Acknowledgments

We thank Dídac Surís, Chengzhi Mao, Mia Chiquier, and Abby Lu for helpful feedback. This research is based on work partially supported by NSF CRII Award #1850069, the DARPA SAIL-ON program under PTE Federal Award No. W911NF2020009, and an Amazon Research Gift. We thank NVIDIA for GPU donations. Part of this work was performed while being the recipient of a Belgian American Educational Foundation Fellowship to BVH. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Government.

Supplementary material

Code is available at https://github.com/basilevh/dissecting-image-crops.

A. Dataset constraints, collection, and description

A.1 Lack of cropping

As a starting point, any sufficiently large collection of natural photos suffices. In order to simulate scenarios where a user only has access to pixels but not the metadata (which commonly happens when downloading photos from *e.g.* social media), no labels are needed. Training and testing data can be retrieved 'for free' by extracting patches and thumbnails from any dataset consisting of real-world images, where the only important constraint is the lack of tampering. However, it turns out that cropping, as well as various other kinds of 'soft tampering', is a natural part of the digital editing process. Because these operations are mostly harmless and probably happen more often than we realize, it becomes almost impossible to know to what extent a given database really is unedited.

A.2 Sufficiently high resolution

Acknowledging the fact that the dataset might be noisy to some degree, we proceed with adding a resolution constraint. Image datasets for deep learning are often down-scaled such that the maximal dimension lies around 500 to 1,000 pixels[2], presumably because the benefit of an even finer level of detail for recognizing object semantics rarely outweighs the extra computational cost. However, in order to better pick up lens flaws that are typically exhibited in subtle pixel-level features, we prefer to keep the resolution higher and closer to the original photo. This matches the observation in image forensics that resizing should be avoided because it tends to damage high-frequency details [38]. We decided to settle for (*i.e.* download images with) a maximal dimension of 2,048 pixels for each sample, which is deemed high enough to detect optical imperfections, but also low enough to avoid exceeding realistic dimensions of photos that may be shared online.

[2]For example, every sample in Open Images V5 [29] has at most 1,024 pixels on its longest side.

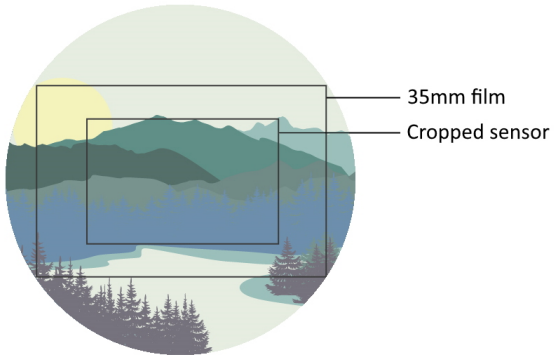


Figure 9: Comparison of a full-frame sensor versus a crop sensor with respect to the lens circle. (Adjusted and reprinted from [56] with permission.)

A.3 Inter-device variation considerations

Every lens and sensor is different, and this variation in standards might make what exactly constitutes a 'crop' less precise. For example, if a full-frame lens is coupled with a crop sensor (*i.e.* the film frame width is less than 35mm) as in Figure 9, every resulting picture can be thought of as inherently cropped because the light captured by the sensor does not fully cover the lens circle. Mobile phones have an especially large crop factor, since their sensors are typically much smaller than those used in professional DSLR camera systems. In fact, there is a vast number of possible configurations, and trying to take all of them into account would become impracticable. We thus clear confusion by defining a 'cropped image' to be any deviation from what was originally captured by the imaging sensor at the time of shooting. Since our method is camera make and model-blind, we rely on the learning-based approach to discover modal values within this combinatorial space of configurations in the dataset, such that our network will learn to take the diversity among devices and settings into account automatically.

A.4 Scraping and dataset bias

We scraped Flickr by querying the API with 10,000 different search terms and downloading up to 500 photos for every tag. The keywords were gathered from an online

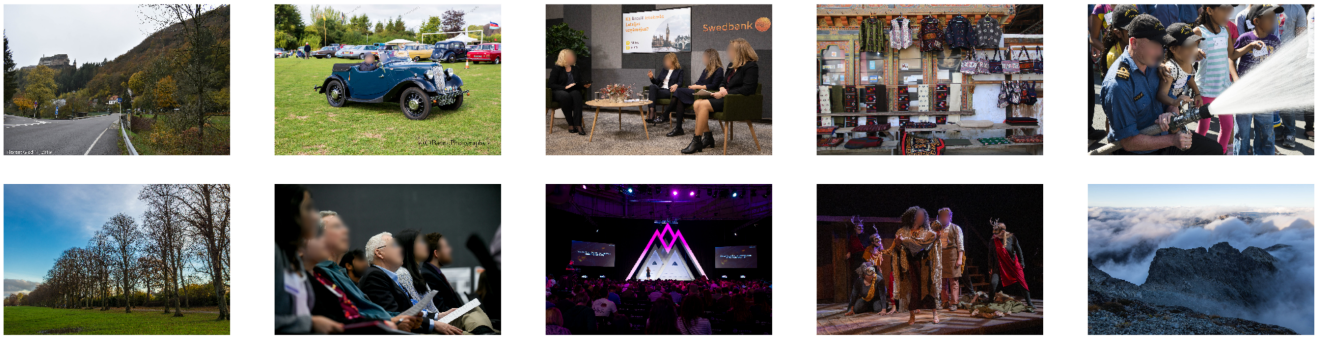


Figure 10: Random examples of the Flickr dataset. (Faced blurred for privacy protection.)

list of the 10,000 most commonly used words in English, which was in turn generated by performing N-gram frequency analysis on the Google Trillion Word Corpus [18, 25]. The resulting database has around 1.3 million images, which would have been 5 million if the search results did not overlap due to many entries having multiple tags. Note that Flickr seems to be biased toward photos (1) depicting persons, (2) of somewhat professional quality, and (3) taken using expensive cameras, but we view neither of these aspects as a drawback considering the relevance of our project to photography patterns and photojournalism.

A.5 Aspect ratio

Mixing training examples with different aspect ratios together also changes the shapes of the grid cells of F_{patch}, which should clearly be avoided. Otherwise, patches that have the same absolute position with respect to the lens circle, might be assigned different labels depending on the aspect ratio of the sensor within said lens circle. Most digital camera systems have a sensor size of 36mm×24mm, corresponding to an aspect ratio of 1.5. We therefore fix the aspect ratio to 1.5 and enforce landscape-only photos (by rotating portrait images either left or right) to further enhance consistency, which shrinks the pool of files meeting all discussed criteria down to 700,000 files.

A.6 Dataset split

Lastly, we perform a 3-way train / validation / test set split distributed as 90% / 5% / 5%. A few samples of the test set are shown in Figure 10.

B. Shortcut mitigation

Convolutional neural networks have been shown to be surprisingly adept at finding and leveraging often irrelevant shortcuts [15, 43]. Here, we present our approach to ensure that the models learn useful features.

B.1 Image patch extraction

Patches are extracted from the centers of a regularly sized 4×4 grid within every image (cropped or not), but we also apply random jittering of ± 8 pixels in both dimensions. This way, we discourage F_{patch} from learning low-level image processing-related shortcuts, for example JPEG block artefact alignment.

B.2 Resizing global images

Since F_{global} uses a downscaled variant of the incoming image with fixed dimensionality 224×149, but cropping an image also changes its raw dimensions, we were obliged to employ some tricks in order to prevent the model from learning glitches that are unrelated to physical imaging aberrations, notably resampling factor detection. Resampling shortcuts have occurred in various previous works [43], and are typically an undesired factor. For example, a neural network is able to trivially distinguish images that have been downsized starting from 2048×1365 as opposed to starting from 1536×1024 based on pixel-level resampling artefacts, even if the interpolation method is randomized [43]. To work around this issue, we perform random resizing in multiple stages to make the original dimensions nearly impossible to recover, without noticeably damaging the image contents.

Given the potentially cropped source image of size $W \times H$, we first resize 3 times to a random $W' \times H'$ where W' is uniformly distributed in $[1024, 2048]$, and $H'|W'$ is conditionally uniformly distributed in $[0.8W'/A, 1.2W'/A]$, with A the aspect ratio. Note that the interpolation method itself is also random, and is chosen from one of {NEAREST, LINEAR, AREA, CUBIC, LANCZOS4} as provided by the OpenCV library [1]. Finally, the whole image is downscaled to 224×149, and from now on it should be nearly impossible to tell what its original resolution was.

Indeed, if we replace the cropping operation with a rescaling to the same dimensions that the cropped image

would otherwise have, the accuracy of our global model drops to chance (50%). This suggests that only altered image contents play a role, while input resolution does not anymore.

Note that the way in which patches are extracted remains unaltered by this procedure; only thumbnails must be treated to ensure that F_{global} predominantly looks at semantically meaningful content.

B.3 Joint model

Another, more sophisticated shortcut arose which occurs only when the model has access to both patches and thumbnails simultaneously. Even if the original dimensions of a global image cannot be inferred, the integrated network could still learn to measure how 'large' the patches are in comparison to the thumbnail, since they are extracted from a 'smaller' image if the input is cropped. To alleviate this issue, we perform an extra random resizing step *before* extracting patches but *after* cropping, where the width is uniformly distributed in $[1024, 2048]$ and the height is chosen proportionally such that the aspect ratio is retained. This guarantees that the fraction of the thumbnail that is being covered by patches loses its predictive power, discouraging G from trying to exploit low-level correlations among the outputs produced by F_{patch} and F_{global}. This approach serves the additional purpose of enforcing our ignorance about both the crop rectangle and the sequence of resizes that images at test time could have undergone; hence, during our evaluations, we also randomize input resolutions the same way.

B.4 Patch labels and intra-batch interaction

We observed a peculiar effect when all the examples within a minibatch have the same ground truth label for absolute patch localization. Specifically, when all patches belonging to the same position class were forwarded through the network, an unnaturally high accuracy could be achieved during training, but not during validation. This does not occur when the batch size is just 1 instead of 64, implying that there exists an architectural feature of the neural network that enables cross-example interaction. Although we have not studied this aspect systematically, we speculate that it may be due to the BatchNorm2D layers contained in a ResNet [21], whereby the mutual information across different examples within every minibatch is somehow leaked and exploited. To counter this shortcut, we cyclically shift image patches across minibatches in order to ensure that every minibatch contains a uniform distribution of all 16 labels, rather than all of them having the same class. The mutual information among examples within a minibatch is therefore minimized, and the peculiar overfitting effect disappeared, as evidenced by the results becoming independent of batch size.

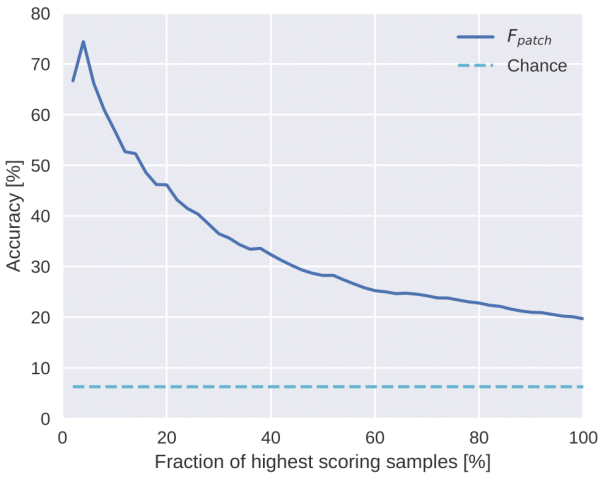
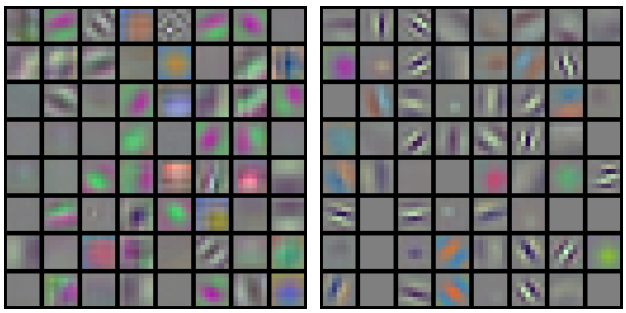


Figure 11: **Sample selectivity versus patch localization performance.** The accuracy improves significantly once we discard more and more predictions that F_{patch} is uncertain about.



(a) F_{patch} (ResNet-18). (b) ImageNet-trained ResNet-34.

Figure 12: **First convolutional layer filter visualization.** At the lowest level, the absolute patch localization model is clearly more sensitive to alternations between green and magenta (*i.e.* lack of green) pixel values in various directions, as compared to a vanilla ImageNet-trained neural network.

C. Patch localization accuracy versus confidence

Figure 11 plots the accuracy of F_{patch} as a function of the response rate, where moving to the left on the horizontal axis means that an increasingly smaller fraction of only the patches with the highest scores are considered. This supports the earlier claim that the maximum value in the output distribution correlates positively with the correctness of the pretext model.

D. Convolutional filter visualization

We display and compare the values of the convolution operations applied by the very first layers of both F_{patch}

Model	Color	Grayscale	Chance
F_{patch} (patch loc.)	21%	15%	6%
Joint (crop det.)	86%	81%	50%
Global (crop det.)	79%	78%	50%
Patch (crop det.)	77%	72%	50%

Table 2: **Accuracies with or without color.** Removing all color information on the test set decreases the model's performance, but only considerably so when a model relies on patches.

and a regular ImageNet classifier in Figure 12. These visualizations suggest that the network is particularly sensitive to green transverse chromatic aberration.

E. Additional experiments for lens-related clues

E.1 Effect of red and blue chromatic aberration

As shown in Figures 13a and 13b, the patch localization accuracy plots appear horizontally flipped with respect to Figure 5a. This indicates that the modal value of purple fringing in our dataset corresponds to the green channel being scaled toward one preferred direction more often than in the other direction. (Inward green TCA is visually the same as a combination of outward red and blue TCA.)

E.2 Effect of color saturation and grayscale

In order to quantify the significance of color information in general beyond just chromatic aberration, it may be instructive to control the saturation of the test set. A saturation factor of 0% is equivalent to grayscale imagery, 100% is identity, and larger numbers represent exaggerated colors. The result is shown in Figure 13c. This feature does not depend on the location of a patch, therefore it is not unexpected that the best performance corresponds with untouched images. Any other value simply moves the images away from the expected distribution.

Table 2 also compares the performance of the model when tested on grayscale and regular color images. Although color information clearly constitutes a respectable gain to the network's correctness relative to chance levels, there is a large residual gap that does not rely on color. Apart from vignetting, we hypothesize this is mostly related to photography patterns and object priors, which we discussed in Section 5.3. Moreover, the only model that is likely unable to perceive lens aberrations in the first place (*global*) seems to care the least about color information, suggesting that the object priors involved in revealing crops can be learned with minimal dependence on color.

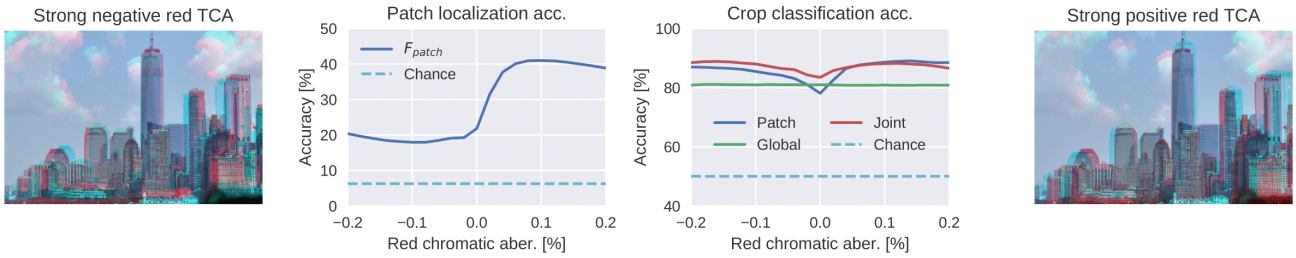
E.3 Effect of radial lens distortion

Pincushion or barrel distortion, illustrated left and right respectively in Figure 13d, arises from the fact that the magnification of a scene through a lens does not stay constant across the image plane, but depends on the radius $r = \sqrt{x^2 + y^2}$ from the optical center [33]. We replicate this distortion by applying a geometric coordinate transformation with a simple square law that scales every destination pixel (x_d, y_d) relative to its source (x_s, y_s) as follows:

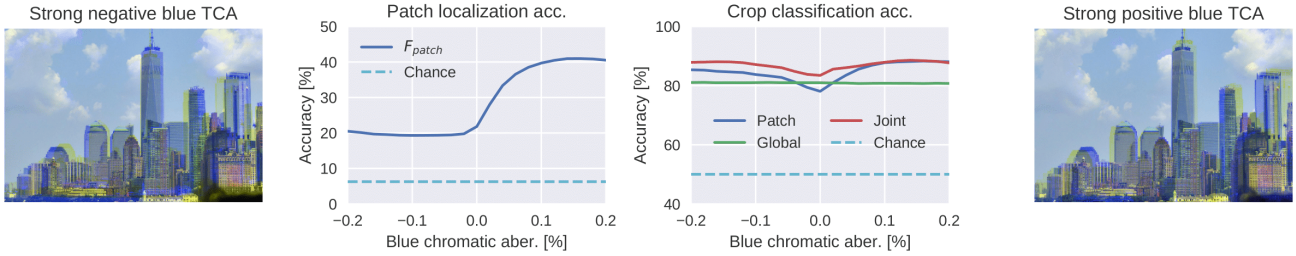
$$d = 1 + k_1 r^2 \tag{7}$$

$$(x_d, y_d) = (dx_s, dy_s) \tag{8}$$

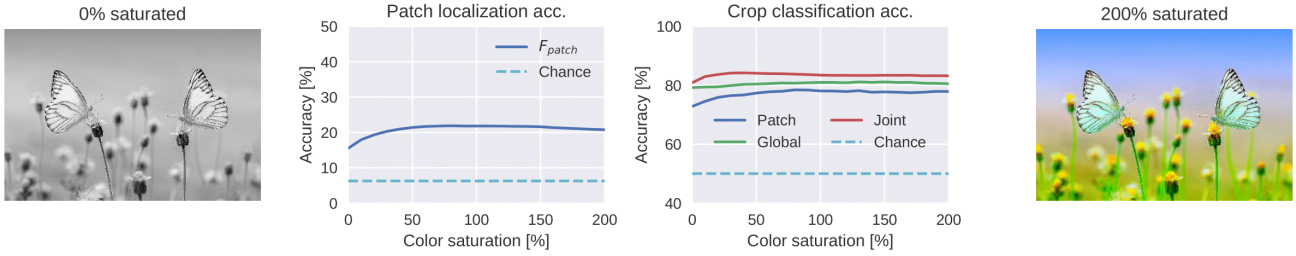
Figure 13d shows the effect of inflating lens distortion on the test set according to Equation (7-8).



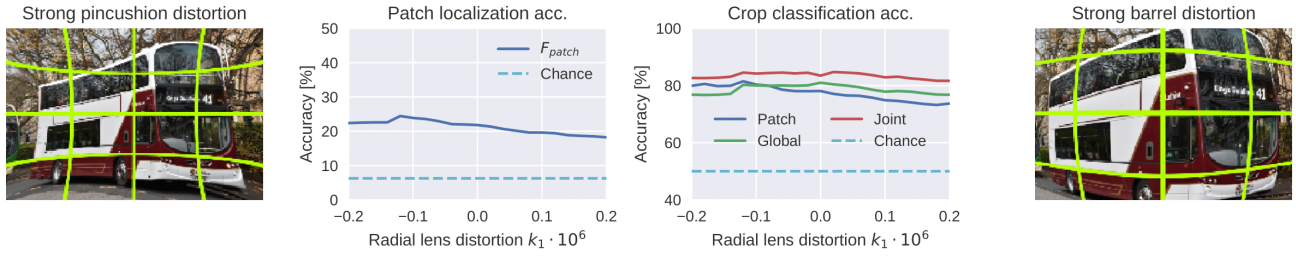
(a) **Red transverse chromatic aberration** in the positive (outward) direction boosts performance.



(b) **Blue transverse chromatic aberration** in the positive (outward) direction boosts performance.



(c) Adjusting **color saturation** away from 100% (= identity) slightly degrades performance.



(d) The degree of **radial lens distortion** in our dataset may be too subtle to substantially affect the integrated crop detection model, although due to the noisy results, this is inconclusive.

Figure 13: **Extended breakdown of image attributes.** See Figure 5 for the main results.