Contextual Convolutional Neural Networks

Ionut Cosmin Duta*, Mariana Iuliana Georgescu, Radu Tudor Ionescu
University of Bucharest, Romania; SecurifAI, Romania
*icduta@gmail.com

Abstract

We propose contextual convolution (CoConv) for visual recognition. CoConv is a direct replacement of the standard convolution, which is the core component of convolutional neural networks. CoConv is implicitly equipped with the capability of incorporating contextual information while maintaining a similar number of parameters and computational cost compared to the standard convolution. CoConv is inspired by neuroscience studies indicating that (i) neurons, even from the primary visual cortex (V1 area), are involved in detection of contextual cues and that (ii) the activity of a visual neuron can be influenced by the stimuli placed entirely outside of its theoretical receptive field. On the one hand, we integrate CoConv in the widely-used residual networks and show improved recognition performance over baselines on the core tasks and benchmarks for visual recognition, namely image classification on the ImageNet data set and object detection on the MS COCO data set. On the other hand, we introduce CoConv in the generator of a state-of-the-art Generative Adversarial Network, showing improved generative results on CIFAR-10 and CelebA. Our code is available at https://github.com/iduta/coconv.

1. Introduction

Contextual information is vital for a visual perception system. A point (or a small patch) in a scene (image) is mostly meaningless without the surrounding contextual information. For instance, it is very difficult for a person to provide a semantic label or a description for a small patch (taken from an image) without providing a broader visual context. As shown in the example illustrated in Figure 1, it is even hard to label an entire object without context, let alone some part of the respective object. In neuroscience research, the critical role of the contextual influences on visual perception systems is well proven since long time ago [1, 4, 38]. For example, Zipser et al. [38] studied the contextual modulation in the primary visual cortex of awake, behaving macaque monkeys. The work shows that the activity of a visual neuron is influenced by the stimuli placed entirely outside of its default receptive field. Furthermore, the work demonstrates that the influence of the context on the activity of a visual neuron is present even at the early stages of the visual system (V1 area). Albright et al. [1] stated that for each local region of an image, the extraction of semantic meaning is only possible if information from other regions is taken into account. This clearly highlights the importance of contextual information in the natural visual systems studied in the field of neuroscience.

(a) (b)

Figure 1: An example in which the context is crucial in labeling the object (kitchen glove), which can easily be mistaken for something else if the rest of the image is not seen. (a) A picture of a kitchen glove. (b) A picture of the same glove with context.

Convolutional neural networks (CNNs) [18, 19] represent the backbone of nearly every current computer vision task and application [3, 7, 8, 9, 11, 12, 13, 17, 21, 20, 30, 31, 34, 35, 39]. Although the neuroscience studies mentioned above [1, 4, 38] clearly demonstrate the presence and the importance of contextual influence in a visual neuron of a biological visual cortex, in the current artificially-built visual systems, the core building block of CNNs, represented by the convolutional layer (with spatial filters that activate on local patterns), is not implicitly equipped with the ability of integrating contextual information. In general, the convolutional filters have a limited receptive field, usually corresponding to a 3×3 spatial kernel size, due to the fact that increasing the kernel size brings additional costs in terms of parameters and computational resources. There are many approaches that address the integration of contextual information, e.g. [32], however, most of them follow the direction of integrating additional building blocks in the CNN to incorporate contextual information. However, this line of research results in additional costs for the CNN in terms

of both parameters and computation, which is not in line with the neuroscience findings, pointing out that the visual system is extremely efficient and that the integration of the contextual information is an implicit capability of the visual neuron. Inspired by the aforementioned neuroscience studies and addressing the above limitations, this work makes the following contributions:

- We propose contextual convolution (CoConv), a direct replacement of the standard convolution that can be used at any stage in CNN architectures. CoConv is implicitly equipped with the ability of accessing contextual information at multiple levels without increasing the demands in terms of parameters and computational cost, compared to the standard convolution (see Section 3).
- We integrate CoConv in convolutional and generative networks of various depths, presenting novel architectures based on CoConv for visual recognition and generation (see Section 4).
- We show improved detection, recognition and generation performance obtained by CoConv over the standard convolution and competing methods on the core tasks and benchmarks for visual recognition and generation (see Section 5).

We underline that our approach, CoConv, is both effective and efficient. We believe that its simplicity coupled with its effectiveness generates a great potential to become widely-adopted.

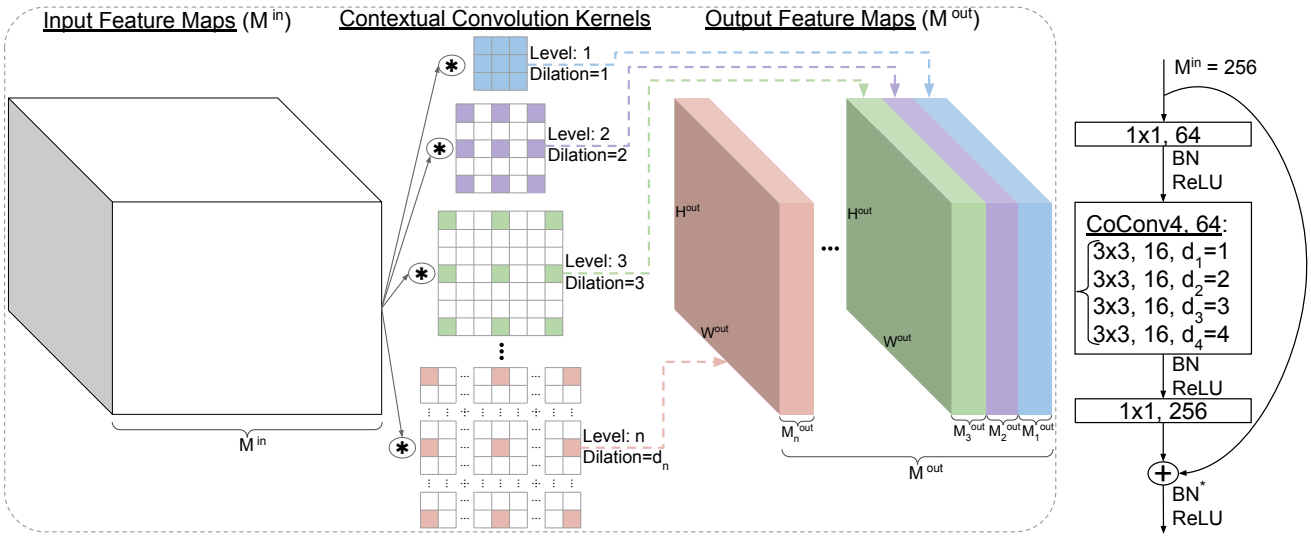
2. Related Work

There are numerous works with the goal of integrating contextual information in an artificial visual system. The work of Wang et al. [32] introduced a non-local block to capture contextual information in a CNN. In [11], the authors proposed a squeeze-and-excitation block to capture global information and scale each feature map accordingly. However, these works propose additional building blocks that need to be inserted in the CNN, therefore bringing significant supplementary parameters and computational costs that can negatively impact the efficiency of the visual system. In contrast, we propose to implicitly integrate the process of capturing contextual information in the core component (the convolutional layer), without increasing the number of parameters and computational costs. Furthermore, our approach can be complementary to these works, as they still need to use convolutional layers, in their overall CNN architectures. Dilated convolution [2, 36, 37] is an approach to enlarge the receptive field of the convolution kernel. Our work makes use of dilated convolution, however, there are significant differences in the approach and usage from previous works. For example, Chen et al. [2] proposed atrous spatial pyramid pooling (ASPP) to segment objects at multiple scales. There are fundamental differences that distinguish our work from that of Chen et al. [2], as explained next. First, ASPP is proposed just as a head module for image segmentation, while our approach is designed as a direct replacement of the convolution along all stages of the CNN architecture, irrespective of the visual recognition task. Second, different from ASPP, our approach is specifically designed to integrate contextual information at different levels while maintaining the same number of parameters and computational costs as the standard convolution. Importantly, as in the neuroscience findings [1, 38] showing that contextual influence is present and relevant, even in the primary visual cortex (V1 area), we integrate the contextual information within all network layers, including the early convolutional layers as well. Contrary to the neuroscience findings, Chen et al. [2] employed contextual modeling only at the end of the CNN, just as an additional head before the final classification layer for semantic segmentation. A more closely related work to our own is [37], which proposes dilated residual networks by integrating dilated convolution just towards the end of the network. Thus, this work is also not in line with the neuroscience conclusion regarding the importance of the contextual modeling at the early stages of the visual cortex. Furthermore, our approach is different from that of Yu et al. [37], as we employ different levels of contextual information, being able to capture information about local details and various levels of contextual information in the same time. Importantly, integrating the default approach of Yu et al. [37] into residual networks for image recognition drastically increases the demands in computational resources. As shown in the experiments, our approach delivers improved recognition performance without increasing the computational costs.

Another contribution that is aimed at capturing context in all stages of neural architectures is represented by capsule networks (CapsNets) [28]. Although CapsNets are also backed by neurosciene studies and showed promising results on small data sets such as MNIST and CIFAR-10, their low accuracy gains come with a large computational cost. Furthermore, since their introduction by Sabour et al. [28], CapsNets have failed to show their effectiveness on very deep neural networks and on large image recognition benchmarks, mostly due to their extremely large computational costs. Different from [28], we present empirical evidence showing that CoConv improves the accuracy of very deep models, e.g. ResNet-152, on very large benchmarks, e.g. ImageNet, at no additional cost. Another plus is that our contribution is fairly easy to implement, having the right ingredients (simple, effective, no additional computational cost) to be widely adopted by the community.

3. Contextual Convolution

The standard convolution in state-of-the-art CNN architectures uses a single type of kernel with a fixed receptive field, usually corresponding to a kernel size of 3×3, since



(a) (b)

Figure 2: (a) Contextual Convolution (CoConv). Instead of using standard or dilated convolution, we propose to integrate multiple levels of kernels with different dilation ratios in the convolutional layer. At each level, we have multiple kernels. We emphasize the fact that, in this illustration, $d_1 = 1$, $d_2 = 2$ and $d_3 = 3$ is a coincidence that facilitates visualization, yet, in general, we do not constrain d_i to be equal to i. Best viewed in color. (b) An example of CoConv residual building block.

increasing the kernel size brings additional costs in terms of the number of learnable parameters and computational time, respectively. The number of learnable parameters (weights) and FLOPs (floating point operations) for the standard convolution can be computed as:

$$\begin{aligned} \text{params} &= M^{in} \cdot K^w \cdot K^h \cdot M^{out}, \\ \text{FLOPs} &= M^{in} \cdot K^h \cdot K^w \cdot M^{out} \cdot W^{out} \cdot H^{out}, \end{aligned} \tag{1}$$

where, M^{in} and M^{out} represent the number of input and output feature maps, K^w and K^h are the width and height of the convolution kernel, and finally, W^{out} and H^{out} are the width and height of the output feature maps. For the sake of simplicity, we ignored the bias terms and hyperparameters such as stride and padding in Equation (1).

Contextual convolution (CoConv), illustrated in Figure 2a, receives a number of input feature maps M^{in}, over which we apply different levels $L = \{1, 2, 3, ..., n\}$ of convolution kernels with varying dilation ratios $D = \{d_1, d_2, d_3, ..., d_n\}$. In other words, the kernels at level i have the dilation ratio $d_i, \forall i \in L$. By gradually increasing the dilation ratio (basically introducing increasingly larger "holes" into the kernels), the filters can have access to increasingly broader contextual information. As we increase the dilation ratio, the kernels become sparser, thus, being applied over the input feature maps in a sparse pattern, skipping elements in the computation. As depicted in Figure 2a, only the colored spatial locations of the kernels are involved in the computation of the output feature maps. Thus, each level of dilated kernels maintains a similar number of parameters and FLOPs, while increasing the spatial receptive field to integrate more contextual information. The kernels with lower dilation ratios are responsible for capturing information about local details from the input feature maps, while the kernels with higher dilation ratios are empowered with the ability of incorporating contextual information for helping the recognition process. At each level i, the kernels provide a number of output feature maps M_i^{out}, for all $i \in L$, each map having the width W^{out} and the height H^{out}. Hence, the total number of learnable parameters and FLOPs of CoConv is computed as follows:

$$\begin{aligned} \text{params} = & M^{in} \cdot (K^w \cdot K^h)^{(d_1)} \cdot M_1^{out} + \\ & ... + M^{in} \cdot (K^w \cdot K^h)^{(d_n)} \cdot M_n^{out}, \\ \text{FLOPs} = & M^{in} \cdot (K^w \cdot K^h)^{(d_1)} \cdot M_1^{out} \cdot W^{out} \cdot H^{out} + \\ & ... + M^{in} \cdot (K^w \cdot K^h)^{(d_n)} \cdot M_n^{out} \cdot W^{out} \cdot H^{out}, \end{aligned} \tag{2}$$

where, $(K^w \cdot K^h)^{(d_i)}$ refers to the spatial size of the kernel of width w and height h (basically, how many spatial locations of the kernel are involved in the computation of an output feature map), d_i points to the dilation ratio used for the kernels at level i, and:

$$M^{out} = \sum_{i=1}^{n} M_i^{out}, \forall i \in L. \tag{3}$$

Although we use multiple levels of kernels with different dilation ratios, the total number of kernels in CoConv is equal to the total number of kernels in the standard convolution, as it results from Equation (3). We emphasize that all levels and kernels in CoConv are independent, allowing parallel execution, just as in a standard convolutional

layer. CoConv is a direct replacement of the standard convolution with the capability of integrating contextual information. Moreover, as formally presented in Equations (2), the number of learnable parameters and FLOPs involved in CoConv is equal to those involved in the standard convolution from Equation (1). We emphasize that the number of dilation levels of CoConv can be adjusted, for instance, based on the resolution of the feature maps. We present various practical examples in the next section.

4. Contextual Convolutional Neural Networks

We describe some neural architectures based on contextual convolution (CoConv). First, we integrate CoConv in the widely-used residual networks (ResNets) [8]. ResNets can be split into four main stages depending on the proximity of the layers with respect to the input and, implicitly, on the resolution of the feature maps, as shown in Table 1. Figure 2b shows an example of a CoConv residual building block used in the first stage of a network. The CoConv residual block uses a 1×1 convolution to reduce the number of feature maps to 64, followed by a CoConv with four levels to capture contextual information. Each CoConv level has a different dilation ratio. The number of output feature maps at each level is 16, regardless of the dilation rate. Then, a 1×1 convolution is used to restore the number of feature maps to 256. As in the standard residual blocks, batch normalization (BN) [13] and Rectified Linear Unit (ReLU) activations [25] follow each convolutional block.

Our network for image classification, termed contextual residual network (CoResNet), is formally presented in Table 1. Although we illustrate our updates on ResNet-50, thus obtaining CoResNet-50, the changes can be analogously operated on models of different depths. Since the size of the feature maps decreases as the layers are farther away from the input, we also adapt the number of levels in our CoConv layers with respect to the resolution of the feature maps. Hence, the first main stage of the network uses four levels in the CoConv layers, with different dilation ratios. Further, the second stage uses three levels in the CoConv layers, while the third stage uses two levels. As the spatial resolution of the feature maps in the last stage is reduced to 7×7, we consider that using multiple dilation ratios is no longer justified. Thus, the last stage employs just one level of CoConv. In the experimental section, we provide an ablation study on the number of levels in the CoConv layers. However, the number of levels can be tuned for each particular task or application, based, for instance, on the resolution of the feature maps along the network. Based on empirical evidence, we consider that our network has improved recognition capabilities compared to the standard ResNet, as the convolution is equipped with the ability of integrating contextual information at multiple levels and, as can be seen in Table 1, CoConv does not add any additional

stage	output	ResNet-50	CoResNet-50
	112×112	7×7, 64, $s=2$	7×7, 64, $s=2$
	56×56	3×3 maxpool $s=2$	3×3 maxpool $s=2$
1	56×56	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\\underline{\text{CoConv4, 64:}}\\\begin{bmatrix}3\times3, 16, d_1{=}1\\3\times3, 16, d_2{=}2\\3\times3, 16, d_3{=}3\\3\times3, 16, d_4{=}4\end{bmatrix}\\1\times1, 256\end{bmatrix}\times3$
2	28×28	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\\underline{\text{CoConv3, 128:}}\\\begin{bmatrix}3\times3, 64, d_1{=}1\\3\times3, 32, d_2{=}2\\3\times3, 32, d_3{=}3\end{bmatrix}\\1\times1, 512\end{bmatrix}\times4$
3	14×14	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\\underline{\text{CoConv2, 256:}}\\\begin{bmatrix}3\times3, 128, d_1{=}1\\3\times3, 128, d_2{=}2\end{bmatrix}\\1\times1, 1024\end{bmatrix}\times6$
4	7×7	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\\underline{\text{CoConv1, 512:}}\\\begin{bmatrix}3\times3, 512, d_1{=}1\end{bmatrix}\\1\times1, 2048\end{bmatrix}\times3$
	1×1	global avgpool 1000-d fc	global avgpool 1000-d fc
# params		25.56×10^6	25.56×10^6
FLOPs		4.14×10^9	4.14×10^9

Table 1: A side-by-side comparison of ResNet-50 and CoResNet-50. Although we illustrate our updates on ResNet-50, the changes can be analogously operated on models of different depths, e.g. ResNet-152.

weights nor it enlarges the computational cost compared to the original network.

We also show the benefits of integrating CoConv in a generative adversarial network (GAN) [5]. We specifically consider progressive GAN (ProGAN) [14], a model that generates high-resolution images starting with a low-resolution output (4×4 pixels) and gradually adding layers to the network to produce a high-resolution output (up to 1024×1024 pixels). We used the exact same architecture described in [14], only replacing the convolutional layers with CoConv layers. The ProGAN and CoProGAN gener-

output	ProGAN	CoProGAN
4×4	3×3, 512	3×3, 512
8×8	2×2 upsample	2×2 upsample
8×8	3×3, 512	$\begin{bmatrix} 3\times3, 256, d_1{=}1 \\ 3\times3, 256, d_2{=}2 \end{bmatrix}$
16×16	2×2 upsample	2×2 upsample
16×16	3×3, 512	$\begin{bmatrix} 3\times3, 172, d_1{=}1 \\ 3\times3, 170, d_2{=}2 \\ 3\times3, 170, d_3{=}3 \end{bmatrix}$
32×32	2×2 upsample	2×2 upsample
32×32	3×3, 512	$\begin{bmatrix} 3\times3, 128, d_1{=}1 \\ 3\times3, 128, d_2{=}2 \\ 3\times3, 128, d_3{=}3 \\ 3\times3, 128, d_4{=}4 \end{bmatrix}$
64×64	2×2 upsample	2×2 upsample
64×64	3×3, 256	$\begin{bmatrix} 3\times3, 64, d_1{=}1 \\ 3\times3, 64, d_2{=}2 \\ 3\times3, 64, d_3{=}3 \\ 3\times3, 64, d_4{=}4 \end{bmatrix}$
128×128	2×2 upsample	2×2 upsample
128×128	3×3, 128	$\begin{bmatrix} 3\times3, 32, d_1{=}1 \\ 3\times3, 32, d_2{=}2 \\ 3\times3, 32, d_3{=}3 \\ 3\times3, 32, d_4{=}4 \end{bmatrix}$
128×128	1×1, 3	1×1, 3
# params	27.21×10^6	27.21×10^6
FLOPs	54.76×10^9	54.76×10^9

Table 2: A side-by-side comparison of ProGAN and CoProGAN.

ator for the CelebA data set [24] is illustrated in Table 2. The CoProGAN generator designed for the final output of 128×128 pixels starts with a feature map size of 4×4 pixels. Hence, at the first layer, we only use a dilation rate of 1. When the output size increases to 8×8 pixels, we add two dilation rates of 1 and 2. Similarly, we add three dilation rates (1, 2 and 3), when the output size is 16×16 pixels. When the output size is 32×32, 64×64 or 128×128 pixels, we use four dilation rates of 1, 2, 3 and 4, regardless of the size of the output. The number of filters in each CoConv layer matches the number of filters in the corresponding conv layer from ProGAN. Thus, the number of parameters and FLOPs in ProGAN and CoProGAN are identical.

5. Experiments

5.1. Experimental setup

We perform object recognition experiments on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [27], which is one of the most popular benchmarks in visual recognition. The ImageNet data set consists of 1000 classes of objects, 1.28 million training images and 50K validation images. As common, we report both the top-1 and top-5 error rates. We follow the standard settings in [6, 8, 9] and employ the Stochastic Gradient Descent (SGD) optimizer with a standard momentum rate of 0.9 and a weight decay of 0.0001. We perform the training for 90 epochs, starting with a learning rate of 0.1, reducing it by $1/10$ at the 30-th, 60-th and 80-th epochs, similarly to [6, 8]. Each model is trained using 8 GPUs. We use the standard mini-batch size of 256 for training and data augmentation as in [6, 31], training and testing on 224×224 image crops.

For the object detection task, we consider the MS COCO data set [22], which contains 80 object categories. We use COCO 2017 train (118K images) for training and COCO 2017 val (5K images) for testing. We train each model for 130 epochs on 8 GPUs using mini-batches of 32 examples, resulting in 60K training iterations. The training is performed using the SGD optimizer with momentum 0.9, weight decay 0.0005, with the learning rate 0.02 (reduced by $1/10$ before the 86-th and 108-th epochs). We also use a linear warm-up in the first epoch, following [6]. For data augmentation, we perform random crop as in [23], color jittering and horizontal flip. We consider an input image size of 300×300 pixels. As evaluation metrics, we report the average precision (AP) and the AP@IoU=0.5.

We conduct image generation experiments on CIFAR-10 [16] and CelebA [24], considering only fully unsupervised (not class conditional) GAN models. The CIFAR-10 training set is composed of 50K images of 32×32 pixels, while the CelebA training set is formed of 202K images of 128×128 pixels. Following Karras et al. [14], the optimization is performed using the Adam [15] optimizer with $\beta_1 = 0$, $\beta_2 = 0.99$, $\epsilon = 10^{-8}$ and the learning rate set to 10^{-3}. We train each model until the discriminator sees 12 million real images in total. Each model is trained on a single GeForce GTX 3090 GPU. As evaluation measures, we report the Inception Score (IS) [29] and the Fréchet Inception Distance (FID) [10] for CIFAR-10, and the multiscale structural similarity index measure (MS-SSIM) [33] and the Sliced Wasserstein Distance (SWD) [26] for CelebA. To reproduce the results of ProGAN, we used the official TensorFlow implementation available at https://github.com/tkarras/progressive_growing_of_gans.

5.2. Ablation experiments on CoConv dilation levels

In Table 3, we present ablation experiments with different configurations generated by varying the number of dilation levels in the ConConv residual blocks corresponding to each stage of the network. The first column indicates the number of levels in the CoConv residual block used in each

CoConv levels	top-1	top-5	params	GFLOPs
(1,1,1,1)	23.88	7.06	25.56	4.14
(2,2,2,1)	23.24	6.79	25.56	4.14
(3,3,2,1)	23.23	6.66	25.56	4.14
(3,3,3,3)	23.33	6.71	25.56	4.14
(4,3,2,1)	22.73	6.49	25.56	4.14
top(4,3,2,1)	25.58	8.05	25.56	4.14

Table 3: Ablation experiments on ImageNet with various CoResNet-50 configurations, considering different numbers of dilation levels in the CoConv residual blocks. The configuration $(1, 1, 1, 1)$ corresponds to ResNet-50 [8]. Lower values are better.

Network	stride	top-1	top-5	params	GFLOPs
ResNet-50 [8]	32	23.88	7.06	25.56	4.14
NL ResNet-50 [32]	32	22.91	6.42	36.72	6.18
CoResNet-50 [ours]	32	22.73	6.49	25.56	4.14
DRN-50 [37]	8	22.44	6.47	25.56	19.20
CoResNet-50 [ours]	8	21.93	6.17	25.56	19.20

Table 4: Comparison of CoResNet with closely related works [32, 37] on ImageNet, considering neural models of 50 layers in all cases. Lower values are better.

of the four main stages of the network. The configuration $(1, 1, 1, 1)$ refers to the case where a single CoConv level with dilation $d_1 = 1$ is used in all four stages, thus being completely equivalent to the original ResNet [8], which is the baseline in our object recognition experiments.

Integrating our CoConv with two levels for the first three stages of the network, resulting in the configuration $(2, 2, 2, 1)$, significantly improves the top-1 error rate from 23.88% to 23.24%, while maintaining the same number of parameters and FLOPs. In general, adding more levels to our CoConv residual blocks further improves the results.

We obtain the best results with the configuration $(4, 3, 2, 1)$ for the number of levels in the CoConv blocks involved in the four main stages of the network. We hereby note that we performed experiments with even more levels of CoConv, but we did not notice further significant improvements in terms recognition performance. Hence, we find the configuration $(4, 3, 2, 1)$ optimal for an input size of 224×224 pixels. If the input size would be higher, then perhaps another configuration considering more dilation levels can provide even further improvements. All in all, the flexibility of adapting the CoConv levels for each network stage with respect to the input resolution is an important strong point of our approach.

To show the importance of having different levels of dilation in CoConv, we include the configuration top$(4, 3, 2, 1)$ in Table 3, which considers only the highest level of dilation in each stage of the network. For instance, in the first stage, only the convolution with dilation ratio 4 is used, the second stage uses only the convolution with dilation 3, and so on. For a fair comparison, we stress out that the number of filters in each CoConv layer is always equal to the number of filters in the original ResNet model. Regarding the configuration top$(4, 3, 2, 1)$, we can notice a significant drop in recognition performance. This is basically the opposite case of the baseline $(1, 1, 1, 1)$. We can observe from the results that both (extreme) cases have significantly lower recognition performance than CoConv with multiple levels. This is due to the fact that the baseline $(1, 1, 1, 1)$ is only able to capture information about local details (as it uses the lowest dilation), without being equipped with the ability to capture contextual information. On the other side, the case top$(4, 3, 2, 1)$ is only able to capture contextual information, lacking the ability of capturing information about local details. This set of experiments proves an important and strong point of our approach: CoConv captures information regarding both local details and global context, providing a more complete visual representation which improves the recognition performance.

5.3. Comparison of CoResNet to closely related works

In Table 4, we present the results of closely related methods [32, 37] by training all neural networks with the same standard settings for providing a direct and fair comparison. All methods are applied on top of the 50-layer deep residual network. First, we observe that our CoResNet-50 outperforms more complex architectures, such as non-local (NL) networks [32]. It is important to highlight the increase in the number of learnable parameters and computational costs brought by the introduction of the non-local block, while our CoResNet-50 maintains the same number of parameters and computational cost as the baseline ResNet-50 [8].

The dilated residual network (DRN) proposed by Yu et al. [37] requires to decrease the overall stride of the network from the default 32 to 8. Basically, DRN does not use downsampling of the feature maps for the last two stages of the network. In Table 4, we can observe that DRN has a significant impact in increasing the requirements in terms of computational resources, increasing the GFLOPs from 4.14 to 19.20. This significant increase in computational cost makes DRN not feasible for standard image classification as, for instance, increasing the depth of the baseline ResNet [8] from 50 to 101 layers provides a top-1 error improvement from 23.88% to 22.00%, while the requirements in GFLOPs increase from 4.14 to only 7.88. In the same time, DRN-50 improves the baseline top-1 error from 23.88% to 22.44% at a higher computational cost. In comparison, our CoResNet-50 can improve the recognition performance of the baseline without impacting the demands in terms of computational resources.

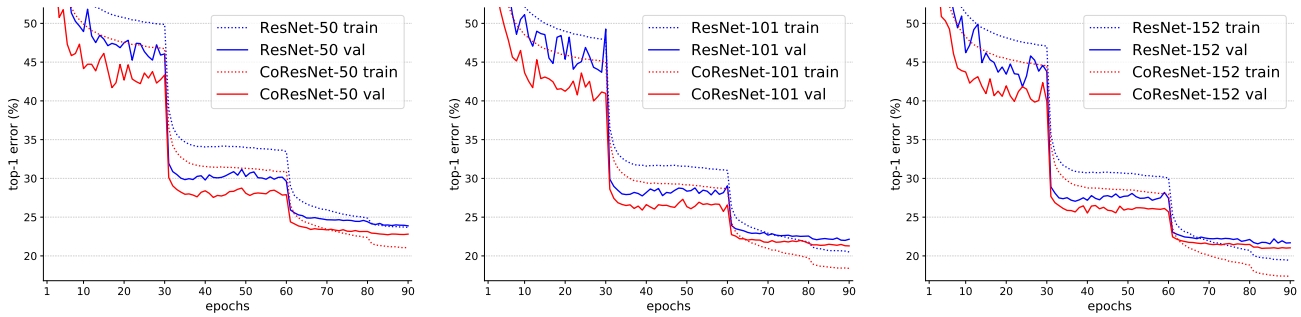


Figure 3: Training and validation curves on ImageNet for ResNet and CoResNet architectures of 50, 101 and 152 layers, respectively. Best viewed in color.

Network	network depth: 50				network depth: 101				network depth: 152			
	top-1	top-5	params	GFLOPs	top-1	top-5	params	GFLOPs	top-1	top-5	params	GFLOPs
ResNet [8]	23.88	7.06	25.56	4.14	22.00	6.10	44.55	7.88	21.55	5.74	60.19	11.62
pre-act. ResNet [9]	23.77	7.04	25.56	4.14	22.11	6.26	44.55	7.88	21.41	5.78	60.19	11.62
SE ResNet [11]	22.74	6.37	28.07	4.15	21.31	5.79	49.29	7.90	21.38	5.80	66.77	11.65
NL ResNet [32]	22.91	6.42	36.72	6.18	21.40	5.83	55.71	9.91	21.91	6.11	71.35	13.66
CoResNet [ours]	22.73	6.49	25.56	4.14	21.29	5.72	44.55	7.88	20.97	5.48	60.19	11.62

Table 5: ImageNet results of CoResNet in comparison with other state-of-the-art methods [9, 11, 32], considering architectures on different depths, ranging from 50 layers to 152 layers.

To make a direct comparison between our approach and DRN [37] under the same number of parameters and FLOPs, we also perform an experiment by setting the stride of CoResNet-50 to 8 instead of 32. As shown in Table 4, our approach provides improved recognition performance in comparison to DRN. As another evidence that our approach is superior to DRN, we can link the DRN results from Table 4 with our configuration $top(4, 3, 2, 1)$ from Table 3. More precisely, DRN and $top(4, 3, 2, 1)$ are from the same category of methods, as both use only the top dilation for convolution. We have already shown that this is not the optimal case for attaining good recognition performance, as it is necessary to have kernels that can capture detailed (local) information, as well as kernels that capture contextual information. We conjecture that the range of kernels from local to contextual is important for visual perception, as different levels of kernels bring complementary information into the visual system.

5.4. Comparison of CoResNet on other architectures of different depths

In Figure 3, we present the training and validation learning curves on ImageNet, considering ResNet and CoResNet architectures of 50, 101 and 152 layers, respectively. Comparing our CoResNet with the baseline ResNet [8], we can see an improved convergence during training, this being due to the fact that CoConv provides a more complete visual representation of the input.

Backbone	AP	AP@IoU=0.5	params	GFLOPs
ResNet-50	26.20	43.97	22.89	20.92
CoResNet-50	28.63	46.71	22.89	20.92
ResNet-101	29.58	47.69	41.89	48.45
CoResNet-101	31.19	49.89	41.89	48.45

Table 6: Results of SSD with various ResNet and CoResNet backbones of 50 or 101 layers on the MS COCO data set, for input images of 300×300 pixels. Higher AP and AP@IoU=0.5 values are better.

In Table 5, we provide the comparative results between CoResNet and several other works [8, 9, 11, 32], considering neural networks of 50, 101 and 152 layers deep, respectively. CoResNet outperforms the baseline ResNet [8] on all network depths. We also outperform the pre-activation ResNet [9] by a consistent margin, while maintaining the same number of parameters and computational cost. In terms of the top-1 error rate, we outperform the non-local block of Wang et al. [32] on all three tested network depths, namely 50, 101 and 152. We qualify our results as even more impressive, considering that the work of Wang et al. [32] significantly increases the number of parameters and FLOPs of the model. Interestingly, our CoResNet provides competitive results even when we compare it to the work of Hu et al. [11], although their work proposes an additional attention block (squeeze-and-excitation block) that needs to be inserted into the CNN, thus increasing the number of

Method	CIFAR-10		CelebA					
	IS	FID	MS-SSIM	SWD $\times 10^3$				
				128	64	32	16	Avg.
ProGAN [14]	7.60±0.09	20.70	0.2894	3.65	2.48	2.66	7.29	4.02
CoProGAN	7.71±0.06	19.66	0.2875	3.29	2.43	2.27	5.35	3.34

Table 7: ProGAN versus CoProGAN results on CIFAR-10 and CelebA. Higher IS values are better. Lower FID, MS-SSIM and SWD are better.

Figure 4: Examples generated by ProGAN and CoProGAN, selected by a human annotator from a set of 50 images generated from CIFAR-10 and CelebA, respectively. Best viewed in color.

learnable parameters of the model.

5.5. Object detection on MS COCO

In order to show the generality and the transfer learning capability of our approach, we integrate CoResNet in an object detection pipeline, namely the Single Shot Multi-Box Detector (SSD) [23]. As in [23], we remove all the layers of the ResNet backbones after the third stage to maintain the efficiency of the SSD. We also set the stride to 1 for the third stage to obtain 38×38 output feature maps from the backbones. The corresponding results, which are presented in Table 6, show that our approach provides significant improvements in detection performance, without affecting the number of parameters and computational cost.

5.6. Image generation on CIFAR-10 and CelebA

In Table 7, we compare our CoProGAN with ProGAN [14] on CIFAR-10 and CelebA, respectively. While the IS on CIFAR-10 indicates that our model is slightly better, the FID points to a lager difference in favor of CoProGAN. Analogously, on CelebA, the MS-SSIM indicates slight performance gains brought by CoConv, but the improvements measured by SWD are much higher, regardless of the resolution of the output (from 16×16 to 128×128 pixels). Overall, the empirical evidence indicates that CoProGAN produces superior results, regardless of the metric.

In Figure 4, we show the best and worst looking examples generated by ProGAN and CoProGAN selected by an independent human annotator from a set of 50 images generated from each data set. On CIFAR-10, we observe that our successful results seem more realistic, while our failure cases seems to contain objects with a global structure. On CelebA, our successful faces seem more symmetrical, while the faces seen in the CoProGAN failure cases are still distinguishable as faces.

6. Conclusion

We proposed contextual convolution (CoConv) as a direct replacement of the standard convolution, aiming to integrate contextual information at different levels of neural architectures. CoConv is efficient, maintaining the same requirements in the number of parameters and computational costs as the standard convolution, while providing improved visual recognition capabilities. Our contextual convolutional neural network (CoCNN) architectures are motivated by a series of neuroscience studies which clearly indicate the presence and importance of contextual modulation, even at the early stages of the biological visual systems, specifically in the V1 area. In this work, we showed that the findings in neuroscience can be applied to the artificial visual systems for object detection, recognition and generation, where we obtain significant improvements over several state-of-the-art baselines.

Acknowledgments

This work was supported by a grant of the Romanian Ministry of Education and Research, CNCS - UEFISCDI, project number PN-III-P1-1.1-TE-2019-0235, within PNCDI III. This article has also benefited from the support of the Romanian Young Academy, which is funded by Stiftung Mercator and the Alexander von Humboldt Foundation for the period 2020-2022.

References

[1] Thomas D Albright and Gene R Stoner. Contextual influences on visual processing. *Annual Review of Neuroscience*, 25(1):339–379, 2002. 1, 2

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. 2

[3] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of CVPR*, pages 1251–1258, 2017. 1

[4] Charles D Gilbert and Torsten N Wiesel. The influence of contextual stimuli on the orientation selectivity of cells in primary visual cortex of the cat. *Vision Research*, 30(11):1689–1701, 1990. 1

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of NIPS*, pages 2672–2680, 2014. 4

[6] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *arXiv:1706.02677*, 2017. 5

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of ICCV*, pages 2961–2969, 2017. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of CVPR*, pages 770–778, 2016. 1, 4, 5, 6, 7

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and J. Sun. Identity Mappings in Deep Residual Networks. In *Proceedings of ECCV*, pages 630–645, 2016. 1, 5, 7

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Proceedings of NIPS*, pages 6626–6637, 2017. 5

[11] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):2011–2023, 2020. 1, 2, 7

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely Connected Convolutional Networks. In *Proceedings of CVPR*, pages 4700–4708, 2017. 1

[13] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of ICML*, pages 448–456, 2015. 1, 4

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *Proceedings of ICLR*, 2018. 4, 5, 8

[15] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic gradient descent. In *Proceedings of ICLR*, 2015. 5

[16] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 5

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of NIPS*, pages 1097–1105, 2012. 1

[18] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. 1

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *Proceedings of CVPR*, pages 2117–2125, 2017. 1

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *Proceedings of ICCV*, pages 2980–2988, 2017. 1

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Proceedings of ECCV*, pages 740–755, 2014. 5

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single Shot Multibox Detector. In *Proceedings of ECCV*, pages 21–37, 2016. 5, 8

[24] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *Proceedings of ICCV*, pages 3730–3738, 2015. 5

[25] Vinod Nair and Geoffrey E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of ICML*, pages 807–814, 2010. 4

[26] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein Barycenter and Its Application to Texture Mixing. In *Proceedings of SSVM*, pages 435–446, 2012. 5

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 5

[28] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In *Proceedings of NIPS*, pages 3856–3866, 2017. 2

[29] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Proceedings of NIPS*, pages 2234–2242, 2016. 5

[30] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556*, 2014. 1

[31] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of CVPR*, pages 1–9, 2015. 1, 5

[32] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local Neural Networks. In *Proceedings of CVPR*, pages 7794–7803, 2018. 1, 2, 6, 7

[33] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *Proceedings of ACSSC*, volume 2, pages 1398–1402, 2003. 5

[34] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of ECCV*, pages 3–19, 2018. 1

[35] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. In *Proceedings of CVPR*, pages 1492–1500, 2017. 1

[36] Fisher Yu and Vladlen Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. In *Proceedings of ICLR*, 2016. 2

[37] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated Residual Networks. In *Proceedings of CVPR*, pages 472–480, 2017. 2, 6, 7

[38] Karl Zipser, Victor AF Lamme, and Peter H Schiller. Contextual modulation in primary visual cortex. *Journal of Neuroscience*, 16(22):7376–7389, 1996. 1, 2

[39] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning Transferable Architectures for Scalable Image Recognition. In *Proceedings of CVPR*, pages 8697–8710, 2018. 1