagation algorithm, that search for a solution that minimizes a loss function that depends on the visual task the network is trained for. Mainly, two types of layers are used: convolutional and pooling. The main responsible layers on the encoding process are the convolutional ones, which apply a convolution operator between the input image representation and the set of filters of the layer trying to extract specific information from images. More explicitly, with this operation the image locations where the filter template best matches are highly activated. A CNN learns to match filter shapes with image structures which are important for the goal task. As layers are stacked, each filter shape is expressed in terms of the previous layer: each neuron is specialized through connections with other cells of the previous layer. This interpretation of the process also support with the fact that the filter bank of the first convolutional layer is easily understood compared to the rest. The pooling layers are devoted to reduce the image size to introduce some tolerance to spatial shifts and increasing the number of a spatial features at different scales. To sum up, each neuron at every layer is specialized in encoding specific image parts depending on the previous layers. Understanding how a CNN encodes image information to achieve such remarkable results is a focus of attention in computer vision [1, 2, 6, 18, 15, 17] that is not solved yet.

In this work, to explore the activation of a neuron to a

given input pattern we will work on a decoded version of the neuron filter. A decoded filter should be a projection of the neuron activity towards the image space. The convolution of an input image with a decoded filter should give an activation similar to the net activation. It should be computed by the network inversion. However, polling is not invertible, and convolution with a kernel is linear but not all kernels can be inverted. Therefore, both operations do not allow to compute the perfect inverse network. Consequently, we build an estimation of this decoded filter by making some specific assumptions on the lost information. We will denote the i th neuron of a specific layer L , as $n^{L,i}$, which is initialized with its corresponding filter, $F^{L,i}$ at level L . In what follows we explain the decoding process in two separate parts: convolution and pooling. The estimation of the neuron inversion will be iteratively computed from layer L through all intermediate layers l , this is denoted as $\hat{n}^{L,l,i}$, ending at layer 1, which represents the image space.

Inversion of Convolutional Layers: Inverting the encoding of a convolutional layer was firstly approached by Kavukcuoglu *et al.* [9] and afterwards, the stacking of several layers was performed in [19]. In both works, this inversion is approximated by the convolution with the transposed filter, which is called deconvolution. We project a neuron onto an inferior layer by deconvolving it with the set of filters of the neurons it is connected to. This step is different from what is done in [18], where the authors back the feature activities in intermediate levels to the input pixel space through the deconvolution. We map the filter pattern to the input pixel space. By doing this, instead of analyzing the image appearance that highly activate the neuron at a certain layer, we will focus on the properties of the built filter that can help in understanding the intrinsic neuron activation. The inversion of the convolutional layer is computed as:

$$\{\hat{n}_j^{L,l,i}\}_{j=1..c_l} = \left\{ \sum_{k=1}^{s_l} \hat{n}_k^{L,l+1,i} * f_j^{l,k} \right\}_{j=1..c_l} \quad (1)$$

where s_l denotes the number of filters of the layer l , c_l the number of channels of these filters, $\hat{n}_k^{L,l+1,i}$ is the k th channel of the estimated mapping of the neuron we are exploring, $n^{L,i}$, at layer $l + 1$, and $f_j^{l,k}$ is de j th transposed channel of the k th filter $F^{L,k}$. The symbol $*$ denotes the convolution operation.

Inversion of Pooling layers: Inverting pooling layers is not possible, they reduce the image size usually with a max pooling operation performed on a neighborhood region to keep the strongest activations in this zone. These layers simplify the information by capturing what is most relevant

in the area tolerating small spatial shifts. One way to approximate the inversion is to preserve the specific location where the maximum values of the activations came from [20], this is useful to recover the intermediate feature activities which are dependent of the image. Since we are recovering the filter itself without consider the activations of the images, this inversion can be approximate by a simple upsampling of the representation, then we define the unpooling operation as:

$$\Phi(\hat{n}_k^{L,l+1,i}) \quad (2)$$

where Φ denotes the image upsampling function that recovers the previous size of the representation considering the layer parameters. The loss of information is recovered by an interpolation method.

Other types of layers such as the Rectified-Linear Units (ReLU) layers are not considered in the inversion process of filters. These layers are usually devoted to inhibit negative responses of image activations. Taking into account that filters have been learned without these negative responses, inverting them would imply to insert information that did not participate in the training process.

Finally, we want to point out about the stride parameter. It has an important effect on the inverted shapes, since it is related to the spatial relationship between pixels. It introduces more errors in the inversion process of both, convolution and pooling. To deal with it we estimate its effect by a direct upsampling of the representation previous to any inversion process.

The results of applying the method explained in this section can be seen in figure 1. We can observe the decoded estimation of the neurons at different 5 convolutional layers of the CNN we study in this work. Its architecture is given in table 1 and it is explained with more details in section 5.

3. Extracting layer color axes

In this section we explore the decoded filters we have obtained above. As we can observe in figure 1, the appearance of the filters presents a huge variety both in shape and color. In this work we try to approach the understanding of color representation in CNNs through the analysis of these decoded filters. This exploration is a hard task, considering the amount of filters and their variety. We have observed an important correlation between the color of these filters and the color of their maximum activation images. This correlation is specially important in the first layers of the network. Following this idea, we explore the color axis represented by each filter and based on these axes, we will estimate the main color axis of each layer.

To this end, we perform a classification according to their color properties, which we use to get subsets of different decoded filters sharing some properties and be able to search

for those filters that define the color space for each layer. The analysis is done on the following opponent-color space, based on [11] but with all axis ranges compressed between -1 and 1 values:

$$\begin{aligned} O_1 &= (R + G + B - 1.5)/1.5, \\ O_2 &= (R - G), \\ O_3 &= (R + G - 2 \times B)/2 \end{aligned} \quad (3)$$

First, we classify the neurons considering their color correlation with a linear axis. We use the Principal Component Analysis method (PCA) to obtain the main axis that fits the color distribution. The line of the axis is given by the eigenvector direction with highest eigenvalue and the point given by the color mean. Depending on the dispersion of these color pixels in the space, we separate between linearly correlated filters (called aligned) against dispersed filters. In fact, color pixels of linearly correlated neurons can be expressed by the colored-axis obtained by this regression. This classification arises from the observation that aligned filters present an strong spatial correlation between channels, and consequently simpler spatial shapes. While non-aligned filters present complex shapes which are difficult to understand from the decoded filter.

Second, once aligned filters are found, we analyze the color variability of these aligned decoded filters in terms of the amount of color names labels that can be assigned. To this end we have applied the parametric model for the universal terms developed by Benavente et al in [3]. It categorizes each color pixel in one of the 11 basic colors defined by Berlin and Kay [4]: red, orange, brown, yellow, green, blue, purple, pink, black, gray and white. In this way, neurons can be classified as single-color neurons the ones which are categorized with one color, double-color neurons having two colors and we also consider multiple-color neurons when they presents a higher number of colors. In other hand, aligned neurons are also classified according to their color axis position in the opponent color space. Centered neurons are those whose axes cross the origin of the color space, and shifted neurons, those whose axes are away from the origin.

Given the previous classification, we finally focus on one of the classes, which is the one of double-color neurons, having a centered aligned axis. This specific subset is, somehow, defining the color space at the corresponding layer. Two main categorization can be done observing the amount of variance on the intensity channel: high variance corresponds to black and white selectivity, while low variance to double-color selectivity. From that point, we are able to select the main color axis for each layer: we project each color axis to the chromaticity plane RG-BY by subtracting intensity information of the double-color projected neuron axis and we analyze their distribution along this plane. Specifically, we consider the angle between their

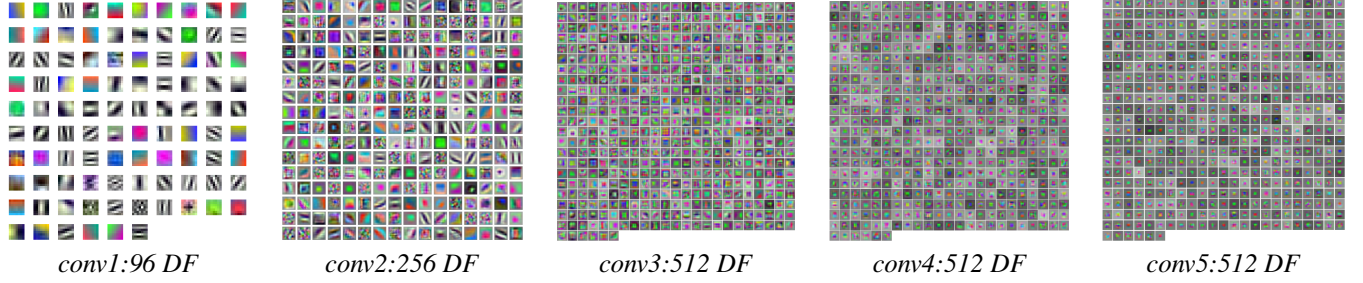


Figure 1: Neuron projection to image space. Decoded filters (DF) approximated for five convolutional layers of CNN-M net.

axes and the RG axis of the color space. This distribution is modeled through an expectation-maximization algorithm (EM) to find the mixture of gaussians that fits them. With this procedure, we obtain the color axis of each layer as the ones defined by the mean of each component distribution.

To sum up, we propose a neuron categorization process in order to get a subset of decoded filters that allow to explore relevant color directions of the layer. With the EM model we are able to extract the axes of the color space that emerge from these neurons.

4. Analyzing CNN color selectivity

Once we have extracted relevant axes of net layers, which were derived from decoded filters, another interesting point is the study individual neuron selectivity to a particular property. Several research studies on Convolutional Neural Networks have demonstrated that internal neurons are selective to a particular appearance of the object, or object style, viewpoint, etc. [15, 23, 18, 17, 1, 2]. In this section, we focus on the study of color selectivity of the neurons, trying to answer the question about how important is color selectivity in these architectures.

In order to do this, we apply an unsupervised classification method to analyze the color selectivity of a neuron from the study of the image parts having maximum activations. This process should give us the number of different colors appeared in these top images. We consider that a neuron is color selective if it is activated by images presenting a subset of specific predominant colors. Heretofore, we analyze the colors of the t top activations for each neuron in a random subset of the dataset.

Following this idea, we need to characterize the color of the image patches corresponding to a high activation of a certain neuron. Since we need a global color description, we will use the labels of a color naming approach [3]. Each pixel is transformed into a 11-dimensional space from its color probabilities to belong to a certain basic color. We build our wide-range description of this cropped image by clustering the set of pixels in k categories using the k -means

method onto the 11D space. The obtained centroids will allow to obtain compound color categories that capture more than eleven labels but preserving a global description of the image colors. In this way, each cropped image can be described by the histogram of the labels defining its predominant colors, this is the probability of finding a pixel with an specific label in the image.

To quantify the color selectivity of a neuron we will use the descriptors computed on the t top activation images. We will combine the histograms of labels of the t images, that is denoted as h . It is representing the probability to find a pixel with a specific label in the t images. A neuron with high index of color selectivity will concentrate most of the pixels on a small subset of labels, on the contrary, a neuron with low index of color selectivity will present a flat histogram with pixels in all the bins or labels.

We define the index of color selectivity index of a neuron, S_p , as the ratio of pixels contained by the p bins with highest probabilities, given by

$$S_p(n^{L,i}) = \frac{\sum_{m=1}^p h_m(n^{L,i})}{p} \quad (4)$$

where h_m is the m th maximum of the histogram descriptor h .

5. Results and discussion

In this paper we analyze the neurons of a CNN architecture trained on ImageNet ILSVRC dataset [12] for a generic visual task of object recognition, which contains around 1.2M of images classified in 1.000 categories. We use the CNN trained by Chatfield *et al.* in [6], where was referred as a medium net (CNN-M). Its architecture is summarized in table 1. This network is of note due to similar net shows a good representational performance when is compared to human one [5].

In this section we perform two experiments on CNN-M network, but we only focus on its convolutional layers without considering the last fully connected layers. First, we study the color axes in each convolutional layer using the

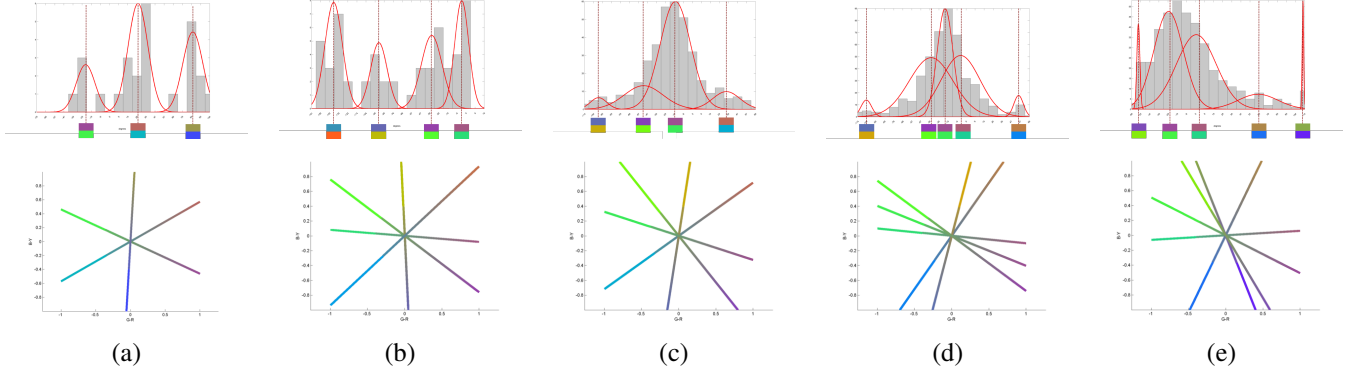


Figure 2: Color axes emerging from the convolutional layers of the explored CNN. (a) 3 color axes in conv1. (b) and (c) 4 color axes in conv2 and conv3, respectively. (d) and (e) 5 color axes in conv4 and conv5, respectively. For each convolutional layer, the first row is the distribution of double-color neurons for each chromaticity angle on an opponent space (RG-BY), gray bars correspond to the number of neurons, in red the estimated Gaussian mixtures modeling the emerging color axes. Second row, are the chromaticity axes corresponding to the estimated means of the mixture model provided given by the EM algorithm.

process explained in section 3 and second, we compute the index of color selectivity for all the neurons in convolutional layers following the method shown in section 4.

Figure 1 shows the set of neuron projections obtained as explained in section 2 in the different convolutional layers. From these projections, we can observe that decoded filters present simpler shapes in first layers, so complexity of shapes is increasing through layers and is more difficult to interpret these shapes in last layers. Moreover, lower layers present more separation between colors and black-white neurons. Nevertheless, we have to consider that the process of projecting each neuron accumulates errors through layers.

The aim of the first experiment is to analyze the color spaces emerged from this deep convolutional network. This color spaces are defined by the set of axes of the aligned, centered and double-color neurons. Table 2 illustrates the results of our classification for conv5. Aligned filters show simpler spatial shapes compared with the dispersed ones. As we explained in section 3 we use the EM Gaussian mixture model to get the principal color axis that this specific subset of neurons are defining. Note that this clustering algorithm is done from the chromaticity angle on the opponent space (RG-BY), and there are only considered angles between 0 and 180 degrees. This fact implies to fit our data in a kind of circular space. For this reason, we successively shift the last bin into the first position of the histogram and evaluate the EM algorithm by the AIC measure to choose the best one. We also consider different possible number of classes (from 3 to 10) and again AIC is used to get the final fitting. Figure 2 shows the probability of double-color neurons along the RG axes in all of the convolutional layers

studied in this paper. Surprisingly, in the first layer emerges a color space with 3 color axes. It is important to clarify that black and white axis is not considered in this study, so that first convolutional layer is somehow defining a 4-D color space: black-white, red-cyan, magenta-green and blue-yellow. This result can be related with the controversial debate of the existence of this third color channel also exposed in [21]. From second convolutional layer a 5D color space is emerged, adding a red-green layer compared to the color space emerged in conv1. In the same figure, we can observe that as we go deeper in the layers, the neuron color axes are covering a major range of hues. Nevertheless, we have to specify that white and black axes disappear from conv3. Moreover, from this results we can observe that most neurons tend to be expressed in reddish-greenish terms.

Finally, our second experiment determines the index of color selectivity through the different convolutional layers. Each neuron is studied from the 9th parts of images which provoke a high activation of that neuron ($t = 9$). We apply the methodology explained in section 4 to analyze the degree of selectivity of each layer. In figure 3 we plot the percentage of neurons for each convolutional layer that has a greater index of color selectivity value than a threshold ($th = 0.30, 0.40, 0.50, 0.60, 0.70$) fixing $p = 3$. This graphic shows that color selectivity clearly decreases as layer is deeper in the architecture. This fact implies that CNN is more color invariance as we go up in the net, as expected. Another interesting observation on these results, is that the highest decrement of selectivity neurons is done between conv2 and conv3, where there are no more black-white neurons.

<i>conv1</i>	<i>conv2</i>	<i>conv3</i>	<i>conv4</i>	<i>conv5</i>	<i>full6</i>	<i>full7</i>	<i>full8</i>
96x7x7	256x5x5	512x3x3	512x3x3	512x3x3	4096	4096	1000
<i>st. 2, pad. 0</i>	<i>st. 2, pad. 1</i>	<i>st. 1, pad. 1</i>	<i>st. 1, pad. 1</i>	<i>st. 1, pad. 1</i>	<i>dropout</i>	<i>dropout</i>	<i>softmax</i>
<i>LRN, x2 pool</i>	<i>LRN, x2 pool</i>			<i>x2 pool</i>			

Table 1: CNN-M architecture designed by Chatfield et al. in [6]. We use their notation, where $M \times N \times P$ corresponds to number of filters, number of rows and number of columns of the filters respectively. *St.* and *pad.* refers to stride and padding respectively; *LRN* is a Relu and the corresponding pooling (*pool*) if applied.

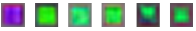
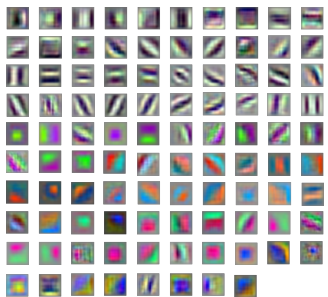
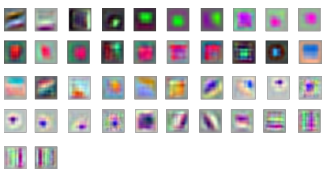
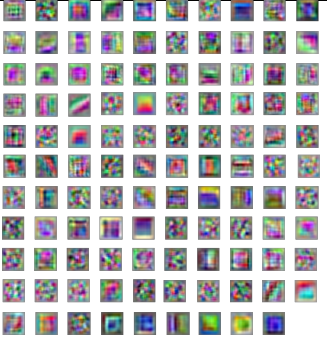
	<i>Aligned</i>		<i>Dispersed</i>
	<i>Centered Axes</i>	<i>Shifted Axes</i>	
<i>Single-color</i>			
<i>Double-color</i>			

Table 2: Classification of neurons in conv5 layer of the CNN-M. Neurons are classified in 4 different classes, depending on their linear correlation and their number of appearing colors.

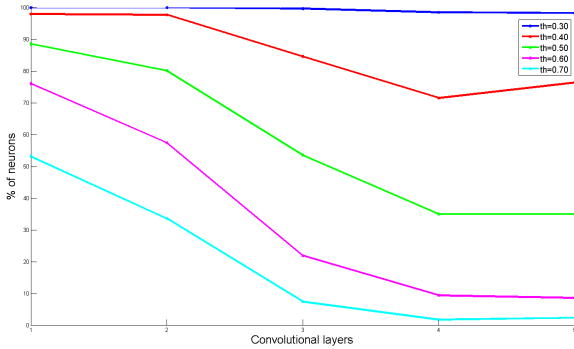


Figure 3: Color selectivity behavior through the different convolutional layers of the analyzed CNN. This plot compares the percentage of neurons per layer that has a index selectivity greater than a fixed threshold.

6. Conclusion

In this paper we have explored how color is represented in a Convolutional Neural Network from a projected version of each neuron. As a result we conclude that 3 chromaticity axes emerge in the first layer, instead of the classical Red-

Green and Blue-Yellow. We also can state that as we go up in the hierarchy, the estimated axes in the first two layers try to equally cover the full hue space, with the 3rd layer a major concentration on one specific axes is emerging, which in decoded filters is more aligned with a Red-Green axis. Black and White color axes is defined in first two layers, and a bit in the 3rd; but deeper layers have no specific black and white neuron. Finally, we also observe that color selectivity is an important feature in the first convolutional layers but it is decreasing through layers, this is an obvious conclusion since the decrease in selectivity must be accompanied by an increase in invariance that is a must for a good behavior of the net.

References

- [1] T. B. Alexey Dosovitskiy, Jost Tobias Springenberg. Learning to generate chairs with convolutional neural networks. In *CVPR*, 2015. 2, 4
- [2] M. Aubry and B. C. Russell. Understanding deep features with computer-generated imagery. In *ICCV*, 2015. 2, 4
- [3] R. Benavente, M. Vanrell, and R. Baldrich. Parametric fuzzy sets for automatic color naming. *Journal of the Optical Society of America A*, 25(10):2582–2593, Oct 2008. 3, 4
- [4] B. Berlin and P. Kay. *Basic Color Terms: their Universality and Evolution*. University of California Press, 1969. 3

- [5] C. F. Cadieu, H. Hong, D. L. K. Yamins, N. Pinto, D. Ardila, E. A. Solomon, N. J. Majaj, and J. J. DiCarlo. Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS computational biology*, 10:e1003963, 2014 Dec 2014. 4
- [6] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014. 2, 4, 6
- [7] B. R. Conway and D. Y. Tsao. Color-tuned neurons are spatially clustered according to color preference within alert macaque posterior inferior temporal cortex. *Proc Natl Acad Sci U S A.*, 42(106):1803418039, 2009. 1
- [8] M. D. Fairchild. *Color appearance models*. Wiley Ed., 3rd edition, 2013. 1
- [9] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. LeCun. Learning convolutional feature hierarchies for visual recognition. In *Advances in Neural Information Processing Systems (NIPS 2010)*, volume 23, 2010. 2
- [10] N. Kriegeskorte. Deep neural networks: A new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science.*, 1:417–446, 2015. 1
- [11] K. Plataniotis and A. Venetsanopoulos. *Color Image Processing and Applications*. Springer, 2000. 3
- [12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 4
- [13] T. Serre, L. Wolf, S. M. Bileschi, M. Riesenhuber, and T. A. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(3):411–426, 2007. 1
- [14] R. Shapley and M. Hawken. Color in the cortex: single- and double-opponent cells. *Vision Research*, 51(7):701–717, 2011. 1
- [15] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In ICLR 2015*, 2015. 2, 4
- [16] K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, Sept 2010. 1
- [17] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. In *Deep Learning Workshop, International Conference on Machine Learning (ICML)*, 2015. 2, 4
- [18] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *14th European Conference on Computer Vision (ECCV'14)*. 2, 4
- [19] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *CVPR*, 2010. 2
- [20] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, 2011. 3
- [21] J. Zhang, Y. Barhomi, and T. Serre. A new biologically inspired color image descriptor. In *12th European Conference on Computer Vision (ECCV'12)*, pages 312–324, 2012. 5
- [22] X. Zhang, B. A. Wandell, and B. A. W. A spatial extension of cielab for digital color image reproduction. *SID Journal*, 1996. 1
- [23] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In *In ICLR 2015*, 2015. 4