Crop-Equivariant Convolutional Neural Networks for Self-Super-Resolution

Samuel W. Remedios[1,2] SAMUEL.REMEDIOS@JHU.EDU
[1] *Department of Computer Science, Johns Hopkins University, MD, USA*
[2] *Department of Radiology, National Institutes of Health Clinical Center, MD, USA*

Aaron Carass[3] AARON_CARASS@JHU.EDU
[3] *Department of Electrical Engineering, Johns Hopkins University, MD, USA*

Yuan-Ting Wu[4] YMW5168@PSU.EDU
[4] *Department of Neural and Behavioral Sciences, The Pennsylvania State University, PA, USA*

John A. Butman[2] JBUTMANA@CC.NIH.GOV

Michael Schär[5] MSCHAR3@JHU.EDU
[5] *Department of Radiology, Johns Hopkins University, MD, USA*

Dzung L. Pham[6] DZUNG.PHAM@NIH.GOV
[6] *Center for Neuroscience and Regenerative Medicine, Henry M. Jackson Foundation, MD, USA*

Yongsoo Kim[4] YUK17@PSU.EDU

Jerry L. Prince[3] PRINCE@JHU.EDU

Editors: Under Review for MIDL 2022

Abstract

Practical implementations of convolutional neural networks (CNNs) involve training and using models on the GPU. In order for this to work, memory must be allocated for not only the CNN's weights but also all intermediate feature maps, the size of which depends on the input image size. Medical image data can routinely exceed the available GPU RAM. Additionally, CNNs cannot naively stitch together cropped inputs and expect identical results compared to the scenario when CNNs run on the full image. In this work, we propose an architectural design which allows test-time patch evaluation with identical results to full-slice evaluation. We call this property *crop-equivariance* and show its equivalence to scenarios where the entire image is loaded into vRAM. We showcase our approach in a self-super-resolution task on test data where we can compare test-time patch evaluation to full slice evaluation as well as on light-sheet fluorescence microscopy images that are too large to fit into vRAM. An added benefit of our approach is dramatically reduced network size and train-/test-times.

Keywords: Convolutional neural networks, crop-equivariance, super-resolution

1. Introduction

Medical images often are too large to fit into GPU RAM (or vRAM). This affects training and testing; researchers make a trade-off between model size, batch size, and image size. At train-time this is remedied by training on patches—a contiguous smaller portion of the image(s). Patch-wise training alleviates the train-time vRAM burden; however at test-time, naively running on input patches is not equivalent to running on the entire image.

CNN architectures can be designed to satisfy certain mathematical properties to enhance their performance for specific tasks. For example, a network is shift-equivariant if spatially shifting the input equally shifts the output. Shift-equivariance is often desired for consistency in image-to-image translation, segmentation, and super-resolution (SR) tasks. Shift-invariance is desirable for classifications tasks. Mathematically, the definitions of shift-equivariance and shift-invariance for the function f can be stated as:

$$f(\texttt{Shift}(x)) = \texttt{Shift}(f(x)) \qquad \text{Shift-equivariance}$$
$$f(\texttt{Shift}(x)) = f(x) \qquad \text{Shift-invariance}$$

From Zhang (2019), we note that a CNN is shift-equivariant if we do not change the sizes of feature maps. We propose to extend shift-equivariance to crop-equivariance to permit a test-time patch evaluation with identical results to full-slice evaluation. For a function to be crop-equivariant it is necessarily shift-equivariant, which we will expand upon in Sec. 2.1.

Other types of equivariances have been explored for neural networks. Han et al. (2020) developed reflection-equivariant networks to take advantage of the left-right similarity that occurs in the brain, thalamus, and cerebellum. Rotation-equivariant (Ecker et al., 2019) CNNs were proposed as a means to model the human visual system. Scale-equivariant networks (Sosnovik et al., 2021) aim to capture and respect features across scales. However, to the best of our knowledge, crop-equivariant networks have not been previously explored.

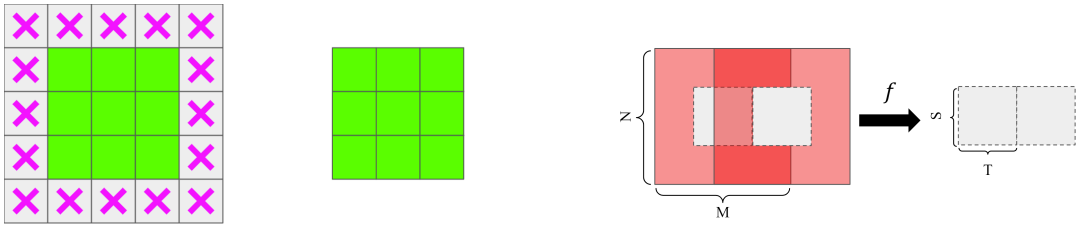
In this work we first make three contributions: 1) we define *crop-equivariance* as a stricter extension of shift-equivariance; 2) we propose and validate an architectural constraint that restricts the total receptive field of a CNN; 3) we develop a test-time patch extraction strategy. Both 2) and 3) are needed to achieve crop-equivariant CNNs, which produce identical results between test-time patches and full images. We then make a fourth contribution: the application of our architecture to the task of self-SR with validation on the ADNI phantom and evaluation on 50 T1-weighted brain MR images from the OASIS-3 dataset. A fifth contribution is to demonstrate our approach on light-sheet fluorescence microscopy images, which are too large to fit into vRAM. A consequence of our crop-equivariant CNNs is a dramatically reduced network size and train-/test-times.

2. Methods

2.1. Crop-equivariance

Without loss of generality we proceed with 2D definitions, understanding that this exposition readily generalizes to higher dimensions. For a discrete image $I[N, M]$ with coordinate support $\Omega_{NM} = \{(1,1),(2,1),\cdots,(N,1),\cdots,(N,M)\}$ and a model f, we define crop-equivariance as

$$f(\texttt{Crop}_{\boldsymbol{LK}}(I[N,M])) = \texttt{Crop}_{\boldsymbol{LK}}(f(I[N,M])). \tag{1}$$



(a) **(b)**

Figure 1: **(a)** On the left the output produced by a single kernel of size 3×3 with padding on an input of size 5×5 is of the same size as the input; tainted pixels are marked with a magenta '$\times$'. However on the right, the output produced by the same kernel but without padding cannot have tainted pixels. **(b)** Test-time patch extraction strategy: f uses valid convolutions, its output has smaller spatial size than its input. Patches overlap with a stride of $k \times k$ and are concatenated together.

Here, $\texttt{Crop}_{\boldsymbol{LK}} : \Omega_{NM} \to \Omega_{LK}$ is a function that reduces the discrete coordinate support of the image, with $\Omega_{LK} = \{(n_1, m_1), (n_1 + 1, m_1), \cdots, (n_2, m_2) | 1 \leq n_1 \leq n_2 \leq N, 1 \leq m_1 \leq m_2 \leq M\}$, $\boldsymbol{L} = (n_1, m_1)$ and $\boldsymbol{K} = (n_2, m_2)$. That is, the cropped image is a contiguous portion of equal or smaller size to its input. We note that crop-equivariant models f must also be shift-equivariant, since realizations of Ω_{LK} can occur at various shifts of one another.

When f is the convolution of a single kernel of size $k \times k$ with input $I[N, M]$, two scenarios commonly occur. First, if f pads its input to permit the kernel operation at corners and edges, the resulting image will have support Ω_{NM}. Alternatively, if f does not pad its input then the resulting image will have support Ω_{XY}, with $X = N - 2\lfloor \frac{k}{2} \rfloor$ and $Y = M - 2\lfloor \frac{k}{2} \rfloor$. The second case (where a convolution does not pad the input) is known as a "valid" convolution. For example, if $k = 3$ and $N = M = 5$, then in the first case f produces a 5×5 image where all pixels around the border use the padding as input, and in the second it produces a 3×3 image where none of the pixels use the padding as input. This is illustrated in Fig. 1(a). We consider any output pixels that incorporate information from padding as "*tainted*". Tainted pixels manifest as "seam" artifacts after aggregation (concatenation or weighted summation) of test-time patches. In such a situation, the result after test-time patch evaluation will differ from full image evaluation. Clearly, when padding is not used, no pixel can be tainted by definition. To maintain crop-equivariance, we must avoid or discard tainted pixels.

In the single-kernel case, we can simply remove all tainted pixels. For a cascade of kernels, we can calculate which pixels are tainted and then discard those. Thus, the number of tainted pixels in an image is a function of the total receptive field (Long et al., 2014; Araujo et al., 2019). In practice, if f is a deep CNN with zero-padding, it has been shown that the network learns to assign near-zero weights at kernel edges, yielding an empirically smaller receptive field than theoretical predicted (Zhou et al., 2015). However, there is no guarantee that the empirical receptive field optimizes to a particular value, and therefore there is no control over how many pixels are tainted by padding.

When f processes the image $I[N, M]$, no pixels are tainted except those that incorporate padded pixels. Precise removal of tainted pixels in padded networks requires layer-by-layer

adjustment; however, in networks without padding from layer to layer, no pixels are tainted. As f does not pad its input, the coordinate support is reduced. It is reduced by the receptive field of f; for a single-path CNN with layer-constant kernel size $k \times k$ and l layers, the reduction in the receptive field can be expressed as,

$$f : \mathbb{R}^{N \times M} \rightarrow \mathbb{R}^{S \times T} \quad \text{where} \quad S = N - 2l \left\lfloor \frac{k}{2} \right\rfloor \quad \text{and} \quad T = M - 2l \left\lfloor \frac{k}{2} \right\rfloor . \tag{2}$$

Equation 2 dictates our patch extraction strategy at test-time to guarantee slice-perfect test-time patch-wise evaluation. When performing conventional full-slice evaluation, the CNN kernels stride along the input uninterrupted. For our construction to be equivalent, we must draw patches at test-time that overlap in a particular way which allows concatenation of the resulting patches. We illustrate this in Fig 1(b). Patches of size $N \times M$ are drawn from the input image with a stride of $S \times T$, (as defined in Eq. 2) which is equivalent to the size of the output produced by our network. This allows the results to be concatenated together in order. We note that if we try this same strategy with a padded CNN, we do not have control over which pixels are tainted.

However, in many imaging applications, it is undesirable to reduce the size of the input image. To address this, we pre-pad the input image prior to test-time patch extraction such that $N_{\text{pad}} = N + 2l\lfloor \frac{k}{2} \rfloor$ and $M_{\text{pad}} = M + 2l\lfloor \frac{k}{2} \rfloor$. This does admit tainted pixels at the true boundaries of the input, but only at the boundaries and nowhere else. This same phenomenon occurs with full-slice evaluation as well.

We now have the elements of crop-equivariant CNNs. Specifically, we propose to design crop-equivariant CNNs by 1) omitting any resampling layers from the network and restricting all kernel strides to 1 for all layers; 2) removing all padding from all convolutional layers in the input; and 3) performing test-time patch extraction as illustrated in Fig. 1.

3. Experiments

We conducted two experiments to validate our method. First, we compared our crop-equivariant approach to the conventional method on a slice-vs-patch test-time evaluation to determine whether our method is correct. Then, we evaluated our approach on light-sheet fluorescence microscopy (LSFM) data, which is too large to fit into vRAM conventionally.

3.1. Validation: Slice vs. Patch Evaluation

To validate the correctness of our approach, we first consider the SR task of a magnetic resonance volume, using a similar internal training scheme to Zhao et al. (2021). Briefly, we create training data by drawing high-resolution (HR) 2D patches from the in-plane slices and simulate their low-resolution (LR) counterparts, then train a CNN to learn the mapping from LR to HR. The implementation here forgoes the anti-aliasing network in Zhao et al. (2021), learning both anti-aliasing and SR with one network.

We compare two architectures under this task: an EDSR-based network as in Zhao et al. (2021) and our proposed network, illustrated in Fig. 2. We will refer to these as "zero-padded" and "valid" networks respectively. The first network zero-pads the feature maps between convolutional layers while the second network does not, and only performs convolutions over "valid" portions of the input.

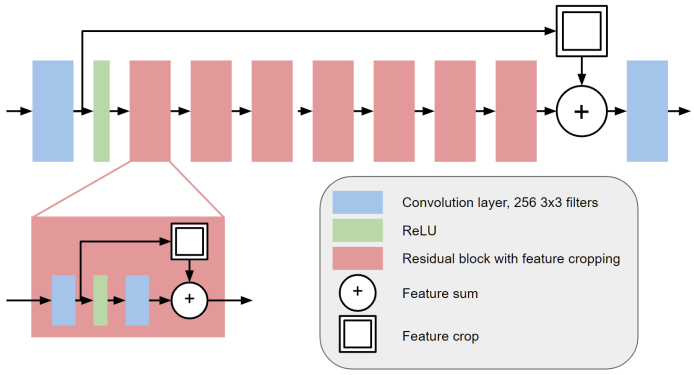


Figure 2: Our proposed architecture is 16 convolution layers, 14 of which comprise 7 residual blocks. Since valid convolutions are used, feature maps must be spatially cropped before the residual sum.

The summary of important differences in the networks is shown in Table 1. The valid network is necessarily much shallower than the zero-padded network due to the reduction in feature map size from layer to layer. Since the number of filters per layer is fixed, this reduces the number of trainable parameters considerably. Notably, the total theoretical receptive field of the zero-padded network is much larger than the proposed valid network; this is because the receptive field is a function of the number of layers from Eq. 2.

3.2. Data

We evaluate these two networks on MR images of the ADNI phantom with dimensions $256 \times 256 \times 180$ (as a sanity check) and on 50 T_1-weighted MR brain volumes with similar dimensions from the OASIS-3 dataset. The ADNI phantom was acquired at $0.8 \times 0.8 \times z$ mm^3 resolution, with $z \in 1, 2, 4$ and resampled to 1×1 mm^2 in-plane resolution, resulting in one isotropic volume and two LR volumes at $2\times$ and $4\times$ scaling.

The OASIS volumes were acquired at varying isotropic resolutions, and we individually downsampled each of them $2\times$, $3\times$, $4\times$, and $5\times$ to evaluate our SR method at each of these scales. Our downsampling process uses through-plane convolution with a Gaussian filter of FWHM equal to the downsampling factor followed by quintic b-spline interpolation to the corresponding scale.

Table 1: Network architecture differences used in the slice vs. patch experiment.

	Zero-padded	**Valid**
Number of Layers	66	16
Receptive field	133×133	33×33
Training patch size	32×32	64×64
Filters / layer	256	256
Trainable parameters	38,360,065	2,365,185

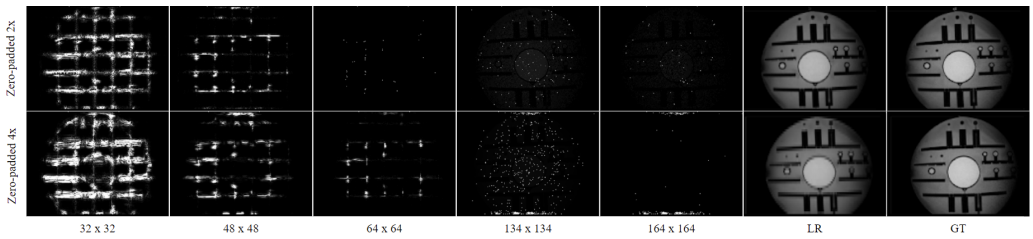


Figure 3: Difference maps comparing slice evaluation to patch evaluation for the the zero-padded network at 2× and 4× scale. LR and ground truth (GT) images shown on the right for reference. Our proposed network has zero difference between patch and slice evaluation, and are omitted. See Table 2 for numeric comparison.

3.3. Training Details

Our approach uses internal supervised learning; thus, the networks must be re-trained on each input volume. However, training is patch-based in both cases and we therefore train each model once and use the same set of weights for both slice and patch evaluation. Our training scheme is similar to Zhao et al. (2021) and creates training data from in-plane patches. To expand the training set, prior to patch extraction we augment in-plane slices with rotation, flips, and intensity shifts.

The training hyperparameter setups for both networks are identical: We used `AdamW` as our optimizer with a learning rate of 10^{-4} and an L1 + Sobel Edge loss. As internal supervision still permits a validation set, we set aside one rotation of the data as validation and define convergence as no improvement of the validation loss by 10^{-2} after 10 epochs. We train and validate for 320,000 and 640 patches, respectively, with a batch size of 32.

3.4. Results

ADNI Phantom: We compare slice vs. patch evaluation for the two network architectures on the ADNI phantom. Slice evaluation takes a full 256×256 slice as input, and patch evaluation takes $p \times p$ patches as input using the test-time patch extraction method illustrated in Fig. 1(b), with p varying as shown in Fig. 3. For both evaluation approaches, the same set of weights were used, since training is patch-wise regardless. We compared results at different test-time patch sizes and show the difference maps in Fig. 3. We see clear "seams" in the zero-padded network's results, especially at smaller patch sizes. Line profiles across the seams are shown in Fig. 4. This suggests that, although the network is trained on 32×32 patches, the effective receptive field is between 32×32 and 133×133. However, as apparent from Fig. 3, this differs at different scales (and potentially with different realizations after training); e.g., at 2× scale, the seams start to vanish at 64×64 patches, but at 4× scale they are still visible. In contrast, our proposed valid network is crop-equivariant, and has zero difference between slice and patch evaluation with 64×64 input patches, the same as its training patch size.

We report the mean squared error between slice and patch prediction in Table 2. We note that the intensity of error at the seams tends towards zero even for traditional zero-padded

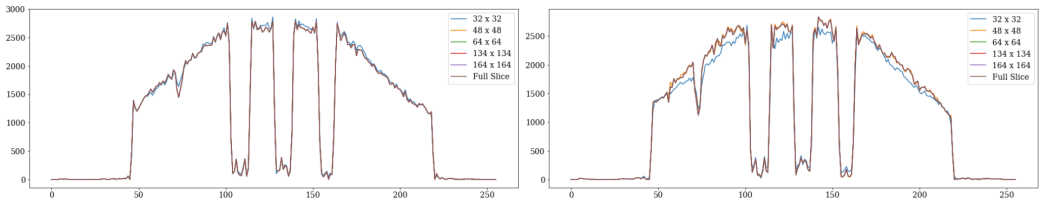


Figure 4: Line profiles from the zero-padded network at 2× (left) and 4× scale. As the test-time patch size increases, they approach the full-slice evaluation result. Our proposed network is identical for patch and slice evaluation and thus is omitted.

Table 2: Quantitative comparative results on the ADNI phantom for both architectures between patch and slice evaluation. While larger test-time patches tend towards zero for zero-padded networks, our proposed method (Valid) has exactly zero error. MSE denotes mean squared error.

		Scale 2×		**Scale** 4×	
	Patch Size	**MSE**	**Time (s)**	**MSE**	**Time (s)**
Zero-padded	Full slice	0	128	0	128
	32 × 32	275.138829	154	3049.724542	154
	48 × 48	0.137236	300	32.134624	300
	64 × 64	0.000008	526	0.20588	526
	134 × 134	0.000017	3736	0.000041	3736
	164 × 164	0.000006	6653	0.000022	6653
Valid	Full slice	0	45	0	45
	64 × 64	0	110	0	110

networks, yet the compute time increases due to redundant computation. Notably, naively stitching together patches at the training size leads to visible seams and the worst error with zero-padded networks, whereas our proposed approach yields identical results to full-slice evaluation.

OASIS-3: To evaluate whether the proposed method sacrifices performance, we trained and ran each method on 50 T_1-weighted MR volumes at four different scales. We used slice predictions from the zero-padded network. We show the comparisons at each scale in Fig. 5; we see that our proposed method does not sacrifice significant performance compared to the zero-padded formulation.

4. Super-Resolution of Light-Sheet Fluorescence Microscopy

Equipped with a crop-equivariant approach for SR, we ran our method on light-sheet fluorescence microscopy (LSFM) data. This data consists of two channels that capture lectin and neurons of a mouse brain, acquired at 0.411^2 μm^2 in-plane with a through-plane PSF FWHM

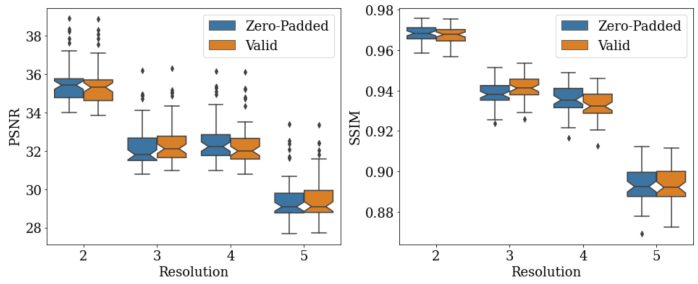


Figure 5: PSNR and SSIM at each scale for both the zero-padded and valid networks on the set of 50 OASIS-3 human brain volumes.

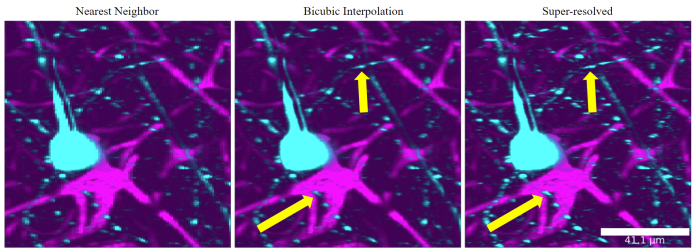


Figure 6: Zoomed region of the maximum intensity projection of a through-plane direction. Cyan: neuron; magenta: lectin-stained cerebrovasculature. The scale bar on the super-resolved image indicates 41.1 μm. The horizontal axis is the direction of SR. Yellow arrows indicate improved depictions of neurons.

of 2.2μm and through-plane step-size of 1μm. This results in a LR volume of pixel dimensions $2000 \times 2000 \times 2400$. After SR to isotropy, this volume has dimensions $2000 \times 2000 \times 5839$; evaluation on slices at this scale are possible only due our crop-equivariant network.

We trained our valid network as in the above experiment with slight adjustments: we sample a total of 500,000 patches, we also add Poisson and uniform noise to the LR training patches as augmentation, and the in-plane patch sampling strategy selects patches with high gradient values. Overall, the train time is about 45 mins on a Tesla V100, and the test-time with patches of size 64×64 is 66 hours on a single GPU. We compare the super-resolved maximum intensity projection to bicubic interpolation on a zoomed region in Fig. 6.

5. Discussion

In this work, we have introduced crop-equivariant CNNs to address the "seam" artifacts that occur when naively concatenating test-time results from conventional zero-padded architectures. We showed that our method produces identical results to full-slice prediction without sacrificing performance and showed qualitative results on LSFM data acquired at the micron scale. We hope this method of architecture design will allow deep networks to feasibly be run on larger images in other domains.

References

André Araujo, Wade Norris, and Jack Sim. Computing receptive fields of convolutional neural networks. *Distill*, 2019. doi: 10.23915/distill.00021.

Alexander S. Ecker, Fabian H. Sinz, Emmanouil Froudarakis, Paul G. Fahey, Santiago A. Cadena, Edgar Y. Walker, Erick Cobos, Jacob Reimer, Andreas S. Tolias, and Matthias Bethge. A rotation-equivariant convolutional neural network model of primary visual cortex. In *International Conference on Learning Representations*, 2019.

Shuo Han, Jerry L. Prince, and Aaron Carass. Reflection-equivariant convolutional neural networks improve segmentation over reflection augmentation. In Ivana Išgum and Bennett A. Landman, editors, *Medical Imaging 2020: Image Processing*, volume 11313, pages 806 – 813. International Society for Optics and Photonics, SPIE, 2020. doi: 10.1117/12.2549399.

Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Ivan Sosnovik, Artem Moskalev, and Arnold Smeulders. Scale equivariance improves siamese tracking. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2764–2773, 2021. doi: 10.1109/WACV48630.2021.00281.

Richard Zhang. Making convolutional networks shift-invariant again. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7324–7334. PMLR, 09–15 Jun 2019.

Can Zhao, Blake E. Dewey, Dzung L. Pham, Peter A. Calabresi, Daniel S. Reich, and Jerry L. Prince. Smore: A self-supervised anti-aliasing and super-resolution algorithm for mri using deep learning. *IEEE Transactions on Medical Imaging*, 40(3):805–817, 2021. doi: 10.1109/TMI.2020.3037187.

Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015.