Visual Chirality

Zhiqiu Lin[1] Abe Davis[2] Jin Sun[2] Noah Snavely[2]
Cornell University[1] Cornell Tech[2]

Abstract

How can we tell whether an image has been mirrored? While we understand the geometry of mirror reflections very well, less has been said about how it affects distributions of imagery at scale, despite widespread use for data augmentation in computer vision. In this paper, we investigate how the statistics of visual data are changed by reflection. We refer to these changes as "visual chirality," after the concept of geometric chirality—the notion of objects that are distinct from their mirror image. Our analysis of visual chirality reveals surprising results, including low-level chiral signals pervading imagery stemming from image processing in cameras, to the ability to discover visual chirality in images of people and faces. Our work has implications for data augmentation, self-supervised learning, and image forensics.

1. Introduction

> "...there's a room you can see through the glass—that's just the same as our drawing room, only the things go the other way."
>
> — Lewis Carroll,
> "Alice's Adventures in Wonderland & Through the Looking-Glass"

There is a rich history of lore involving reflections. From the stories of Perseus and Narcissus in ancient Greek mythology to the adventures of Lewis Carroll's Alice and J.K. Rowling's Harry Potter, fiction is full of mirrors that symbolize windows into worlds similar to, yet somehow different from, our own. This symbolism is rooted in mathematical fact: what we see in reflections is consistent with a world that differs in subtle but meaningful ways from the one around us—right hands become left, text reads backward, and the blades of a fan spin in the opposite direction. What we see is, as Alice puts it, "just the same... only the things go the other way".

Geometrically, these differences can be attributed to a world where distances from the reflecting surface are negated, creating an orientation-reversing isometry with objects as we normally see them. While the properties of such isometries are well-understood in principle, much less is known about how they affect the statistics of visual data at scale. In other words, while we understand a great deal about how reflection changes image data, we know much less about how it changes what we learn from that data—this, despite widespread use of image reflection (e.g., mirror-flips) for data augmentation in computer vision.

(a) (b) (c)

Figure 1. **Which images have been mirrored?** Our goal is to understand how distributions of natural images differ from their reflections. Each of the images here appears plausible, but some subset have actually been flipped horizontally. Figuring out which can be a challenging task even for humans. Can you tell which are flipped? Answers are in Figure 2.

This paper is guided by a very simple question: How do the visual statistics of our world change when it is reflected? One can understand some aspects of this question by considering the images in Figure 1. For individual objects, this question is closely related to the concept of *chirality* [12]. An object is said to be *chiral* if it cannot be rotated and translated into alignment with its own reflection, and *achiral* otherwise.[1] Put differently, we can think of chiral objects as being fundamentally changed by reflection—these are the things that "go the other way" when viewed through a mirror—and we can think of achiral objects as simply being moved by reflection. Chirality provides some insight into our guiding question, but remains an important step removed from telling us how reflections impact learning. For this, we need a different measure of chirality—one we call *visual chirality*—that describes the impact of reflection on distributions of imagery.

1.1. Defining Visual Chirality

To define visual chirality, we first consider data augmentation for learning in computer vision. Machine learning algorithms build on the idea that we can approximate distributions by fitting functions to samples drawn from those

[1] More generally, any figure is achiral if its symmetry group contains any orientation-reversing isometries.

(a) (b) (c)

Figure 2. Images from Figure 1 with chirality-revealing regions highlighted. These regions are automatically found by our approach to chiral content discover. (a, **flipped**) *Text chirality.* Text is a strong cue. (b, **not flipped**) *Object chirality.* The shirt collar, and in particular which side the buttons are on, are a more subtle cue. (c, **flipped**) *Object interaction chirality.* While guitars are sometimes (nearly) symmetric, the way we hold them is not (the left hand is usually on the fretboard).

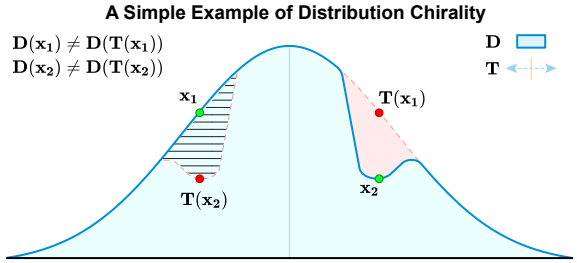


Figure 3. Using a transformation $\mathbf{T}$ to augment a sample-based approximation of the distribution $\mathbf{D}$ assumes symmetry with respect to $\mathbf{T}$. We define visual chirality in terms of approximation error induced by this assumed symmetry when $\mathbf{T}$ is image reflection.

distributions. Viewed in this light, data augmentation can be seen as a way to improve sampling efficiency for approximating a distribution $\mathbf{D}(\mathbf{x})$ by assuming that $\mathbf{D}$ is invariant to some transformation $\mathbf{T}$. More precisely, augmenting a training sample $\mathbf{x}$ with the function $\mathbf{T}$ assumes symmetry of the form:

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{T}(\mathbf{x})) \tag{1}$$

which allows us to double our effective sampling efficiency for $\mathbf{D}$ at the cost of approximation error wherever the assumed symmetry does not hold (see Figure 3).

Recall that for achiral objects reflection is equivalent to a change in viewpoint. Therefore, if we consider the case where $\mathbf{D}$ is a uniform distribution over all possible views of an object, and $\mathbf{T}$ is image reflection, then Equation 1 reduces to the condition for achirality. We can then define *visual chirality* by generalizing this condition to arbitrary visual distributions. In other words, we define visual chirality as a measure of the approximation error associated with assuming visual distributions are symmetric under reflection. Defining visual chirality in this way highlights a close connection with data augmentation. Throughout the paper we will also see implications to a range of other topics in computer vision, including self-supervised learning and image forensics.

Note that our definition of visual chirality can also be generalized to include other transformations. In this paper we focus on reflection, but note where parts of our analysis could also apply more broadly.

Notes on Visual Chirality vs. Geometric Chirality:

Here we make a few clarifying observations about visual chirality. First, while geometric chirality is a binary property of objects, visual chirality can be described in terms of how *much* Equation 1 is violated, letting us discuss it as a continuous or relative property of visual distributions, their samples, or, as we will see in Section 4, functions applied to visual distributions. Second, visual chirality and geometric chirality need not imply one another. For example, human hands have chiral geometry, but tend to be visually achiral because the right and left form a reflective pair and each occurs with similar frequency. Conversely, an achiral object with one plane of symmetry will be visually chiral when it is only viewed from one side of that plane. For the remainder of the paper we will refer to geometric chirality as such to avoid confusion.

2. Related work

Chirality is closely related to symmetry, a long-studied topic in computer vision. Closely related to our work is recent work exploring the asymmetry of *time* (referred to as "Time's arrow") in videos, by understanding what makes videos look like they are being played forwards or backwards [20, 24]—a sort of temporal chirality. We explore the spatial version of this question, by trying to understand what makes images look normal or mirrored. This spatial chirality is related to other orientation problems in graphics in vision, such as detecting "which way is up" in an image or 3D model that might be oriented incorrectly [23, 4]. Compared to upright orientation, chirality is potentially much more subtle—many images may lack strong chirality cues, including a couple of the images in Figure 1. Upright orientation, as well as other related tasks, have also been used as proxy tasks for unsupervised learning of feature representations [5]. Such tasks include the arrow of time task mentioned above [24], solving jigsaw puzzles [19], and reasoning about relative positions of image patches [2].

Our problem represents an interesting spin on the classic task of detecting symmetries in images [15]. As such, our work is related to the detection and classification of asymmetric, chiral objects, as explored by Hel-Or *et al.*in their work on "how to tell left from right" [9], e.g., how to tell a left hand in an image from a right hand. However, this prior work generally analyzed *geometric* chirality, as opposed to the *visual chirality* we explore, as defined above—for instance, a right hand might be geometrically chiral but not visually chiral, while a right hand holding a pencil might visually chiral due to the prevalence of right-handed people.

Our work also relates to work on unsupervised discovery from large image collections, including work on identifying distinctive visual characteristics of cities [3] and of yearbook photos over time [6].

Finally, a specific form of chirality (sometimes referred to as *cheirality*) has been explored in geometric vision. Namely, there is an asymmetry between 3D points in front of a camera and points in back of a camera. This asymmetry can be exploited in various geometric fitting tasks [7].

3. Measuring Visual Chirality

In principle, one way to measure visual chirality would be to densely sample a distribution and analyze symmetry in the resulting approximation. However, this approach is inefficient and in most cases unnecessary; we need not represent an entire distribution just to capture its asymmetry. Instead, we measure visual chirality by training a network to distinguish between images and their reflections. Intuitively, success at this task should be bound by the visual chirality of the distribution we are approximating.

Given a set of images sampled from a distribution, we cast our investigation of visual chirality as a simple classification task. Let us denote a set of training images from some distribution as $C_{\text{positive}} = \{I_1, I_2, \cdots, I_n\}$ (we assume these images are photos of the real world and have not been flipped). We perform a horizontal flip on each image I_i to produce its reflected version I_i'. Let us denote the mirrored set as $C_{\text{negative}} = \{I_1', I_2', \cdots, I_n'\}$. We then assign a binary label y_i to each image I_i in $C_{\text{positive}} \cup C_{\text{negative}}$:

$$y_i = \begin{cases} 0 & \text{if } I_i \in C_{\text{negative}}, \text{ i.e., flipped} \\ 1 & \text{if } I_i \in C_{\text{positive}}, \text{ i.e., non-flipped} \end{cases} \quad (2)$$

We train deep Convolutional Neural Nets (CNNs) with standard classification losses for this problem, because they are good at learning complex distribution of natural images [13]. Measuring a trained CNNs performance on a validation set provides insight on the visual chirality of data distribution we are investigating on.

Next we discuss details on training such a network and the techniques we use to discover the sources of visual chriality of the data distribution using a trained model as a proxy.

Network architecture. We adopt a ResNet network [8], a widely used deep architecture for image classification tasks. In particular, we use ResNet-50 and replace the last average pooling layer of the network with a global average pooling layer [16] in order to support variable input sizes.

Optimization. We train the network in a mini-batch setting using a binary cross-entropy loss. We optionally apply random cropping, and discuss the implications of such data augmentation below. We normalize pixel values by per-channel mean-subtraction and dividing by the standard deviation. We use a stochastic gradient descent [1] optimizer, with momentum 0.9 and L_2 weight decay of 10^{-5}.

(a) Resizing (b) Random Cropping

Figure 4. **Resizing vs. random cropping as dataset preprocessing**. This figure shows CAM heatmaps for an image from models trained with two preprocessing methods: (a) resizing and (b) random cropping. We observe that the resizing scheme learns cues in the edges or corners of images (note the focus on the lower left corner of (a)), where JPEG encoding can be asymmetric. On the other hand, the random cropping scheme captures the meaningful high-level cue—the chiral shirt collar.

Hyperparameter selection. Finding a suitable learning rate is important for this task. We perform a grid search in the log domain and select the best learning rate for each experiment by cross-validation.

Shared-batch training. During training, we include both I_i and I_i' (i.e., positive and negative chirality versions of the same image) in the same mini-batch. We observe significant improvements in model performance using this approach, in alignment with prior self-supervised learning methods [5].

Discovering sources of visual chirality. If a trained model is able to predict whether an image is flipped or not with high accuracy, it must be using a reliable set of visual features from the input image for this task. We consider those cues as the source of visual chrility in the data distribution.

We use Class Activation Maps (CAM) [26] as a powerful tool to visualize those discriminative regions from a trained model. Locations with higher activation values in CAM make correspondingly larger contributions to predicting flipped images.

Throughout this paper, we visualize these activation maps as heatmaps using the Jet color scheme (red=higher activations, blue=lower activations). We only compute CAM heatmaps corresponding to an image's correct label. Figure 2 shows examples of such class activation maps.

In the following sections, we analyze visual chirality discovered in different settings using the tools we described in this section.

4. The Chirality of Image Processing

When we first trained our model to distinguish between images and their reflections, we quickly observed that the network would find ways to accomplish this task using low-level cues that appeared only loosely correlated with the image's content. Furthermore, the strength of these cues seemed to vary a great deal with changes in how data was prepared; for example, Figure 4 shows two different CAM maps for the same sample image. The left is derived from a network trained on resized data, and the right is derived from a network trained on random crops of the same data. Both maps identify a dark corner of the image as being discriminative, as well as part of the shirt on one of the the image's human subjects. However, these networks appear to disagree about the relative stretch of the chiral cues in these regions. This result illustrates how the way we capture and process visual data—even down to the level of Bayer mosaics in cameras or JPEG compression—can have a significant impact on its chirality. In this section we develop theory to help reason about that impact, and use that theory to predict what networks will learn in experiments.

4.1. Transformation Commutativity

The key challenge of predicting how an imaging process will affect chirality is finding a way to reason about its behavior under minimal assumptions about the distribution to which it will be applied. For this, we consider what it means for an arbitrary imaging transformation $\mathbf{J}$ to preserve the chirality of a distribution $\mathbf{D}$ under the transformation $\mathbf{T}$. There are two conditions for this to happen. First, $\mathbf{J}$ must preserve any existing symmetries that affect the chirality of $\mathbf{D}$. We can formalize this by applying $\mathbf{J}$ to the arguments of $\mathbf{D}$ on both sides of Equation 1:

$$\mathbf{D}(\mathbf{J}(\mathbf{x})) = \mathbf{D}(\mathbf{J}(\mathbf{T}(\mathbf{x}))) \tag{3}$$

Second, $\mathbf{J}$ must not introduce any new asymmetries that could change the chirality of $\mathbf{D}$. We can formalize this by simply substituting $\mathbf{J}(\mathbf{x})$ for $\mathbf{x}$ in Equation 1:

$$\mathbf{D}(\mathbf{J}(\mathbf{x})) = \mathbf{D}(\mathbf{T}(\mathbf{J}(\mathbf{x}))) \tag{4}$$

Now, combining Equations 3 and 4, we get:

$$\mathbf{D}(\mathbf{J}(\mathbf{T}(\mathbf{x}))) = \mathbf{D}(\mathbf{T}(\mathbf{J}(\mathbf{x}))) \tag{5}$$

which shows that $\mathbf{J}$ preserves symmetry under $\mathbf{D}$ with respect to $\mathbf{T}$ if and only if $\mathbf{T}$ and $\mathbf{J}$ are commutative under $\mathbf{D}$.

4.2. Predicting Chirality With Commutativity

Tying the chirality of transformations to commutativity gives us a powerful way to predict their effect on learning given a minimal representative sample of training data and trivial amount of compute. We simply have to check whether a given transformation $\mathbf{J}$ commutes with reflection. In our supplemental material we use this strategy to predict the chirality of Bayer demosaicing, JPEG compression, demosaicing + JPEG compression, and all three of these again combined with random cropping. We then show that, in all 6 cases, our analysis predicts the performance of a network trained from scratch to distinguish between random noise images and their reflection. These predictions also explain our observations in Figure 4. While the full details are presented in the supplemental material, some key highlights include:

- Demosaicing and JPEG compression are both individually chiral and chiral when combined.
- When random cropping is added to demosaicing or JPEG compression individually, they become achiral.
- When demosaicing, jpeg compression, and random cropping are all combined, the result is chiral.

These conclusions have implications on image forensics—for instance, our analysis gives us new theoretical and practical tools for determining if image content has been flipped, a common operation in image editing.

Note that none of Equations 1-5 make assumptions about our choice of $\mathbf{J}$, $\mathbf{T}$, or $\mathbf{D}$, which lets us generalize our conclusion about commutativity to the preservation of more arbitrary symmetries in visual distributions. For example, Doersch *et al.* [2] found that when they used the relative position of different regions in an image as a signal for self-supervised learning, the networks would "cheat" by utilizing chromatic aberration for prediction. Identifying the relative position of image patches requires asymmetry with respect to image translation. Applied to their case, Equation 5 suggests that any aspect of the imaging process that does not commute with translation would provide a solution to their training task. And, indeed, this is true of chromatic aberration, which has radial but not translational symmetry in the image plane.

5. High-level visual chirality

While analysis of chiralities that arise in image processing have useful implications in forensics, we are also interested in understanding what kinds of high-level visual content (objects, object regions, etc.) reveals visual chirality, and whether we can discover these cues automatically. As described in Section 4, if we try to train a network from scratch, it invariably starts to pick up on uninterpretable, low-level image signals. Instead, we hypothesize that if we start with a ResNet network that has been pre-trained on ImageNet object classification, then it will have a familiarity with objects that will allow it to avoid picking up on low-level cues. Note, that such ImageNet-trained networks should *not* have features sensitive to specifically to chirality—indeed, as noted

above, many ImageNet classifiers are trained using random horizontal flips as a form of data augmentation.

Data. What distribution of images do we use for training? We could try to sample from the space of all natural images. However, because we speculate that many chirality cues have to do with people, and with manmade objects and scenes, we start with images that feature *people*. In particular, we utilize the StreetStyle dataset of Matzen *et al.* [17], which consists of millions of images of people gathered from *Instagram*. For our work, we select a random subset of 700K images from StreetStyle, and refer to this as the *Instagram* dataset; example images are shown in Figures 1 and 5. We randomly sample 5K images as a test set S_{test}, and split the remaining images into training and validation sets with a ratio of 9:1 (unless otherwise stated, we use this same train/val/test split strategy for all experiments in this paper).

Training. We trained the chirality classification approach described in Section 3 on *Instagram*, starting from an ImageNet-pretrained model. As it turns out, the transformations applied to images before feeding them to a network are crucial to consider. Initially, we downsampled all input images bilinearly to a resolution of 512×512. A network so trained achieves a 92% accuracy on the *Instagram* test set, a surprising result given that determining whether an image has been flipped can be difficult even for humans.

Without careful preprocessing, a CNN trained on chirality classification will pick up on traces left by low-level processing, such as boundary artifacts produced by JPEG encoding. This was manifest in CAM heatmaps that often fired near the corners of images. For now, we find that networks can be made resistant to the most obvious such artifacts by performing random cropping of input images. In particular, if we randomly crop a 512×512 window from the input images during training and testing, rather than simply resizing the images, then a trained CNN achieves a test accuracy to 80%, still a surprisingly high result.

Non-text cues. Examining the most confident classifications, we found that many involved text (e.g., on clothing or in the background), and that CAM heatmaps often predominantly focused on text regions. Indeed, text is such a strong signal for chirality that it can evidently drown out other signals. This yields a useful insight: we may be able to leverage chirality to learn a text detector via self-supervision, for any language (so long as the writing is chiral, which is true for many if not all languages).

However, for the purpose of the current analysis, we wish to discover non-text chiral cues as well. To make it easier to identify such cues, we ran an automatic text detector [25] on *Instagram*, split it into text and no-text subsets, and then randomly sampled the no-text subset to form new training and text set. On the no-text subset, chirality classification accuracy drops from 80% to 74%—lower, but still well above chance.

Training set	Preprocess	Test Accuracy	
		Instagram	F100M
Instagram	Resizing	0.92	0.57
Instagram	RandCrop	0.80	0.59
Instagram (no-text)	RandCrop	0.74	0.55

Table 1. **Chirality classification performance of models trained on *Instagram*.** Hyper-parameters were selected by cross validation. The first column indicates the training dataset, and the second column the processing that takes place on input images. The last columns report on a held-out test set, and on an unseen dataset (Flickr100M, or F100M for short). Note that the same preprocessing scheme (resize vs. random crop) is applied to both the training and test sets, and the model trained on *Instagram* without text is also tested on *Instagram* without text.

Generalization. Perhaps our classifier learns features specific to Instagram images. To test this, Table 1 (last column) shows the evaluation accuracy of all models (without fine-tuning) on another dataset of Internet photos, a randomly selected subset of photos from Flickr100M [22]. Note that there is a significant domain gap between *Instagram* and Flickr100M, in that images in our *Instagram* dataset all contain people, whereas Flickr100M features more general content (landscapes, macro shots, etc.) in addition al people. While the performance on Flickr100M is naturally lower than on *Instagram*, our *Instagram*-trained models still perform above chance rates, with an accuracy of 55% (or 59% if text is considered), suggesting that our learned chiral features can generalize to new distributions of photos.

5.1. Revealing object-level chiral features

Inspecting the CAM heatmaps derived from our non-text-trained *Instagram* model reveals a network that focuses on a coherent set of local regions, such as smart phones and shirt pockets, across different photos. To further understand what the network has learned, we develop a way to group the images, as well as their CAM heatmaps, to determine which cues are most common and salient. Inspired by work on mid-level discriminative patch mining [3, 21, 14, 18], we propose a method built upon CAM that we call *chiral feature clustering*, which automatically groups images based on the similarity of features extracted by the network, in regions deemed salient by CAM.

Chiral feature clustering. First, we extract the most discriminative local chiral feature from each image to use as input to our clustering stage. To do so, we consider the feature maps that are output from the last convolutional layer of our network. As is typical of CNNs, these features are maps with low spatial resolution, but with high channel di-

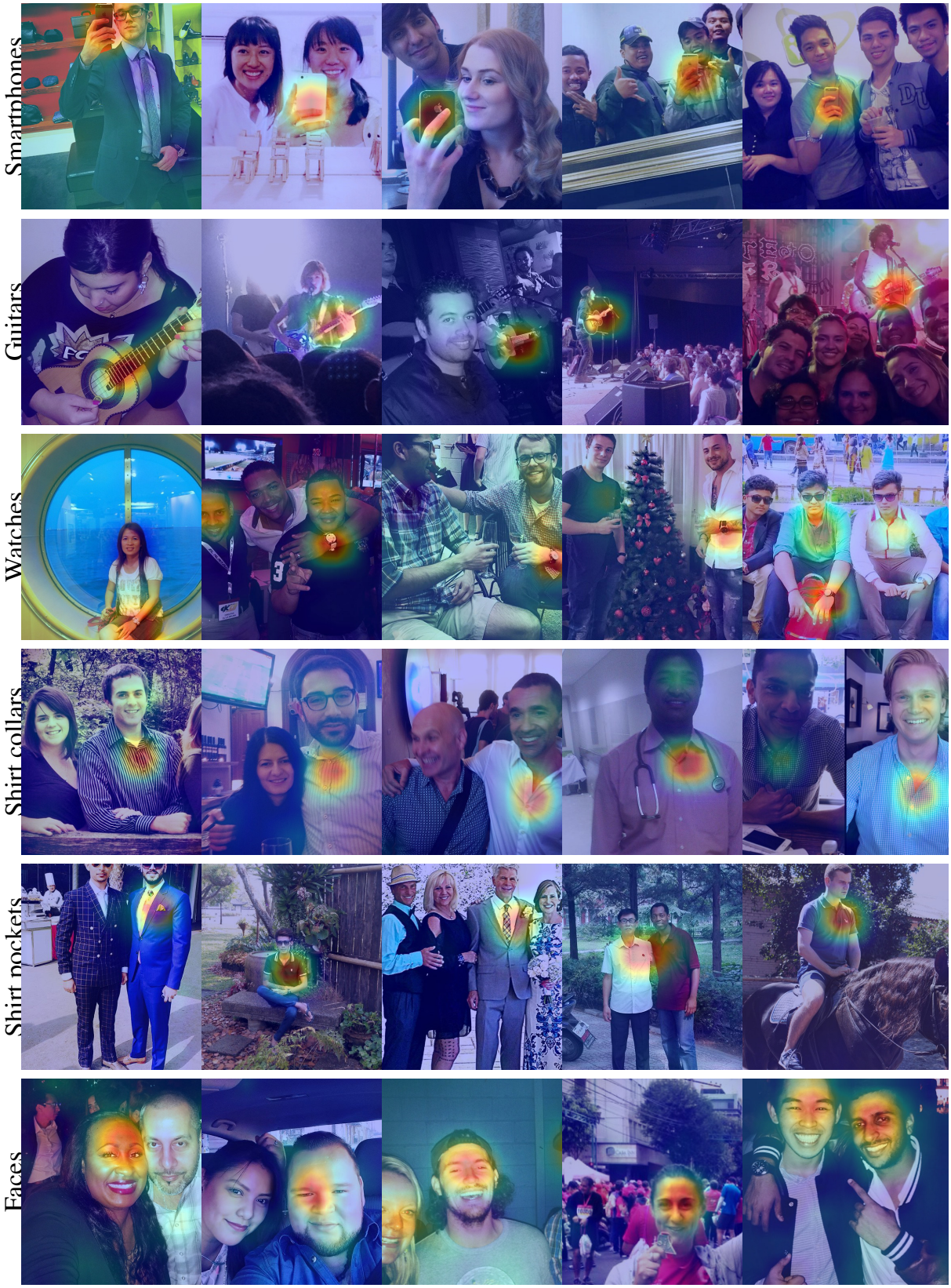


Figure 5. **Chiral clusters discovered in the Instagram dataset.** Each row shows selected images from a single discovered cluster. Each image is shown with its corresponding CAM heatmap superimposed, where red regions are highly correlated with its true chirality. We discover a range of object-level chiral clusters, such as cellphones, watches, and shirts.

mensionality (e.g., 2048).

Given an input image, let us denote the output of this last convolutional layer as $\mathbf{f}$, which in our case is a feature map of dimensions $16 \times 16 \times 2048$ ($w \times h \times c$). Let $\mathbf{f}(x, y)$ denote the 2048-D vector at location (x, y) of $\mathbf{f}$. We apply CAM, using the correct chirality label for the image, to obtain a 16×16 weight activation map A. Recall that the higher the value of $A(x, y)$, the higher the contribution of the local region corresponding to (x, y) to the prediction of the correct chirality label.

We then locate the spatial maxima of A, $(x^*, y^*) = \arg\max_{(x,y)} A(x, y)$ in each image. These correspond to points deemed maximally salient for the chirality task by the network. We extract $\mathbf{f}(x^*, y^*)$ as a local feature vector describing this maximally chiral region. Running this procedure for each image yields a collection for feature vectors, on which we run k-means clustering.

Results of chiral feature clustering. We apply this clustering procedure to our no-text *Instagram* test set, using $k = 500$ clusters. We observe that this method is surprisingly effective and identifies a number of intriguing object-level chiral cues in our datasets. We refer to these clusters as *chiral clusters*. Examples of striking high-level chiral clusters are shown in Figure 5, and include phones (e.g., held in a specific way to take photos in a mirror), watches (typically worn on the left hand), shirt collars (shirts with buttoned collared typically button on a consistent side), shirt pockets, pants, and other objects.

Many of these discovered chiral clusters are highly interpretable. However, some clusters are difficult to understand. For instance, in the face cluster shown in the last row of Figure 5, the authors could not find obvious evidence of visual chirality, leading us to suspect that there may be subtle chirality cues in faces. We explore this possibility in Section 6. We also observe that some clusters focus on sharp edges in the image, leading us to suspect that some low-level image processing cues are being learned in spite of the ImageNet initialization.

6. Visual chirality in faces

Inspired by our results on the Instagram dataset in Section 5, we now analyze chirality in face images.

To do so, we use the FFHQ dataset [11] as the basis for learning. FFHQ is a recent dataset of 70K high-quality faces introduced in the context of training generative methods. We use 7% of the images as a test set and the remaining images for training and validation. We train various models on FFHQ, first downsampling images to a resolution of 520×520, then randomly cropping to 512×512. We train a standard model starting from ImageNet pre-trained features. This model achieves an accuracy of 81%, which is a promising indicator that our network can indeed learn to predict the chirality of faces with accuracy significantly better than chance.

However, perhaps there is some bias in FFHQ that leads to spurious chirality signals. For instance, since a face detector is used to create FFHQ, there is the possibility that the detector is biased, e.g., for left-facing faces vs. right-facing faces. To test this, we evaluate how well our FFHQ-trained model generalizes to other independent datasets. In particular, we evaluate this model (without fine-tuning) on another dataset - LFW, a standard face dataset [10] (we upsample the low-resolution images in LFW to 512×512 to match our input resolution), and get an accuracy of 60%—not as high as FFHQ, perhaps due to different distributions of faces, but still significantly better than chance.

To qualitatively explore the chirality cues the model has identified, we show a sample of chiral clusters derived from the FFHQ test set in Figure 6. We can see that the CAM heatmaps in each cluster focus on specific facial regions. Based on these cluster, we have identified some intriguing preliminary hypotheses about facial chirality:

Hair part. The first cluster in Figure 6 indicates a region around the part of the hair on the left side of the forehead. We conjectured that this could be due to bias in hair part direction. We manually inspected a subset of the FFHQ test set, and found that a majority of people pictured parted their hair from left to right (the ratio is ∼2:1 for photos with visible hair part), indicating a bias for asymmetry in hair, possibly due to people preferentially using their dominant right hand to part their hair.

Predominant gaze direction, aka ocular dominance[2]. The second cluster cluster in Figure 6 highlights a region around the corner of the right eye. We conjectured that this may have to do with bias in gaze direction, possibly due to ocular dominance. We use gaze detection software[3] to determine and compare the locations of the pupil in the left and right eyes. We found that indeed more than two thirds of people in portrait photographs gaze more towards the left.

Note that there are some other clusters left to be explained (for example the "beard" cluster, which may be due to that males tend to use right hands to shave beard). Exploring such cues would make for interesting future work.

7. Conclusion

We propose to discover visual chirality in image distributions using a self-supervised learning approach by predicting whether a photo is flipped or not, and by analyzing properties of transformations that yield chirality. We report various visual chirality cues identified using our tool on a variety of datasets such as Instagram photos and FFHQ face images.

[2]https://en.wikipedia.org/wiki/Ocular_dominance
[3]https://github.com/shaoanlu/GazeML-keras

Figure 6. **Chiral clusters found in FFHQ.** It shows 3 chiral clusters of FFHQ dataset. The leftmost image of each row is the average face + CAM heatmap for all non-flipped images inside the each cluster. We also show some random non-flipped examples for each cluster.

We also find that low-level chiral cues are likely pervasive in images, due to chiralities inherent in standard image processing pipelines. Our analysis has implications in data augmentation, self-supervised learning, and image forensics. Our results implies that visual chirality indeed exists in many vision datasets and such properties should be taken into account when developing real-world vision systems. However, our work suggests that it can also be used as a signal that can be leveraged in interesting new ways. For instance, since text is highly chiral, our work points to interesting future direction in utilizing chirality in a self-supervised way to learn to detect text in images in the wild.

References

[1] L. Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade. Springer*, page 421–436, 2012. 3

[2] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. *ICCV*, 2015. 2, 4

[3] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes Paris look like Paris? *SIGGRAPH*, 31(4), 2012. 3, 5

[4] H. Fu, D. Cohen-Or, G. Dror, and A. Sheffer. Upright orientation of man-made objects. In *SIGGRAPH*, 2008. 2

[5] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 2, 3

[6] S. Ginosar, K. Rakelly, S. Sachs, B. Yin, and A. A. Efros. A Century of Portraits: A visual historical record of american high school yearbooks. In *ICCV Workshops*, December 2015. 3

[7] R. Hartley. Cheirality invariants. In *Proc. DARPA Image Understanding Workshop*, 1993. 3

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[9] Y. Hel-Or, S. Peleg, and H. Hel-Or. How to tell right from left. In *CVPR*, 1988. 2

[10] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 7

[11] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018. 7

[12] W. T. Kelvin. The molecular tactics of a crystal. *J. Oxford Univ. Jr. Sci. Club*, 18:3–57, 1894. 1

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 3

[14] Y. Li, L. Liu, C. Shen, and A. van den Hengel. Mid-level deep pattern mining. In *CVPR*, 2015. 5

[15] Y. Liu, H. Hel-Or, C. S. Kaplan, and L. J. V. Gool. Computational symmetry in computer vision and computer graphics. *Foundations and Trends in Computer Graphics and Vision*, 5(1-2):1–195, 2010. 2

[16] Q. C. M. Lin and S. Yan. Network in network. In *International Conference on Learning Representations*, pages 2921–2929, 2014. 3

[17] K. Matzen, K. Bala, and N. Snavely. StreetStyle: Exploring world-wide clothing styles from millions of photos. *CoRR*, abs/1706.01869, 2017. 5

[18] K. Matzen and N. Snavely. BubbLeNet: Foveated imaging for visual discovery. In *ICCV*, 2015. 5

[19] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 2

[20] L. C. Pickup, Z. Pan, D. Wei, Y.-C. Shih, C. Zhang, A. Zisserman, B. Schölkopf, and W. T. Freeman. Seeing the arrow of time. In *CVPR*, 2014. 2

[21] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 5

[22] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. Yfcc100m: The new data in multimedia research. *CACM*, 59(2), Jan. 2016. 5

[23] A. Vailaya, H. Zhang, C. Yang, F.-I. Liu, and A. K. Jain. Automatic image orientation detection. *Trans. Image Processing*, 11(7):746–55, 2002. 2

[24] D. Wei, J. Y. S. Lim, A. Zisserman, and W. T. Freeman. Learning and using the arrow of time. In *CVPR*, 2018. 2

[25] H. W. Y. W. S. Z. W. H. X. Zhou, C. Yao and J. Liang. East: An efficient and accurate scene text detector. In *CVPR*, 2017. 5

[26] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 3

Visual Chirality—Supplemental Material: Commutativity and The Chirality of Imaging Processes

Zhiqiu Lin[1] Abe Davis[2] Jin Sun[2] Noah Snavely[2]
Cornell University[1] Cornell Tech[2]

Abstract

In this document we explore how commutativity can be used to predict the chirality of different imaging processes, including demosaicing, JPEG compression, and random cropping.

1. Introduction

A key goal of our work is to understand how reflection changes what we learn from image data. We can think of this change as the difference between two distributions: one represented by a data set, and the other represented by its reflection. We found that when training a network to distinguish between samples from these two different distributions, it can often accomplish this task by looking at low-level artifacts left by various imaging processes. This leads us to ask, when can we attribute visual chirality to the content being imaged, and when might it instead be the result of the imaging process? To answer this question, we develop theory relating the chirality of an imaging process to its commutativity with reflection. We show that by examining this commutativity on representative samples we can predict whether an imaging process might introduce visual chirality into a dataset.

Section 2 introduces the basic connection between the commutativity of an imaging process $\mathbf{J}$ with a transformation $\mathbf{T}$—in our case, reflection—and whether $\mathbf{J}$ can create new asymmetries with respect to $\mathbf{T}$. Section 3 applies our theory to analyzing Bayer demosaicing and JPEG compression, and Section 4 examines how translation invariance, as incorporated through random cropping, can change the chirality of imaging processes. A surprising finding is that by taking a collection of images that are initially achiral, and passing them through Bayer demosaicing and JPEG compression, this processed collection can become *chiral*, even when looking at random crops of these images. This holds even though Bayer demosaicing and JPEG compression alone are insufficient to introduce chirality to an achiral distribution when random cropping is applied. Our results suggests that imperceptible chiral traces are left in photos by imaging pipelines, which has implications on self-supervised learning, image forensics, etc.

2. Commutativity

We begin by reviewing the derivation from our paper. Consider a distribution $\mathbf{D}$ over images. For our purposes, $\mathbf{D}$ can be thought of as a probability distribution over images from which one has a number of samples (e.g., to form a dataset), in which case $\mathbf{D}(\mathbf{x})$ denotes the probability of image $\mathbf{x}$. Alternatively, we will later consider $\mathbf{D}$ to be a set of images, which we take to be equivalent to a uniform probability distribution over the elements of that set.

Our main derivation shows that an imaging process $\mathbf{J}$ (e.g., JPEG compression) preserves the achirality of a distribution $\mathbf{D}$ with respect to some transformation $\mathbf{T}$ when $\mathbf{J}$ and $\mathbf{T}$ commute under $\mathbf{D}$. We derive this generally for any transformation $\mathbf{T}$ (not just mirror flips). A more intuitive explanation of how we use this in the specific case where $\mathbf{T}$ is a mirror reflection is provided in Section 2.2.

For clarity, we will use $\mathbf{D_J}$ to refer to the distribution that results from applying $\mathbf{J}$ to the domain of $\mathbf{D}$:

$$\mathbf{D_J}(\mathbf{x}) = \sum_{\forall \mathbf{x}_i : \mathbf{J}(\mathbf{x}_i) = \mathbf{x}} \mathbf{D}(\mathbf{x}_i) \tag{1}$$

First, we have the definition of the symmetry of the distribution $\mathbf{D}$ under $\mathbf{T}$:

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{T}(\mathbf{x})) \tag{2}$$

Now, we can state our proposition as follows:

Proposition 1 *If operation* $\mathbf{J}$ *and* $\mathbf{T}$ *are commutative under* $\mathbf{D_J}$*, then* $\mathbf{J}$ *preserves the symmetry of* $\mathbf{D}$ *with respect to* $\mathbf{T}$.

Now recall the conditions for preservation. First, $\mathbf{J}$ must preserve the existing symmetries of $\mathbf{D}$ under $\mathbf{T}$. We can formalize this by applying $\mathbf{J}$ to the arguments on both sides of Equation 2 applied to $\mathbf{D_J}$:

$$\mathbf{D_J}(\mathbf{J}(\mathbf{x})) = \mathbf{D_J}(\mathbf{J}(\mathbf{T}(\mathbf{x}))) \tag{3}$$

The second condition is that $\mathbf{J}$ must not create new asymmetries under $\mathbf{T}$ that affect achirality. We can write this by substituting $\mathbf{J}(\mathbf{x})$ for $\mathbf{x}$ in Equation 2:

$$\mathbf{D_J}(\mathbf{J}(\mathbf{x})) = \mathbf{D_J}(\mathbf{T}(\mathbf{J}(\mathbf{x}))) \tag{4}$$

Now, combining Equations 3 and 4, we get:

$$\mathbf{D_J}(\mathbf{J}(\mathbf{T}(\mathbf{x}))) = \mathbf{D_J}(\mathbf{T}(\mathbf{J}(\mathbf{x}))) \tag{5}$$

which implies that $\mathbf{J}$ preserves symmetry of $\mathbf{D}$ with respect to $\mathbf{T}$ if $\mathbf{T}$ and $\mathbf{J}$ are commutative under $\mathbf{D_J}$.

Preserving Achirality vs Chirality: Note that we have not accounted for the scenario where $\mathbf{J}$ removes asymmetries. This means that while achirality is preserved, chirality may not be.[1] In fact, loss of chirality is almost certain to happen, as imaging is necessarily lossy and therefore trivially creates symmetry. However, our primary concern is determining whether the asymmetries we learn from data are the result of content or an artifact of processing. This question does not apply to asymmetries we never observe. The design of a chirality-preserving imaging system could be an interesting problem related to computational photography, but we leave this to future work.

2.1. Commutative Residual

Equation 5 gives us a simple, abstract way to reason about the chirality of transformations. First, note that equality of two vectors is trivially sufficient to assume they are equal under some distribution:

$$x_1 = x_2 \longrightarrow \mathbf{D}(\mathbf{x}_1) = \mathbf{D}(\mathbf{x}_2) \tag{6}$$

Combining this with Equation 5, we can say that achirality is preserved when:

$$\mathbf{J}(\mathbf{T}(\mathbf{x})) - \mathbf{T}(\mathbf{J}(\mathbf{x})) = 0 \tag{7}$$

Now we define the *commutative residual image* of $\mathbf{x}$, denoted $\mathbf{E}_J(\mathbf{x})$:

$$\mathbf{E}_J(\mathbf{x}) = \mathbf{J}(\mathbf{T}(\mathbf{x})) - \mathbf{T}(\mathbf{J}(\mathbf{x})) \tag{8}$$

We can get a rough measure of the chirality of a transformation on some representative sample $\mathbf{x}$ by looking at the value of $|\mathbf{E}_J(\mathbf{x})|$, which we summarize by its average $\hat{\mathbf{e}}_J(\mathbf{x})$. We refer to $\hat{\mathbf{e}}_J(\mathbf{x})$ as a *commutative residual*.

2.2. Intuition for Commutative Residuals

An alternative derivation of commutative residuals can be generalized from the case where $\mathbf{T}$ is its own inverse, as is true of reflections. Consider the effect of $\mathbf{J}$ on a distribution represented by the dataset $\mathbf{D} = \{\mathbf{x}, \mathbf{T}(\mathbf{x})\}$. There is no way for a network to tell between a random sample from $\mathbf{D}$ and a flipped random sample, as $\mathbf{D}$ is closed under $\mathbf{T}$. But what happens when we apply $\mathbf{J}$? $\mathbf{D}$ becomes $\mathbf{D_J} = \{\mathbf{J}(\mathbf{x}), \mathbf{J}(\mathbf{T}(\mathbf{x}))\}$, and we can measure the asymmetry of this new distribution by taking the difference between one element and the reflection of the other:

$$\mathbf{J}(\mathbf{T}(\mathbf{x})) - \mathbf{T}(\mathbf{J}(\mathbf{x})) \tag{9}$$

which is precisely how we define the commutative residual image.

[1] There is a small, unfortunate typo in our paper swapping these two. We will fix this in our final draft.

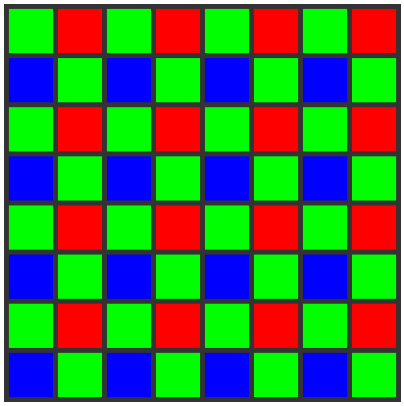


Figure 1. **Example 8× 8 Bayer pattern mosaic**: A typical Bayer filter mosaic consists of tiled 2×2 blocks of pixels with two green filters and one red and on blue filter. Note that a even-sized Bayer filter, like the one pictured, is asymmetric (mirror flipped version is not equal to itself), while an odd-sized version of this filter pattern would be symmetric.

2.3. Evaluating the Chirality of Operations

We will compare two approaches to evaluating the chirality of an operation $\mathbf{J}$. The first approach, based on the theory we have derived about commutativity, is to evaluate the commutative residual with respect to $\mathbf{J}$ on a small representative set of sample images. The second method, as described in our paper, is to train a neural network to distinguish between flipped and unflipped images sampled from a much larger, symmetric data set that after transforming every image in that dataset by $\mathbf{J}$. In the latter case, it is important that the initial dataset be symmetric or drawn from a symmetric distribution to ensure that any learned chirality can be attributed solely to the effect of $\mathbf{J}$.

3. Demosaicing & JPEG Compression

We evaluate two standard imaging processes: Bayer demosaicing (we consider the method described in [2]), and JPEG compression. We start with a brief summary of these two operations.

Bayer filters and demosaicing. Many modern digital cameras (including cellphone cameras) capture color by means

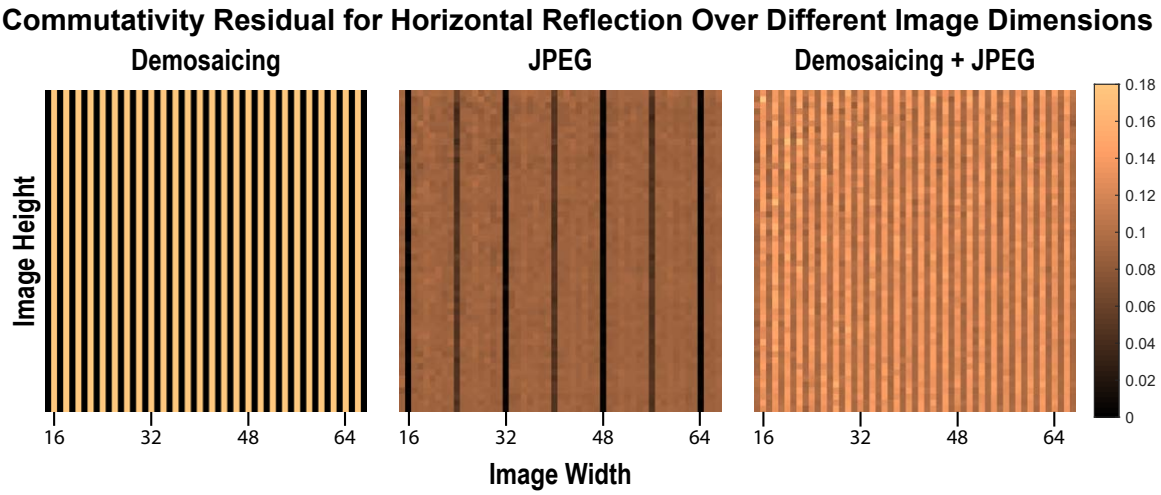


Figure 2. **Commutativity Residuals for demosaicing (left), JPEG compression (middle) and their composition (right)**: Each image shows how commutativity residual, measured in absolute average percent error per pixel, varies with different image sizes. For integers n we see commutativity in demosaicing at image widths of $2n-1$ (i.e., odd widths), and in JPEG compression at widths of $16n$. We do not see commutativity when both are applied.

of a square grid of colored filters that lies atop of the grid of photosensors in the camera. An $8\times$ 8 example of such a color filter grid, known as a Bayer filter mosaic, is shown in Figure 1. In such cameras, each pixel's sensor measures intensity for a single color channel (red, green, or blue), and so to produce a full color image at full resolution, we must interpolate each color channel such that each pixel ultimately has an R, G, and B value. This interpolation process is known as *demosaicing*. For our analysis we assume, as is typical, that a Bayer filter mosaic pattern consists of a tiled 2×2 element (GRBG in the case of Figure 1).

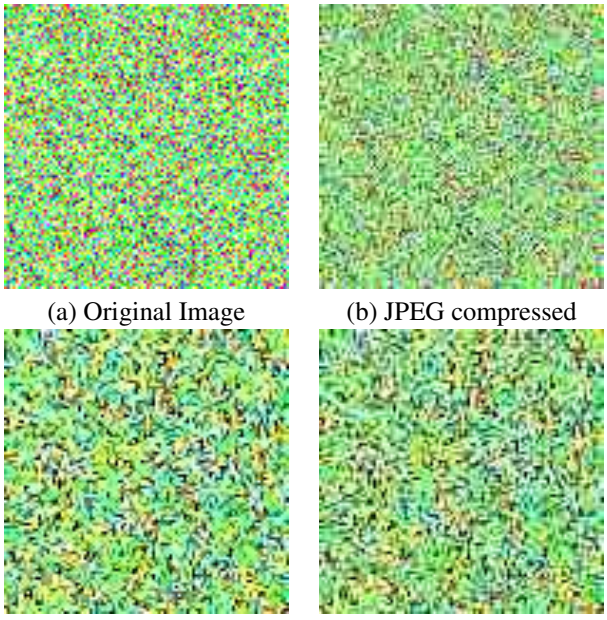
Note that the 8×8 Bayer filter mosaic in Figure 1 has interesting symmetry properties. The 8×8 pattern as a whole is asymmetric—flipping it horizontally will result in a red pixel in the upper-left corner, rather than a green pixel. The same is true for any even-sized Bayer filter mosaic. However, from the perspective of the center of any pixel, the pattern is locally symmetric. Moreover, if we imagine a 9×9 version of this mosaic (or indeed any odd-sized pattern), that mosaic would be symmetric.

JPEG compression. JPEG is one of the most common compression schemes for images. There are two main ways that JPEG compresses image data. First, it converts images into the $\mathrm{Y'C_bC_r}$ colorspace and downsamples the chroma channels ($\mathrm{C_b}$ and $\mathrm{C_r}$), typically by a factor of two. Then it splits each channel into a grid of 8×8 pixel blocks and computes the discrete cosine transform (DCT) of each block. In the luminance ($\mathrm{Y'}$) channel, each block covers an 8×8 pixel region of the original image, while for the chroma channels, each block corresponds to a 16×16 pixel region in the original image, due to the $2\times$ downsampling. Finally, the DCT of each block is strategically quantized to further compress the data at low perceptual cost.

For the purposes of our analysis, one noteworthy aspect of JPEG compression is that for images with dimensions that are not a multiple of 16, there will be boundary blocks that do not have a full 8×8 complement of pixels. These are handled specially by the JPEG algorithm.

3.1. Commutative Residuals and Image Size

If we evaluate commutative residuals on arbitrarily random images for demosaicing, they will be nonzero about half of the time. For JPEG, they will be nonzero over 90% of the time. But if we sample over different image sizes more systematically, a pattern begins to emerge. Figure 2 visualizes the commutative residuals for random noise images as a function of image width and height. We can see that demosaicing appears to be chiral for images with even widths, while JPEG compression seems to be chiral for image widths that are not divisible by 16. We can explain this result by considering the geometry of Bayer patterns and JPEG block grids. Bayer patterns (Figure 1) have horizontal symmetry when reflected about any line centered on a pixel column, while the JPEG block grid, which consists of 8×8 blocks, is horizontally symmetric only around grid lines, which rest between columns at 16-pixel intervals. Figure 2 shows that our black-box analysis of commutative residuals is able to reveal the grid structures underlying both algorithms and show how each grid structure impacts chirality. We can also hypothesize that the combination of demosaicing followed by JPEG compression is chiral because the two processes are never achiral for the same image width (and, indeed, our commutative residual analysis predicts this as well). The chiralities predicted by our analysis for Demosaicing, JPEG,



(a) Original Image (b) JPEG compressed

(c) Demosaicing (d) Demosaicing + JPEG

Figure 3. **A sample image from our Gaussian noise image distribution after different imaging operations**. This image is of size (100,100) and is generated using the Gaussian noise method described in Section 3.1

Imaging Operation	Image size		
	99	100	112
Demosaicing	X	C	C
JPEG	C	C	X
Demosaicing+JPEG	C	C	C

Table 1. **Predicted chirality of three (initially achiral) Gaussian noise image distributions (corresponding to three different square image sizes) under each of three processing schemes.** 'C' means chiral, and 'X' means achiral. Explanation: 99px images should remain achiral under demosaicing, since it is odd size. 112 should remain achiral under JPEG compression since it is divisible by 16. Everything else becomes chiral as discussed earlier. We verify this table empirically by training network models on the nine distributions resulting from these transformations.

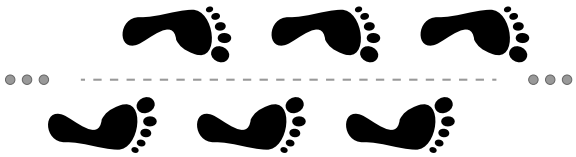


Figure 4. **Glide Symmetry**: Human footprints often exhibit glide symmetry. The infinitely repeating footstep pattern shown here is equivalent to the reflection of a shifted version of itself.

and their composition are summarized in Table 1.

The analysis involved in generating Figure 2 and the predictions of Table 1 was performed in just a few minutes on a laptop using unoptimized MATLAB code. Our hope is that this very quick test will predict the results of a much more expensive and parameter-sensitive learning process. To verify this we compare our predictions against results of the learning task described in our paper. We trained deep network models on three achiral distributions of Gaussian noise images, corresponding to three different square images sizes: one with odd width (99×99), one with even width that is not an integer multiple of 8 (100×100), and one that is a multiple of 16 (112×112). At each pixel of each sample image, the value of each channel was sampled from a Gaussian. For each channel we used a different mean (R:0.6, G:0.5, B:0.9) and standard deviation (R:0.3, G:0.25, B:0.4) to reduce the number of symmetries present other than $\mathbf{T}$. A sample image, before and after processing, can be found in figure 3.

We used the same ResNet model as in main paper (with randomly initialized weights) on the chirality (flip/no-flip) task for each of these nine datasets, performing a grid search for an optimal learning rate using a log scale. As expected, our network model achieved 100% accuracy on distributions that were predicted by our analysis to be chiral, and failed to learn on the distributions our analysis predicts to be achiral (i.e., accuracy was stuck at 50% no matter how we set the hyperparameters). This demonstrates that our analysis was able to predict when an achiral image distribution would become chiral after going through the imaging processes we examined.

Note that this analysis assumes that no cropping has taken place on the images after Bayer demosaicing and/or JPEG compression. Interestingly, these results mirror the situation of training networks on *real* images with no random cropping, as described in the main paper. Figure 6 shows that networks trained to classify chirality on resized (but not cropped) Instagram images often seem to focus on image evidence near boundaries (first row), which we hypothesis is due exactly to the kinds of chiral artifacts discussed in this section. However, training with random cropping data augmentation yields networks that appear to focus on much more high-level features (second row). In the next section, we discuss the interaction of processing with random cropping (or image translation) and how the addition of random cropping can make a chiral imaging process achiral.

4. Random Cropping and Glide Symmetry

One advantage of the more general distribution-based proof provided in Section 2 is that it lets us reason through interactions with other symmetries and data augmentation strategies used in learning. For example, translational invariance is a common and useful prior that is often applied to data through the use of random crops as a type of data augmentation. Doersch *et al*. [1] found that when they trained a network to predict the relative position of different regions in

an image, it would "cheat" by utilizing chromatic aberration for prediction. We can use our observation about commutativity to explain this behavior by considering a family of transformations in the plane. The self-supervision task used in [1] requires the network to distinguish between different translations, which is only possible when the following symmetry does not hold:

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{T}_\mathbf{v}(\mathbf{x})) \tag{10}$$

where $\mathbf{T}_\mathbf{v}$ is translation by some vector $\mathbf{v} \in \mathbb{R}^2$. Equation 5 tells us that this symmetry can be broken by any $\mathbf{J}$ that does not commute with translation. This agrees with the findings of Doersch *et al*. that the network was able to "cheat" using artifacts caused by chromatic aberration, which is not translation-invariant.

4.1. Glide Symmetries

If we revisit our analysis of commutative residuals under an assumption of translation invariance as described by the symmetry in Equation 10, we can draw new conclusions about the chirality of demosaicing and JPEG compression.

First, let's rewrite both sides of Equation 5 using Equation 1 to get:

$$\mathbf{D}_\mathbf{J}(\mathbf{J}(\mathbf{T}(\mathbf{x}))) = \sum_{\{\mathbf{x}_a | \mathbf{J}(\mathbf{x}_a) = \mathbf{J}(\mathbf{T}(\mathbf{x}))\}} \mathbf{D}(\mathbf{x}_a) \tag{11}$$

$$\mathbf{D}_\mathbf{J}(\mathbf{T}(\mathbf{J}(\mathbf{x}))) = \sum_{\{\mathbf{x}_b | \mathbf{J}(\mathbf{x}_b) = \mathbf{T}(\mathbf{J}(\mathbf{x}))\}} \mathbf{D}(\mathbf{x}_b) \tag{12}$$

We see that Equation 5 is trivially true when the commutative residual is zero because the summations in Equations 11 and 12 happen over the same exact elements. However, Equation 5 also holds whenever these summations happen over sets with equal likelihood under $\mathbf{D}$. Therefore, if we assume that $\mathbf{D}$ is invariant to translation, we can substitute elements from either of the summations with translations of those elements without changing the total sum. In other words for image pairs $\{(\mathbf{x}_a, \mathbf{x}_b) | \mathbf{x}_b = \mathbf{T}_\mathbf{v}(\mathbf{x}_a)\}$ Equation 10 gives us $\mathbf{D}(\mathbf{x}_a) = \mathbf{D}(\mathbf{x}_b)$, and we can adapt our notion of commutativity to one of glide-commutativity by considering some translation vector $\mathbf{v}_g \in \mathbb{R}^2$:

$$\mathbf{v}_g(\mathbf{x}) = \underset{\mathbf{v} \in \mathbb{R}^2}{\arg\min} \sum_{\text{pixels}} |\mathbf{J}(\mathbf{T}(\mathbf{x})) - \mathbf{T}(\mathbf{J}(\mathbf{T}_\mathbf{v}(\mathbf{x})))| \tag{13}$$

the *glide-commutative residual image* $\mathbf{E}_{Jg}(\mathbf{x})$:

$$\mathbf{E}_{Jg}(\mathbf{x}) = \mathbf{J}(\mathbf{T}(\mathbf{x})) - \mathbf{T}(\mathbf{J}(\mathbf{T}_{\mathbf{v}_g}(\mathbf{x}))) \tag{14}$$

and the *glide-commutative residual* $\hat{\mathbf{e}}_{Jg}(\mathbf{x})$:

$$\hat{\mathbf{e}}_{Jg}(\mathbf{x}) = \frac{1}{k} \left(|\mathbf{E}_{Jg}(\mathbf{x})| \right) \tag{15}$$

Where k is the number of pixels in $\mathbf{x}$. Glide-commutativity is closely connected to the concept of glide symmetries in geometry. A glide symmetry is one where a figure or infinitely repeating pattern may not be equal to its reflection, but is equal to the reflection of some translated version of itself (see Figure 4).

4.2. Testing for Glide-Commutativity

To test for glide-commutativity, we first define a way of phase-shifting $\mathbf{T}(\mathbf{J}(\mathbf{x}))$ and $\mathbf{J}(\mathbf{T}(\mathbf{x}))$. For this, we define $\mathbf{JT}_\phi(\mathbf{x})$ and $\mathbf{TJ}_\phi(\mathbf{x})$ as the process of:

1. Padding $\mathbf{x}$ with a large, constant number of pixels on all sides.
2. Translating the padded image by ϕ.
3. Applying $\mathbf{T}$ then $\mathbf{J}$ for $\mathbf{JT}_\phi(\mathbf{x})$, or $\mathbf{J}$ then $\mathbf{T}$ for $\mathbf{TJ}_\phi(\mathbf{x})$.
4. Translating by $\mathbf{T}(-\phi)$.
5. Cropping out the previously padded pixels.

This has the effect of performing $\mathbf{J}$ and $\mathbf{T}$ as if the image had occurred at a translation of ϕ from its original position. For grid-based algorithms like demosaicing and JPEG compression, this effectively phase-shifts the grid structure used in the algorithm.

To test for glide-commutativity we simply look for some repeating pattern of zeros in residuals of the form:

$$\mathbf{e}_\mathbf{J}(\mathbf{x}, \phi_1, \phi_2) = \frac{1}{k} \sum_{\text{pixels}} |\mathbf{JT}_{\phi_1}(\mathbf{x}) - \mathbf{TJ}_{\phi_2}(\mathbf{x})| \tag{16}$$

As the results in Figure 2, we verified that the vertical components of ϕ_1 and ϕ_2 do not matter. We therefore set them only to vary in the x dimension of the image. Figure 5 shows the residuals calculated for a range of phase shifts. We see that both demosaicing and JPEG compression appear to be glide-commutative due to the regular repeating pattern of zeros. However, the combination of demosaicing and JPEG compression does not appear to be glide-commutative, and we can see this is because zeros always occur at different phase shifts for each of the two operations.

4.3. Results of Learning With Random Crops

The analysis from the previous section has simple implications (in terms of random cropping on images): (1) The distribution of random crops (while avoiding cropping from the boundary of 16 pixels) from an originally achiral distribution of images that has undergone either demosaicing or JPEG compression (but not both) should remain achiral. (2) On the other hand, surprisingly, random crops (avoiding a 16-pixel margin around the boundary in the cropped image)

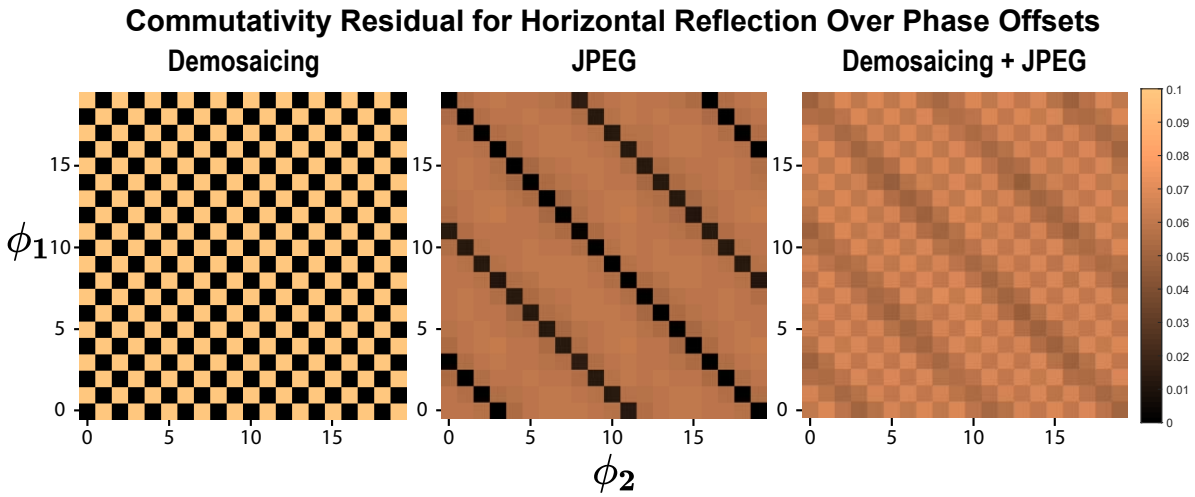


Figure 5. **Glide Commutativity Residuals for demosaicing (left), JPEG compression (middle) and their composition (right)**: Each image shows the glide commutativity residual, measured in absolute average percent error per pixel, measured over different phase shifts. For integers n we see commutativity in demosaicing at image widths of $2n-1$, and in jpeg compression at widths of $16n$. We do not see commutativity when both are applied.

Figure 6. Class Activation Maps (CAM) resulting from two preprocessing procedures used in training ImageNet-pretrained models on the chirality task: (top row) simple bilinear resizing and (bottom row) random cropping. Recall from the main paper that the CAM tends to fire on discriminative regions for classification. Note the heavy focus on edge and corner regions on bilinear resized images, likely due to edge artifacts caused by JPEG compression or demosaicing (or both). These artifacts disappear when random cropping is applied.

on that achiral distribution of images after both demosaicing and JPEG compression may likely become chiral.

To verify this analysis empirically, we again train ResNet models on the same achiral Gaussian distributions as introduced in 3.1. Specifically, we take random crops of size (480, 480) from the center (496, 496) of the (512,512) Gaussian noise images to avoid possible boundary effects from a 16-pixel margin. We train separate networks on each of the three output image distributions obtained from applying each of the three imaging operations (Demosaicing, JPEG compression, and Demosaicing+JPEG compression) on the initial Gaussian noise image distributions. Note that, as before, we perform a log-scale grid search over learning rates.

Our empirical results show that neither Demosaicing nor JPEG compression alone is sufficient to produce a chiral distribution under random cropping: models trained with them stuck at 50% test accuracy. This indicates that chirality has been preserved when those operations are applied

in isolation. But once both of them are applied, the image distribution becomes chiral: the trained network achieves 100% accuracy. This accords with our theoretical analysis of the glide-commutativity under these operations. To further verify the robustness of this experiment, we also generate Gaussian noise images of variable square sizes (e.g., 544, 545, 550px) for testing our 512px image–trained network and evaluated it on random crops (still avoiding a 16-pixel boundary) on such images. In the case of Demosaicing+JPEG compression combined operation, the trained model generalizes to random crops from images of different sizes (i.e. 100% testing accuracy). Together, our analysis and empirical study suggest that chiral traces are left in photographs via the Bayer demosaicing and JPEG compression imaging processes.

References

[1] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. *ICCV*, 2015. 4, 5

[2] K. Hirakawa and T. W. Parks. Adaptive homogeneity-directed demosaicing algorithm. *IEEE Transactions on Image Processing*, 14(3):360–369, March 2005. 2