Article

Not peer-reviewed version

A Long Skip Connection for Enhanced Color Selectivity in CNN Architectures

Oscar Sanchez-Cesteros *, Mariano Rincon, Margarita Bachiller, Sonia Valladares-Rodriguez

Posted Date: 2 August 2023

doi: 10.20944/preprints202307.1957.v1

Keywords: Color selectivity; skip connections; long skip connection; CNN; VGG16; Densenet121; Resnet50; feature map visualization

Article

A Long Skip Connection for Enhanced Color Selectivity in CNN Architectures

Oscar Sanchez-Cesteros [1,*], Mariano Rincon [1], Margarita Bachiller [1] and Sonia Valladares-Rodriguez [1,2]

[1] Department of Artificial Intelligence, UNED, Juan del Rosal, 16 Madrid 28040
[2] Dept. of Electronics and Computing. University of Santiago de Compostela, USC
[*] Correspondence: osanchez75@alumno.uned.es

Abstract: Some recent studies show that filters in convolutional neural networks (CNNs) have low color selectivity in datasets of natural scenes such as Imagenet. CNNs, bio-inspired in the visual cortex, are characterized by their hierarchical learning structure, which appears to gradually transform the representation space. Inspired by the direct connection between the LGN and V4, which allows V4 to handle low-level information closer to the trichromatic input in addition to processed information that comes from V2/V3, we propose the addition of a long skip connection (LSC) between the first and last blocks of the feature extraction stage to allow deeper parts of the network to receive information from shallower layers. This type of connection improves classification accuracy by combining simple-visual and complex-abstract features to create more color-selective ones. We have applied this strategy to classic CNN architectures and quantitatively and qualitatively analyzed the improvement in accuracy while focusing on color selectivity. The results show that, in general, skip connections improve accuracy, but LSC improves it even more and enhances the color selectivity of the original CNN architectures. As a side result, we propose a new color representation procedure for organizing and filtering feature maps, making their visualization more manageable.

Keywords: color selectivity; skip connections; long skip connection; CNN; VGG16; Densenet121; Resnet50; feature map visualization

1. Introduction

Color selectivity refers to the ability of a system to respond selectively to specific colors or wavelengths of light inside its receptive field, that is, to locate patterns based on chromatic characteristics alone or color mixed with other visual features such as texture or shape. Taylor et al. [1] showed that CNNs presented near-orthogonal color and form processing in early layers, but increasingly intermixed feature coding in higher layers. Previously, several authors [2–4] have shown that CNN color selectivity is low in models trained with large datasets of natural scenes such as Imagenet, which are biased towards grayish, orangish, and bluish colors. It seems clear that increasing color selectivity is necessary to take full advantage of the information in the color channels, especially in scenarios where color is a primary characteristic.

The reason for this low color selectivity may be in the architecture itself. CNNs were inspired by early findings from the study of biological vision[5]. Since Fukushima with the Neocognitron [6], who built a basic block based on V1, or Lecun et al. with the Lenet architecture [7], who continued the idea of aggregating simpler features into more complex ones with a sequential architecture of repetitive building blocks, CNNs have basically maintained the same architecture. This representational hierarchy appears to gradually transform a space-based visual into a shape-based and semantic representation [8].

However, in the last thirty years, neuroscience has evolved in the understanding of the visual area and its active role in perception, as we can see in Hubel's later work [9] or that of Davila Teller [10] , where they advanced the study of the relationship between vision and the vision system using physiological and perceptual techniques. Although the process is mainly sequential and follows the scheme "$retina \rightarrow LGN \rightarrow V1 \rightarrow V2 \rightarrow V3 \rightarrow V4$", there are connections between these areas that do not follow this sequentiality. Some models have been proposed that partially skip this sequential structure [11–14] but none have been proposed to improve color selectivity.

In 1994, Ferrera et al. [15] verified that the information structure of the LGN was maintained in the macaque visual area V4, which indicates that this area, specialized in the detection of shapes and colors [16,17], processes information coming from areas V1, V2, and V3, where color is intermixed for the detection of edges, shapes, and textures, and, in addition, from the LGN, where color and other features are present but at a lower level of processing. The author hypothesized that this structure of unprocessed information from the LGN is maintained in areas V1, V2, and V3 in mammals. Recently, these direct connections between the precortical area and neocortex visual areas have been analyzed in humans, indicating that there is a direct and bidirectional connection between the LGN and V4 [18].

In this study, inspired by the direct connection $LGN \rightarrow V4$, we propose the following hypothesis: it is possible to increase the color selectivity of a feed-forward architecture and improve its performance in classification tasks by modifying the structure of the network to reflect this direct connection. We do this by creating a long skip connection (LSC) between the output of the initial block, equivalent to the LGN, and the input of the last block of the feature extraction stage of the network so that this last block, similar to area V4, processes both branches together.

To evaluate this proposal, we selected several classic CNN architectures, VGG16, Densenet121, and Resnet50, which have sufficient blocks for the proposed connection to be functional and present different types of skip connections. Specifically, VGG16 does not have skip connections, Densenet121 uses skip connections via concatenation and Resnet50 uses skip connections via addition, which allows us to analyze the LSC extensively.

Accordingly, the remainder of the paper is organized as follows. Section 2 presents preliminary background information related to studying color selectivity in CNNs and different skip connection typologies. Section 3 describes the implementation of LSC on several classic CNN architectures, and a simplified procedure to analyze quantitatively and qualitatively color selectivity. Section 4 describes the experiments conducted to demonstrate color selectivity improvements of models with LSC. Section 5 shows experimental results and discusses the results along with observations. Finally, Section VI concludes the paper.

2. Related work

2.1. Color selectivity in CNN

Few studies have analyzed color selectivity in neural networks. Usually, the problem has been addressed pragmatically, testing different color representations and choosing the one with the best performance. In the last years, color selectivity in CNNs has been studied by analyzing the activation of neurons in the network in response to particular stimuli. Building on the work of Shapley and Hawken [19] on how color is encoded in the human vision system, Rafegas et al. [3,20] proposed a method to explore how color is encoded in a trained artificial network. They defined a representation to visualize the neuron activity that they named "Neuron Feature" (NF) and consists of an image obtained by averaging the set of N-top image crops that maximally activate the neuron. Although this representation has certain limitations since it is based on the assumption that the color properties of kernels equal the color properties of their corresponding mean image patches, this qualitative representation is a proxy for what the neuron responds to: If NF has a well-defined hue, the neuron responds only to that range of hues, whereas if it has a grayish color, it responds to a wider range of colors, i.e., its response is less color dependent. In addition, they defined a color selectivity index (CSI) for each neuron in the network by comparing its activation in response to color and grayscale images. Engilberge et al. [2], also defined a color sensitivity index by comparing neuronal activation in color and grayscale images, but focused exclusively on quantitative metrics. Later, several methodologies for the analysis of color opponency and spatial and color tuning have been defined [4,21,22], but they use prepared datasets which are beyond the scope of our study.

2.1.1. Color selectivity analysis of individual neurons

According to Rafegas and Vanrell's methodology [3,20], an "image crop" represents the image area of the receptive field that activates a neuron. For qualitative analysis, they visualized the kind of inputs that trigger the neuron's response in the spatial-chromatic domain by means of the concept of Neuron Feature (NF), which is the weighted average of the N image crops that activate the neuron the most in the dataset (Eq. 1).

$$NF(n_i) = \frac{1}{N_{max}} \sum_{j=1}^{N_{max}} w_{j,i} I_j \tag{1}$$

where n_i is neuron i (neuron i that belongs to a feature map f of a layer L); N_{max} is the configurable parameter that sets the number of images that most activate the neuron to be considered; and $w_{j,i}$ is the weighting factor for the image crop of image j, I_j. This weighting factor $w_{j,i}$ is defined as the activation value normalized to the highest activation of the neuron in the entire dataset (Eq. 2).

$$w_{j,i} = \frac{a_{j,i}}{a_{max,i}} \tag{2}$$

where $a_{j,i}$ is the activation value obtained by n_i in I_j, and $a_{max,i} = max(a_{k,i}) \forall k$ is the highest activation of n_i in the entire dataset.

For quantitative analysis, they defined the color selectivity index (CSI), which was obtained by comparing the value of $w_{j,i}$ of the original color image with the value $w'_{j,i}$ of the same image in grayscale (Eq. 3). CSI values are in the interval [0,1], with the extreme cases CSI = 0 (no color selectivity) when $w_{j,i}$ and $w'_{j,i}$ are equal, and CSI = 1 (maximum color selectivity) when $w'_{j,i}$ is 0.

$$CSI(n_i) = 1 - \frac{\sum_{j=1}^{N_{max}} w'_{j,i}}{\sum_{j=1}^{N_{max}} w_{j,i}} \tag{3}$$

Figure 1 shows five NF from the last layer of the fifth convolution block with low, medium, and high CSI values. We choose neurons at this level to show the relationship between CSI and NF because this is where we will further analyze the benefits of the proposed architecture. When CSI is low, the image crops that activate the neuron the most (Figure 1b) have heterogeneous hue ranges (yellows, whites, greens, blues, etc.), whereas when CSI is high, the images have highly localized hue ranges. This is also visible in the color hue distribution of the pixels in the top 100 images of Figure 1c, which shows that it is unimodal and narrow when the CSI is high, whereas it is multimodal and flat when the CSI is low.

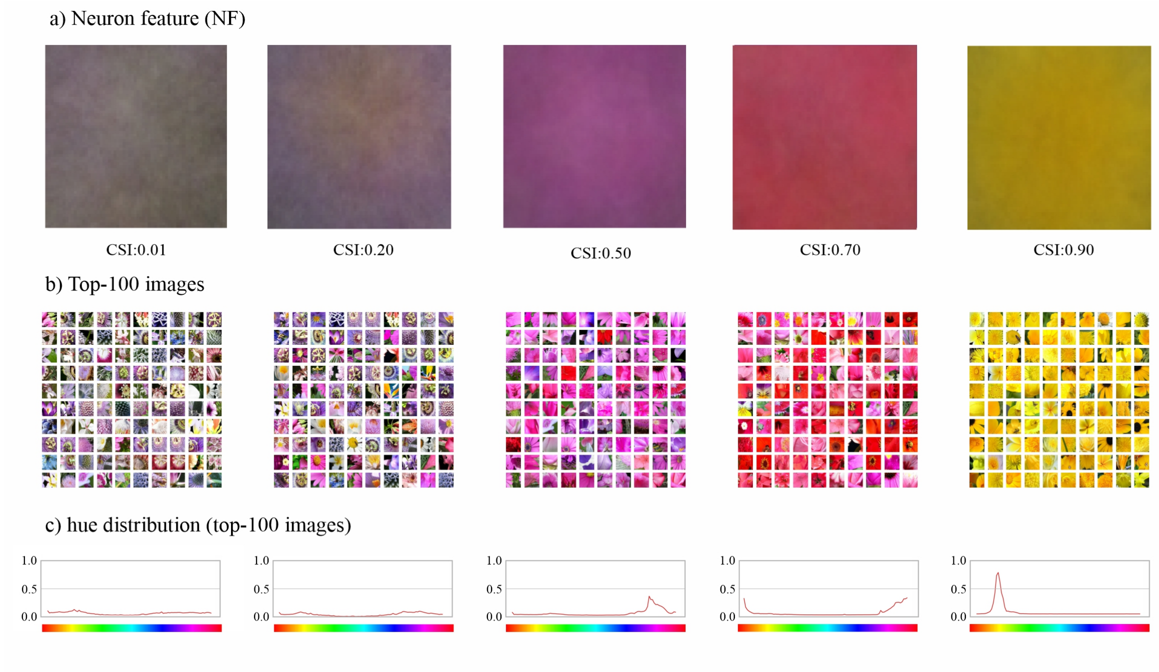


Figure 1. Examples of NF associated to neurons of the last layer of the fifth convolution block with different CSI values: a) NF images; b) image crops of the top 100 images that activate the neuron the most; and c) color hue distribution of the 100 image crops.

2.2. Skip connections

Skip connections are a type of shortcut that connects the output of one layer to the input of another layer that is not adjacent to it. Srivastava et al. [11] introduced the concept of skip connections with their Highway Networks, which are characterized by the use of gating units which learn to regulate the flow of information between layers. However, skip connections achieved fame with He et al.'s work on Residual Networks (ResNet) [12], which, although a particular case of the previous ones, was the foundation of the 1st-place winning entries in all five main tracks of the ImageNet and COCO 2015 competitions, which covered image classification, object detection, and semantic segmentation. These networks use skip connections via addition to learning residual functions with reference to the layer inputs. They allowed to solve the degradation problem, which facilitates the design of deeper neural networks.

Huang et al. [13] proposed another architecture that also uses skip connections: Densenet. In this case, Densenet uses skip connection via concatenation of each layer with the following ones to encourage feature reuse and also to alleviate the vanishing-gradient problem. However, an essential part of CNNs is the down-sampling of layers, which reduces the size of feature maps through dimensionality reduction to achieve higher computation speeds. To enable this, Densenets are divided into Dense Blocks sequentially connected, with no skip connections between them.

Another architecture that uses skip connections is U-net (Ronneberger et al. [14]), which is a symmetrical encoder-decoder architecture in which each decoder layer receives the output of the same layer of the encoder concatenated with the input of the previous layer of the decoder. This architecture was used to learn fine-grained details, making it ideal for semantic segmentation.

There are other popular architectures, such as Inception [23] or Xception [24], that we have not analyzed; however, the reader can note that skip connections have always been used within the layers of a block or between symmetrical layers of an encoder-decoder architecture, but not between blocks in classification tasks, and not for the purpose of enhancing color selectivity.

doi:10.20944/preprints202307.1957.v1

3. Methods

In this section, we describe the implementation of LSC on the VGG16, Densenet121 and Resnet50 architectures and introduce two new procedures for analyzing color selectivity to show the improvements it introduces. To simplify the notation, we will refer to a generic architecture with M (Model) accompanied by the subscripts O when referring to the original architecture (M_O) and with LS when referring to the architecture with LSC (M_{LS}). In addition, we will use VGG to refer to VGG16, *Densenet* for Densenet121 and *Resnet* for Resnet50. Therefore, for example, to refer specifically to the VGG16 architecture modified with LSC, we will use the notation VGG_{LS}.

3.1. The proposed LSC architecture

As mentioned in the introduction, we propose establishing an LSC between the output of the first block and the input of the last block of the feature extraction stage of the network. Our proposal was tested on three classic CNN architectures with and without skip connections: VGG, *Densenet*, and *Resnet*. All of these architectures use a similar down-sampling strategy along the network. However, there were differences between them.

We numbered the blocks of the feature extraction stage according to the pooling operations: 112x112 (first block), 56x56 (second block), 28x28 (third block), 14x14 (fourth block), and 7x7 (fifth block). The classification stage consists of one or more dense layers. Figure 2 shows the structure of both architectures: the original, M_O, and the one with LSC, M_{LS}. Figure 2c shows the receptive field of a neuron in the fifth block of M_{LS}. In the fifth block, the receptive field is composed (by addition or concatenation) of the fourth block output (forward path) and the LSC that originates from the first block. The LSC reduces the dimensions of the first block output by max-pooling, allowing the composition.

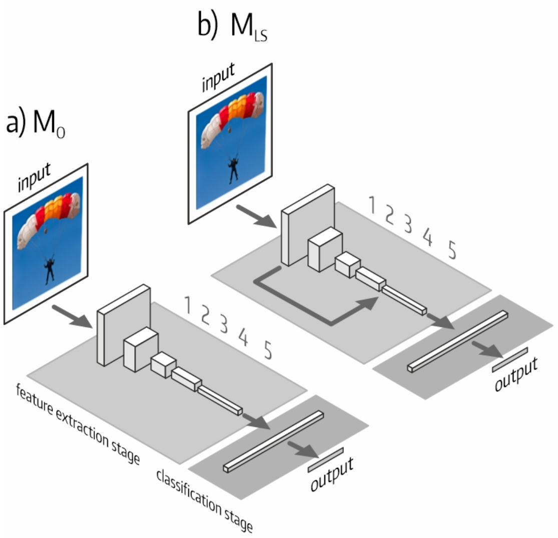


Figure 2. LSC architecture inspired in the connection between LGN and V4. a) Original architecture (M_O); b) new LSC architecture (M_{LS}); and c) detail of the receptive field of a neuron of the fifth block, which results from the composition of the fourth block output (forward path) and the LSC that comes from the first block.

Tables 1–3 list the details of the implementation of the three architectures. In VGG, the architecture is based on similar blocks (convolution layers plus an output max-pooling layer). In *Densenet* and *Resnet*, there is a convolution layer at the input and then five and four functional blocks respectively. While *Resnet* and VGG maintain the same dimensionality reduction pattern in the last block (the last block input is 14x14 and its output is 7x7), in *Densenet*, the last block input is 7x7 and its output is also 7x7. We now examine each model in detail.

Table 1. VGG and VGG_{LS} architectures. The blocks are composed of several convolution layers and a max-pooling layer at the output. The max-pooling is 2x2 and stride=2. The LSC of VGG_{LS} performs three max-pooling operations at the output of the first block.

block	output	VGG	filters	VGG_{LS}	filters
1	224x224	3x3 conv x2	64	3x3 conv x2	64
	112x112	max-pool	64	max-pool	64
2	112x112	3x3 conv x2	128	3x3 conv x2	128
	56x56	max-pool	128	max-pool	128
3	56x56	3x3 conv x2	256	3x3 conv x2	256
	56x56	1x1 conv	256	1x1 conv	256
	28x28	max-pool	256	max-pool	256
4	28x28	3x3 conv x2	512	3x3 conv x2	512
	28x28	1x1 conv	512	1x1 conv	512
	14x14	max-pool	512	max-pool	512
LS	14x14			max-pool	64
				concat	
5	14x14	3x3 conv x2	512	3x3 conv x2	512
	14x14	1x1 conv	512	1x1 conv	512
	7x7	max-pool	512	max-pool	512
classification stage	1x1	FC	1	FC	1
	1x1	FC	1	FC	1
	1x1	soft-max		soft-max	

The VGG architecture consists of five blocks in the feature extraction stage, and two fully connected layers and one softmax layer in the classification stage (Table 1). Each block has several convolution layers and a max-pooling layer that halves the size of the feature maps. The number of filters increases from 64 in the first block to 512 in the last block. The LSC in VGG_{LS} is established between the output of the first block (112x112) and the fourth (14x14) using three 2x2 max-pooling operations. Therefore, the fifth block receives the concatenated output of the fourth block and the down-sampled output of the first block. The classification stage has two dense layers of 4016 units and a soft-max layer that depends on the number of classes in each dataset.

Table 2. $Densenet$ and $Densenet_{LS}$ architecture. The max-pooling layer of the first block is 3x3 and stride=2. Each Dense block is composed of a 1x1 convolution layer and a 3x3 convolution layer. The transition blocks have a 1x1 convolution layer and a 2x2 average-pooling layer with stride=2 The average pool of the first layer and that of the output layer is 7x7 with stride=2. For the LSC of $Densenet_{LS}$, four 3x3 max-pooling are performed to reduce the dimension of the first block and concatenate with the fourth block.

block	output	$Densenet$	filters	$Densenet_{LS}$	filters
1	112x112	7x7 conv	48	7x7 conv	48
	56x56	max-pool	48	max-pool	48
2	56x56	dense x6	192	dense x6	192
	28x28	transition	96	transition	96
3	28x28	dense x12	384	dense x12	384
	14x14	transition	192	transition	192
4	14x14	dense x24	768	dense x24	768
	7x7	transition	384	transition	384
LS	7x7			max-pool	48
				concat	
5	7x7	dense x16	768	dense x16	816
	7x7	1x1 conv	768	1x1 conv	816
classification stage	1x1	ave-pool	1	ave-pool	1
	1x1	soft-max		soft-max	

The *Densenet* architecture is composed of "dense blocks" followed by a "transition layer", which halves the size of the feature maps (see Table 2). Each dense block has three convolution layers: 1x1, 3x3, and 1x1. The number of filters in each layer varies as the network advances in depth from 64 to 2048, as well as the number of repetitions, which are concatenated. The transition layer is composed of a 1x1 convolution layer and a 2x2 and stride=2 average pooling layer. The first block comprises the first convolution layer and the max-pooling layer, which reduces the dimension to 56x56. In the fifth block, unlike in *VGG*, where the dimension is reduced from 14x14 to 7x7, the fifth block has an input of 7x7 and no reduction. Therefore, the LSC from the first block reduces the dimension to 7x7. The classification stage has a soft-max layer that depends on the number of classes in each dataset.

Table 3. *Resnet* and $Resnet_{LS}$ architecture. The max-pooling is 3x3 with stride=2, and the average-pooling is 7x7 and stride=2. Each Resnet block is a 1x1 conv layer, followed by a 3x3 conv layer, and ending with a 1x1 conv layer. For the LSC of $Resnet_{LS}$, three 3x3 max-pooling are performed to reduce the dimension of the first block and a 1x1 conv layer with 1024 filters for adding it to the fourth block output.

block	output	*Resnet*	filters	$Resnet_{LS}$	filters
1	112x112	7x7 conv	64	7x7 conv	64
2	56x56	max-pool	64	max-pool	64
	56x56	resnet x3	256	resnet x3	256
3	28x28	resnet x4	512	resnet x4	512
4	14x14	resnet x6	1024	resnet x6	1024
LS	14x14			max-pool	62
	14x14			conv 1x1	1024
				addition	
5	7x7	resnet x3	2048	resnet x3	2048
classification stage	1x1	ave-pool	1	ave-pool	1
	1x1	soft-max		soft-max	

The *Resnet* architecture is composed of "resnet blocks" containing three convolution layers: 1x1, 3x3, and 1x1, with a max-pooling of 2x2 and stride=2 at the end to reduce the dimension. The output block has a 7x7 average-pooling and a soft-max layer that depends on the number of classes in each dataset. The LSC is established between the output of the first block and the output of the fourth block, performing an adding operation to maintain the type of skip connections used in this architecture. Owing to this addition operation, we increase the number of filters of the output of the first block from 64 to 1024 using a 1x1 convolution layer.

3.2. Evaluation of color selectivity

To demonstrate that the improvements introduced by the LSC are related to color processing, we performed two types of analysis: a quantitative analysis of the filter distribution according to color selectivity and a qualitative analysis of the filter's response to color hue.

As mentioned above, we focused our analysis on the fifth block, as this is where the information coming from the different layers and from the LSC connection is combined to create the final features to be used in the classification stage.

3.2.1. Color selectivity properties of filters.

To analyze color selectivity, we will use the method of Rafegas and Vanrell described in Section 2.1.1 to obtain the color selectivity index (CSI) and the neuronal feature image (NF) of each neuron. However, these values are oriented towards a neuron-level analysis. As it is impractical to analyze the response of all neurons individually, we simplified the analysis by selecting a representative neuron of each layer, reducing the number of elements to manage.

We evaluated the CSI values in the neurons of each feature map of the fifth block output and found that their variance followed a decreasing exponential distribution (a^{bx}; a= 109.76; b= -73.60), which is very narrow and indicates that any neuron in the feature map can be representative of the filter behavior. Therefore, for each feature map, we selected an active and centered neuron to obtain the CSI and NF values of the filters. The CSI difference of this representative neuron with respect to the average of the feature map is $0.07(0.000025SD)$, which we consider insignificant for this study.

3.2.2. Qualitative Color selectivity analysis

A feature map shows the filter response to an input image. In order to analyze differences between M_O and M_{LS} responses, we can compare all the feature maps of each block output. However, it is very complicated and cumbersome when the block has many feature maps because it is necessary to visualize all these monochrome images together (512, 768, and 2048 feature maps in VGG, Densenet, and Resnet, respectively).

To facilitate the analysis of color selectivity for a particular image, we propose the following color representation of the feature maps in the HSV space: H (hue) is the average hue of the NF of the filter that generates the feature map, S (saturation) is the CSI value of the filter, and V (value) is the normalized activation value $w_{j,i}$ for image j. With this representation, we can reorganize the feature maps according to the CSI intervals and select the most active filters for each one. This reduces the number of images displayed, which remarkably simplifies the analysis and, as will be seen later, does not influence the results.

In Figure 3, we show an example of how the new representation facilitates the analysis of the feature maps at the fifth block output in VGG_{LS} (we just show a sample of 100 out of 512 feature maps for the sake of clarity). Given the input image shown in Figure 3a, Figure3b shows the feature maps ordered according to their positions in the architecture, which is difficult to follow. Figure 3c is an intermediate representation in the HSV space that groups feature maps in 12 hue ranges. The light gray boxes represent hue ranges without feature maps. In this representation, the maps are sorted by their hue range and intensity (average activation), facilitating the visualization of the visuospatial features of the original image they capture, such as contours, shapes, textures, and specific areas. Finally, to further simplify the analysis, the highest intensity feature maps for each hue range and CSI interval are selected (Figure 3d).

doi:10.20944/preprints202307.1957.v1

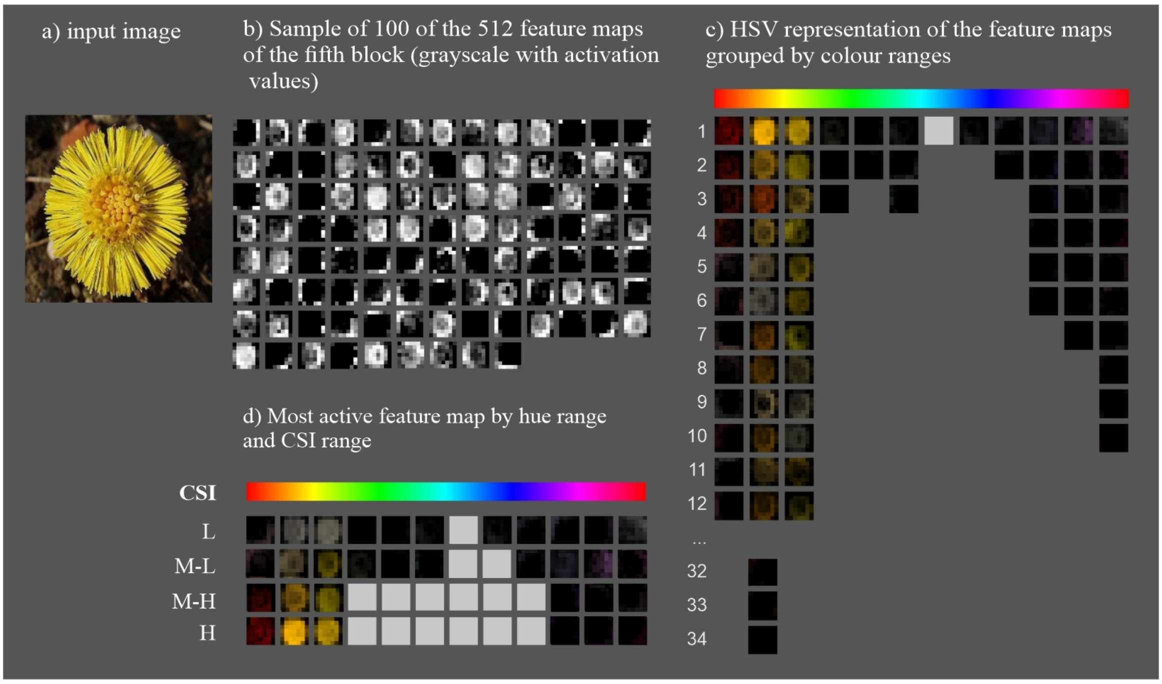

a) input image
b) Sample of 100 of the 512 feature maps of the fifth block (grayscale with activation values)
c) HSV representation of the feature maps grouped by colour ranges
d) Most active feature map by hue range and CSI range
CSI
L
M-L
M-H
H
1
2
3
4
5
6
7
8
9
10
11
12
...
32
33
34

Figure 3. Color representation of the feature maps of the fifth block output layer to facilitate color selectivity analysis: a) input image; b) grayscale representation of the sample of 100 out 512 feature maps in the fifth block output layer in VGG_{LS} architecture; c) HSV representation of the feature maps grouped by hue ranges and sorted by average activation value; and d) simplified visualization consisting of the feature map with the highest average activation per hue range and CSI interval.

4. Experimental setup

4.1. Datasets

Table 4 and Figure 4 summarize the characteristics of the four datasets used in our experiments: Imagenette [25], Tiny Imagenet [26], Birds [27], and Flowers [28]. We chose these datasets because of their varied themes (natural scenes, people, artifacts, animals, flowers, etc.) and different color characteristics, with images containing monochrome and polychrome elements, with high and low saturation, and with a variable number of classes (from 10 to 315) and images (between 5,000 and 100,000). Furthermore, there is an interest in the community to improve the performance of models with small datasets [29], since in the industry it is complex to obtain enough examples with homogeneous quantities to represent each class. In this scenario, where it is necessary to improve the feature extraction capability, the comparison of the proposed improvement will be more effective than using large datasets. Imagenette and Tiny Imagenet are two datasets derived from Imagenet [30]. The objects of interest appear in different sizes and positions, complete in the foreground or segmented and occluded by other objects. Birds and Flowers are two specialized datasets, one on birds and the other on flowers. The first one has the objects of interest in the foreground, and the second one in the foreground and background. All four datasets contain mainly natural scenes, so their color hue distribution shows a high concentration of pixels in the orange range (due to the presence of organic objects) and somewhat less in the bluish range (mainly due to the presence of sky or water regions in the background). Only the Flowers dataset contained no peak in the bluish range.

Table 4. Datasets used in the experiments.

Dataset	Classes	train/validation	Theme
Imagenette [25]	10	9,469/3,925	varied
Tiny Imagenet [26]	200	100,000 / 10,000	varied
Birds [27]	315	45,995 / 1,590	specialized
Flowers [28]	102	6,551 / 1,637	specialized

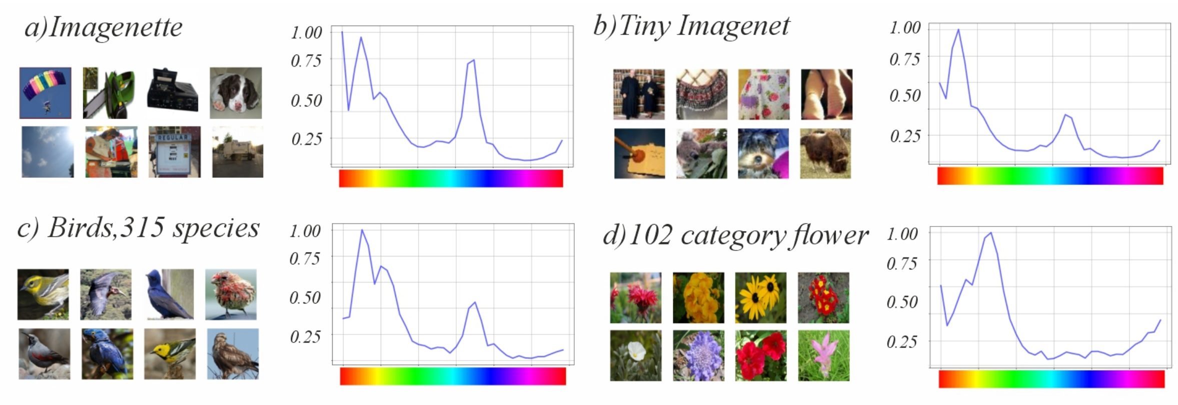


Figure 4. Overview of the four datasets used to evaluate the models.

4.2. Setup

Because our purpose is to analyze the performance improvement of the new architecture with respect to color selectivity, we used the default configuration parameters offered by Keras to train the models, without any technique to improve accuracy (e.g., preprocessing, batch normalization, etc.). We used a learning rate lr=0.00001 in VGG to prevent dead neurons, and lr=0.0001 in Densenet and Resnet; batch size=32; Adam optimizer; and cross-entropy as the loss function. We selected the model with the highest validation accuracy in each training session (usually obtained in approximately 15 epochs), and then chose the one with the best performance in five training sessions.

4.3. Evaluation

A quantitative and qualitative analysis of the effect of color selectivity on the performance of the different architectures was carried out on the fifth block output, where the impact of LSC appears. This is the last block of the feature extraction stage, where the information from the forward path and the LSC is merged, and the final visuospatial features used in the classification stage are generated.

First, we compared the global accuracy of M_O and M_{LS} architectures. Next, to understand the effect on performance of filters of different CSI intervals, we conduct an ablation study where we "turn off" filters based on their CSI. Finally, we used the method proposed in subsection Methods.B to evaluate the model response to color hues.

4.3.1. Performance analysis

We followed the methodology described in the subsection Related Work.A to obtain the NF and CSI values of every filter of the fifth block output based on a subset of 3000 images from each validation dataset, except for Flowers, where there are only 1500 images. The 100 images that most activated the selected neuron of each filter were used. We assessed whether there was any improvement in performance with M_{LS} by calculating the difference in accuracy between M_O and M_{LS}.

4.3.2. Filter CSI analysis

We conducted an ablation study of the filters of the fifth block output as a function of their CSI to evaluate their impact on classification accuracy. Although ablation techniques have been used in

CNNs [31], there have been no previous studies on color selectivity. We checked several CSI interval sizes to group filters. Initially, we tested a small size CSI interval (bin size = 0.1, ten intervals) and detected no significant changes. Therefore, we increased the interval size (bin size = 0.25) and obtained four CSI intervals: low (L) = [0, 0.25], medium-low (M-L) = [0.25, 0,50], medium-high (M-H)= [0.50, 0,75], and high (H) = [0.75, 1]. Significant differences were observed in this case.

The procedure to analyze the different architectures for each CSI interval first calculates the effect of ablation on the model accuracy, $DA_{M_X,csi}$ (Eq. 4), and then the difference between this effect in the M_O and M_{LS} variants, $DM_{M,csi}$ (Eq. 5).

$$DA_{M_X,csi} = Acc(M_X) - Acc(M_{X,A_{csi}}) \tag{4}$$

where M_X in $\{M_O, M_{LS}\}$ represents a particular architecture; $csi \in \{L, M-L, M-H, H\}$ is the CSI interval; A_{csi} represents the ablation of the filters on CSI interval csi; $Acc(M_X)$ is the accuracy without ablation; and $Acc(M_{X,A_{csi}})$ is the accuracy after ablation of filters in CSI interval csi.

$$DM_{M,csi} = DA_{M_{LS},csi} - DA_{M_O,csi} \tag{5}$$

where $M \in \{VGG, Dense, Resnet\}$ and $csi \in \{L, M-L, M-H, H\}$. $DA_{M_X,csi}$ indicates the contribution of the feature maps of a particular CSI interval to the final performance and $DM_{M,csi}$ indicates whether the ablation affects M_O or M_{LS} more. Therefore, the M_{LS} variant improves color selectivity if the difference $DM_{M,csi}$ is positive and larger in higher CSI intervals, where the filters are more color selective.

4.3.3. Color hue analysis

We conducted an experiment to compare the responses of the fifth block output filters to different hues. Instead of using a monochrome image, which would provoke an unknown filter response because it is outside the dataset population, we used an image of the "green broadbill" in the Birds dataset in which hue is one of the most representative features of the bird and whose shape-color combination is not very present in the dataset. We created two synthetic sets of images with different hues (red, yellow, green, cyan, blue, and magenta), one with texture details and the other with only the silhouette, as in the work of Taylor et al. [1]. We use both sets of images to determine whether the figure is detected solely based on its contours or whether its texture, including different shades of color, also plays a role. Hue is a significant color characteristic, and this experiment will evaluate whether there is an improvement in selectivity concerning hue, both in monochromatic signals (silhouettes) and in those with intensity variation (texture), and in what hue ranges (those very present or not so present in the dataset).

4.3.4. Qualitative analysis of filter response

Finally, to complete the comparative study, we used the color representation of the feature maps of the fifth block output to qualitatively characterize the differences in the types of features to which the filters respond. To achieve this, we analyzed the feature maps of several randomly selected images from each dataset.

5. Experimental Results

5.1. Global performance analysis

Table 5 lists the accuracy of each model for the validation datasets. Analyzing accuracy, we draw attention to the following results:

- All M_{LS} variants improved accuracy compared to M_O.

- The accuracy achieved in each dataset varied significantly: [64-75]% in Imagenette, [20-29]% in Tiny Imagenet, [47-77]% in Birds, and [25-57]% in Flowers.
- Among the architectures, *Densenet* achieved the highest accuracy in almost all datasets, with few differences with respect to *Resnet*. *VGG* performed worse, especially in Birds and Flowers.
- VGG_{LS} achieved accuracy values similar to the original $Densenet_O$ and $Resnet_O$.

Table 5. Accuracy obtained by the M_O and M_{LS} architectures in the four datasets and absolute (abs.) and relative (rel.) differences between them.

	VGG				Densenet				Resnet			
	accuracy		difference		accuracy		difference		accuracy		difference	
Dataset	***o***	***ls***	**abs.**	**rel.**	***o***	***ls***	**abs.**	**rel.**	***o***	***ls***	**abs.**	**rel.**
IM	64%	74%	10%	14%	**68%**	**75%**	7%	9%	66%	70%	4%	6%
TI	20%	23%	3%	13%	**28%**	**29%**	1%	3%	24%	27%	3%	11%
BI	47%	62%	15%	24%	60%	**77%**	17%	22%	**65%**	74%	9%	12%
FL	25%	43%	18%	42%	**48%**	**57%**	9%	16%	45%	**57%**	12%	21%

IM: Imagenette; TI: Tiny Imagenet; BI: Birds; and FL: Flowers.

Regarding the differences in accuracy between M_O and M_{LS}, we highlight that:

- The highest differences were produced in Birds and Flowers, independently of the architecture.
- *VGG* had the highest relative difference in all datasets (42% trained on Flowers stands out).
- *Densenet* had the lowest relative difference (3%) on Tiny Imagenet.

In brief, LSC improved the accuracy of the architecture regardless of whether the original architecture already had skip connections.

5.2. Filter CSI analysis

Figure 5 shows the distribution of the fifth block output filters per CSI interval. We highlight that the distribution of filters per CSI interval varied with the dataset and was quite similar for all the architectures.

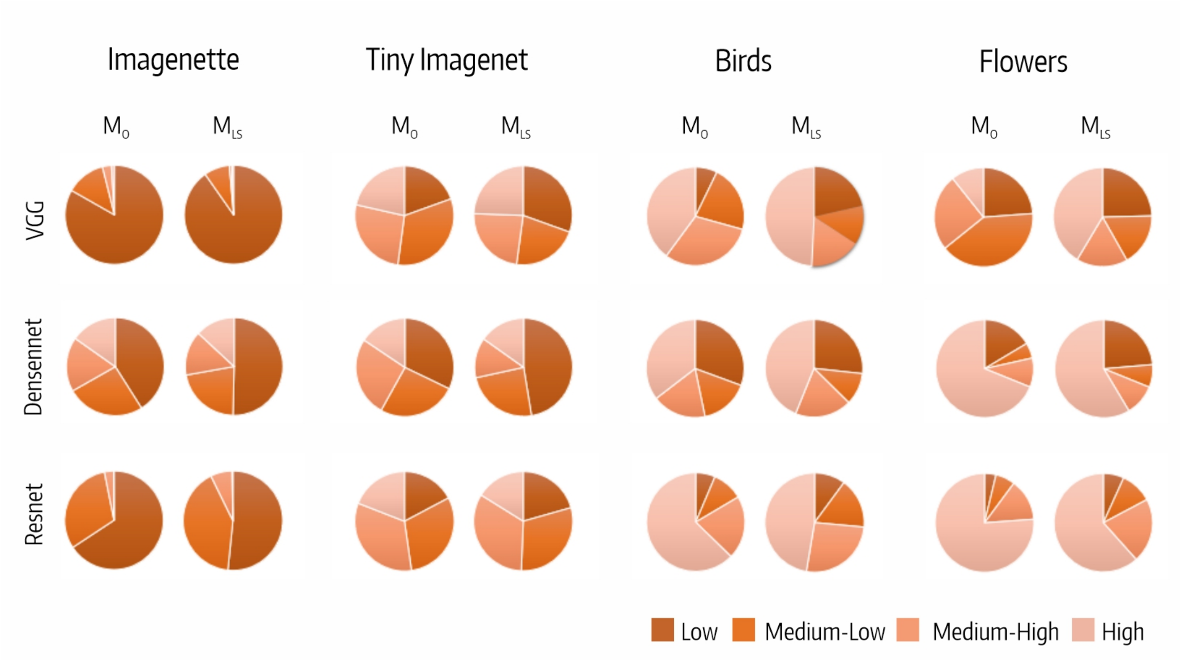
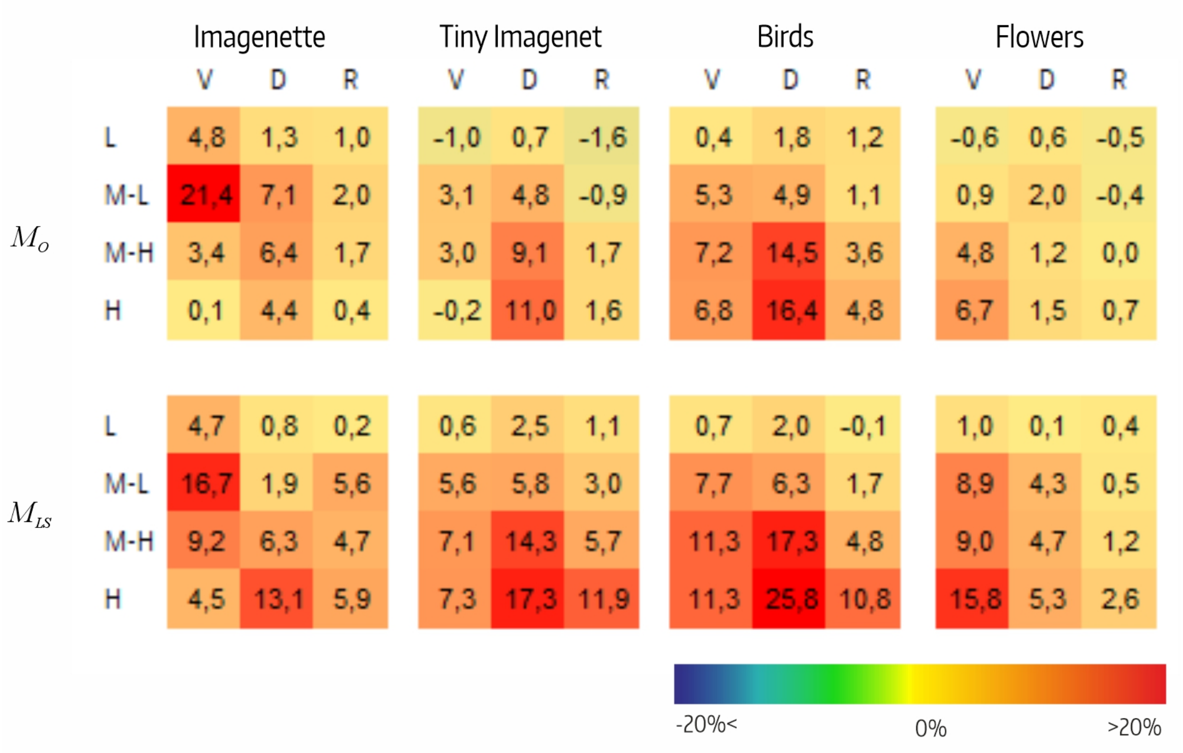


Figure 5. Distribution of fifth block output filters per CSI interval.

Figure 6 shows the reduction in accuracy resulting from filter ablation in each of the four CSI intervals. Analyzing, we highlight the following:

doi:10.20944/preprints202307.1957.v1

- In M_O, the decrease in performance tended to be greater at high CSI intervals. In VGG_O and $Densenet_O$, it varied between -1% and 21%, whereas in $Resnet_O$, it did not exceeded 2%. Of note were the M-L interval in VGG_O for Imagenette (21%) and the M-H and H intervals in $Densenet_O$ for Tiny Imagenet (14% and 16%, respectively). There were three small negative values.
- In M_{LS}, the performance decrease also tended to be greater at higher CSI intervals. The decrease varied between 1% and 17% in VGG_{LS}, between 0% and 26% in $Densenet_{LS}$, and between 0% and 12% in $Resnet_{LS}$. Negative values were not observed.



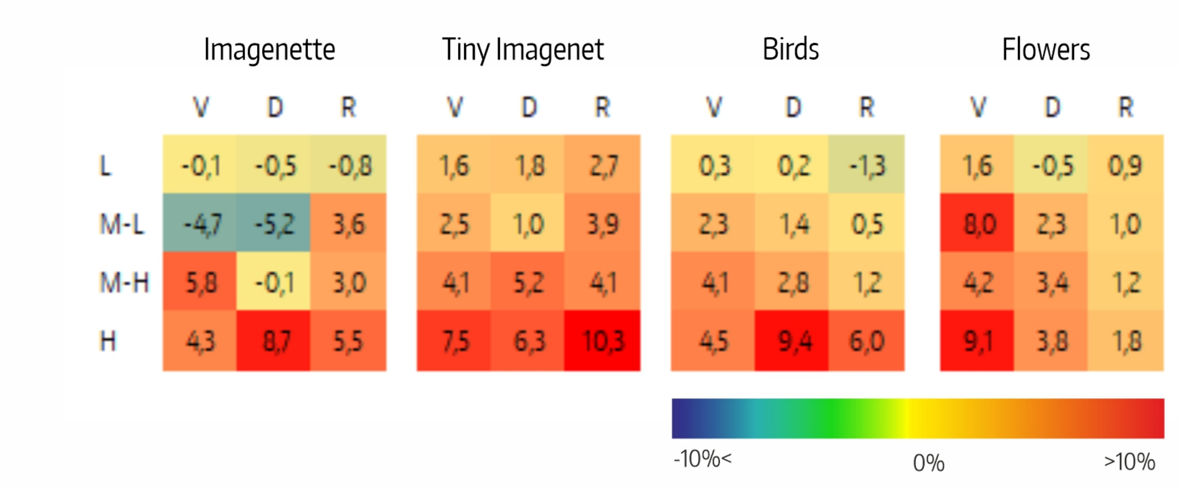
		Imagenette			Tiny Imagenet			Birds			Flowers		
		V	D	R	V	D	R	V	D	R	V	D	R
M_o	L	4,8	1,3	1,0	-1,0	0,7	-1,6	0,4	1,8	1,2	-0,6	0,6	-0,5
	M-L	21,4	7,1	2,0	3,1	4,8	-0,9	5,3	4,9	1,1	0,9	2,0	-0,4
	M-H	3,4	6,4	1,7	3,0	9,1	1,7	7,2	14,5	3,6	4,8	1,2	0,0
	H	0,1	4,4	0,4	-0,2	11,0	1,6	6,8	16,4	4,8	6,7	1,5	0,7
M_{LS}	L	4,7	0,8	0,2	0,6	2,5	1,1	0,7	2,0	-0,1	1,0	0,1	0,4
	M-L	16,7	1,9	5,6	5,6	5,8	3,0	7,7	6,3	1,7	8,9	4,3	0,5
	M-H	9,2	6,3	4,7	7,1	14,3	5,7	11,3	17,3	4,8	9,0	4,7	1,2
	H	4,5	13,1	5,9	7,3	17,3	11,9	11,3	25,8	10,8	15,8	5,3	2,6

Figure 6. Effect of ablation in $M_{O,csi}$ and $M_{LS,csi}$ variants per CSI interval ($DA_{M_X,csi}$) for VGG (V), Densenet (D), and Resnet (R) architectures on the four datasets.

The difference between the effects on the accuracy of filter ablation per CSI interval, $DM_{M,csi}$, is shown in Figure 7. The H intervals in VGG for Flowers (9%), in $Densenet$ for Imagenette and Birds (9%), and in $Resnet$ for Tiny Imagenet (10%) stand out. Also noteworthy are the negative values in the M-L intervals of VGG and $Densenet$ for Imagenette (-5%). The positive trend of $DM_{M,csi}$ as CSI increases is clearly shown in Figure 8, but the correlation analysis between the accuracy variation and the number of filters per CSI shows that the correlation is low in all cases (Table 6).

Table 6. Pearson correlation between the percentage of filters per CSI interval and the accuracy variation by ablation of filters.

	M_O	M_{LS}
VGG.	0.009	0.0582
Densenet	-0.0933	-0.0940
Resnet	0.3465	0.1276



	Imagenette V	Imagenette D	Imagenette R	Tiny Imagenet V	Tiny Imagenet D	Tiny Imagenet R	Birds V	Birds D	Birds R	Flowers V	Flowers D	Flowers R
L	-0,1	-0,5	-0,8	1,6	1,8	2,7	0,3	0,2	-1,3	1,6	-0,5	0,9
M-L	-4,7	-5,2	3,6	2,5	1,0	3,9	2,3	1,4	0,5	8,0	2,3	1,0
M-H	5,8	-0,1	3,0	4,1	5,2	4,1	4,1	2,8	1,2	4,2	3,4	1,2
H	4,3	8,7	5,5	7,5	6,3	10,3	4,5	9,4	6,0	9,1	3,8	1,8

Figure 7. Difference between the effects of ablation in the $M_{O,csi}$ and $M_{LS,csi}$ variants per CSI interval ($DM_{M,csi}$) for VGG (V), Densenet (D), and Resnet (R) architectures on the four datasets.

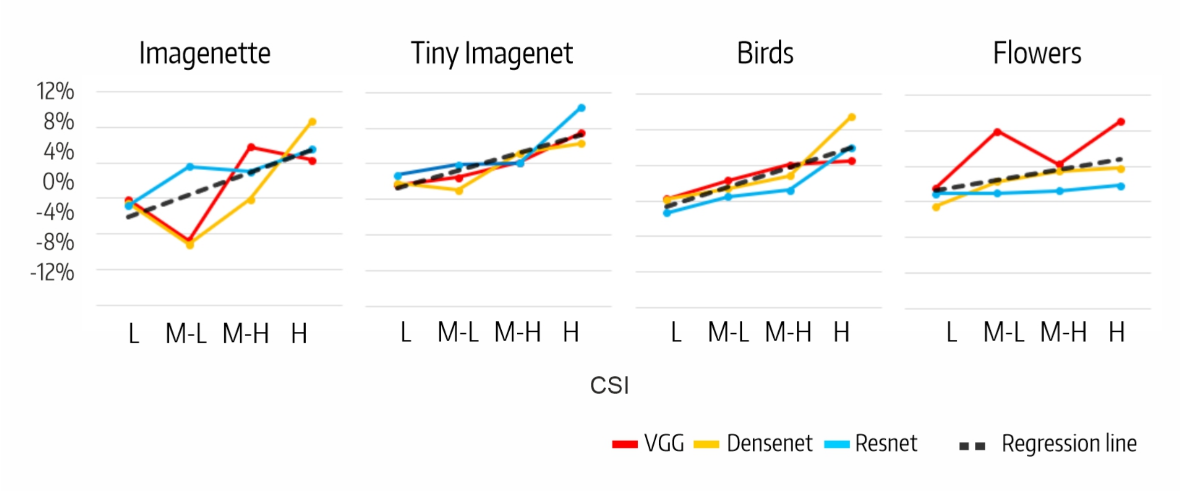


Figure 8. Relationship between $DM_{M,csi}$ and CSI interval.

In summary, the models increased the filters in the L (or even M-L) interval; however, the highest contribution to accuracy occurred in the H interval, supporting our hypothesis that LSC increases color selectivity.

5.3. Color hue analysis

Table 7 shows the Pearson correlation, for each dataset, between the hue distribution of pixels in the dataset and the number of filters in the model trained with the same dataset. It is above 0.60 in three out of four datasets (Imagenette, Tiny Imagenette, and Birds); however, it is low in Flowers (between 0.17 and 0.32). In this case, the yellow-green range is the most prevalent in the distribution of pixels in the dataset, whereas the orange-yellow range contains more filters. This difference can be explained by the fact that the former is more prevalent in the background and the latter in the figure.

Figures 9–11 show the most activated feature maps for each CSI interval for the silhouette and texture datasets. We highlight the following results:

- There were no significant differences between the maps of the silhouette and texture datasets.
- The models had maps with more defined silhouettes in the H interval.
- The difference between models was minimal in the L, M-L, and M-H intervals, except for VGG_{LS} and $Resnet_{LS}$, which had maps with active areas in all hue ranges.

- In the H interval, all M_{LS} variants had feature maps in more hue ranges than their M_O counterparts, especially in the red-magenta-blue range; in particular, VGG_{LS} achieved higher activation.
- In addition, in the H interval, VGG_{LS} had maps in all hue ranges of the dataset, whereas VGG_O only had less active maps in the red and orange ranges. $Densenet_O$ and $Densenet_{LS}$ had maps in the most frequent ranges of the Birds dataset but with incomplete silhouettes except on the orange hue, where $Densenet_{LS}$ obtained a complete figure. $Resnet_O$ has maps in all dataset hue ranges, like VGG_{LS}, and $Resnet_{LS}$, except for the magenta to red range.

Table 7. Pearson correlation, for each dataset, between the hue distribution of pixels in the dataset and the number of filters in the model trained with the same dataset.

	VGG16		**Densenet**		**Resnet**	
	M_O	M_{LS}	M_O	M_{LS}	M_O	M_{LS}
Imagenette	0.67	0.64	0.85	0.84	0.61	0.60
Tiny Imagenet	0.88	0.87	0.79	0.77	0.75	0.83
Birds	0.75	0.79	0.63	0.71	0.75	0.77
Flowers	0.17	0.32	0.28	0.28	0.20	0.22

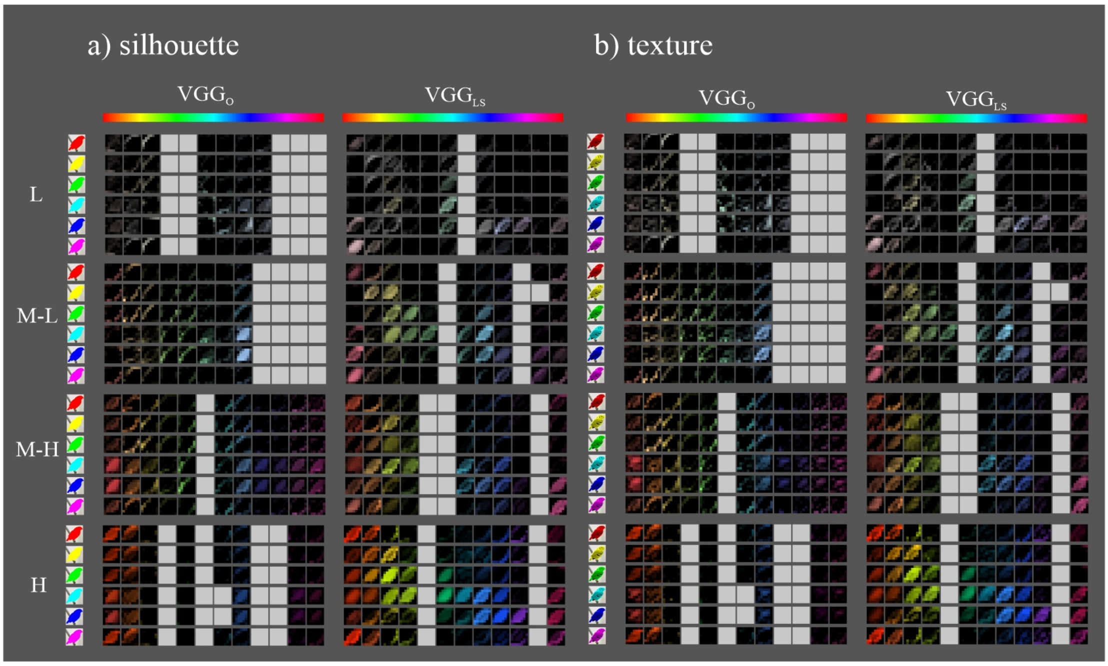


Figure 9. Fifth block filters response in VGG variants to images with objects of interest of different hues: a) only silhouette; b) with texture details.

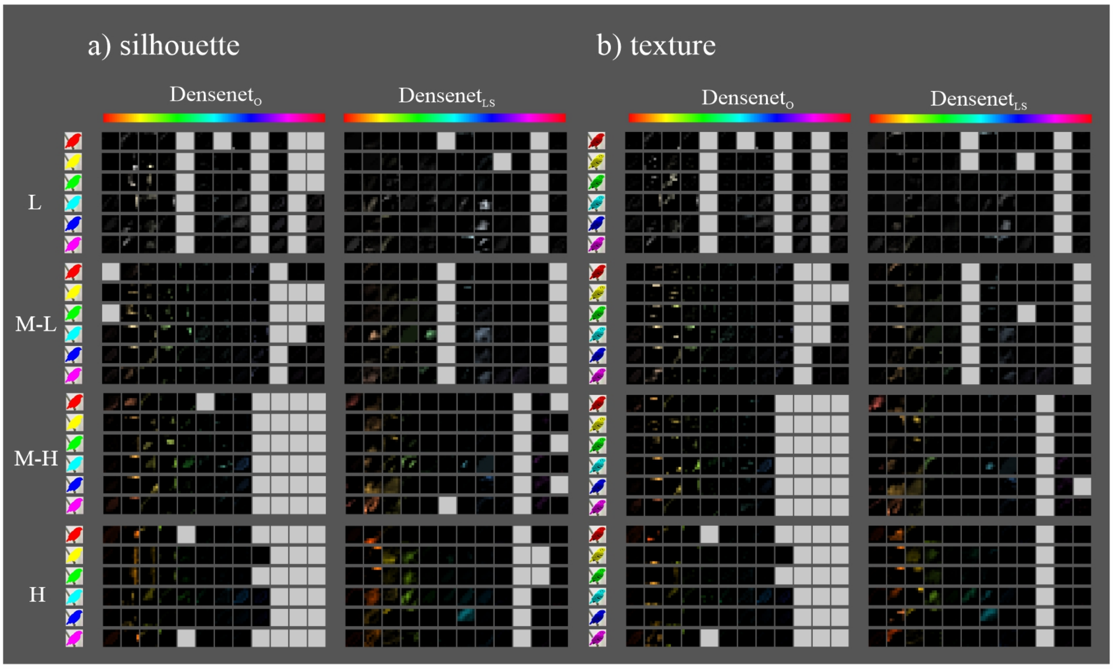


Figure 10. Fifth block filters response in *Densenet* variants to images with objects of interest of different hues: a) only silhouette; b) with texture details.

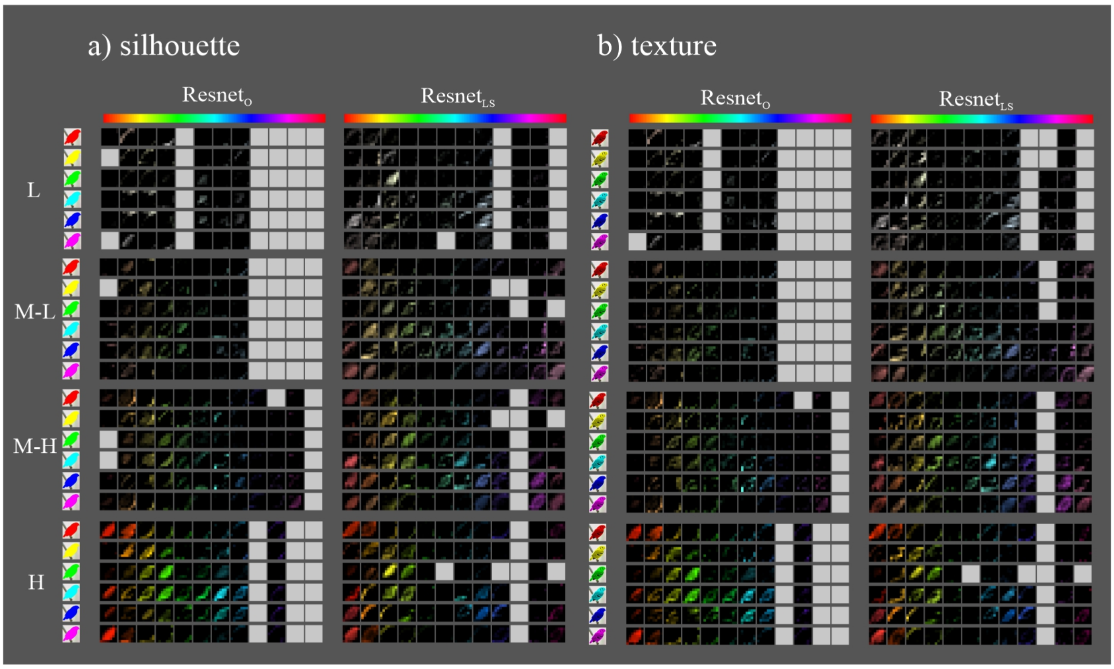


Figure 11. Fifth block filters response in *Resnet* variants to images with objects of interest of different hues: a) only silhouette; b) with texture details.

In summary, the models with LSC improved the response of the filters to the hue ranges present in the dataset, for both silhouettes and textures. The difference was higher in the architecture without skip connections. On the other hand, the ability of models to detect less present hue ranges varies. VGG_{LS} is the model with the strongest ability to detect both strongly and weakly present nuances. $Resnet_O$ and $Resnet_{LS}$ also detected them, but at a lower activation level.

5.4. Qualitative analysis of filter response

Figure 12 shows the color representation of the feature maps of the fifth block output for four randomly selected images, one from each dataset. In the visual analysis, we highlight the following points:

- The image from Imagenette (Figure 12a) has a strong contrast yellow-cyan between the figure and the background. The hue histogram shows that the cyan range is the most present in the image. Both variants of VGG get maps detecting the figure in yellow and the background in cyan; however, VGG_{LS} has maps of the orange areas in addition to the yellow parachute areas. $Densenet_O$ only has maps with very specific edges of the parachute or the silhouette in the low CSI intervals, whereas $Densenet_{LS}$ detects the silhouette of the figure in the M-H interval of the red range, although the activation area is larger than the corresponding figure in the image. $Resnet_O$ detects the background but not the figure, and $Resnet_{LS}$ detects both the figure and the background.
- The image from Tiny Imagenet (Figure 12b) shows less contrast between the background and the figure. The hue histogram shows that the yellow-green range is the most present. VGG_O detects textures of the background in orange and yellow hues, while VGG_{LS} detects textures of both the figure and background areas in green ranges as well. $Densenet_O$ and $Densenet_{LS}$ only have maps with very small areas of the figure in the predominant hues in the image: red, orange, yellow, or green; less in $Densenet_O$ than in $Densenet_{LS}$. $Resnet_O$ does not detect areas of the figure and $Resnet_{LS}$ does, even the silhouette in the orange or yellow ranges.
- The image from Birds (Figure 12c) shows a hue histogram where the yellow and cyan ranges are the highest, and green has a lower presence, representing areas in the background. VGG_O and VGG_{LS} detect the silhouettes of the figure in their yellow hues. In addition, VGG_{LS} detects the texture. $Densenet_O$ only detects reduced areas of the figure, whereas $Densenet_{LS}$ gets larger zones, particularly in the figure. $Resnet_O$ has maps detecting background textures and $Resnet_{LS}$ also detects the silhouette and background areas but with less definition than VGG_{LS}.
- Finally, the image from Flowers (Figure 12d) shows a hue histogram with the yellow-green range as the most relevant, followed by orange. VGG_{LS} detects the full silhouettes in the orange-yellow ranges and VGG_O either the edges or only one of the flowers. $Densenet_O$, $Densenet_{LS}$, and $Resnet_O$ detect reduced areas of the figure in the orange-yellow ranges. $Resnet_{LS}$, like VGG_{LS}, detects the full silhouettes but with less detail.

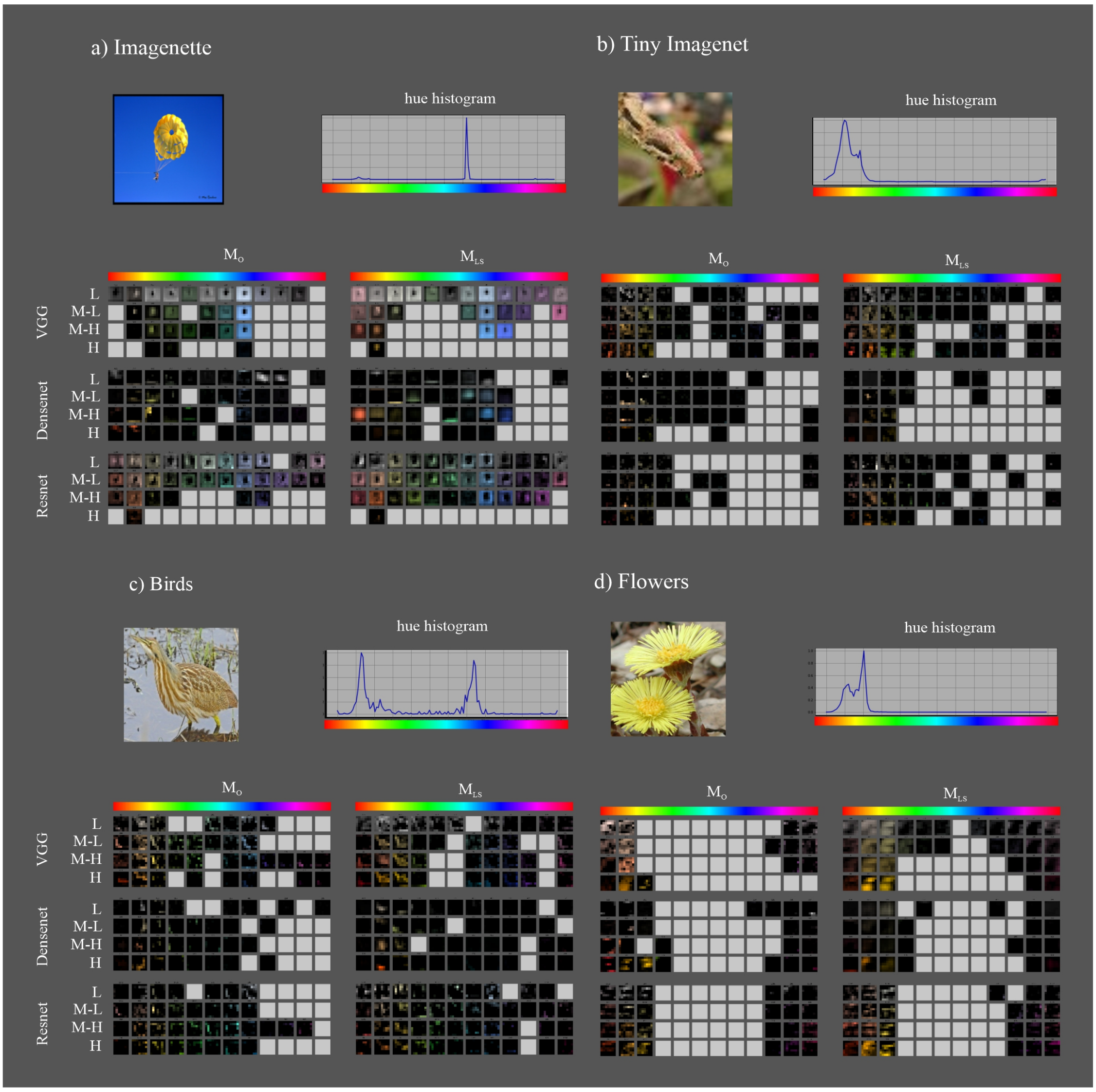


Figure 12. Randomly selected images in each dataset as an example for the feature map visualization according to the proposed methodology.

In cases with a higher contrast between the figure and background, such as in the Imagenette parachute (cyan range in the background and orange range in the figure), the improvements from LSC are smaller. However, in cases with less contrast (such as the Tiny Imagenet case or the Flowers case), LSC detects more features in the figure. This effect is higher for VGG and ResNet than for Densenet. This shows that LSC improves color selectivity in complex patterns, where contrast is more difficult to achieve, and facilitates feature extraction.

6. Conclusion and future work

In this paper, we focused on enhancing the color selectivity of CNN architectures. Bioinspired by the direct connection between the LGN and area V4, we presented an LSC architecture that connects the first and last blocks of the feature extraction stage, which allows the incorporation of low-level features, detected near the RGB input, to the definition of the more abstract features used in the classification stage.

It has been demonstrated that our proposal improves the performance of the original CNN architectures by enhancing color selectivity, regardless of whether they have skip connections. On the one hand, this improvement correlates with CSI (the higher the CSI interval, the higher the improvement). On the other, the size of the improvement is more prominent in datasets where color is a more significant feature but the particular CSI redistribution of filters depends exclusively on the characteristics of the training dataset.

However, not all improvements were due to color processing. Ablation studies show that, although high CSI filters produce the most significant improvements, there are often improvements due to low CSI filters. Therefore, although we focused on the analysis of color selectivity in this study, the LSC connection probably also improves the treatment of low saturated colors and achromatic textures. This point should be studied experimentally using more controlled datasets.

Hue analysis was facilitated by the proposed qualitative methodology, which allows us to group and sort the feature maps. Although the hue distribution of filters correlates with that of pixels in the dataset, the LSC connection enables the setting of selective filters for color hues that are poorly represented in the dataset.

To conclude, in this study, we analyzed how the filters of the last block of the feature extraction stage improved their color selectivity when using LSC. In future studies, it would be interesting to establish the relationship between color selectivity in the feature extraction and classification stages, especially by analyzing the role of color selectivity in per-class accuracy. On the other hand, the study of the ablation of filters by CSI and hue could be used to improve performance and reduce computational costs.

References

1. Taylor, J.; XuI, Y. Joint representation of color and form in convolutional neural networks: A stimulus-rich network perspective. *PLoS One* **2021**, *16*, :e0253442. doi:10.1371/journal.pone.0253442.
2. Engilberge, M.; Collins, E.; Süsstrunk, S. Color representation in deep neural networks. 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017, pp. 2786–2790.
3. Rafegas, I.; Vanrell, M. Color encoding in biologically-inspired convolutional neural networks. *Vision research* **2018**, *151*, 7–17.
4. Flachot, A.; Gegenfurtner, K.R. Color for object recognition: Hue and chroma sensitivity in the deep features of convolutional neural networks. *Vision Research* **2021**, *182*, 89–100. doi:10.1016/j.visres.2020.09.010.
5. Lindsay, G.W. Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future. *J. Cogn Neuroscience* **2021**, *33*, 2017–2031. doi:10.1162/jocn_a_01544.
6. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics* **1980**, *36*, 193–202.
7. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*, 2278–2324.
8. N., K. Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing. *Annu Rev Vis Sci.* **2015**, *1*, 417–446. doi:10.1146/annurev-vision-082114-035447.
9. Hubel, D. *Eye, Brain and Vision. Scientific American Libarary*; Scientific American Libarary. New York, USA, 1988.
10. Teller, D.; Palmer, J. Vision and the Visual System, 2016.
11. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Highway networks. *arXiv preprint arXiv:1505.00387* **2015**.
12. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
13. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks, 2018. arXiv:1608.06993 [cs].
14. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer Assisted Intervention, MICCAI 2015; Springer International Publishing: Cham, 2015; pp. 234–241.
15. Ferrera, V.; Nealey, T.; Maunsell, J. Responses in macaque visual area V4 following inactivation of the parvocellular and magnocellular LGN pathways. *Journal of Neuroscience* **1994**, *14*, 2080–2088.
16. Conway, B.R. Color vision, cones, and color-coding in the cortex. *The neuroscientist* **2009**, *15*, 274–290.

17. Roe, A.W.; Chelazzi, L.; Connor, C.E.; Conway, B.R.; Fujita, I.; Gallant, J.L.; Lu, H.; Vanduffel, W. Toward a unified theory of visual area V4. *Neuron* **2012**, *74*, 12–29.
18. Arrigo, A.; Calamuneri, A.; Mormina, E.; Gaeta, M.; Quartarone, A.; Marino, S.; Anastasi, G.P.; Aragona, P. New insights in the optic radiations connectivity in the human brain. *Investigative Ophthalmology & Visual Science* **2016**, *57*, 1–5.
19. Shapley, R.; Hawken, M.J. Color in the cortex: single-and double-opponent cells. *Vision research* **2011**, *51*, 701–717.
20. Rafegas, I.; Vanrell, M.; Alexandre, L.A.; Arias, G. Understanding trained CNNs by indexing neuron selectivity. *Pattern Recognition Letters* **2020**, *136*, 318–325.
21. Harris, E.; Mihai, D.; Hare, J. Spatial and colour opponency in anatomically constrained deep networks. *arXiv preprint arXiv:1910.11086* **2019**.
22. Harris, E.; Mihai, D.; Hare, J. How Convolutional Neural Network Architecture Biases Learned Opponency and Color Tuning. *Neural Computation* **2021**, *33*, 858–898.
23. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, 2016. arXiv:1602.07261 [cs].
24. Chollet, F. Xception: Deep learning with depthwise separable convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.
25. Howard, J. imagenette, 2019.
26. Le, Y.; Yang, X. Tiny imagenet visual recognition challenge. *CS 231N* **2015**, *7*, 3.
27. Gerry. Birds 315; species image classification dataset, 2021.
28. Nilsback, M.E.; Zisserman, A. Automated Flower Classification over a Large Number of Classes. Indian Conference on Computer Vision, Graphics and Image Processing, 2008.
29. Brigato, L.; Barz, B.; Iocchi, L.; Denzler, J. Image classification with small datasets: overview and benchmark. *IEEE Access* **2022**, *10*, 49233–49250.
30. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A.C.; Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* **2015**, *115*, 211–252. doi:10.1007/s11263-015-0816-y.
31. Meyes, R.; Lu, M.; de Puiseau, C.W.; Meisen, T. Ablation studies in artificial neural networks. *arXiv preprint arXiv:1901.08644* **2019**.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.