Contextual Convolution Blocks

David Marwood
marwood@google.com
Shumeet Baluja
shumeet@google.com

Google Research
Google, Inc.
Mountain View, CA

Abstract

A fundamental processing layer of modern deep neural networks is the 2D convolution. It applies a filter uniformly across the input, effectively creating feature detectors that are translation invariant. In contrast, fully-connected layers are spatially selective, allowing unique detectors across the input. However, full connectivity comes at the expense of an enormous number of free parameters to be trained, the associated difficulty in learning without over-fitting, and the loss of spatial coherence. We introduce Contextual Convolution Blocks, a novel method to create spatially selective feature detectors that are locally translation invariant. This increases the expressive power of the network beyond standard convolutional layers and allows learning unique filters for distinct regions of the input. The filters no longer need to be discriminative in regions not likely to contain the target features. This is a generalization of the Squeeze-and-Excitation architecture that introduces minimal extra parameters. We provide experimental results on three datasets and a thorough exploration into how the increased expressiveness is instantiated.

1 Introduction

Modern deep neural networks rely on large sets of learned convolutions for image and video processing tasks. The convolution operator learns translation invariant features across the spatial dimensions at each layer. When applied hierarchically, the detection of numerous features found in spatially local neighborhoods across the image are integrated in subsequent layers. This integration occurs through expanding regions as information is mixed through deeper layers. The benefits of convolutional neural networks (CNNs) have been demonstrated over three decades of study [16, 17, 18, 26, 30, 33].

Perhaps the most ubiquitous avenue of research within the neural network community is the development of more powerful network architectures. The architecture model dictates the features that are developed within the hidden layers by specifying how the co-occurrences and spatial correlations are captured. In this paper, we present a novel method to incorporate both spatial- and channel-focusing using a single, easily trained module, the Contextual Convolution Block (*CC-block*). The *CC-block* allows the network to emphasize feature-detectors in learned spatial regions of the input image. This gives the network the expressive power to maintain the translation invariant properties of convolutions where beneficial while simultaneously considering the image context, all using far fewer parameters than fully-connected layers. The primary contributions of this paper are:

- A novel combination of fully-connected and convolution layers. The *CC-block* creates detectors that (a) are effectively locally translation invariant, a feature not present in fully-connected layers, and (b) can be applied spatially selectively, a feature not present in convolution layers.
- Demonstrating how the *CC-block* can be instantiated with minimal extra parameters.
- We frame this work as a generalization of the *SE-block* [11] with spatial processing.
- As a secondary benefit, we present a method for learning filter weightings during training that are *not* dependent on the inputs at inference-time. Though counter-intuitive, these *effectively constant* filter weightings provide improvements in accuracy.

1.1 Representative Related Work

One of the early building blocks of modern neural architectures that integrated multi-scale convolutional features was the Inception architectures [30]. Since then, neural networks have grown rapidly in depth and size, for example the sizes of VGGNets to ResNet-152 and then residual attention networks [2, 10, 27, 35]. These increasingly deep architectures provide ample opportunity for continuous recalibration of the importance of information, for which various explicit methods have been proposed. For example, Highway Networks [29] use gates to control flow in shortcut connections. In other architectures, multiple and selective propagation path networks have been explored [6, 17, 31]. Past studies have analyzed layers independently as well as in groups [8, 14]. Increased depth has also provided the means to integrate information from across the image; this has been particularly useful in tasks that required the synthesis of images or textures [22, 25].

Traditional attention mechanisms apply a learned multiplicative or gating value to bias processing towards some features or away from others [13, 24, 39]. Attention mechanisms have performed well in numerous applications including sequence learning [38], image understanding and medical diagnosis [3], and captioning [4, 37]; the latter of which also uses channel-wise attention. Transformers are large networks that use positional encoding in a self-attention mechanism [34] with success in text processing [1] as well as image understanding and image generation [5, 23]. Squeeze-and-Excitation networks [11] use aspects of both, leveraging traditional attention by acting directly on the input while applying self-attention to gate individual channels of a CNN, an idea we will further extend.

In a CNN, each convolutional filter produces a single-channel activation image that becomes part of the inputs for the next layer. Under the hypothesis that some channels may be more important than others, the Squeeze-and-Excitation network's *SE-block* provides a lightweight architecture for (de-)emphasizing individual channels. Each channel is gated by a scalar multiplier derived from the layer's input. Approaches that use the *SE-block* have been deployed in a variety of scenarios including super resolution [7], removing rain from images [20] and medical diagnosis [19]. The EfficientNet framework has also employed the *SE-block* with state-of-the-art results on ImageNet [9, 32].

The *SE-block* is the closest related work to ours, both in implementation and purpose; we will contrast our approach with it throughout this paper. There have been other recent studies that examined the difficulties in extracting and preserving spatial information within CNNs. For example, [15] demonstrates a clever method to extract position information from CNNs by exploiting boundary effects on convolutions that are learned with standard finite sized inputs. [36] and [21] present methods to re-introduce spatial information into convolutions through the use of explicit position encoding channels. These studies have

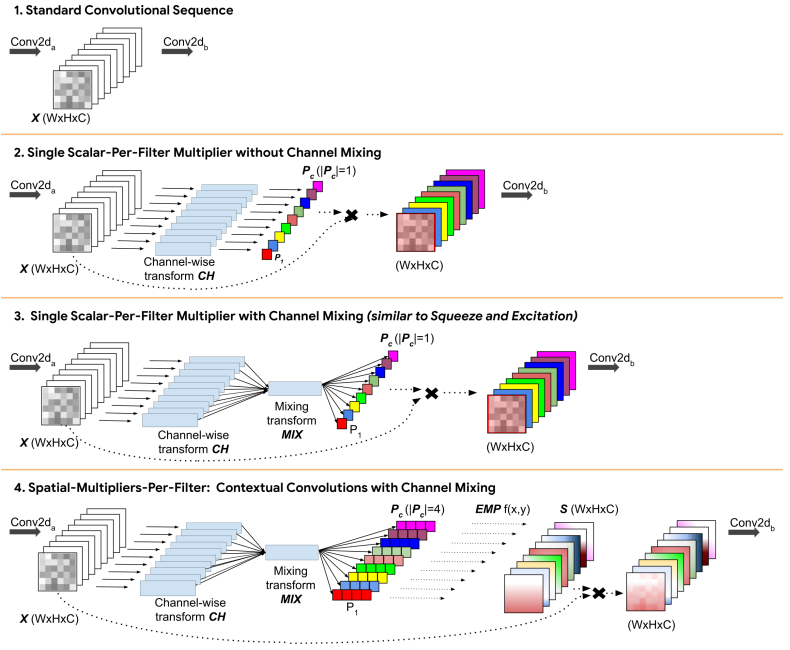


Figure 1: Spatial and channel excitation in the *CC-block*. (1) A standard CNN Convolution layer. (2) A single scalar multiplier $\boldsymbol{P_c}$ is produced per channel, gating the layer's input channels independently. (3) A mixing transform shares information across channels. This is similar to the Squeeze-and-Excitation block of [▭]. (4) Our work: rather than gating entire channels with one scalar, we gate spatially using a spatial excitation map, $\boldsymbol{S}$. #4 without mixing is also attempted; it is not shown for space considerations.

shown that maintaining spatial information in CNNs has benefits in classification, image synthesis and reinforcement learning tasks. In our work, we demonstrate a novel procedure in which the network determines both which spatial information is necessary to maintain and how to use it as a weighting for each layers' features.

2 Contextual Convolution Blocks

The Contextual Convolution Block (*CC-block*) is a series of network layers that compute soft attention on both the spatial and channel dimensions of its input. Internally, it learns a network that outputs just a few coefficients that become inputs to a spatial gating function. Figure 1 shows three possible designs of a *CC-block*.

We define the input of size $(W \times H)$ with C channels as $\boldsymbol{X} \in \mathbb{R}^{W \times H \times C}$. In Figure 1(#4), $\boldsymbol{X}$ is modified by a channel-wise transform $\boldsymbol{CH}$ and a mixing transform $\boldsymbol{MIX}$, producing our few coefficients. The coefficients are then rendered as a soft-gating *excitation map* the same size as the input $(W \times H \times C)$ using a pre-defined function; we will refer to this as the Excitation Map Producer module ($\boldsymbol{EMP}$). The composition of these steps, $\boldsymbol{S}(\boldsymbol{X}) = \boldsymbol{EMP} \circ \boldsymbol{MIX} \circ \boldsymbol{CH}(\boldsymbol{X})$ where $\circ$ is function composition, is element-wise multiplied by the input,

$$\textit{CC-block}(X) = \boldsymbol{S}(\boldsymbol{X}) \times \boldsymbol{X} = \boldsymbol{EMP} \circ \boldsymbol{MIX} \circ \boldsymbol{CH}(\boldsymbol{X}) \times \boldsymbol{X} \tag{1}$$

$\boldsymbol{CH}$ operates on channels independently so is decomposed as $\boldsymbol{CH} = \{\boldsymbol{CH_1}...\boldsymbol{CH_C}\}$. $\boldsymbol{EMP} = \{\boldsymbol{EMP_1}...\boldsymbol{EMP_C}\}$ is also a channel-wise operation, akin to depthwise separable convolutions [8]. All cross-channel communication, when it is used, is encompassed in $\boldsymbol{MIX}$.

The Squeeze-and-Excitation [11] architecture fits well within this model, see Figure 1(#3). The squeeze is a channel-wise transform, like our $\boldsymbol{CH}$. It creates a single scalar summary statistic *per channel* using a non-learned global average pool operation that intentionally drops any spatial information. The squeeze operation is followed by the channel recalibration step, like our $\boldsymbol{MIX}$, that communicates across channels. It employs two fully-connected layers with relu and sigmoid activations, respectively, to create an *excitation vector* of values between $\{0...1\}$, $\boldsymbol{M} \in \{0...1\}^C$, analogous to $\boldsymbol{P}$ in Figure 1(#3). To recalibrate the channels, *SE-block* broadcasts $\boldsymbol{M}$ across the spatial dimensions and multiplies it by the input $\boldsymbol{X}$. This performs a selective excitation, or soft attention, on the individual channels of $\boldsymbol{X}$.

We draw attention to two aspects of *SE-block*'s channel recalibration. First, note that it trivially maintains the translation invariance of the convolution layer by applying a uniform multiplier across all pixels in a channel. Second, note that $\boldsymbol{M}$ is itself a function of $\boldsymbol{X}$. As such, the excitation is dynamic — it is based on the input, and can be regarded as a self-attention function that operates at the granularity of individual channels.

Unlike *SE-block*, *CC-block* is spatially selective, similar to a fully-connected layer. To avoid the excessive number of learned weights in fully-connected layers, we allow only a few input values per channel, $\boldsymbol{P} = \{\boldsymbol{P_1}...\boldsymbol{P_C}\}$, to the excitation map producer $\boldsymbol{EMP}$, where the number of $\boldsymbol{EMP}$ inputs $|\boldsymbol{P_c}| \ll W \times H$. The $\boldsymbol{EMP}$ maps these few coefficients to the full $W \times H$ size. In contrast, *SE-block*'s M parameter is a per-channel multiplier that is applied uniformly to the entire channel, analogous to $|\boldsymbol{P_c}| = 1$. As such, *CC-block* is a relaxation of *SE-block*. We will provide quantitative evidence that the extra spatial expressiveness is used effectively in the *CC-block*s.

To approximate the translation invariance of a traditional convolutional layer in *CC-block*, we select a function for $\boldsymbol{EMP}$ that varies only slightly locally while still being globally spatially selective. Thus, *CC-block* approximates local translation invariance. We experimented with a variety of low-coefficient functions for the $\boldsymbol{EMP}$ (see Section 3.1.2). The simplest and the best performing was a bilinear interpolation from the image corners; we use it in the experiments below. This is used to create a smooth two-dimensional gradient across the input channel through a standard bilinear interpolation where, per channel, c, the inputs to $\boldsymbol{EMP_c}$ are $\boldsymbol{P_c} = \{initial_c^w, end_c^w, initial_c^h, end_c^h\}$. The interpolation is therefore:

$$f_c(x,y) = (initial_c^w \times (W-x)/W + end_c^w \times x/W) \times (initial_c^h \times (H-y)/H + end_c^h \times y/H) \quad (2)$$

Many of the extra network weights in the *CC-block* are devoted to the cross channel mixing, $\boldsymbol{MIX}$. Here, information *across* channels is integrated. In the experiments in which $\boldsymbol{MIX}$ is used, it outputs $|\boldsymbol{P}| = |\boldsymbol{P_c}| * C$ values. In the experiments in which $\boldsymbol{MIX}$ is not used, we instead define $\boldsymbol{S}(\boldsymbol{X}) = \boldsymbol{EMP} \circ \boldsymbol{CH}(\boldsymbol{X})$, where all operations are channel-wise. To further emphasize the channel independence, we can slice $\boldsymbol{S} = \{\boldsymbol{S_1}...\boldsymbol{S_C}\}$ and $\boldsymbol{X} = \{\boldsymbol{X_1}...\boldsymbol{X_C}\}$ and write $\boldsymbol{S_c}(\boldsymbol{X_c}) = \boldsymbol{EMP_c} \circ \boldsymbol{CH_c}(\boldsymbol{X_c})$, where each $\boldsymbol{CH_c}$ directly produces the inputs to the $\boldsymbol{EMP}$ ($|\boldsymbol{P_c}|$ values). As defined by [11], the *SE-block* employs cross-channel mixing. In the experimental section, we will explicitly examine how important this is to achieve the performance improvements.

Figure 2(A) shows a more detailed *CC-block-Full* design containing all the features of our implementation. To minimize the number of extra parameters introduced by the *CC-block*, the first step is to downsample the full $W \times H$ channel to 11×11 (or not change if

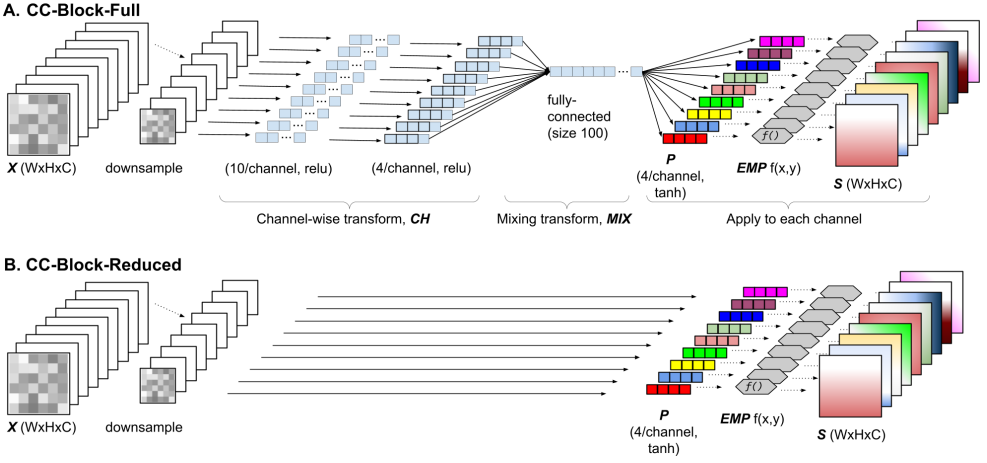


Figure 2: A fully expanded *CC-block*(top). Solid lines are learned layers and dashed are data-flow. ***CH*** is composed of 2 hidden layers, each applying a fully-connected layer to *each* channel independently. S is then multiplied by X to produce the output. Bottom: Reduced *CC-block* that uses neither ***CH*** nor ***MIX***. Experiments will be conducted with both A & B.

the channel is smaller) using a nearest-neighbor image resize. The size is not a sensitive parameter; other sizes yielded similar findings. ***CH*** is implemented as two per-channel hidden layers. ***MIX*** is a fully connected layer taking the concatenation of all the per-channel outputs from ***CH*** and outputs the $|\boldsymbol{P}|$ values described in the previous paragraph.

All the components of *CC-block* are differentiable. Training a system that uses *CC-block*s proceeds exactly the same as networks that do not – the *CC-block*s are trained simultaneously with any other layers using standard SGD.

2.1 A Step Back: Input-*Independent CC-block*

Before presenting the experiments with our complete system, we briefly examine the underlying assumptions of our model. The authors of [11] suggest that the dynamic reweighting behavior enhances the representational power of the network. Let's re-examine this fundamental assumption by hobbling our system to create an even simpler one. With a modification to Figure 1(#4), we create a version of the system that retains the full spatial and channel excitation map but is **independent of the actual inputs**, see Figure 3. This is done by altering $\boldsymbol{P}$: instead of making the coefficients $\boldsymbol{P}$ a function of the inputs, they are directly learned in training.[1] Then, to remove any dependence on the input, we redefine $\boldsymbol{S} = \boldsymbol{EMP}$, making $\textit{CC-block}(\boldsymbol{X}) = \boldsymbol{EMP} \times \boldsymbol{X}$. Since the excitation map $\boldsymbol{S}$ has no dependence on $\boldsymbol{X}$, it is **constant** at inference time.

Why might this simplified system work where a CNN might struggle? The answer lies in the training dynamics with this more expressive architecture, which is best illustrated by an example. Let us consider the sample task of face recognition with pre-aligned faces. A typical feature detector to emerge in these systems is one that detects eyes. However, when such a translation invariant detector is applied across the entire image, the shape will also

[1] Training these input-independent coefficients is accomplished through the usual back-propagation through the excitation map, $\boldsymbol{S}$. Conceptually, it is similar to training the parameters of a spatially dependent bias term.

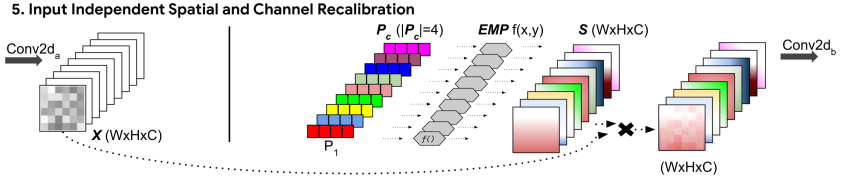


Figure 3: Input independent recalibration. The excitation map is *constant* at inference.

Table 1: Underlying Network Architectures for Experiments

Dataset	CNN Layers	Output Shape
MNIST	[Conv2D(3x3, relu), MaxPool(4x4)] repeated twice	$(2,2,192)$
CIFAR-10	[Conv2D(3x3, relu), MaxPool(2x2)] repeated four times	$(2,2,384)$
CIFAR-100	[Conv2D(3x3, relu), Conv2D(3x3, relu), MaxPool(2x2)] four times	$(2,2,384)$

trigger on various poses of the mouth. Therefore, in standard CNNs, the actual detector developed must be good at both recognizing eyes and not triggering false-positives on mouths. By adding the *CC-block* and simultaneously learning spatial excitation maps per channel, the network can learn the specific portion of an image to which a filter should be applied. With this added expressiveness, specific filters, such as the eye detector, can be weighted more heavily in the top half of the image. This makes the learning problem simpler, even when the mechanisms at inference are input-independent. Such a need for spatial selectivity is also found in a variety of images detection tasks in which the image is not pre-aligned; elements such as blue skies, green grass, pavement, *etc.*, have strong spatial priors that are easily exploited. We will include experiments using an input-independent *CC-block* next.

3 Experiments

In this section, we empirically evaluate the *CC-block* using simple VGG-like CNNs [28], shown in Table 1, on the MNIST, CIFAR-10, and CIFAR-100 datasets. We vary the number of channels output by the first Conv2D to experiment with different numbers of learned weights in the CNN, selecting from {1, 2, 4, 8, 16, 32, and 48}. The number of channels then doubles (quadruples for MNIST) in the output of each Conv2D following a MaxPool. In Table 1, the "Output Shape" is the (W,H,C) of the output of the final MaxPool in the case where we choose 48 channels of output (our largest network) for the first Conv2D.

In every experimental trial, the head of the CNN is two fully-connected layers: FC(128, relu) and FC(*num_classes*, softmax). In our experiments that use the *CC-block*, it appears after every Conv2D through the entire network. The only modification from Figure 2 is that input-dependent *CC-block*s modify the $\boldsymbol{EMP}$ input to be $tanh(\boldsymbol{P})/2+1$ to force the coefficients into the range $\{0.5...1.5\}$. This range was chosen because the gating/recalibration is multiplicative, and this centers the gating around 1.0 — a simple pass-through. Other ranges were tried, including $\{-1...1\}$; based on extensive testing, this range performed best.

Numerous empirical studies with architectures and settings for the *CC-block* were attempted, starting with the architecture in Figure 2 and successively *removing* portions to whittle down the number of extra parameters. The full suite of 5 experiments are:

1. **Standard**: Standard CNN with no *CC-block*s.
2. **Scalar Recalibration**: As in Figure 1(#2), we use $|\boldsymbol{P_c}| = 1$ so there is no spatial selectivity. This mimics the *SE-block* functionality.

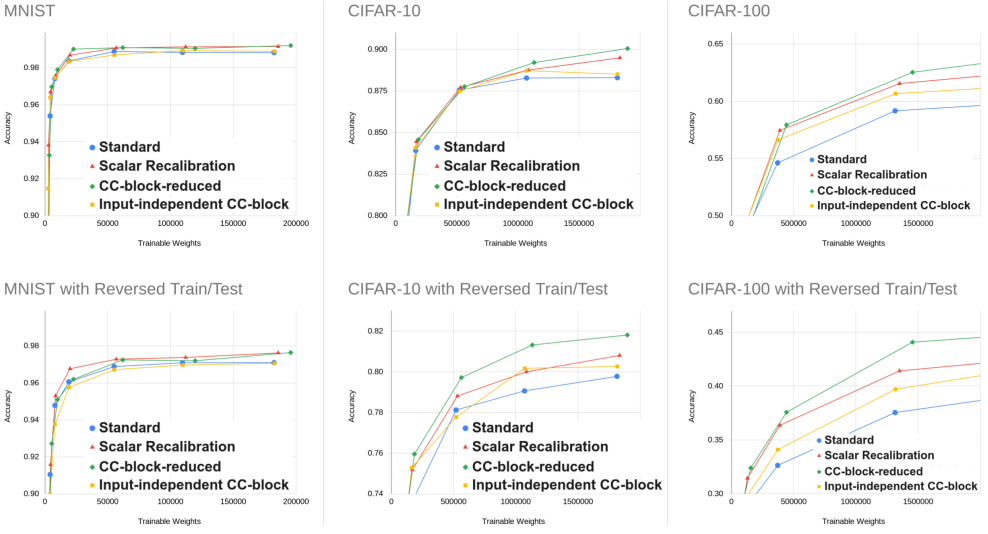


Figure 4: Top Row: Results from the 4 methods on MNIST, CIFAR-10, and CIFAR-100. Bottom Row: Results when the role of test and training sets are reversed – an opportunity to see how the systems compare in the presence of far less training data.

3. ***CC-block-Full***: Our approach in Figure 2(A) that uses both spatial and channel excitation, $|\boldsymbol{P_c}| = 4$. This is a direct extension to above with our spatial approach.
4. ***CC-block-Reduced***: The reduced version of *CC-block* (Figure 2(B)) in which both ***CH*** and ***MIX*** are removed. The downsampled channels directly create ***P***.
5. **Input-independent *CC-block***: Excitation is spatial, like *CC-block*, but *constant* at inference time (Figure 3). As a reminder, this is an ablative study.

In addition to the standard method of using these three datasets, we also repeated the entire set of experiments reversing the roles of the training and testing sets. In this manner, we examine the accuracies in the presence of much smaller training sets. In Figure 4, we present top-1 accuracy as a function of the number of learned weights (X-axis); this accounts for the weights required for the extra functionality.

Even though the performance of *CC-block-Full* and *CC-block-Reduced* are similar when using the same number of channels, when measuring the number of trainable parameters, *CC-block-Full* performs worse. We omit the results from *CC-block-Full* in the graphs in the interest of readability and will return to it in the ablative studies in the next section.

In the easiest task, MNIST, whether standard or reversed training/test sets, the task is easy enough for all the networks that the accuracies are largely indistinguishable until almost 10^5 weights are used, in which case the recalibration-based approaches start to do better.

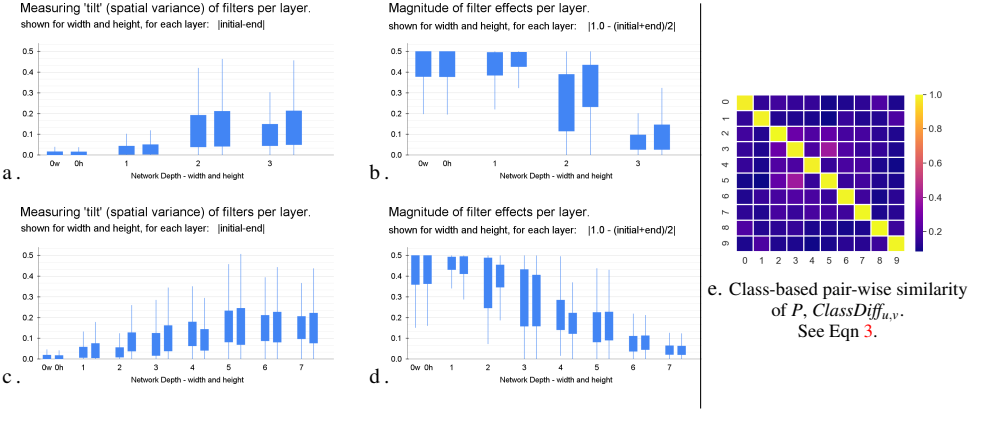
In the CIFAR-10 and CIFAR-100 tests, a clear pattern of performance emerges early and holds across both datasets and both the standard and reversed train/test set experiments. As the number of weights increases, the performance of the standard CNN (blue circles) does not keep up with the other three approaches. The best performance is achieved using *CC-block-Reduced* (green diamonds). Using Scalar Recalibration rather than *CC-block-Reduced*'s spatial excitation map performs second best. The Input-independent *CC-block*, that creates a constant excitation map per channel, also out-performs standard CNNs.

Several points should be noted about the experiments. First, the trends are consistent even when there are many fewer training examples in the reversed train/test trials. Second, the performance above standard CNNs grows as the number of trainable weights grows.

Third, we verified the result that even when the recalibration is *not* dynamic, and is independent of the input (Input-independent *CC-block*), performance improves. This intriguing finding implies that either the network learns different features and/or it learns the same features but that they are spatially recalibrated. Delving deeper, we performed an additional experiment in which we replaced spatial recalibrations with the simple scalar recalibrations found in *SE-block* ($|\boldsymbol{P_c}| = 1$) – *e.g.* a *static* per channel weighting. Surprisingly, this also yielded improvement over no-recalibration networks. What does this mean? The dynamic nature of the recalibrations is most likely *not* the sole contributor to the improved performance, as is commonly thought, neither in *SE-block* nor in *CC-block*. Rather, the network learns an alternate set of features than it would in the absence of expressive power afforded by either of the recalibration schemes. Next, we delve deeper into how the spatial recalibration is actually used within the network.

3.1 Analysis

To gain an understanding into how the *CC-block* recalibrations improve performance, let us examine how much they modify the network's processing. First, we measure how much spatial re-adjustment is made by each filter. For each channel (in every layer), we examine the absolute differences $|initial_c^w - end_c^w|$ and $|initial_c^h - end_c^h|$. If there is a large difference between the initial and end values then the range of adjustments, or "tilt", in the corresponding spatial dimension is large. The distribution of tilts for all the channels in the network are shown in Figure 5(a). Second, we examine the magnitude of the recalibration. Recall that the values in $\boldsymbol{P}$ have ranges of $\{0.5...1.5\}$. An average value of 1.0 indicates the average recalibration is identity while values farther from 1.0 indicate a larger magnitude of the av-



e. Class-based pair-wise similarity of P, $ClassDiff_{u,v}$. See Eqn 3.

Figure 5: (a,b) Tilt and magnitude of the recalibration described by $\boldsymbol{P}$ in the CIFAR-10 network in Table 1. (c,d) Same analysis with an 8 layer network (0=first layer after inputs, 7=last layer before fully-connected portion of classification). Statistical outliers omitted. (e) When examined by all pairs of classes, there is a strong distinction of values in $\boldsymbol{P}$ across network layers: different classes are unlikely to produce the same values of $\boldsymbol{P}$.

erage recalibration. For each filter, we compute $(initial_c^w + end_c^w)/2$ and $(initial_c^h + end_c^h)/2$ and plot the absolute difference from 1.0; see Figure 5(b).

In the early layers of the network's computation (lower numbers), there is little tilt, but a high magnitude, meaning that the recalibration is more akin to Squeeze-and-Excitation networks — little spatial variation, but large changes due to excitation. However, the opposite holds true deeper in the network (right side of each graph). The magnitude decreases but the tilt, or the amount of spatial variation, *increases* – thereby using more of the expressive power introduced in the *CC-block*. These measurements were conducted using *CC-block-Reduced* and CIFAR-10 (see Table 1). To verify these trends, we repeated the same experiments using an 8-layer version of the CIFAR-10 network. Results are shown in Figure 5(c&d). The same strong trends are visible in both measurements. Note that the whiskers of the box plots indicate a large range of values. This corresponds well to intuition; some filters trigger for some examples/classes, while others do not. We verify this next.

A second analysis attempts to uncover the sensitivity of *CC-block* to different input examples. Specifically, we examine whether examples from different classes produce different values in $\boldsymbol{P}$. Using the 8-layer CIFAR-10 network, we bucket examples by their class and compute the channel distribution of each value in $\boldsymbol{P}$ within each class, D_{class}. $\boldsymbol{P_c} = 4$ and there are 179 channels across the 8 *CC-block*s so each example produces $4 \times 179 = 716$ values in $\boldsymbol{P}$. Accordingly, there are also 716 distributions in each class in D_{class}. The channel distributions for each pair of classes, $(D_u \in D_{class}, D_v \in D_{class})$, are compared using a Mann-Whitney test, a non-parametric test of the difference of distributions.

$$ClassDiff_{u,v} = \sum_{d_i \in D_u} \sum_{d_j \in D_v} \begin{cases} 1, & \text{if } MannWhitney(d_i, d_j) < 0.01 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The pairwise results are shown in Figure 5(e). In the special case of the diagonal ($u = v$), the examples are split in two equal halves and the same process repeated. The diagonal distributions are correctly recognized as much more similar than the off-diagonal distributions, showing the spatial recalibration is strongly dependent on the specific class being inferred.

3.1.1 Network Ablation: *CC-block-Full* to *CC-block-Reduced*

A number of ablative experiments were conducted using CIFAR-100. Here, we summarize our most interesting and counter-intuitive results; see Figure 6. The original *SE-block* employs a squeeze operation, analogous to our ***CH***, as well as a channel recalibration step analogous to our ***MIX***. Surprisingly, we found that neither of these blocks were necessary for good performance. Rather, they often *hurt* the final classifications. The smallest architecture — without ***CH*** and ***MIX*** layers — performed best. It will be interesting to determine

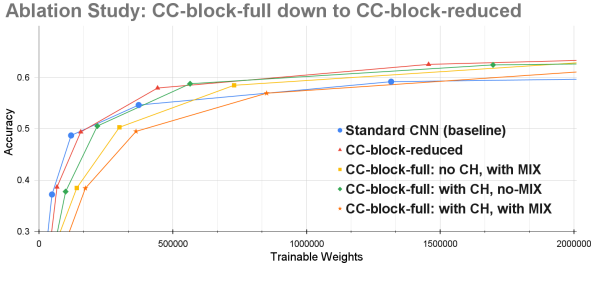


Figure 6: Ablation study. Three versions of *CC-block-Full*: (a) with ***CH*** and ***MIX***, (b) without ***MIX***, and (c) without ***CH***. *CC-block-Reduced* is also shown; from Figure 2(B).

if, and how, this finding changes as problem complexity increases. This is left for future exploration, as are its ramifications to current *SE-block* practice.

3.1.2 Alternative EMPs

In the results presented thus far, we employed bilinear interpolation in the ***EMP***. Other low-coefficient bases were also tried; Table 2 shows two alternatives, 2D-Gaussians and 2D-Sine, both with 4 coefficients. The results with these on CIFAR-100 (Table 1 with 32 channels) are shown in Table 3. As with the original *CC-block*, the values of $\boldsymbol{P}$ were offset to force the coefficients into the range $\{0.5...1.5\}$. The bilinear ***EMP*** was both the simplest and had the best performance; therefore was used in our full system.

Table 2: Two alternative ***EMP*** functions.

Function	Inputs $\boldsymbol{P_c}$	$f_c(x,y)$
2D-Gaussian	$\{\mu_c^w, \sigma_c^w, \mu_c^h, \sigma_c^h\}$	$(G(x\|\mu_c^w, \sigma_c^w)+0.5)*(G(y\|\mu_c^h, \sigma_c^h)+0.5)$
2D-Sine	$\{v_c^w, \phi_c^w, v_c^h, \phi_c^h\}$	$(\sin(v_c^w \times x+\phi_c^w)/2+1)*(\sin(v_c^h \times y+\phi_c^h)/2+1)$

Table 3: Different ***EMP*** functions on Table 1's CIFAR-100 architecture (32 channels).

Variant	Parameters	Accuracy
CC-block (bilinear)	1.5M	62.5%
Input-independent *CC-block* (bilinear)	1.3M	60.7%
Input-independent *CC-block* using 2D-Gaussian	1.3M	59.4%
Input-independent *CC-block* using 2D-Sine	1.3M	57.1%

4 Conclusions

We have presented a novel method for dynamically recalibrating distinct regions of each individual channel within each layer of deep CNNs. This increases the expressiveness of Squeeze and Excitation networks by learning spatial selectivity. By using a simple architecture and a non-learned, low-parameter ***EMP*** function to approximate local translation invariance, the number of added weights is minimized. Nonetheless, even when the cost of the small number of additional weights is considered, the *CC-block* shows consistent improvement over standard CNNs with no recalibration as well as over CNNs with the scalar recalibration like *SE-block*.

We also presented detailed experimental insights into *why* the improvements are so consistent. The commonly accepted notion that dynamic recalibration of the channels is responsible for the improved performance in *SE-block* (and by extension *CC-block*) is only part of the explanation. Rather, the features that are learned by networks with the ability to spatially recalibrate the channels are different than those learned without this expressive power. Lastly, through our ablative studies, we found that the even a reduced *CC-block*, one that does not utilize cross-channel information, worked as well as, and often better than, those with cross channel mixing. This counter-intuitive finding is relevant to *CC-block* and has implications for *SE-block* — simple, small-parameter architectures are sufficient.

Numerous avenues are open for future exploration. Three immediate areas include: (1) Measuring the contributions of the *CC-block* as a function of problem difficulty and network capacity and demonstrating our results on larger problems such as ImageNet with ResNet [10]; (2) The use of the *CC-block* in post-training regimes, *e.g.* for either network refinement or domain transfer; and (3) In terms of network analysis, visualizing the differences of features learned in networks with and without the *CC-block*.

References

[1] Jay Alammar. The illustrated transformer. http://jalammar.github.io/illustrated-transformer/, 2017. accessed: March 23, 2021.

[2] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.

[3] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2956–2964, 2015.

[4] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5659–5667, 2017.

[5] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020.

[6] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. *arXiv preprint arXiv:1707.01629*, 2017.

[7] Xi Cheng, Xiang Li, Jian Yang, and Ying Tai. Sesr: Single image super resolution with recursive squeeze and excitation networks. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 147–152. IEEE, 2018.

[8] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[12] Yani Ioannou, Duncan Robertson, Roberto Cipolla, and Antonio Criminisi. Deep roots: Improving cnn efficiency with hierarchical filter groups. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1231–1240, 2017.

[13] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001.

[14] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014.

[15] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14274–14285, 2020.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[17] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[18] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, pages 253–256. IEEE, 2010.

[19] Meng Lei, Jia Li, Ming Li, Liang Zou, and Han Yu. An improved unet++ model for congestive heart failure diagnosis using short-term rr intervals. *Diagnostics*, 11(3):534, 2021.

[20] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 254–269, 2018.

[21] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *arXiv preprint arXiv:1807.03247*, 2018.

[22] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[23] A. Ramesh, M. Pavlov, G. Goh, and S. Gray. Dall·e: Creating images from text. https://openai.com/blog/dall-e/, 2021. accessed: March 23, 2021.

[24] Adria Recasens, Petr Kellnhofer, Simon Stent, Wojciech Matusik, and Antonio Torralba. Learning to zoom: a saliency-based sampling layer for neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 51–66, 2018.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[26] Henry A Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 20(1):23–38, 1998.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR: International Conference on Learning Representations*, 2015.

[29] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[30] 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 the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[31] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[32] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[33] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[35] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017.

[36] Zhenyi Wang and Olga Veksler. Location augmentation for CNN. *CoRR*, abs/1807.07044, 2018. URL http://arxiv.org/abs/1807.07044.

[37] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention, 2016.

[38] Kun Xu, Lingfei Wu, Zhiguo Wang, Yansong Feng, Michael Witbrock, and Vadim Sheinin. Graph2seq: Graph to sequence learning with attention-based neural networks. *arXiv preprint arXiv:1804.00823*, 2018.

[39] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.