Self-supervised Feature Learning for Semantic Segmentation of Overhead Imagery
—Supplementary Materials—

Abstract

The supplementary material contains (1) Additional architecture details for inpainting, coach, and segmentation networks, (2) Additional details of datasets used in experiments, (3) Qualitative examples from datasets, and (4) additional qualitative results.

1 Architecture Details

AlexNet architecute for Baselines For pre-training with self-supervised methods Context Prediction [3], Context Encoder [8], and Split-Brain Autoencoder, we use a common AlexNet [6] architecture as an encoder till Conv5 (see Table 1) and add `BatchNorm` layer after each convolution layer (similar to [10]).

Following Doersch *et al.*[3], we train the Context Prediction network with 2 patches of size 96 × 96 from spatial grid configuration. The patches are spatially separated by 16 pixels and the locations are further randomly jitter by 7 pixels. We replace `pool5` layer with `AvgPool` layer. This is then followed by `fc6 Linear` layer. Features of both patches are concatenated and fed to `fc7` layer which predicts the relative spatial location.

We use the similar architecture for Context Encoder described by [8] i.e., AlexNet [6] till `pool5` as encoder followed by channel wise fully connected and five `Deconvolution` layers. We randomly erase 16 patches of size 16 × 16 from input image (227 × 227) and the network tries to reconstruct the image.

For the Split-Brain Autoencoder, we follow the training procedure proposed by Zhang *et al.*[10] and train 2 disjoint networks halved along the channel dimensions. Table 1 describes the network details for one half i.e. predicting `ab` channel from `L` channel. We use the same architecture for the second network and reverse the input and output channels. Mean squared loss is calculated with respect to heavily downscaled ground truth of size 12 × 12.

We adapt and fine-tune the pre-trained networks for semantic segmentation task using `FCN` [7]. We use `SGD` optimizer to train the network for 100 epochs and step learning rate starting at 0.001, step size 0.1, momentum 0.9 and weight decay of 0.0005.

Coach Network We use ResNet-18 as backbone network for the coach network. The input to the network is mean subtracted. The score regressor module consists of three 1×1 convolutional layers for prediction of mean, standard deviation, and the final score (see Figure 1). Reparameterization [5] is used to inject noise into score prediction. We use 100 filters for predicting mean and standard deviation, whereas, a single filter to predict the final score. We remove `maxpool` from ResNet-18 to obtain prediction map of resolution 8×8 or downsampling factor of $\frac{1}{16}$. This is followed by point-wise sigmoid function during training of coach model and step-function during training of inpainting network. For input of size 128×128, each pixel in the output score corresponds to 16×16 region in the input. We use 16× nearest neighbor upsampling to obtain mask at input resolution. There are 11.28 Million parameters in this network.

* Equal Contributions.

Code available at: https://github.com/suriyasingh/Self-supervision-for-segmenting-overhead-imagery

Layer	**Context Encoder**						**SplitBrain Autoencoder**					
	X	C	K	S	P	D	X	C	K	S	P	D
data	227	3	–	–	–	–	180	1	–	–	–	–
conv1	56	96	11	4	2	1	45	48	11	4	5	1
pool1	27	96	3	2	0	1	23	48	3	2	1	1
conv2	27	256	5	1	2	1	23	128	5	1	2	1
pool2	13	256	3	2	0	1	12	128	3	2	1	1
conv3	13	384	3	1	1	1	12	192	3	1	1	1
conv4	13	384	3	1	1	1	12	192	3	1	1	1
conv5	13	256	3	1	1	1	12	128	3	1	1	1
pool5	6	256	3	2	0	1	12	128	3	1	1	1

Table 1: AlexNet architecture used as encoder network in all baseline experiments. X : input spatial resolution for the layer, C : number of channels/filters in the layer, K : convolution or pooling kernel size, S : stride, P : padding, and D : kernel dilation.

Inpainting Network We use ResNet-18 as backbone network for the inpainting network as well followed by a series of 5 deconvolution layers, each upsample the feature maps by `2x` (see Figure 1). Inputs and outputs are both of size 128 × 128. The input to this network is also mean subtracted. Target are normalized to [1, -1] with the dataset's mean and standard deviation and point-wise `tanh` function is applied to the output of the decoder network. The total number of parameters in this network is 18.16 Million.

Segmentation Networks We adapt the encoder-decoder inpainting network for semantic segmentation by attaching pixel-wise 1 × 1 convolutional classifiers at each deconvolution layers (see Figure 1). The input is of size 256 × 256 and is also mean subtracted. There are 18.164 Million parameters in this network.

2 Additional Details for Datasets

Potsdam [4] This dataset is used for scene parsing of the Potsdam city. Pixel-level annotations are provided for 6 classes: `impervious surface`, `building`, `tree`, `low vegetation`, `car`, and `background`. We create crops of size 600 × 600 with a stride of 200 × 200 for 20 training images and non-overlapping stride for 4 validation images. Post-processing artifacts are present in significant number of images.

SpaceNet [9] This dataset is used for road network estimation in four cities: `Paris`, `Las Vegas`, `Shanghai`, and `Khartoum`. The annotations are provided in the form of line-strings corresponding to the mid-line of a road. We obtain the binary masks by computing distance transform with respect to the mid-line, apply a Gaussian kernel of standard deviation of 15 on the distance transform outputs to obtain a heatmap, and then threshold the heatmap at 0.4. This results in foreground road masks of roughly 12 meters. We create crops of 650 × 650 with a stride of 250 × 250 for 2000 training images and non-overlapping stride for 567 validation images. The images are very diverse and consists of motorway, primary highway, secondary highway, tertiary highway, residential road, cart track, and unclassified roads. A significant number of roads have not been labeled.

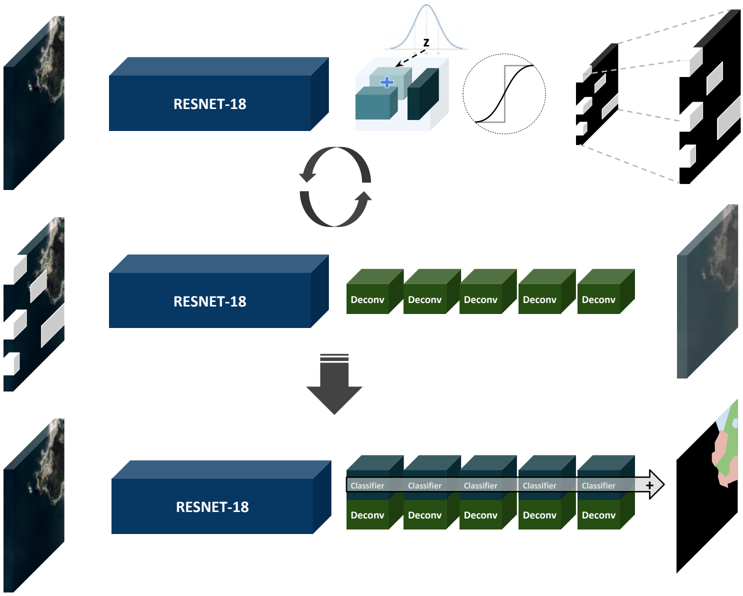


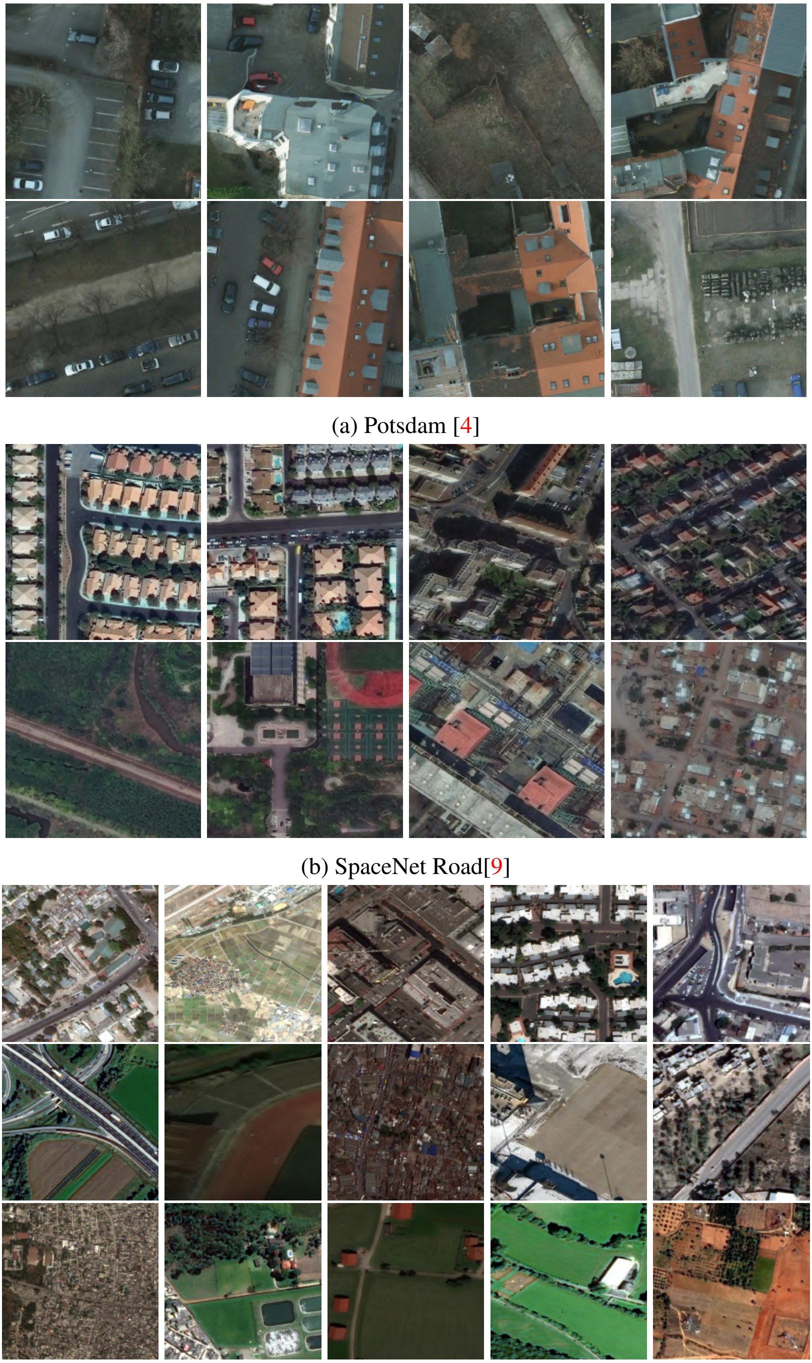
Figure 1: Architectures of coach network (top), inpainting network (middle), and segmentation network (bottom).

DeepGlobe Lands [1] This dataset is used for land cover estimation. Pixel-level annotations are provided for 7 classes: `urban`, `agriculture`, `range land`, `forest land`, `water`, `barren land`, and `unknown` (not used for evaluation). We create crops of 612 × 612 with stride of 228 × 228 for training images and non-overlapping stride for validation images.

DeepGlobe Roads [1] This dataset is used for road network estimation. Pixel-level annotations are provided for `road` and `background` classes. We create crops of 512 × 512 with stride of 256 × 256 for training images and non-overlapping stride for validation images.

Functional Map of the World [2] We use only the images from this dataset to study the feature quality learned with respect to the number of unlabeled examples. We use only the `train` split in our experiments. The images are taken from all over the world and significant diversity in terms of geography, terrain, weather condition, illumination exist in the images. We resize the images preserving the original aspect-ratio such that the minimum image dimension becomes 1024 pixels. We then create crops of 512 × 512 with non-overlapping stride for training as well as validation images.

3 Additional Qualitative Results



(a) Potsdam [4]

(b) SpaceNet Road[9]

(c) Functional Map of the World (fMoW) [2]

Figure 2: Qualitative examples of datasets used for experiments.

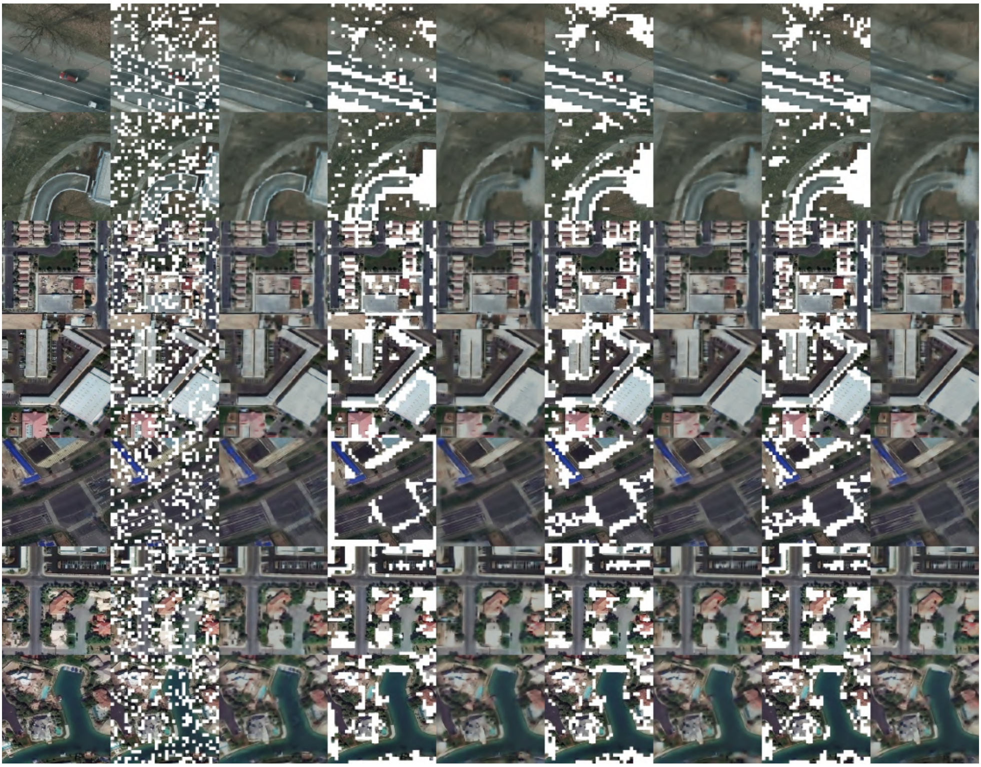


Figure 3: Coach model predicts an increasingly difficult masks for semantic inpainting. For each row, from left to right: Input image (512 × 512) for the coach network, masks predicted at iterations 0, 1, 6, and 8 with corresponding inpainting output. Note that at iteration 0 the coach predicts random masks.

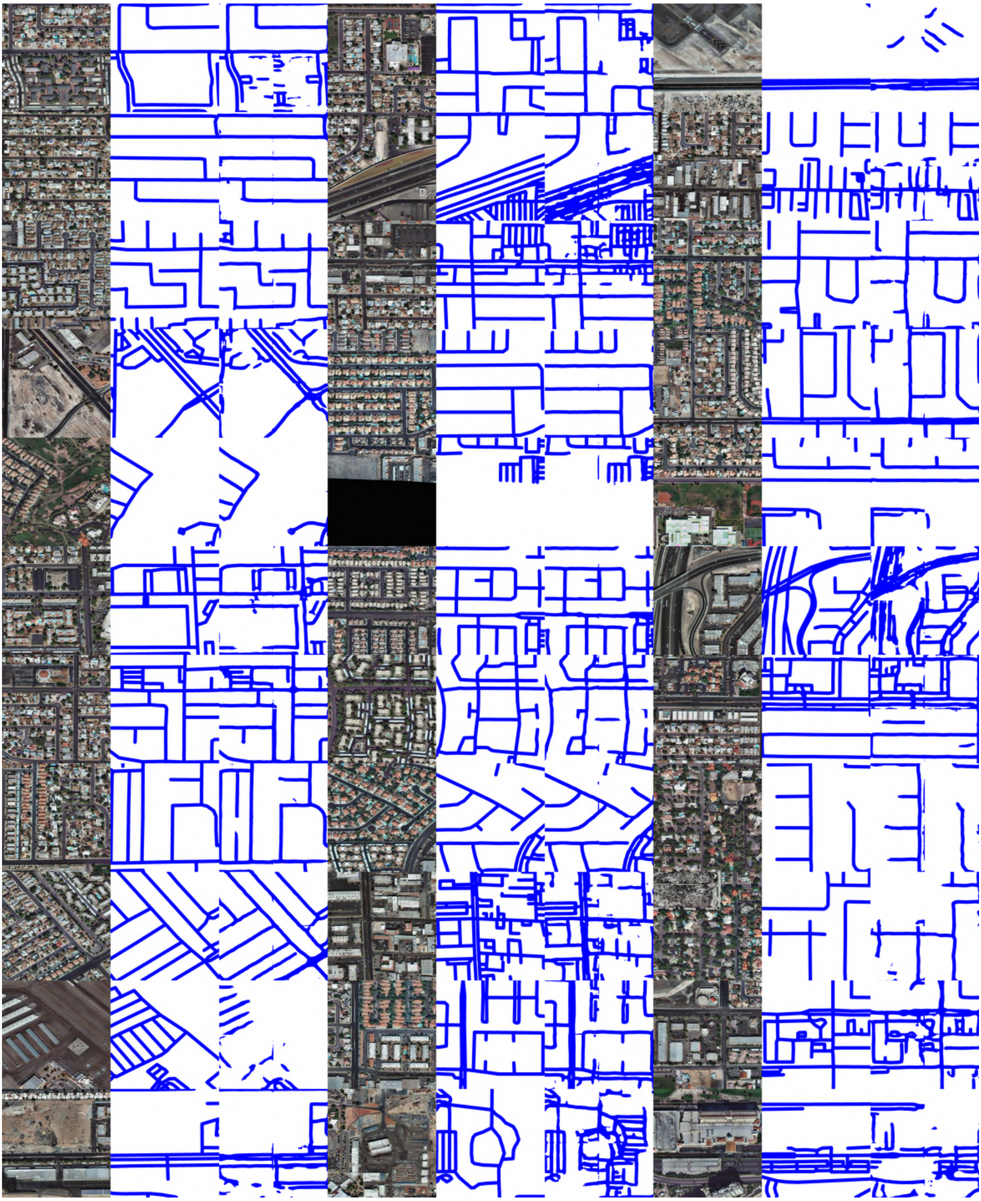
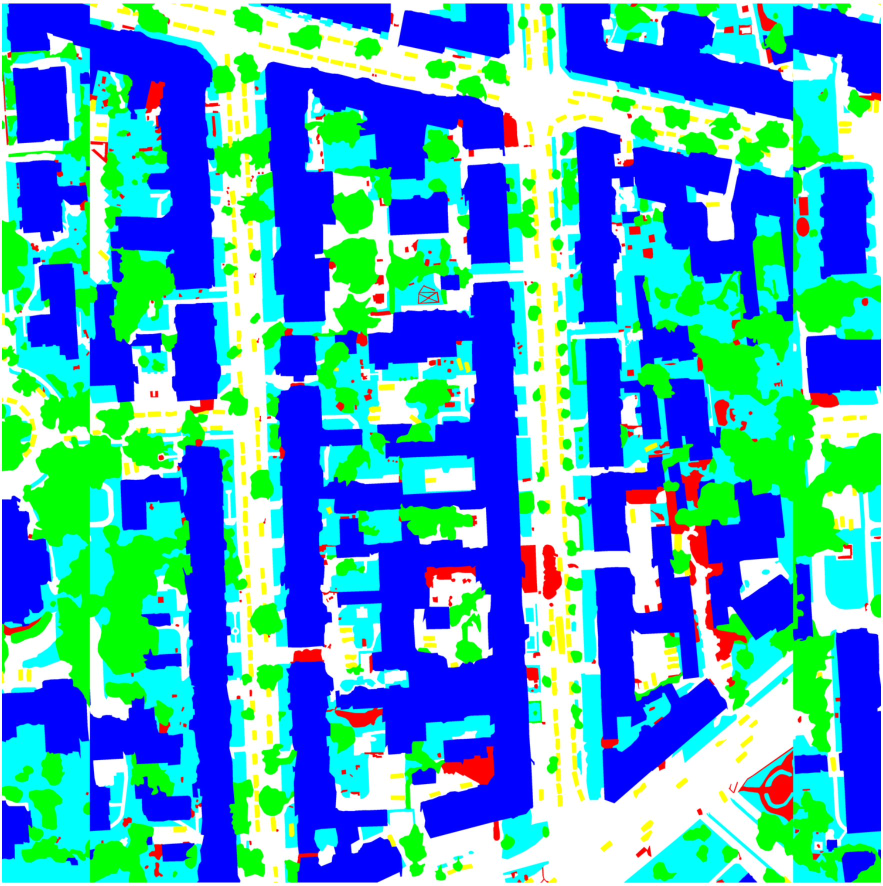


Figure 4: Road network extraction. Input image (left), ground truth segmentation map (middle), and predicted segmentation with coach training and 10% labeled data used for fine-tuning (right).



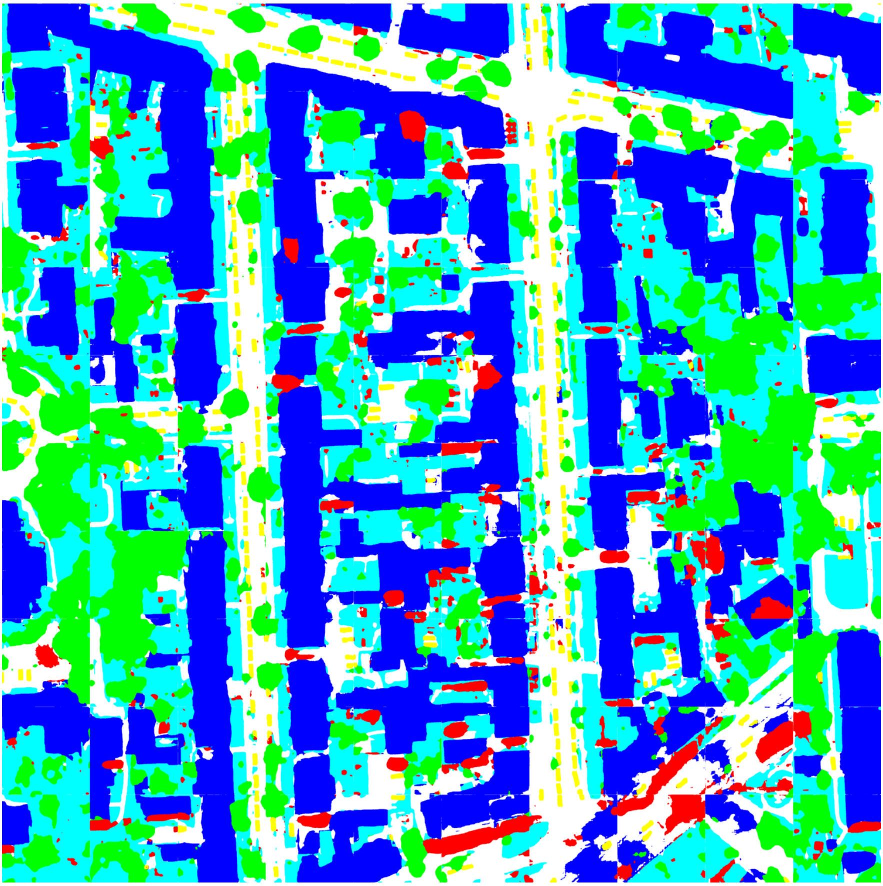


Figure 4: Parsing of an unseen region of Potsdam city. Input image (top), ground truth segmentation map (middle), and predicted segmentation with coach training and 10% labeled data used for fine-tuning (bottom).

References

[1] Deepglobe challenge. http://deepglobe.org/challenge.html.

[2] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *CoRR abs/1711.07846*, 2017.

[3] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[4] ISPR. Potsdam 2d semantic labeling contest. http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html.

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[7] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[8] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[9] SpaceNet. Spacenet on amazon web services (aws). https://spacenetchallenge.github.io/datasets/datasetHomePage.html, 2017.

[10] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017.