Counterfactual Visual Explanations

Yash Goyal [1] **Ziyan Wu** [2] **Jan Ernst** [2] **Dhruv Batra** [1] **Devi Parikh** [1] **Stefan Lee** [1]

Abstract

In this work, we develop a technique to produce *counterfactual visual explanations*. Given a 'query' image I for which a vision system predicts class c, a counterfactual visual explanation identifies how I could change such that the system would output a different specified class c'. To do this, we select a 'distractor' image I' that the system predicts as class c' and identify spatial regions in I and I' such that replacing the identified region in I with the identified region in I' would push the system towards classifying I as c'. We apply our approach to multiple image classification datasets generating qualitative results showcasing the interpretability and discriminativeness of our counterfactual explanations. To explore the effectiveness of our explanations in teaching humans, we present machine teaching experiments for the task of fine-grained bird classification. We find that users trained to distinguish bird species fare better when given access to counterfactual explanations in addition to training examples.

1. Introduction

When we ask for an explanation of a decision, either implicitly or explicitly we do so expecting the answer to be given with respect to likely alternatives or specific unselected outcomes – *"For situation X, why was the outcome Y and not Z?"* A common and useful technique for providing such discriminative explanations is through counterfactuals – i.e. describing what changes to the situation would have resulted in arriving at the alternative decision – *"If X was X*, then the outcome would have been Z rather than Y."* As computer vision systems achieve increasingly widespread and consequential applications, the need to explain their decisions in arbitrary circumstances is growing as well – for example, questions of safety *"Why did the self-driving car misidentify the fire hydrant as a stop sign?"* or fairness

[1]Georgia Institute of Technology [2]Siemens Corporation. Correspondence to: Yash Goyal <ygoyal@gatech.edu>.
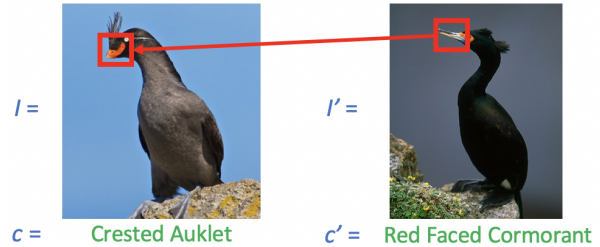
Figure 1. Our approach generates counterfactual visual explanations for a query image I (left) – explaining why the example image was classified as class c (*Crested Auklet*) rather than class c' (*Red Faced Cormorant*) by finding a region in a distractor image I' (right) and a region in the query I (highlighted in red boxes) such that if the highlighted region in the left image looked like the highlighted region in the right image, the resulting image I^* would be classified more confidently as c'.

"Why did the traveler surveillance system select John Doe for additional screening?" will need answers.

While deep learning models have shown unprecedented (and occasionally super-human) capabilities in a range of computer vision tasks (Deng et al., 2009; Huang et al., 2007), they achieve this level of performance at the cost of becoming increasingly inscrutable compared to simpler models. However, understanding the decisions of these deep models is important to guide practitioners to design better models, evaluate fairness, establish appropriate trust in end-users, and to enable machine-teaching for tasks where these models have eclipsed human performance.

As these systems are increasingly been deployed in real world applications, interpretability of machine learning systems (particularly deep learning models) has become an active area of research (Simonyan et al., 2013; Springenberg et al., 2015; Ribeiro et al., 2016; Adebayo et al., 2018; Doshi-Velez & Kim, 2017). Many of these works have focused on identifying regions in an input image that most contributed to the final model decision (Springenberg et al., 2015; Ribeiro et al., 2016; Simonyan et al., 2013; Selvaraju et al., 2017; Zhang et al., 2016) – i.e. producing explanations via feature attribution. However, these approaches do not consider alternative decisions or identify hypothetical adjustments to the input which could result in different outcomes – i.e. they are neither discriminative nor counterfactual.

In this work, we study how such **counterfactual visual explanations** can be generated to explain the decisions of

deep computer vision systems by identifying what and how regions of an input image would need to change in order for the system to produce a specified output. Consider the example in Figure 1, a computer vision system identifies the left bird as *Crested Auklet*. A standard feature attribution explanation approach may identify the bird's crown, slender neck, or colored beak as important regions on this image. However, when considering an alternative such as the *Red Faced Cormorant* (shown right) many of these key regions would be shared across both birds. In contrast, our approach provides a counterfactual explanation by identifying the beak region in both images – indicating that if the bird on the left had a similar beak to that on the right, then the system would have output *Red Faced Cormorant*.

More concretely, given a query image I for which the system predicts class c, we would like to generate a faithful, counterfactual explanation which identifies how I could change such that the system would output a specified class c'. To do this, we select a distractor image I' which the system predicts as class c' and identify spatial regions in I and I' such that replacing the identified region in I with the identified region in I' would push the system towards classifying I as c'. We formalize this problem and present greedy relaxations that sequentially execute such region edits.

We apply our approach to SHAPES (Andreas et al., 2016), MNIST (LeCun et al., 1998), Omniglot (Lake et al., 2015) and the Caltech-UCSD Birds (CUB) 2011 (Wah et al., 2011) datasets generating qualitative results showcasing the intepretability and discriminativeness of our counterfactual explanations. Our design of SHAPES dataset also enables us to quantitatively evaluate our generated explanations. The simplistic nature of MNIST and Omniglot images allows us to generate an imagination of how the query image would "look" like if the discriminative region in the query image is replaced by the discriminative region in the distractor image. For CUB, we present an analysis of our counterfactual explanations utilizing segmentation and keypoint annotations present in the dataset which shows that our explanations highlight discriminative parts of the birds.

In addition to being more discriminative and interpretable, we think this explanation modality is also compelling from a pedagogical perspective, an important and relatively less studied application for interpretability. Good teachers explain why something is a particular object and why it's not some other object. Similarly, these counterfactual explanations can be useful in the context of machine teaching, *i.e.* AI teaching humans, especially for tasks where AI systems outperform untrained humans.

We apply our approach to a fine-grained bird classification task in which deep models perform significantly better than untrained humans on the Caltech-UCSD Birds (CUB) 2011 dataset (Wah et al., 2011). We hypothesize that our counterfactual visual explanations from a deep model trained for

this task can help in teaching humans where in the image to look at (*e.g.* neck, beak, wings, *etc*. of the bird) in order to identify the correct bird category. For example, for the birds shown in Figure 1, most people not specifically trained in bird recognition would not know the difference between these two birds. But, given a counterfactual visual explanation from our approach "If the highlighted region in the left image looked like the highlighted region in the right image, the left image would look more like a *Red Faced Cormorant*", an untrained human is more likely to learn the differences between these two birds as compared to only showing example images for both birds.

To explore the effectiveness of our explanations in teaching humans the fine-grained differences between birds, we designed a human study where we train and test humans for this task of bird classification. Through our human studies, we found that users trained to discern between bird species fare better when given access to counterfactual explanations in addition to training examples.

In summary, we make the following contributions: we

1. propose an approach to generate counterfactual visual explanations, *i.e.* what region in the image made the model predict class c instead of class c'?
2. show that our counterfactual explanations from deep models can help in teaching humans via human studies.

2. Approach

In order to explain a query image I relative to a distractor I' under some trained network, we seek to identify the key discriminative regions in both the images such that replacing these regions in I' with those in I would lead the network to change its decision about the query to match that of the distractor. In Section 2.1 we formalize this problem and then present two greedy solutions: exhaustive search (Section 2.2) and a continuous relaxation (Section 2.3).

2.1. Minimum-Edit Counterfactual Problem

Consider a deep convolutional network taking as input an image $I \in \mathcal{I}$ and predicting log-probability output $\log P(\mathbf{Y}|I)$ over the output space $\mathbf{Y}$. For the sake of this discussion and without loss of generality, we will consider a decomposition of the network into two functional components – a spatial feature extractor and a decision network – as shown in Figure 3. First, $f : \mathcal{I} \rightarrow \mathbb{R}^{h*w \times d}$ maps the image to a $h \times w \times d$ dimensional spatial feature which we reshape to a $hw \times d$ matrix, where h and w are the spatial dimensions and d is the feature size (*i.e.* number of channels). Second, $g : \mathbb{R}^{hw \times d} \rightarrow \mathbb{R}^{|\mathbf{Y}|}$ takes this feature and predicts log-probabilities over output classes $\mathbf{Y}$. We can then write the network as a whole as $\log P(\mathbf{Y}|I) = g(f(I))$. For notational convenience, we let $g_c(f(I))$ denote the log-probability of class c for image I under this network.
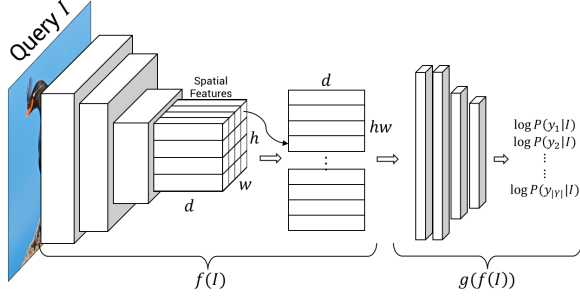
Figure 3. We decompose a CNN as a spatial feature extractor $f(I)$ and a decision network $g(f(I))$ as shown above.

Given a query image I for which the network predicts class c, we would like to produce a counterfactual explanation which identifies how I could change such that the network would output a specified distractor class c'. However, the space of possible changes to I is immense and directly optimizing pixels in I to maximize $\log P(c'|I)$ is unlikely to yield interpretable results (Tyka, 2015). Instead, we consider changes towards a distractor image I' which the network already predicts as class c'.

Given these two images, we would like to make a transformation T from I to $I^* = T(I, I')$ such that I^* appears to be an instance of class c' to the trained model $g(f(\cdot))$. One natural way of perform this transformation is by replacing regions in image I with regions in image I'. However at the extreme, we could simply set $I^* = I'$ and replace I entirely. To avoid such trivial solutions while still providing meaningful change, we would like to apply a minimality constraint on the number of transferred regions. Hence, our approach tries to find the minimum number of region replacements from I' to I to generate I^* such that the trained model classifies I^* as an instance of class c'. We call this the *minimum-edit counterfactual problem*. Rather than considering the actual image regions themselves, we consider the spatial feature maps $f(I), f(I') \in \mathbb{R}^{hw \times d}$ corresponding to image regions.

We formalize this transformation as depicted in Figure 2. Let $P \in \mathbb{R}^{hw \times hw}$ be a permutation matrix that rearranges the spatial cells of $f(I')$ to align with spatial cells of $f(I)$ and let $a \in \mathbb{R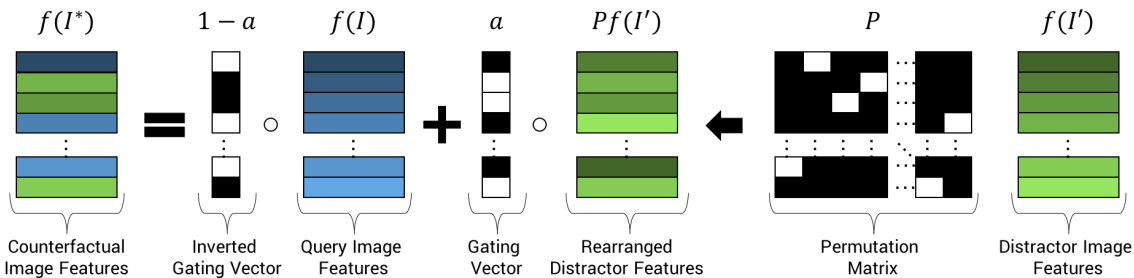}^{hw}$ be a binary vector indicating whether to replace each spatial feature in image I with spatial features from image I' (value of 1) or to preserve the features of I (value of 0). We can then write the transformation from I to I^* in spatial feature space $f(*)$ as

$$f(I^*) = (\mathbb{1} - \mathbf{a}) \circ f(I) + \mathbf{a} \circ P f(I') \qquad (1)$$

where $\mathbb{1}$ is a vector of all ones and $\circ$ represents the Hadamard product between a vector and a matrix obtained by broadcasting the vector to match the matrix's dimensions and then taking the Hadamard product between the broadcasted vector and the matrix. Note that as $\mathbf{a}$ is a binary vector, minimizing its norm corresponds to minimizing the number of edits from I' to I.

With this notation in hand, we can write the *minimum-edit counterfactual problem*, *i.e.* minimizing the number of edits to transform I to I^* such that the predicted class for the transformed image features $f(I^*)$ as defined in Eq. 1 is the distractor class c', as the following:

$$\underset{P, \mathbf{a}}{\text{minimize}} \quad ||a||_1$$
$$\text{s.t.} \quad c' = \operatorname{argmax} \ g((\mathbb{1} - \mathbf{a}) \circ f(I) + \mathbf{a} \circ P f(I'))$$
$$a_i \in \{0, 1\} \ \forall i \ \text{and} \ P \in \mathcal{P}$$
$$(2)$$

where $\mathcal{P}$ is the set of all $hw \times hw$ permutation matrices. Given the resulting $\mathbf{a}$ and P, we can extract the set of pairs of spatial cells involved in the edits as $\mathcal{S} = \{(i, j, i', j')\}) \mid a_{i*j} = 1 \wedge P_{i*j, i'*j'} = 1\}$.

After optimization, the resulting vector $\mathbf{a}$ provides the discriminative attention map on image I indicating which spatial cells in I were edited with features copied from I' and P_{i^*}, where $i^* = \operatorname*{argmax}_i a_i$, provides the discriminative attention map on the distractor image I' indicating which source cells those features were taken from.

Solving this problem directly is quite challenging – requiring identifying the minimum subset of $hw * hw$ possible edits that changes the model's decision. To put this in perspective, there are $O((h * w)^{2+k})$ such subsets of size k *i.e.* k cells in I being replaced by k cells in I'. Even for modest feature sizes of $h{=}w{=}16$, this quickly scales over a million candidates for $k = 2$.

Figure 2. To parameterize our counterfactual explanations, we define a transformation that replaces regions in the query image I with those from a distractor I'. Distractor image features $f(I')$ are first rearranged with a permutation matrix P and then selectively replace entries in $f(I)$ according to a binary gating vector a. This allows arbitrary spatial cells in $f(I')$ to replace arbitrary cells in $f(I)$.
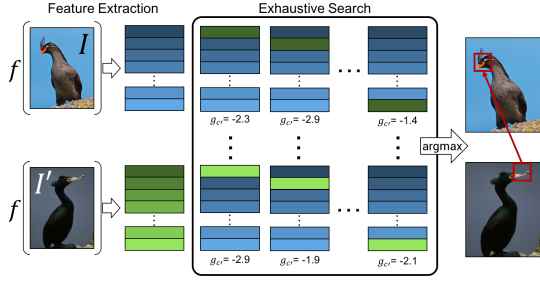
Figure 4. In our exhaustive best-edit search, we check all pairs of query-distractor spatial locations and select whichever pair maximizes the log probability of the distractor class c'.

In the following sections, we present two greedy sequential relaxations – first, an exhaustive search approach keeping $\mathbf{a}$ and P binary and second, a continuous relaxation of $\mathbf{a}$ and P that replaces search with an optimization.

2.2. Greedy Sequential Exhaustive Search

Rather than solve Eq. 2, we consider greedily making edits to I until the cumulative effect changes the model's decision. That is to say, we sequentially find single edits that maximizes the gain in output log-probability $g_{c'}(\cdot)$ for c'. We can write this *best-edit* sub-problem as

$$\underset{P,a}{\text{maximize}} \quad g_{c'}((\mathbb{1} - \mathbf{a}) \circ f(I) + \mathbf{a} \circ P f(I'))$$
$$\text{s.t.} \quad ||\mathbf{a}||_1 = 1, \; a_i \in \{0,1\} \; \forall i \quad\quad (3)$$
$$P \in \mathcal{P}$$

where $\mathbf{a}$ is a binary vector and P is a permutation matrix as before. Rather than minimizing $||\mathbf{a}||_1$ as in Eq. 2, we instead constrain it to be one-hot – indicating the edit in I which maximizes the model log-probability $g_{c'}$.

One straight-forward approach to solving Eq. 3 is through exhaustive search – that is to say, evaluating $g_{c'}$ after replacing features for each of the $h * w \times h * w$ cells in $f(I)$ with those of each of the cells in $f(I')$. As shown in Fig. 4, one step of our exhaustive search approach consists of first replacing the features in one spatial cell of $f(I)$ by features of one spatial cell in $f(I')$, and then passing these modified convolutional features through the the rest of the classifier network $g(.)$ to compute the log-probability of the distractor class c'. We then repeat this procedure for all permutations of cell locations in $f(I)$ and $f(I')$. The pair of cells that result in the greatest log-probability for the distractor class c' are the most discriminative spatial cells in $f(I)$ and $f(I')$.

In order to approximately minimize the objective in Eq. 2 *i.e.* the number of edits, we run this search sequentially multiple times (excluding previously selected edits) until the predicted class changes from c to c', i.e. $g_{c'} > g_c$. We outline this procedure in Algorithm 1.

This procedure requires evaluating $g(\cdot)$ $O(h^2 w^2 k)$ times where k is the average number of edits before the decision

---

Algorithm 1 Greedy Sequential Search

Data: query image I with class c, distractor I' with class c'
Result: list of edits $\mathcal{S}$ that change the model decision
$\mathcal{S} \leftarrow [\,] \quad F^* \leftarrow f(I) \quad F' \leftarrow f(I')$
/* Until decision is changed to c' */
while $c' \neq argmax\ g(F^*)$ **do**

    /* Find single best edit excluding
       previously edited cells in $\mathcal{S}$ */
    $i, j' \leftarrow \text{BestEdit}(F^*, F', \mathcal{S})$

    /* Apply the edit and record it */
    $F^*_{i,*} = F'_{j',*}$
    $\mathcal{S}.append(\{i, j'\})$

end

---

changes. In the next subsection, we provide a continuous relaxation of the best-edit problem amicable to gradient based solutions – resulting in fewer evaluation calls on average.

2.3. Continuous Relaxation

We formulate the best-edit problem defined in Eq. 3 as a tractable optimization problem by relaxing the constraints.

First, we loosen the restriction that $\mathbf{a}$ be binary – allowing it to instead be a point on the simplex (i.e. non-negative and summing to one) corresponding to a distribution over which cells in $f(I)$ to edit. Second, we allow a similar softening of the constraints on P, restricting it to be a right stochastic matrix (i.e. non-negative with rows $\mathbf{p_i}^T$ summing to one) corresponding to distributions over cells in $f(I')$ to be copied from. We write this relaxed objective as:

$$\underset{P,a}{\text{maximize}} \quad g_{c'}((\mathbb{1} - \mathbf{a}) \circ f(I) + \mathbf{a} \circ P f(I'))$$
$$\text{s.t.} \quad ||a||_1 = 1, \; a_i \geq 0 \; \forall i \quad\quad (4)$$
$$||p_i||_1 = 1 \quad \forall i, \; P_{i,j} \geq 0 \; \forall i, j$$

To always satisfy the constraints in Eq. 4, we reparameterize $\mathbf{a}$ and P in terms of auxilliary variables α and M respectively. Specifically, we define $\mathbf{a} = \sigma(\alpha)$ and $\mathbf{p_i}^T = \sigma(\mathbf{m_i}^T)$ where $\sigma(\cdot)$ is the softmax function: $a_i = \frac{e^{\alpha_i}}{\sum_j e^{\alpha_j}}$. In this way, the non-negativity and unit norm constraints on $\mathbf{a}$ and P are ensured while we are free to optimize α and M unconstrained via gradient descent. We use gradient descent with a learning rate of 0.3.

In this soft version, cells in $f(I')$ can be copied to more than one location or copied fractionally through non-binary entries in P or $\mathbf{a}$; however, by applying entropy losses on $\mathbf{a}$ and rows of P (minimizing their entropy), we can recover a nearly binary solution for $\mathbf{a}$ and the rows of P.

We apply this approach as the *best-edit* search procedure in lieu of exhaustive search presented in Section 2.2 – iteratively selecting the best-edit until the decision changes.

To summarize our approach, we defined a formulation of

minimum-edit counterfactual problem and introduced two greedy sequential relaxations – an exhaustive search approach in Sec. 2.2 and a continuous relaxation in Sec. 2.3.

3. Related Work

Visual Explanations. Various feature attribution methods have been proposed in the recent years which highlight "important" regions in the input image which led the model to make its prediction. Many of these approaches are gradient based (Simonyan et al., 2013; Springenberg et al., 2015; Selvaraju et al., 2017), using backpropagation-like techniques and upsampling to generate visual explanations. Another type of feature attribution methods are reference based approaches (Fong & Vedaldi, 2017; Dabkowski & Gal, 2017; Zintgraf et al., 2017; Dhurandhar et al., 2018; Chang et al., 2018), which focus on the change in classifier outputs with respect to perturbed input images *i.e.* input images where parts of the image have been masked and replaced with various references such as mean pixel values, blurred image regions, random noise, outputs of generative models, etc. In similar spirit, a concurrent work Chang et al. (2018) use a trained generative model to fill in the masked image regions from the unmasked regions. More relevant to our approach, Dhurandhar et al. (2018) find minimal regions in the input image which should be necessarily present/absent for a particular classification. But, all the above works focus on generating visual explanations that highlight regions in an image which made the model predict a class c. On the other hand, we focus on a more specific task of generating counterfactual visual explanations that highlight what and how regions of an image would need to change in order for the model to predict a distractor class c' instead of the predicted class c.

Counterfactual Explanations. To tackle a similar task as ours, Hendricks et al. (2018) learn a model to generate a counterfactual explanation for why a model predicted a class c instead of class c'. But our approach is different from theirs in 3 significant ways: 1) their explanation is in natural language while our explanation is visual, 2) their approach requires additional attribute annotations while ours doesn't, and 3) their explanation is the output of a separate learned model (raising concerns regarding its faithfulness to the target model's prediction) while our explanation is directly generated from the target model based on the receptive field of the model's neurons and, hence, is faithful by design.

Machine Teaching. Machine teaching (Zhu, 2015) works have mainly focused on how to show examples to humans so that they can learn a task better. Many of these approaches focus on selecting or ordering examples to be shown to humans in order to maximize their information gain. As deep learning models achieve superhuman performance in some tasks, it is natural to ask if they can in turn act as instructors to help improve human ability. To our knowledge,

only one other work (Mac Aodha et al., 2018) uses visual explanations for machine teaching. They use saliency map explanations generated from Zhou et al. (2016) along with heuristics to select good examples to be shown to human learners. To compare with an equivalent setting of this work to ours, we ran a baseline human study with explanations generated from GradCAM (Selvaraju et al., 2017), which has been shown to generate more discriminative visual explanations as compared to Zhou et al. (2016).

4. Experiments

We apply our approach on four different datasets – SHAPES (Andreas et al., 2016) (in supplement due to space constraints), MNIST (LeCun et al., 1998), Omniglot (Lake et al., 2015) and Caltech-UCSD Birds (CUB) 2011 Dataset (Wah et al., 2011), and present results showcasing the intepretability and discriminativeness of our counterfactual explanations.

Common Experimental Settings In all our experiments, we operate in output space of the last convolutional layer in the CNN but our approach is equally applicable to the output of any convolutional layer. Further, all qualitative results shown are with the exhaustive search approach presented in Section 2.2 as we are operating on relatively small images.

In our experiments on CUB (Sec. 4.3), we find the continuous relaxation presented in Sec. 2.3 achieves identical solutions to exhaustive search for 79.98% of instances and on average achieves distractor class probability that is 92% of the optimal found via exhaustive search – suggesting its usefulness for larger feature spaces.

4.1. MNIST

We begin in the simple setting of hand-written digit recognition on the MNIST dataset (LeCun et al., 1998). This setting allows us to explore our counterfactual approach in a domain well-understood by humans.

We train a CNN model consisting of 2 convolutional layers and 2 fully-connected layers on this dataset. This network achieves 98.4% test accuracy – note this is well below state-of-the-art but this is not important for our purposes. Under this model, the size of spatial features is $4 \times 4 \times 20$.

Qualitative Results. We examine counterfactual explanations for randomly selected distractor class c' and corresponding image I'. Sample results are shown in Fig. 5.

The first two columns show the query and distractor images and highlight the *best-edit* regions. We produce this highlight based on the receptive field of the convolutional feature selected through our approach. The third column depicts a composite image generated in pixel space by aligning and superimposing the highlighted region centers We note that our approach operates in the convolutional feature space and we present this composite as visualization.
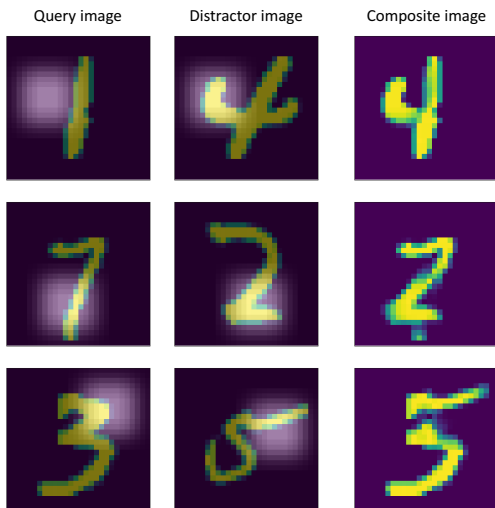
Figure 5. Results on MNIST (LeCun et al., 1998) dataset. The first two columns show the query and distractor images, each with their identified discriminative region highlighted. The third column shows composite images created by making the corresponding replacement in pixel space.
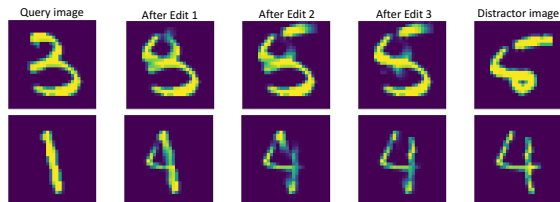

Figure 6. Examples of multiple edits on MNIST digits.

For the first example (first row), our approach finds that that if the left side stroke (the highlighted region) in the distractor image (2nd column) replaced the highlighted region (background) in the query image, the query image would be more likely to belong to the distractor class '4'. As we can see in the composite, the resulting digit does appear to be a '4'. As a reminder, the network was not trained to consider such transformation, rather our approach is identifying the key discriminative edits. In the third example, our approach finds that if the upper curve of the '3' in the query image instead looked like the horizontal stroke of the '5' in the distractor image, the query image would be more likely to belong to the distractor class '5'.

Quantitative Analysis. On average, it takes our approach 2.67 edits to change the model's prediction from c to c'. Examples with multiple (3) edits are shown in Fig. 6. As edits are taken greedily, often the first edit makes the most significant change (second row); however, for complex transformation like $3 \rightarrow 5$ (top row) multiple edits are needed. Our approach takes 15 μs per image on a Titan XP GPU.

4.2. Omniglot

We move on to the Omniglot dataset (Lake et al., 2015) containing images of hand-written characters from 50 writing systems. Like MNIST, these images are composed of
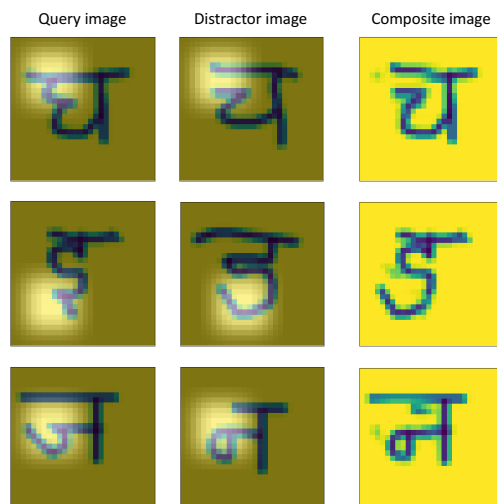

Figure 7. Qualitative results on the Omniglot dataset.

simple pen strokes; however, most humans are not going to a priori know the difference between characters. Hence, Omniglot is an ideal "mid-way point" between our MNIST and CUB experiments.

We experiment with the 'Sanskrit' writing system consisting of 42 characters with 20 images each. We created a random train/test split of 80/20% to train the classification model. We use the same architecture as in the MNIST experiments, resizing the Omniglot images to match. This network achieves 66.8% test accuracy.

Qualitative Results. As before, we examine single-edit counterfactual explanations for randomly selected distractors. Qualitative results are shown in Fig. 7. Our approach finds appropriate counterfactual edits to shift the character towards the distractor even given their complex shape.

Quantitative Analysis. On average, it takes our approach 1.46 edits to change the model's prediction from c to c'. Runtime is 9 μs per image on a Titan XP GPU.

4.3. Caltech-UCSD Birds (CUB)

Finally, we apply our approach to the Caltech-UCSD Birds (CUB) 2011 dataset (Wah et al., 2011) consisting of 200 bird species. It is one of the most commonly used datasets for fine-grained image classification and can be challenging for non-expert humans. Consequentially, we use this dataset for our machine teaching experiments.

We trained a VGG-16 (Simonyan & Zisserman, 2015) model on this dataset, which achieves 79.4% test accuracy. The size of this feature space is 7 x 7 x 512.

Given an image I with the predicted class c, we consider two ways of choosing the distractor class c' – random classes different from c and nearest neighbor classes of c in terms of average attribute annotations (provided with the dataset). The latter helps us in creating pairs of images (I, I') which are very similar looking to each other. We sample the distractor image I' with the predicted class c' in two ways –

Query image	Distractor image	Composite image

Eared Grebe · Horned Grebe

Olive sided Flycatcher · Myrtle Warbler

Blue Grosbeak · Indigo Bunting

Northern Fulmar · Glaucous winged Gull
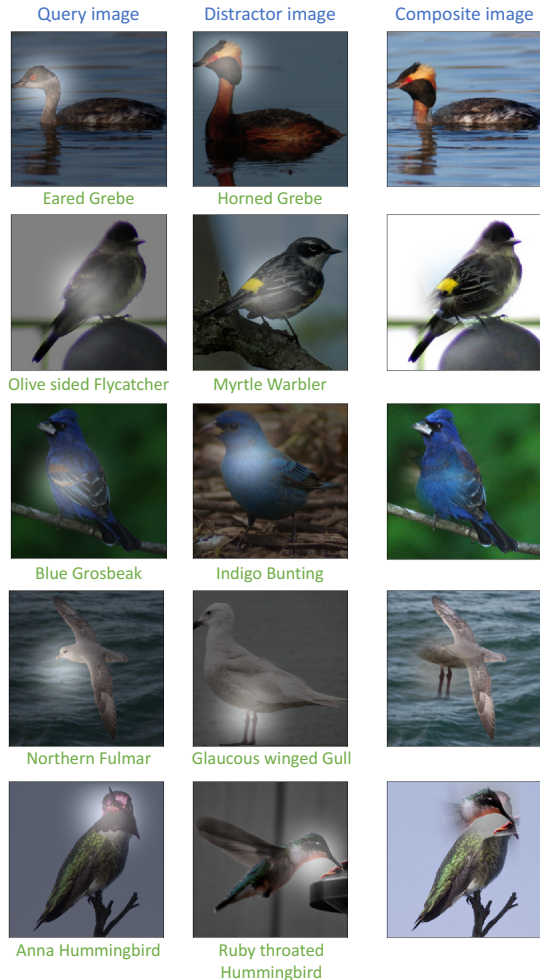
Anna Hummingbird · Ruby throated Hummingbird

Figure 8. Qualitative results on CUB. Our counterfactual explanation approach highlights important attributes in the birds such as head plumage, yellow wing spots and texture on the wings.

random images and nearest neighbor images of I in terms of keypoint locations of the bird (provided with the dataset). The latter helps us in creating pairs of images (I, I') in which the birds are in similar orientation to each other.

Qualitative Results. As before, single-edit qualitative results are shown in Fig. 8. The first three examples depict success cases where our approach identifies important features like head plumage, yellow wing spots, and wing coloration. In these cases, the simple composite visualization is quite telling. However, the bottom two rows show less interpretable results. In the fourth row, the query bird's head is replaced by the long legs of the distractor bird – perhaps in an attempt to turn the bird shape around as shown in the composite. The fifth row seems to correctly identify the need to recolor the neck of the query bird; however, the composite looks poor due to pose misalignment.

Our explanations can also be helpful in debugging a model's mistakes. Such examples are shown in Fig. 9. In the first example, the query image is incorrectly classified as *Bronzed*

Query image	Distractor image	Composite image

Bronzed Cowbird · Red winged Blackbird

Ringed Kingfisher · Green Kingfisher

Figure 9. Qualitative results where the model's predictions are incorrect. Our counterfactual explanations with respect to the correct class highlights important attributes of the correct class which are not clearly visible in the query images such as red wing spot and texture on the wings.

Cowbird instead of *Red winged Blackbird* probably because the distinct feature of the correct class (a red spot on the wing) is not clearly visible. When an explanation is generated with respect to an image of the correct class, our approach copies over the red spot in order to increase the score of the correct class. Similarly, in the second example, our approach highlights the distinct texture on the wings which is not clearly visible in the query image.

Quantitative Analysis. On average, it takes our approach 7.4 edits to change the model's prediction from c to c' if c' is a random distractor class while it takes on average 5.3 edits if c' is a nearest neighbor in terms of attributes. Runtimes are 1.85 and 1.34 sec/image for random and NN distractor classes respectively on a Titan XP GPU.

To check the degree of dependence of our explanations on the choice of the distractor image, we compute "agreement" in the most discriminative spatial cell locations *i.e.* outputs of the best-edit subproblem in image I for different distractor images with the same predicted class c'. This agreement is 78% (a high value) implying that our approach highlights similar regions in image I for different choices of distractor image I' from class c'. Similarly, we compute "agreement" in the most discriminative spatial cell locations in image I for different distractor classes c'. This agreement is 42% (a low value) implying that our explanations on image I differ based on the choice of the distractor class.

The CUB dataset also comes with dense annotations of bird regions and parts which we use to further analyze our approach. First, we compute how often our discriminative regions lie inside the bird segmentations and find that regions in both the images lie inside the bird segmentation 97% of the times.

Further, we compute how often our discriminative regions lie near the important keypoints of birds such as neck, crown, wings, legs, *etc.* provided with the dataset. After running
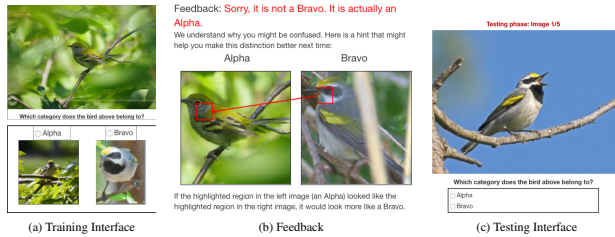
Figure 10. Our machine teaching interface. During training phase (shown in (a)), if the participants choose an incorrect class, they are shown a feedback (shown in (b)) highlighting the fine-grained differences between the two classes. At test time (shown in (c)), they must classify the birds from memory.

our explanation procedure until the model decision changes, we find they are near the keypoints of the bird 75% of the times in the query image I and 80% of the times in the distractor image I'. Our discriminative regions also highlight the same keypoint in both the birds 20% of the times. This shows that our approach often replaces semantically meaningful keypoints between birds – indicating that the underlying CNN has likely picked up on these keypoints without explicit supervision.

5. Machine Teaching

We apply our approach on CUB 2011 dataset (Wah et al., 2011) to guide humans where to look to distinguish bird categories. Towards this goal, we built a machine teaching interface for human subjects shown in Figure 10.

Since it is a hard task for humans, the interface consists of two phases– Training and Testing. In both the phases, we show the subjects an image and 2 bird category options denoted Alpha and Bravo, and ask them to assign the shown bird images to one of these categories. During the Training phase, each of the category option is accompanied by an example image for subjects to compare against the presented image. Our training interface is shown in Fig. 10a.

If they choose an incorrect category, we show them some feedback. The feedback shows the discriminative attention maps generated by our approach using an image of the distractor class (Alpha if the correct class is Bravo and vice-versa) closest to the query image in keypoint space as the distractor image. The feedback in our interface is shown in Fig. 10b.

After the training phase, we test the human subjects. During the testing phase, they are shown an image and the bird category options, but without the example images. As in training, they have to select the option they think best fits the query test image. No feedback is given during testing. Our test interface is shown in Fig. 10c.

We compare this human study with two baselines which only differ in terms of the feedback shown to human subjects. In the first baseline case, the feedback only con-

veys the information that the subject chose the incorrect category. In the second and harder baseline, in place of our counterfactual explanations, the feedback shows a non-counterfactual, feature-attribution explanation generated via GradCAM (Selvaraju et al., 2017) highlighting the region in the image most salient for the predicted category.

The studies were conducted in smaller sessions. Each session consisted of teaching a participant 2 bird classes with 9 training and 5 test examples. We conducted all studies with graduate students studying Machine Learning (ML) because of the level of difficulty of the task. Total 26 graduate students participated, each participant (on average) worked on 6.2 sessions and spent 3 min 52 sec per session.

The mean test accuracy with counterfactual explanations is 78.77%, with GradCAM explanations, it is 74.29% while the mean test accuracy without any explanations is 71.09%. We find that our approach is statistically significantly better than the no-explanation baseline at a 87% confidence level, and than the GradCam baseline at 51% – implying the need for further study to assess if counterfactual explanations improve machine teaching compared to feature attribution approaches. To examine the effect of a participant's familiarity to ML on their performance, we conducted a small study with 9 participants with no knowledge of ML, each working on 5 sessions. The test accuracy is 61.7% without explanations and 72.4% with our counterfactual explanations, trends consistent with the previous human study. Overall, this shows that counterfactual explanations from deep models can help teach humans by pointing them to appropriate parts to identify the correct bird category.

6. Conclusion

In this work, we present an approach to generate counterfactual visual explanations – answering the question "How should the image I be different for the model to predict class c' instead?" We show our approach produces informative explanations for multiple datasets. Through a machine teaching task on fine-grained bird classification, we show that these explanations can provide guidance to humans to help them perform better on this classification task.

References

Adebayo, J., Gilmer, J., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. In *NeurIPS*, 2018.

Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. Neural module networks. In *CVPR*, 2016.

Chang, C.-H., Creager, E., Goldenberg, A., and Duvenaud, D. Explaining image classifiers by counterfactual generation. *arXiv preprint arXiv:1807.08024*, 2018.

Dabkowski, P. and Gal, Y. Real time image saliency for black box classifiers. In *NIPS*, 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

Dhurandhar, A., Chen, P.-Y., Luss, R., Tu, C.-C., Ting, P., Shanmugam, K., and Das, P. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *NeurIPS*, 2018.

Doshi-Velez, F. and Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Fong, R. C. and Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*, 2017.

Hendricks, L. A., Hu, R., Darrell, T., and Akata, Z. Grounding visual explanations. In *ECCV*, 2018.

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

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Mac Aodha, O., Su, S., Chen, Y., Perona, P., and Yue, Y. Teaching categories to human learners with visual explanations. In *CVPR*, 2018.

Ribeiro, M. T., Singh, S., and Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *SIGKDD*, 2016.

Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., and Batra, D. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. In *ICCV*, 2017.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for Simplicity: The All Convolutional Net. In *ICLR Workshop Track*, 2015.

Tyka, M. Inceptionism: Going deeper into neural networks, Jun 2015. URL https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Zhang, J., Lin, Z., Brandt, J., Shen, X., and Sclaroff, S. Top-down Neural Attention by Excitation Backprop. In *ECCV*, 2016.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. Learning deep features for discriminative localization. In *CVPR*, 2016.

Zhu, X. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *AAAI Conference on Artificial Intelligence*, 2015.

Zintgraf, L. M., Cohen, T. S., Adel, T., and Welling, M. Visualizing deep neural network decisions: Prediction difference analysis. In *ICLR*, 2017.