On Translation Invariance in CNNs: Convolutional Layers can Exploit Absolute Spatial Location

Osman Semih Kayhan
Computer Vision Lab
Delft University of Technology

Jan C. van Gemert
Computer Vision Lab
Delft University of Technology

Abstract

In this paper we challenge the common assumption that convolutional layers in modern CNNs are translation invariant. We show that CNNs can and will exploit the absolute spatial location by learning filters that respond exclusively to particular absolute locations by exploiting image boundary effects. Because modern CNNs filters have a huge receptive field, these boundary effects operate even far from the image boundary, allowing the network to exploit absolute spatial location all over the image. We give a simple solution to remove spatial location encoding which improves translation invariance and thus gives a stronger visual inductive bias which particularly benefits small data sets. We broadly demonstrate these benefits on several architectures and various applications such as image classification, patch matching, and two video classification datasets.

Figure 1. We place an identical image patch on the top-left or on the bottom-right of an image. We evaluate a standard fully convolutional network [35, 43, 61, 93, 95, 105] if it can classify the patch location (top-left vs bottom-right). We use 1 layer, a single 5x5 kernel, zero-padding, same-convolution, ReLu, global max pooling, SGD, and a soft-max loss. Surprisingly, this network can classify perfectly, demonstrating that current convolutional layers can exploit the absolute spatial location in an image.

1. Introduction

The marriage of the convolution operator and deep learning yields the Convolutional Neural Network (CNN). The CNN arguably spawned the deep learning revolution with AlexNet [54] and convolutional layers are now the standard backbone for various Computer Vision domains such as image classification [35, 89, 95], object detection [65, 77, 79], semantic segmentation [34, 52, 81], matching [66, 32, 107], video [12, 33, 88], generative models [25, 29, 51], *etc*. The CNN is now even used in other modalities such as speech [1, 58, 73], audio [15, 40, 82], text [14, 20, 56], graphs [11, 21, 84], *etc*. It is difficult to overstate the importance of the convolution operator in deep learning. In this paper we analyze convolutional layers in CNNs which is broadly relevant for the entire deep learning research field.

For images, adding convolution to neural networks adds a visual inductive prior that objects can appear anywhere. Convolution can informally be described as the dot product between the input image and a small patch of learnable weights –the kernel– sliding over all image locations. This shares the weights over locations yielding a huge reduction in learnable parameters. Convolution is equivariant to translation: If an object is shifted in an image then the convolution outcome is shifted equally. When convolution is followed by an operator that does not depend on the position, such as taking the global average or global maximum, that gives translation invariance and absolute location is lost. Translation invariance powers the visual inductive prior of the convolution operator, and we will demonstrate that improving translation invariance improves the prior, leading to increased data efficiency in the small data setting.

In this paper we challenge standard assumptions about translation invariance and show that currently used convolutional layers can exploit the absolute location of an object in an image. Consider Fig. 1, where the exactly identical image patch is positioned on the top left (class 1) or on the bottom right (class 2) in an image. If a fully convolutional CNN is invariant, it should not be able to classify and give random performance on this task. Yet, surprisingly, a simple standard 1-layer fully convolutional network with a global max pooling operator can perfectly classify the location of the patch and thus exploit absolute spatial location.

We show that CNNs can encode absolute spatial location by exploiting image boundary effects. These effects occur because images have finite support and convolving close to the boundary requires dealing with non-existing values beyond the image support [47, 94]. Boundary effects allow CNNs to learn filters whose output is placed outside the image conditioned on their absolute position in the image. This encodes position by only keeping filter outputs for specific absolute positions. It could, for example, learn filters that only fire for the top of the image, while the bottom responses are placed outside the image boundary. Boundary effects depend on the size of the convolution kernel and are small for a single 3x3 convolution. Yet, CNNs stack convolution layers, yielding receptive fields typically several times the input image size [4]. Boundary effects for such huge kernels are large and, as we will demonstrate, allows CNNs to exploit boundary effects all over the image, even far away from the image boundary.

We have the following contributions. We show how boundary effects in discrete convolutions allow for location specific filters. We demonstrate how convolutional layers in various current CNN architectures can and will exploit absolute spatial location, even far away from the image boundary. We investigate simple solutions that removes the possibility to encode spatial location which increases the visual inductive bias which is beneficial for smaller datasets. We demonstrate these benefits on multiple CNN architectures on several application domains including image classification, patch matching, and video classification.

2. Related Work and Relevance

Fully connected and fully convolutional networks. Initial CNN variants have convolutional layers followed by fully connected layers. These fully connected layers can learn weights at each location in a feature map and thus can exploit absolute position. Variants of the seminal LeNet that included fully connected layers experimentally outperformed an exclusively convolutional setup [59]. The 2012 ImageNet breakthrough as heralded by AlexNet [54] followed the LeNet design, albeit at larger scale with 5 convolutional and 2 fully connected layers. Building upon AlexNet [54], the VGG [89] network family variants involve varying the depth of the convolutional layers followed by 3 fully connected layers. The fully connected layers, however, take up a huge part of the learnable parameters making such networks large and difficult to train.

Instead of using fully connected layers, recent work questions their value. The Network In Network [61] is a fully convolutional network and simply replaces fully connected layers by the global average value of the last convolutional layer's output. Such a global average or global max operator is invariant to location, and makes the whole network theoretically insensitive to absolute position by building on top of equivariant convolutional layers. Several modern networks are now using global average pooling. Popular and successful examples include the The All Convolutional Net [93], Residual networks [35], The Inception family [95], the DenseNet [43], the ResNext network [105] *etc*. In this paper we show, contrary to popular belief, that fully convolutional networks will exploit the absolute position.

Cropping image regions. Encoding absolute location has effect on cropping. Examples of region cropping in CNNs include: The bounding box in object detection [27, 34, 79]; processing a huge resolution image in patches [42, 86]; local image region matching [32, 66, 108, 107]; local CNN patch pooling encoders [3, 6, 8]. The region cropping can be done *explicitly* before feeding the patch to a CNN as done in R-CNN [27], high-res image processing [42] and aggregation methods [80, 87]. The other approach to cropping regions is *implicitly* on featuremaps after feeding the full image to a CNN as done in Faster R-CNN [79], BagNet [8], and CNN pooling methods such as sum [6], BoW [76], VLAD [3, 28], Fisher vector [16]. In our paper we show that CNNs can encode the absolute position. This means that in contrast to explicitly cropping a region before the CNN, cropping a region after the CNN can include absolute position information, which impacts all implicit region cropping methods.

Robustness to image transformations. The semantic content of an image should be invariant to the accidental camera position. Robustness to such geometric transformation can be learned by adding them to the training set using data augmentation [18, 24, 39, 41, 50]. Instead of augmenting with random transformations there are geometric adverserial training methods [22, 23, 49] that intelligently add the most sensitive geometric transformations to the training data. Adding data to the training set by either data augmentation or adverserial training is a brute-force solution adding additional computation as the dataset grows.

Instead of adding transformed versions of the training data there are methods specifically designed to learn geometric transformations in an equivariant or invariant representation [7, 53, 60] where examples include rotation [19, 69, 102, 103, 110], scale [68, 92, 99, 104, 106] and other transformations [17, 26, 38, 57, 90]. Closely related is the observation that through subsequent pooling and subsampling in CNN layers translation equivariance is lost [5, 109]. In our paper, we also investigate the loss of translation equivariance, yet do not focus on pooling but instead show that convolutional layers can exploit image boundary effects to encode the absolute position which was also found independently by Islam *et al*. [45].

Boundary effects. Boundary effects cause statistical biases in finitely sampled data [30, 31]. For image processing this is textbook material [47, 94], where boundary handling has applications in image restoration and deconvolu-

tions [2, 63, 78]. Boundary handling in CNNs focuses on minimizing boundary effects by learning separate filters at the boundary [44], treating out of boundary pixels as missing values [62], circular convolutions for wrap-around input data such as 360° degree images [85] and minimizing distortions in 360° degree video [13]. We, instead, investigate how boundary effects can encode absolute spatial location.

Location information in CNNs. Several deep learning methods aim to exploit an absolute spatial location bias in the data [64, 100]. This bias stems from how humans take pictures where for example a sofa tends to be located on the bottom of the image while the sky tends to be at the top. Explicitly adding absolute spatial location information helps for patch matching [67, 71], generative modeling [101], semantic segmentation [36, 100], instance segmentation [72]. In this paper we do not add spatial location information. Instead, we do the opposite and show how to remove such absolute spatial location information from current CNNs.

Visual inductive priors for data efficiency. Adding visual inductive priors to deep learning increases data efficiency. Deep networks for image recognition benefit from a convolutional prior [97] and the architectural structure of a CNN with random weights already provides an inductive bias [48, 83, 96]. The seminial Scattering network [10] and its variants [74, 75] design a convolutional architecture to incorporate physical priors about image deformations. Other work shows that adding priors increases data efficiency by tying parameters [26], sharing rotation responses [103], and a prior scale-space filter basis [46]. In our paper we show that removing the ability of convolutional layers to exploit the absolute position improves translation equivariance and invariance which enforces the visual inductive prior of the convolution operator in deep learning.

3. How boundary effects encode location

We explore common convolution types for boundary handling with their image padding variants and explore their equivariant and invariant properties. In Fig. 2 we illustrate the convolution types. For clarity of presentation we mostly focus on $d = 1$ dimensional convolutions in a single channel, although the analysis readily extends to the multi-dimensional multi-channel case. We use the term 'image' broadly and also includes feature maps.

Boundaries for convolution on finite samples. Let $\mathbf{x} \in \mathbb{R}^n$ be the 1-D single channel input image of size n and $\mathbf{f} \in \mathbb{R}^{2k+1}$ denote a 1-D single channel filter where for convenience we only consider odd sized filters of size $2k + 1$. The output $y[t]$ for discrete convolution is

$$\mathbf{y}[t] = \sum_{j=-k}^{k} \mathbf{f}[j]\mathbf{x}[t-j]. \tag{1}$$

Images have finite support and require handling boundary cases, for example where $t - j < 0$ and $x[t - j]$ falls outside the defined image. Providing values outside the image boundary is commonly referred to as padding. We consider two cases. *Zero padding* assumes that all values outside of the images are zero. *Circular padding* wraps the image values on one side around to the other side to provide the missing values.

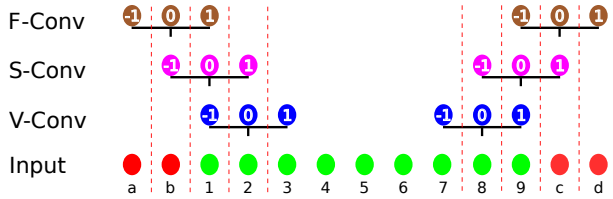

Figure 2. How convolution ignores positions close to the border. We show the first and the last position for three convolution types: Valid (V-Conv), Same (S-Conv) and Full (F-Conv) applied to an input with finite support (green) and border padding (red). Note that for V-conv, the blue filter at position 1 is never applied to the green input positions 1 and 2. For S-Conv, the pink filter position 1 is never applied to green input position 1. F-Conv has all filter values applied on the image.

3.1. Common convolutions for boundary handling

Valid convolution (V-Conv). V-Conv does not convolve across image boundaries. Thus, V-conv is a function $\mathbb{R}^n \rightarrow \mathbb{R}^{n-2k}$ where the output range of Eq. (1) is in the interval:

$$t \in [k+1, n-k]. \tag{2}$$

It only considers existing values and requires no padding. Note that the support of the output y has $2kd$ fewer elements than the input x, where d is the dimensionality of the image, *i.e.*, the output image shrinks with k pixels at all boundaries.

Same convolution (S-Conv). S-Conv slides only the filter center on all existing image values. The output range of Eq. (1) is the same as the input domain; *i.e.* the interval:

$$t \in [1, n]. \tag{3}$$

The support of the output y is the same size as the support of the input x. Note that $2kd$ values fall outside the support of x, *i.e.*, at each boundary there are k padding values required.

Full convolution (F-Conv). F-Conv applies each value in the filter on all values in the image. Thus, F-conv is a function $\mathbb{R}^n \rightarrow \mathbb{R}^{n+2k}$ where the output range of Eq. (1) is in the interval:

$$t \in [-k, n+k]. \tag{4}$$

The output support of y has $2kd$ more elements than the input x, *i.e.*, the image grows with k elements at each boundary. Note that $4kd$ values fall outside of the support of the input x: At each boundary $2k$ padded values are required.

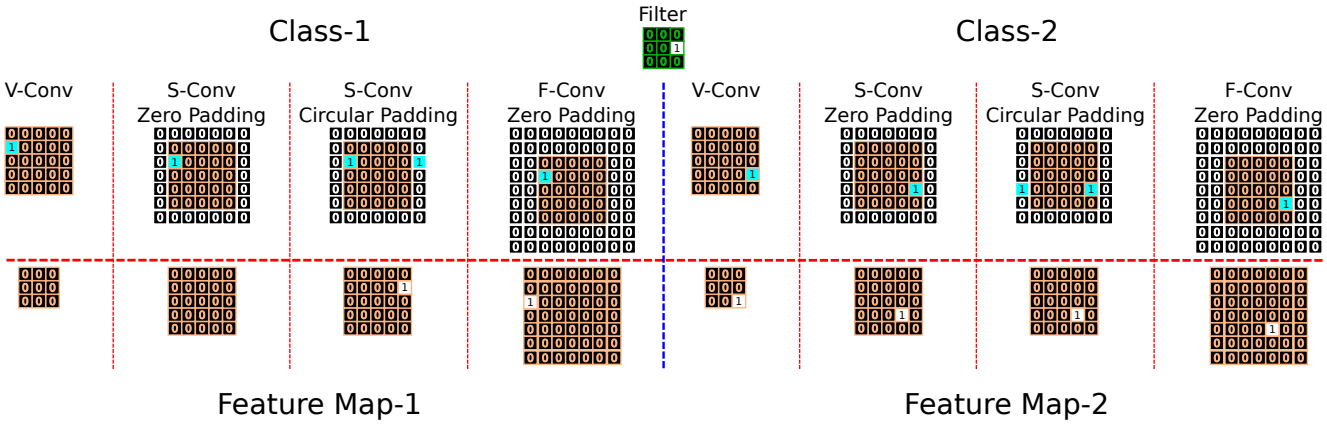

Figure 3. A 2D Example where a pixel on the top left input (Class-1) and the same pixel on the bottom-right input (Class-2) can be classified using convolution. Comparing the output of 4 convolution types shows that V-Conv and S-Conv for Class-1 can no longer detect the pixel, while Class-2 still has the pixel. S-Conv with circular padding and F-Conv always retain the pixel value.

3.2. Are all input locations equal?

We investigate if convolution types are equally applied to all input position in an image. In Fig. 2 we illustrate the setting. To analyze if each location is equal, we modify Eq. (1) to count how often an absolute spatial position a in the input signal x is used in the convolution. The count $C(\cdot)$ sums over all input positions i where the convolution is applied,

$$C(a) = \sum_{i} \sum_{j=-k}^{k} [\![i = a - j]\!], \tag{5}$$

where $[\![\cdot]\!]$ are Iverson Brackets which evaluate to 1 if the expression in the brackets is true. Without boundary effects $C(a)$ always sums to $2k+1$ for each value of a.

When there are boundary effects, there will be differences. For V-Conv, the input locations i are determined by Eq. (2) and the equation becomes

$$C_V(a) = \sum_{i=k+1}^{n-k} \sum_{j=-k}^{k} [\![i = a - j]\!], \tag{6}$$

where i no longer sums over all values. Thus, for all locations in the input image the function $C_V(t)$ no longer sums to $2k+1$ as it does in Eq. (5), instead they sum to a lower value. In fact, it reduces to

$$C_V(a) = \begin{cases} a & \text{if } a \in [1, 2k] \\ n - a + 1 & \text{if } a \in [n-2k, n] \\ 2k+1 & \text{Otherwise.} \end{cases} \tag{7}$$

This shows that for V-Conv there are absolute spatial locations where the full filter is not applied.

For S-Conv, where Eq. (3) defines the input, the count is

$$C_S(a) = \sum_{i=1}^{n} \sum_{j=-k}^{k} [\![i = a - j]\!], \tag{8}$$

where i sums over all values, and slides only the filter center over all locations. Thus, for S-Conv, when the locations are $a \leq k$ or $a \geq n-k$, the function $C_S(a)$ no longer sums to $2k+1$. This reduces to

$$C_S(a) = \begin{cases} a + k & \text{if } a \in [1, k] \\ n - a + (k+1) & \text{if } a \in [n-k, n] \\ 2k+1 & \text{Otherwise.} \end{cases} \tag{9}$$

This means that also for S-Conv there are absolute spatial locations where the full filter is not applied.

S-Conv with circular padding 'wraps around' the image and uses the values on one side of the image to pad the border on the other side. Thus, while for S-Conv, Eq. (9) holds for the absolute position i, it is by using circular padding that the *value* $x[i]$ at position i is exactly wrapped around to the positions where the filter values were not applied. Hence, circular padding equalizes all responses, albeit at the other side of the image. Zero padding, in contrast, will have absolute spatial locations where filter values are never applied.

For F-Conv, in Eq. (4), the counting equation becomes

$$C_F(a) = \sum_{i=-k}^{n+k} \sum_{j=-k}^{k} [\![i = a - j]\!]. \tag{10}$$

F-Conv sums the filter indices over all indices in the image and thus, as in Eq. (5), all locations i sum to $2k+1$ and thus no locations are left out.

We conclude that V-Conv is the most sensitive to exploitation of the absolute spatial location. S-Conv with zero padding is also sensitive to location exploitation. S-Conv with circular padding is not sensitive, yet involves wrapping values around to the other side, which may introduce semantic artifacts. F-Conv is not sensitive to location information. In Fig. 3 we give an example of all convolution types and how they can learn absolute spatial position.

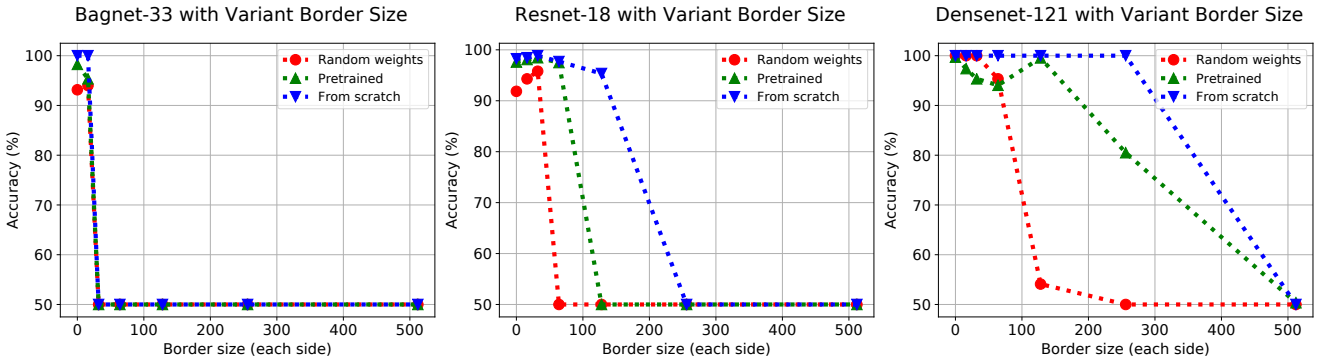

Figure 4. **Exp 1:** Evaluating a BagNet-33 [8] (left), a ResNet-18 [35] (middle) and a DenseNet-121 [43] (right) on how far from the boundary absolute location can be exploited, see Fig. 5. The x-axis is the border size added to all 4 sides of the image and the y-axis is accuracy. All models can classify absolute position. The small RF of the BagNet allows for classification close to the border. The ResNet-18 and DenseNet-121 have larger RFs and can classify location far from the boundary. Random convolutional weights stay relatively close to the boundary while training on ImageNet learns filters that can go further. Training from scratch does best. Note that the most distant location from an image boundary for a kxk image is a border size of $k/2$, *i.e.*, a border size of 128 corresponds to a 256x256 image.

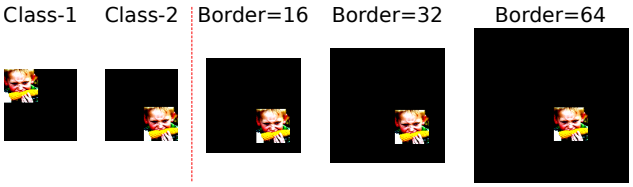

Figure 5. **Exp 1:** Example images. Evaluating how far from the image boundary absolute location can be exploited. The task is to classify the location of a 56x56 resized Imagenet image placed in the top-left (class-1) and bottom-right (class-2), see also Fig. 1. We add a border on all 4 sides of the image, where we increase the border size until location can no longer be classified.

4. Experiments

Implementation details for Full Convolution. For standard CNNs implementing F-Conv is trivially achieved by simply changing the padding size. For networks with residual connections, we add additional zero padding to the residual output to match the spatial size of the feature map. We will make all our experiments and code available[1].

4.1. Exp 1: How far from the image boundary can absolute location be exploited?

CNNs can encode absolute position by exploiting boundary effects. In this experiment we investigate how far from the boundary these effects can occur. Can absolute position be encoded only close to the boundary or also far away from the boundary? To answer this question we revisit the location classification setting in Fig. 1 while adding an increasingly large border all around the image until location can no longer be classified. In Fig. 5 we show the setting.

We randomly pick 3,000 samples from ImageNet validation set, resize them to 56x56 and distribute them equally in a train/val/test set. For each of the 3k samples we create two new images (so, 2,000 images in each of the 3 train/val/test sets) by taking a black 112x112 image and placing the resized ImageNet sample in the top-left corner (class-1) and in the bottom-right corner (class-2), see Fig. 1. To evaluate the distance from the boundary we create 7 versions by adding a black border of size $\in \{0, 16, 32, 64, 128, 256, 512\}$ on all 4 sides of the 112x112 image, see Fig. 5 for examples.

We evaluate three networks with varying receptive field size. BagNet-33 [8] is a ResNet variant where the receptive field is constrained to be 33x33 pixels. ResNet-18 [35] is a medium sized network, while a DenseNet-121 [43] is slightly larger. We evaluate three settings: (i) trained completely from scratch to see how well it can do; (ii) randomly initialized with frozen convolution weights to evaluate the architectural bias for location classification; (iii) ImageNet pre-trained with frozen convolution weights to evaluate the location classification capacity of a converged realistic model used in a typical image classification setting.

Results in Fig. 4 show that all settings for BagNet, ResNet and DenseNet can classify absolute position. Random weights can do it for locations relatively close to the boundary. Surprisingly, the pre-trained models have learned filters on ImageNet that can classify position further away from the boundary as compared to random initialization. The models trained from scratch can classify absolute position the furthest away from the boundary. The BagNet fails for locations far from the boundary. Yet, the medium-sized ResNet-18 can still classify locations of 128 pixels away from the boundary, which fully captures ImageNet as for 224x224 images the most distant pixel is only 112 pixels from a boundary. We conclude that absolute location can even be exploited far from the boundary.

[1] https://github.com/oskyhn/CNNs-Without-Borders

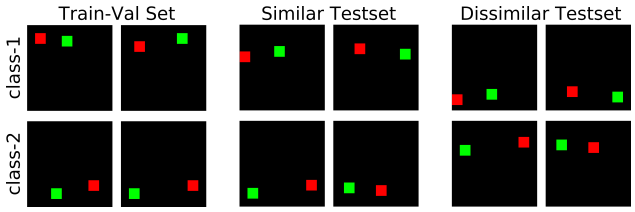

Figure 6. **Exp 2:** Example images of the Red-Green two class classification dataset for evaluating exploitation of absolute position. The upper row of images is class 1: Red-to-the-left-of-Green. The lower row of images is class 2: Green-to-the-left-of-Red. The Similar Testset is matching the Train-Val set in absolute location: Class 1 at the top and class 2 at the bottom. The Dissimilar testset is an exact copy of the Similar testset where absolute location is swapped between classes: Class 1 at the bottom, Class 2 at the top. If absolute location plays no role then classification on the Similar Testset would perform equal to the Dissimilar Testset.

4.2. Exp 2: Border handling variants

Border handling is the key to absolute location coding. Here we evaluate the effect of various border handling variants on absolute location exploitation. To do so, we create an image classification task unrelated to the absolute position and introduce a location bias which should have no effect on translation invariant architectures.

We construct the Red-Green data set for binary image classification of the relative order of colored blocks of 4x4 on a black 32x32 image. Class 1 has Red to the left of Green; class 2 has Green to the left of Red, see Fig. 6. The classification task is unrelated to the absolute position. We introduce a vertical absolute position bias by placing class 1 on the top of the image (8 pixels from the top, on average), and class 2 on the bottom (8 pixels from the bottom, on average). We then construct two test sets, one with similar absolute location bias, and a dissimilar test set where the location bias switched: class 1 at the bottom and class 2 on top, see Fig. 6.

The train set has 2,000 images, the validation and test sets each have 1,000 images. Experiments are repeated 10 times with different initialization of the networks. A 4-layer fully convolutional deep network is used for evaluation. The first two layers have 32 filters and last two layers 64 filter followed by global max pooling. Sub-sampling for layers 2, 3, 4 uses stride 2 convolution.

We evaluate the border handling of Section 3. *V-Conv* uses only existing image values and no padding. For *S-Conv* we evaluate zero and circular padding. *F-Conv* has zero padding. Results are in Table 1. *V-Conv* and *S-Conv*-zero have the best accuracy on the Similar test set, yet they exploit the absolute location bias and perform poorly on the Dissimilar test set, where *V-Conv* relies exclusively on location and confuses the classes completely. *S-Conv*-circ and *F-Conv* perform identical on the Similar and Dissimilar test sets; they are translation invariant and thus cannot exploit the absolute location bias. *F-Conv* does better than *S-Conv*-circ because circular padding introduces new content. *F-Conv* does best on both test sets as it is translation invariant and does not introduce semantic artifacts.

Type	Pad	Similar test	Dissimilar test
V-Conv	-	$\mathbf{100.0 \pm 0.0}$	0.2 ± 0.1
S-Conv	Zero	99.8 ± 0.1	8.4 ± 0.7
S-Conv	Circ	73.7 ± 1.0	73.7 ± 1.0
F-Conv	Zero	89.7 ± 0.5	$\mathbf{89.7 \pm 0.5}$

Table 1. **Exp 2:** Accuracy on the Red-Green dataset shown in Fig. 6. Type is the convolution type, pad is how padding is done. Results are given on the Similar test set with matching absolute positions and the Dissimilar test set with an absolute position mismatch. Stddevs are computed by 10 repeats. *Valid* and *same*-zero exploit location and do poorly on the Dissimilar test set. *Same*-circ is translation invariant yet invents disturbing new content. *Full*-zero is translation invariant, doing well on both test sets.

4.3. Exp 3: Sensitivity to image shifts

Does removing absolute location as a feature lead to robustness to location shifts? We investigate the effect of image shifts at test time on CNN output for various architectures on a subset of ImageNet. We train four different architectures from scratch with S-Conv and F-Conv: Resnet 18, 34, 50 and 101. To speed up training from scratch, we use 20% of the full ImageNet and take the 200 classes from [37] which is still large but 5x faster to train. To evaluate image shifts we follow the setting of BlurPool [109], which investigates the effect of pooling on CNN translation equivariance. As BlurPool improves equivariance, we also evaluate the effect of BlurPool Tri-3 [109].

Diagonal Shift. We train the network with the usual central crop. Each testing image is diagonally shifted starting from the top-left corner towards the bottom-right corner. We shift 64 times 1 pixel diagonally. Accuracy is evaluated for each pixel shift and averaged over the full test set.

Consistency. We measure how often the classification output of a model is the same for a pair of randomly chosen diagonal shifts between 1 and 64 pixels [109]. We evaluate each test image 5 times and average the results.

Results are given in Table 2. For each architecture, using F-Conv improves both the classification performance and the consistency of all the models. The highest classification accuracy gain between S-Conv and F-Conv is 3.6% and the best consistency gain is 2.49% with Resnet-34. BlurPool makes S-Convs more robust to diagonal shifts and increase consistency. When F-Conv and BlurPool are combined, the accuracy on diagonal shifting and consistency are improved further. Resnet-34 (F+BlurPool) obtains more 4.85% of accuracy and 3.91% of consistency compared to

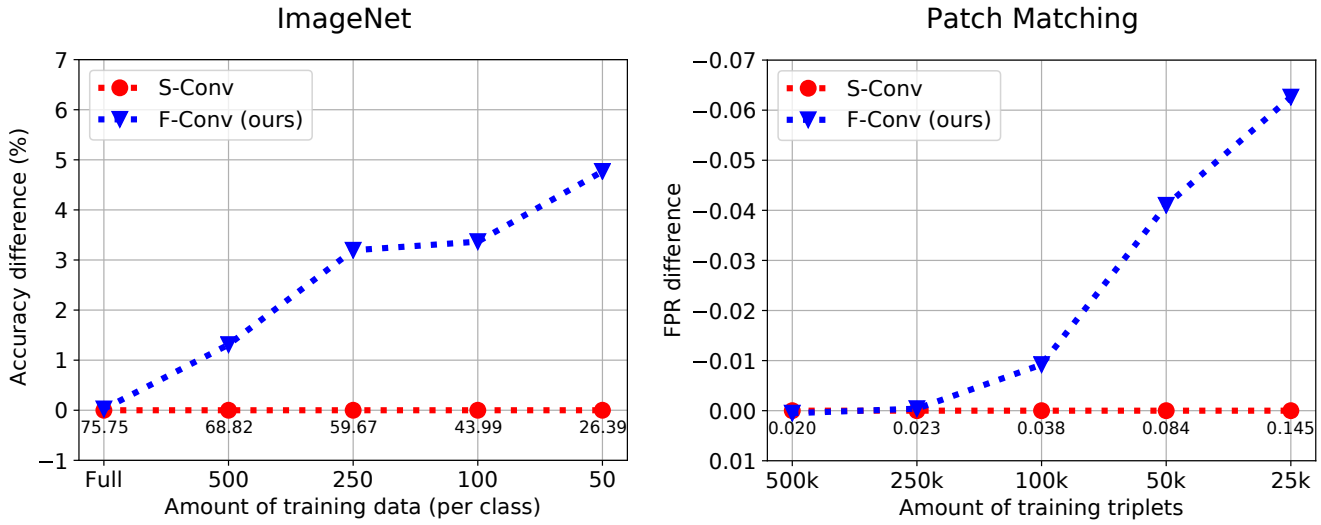

Figure 7. **Exp 4:** Data efficiency experiments. We reduce the amount of training data per class for the 1,000 classes of Imagenet for image classification and full Liberty, Notre Dame and Yosemite for patch matching. F-Conv outperforms S-Conv in both modality with smaller data size. (left) The Imagenet plot demonstrates the obtained accuracy difference when the number of data samples per class. The difference between F-Conv and S-Conv increases when the sample size decreases. (right) Correspondingly, F-Conv results in a performance increase for patch matching.

Diagonal Shift	S-Conv	F-Conv	S+BlurPool	F+BlurPool
RN18	79.43	82.74	81.96	**83.95**
RN34	82.06	85.66	83.73	**86.91**
RN50	86.36	87.92	87.50	**88.93**
RN101	86.95	87.78	88.22	**88.73**
Consistency	S-Conv	F-Conv	S+BlurPool	F+BlurPool
RN18	86.43	88.38	88.32	**90.03**
RN34	87.62	90.12	89.21	**91.53**
RN50	90.21	91.36	91.68	**92.75**
RN101	90.76	91.71	92.36	**92.86**

Table 2. **Exp 3:** Diagonal shift and consistency result for different Resnet architectures. S+BlurPool represents S-Convs with BlurPool Tri-3. Similarly, F+BlurPool corresponds the combination of F-Conv and BlurPool. In the most cases, F-Conv outperforms S-Conv and S+BlurPool (except for Resnet-101) in terms of diagonal shifting accuracy on testing set. Similar trend can be seen for consistency experiment, yet for Resnet-50 and Resnet-101, S+BlurPool has more consistent outputs. F+BlurPool achieves the highest score for both cases with all the architectures.

the S-Conv baseline. If we compare each Resnet architecture, the deepest model of the experiment, Resnet-101, improves the least, both for classification and consistency. Resnet-101 has more filters and parameters and it can learn many more varied filters than other models. By this, it can capture many variants of location of objects and thus the gap between methods for Resnet-101 are smaller.

4.4. Exp 4: Data efficiency

Does improving equivariance and invariance for the inductive convolutional prior lead to benefits for smaller data sets? We evaluate S-Conv and F-Conv with the same random initialization seed for two different settings: Image classification and image patch matching.

Image classification. We evaluate ResNet-50 classification accuracy for various training set sizes of the 1,000 classes in ImageNet. We vary the training set size as 50, 100, 250, 500, and all images per class.

Patch matching. We use HardNet [70] and use FPR (false positive rate) at 0.95 true positive recall as an evaluation metric (lower is better). We evaluate on 3 common patch matching datasets (Liberty, Notre Dame and Yosemite) from Brown dataset [9] where the model is trained on one set and tested on the other two sets. Hardnet uses triplets loss and we vary the training set size as 50k, 100k, 250k, 500k triplet patches. Each test set has 100k triplet patches.

Results are given in Fig. 7. For both image classification as for patch matching S-Conv and F-Conv perform similar for a large amount of training data. Yet, when reducing the number of training samples there is a clear improvement for F-Conv. For ImageNet with only 50 samples per class S-Conv scores 26.4% and F-Conv scores 31.1%, which is a relative improvement of 17.8%. For patch matching, S-Conv scores 0.145 and F-Conv 0.083 which is a relative improvement of 75%. Clearly, removing absolute location improves data efficiency.

	UCF101		HMDB51	
	Baseline (S-Conv)	Ours (F-Conv)	Baseline (S-Conv)	Ours (F-Conv)
RN-18	38.6	40.6	16.1	19.3
RN-34	37.0	46.9	15.2	18.3
RN-50	36.2	44.1	14.3	19.0

Table 3. **Exp 5:** Action recognition with 3D Resnet-18, 34 and 50 by using S-Conv and F-Conv methods. F-Conv outperforms S-Conv on UCF101 and HMDB51 datasets. S-Conv obtains its best result with the most shallow network, Resnet-18, however F-Conv still improves the results even the model becomes bigger.

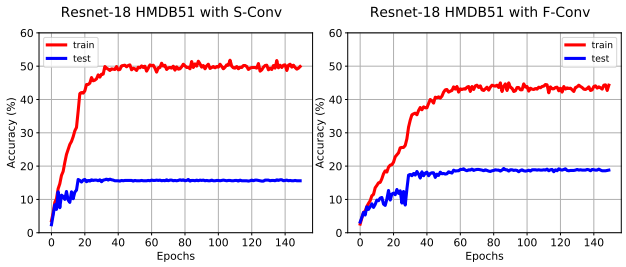

Figure 8. **Exp 5:** Training curves for 3D Resnet-18 S-Conv (left) and F-Conv (right) with HMDB51 dataset. Because the dataset is small, both models overfit. F-Conv achieves relatively 38.8% less overfitting than S-Conv.

4.5. Exp 5: Small datasets

Here we evaluate if the improved data efficiency generalize to two small datasets for action recognition. We select small sized data sets where training from scratch gives significantly worse results due to overfitting and the common practice is pre-training on a huge third party dataset. We compare the standard S-Conv with the proposed F-Conv where both methods are trained from scratch.

Action Recognition. We evaluate on two datasets: UCF101 [91] with 13k video clips from 101 action classes and HMDB51 [55] with 51 action classes and around 7k annotated video clips. We evaluate three 3D Resnet architectures [33], Resnet-18, 34 and 50.

We show results in Table 3. F-Conv models outperform the S-Conv models. Interestingly, in UCF101 experiment, the baseline performance decreased by 2.4% from Resnet-18 to Resnet-50; however, F-Convs still continue to improve the performance by 3.6 % for same architectures. According to Kensho et al [33] a 3D Resnet-18 overfits with UCF101 and HMDB51 which we confirm, yet F-Conv we overfit less than S-Conv. In Fig. 8, the difference between train and test of a 3D Resnet18 with S-Conv is 35.69%, however F-Conv has 25.7% overfitting. Similarly, S-Conv is relatively 41% more overfitted than F-Conv in Fig. 9. Consequently, both methods overfit due to the number of parameter and the lack of data.

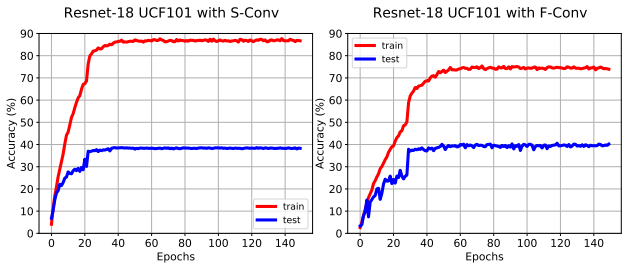

Figure 9. **Exp 5:** Training curves for 3D Resnet-18 S-Conv (left) and F-Conv (right) with UCF101 dataset. Both models overfit, but S-Conv has higher difference between training and testing results (49.1%). F-Conv has 34.8% of gap and thus overfits less.

5. Limitations and Conclusion

One limitation of our method is the extra computation required for padding. There is no extra cost of using circular padding instead of zero padding. For using F-Conv instead of S-Conv, the costs are similar to using S-Conv instead of V-Conv, and we found a Resnet-50 with F-Conv 15% slower to train on Imagenet.

Note that if absolute spatial location is truly discriminative between classes, it *should* be exploited [98], and not removed. For many internet images with a human photographer, there will be a location bias as humans tend to take pictures with the subject in the center, sofas on the bottom, and the sky up. The difficulty lies in having deep networks not exploit spurious location correlations due to lack of data. Addressing lack of data samples by sharing parameters over locations through added convolutions in deep networks is a wonderfully regularizer and we believe that convolutional layers should truly be translation equivariant.

To conclude, we show that in contrary to popular belief, convolutional layers can encode the absolute spatial location in an image. With the strong presence of the convolution operator in deep learning this insight is relevant to a broad audience. We analyzed how boundary effects allow for ignoring certain parts of the image. We evaluated existing networks and demonstrated that their large receptive field makes absolute spatial location coding available all over the image. We demonstrate that removing spatial location as a feature increases the stability to image shifts and improves the visual inductive prior of the convolution operator which leads to increased accuracy in the low-data regime and small datasets which we demonstrate for ImageNet image classification, image patch matching, and two video classification data sets.

References

[1] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM*

Transactions on audio, speech, and language processing, 22(10):1533–1545, 2014. 1

[2] Farzin Aghdasi and Rabab K Ward. Reduction of boundary artifacts in image restoration. *IEEE Transactions on Image Processing*, 5(4):611–618, 1996. 3

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016. 2

[4] Andr Araujo, Wade Norris, and Jack Sim. Computing receptive fields of convolutional neural networks. *Distill*, 2019. https://distill.pub/2019/computing-receptive-fields. 2

[5] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018. 2

[6] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 1269–1277, 2015. 2

[7] Alberto Bietti and Julien Mairal. Group invariance, stability to deformations, and complexity of deep convolutional representations. *The Journal of Machine Learning Research*, 20(1):876–924, 2019. 2

[8] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *ICLR*, 2019. 2, 5

[9] Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, Aug 2007. 7

[10] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013. 3

[11] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, pages http–openreview, 2014. 1

[12] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1

[13] Hsien-Tzu Cheng, Chun-Hung Chao, Jin-Dong Dong, Hao-Kai Wen, Tyng-Luh Liu, and Min Sun. Cube padding for weakly-supervised saliency prediction in 360 videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1420–1429, 2018. 3

[14] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103, 2014. 1

[15] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2392–2396. IEEE, 2017. 1

[16] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, and Andrea Vedaldi. Deep filter banks for texture recognition, description, and segmentation. *International Journal of Computer Vision*, 118(1):65–94, 2016. 2

[17] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999, 2016. 2

[18] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019. 2

[19] Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. In *ICML*, pages 1889–1898. JMLR. org, 2016. 2

[20] Cicero Dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, 2014. 1

[21] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015. 1

[22] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pages 1802–1811, 2019. 2

[23] Alhussein Fawzi and Pascal Frossard. Manitest: Are classifiers really invariant? In *British Machine Vision Conference (BMVC)*, number CONF, 2015. 2

[24] Alhussein Fawzi, Horst Samulowitz, Deepak Turaga, and Pascal Frossard. Adaptive data augmentation for image classification. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3688–3692. Ieee, 2016. 2

[25] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 1

[26] Robert Gens and Pedro M Domingos. Deep symmetry networks. In *Advances in neural information processing systems*, pages 2537–2545, 2014. 2, 3

[27] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[28] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *European conference on computer vision*, pages 392–407. Springer, 2014. 2

[29] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1

[30] Daniel A Griffith. The boundary value problem in spatial statistical analysis. *Journal of regional science*, 23(3):377–387, 1983. 2

[31] Daniel A Griffith and Carl G Amrhein. An evaluation of correction techniques for boundary effects in spatial statistical analysis: traditional methods. *Geographical Analysis*, 15(4):352–360, 1983. 2

[32] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3279–3286, 2015. 1, 2

[33] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 1, 8

[34] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 2

[35] 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. 1, 2, 5

[36] Xiangyu He, Zitao Mo, Qiang Chen, Anda Cheng, Peisong Wang, and Jian Cheng. Location-aware upsampling for semantic segmentation, 2019. 3

[37] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019. 6

[38] Joao F Henriques and Andrea Vedaldi. Warped convolutions: Efficient invariance to spatial transformations. In *ICML*, pages 1461–1469. JMLR. org, 2017. 2

[39] Alex Hernández-García and Peter König. Data augmentation instead of explicit regularization. *arXiv preprint arXiv:1806.03852*, 2018. 2

[40] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017. 1

[41] Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *International Conference on Machine Learning*, pages 2731–2741, 2019. 2

[42] Le Hou, Dimitris Samaras, Tahsin M Kurc, Yi Gao, James E Davis, and Joel H Saltz. Patch-based convolutional neural network for whole slide tissue image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2424–2433, 2016. 2

[43] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1, 2, 5

[44] Carlo Innamorati, Tobias Ritschel, Tim Weyrich, and Niloy J Mitra. Learning on the edge: Investigating boundary filters in cnns. *International Journal of Computer Vision*, pages 1–10. 3

[45] Md Amirul Islam, Sen Jia, and Neil DB Bruce. How much position information do convolutional neural networks encode? In *ICLR*, 2019. 2

[46] Jorn-Henrik Jacobsen, Jan van Gemert, Zhongyu Lou, and Arnold WM Smeulders. Structured receptive fields in cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2610–2619, 2016. 3

[47] Bernd Jahne. *Digital image processing*, volume 4. Springer Berlin, 2005. 2

[48] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th international conference on computer vision*, pages 2146–2153. IEEE, 2009. 3

[49] Can Kanbak, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Geometric robustness of deep networks: Analysis and improvement. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[50] Eric Kauderer-Abrams. Quantifying translation-invariance in convolutional neural networks. *arXiv preprint arXiv:1801.01450*, 2017. 2

[51] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018. 1

[52] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 1

[53] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, pages 2752–2760, 2018. 2

[54] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 2

[55] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011. 8

[56] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015. 1

[57] Dmitry Laptev, Nikolay Savinov, Joachim M Buhmann, and Marc Pollefeys. Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 289–297, 2016. 2

[58] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995. 1

[59] 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. 2

[60] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2

[61] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *ICLR*, 2013. 1, 2

[62] Guilin Liu, Kevin J Shih, Ting-Chun Wang, Fitsum A Reda, Karan Sapra, Zhiding Yu, Andrew Tao, and Bryan Catanzaro. Partial convolution based padding. *arXiv preprint arXiv:1811.11718*, 2018. 3

[63] Renting Liu and Jiaya Jia. Reducing boundary artifacts in image deconvolution. In *2008 15th IEEE International Conference on Image Processing*, pages 505–508. IEEE, 2008. 3

[64] 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. In *Advances in Neural Information Processing Systems*, pages 9605–9616, 2018. 3

[65] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1

[66] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2014. 1, 2

[67] Zixin Luo, Tianwei Shen, Lei Zhou, Siyu Zhu, Runze Zhang, Yao Yao, Tian Fang, and Long Quan. Geodesc: Learning local descriptors by integrating geometry constraints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 3

[68] Diego Marcos, Benjamin Kellenberger, Sylvain Lobry, and Devis Tuia. Scale equivariance in cnns with vector fields. *arXiv preprint arXiv:1807.11783*, 2018. 2

[69] Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation equivariant vector field networks. In *ICCV*, pages 5048–5057, 2017. 2

[70] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pages 4826–4837, 2017. 7

[71] Arun Mukundan, Giorgos Tolias, and Ondrej Chum. Explicit spatial encoding for deep local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9394–9403, 2019. 3

[72] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8837–8845, 2019. 3

[73] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 1

[74] Edouard Oyallon and Stéphane Mallat. Deep rototranslation scattering for object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2865–2873, 2015. 3

[75] Edouard Oyallon, Sergey Zagoruyko, Gabriel Huang, Nikos Komodakis, Simon Lacoste-Julien, Matthew B Blaschko, and Eugene Belilovsky. Scattering networks for hybrid representation learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 3

[76] Nikolaos Passalis and Anastasios Tefas. Learning bag-of-features pooling for deep convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5755–5763, 2017. 2

[77] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[78] Stanley J Reeves. Fast image restoration without boundary artifacts. *IEEE Transactions on image processing*, 14(10):1448–1453, 2005. 3

[79] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2

[80] Alexander Richard and Juergen Gall. A bag-of-words equivalent recurrent neural network for action recognition. *Computer Vision and Image Understanding*, 156:79–91, 2017. 2

[81] 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. 1

[82] Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, 2017. 1

[83] Andrew M Saxe, Pang Wei Koh, Zhenghao Chen, Maneesh Bhand, Bipin Suresh, and Andrew Y Ng. On random weights and unsupervised feature learning. In *ICML*, volume 2, page 6, 2011. 3

[84] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018. 1

[85] Stefan Schubert, Peer Neubert, Johannes Pöschmann, and Peter Pretzel. Circular convolutional neural networks for panoramic images and laser data. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 653–660. IEEE, 2019. 3

[86] Atharva Sharma, Xiuwen Liu, Xiaojun Yang, and Di Shi. A patch-based convolutional neural network for remote sensing image classification. *Neural Networks*, 95:19–28, 2017. 2

[87] Tao Shen, Yuyu Huang, and Zhijun Tong. Facebagnet: Bag-of-local-features model for multi-modal face anti-spoofing.

In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 2

[88] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 1

[89] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 2

[90] Kihyuk Sohn and Honglak Lee. Learning invariant representations with local transformations. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1339–1346. Omnipress, 2012. 2

[91] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. 8

[92] Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders. Scale-equivariant steerable networks. *arXiv preprint arXiv:1910.11093*, 2019. 2

[93] J Springenberg, Alexey Dosovitskiy, Thomas Brox, and M Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015. 1, 2

[94] Gilbert Strang and Truong Nguyen. *Wavelets and filter banks*. SIAM, 1996. 2

[95] 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. 1, 2

[96] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018. 3

[97] Gregor Urban, Krzysztof J Geras, Samira Ebrahimi Kahou, Ozlem Aslan, Shengjie Wang, Abdelrahman Mohamed, Matthai Philipose, Matt Richardson, and Rich Caruana. Do deep convolutional nets really need to be deep and convolutional? In *ICLR*, 2016. 3

[98] Jan C Van Gemert. Exploiting photographic style for category-level image classification by generalizing the spatial pyramid. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, pages 1–8, 2011. 8

[99] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised scale equivariant network for weakly supervised semantic segmentation. *arXiv preprint arXiv:1909.03714*, 2019. 2

[100] Zhenyi Wang and Olga Veksler. Location augmentation for cnn. *arXiv preprint arXiv:1807.07044*, 2018. 3

[101] Nicholas Watters, Loic Matthey, Christopher P Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. In *ICLR workshop*, 2019. 3

[102] Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2018. 2

[103] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5028–5037, 2017. 2, 3

[104] Daniel E Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. *arXiv preprint arXiv:1905.11697*, 2019. 2

[105] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 1, 2

[106] Yichong Xu, Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, and Zheng Zhang. Scale-invariant convolutional neural networks. *arXiv preprint arXiv:1411.6369*, 2014. 2

[107] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361, 2015. 1, 2

[108] Jure Zbontar, Yann LeCun, et al. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016. 2

[109] Richard Zhang. Making convolutional networks shift-invariant again. In *International Conference on Machine Learning*, pages 7324–7334, 2019. 2, 6

[110] Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Oriented response networks. In *CVPR*, pages 519–528, 2017. 2