Improving Invariance and Equivariance Properties of Convolutional Neural Networks

Christopher Tensmeyer & Tony Martinez
Department of Computer Science
Brigham Young University
Provo, UT 84602, USA
tensmeyer@byu.edu
martinez@cs.byu.edu

Abstract

Convolutional Neural Networks (CNNs) learn highly discriminative representations from data, but how robust and structured are these representations? How does the data shape the internal network representation? We shed light on these questions by empirically measuring the invariance and equivariance properties of a large number of CNNs trained with various types of input transformations. We find that CNNs learn invariance wrt all 9 tested transformation types and that invariance extends to transformations outside the training range. We also measure the distance between CNN representations and show that similar input transformations lead to more similar internal representations. Transforms can be grouped by the way they affect the learned representation. Additionally, we also propose a loss function that aims to improve CNN equivariance.

1 Introduction

The overwhelming success of Convolutional Neural Networks (CNNs) is generally attributed to their ability to learn task-specific representations from large quantities of data. This has led to many state of the art results in areas such as image classification (He et al., 2015), semantic segmentation (Liu et al., 2015), and game-playing Go agents (Silver et al., 2016). However, our current understanding of CNN representations is limited, though there is a growing body of literature on visualization techniques for interpreting internal representations (Mahendran & Vedaldi, 2016; Dosovitskiy & Brox, 2015; Nguyen et al., 2016).

In this work, we aim to understand the effect that the training data has on three key properties of representations: invariance, equivariance, and equivalence (Lenc & Vedaldi, 2015). Invariance, with respect to a particular transformation (e.g. rotation by $\theta = 5$), is achieved when the feature vectors for input images at a particular layer do not change when the inputs are transformed. This can be seen as a measure of the robustness of the representation. Equivariance is a generalization of invariance and allows the representation to change in predictable ways in response to input transformations. Equivariance is a rough measure of the structure of the representation space because we can reason about input space transformations in the more abstract representation space. Likewise, measuring equivalence among network representations trained on either the same or different data distributions gives insight into how the training data shapes the representation.

Data augmentation has been considered essential for top CNN performance since the seminal work of Krizhevsky et al. (2012). When input images are stochastically perturbed, less overfitting is observed because all inputs are unique. Since then, others have experimented with various data augmentation strategies with great success (Howard, 2013; Wu et al., 2015; He et al., 2014). While others have noted that data augmentation leads to greater representation invariance (Lenc & Vedaldi, 2015; Wu et al., 2015; Peng et al., 2014), we measure the amount of invariance achieved in response to 9 types of transforms at various magnitudes. We additionally measure the equivariance and pairwise representation distance of the resulting CNNs.

In this work we quantify the invariance and equivariance properties of 70 CNNs on two datasets trained using different input transformations. We find the representation at the penultimate layer (fc7) is structured wrt almost all input transformations when no augmentation is used. Applying data augmentation effectively collapses the structure of the representation space, leading to input transformation invariance for all 9 transforms, including magnitudes of transforms that were not observed during training.

We measure the pairwise distances of the CNN representations by measuring the error of learned mappings between CNN representations. There is a strong bias towards similarity among CNNs trained with the same type of transformation, and we can group transformations based on their mutual average distances. Similar types of transformations (e.g. Color Jitter, Gaussian Noise) yield more similar representations compared to other dissimilar transform pairs.

We also propose a way to increase the equivariance of a CNN by finetuning on a novel loss function that simultaneously minimizes classification error for both transformed and untransformed inputs. This leads to an increase in equivariance while improving or slightly decreasing performance on the untransformed images (dataset dependent), though this trade-off can be controlled by weighting the terms of the loss function.

2 Related Works

Lenc & Vedaldi (2015) studied image representations and gave definitions for representation invariance, equivariance, and equivalence. They measured these properties wrt a limited number of transforms for the convolutional layers of the popular AlexNet architecture. We provide extensions to the definitions of these properties to measure the relative degree to which representations possess these properties. We also consider a wider variety of transformations and use the *fc7* layer of AlexNet because it is the most invariant layer.

Convergent learning is the idea that identical networks (differing in random initialization) converge to the same representation (i.e.. representation equivalence). In Li et al. (2015), the authors show that there are many corresponding convolutional filters among CNNs for one-to-one, one-to-many, and many-to-many relationships. In complementary fashion, we measure representation distances between last fully connected layers in CNNs trained over different transformations of their inputs.

In Krizhevsky et al. (2012), applying random crops, horizontal flips and color jittering during training reduced the top-1 error by over 1%. Howard (2013) applied additional brightness, contrast, crop, and scale transformations at both training and test time to reduce validation top-1 error from 40.7% to 37.0%. Wu et al. (2015) adds rotation and photo filter effects such as lens distortion, vignetting, and color casting for further improvement. Similarly, we explore a series of 10 types of transformations applied at training time; however, we focus on how they affect the learned representation instead of optimizing classifier performance.

In Peng et al. (2014), synthetic images rendered from 3D CAD models are used to explore invariance in object detection networks for factors such as pose, object texture, and foreground/background color. Training a CNN on images that vary in these aspect results in greater invariance than training with less data variety. 3D CAD models are also used in Aubry & Russell (2015) to show that upper layers of CNNs disentangle and (locally) linearize independent object factors such as object viewpoint, style, color, and scale.

ManiTest (Fawzi & Frossard, 2015) defines classifier invariance as the average magnitude of the minimal transform that causes predictions to change. In contrast, we examine internal CNN representations and measure both representation distance and classifier performance.

3 Invariance, Equivariance, and Equivalence

In this section, we provide definitions and intuition on the three properties of interest, borrowing basic definitions and notation from Lenc & Vedaldi (2015). Though we focus on CNN image representations, the following applies to any kind of data representation.

3.1 Equivariance and Invariance

If the domain of interest (e.g. natural images) is $X \subset \mathbb{R}^{nxn}$, a function $\phi : \mathbb{R}^{nxn} \to \mathbb{R}^m$ assigns a feature vector to every input image and thus defines an image representation. We will use X to denote the input space and Z the output space (i.e. $\phi : X \rightarrow Z$). ϕ may be *equivariant* to some transformations, but not others, so we say that ϕ is equivariant wrt $g : X \rightarrow X$ iff

$$\exists\, M_g : Z \rightarrow Z,\ \forall x \in X :\ \phi(gx) \approx M_g \phi(x) \tag{1}$$

This means that transformation by g in X corresponds to a transformation by M_g (if it exists) in Z. Thus if M_g exists, it informs us of the structure of the abstract space Z in terms of the concrete space X. If M_g is the identity transform, then ϕ maps x and gx to the same point in Z, which leads to *invariance* wrt g (i.e. $\phi(gx) \approx \phi(x)$). This is related to the idea of whether CNNs collapse (invariance) or linearize (equivariance) view manifolds of 3D objects (Bakry et al., 2015; Aubry & Russell, 2015).

While the previous definitions suggest equivariance and invariance are all-or-nothing properties, we use the following definition to precisely define the $\approx$ used in Eq 1. We measure the equivariance of ϕ wrt a transform g as

$$\text{Equivaraince}_\phi(g; L) = \min_{M_g} \frac{1}{N} \sum_{i=1}^{N} L(\phi(x_i), M_g \phi(g x_i)) \tag{2}$$

where L is a suitable loss function such as classification error or L2 distance (Section 3.3). Invariance is similarly measured by fixing M_g as the identity function.

In practice, it is difficult to maximize M_g over the space of all functions, so we restrict M_g to be a simple parametric function (e.g. linear) and learn the parameters from data. Thus our results can be considered a lower bound on the actual equivariance of the representation. Eq. 2 also uses the paradigm that M_g attempts to undo in Z the transformation g performed in X. This is also done in Lenc & Vedaldi (2015) and allows us to compute classification based metrics by applying the same classifier to both $\phi(x)$ and $M_g\phi(gx)$.

3.2 Equivalence and Representation Distance

In Lenc & Vedaldi (2015), ϕ_1 is equivalent to ϕ_2 if ϕ_1 has the same information in the sense that

$$\exists\, E_1 : Z_1 \rightarrow Z_2,\ \forall x \in X : \phi_2(x) \approx E_1 \phi_1(x) \tag{3}$$

However, this relation is only symmetric (as the name *equivalence* implies) iff E_1 is invertible. If E_1 is invertible, then E_1^{-1} satisfies $\phi_1(x) \approx E_1^{-1}\phi_2(x)$. In the other direction, if the relation is symmetric, then $\exists E_2$ such that $\phi_1(x) \approx E_2\phi_2(x)$ and such an E_2 is an inverse of E_1.

In light of this, we propose renaming Eq 3 to be the *sub-representation* property with the the terminology that ϕ_2 is a sub-representation of ϕ_1. We then define ϕ_1 and ϕ_2 to be *equivalent* iff an invertible E_1 exists. One example of sub-representation are two reprensetations characterized by (1) the RGB pixels of an image and (2) the corresponding grayscale intensities. The grayscale is a sub-representation of the RGB because we can recover the grayscale from the RGB, but not vice versa.

As with equivariance, we are interested in measuring the extent that two representations are distinct. Thus we define the distance between two representations as

$$D(\phi_1, \phi_2) = \min_{E_1}[L(E_1\phi_1(x),\, \phi_2)] + \min_{E_2}[L(\phi_1,\, E_2\phi_2(x))] \tag{4}$$

As with M_g, finding the optimal E_1, E_2 is difficult, so we learn linear models from data. Choices for the loss function, L, are discussed in Section 3.3.

3.3 Distance Metrics

The two distance metric we employ are normalized euclidean distance in the representation space and classification error.

$$L_{L2}(r_i, r_i') = \frac{||r_i - r_i'||_2}{||r_i||_2}, \quad L_{err}(r_i, r_i') = 1 - \delta(y_i, \hat{y}_i),\ \ \hat{y}_i := \underset{j}{argmax}\, C(r_i')_j$$

where $r_i = \phi(x_i)$, $r_i' = M_g(gx_i)$, $C(r_i)$ is the predicted distribution of class labels, y_i is the ground truth label of r_i, and δ is the Kronecker delta function. For L_{L2}, we normalize each dimension of r to have similar magnitude, similar to the normalization of Li et al. (2015). The classifier, C, is composed of the remaining layers of the CNN after ϕ.

Name	Description
Baseline	No Transform
Color Casting	Add a random integer to each color channel
Crop	Crop a 227x227 sub-window from a larger image
Elastic Deformation	Apply a smoothed random displacement field (Simard et al., 2003)
Gaussian Blur	Blur the image with a Gaussian kernel
Gaussian Noise	Add iid Gaussian noise to each pixel
Mirror	Flip image horizontally, vertically, or both
Perspective	Warp square image to an arbitrary quadrilateral
Rotation	Rotate the image about the center
Shear	Warp square image into trapezoid (horizontal and vertical)

Table 1: Each CNN is trained with one of these types of transformations over parameter ranges that differ in magnitude. Each sampled input to the CNN is stochastically perturbed by sampling transform parameters from the range and applying the transformation to the image.

4 Improving Equivariance

In this section, we propose a loss function for training a network to increase its invariance and equivariance. CNNs for classification tasks are typically trained with a cross entropy loss, i.e.

$$L_d(X,Y) = \frac{1}{N}\sum_{i=1}^{N} H(y_i, C(\phi(x_i))) \tag{5}$$

where X is the training images, Y is the training labels, and H computes cross entropy.

L_d can be augmented by a term that biases the representation ϕ to be equivariant to some set of transforms $G = \{g_1, g_2, ..., g_J\}$. We propose the loss function, L_e, which is minimized when the transformed images are both correctly classified and spatially close to the untransformed image in Z.

$$L_e(X,Y) = \frac{1}{NJ}\sum_{i=1}^{N}\sum_{j=1}^{J}\Big[H(y_i, C(x'_{ij})) + \lambda_1||x'_{ij} - \phi(x_i)||_2^2 + \lambda_2||\phi(g_j x_i) - \phi(x_i)||_2^2\Big] \tag{6}$$

where $x'_{ij} = M_{g_j}\phi(g_j x_i)$. The first term of the sum is the classification loss of the transformed images, while the second and third terms respectively enforce equivariance and invariance. In our experiments, we set $\lambda_1 = \frac{50}{4096}$, $\lambda_2 = \frac{25}{4096}$, though the results do not appear highly sensitive to this particular setting. Each equivariance mapping M_{g_j} constitutes a new portion of the network that attempts to undo the transformation g_j in the Z. As such, the parameters of M_{g_j} must be learned along with the parameters of the network.

Combining Eq. 5 and 6, we arrive at our combined loss:

$$L(X,Y) = L_d(X,Y) + \lambda_3 L_e(X,Y) \tag{7}$$

We found that $\lambda_3 = 0.3$ provides good compromise between performance on the transformed and untransformed data, though λ_3 can be tuned for individual performance requirements.

5 Experimental Setup

Here we detail the experiments we conduct including the transformations, networks, and datasets.

For the first set of experiments, we simply measure the invariance and equivariance of the AlexNet[1] architecture trained under various data augmentation schemes. We measure the representation of layer fc7 (denoted ϕ_{fc7}) after applying the ReLU non-linearity and dropout[2].

[1] We used the reference CaffeNet architecture, which differs slightly from Krizhevsky et al. (2012)

[2] We follow the test time convention of halving neuron activations instead of zeroing out half the activations.

5.1 Transformations

We examine a set of 10 transformation types. See Table 1 for brief descriptions or Appendix A for full details. We train 4 CNNs for each type of transform (only 2 for blurring, 1 for baseline) for a total of 35 networks per dataset. Each of the 4 CNN within a transform type is trained with different magnitudes of transformations. Training with data augmentation is performed by stochastically transforming each network input with a transform taken from a specified range.

After training, we measure invariance and equivariance wrt fixed transforms for each type. For example, the 4 CNNs trained on different ranges of rotations ($\theta \in [-5, 5]$, $\theta \in [-10, 10]$, $\theta \in [-15, 15]$, $\theta \in [-20, 20]$) all have their equivariance measured wrt rotations of $\theta = \pm\{2.5, 5, 10, 15, 20, 25, 30, 40\}$. As a baseline, we trained a model with no data augmentation and measured the invariance and equivariance of that network wrt all transforms.

5.2 Equivariance Mapping

To measure equivariance wrt g_j, we learn M_{g_j} (Eq. 2) and measure the classification accuracy over the transformed images. Due to data augmentation during training, the network may already be invariant wrt g_j, so we model M_{g_j} as a linear residual mapping to bias it towards an identity mapping (He et al., 2015). That is $M_{g_j}(r) = \sigma(r + R(r))$, where R is to be estimated from data and $\sigma(x) = max(x, 0)$ is the ReLU non-linearity. We experimented with R as a linear mapping and as a neural network with a single hidden layer. In nearly all cases, the linear mapping outperformed the neural network, so we present results only for the linear mappings. Therefore, we have

$$M_{g_j}(r) = \sigma(r + W_j r + b_j) = \sigma((W_j + I)r + b_j) \tag{8}$$

where (W_j, b_j) are parameters to be estimated from data.

Each M_{g_j} is trained using fully online SGD for 10 epochs with a static learning rate of 0.001, using an $L2$ weight decay penalty of 10^{-5}. Momentum with $\beta = 0.9$ is used. For learning M_{g_j}, only (W_j, b_j) are updated. The rest of the network parameters are treated as fixed. The loss to be minimized is $||M_{g_j}(\phi_{fc7}(g_j x)) - \phi_{fc7}(x)||_2^2$.

5.3 Representation Distance Mapping

Here we are interested in learning E_1 (or equivalently E_2) from Eq. 4 for each ordered pair (ϕ_i, ϕ_j) in a set of of network representations $\{\phi_i\}$. We consider linear mappings with $E_1(r) = \sigma(Wr + b)$ and estimate (W, b) from data. Hyper-parameters for learning are the same as in Section 5.2. The loss to be minimized for each ordered pair is $||\phi_i(x) - E_1 \phi_j(x)||_2^2$.

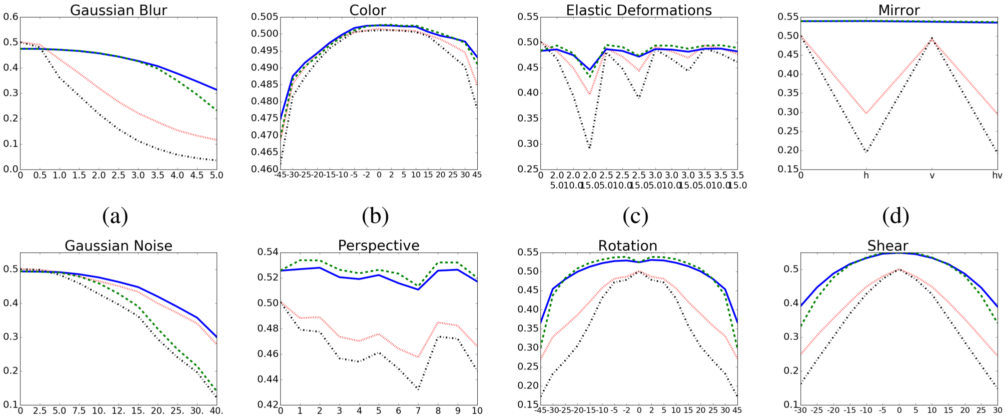
5.4 Finetuning Equivariance

For this experiment, we take the CNNs trained with the most extreme parameter ranges for each transform type and finetune the representation using Eq. 7 as the loss function. For each CNN, we selected 4-6 transformations within the training range to be the G in Eq. 6.

We finetune for 150,000 weight updates with a mini-batch size of 10 for RVL-CDIP and 100 for ILSVRC. The learning rate is 0.0005 and decays by a factor of 10 every 60,000 updates. Momentum of $\beta = 0.9$ was used with no weight-decay regularization.

5.5 Datasets

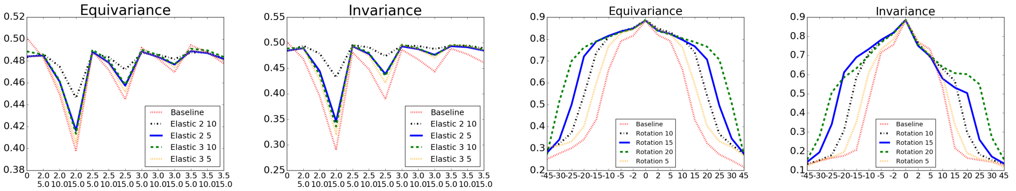
We use two large datasets with distinct properties. The first is the popular ILSVRC 2012 dataset (1.2M train / 50K validation), composed of natural images. The second is the RVL Complex Document Information Processing (RVL-CDIP) dataset (Harley et al., 2015) (320K train / 40K val / 40K test). It is composed of scanned tobacco litigation documents in grayscale format and each document image is labeled with one of 16 categories (e.g. letter, memo, email, form, news article, scientific publication). Examples images can be found in Figure 4 in Appendix A In contrast to ILSVRC, document images in RVL-CDIP are intrinsically 2D objects, have fixed zoom and location, and have little background area. We aim to compare and contrast the invariance and equivariance of the various transforms for these two datasets.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1: Invariance and equivariance accuracy measurements for ILSVRC. Each plot compares a CNN trained with a particular transformation (blue line for equivariance, green for invariance) to the baseline model (red for equivariance, black for invariance) trained with no transformations. The x-axis ranges over different parameters for each type of transformation. Each point represents the error of the M_g learned on that particular parameterization of the transformation.



(a) ILSVRC Elastic (b) ILSVRC Elastic (c) RVL-CDIP Rotation (d) RVL-CDIP Rotation

Figure 2: Equivariance and Invariance measurements of CNNs trained with various parameter ranges. Larger training ranges yields greater robustness for a wider range of transforms.

In our experiments CNNs are trained using the training splits, using the validation split for model selection. For learning equivariance mappings and representation distance mappings, we use a static randomly chosen 50K training images for ILSVRC and the validation split for RVL-CDIP. Thus the reported metrics are over the validation split for ILSVRC and test split for RVL-CDIP. We subsample for ILSVRC for computational reasons and use the validation split for RVL-CDIP because the CNN representation may have overfit the training data.

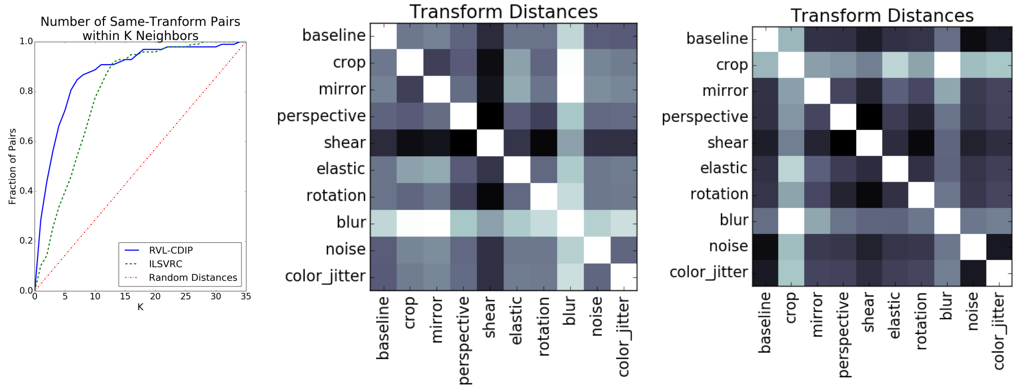
6 Results

6.1 Measuring Invariance and Equivariance

Figure 1 shows the invariance and equivariance measurements for each type of transform on ILSVRC. Additional results for both ILSVRC and RVL-CDIP can be found in Appendix B.1. For each transform type, we compare the CNN trained with the most extreme transforms to the baseline model trained with no data augmentation. The transformations we measured range from no transform to approximately double the transformation magnitude seen during training.

In many instances (e.g. Blur, Noise), the baseline model's performance deteriorates rapidly even for mild transformations. The large difference between the baseline invariance (black line) and equivariance (red line) indicate that Z is structured wrt most transforms (Color is one exception).

For CNNs trained with data augmentation, the equivariance mapping does not improve performance for transforms that are observed during training. This suggests that Z may not be (linearly) struc-



(a) KNN Plot

(b) ILSVRC

(c) RVL-CDIP

Figure 3: (a) shows how often transforms of the same type are K-nearest neighbors. It reveals a strong bias towards networks trained with the same transformation having more similar representations. Heatmaps (b) and (c) show average distances between representations induced by the transforms. Lighter squares indicate greater distance, while darker squares indicate the transforms are closer together. Diagonal entries are distance 0, but were set to white for visualization purposes.

tured wrt those particular transforms possibly because the structure has been collapsed due to the training object. Some transforms (e.g. Noise, Blur, Shear), at large magnitudes, do show a gap between the equivariance and invariance lines, suggesting that Z is structured wrt these transforms. Equivariance performance does drop off outside the training range, but not near as sharply as the baseline model (e.g Figs. 7a, 7c, 7f), so CNN representations generalize somewhat to unseen transforms.

Figure 2 shows how various magnitudes of training augmentation affect the invariance/equivariance properties of the network for select transforms. Other transforms behaved similarly. Larger training ranges yields greater robustness for a wider range of transforms.

6.2 Representation Distance

In this second experiment, we measured the pairwise representation distances of 37 CNNs (34 w/ transforms and 3 baseline networks). Using equation 4 ($L = L_{L2}$), we computed a pairwise distance matrix for the 37 network representations. We attempted a T-SNE embedding (Maaten & Hinton, 2008) visualization, but the distances seem intrinsically difficult to embed in 2D (see Appendix B.3). Comments on the asymmetry of the distances measured can be found in Appendix B.3.

There is, however, a strong bias towards networks of the same training transformation to be nearer to each other compared to random chance. We performed a K-NN analysis by counting the percentage of same-transform pairs that are K nearest neighbors (Figure 3a). Approximately 80% of same-transform pairs are within 5 neighbors of each other for RVL-CDIP (10 neighbors for ILSVRC). This indicates that networks trained with the same transform end up with more similar representations, though the strength of the transformation and network initialization also play a role.

We also visualize the average distance between CNNs grouped by transformation type (Figs 3c and 3b). Patterns emerge in both datasets. In RVL-CDIP, Crop is the most unique transform because the CNNs learn features as a different static image scale. On the other hand, Crop is not as unique for ILSVRC because objects appear at multiple sizes, so the features learned are not targeted at a single size. Blur, especially for ILSVRC, seems to give unique representations because it changes the local textures from which the representation is derived in a bottom-up fashion. Refining the representation in a top-down manner may be a further avenue for improving CNN representation robustness.

Rotation, Shear, and Perspective transforms are mutually similar for both datasets. This is likely because all three move local pixels in a rigid fashion. Another mutually similar group is Baseline, Color Jitter, and Gaussian Noise, which all operate on pixels independently. Elastic Deformations are somewhat similar to Shear, but not to other transforms.

Transform	CDIP Original	CDIP Finetune	Transform	CDIP Original	CDIP Finetune
Color Jitter	88.01	88.58	Crop center from 256x256	88.83	89.07
brightness +10	88.01	88.63	upper left corner	86.46	87.07
brightness -15	88.04	88.60	bottom right corner	86.94	87.57
Elastic Deformations	87.89	88.52	Gaussian Noise	88.15	88.67
$\sigma = 2.5, \alpha = 5$	88.17	88.48	$\sigma = 8$	88.16	88.61
$\sigma = 3, \alpha = 10$	88.05	88.21	$\sigma = 16$	88.09	88.53
Gaussian Blur	86.82	87.18	Mirror	88.42	89.44
$\sigma = 1$	86.79	87.13	Horz.	88.51	88.11
$\sigma = 2$	86.47	86.74	Horz. + Vert.	88.43	88.17
Perspective	88.55	89.13	Rotation	88.49	89.01
	88.69	88.96	$\theta = -10$	87.60	87.41
	88.65	89.07	$\theta = 15$	88.59	88.70
Shear	88.96	89.71			
Horz, $\theta = -10$	88.37	88.55			
Vert, $\theta = -15$	87.40	87.51			

Table 2: Equivariance measurements for several CNNs on RVL-CDIP before and after finetuning using Eq. 7. The first row of each transform type shows performance over untransformed images. In most cases finetuning improves equivariance on the finetuning transforms and on the untransformed images.

6.3 Improved Equivariance

For RVL-CDIP, finetuning using Eq. 7 improves both equivariance and performance on untransformed images by ∼0.5% (Table 2). This may be because the original network was trained with a wide range of transform parameters, so the representation caters to no particular set of transform parameters. By introducing M_{g_j} during finetuning, the representation can individualize to the untransformed image, while still avoiding the overfit that occurs without data augmentation. Results for ILSVRC were mixed, with finetuning improving equivariance for 3 of 9 transformations (see Table 4 in Appendix B).

7 Conclusion

This work gives the results from a large scale empirical study into the invariance and equivariance properties of CNNs trained under different input transformations. We quantify these properties for 70 CNNs across 10 types of transforms for 2 large datasets. CNNs are able to learn invariance to all transforms tried, and this invariance/equivariance extends somewhat to transforms outside the training range. We show evidence that the CNN linearizes its representation wrt Rotation, Perspective, and Shear transforms. In general, the baseline CNN showed very little invariance to even moderate transformations.

We also measured CNN representation distance between all pairs of 37 networks. There is a bias towards CNNs trained with the same type of transformation to have more similar representations. The analysis also revealed that similar types of transforms (e.g. Rotation, Shear) lead to more similar representations. We also proposed a joint loss function that moderately increases accuracy on both untransformed and transformed images, leading to an increase in equivariance for most transforms.

References

Pulkit Agrawal, Ross Girshick, and Jitendra Malik. Analyzing the performance of multilayer neural networks for object recognition. In *European Conference on Computer Vision*, pp. 329–344. Springer, 2014.

Mathieu Aubry and Bryan C Russell. Understanding deep features with computer-generated imagery. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2875–2883, 2015.

Amr Bakry, Mohamed Elhoseiny, Tarek El-Gaaly, and Ahmed Elgammal. Digging deep into the layers of cnns: In search of how cnns achieve view invariance. *arXiv preprint arXiv:1508.01983*, 2015.

Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. *arXiv preprint arXiv:1506.02753*, 2015.

Alhussein Fawzi and Pascal Frossard. Manitest: Are classifiers really invariant? In *British Machine Vision Conference (BMVC)*, number EPFL-CONF-210209, 2015.

Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pp. 991–995. IEEE, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pp. 346–361. Springer, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Andrew G Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 991–999, 2015.

Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent learning: Do different neural networks learn the same representations? *arXiv preprint arXiv:1511.07543*, 2015.

Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1377–1385, 2015.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Aravindh Mahendran and Andrea Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, pp. 1–23, 2016.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*, 2016.

Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. Exploring invariances in deep convolutional neural networks using synthetic images. *CoRR, abs/1412.7122*, 2(4), 2014.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Patrice Y Simard, David Steinkraus, and John C Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pp. 958–962, 2003.

Ren Wu, Shengen Yan, Yi Shan, Qingqing Dang, and Gang Sun. Deep image: Scaling up image recognition. *arXiv preprint arXiv:1501.02876*, 7(8), 2015.

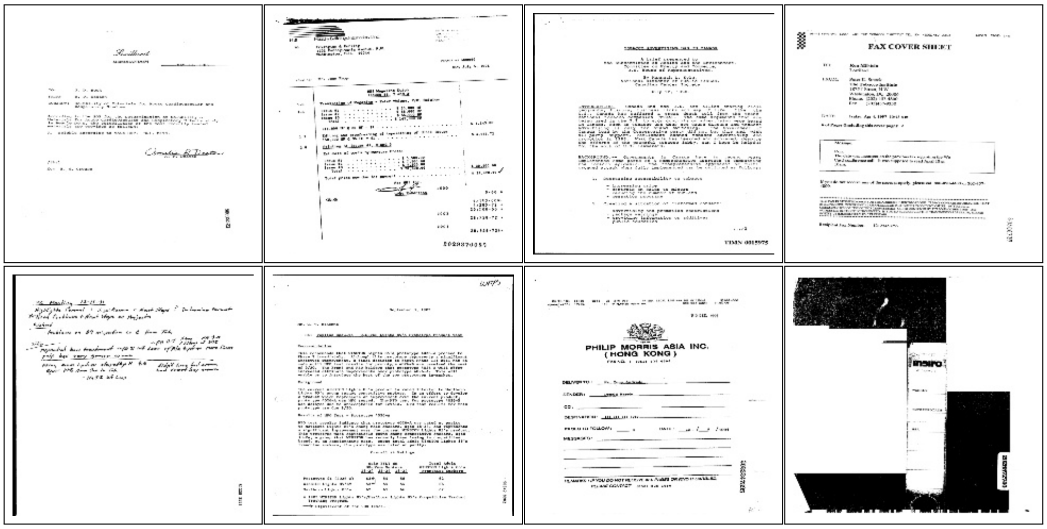


Figure 4: Example instances of the RVL-CDIP dataset.

APPENDIX

A TRANSFORM DETAILS

For space reasons, exact details and parameterizations of the transforms used to train the CNNs were omitted from the main text, but are included here. Table 1 gives a brief explanation of the 9 transforms, but each transform will be explained in detail hereafter. Pixels that move into the image dimensions from outside the image (e.g. pixels rotated in) are set to be intensity 0 (see Figure 6b).

A.1 COLOR JITTER

Color Jitter adds a random value to each color channel (RGB or grayscale) of the input image. The random value drawn is separate for each channel, but the same value is applied to each spatial location, essentially making each color either brighter or darker. If the addition of the random value causes a pixel to go outside the normal $[0, 255]$ range, it is truncated back to the range.

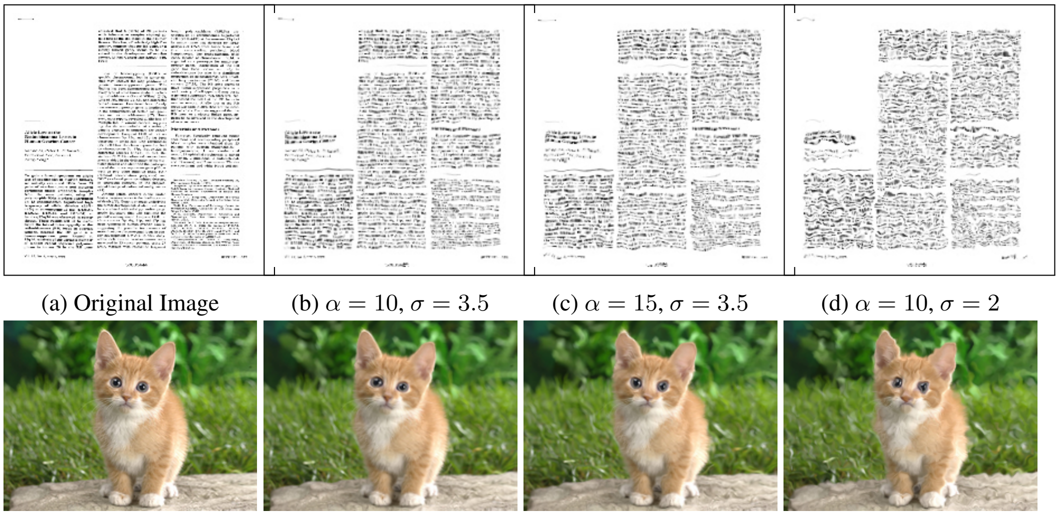
During CNN training, the random values are drawn from a Gaussian Distribution parameterized by a mean μ and standard deviation σ. Four CNNs were trained, with $\mu = 0$ and $\sigma \in \{5, 10, 15, 20\}$.

When measuring invariance/equivariance, we apply a deterministic color jitter transformation that is the same for all input images. We simply adjust the brightness of the image (all color channels tied) by a fixed amount in $\pm\{2, 5, 10, 15, 20, 25, 30, 45\}$.

A.2 CROP

During training with Crop transformations, we first resize the input images to a larger size (such as 240x240, 256x256, 288x288, or 320x320), and then take a 227x227 crop from the larger imgae. The larger the original image, the more detail the CNN gets to see, but also a smaller percentage of the original image is captured in the input window. CNNs trained with other transformations have their input images resized to 227x227.

For measuring invariance/equivariance, the baseline transform for the Crop CNNs is the center crop, and the transforms are crops at other locations. For example, for invariance to upper left corner crops, we measured the difference in CNN activations between the network applied to the center crop and the network applied to the upper left crop. We measured 25 spatial locations arranged in a 5x5 grid.



(a) Original Image (b) $\alpha = 10, \sigma = 3.5$ (c) $\alpha = 15, \sigma = 3.5$ (d) $\alpha = 10, \sigma = 2$

(e) Original Image (f) $\alpha = 10, \sigma = 3.5$ (g) $\alpha = 15, \sigma = 3.5$ (h) $\alpha = 10, \sigma = 2$

Figure 5: Example Elastic Deformations transforms for which we measured invariance and equivariance in Figure 7c.

A.3 Elastic Deformations

Elastic Deformations are a way of locally distorting images by applying a smoothed random displacement field (Simard et al., 2003). Image transformations can be characterized by a backward mapping of pixel locations in the output image to locations in the input image. For elastic deformations, first an random displacement field is sampled that would map output pixels to random input locations in their local neighborhood. The size of the local neighborhood is controlled by an α parameter. This displacement field is then smoothed using Gaussian filtering with a specified σ. This causes local regions of the displacement field to point in the same direction.

For training CNNs, we used a fixed σ and sampled α for each input image. The four CNNs we trained with Elastic Deformations can be described as $(\alpha \in [0, 5], \sigma = 2)$, $(\alpha \in [0, 10], \sigma = 2)$, $(\alpha \in [0, 5], \sigma = 3)$, $(\alpha \in [0, 10], \sigma = 3)$.

When measuring equivariance/invariance (in contrast to training) we apply the same exact transform to every image. For elastic deformations, this means we applied the same displacement field to each image. We did this for a number of displacement fields of various parameter settings. As shown in Figure 7c These settings included every combination of $\alpha \in \{5, 10, 15\}$ and $\sigma \in \{2, 2.5, 3, 3.5\}$.

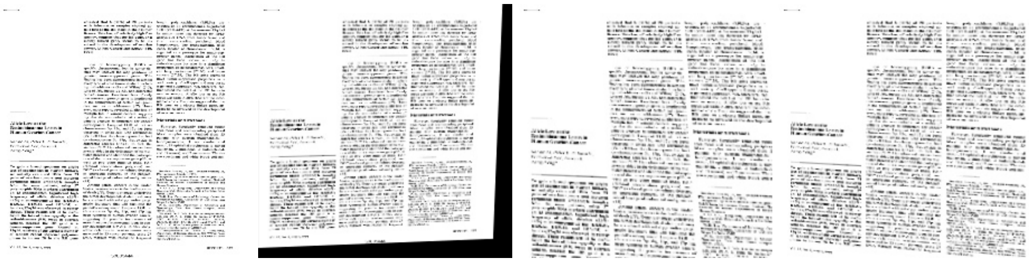
A.4 Gaussian Blur

We used the standard Gaussian Blur transform which replaces each pixel by a weighted average of the neighboring pixels, where weight magnitudes are Gaussian wrt spatial distance. The shape of the Gaussian weighting function is controlled by σ, which is measured in pixels. We trained two CNNs with Gaussian Blur, where for each input image, we sampled a sigma from a uniform distribution (i.e. $\sigma \in [0, 1.5]$ and $\sigma \in [0, 3]$).

For measuring equivariance/invariance, we used $\sigma \in [0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5]$.

A.5 Gaussian Noise

Gaussian Noise is similar to Color Jitter except that we sample different random values for each spatial location. The strength of the noise is controlled by σ, which is measured in pixel intensity ($[0, 255]$ scale). When training the four CNNs, σ is sampled from a uniform distribution for each input image. The four ranges used were $[0, 5]$, $[0, 10]$, $[0, 15]$, $[0, 20]$.



(a) Original Image (b) Transform 4 (c) Transform 6 (d) Transform 9

Figure 6: Example Perspective transforms for which we measured invariance and equivariance in Figure 7f. (a) is the original untransformed images, while (b-d) are ordered in increasing magnitude of transformation.

We measured the invariance/equivariance of Gaussian Noise transforms defined by $\sigma \in [2.5, 5, 7.5, 10, 12.5, 15, 20, 25, 30, 40]$.

A.6 MIRROR

Mirroring refers to reflecting the input image over the horizontal or vertical image axes. We trained 3 CNNs with mirroring. The first performed horizontal mirroring with probability 0.5. The second performed vertical mirroring with probability 0.5. The third performed both types of mirroring with independent probability 0.5 for a total of 4 combinations of flips.

We measures the equivariance/invariance of all 4 combinations of flips.

A.7 PERSPECTIVE

Perspective transforms are a class of transforms with the constraint that straight lines in the input image remain straight in the output image. One way to parameterize (8 parameters) a perspective transform is to specify the output coordinates of the input unit square, so that the unit square is mapped to an arbitrary quadrilateral.

When training a CNN, for each image we first sampled σ from a uniform distribution. Then we sampled the displacement of the coordinates of the unit square from a Gaussian Distribution with mean $\mu = 0$ and standard deviation σ. Thus the output coordinates for the upper left corner of the unit square are $x = N(0, \sigma), y = N(0, \sigma)$. For the lower right corner, they would be $x = 1 + N(0, \sigma), y = 1 + N(0, \sigma)$. Then the image is warped according to the perspective transform defined by the output coordinates of the unit square.

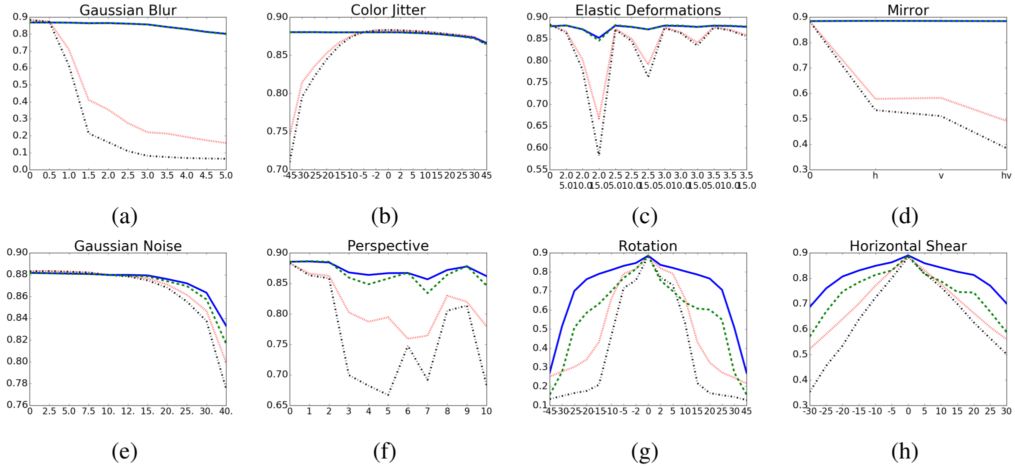
The ranges used for training are $sigma = [0, 0.001], [0, 0.002], [0, 0.003], [0, .004]$. For measuring invariance/equivariance, we sampled 10 perspective transforms from a range of equally spaced σ values in increasing magnitude along the x-axis of Figure 7f. Examples of these transforms are given in Figure 6.

A.8 ROTATION

A Rotation transform is specified by an angle θ and is always performed about the center of the image. During training, we sample θ from a uniform distribution for each image. We trained four CNNs with rotations with θ drawn from the following ranges: $[-5, 5], [-10, 10], [-15, 15], [-20, 20]$. For measuring equivariance/invariance, we used $\theta \in \pm\{2.5, 5, 10, 15, 20, 25, 30, 45\}$.

A.9 SHEAR

A Shear transform warps a square image into a parallelogram. We parameterize it by specifying a shear angle θ and an orientation that is either horizontal or vertical. While the original square image has corners that are $90\,\mathrm{deg}$, the parallelogram that results has corners with angles of $90 + \theta$ and $90 - \theta$.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7: Invariance and equivariance accuracy measurements for RVL-CDIP (similar to Figure 1). Each plot compares a CNN trained with a particular transformation (blue line for equivariance, green for invariance) to the baseline model (red for equivariance, black for invariance) trained with no transformations.

During training, we sample θ from a uniform distribution for each image. We trained four CNNs with shear with θ drawn from the following ranges: $[-5, 5]$, $[-10, 10]$, $[-15, 15]$, $[-20, 20]$. Each shear has equal probability of horizontal or vertical orientation. For equivariance/invariance, we measured only horizontal shears with $\theta \in \pm\{5, 10, 15, 20, 25, 30\}$.

B Additional Results

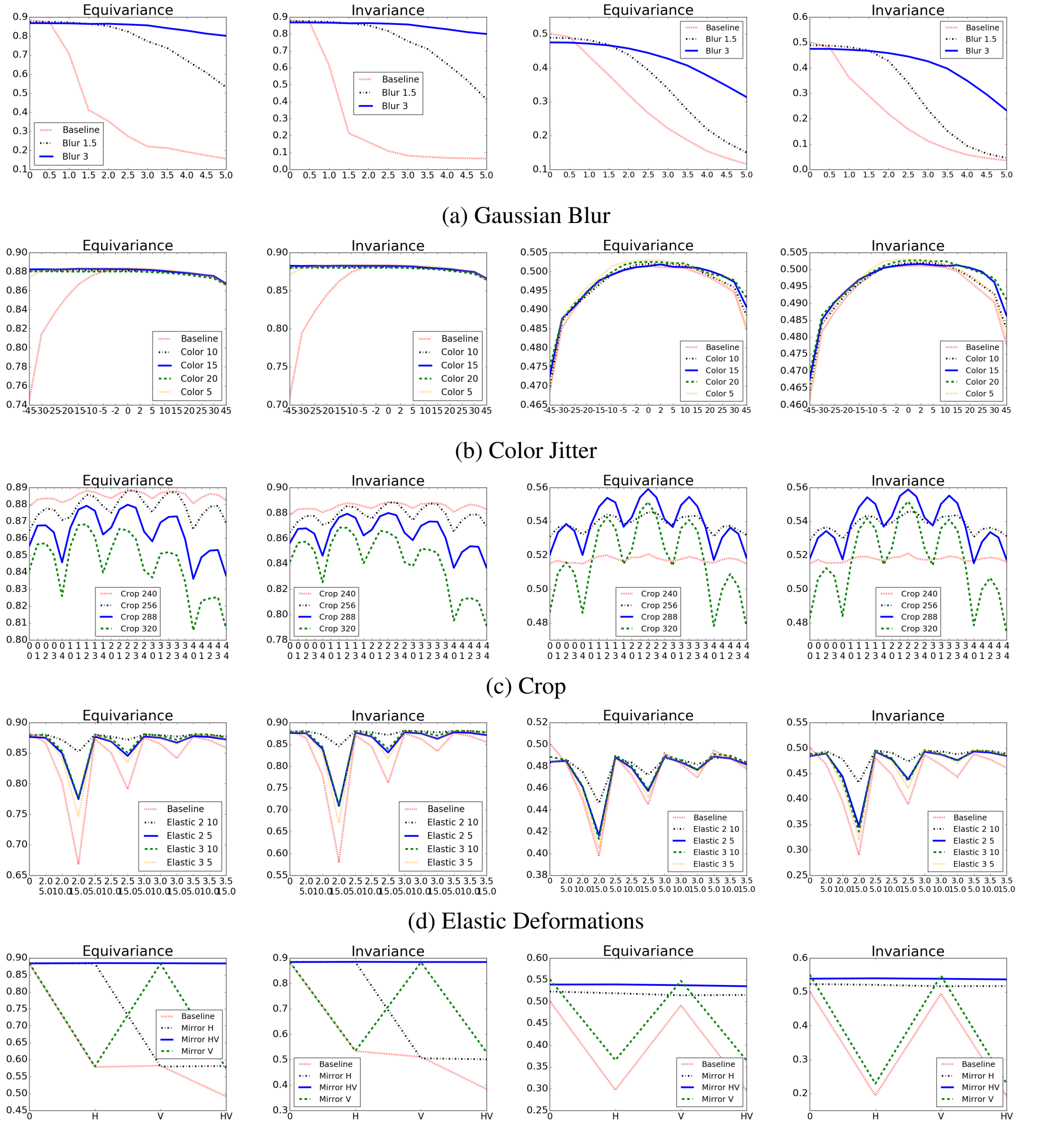
B.1 Invariance and Equivariance

In this section, we include additional results from the experiments in Section 6.1 that did not fit in the main text. Figure 1 presents invariance and equivariance measurements for the ILSVRC dataset (similar to Figure 7).

We observe several interesting differences to the results on RVL-CDIP. The baseline model for ILSVRC is significantly more invariant to vertical mirrors than horizontal mirrors, though the RVL-CDIP baseline performs equally bad on both types of mirrors. Brightness increases affect ILSVRC trained CNNs more, though this is likely because natural images are more susceptible to image saturation because they occupy the full spectrum of pixel intensities, while document images tend to be bi-modal. The RVL-CDIP CNN trained on Gaussian Noise was able to learn invariance, while the corresponding ILSVRC CNN did not. However, both the baseline and noise CNN for ILSVRC can correct for the noise with an equivariant mapping.

Figures 8 and 9 show the invariance/equivariance for all 70 CNNs trained with data augmentation. In general, we see an overwhelming trend that training with greater variety of inputs transforms results in greater invariance and equivariance. The exception seems to be that the magnitude of color jittering does not highly affect the CNN robustness. Early CNN layers can easily become invariant to overall brightness changes by extracting information based on pixel differences, and prior work has shown that later layers encode information mostly by which neurons are non-zero, rather than the magnitude of the neuron activations (Agrawal et al., 2014).

For Crop transforms (Figure 8c), we examined 25 evenly spaced crops arranged in a 5x5 grid. The x-axis of the figure shows these crops ordered in scan-line order. The periodic nature of the graph shows that crops nearer the center of the image yield higher performance. There is virtually no difference between the invariance and equivariance measurements for the Crop CNNs, showing that the high level CNN features trained for classification are not predictive for image extrapolation. This is because the equivariance mapping for Crop transforms uses the CNN activations for some un-centered crop to predict the CNN activations over the center crop.



(a) Gaussian Blur

(b) Color Jitter

(c) Crop

(d) Elastic Deformations

(e) Mirror

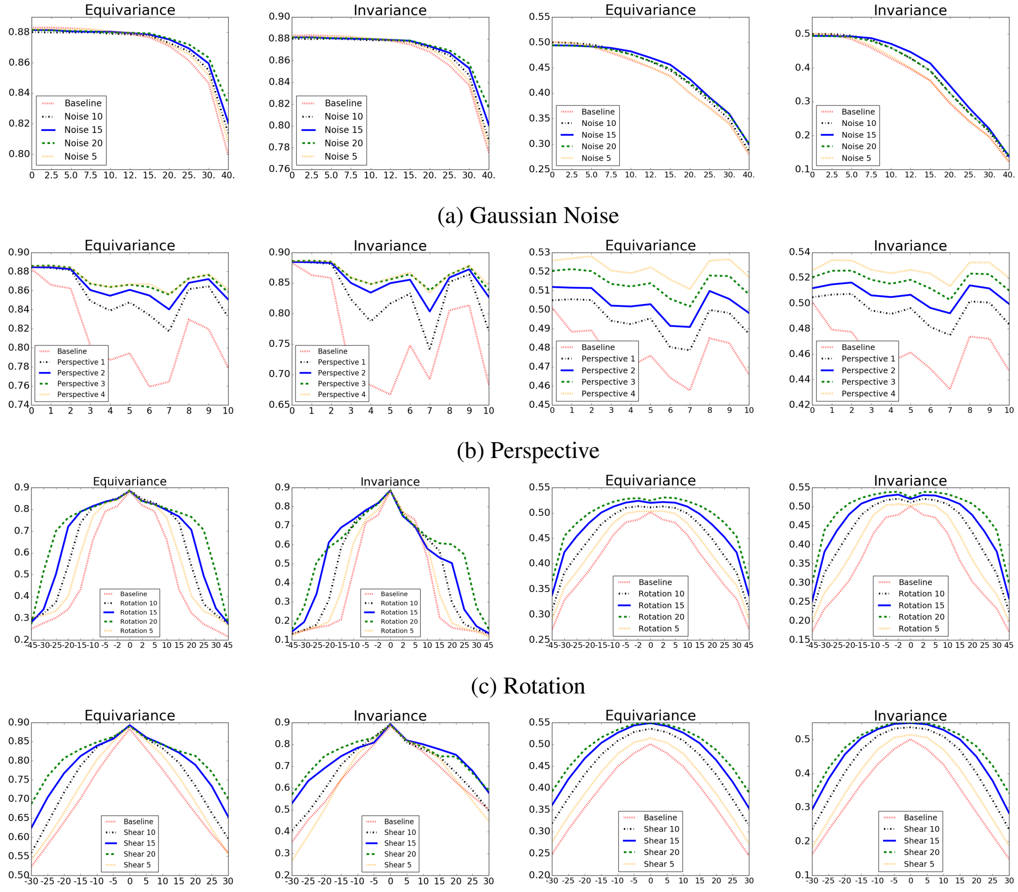
Figure 8: Equivariance and Invariance measurements of CNNs for some transforms. The first two columns are for RVL-CDIP. The last two columns are for ILSVRC.

B.1.1 CNN ACCURACY

We also report the accuracy of each CNN in Table 3. While all transforms help improve invariance, we see that some input transformations help performance on the untransformed images, while other hurt performance. In particular, Crop and Shear transforms improve performance the most, while Elastic Deformations and Gaussian Blur seem to hurt performance most.

B.2 CROSS DATASET INVARIANCE/EQUIVARIANCE MEASUREMENTS

In this section, we describe and give results for a new experiment. While all previous experiments measure invariance/equivariance on the same dataset that the CNN was trained on (though with different splits), here we use a new dataset to measure the invariance/equivariance of CNNs trained on RVL-CDIP. The new dataset, ANDOC, is also composed of grayscale document images, though while RVL-CDIP is composed of scanned office documents, ANDOC is composed of digitized



(a) Gaussian Noise

(b) Perspective

(c) Rotation

(d) Shear

Figure 9: Equivariance and Invariance measurements of of CNNs for some transforms. The first two columns are for RVL-CDIP. The last two columns are for ILSVRC.

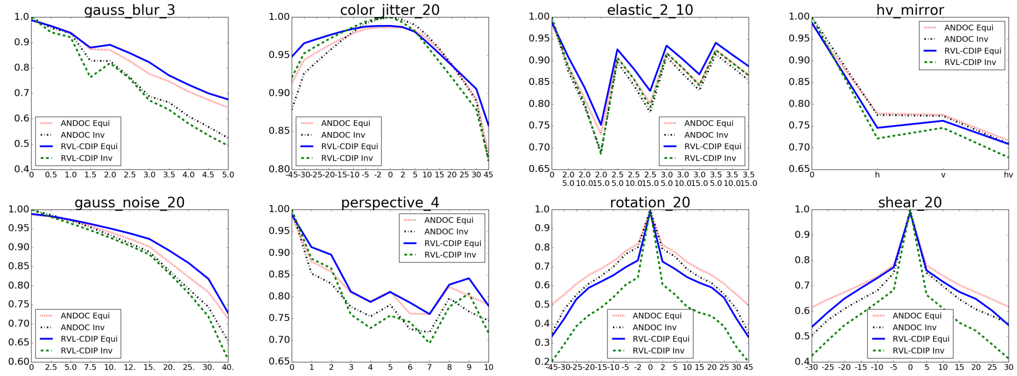
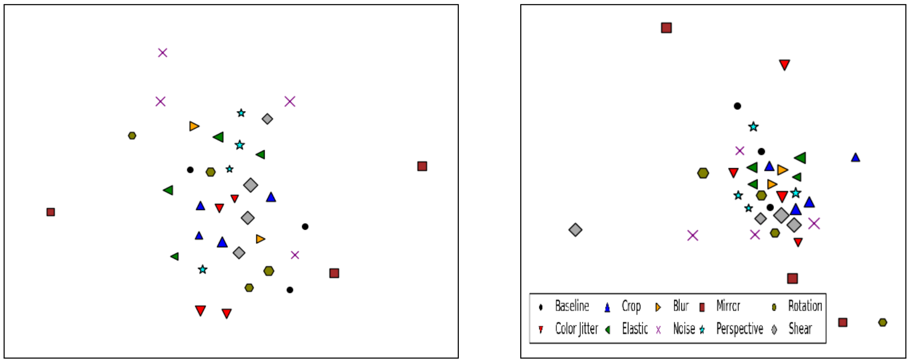


Figure 10: Cross dataset measurements of invariance/equivariance reveal that these properties do extend to data domains beside the one used to train the CNN, though the invariance is slightly weaker. These CNNs were trained on RVL-CDIP and measured using data from a dataset of historical documents. Graphs show $1 - L_{L2}$, so higher is better.

historical documents, which differ significantly from modern office documents. As the RVL-CDIP CNNs were not trained to classify ANDOC documents, we resort to measuring the L_{L2} difference in the feature representations.

CNN	RVL-CDIP	ILSVRC	CNN	RVL-CDIP	ILSVRC
Color Jitter $\sigma = 5$	**88.37**	**50.31**	Crop 240x240	88.82	52.10
Color Jitter $\sigma = 10$	88.10	50.20	Crop 256x256	**88.83**	54.58
Color Jitter $\sigma = 15$	88.26	50.15	Crop 288x288	88.00	**55.89**
Color Jitter $\sigma = 20$	88.01	50.26	Crop 320x320	86.47	55.17
Elastic $\sigma = 2, \alpha = 5$	87.64	48.42	Noise $\sigma = 5$	**88.19**	**50.02**
Elastic $\sigma = 2, \alpha = 10$	87.89	48.37	Noise $\sigma = 10$	88.01	50.00
Elastic $\sigma = 3, \alpha = 5$	87.91	48.42	Noise $\sigma = 15$	88.14	49.42
Elastic $\sigma = 3, \alpha = 10$	**87.94**	**48.85**	Noise $\sigma = 20$	88.15	49.46
Blur $\sigma = 1.5$	**87.50**	**48.95**	Mirror Horz.	**88.51**	52.33
Blur $\sigma = 3$	86.82	47.49	Mirror Vert.	**88.51**	**55.13**
			Mirror Horz./Vert.	88.42	53.91
Perspective $\sigma = 0.001$	88.51	50.48	Rotation $\theta = \pm 5$	88.57	50.30
Perspective $\sigma = 0.002$	88.45	51.12	Rotation $\theta = \pm 10$	**88.74**	51.12
Perspective $\sigma = 0.003$	**88.60**	52.05	Rotation $\theta = \pm 15$	88.48	51.99
Perspective $\sigma = 0.004$	88.55	**52.59**	Rotation $\theta = \pm 20$	88.49	**52.44**
Shear $\theta = \pm 5$	89.02	51.43	Baseline 1	88.08	50.07
Shear $\theta = \pm 10$	89.23	53.62	Baseline 2	88.31	50.14
Shear $\theta = \pm 15$	**89.33**	54.89	Baseline 3	88.33	50.11
Shear $\theta = \pm 20$	88.96	**54.97**			

Table 3: Accuracy of all CNNs over untransformed images (center crops for Crop transforms). Bolded entries indicate best parameter setting per transform type. For RVL-CDIP, test set accuracy is reported, while validation accuracy is reported for ILSVRC. In general, Shear transforms yield the best performance. For RVL-CDIP, it appears that accuracy on the original image is most impacted by the type of transform, rather than the particular parameter settings. For ILSVRC, the larger parameter ranges for Crop, Perspective, Rotation, and Shear worked best.



(a) RVL-CDIP

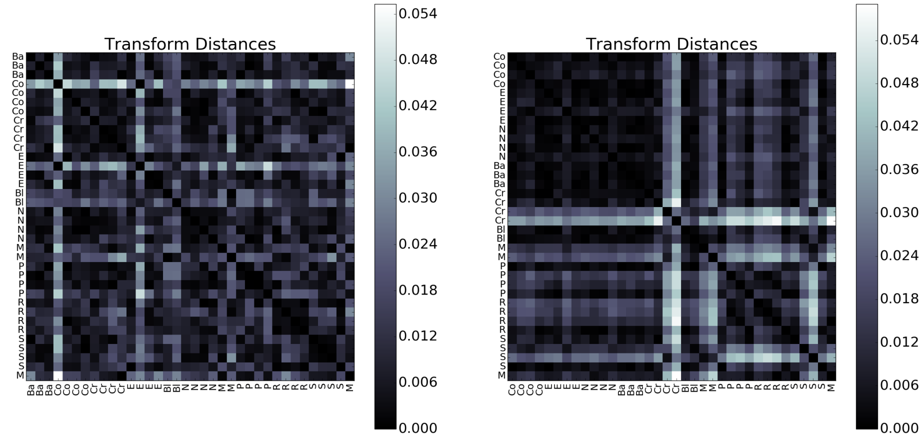
(b) ILSVRC

Figure 11: T-SNE visualization of the pairwise distance matrix for (a) RVL-CDIP and (b) ILSVRC. Best viewed electronically with zoom. Symbols and colors indicate the training transform of the network. The size of the markers indicate the relative magnitude of the training transformation.

In general, the measured invariance/equivariance is lower for ANDOC than for RVL-CDIP (see Figure 10 for examples). This makes sense because the CNNs learn to encode discriminative information about RVL-CDIP documents. However, the invariance and equivariance properties are not lost when switching to a new domain of data. In fact, for Mirror, Rotation, and Shear transforms, the invariance is higher, though this is likely due to less information about the input image being encoded for the new domain.

B.3 Representation Distance

In this section, we include additional results from the experiments in Section 6.2 that did not fit in the main text. Figure 11 shows an example T-SNE embedding of the CNN representations. There appears to be very little cluster structure and the T-SNE embeddings appear to be sensitive to the random initialization of the embedding vectors. The results from multiple runs of the T-SNE



(a) RVL-CDIP

(b) ILSVRC

Figure 12: Absolute Differences in the *one-way* distances (Eq. 9) for all pairs of CNNs using L_{L2} distance. Row/Col labels are abbreviations for the transforms used to train the CNN. For context, a typical L_{L2} representation distance (Eq. 4) is roughly 0.50 for RVL-CDIP and 1.00 for ILSVRC. This means that one-way distance differences of approximately 10% and 5% are observed for RVL-CDIP and ILSVRC respectively.

embedding in general do not agree on either which points are outliers or on nearest neighbor pairs. This is likely because representation distances do not seem to follow the triangle inequality axiom.

While Eq. 4 yields a symmetric function by averaging two *one-way* distances, we find that for some pairs of CNNs, there is significant difference between the magnitudes of those one-way distances. In other words,

$$D'(\phi_1, \phi_2) = \left| \min_{E_1}[L(E_1\phi_1(x),\, \phi_2)] - \min_{E_2}[L(\phi_1,\, E_2\phi_2(x))] \right| \tag{9}$$

is large. Figure 12 plots a heat map of Eq. 9 applied pairwise to each CNN for each dataset. Optimization difficulties are one possible cause of large values in Figure 12. However, a more likely explanation is that some CNNs encode unique information about the input that cannot be predicted from the representations learned from other CNNs.

B.4 Improved Equivariance

Table 4 shows the results of finetuning using Eq. 7 for CNNs trained with different types of data augmentation. Overall, finetuning improves 3 of 9 transform types. We believe that performance does not improve for some transforms because the dataset naturally contains examples of those transforms due to different object poses. This is not the case for RVL-CDIP because the document images are always scanned in the same manner.

Transform	ILSVRC	ILSVRC	Transform	ILSVRC	ILSVRC
	Original	Finetune		Original	Finetune
Color Jitter	50.26	50.67	Crop center from 256x256	54.58	54.18
brightness +10	50.25	50.62	upper left corner	52.90	52.68
brightness -15	49.95	50.63	bottom right corner	53.13	52.66
Elastic Deformations	48.37	50.46	Gaussian Noise	49.46	49.88
$\sigma = 2.5, \alpha = 5$	49.55	49.82	$\sigma = 8$	49.26	49.52
$\sigma = 3, \alpha = 10$	49.44	49.64	$\sigma = 16$	48.79	48.93
Gaussian Blur	47.49	46.59	Mirror	53.91	53.63
$\sigma = 1$	47.12	46.03	Horz.	54.02	51.53
$\sigma = 2$	45.94	44.60	Horz. + Vert.	53.70	51.49
Perspective	52.59	52.44	Rotation	52.44	53.70
	53.31	51.99	$\theta = -10$	53.88	53.11
	53.01	52.27	$\theta = 15$	52.66	51.76
Shear	54.97	54.84			
Horz, $\theta = -10$	54.15	52.62			
Vert, $\theta = -15$	51.85	50.77			

Table 4: Equivariance measurements for several CNNs on ILSVRC before and after finetuning using Eq. 7. The first row of each transform type shows performance over untransformed images. Results are more mixed compared to RVL-CDIP.