Differential Data Augmentation Techniques for Medical Imaging Classification Tasks

Zeshan Hussain[1], Francisco Gimenez[2], PhD, Darvin Yi[2], Daniel Rubin[2], MD, MS
[1]Stanford University, Department of Computer Science, Stanford, CA; [2]Stanford University, Department of Radiology, Stanford, CA

Abstract

Data augmentation is an essential part of training discriminative Convolutional Neural Networks (CNNs). A variety of augmentation strategies, including horizontal flips, random crops, and principal component analysis (PCA), have been proposed and shown to capture important characteristics of natural images. However, while data augmentation has been commonly used for deep learning in medical imaging, little work has been done to determine which augmentation strategies best capture medical image statistics, leading to more discriminative models. This work compares augmentation strategies and shows that the extent to which an augmented training set retains properties of the original medical images determines model performance. Specifically, augmentation strategies such as flips and gaussian filters lead to validation accuracies of 84% *and* 88%*, respectively. On the other hand, a less effective strategy such as adding noise leads to a significantly worse validation accuracy of* 66%*. Finally, we show that the augmentation affects mass generation.*

Introduction

Tremendous progress has been made in using deep learning models for image classification and segmentation. In particular, these methods have been adapted in medical diagnostic tasks, such as the prediction and segmentation of cancerous masses, across different modalities, including lung, liver, and breast scans[1][2][3].

One important preprocessing method that has been shown to be effective in training highly discriminative deep learning models is data augmentation. Data augmentation was initially popularized by Tanner and Wong in order to make simulation more feasible and simple[4]. In computer vision, because there are generally millions or even billions of parameters in CNNs, data augmentation is critical to accumulate enough data to attain satisfactory performance. Multiple data augmentation strategies have been proposed to improve vision tasks for natural images. Conventional strategies including horizontally flipping images, random crops, and color jittering[5]. Krizhevsky et al. employ a technique called *fancy PCA*, which alters the intensities of the RGB channels in training images[6].

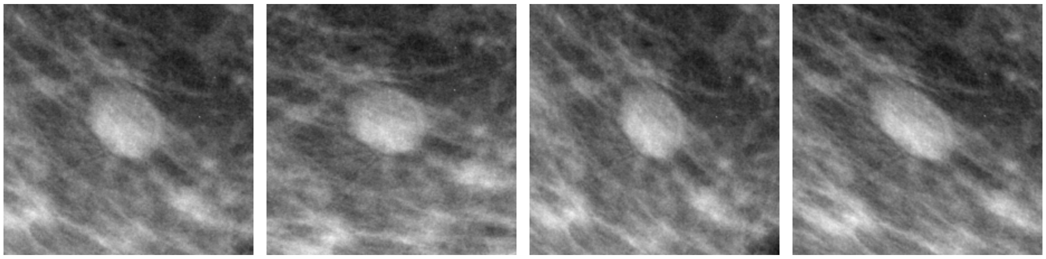
While different augmentation strategies and their combinations have been researched heavily for natural image tasks, there has been little work on finding optimal augmentation strategies for medical imaging tasks. Unlike in the natural image domain, where ImageNet and similar datasets provide millions of images, there are far fewer training images available in medical imaging[7]. This dearth of training data makes it critical to explore methods such as data augmentation, which serves as a regularizer and addresses the data-scarcity problem. In this work, we study different strategies for binary image classification of mass and non-mass mammogram images. We show that some augmentation methods capture medical image statistics more effectively than others, leading to higher training and validation accuracy. Finally, we demonstrate that smarter augmentation may result in fewer artifacts in CNN visualizations.

Methods

We attempt to gain insight into the effect that various augmentation methods have on classification accuracy and visualizations of trained CNNs. Our workflow consists of four main steps, the details of which are given in the following sections. First, for each image I in the entire dataset, we perform initial preprocessing, which includes cropping the full size image and splitting into training and validation sets. Second, for each image I_{tr} in the training set, we perform one of eight augmentations. Third, after performing additional cropping of the augmented images, we train eight VGG-16 nets independently on the eight uniquely augmented sets. Finally, we evaluate performance of the trained CNN by measuring training and validation accuracy of mass/non-mass mammogram classification as well as qualitatively assessing visualizations generated from the CNN.

Dataset & Preprocessing

A set of 1650 mass cases and 1651 non-mass (normal) cases were obtained from the Digital Database for Screening Mammography (DDSM)[8]. The full sized images were initially cropped to 1000 by 1000 images to improve augmentation speed. Each mass image was cropped around the mass lesion, while a random 1000 by 1000 crop of the breast tissue was taken in normal images. Finally, we split the full dataset into a training set and validation set, where approximately 80% of the images are in the training set and 20% are in the validation set.



(a) original **(b)** rotation **(c)** scaled **(d)** sheared

Figure 1: Example Mass Case + Augmentations

Augmentations

Before training the models, we utilize eight augmentation strategies to generate eight new training sets. Each new training set is simply the original training images in addition to the training images augmented by one of the techniques below. Each augmentation is as follows:

- *Flips*: We perform a horizontal and vertical flip for each image I in the training set. Even though only horizontal flips are used in natural images, we believe that vertical flips capture a unique property of medical images, namely, invariance to vertical reflection. Conventionally, for natural images, only horizontal flips of the original images are used, since vertical flips often do not reflect natural images (i.e. an upside-down cat would not generally make a model more discriminative during training). However, a vertical flip of a mass would still result in a realistic mass.
- *Gaussian Noise*: We generate an array, N, where each element in the array is a sample from a gaussian distribution with $\mu = 0$ and with σ^2 in the range of $[0.1, 0.9]$. Then, for each image I, we obtain a noisy image, $I' = I + N$.
- *Jittering*: For each I, we add a small amount of contrast (+/- 1-4 intensity values).
- *Scaling*: We scale each I in either the x or y direction; specifically, we apply an affine transformation, $A = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}$ on I.
- *Powers*: We take each I to a power. To calculate the power, p, we use the following equation, $p = n \cdot r + 1$, where n is a random float taken from a Gaussian distribution with mean 0 and variance 1 while r is a number less than 1. Then, to generated the augmented image, I_a, we have, $I_a = sign(I) * (|I|^p)$. The sign and power are both taken elementwise.
- *Gaussian Blur*: We blur each image I by a gaussian function defined by a variance between 0.1 and 0.9. The filter size is then generated internally by scikit-image[9], where the radius of the kernel is, $r = 4 \cdot \sigma$.
- *Rotations*: The following affine transformation, $A = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$, where θ is between 10 and 175 degrees, is applied.
- *Shears*: Finally, each image I is sheared, represented by the following affine transformation, $A = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}$. s defines the amount that I is sheared, and it is in the range of $[0.1, 0.35]$.

For each augmentation, we vary the augmentation hyperparameters across their specified ranges and generate a final augmented training set of 15673 images. These ranges were chosen such that a single transformation in the range preserves the class while a transformation outside of the range is not guaranteed to be class-preserving. Note that no augmentations were done on the validation set.

Training & Evaluation

Before we begin the training, we further crop each image in both the training and validation set to 500 by 500 pixels. We do so to retain as much information as possible before downsampling to 224 by 224 pixels, which is the input image size expected by the VGG-16 CNN[10]. Then, we train eight VGG-16 CNNs on the eight augmented training sets. The specific experimental parameters are described in the next section.

Before evaluating model performance, we first quantify the augmentation by comparing the similarity between the mean image of the augmented training set, M', and the mean image of the pre-augmented training set, M. The mean image is an intuitive representation of the training set because it captures the basic regularities across all the training images (e.g. general mass shape, location, etc.). Specifically, we compute the mutual information between M and M'. Mutual information has emerged as an effective similarity metric between two images; it captures the reduction in uncertainty of one variable given that we know the other[11]. In the context of this paper, we can think of the mutual information between M and M' as being how effectively M' retains the image statistics reflected by M (i.e. how much information is ”obtained” about M through M'). We evaluate each model by observing the variation in training and validation accuracy across augmentations.

Additionally, we qualitatively assess visualizations generated by the CNNs. The visualizations for each CNN are generated according to the class visualization method detailed by Simonyan et al.[12]. Specifically, we numerically generate an image that represents the class of interest, which in this case is ”mass”, from an input image of normal breast tissue. We do this generation by performing gradient ascent on the target class.

Formally, let I be the the input image and let y be the target class. $s_y(I)$ is the score that a CNN assigns to image I for class y. The image, I^*, that maximizes the score for class y is generated by solving the following optimization problem,

$$I^* = \arg\max_I s_y(I) + \lambda \|I\|_2^2 \,. \tag{1}$$

Note that λ is a regularization parameter. We solve this optimization using backpropagation, where we initialize the scores to be a one-hot vector, with the target class set to 1 and the other class set to 0. Letting dI be the gradients derived from backpropagating from the score layer, the final gradients are defined by the following equation,

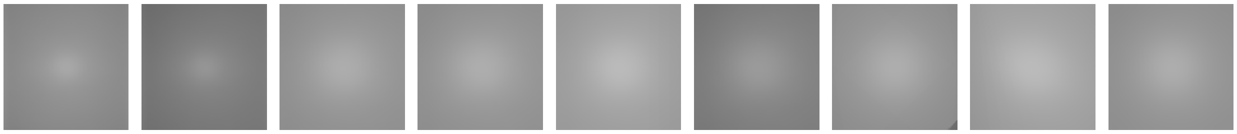
$$dI = dI - 2\lambda \cdot I, \tag{2}$$

where the second term, $2\lambda I$, comes from the derivative of the second term in the objective function[12].

Experiments

We perform eight training experiments, where one experiment consists of training a VGG-16 net on one of the eight augmented sets. For each experiment, the learning rate is set to $1\mathrm{e}{-3}$, L2 regularization is set to $1\mathrm{e}{-7}$, and the dropout parameter p is 0.5. The network is then trained over its corresponding augmented training set for 2500 iterations. We train each network for approximately 1.5 epochs, since the classification accuracy generally plateaus around this point.

For the visualization experiments, we find the gradients that need to be applied to image X at each iteration via backpropagation (derivation not shown). If we let dX be the gradients derived from the backpropagation at iteration i, then the update step at i will be $X = X + \alpha \cdot dX$, where α is the learning rate. For each of the eight visualizations, the learning rate is set to $1\mathrm{e}{-1}$, the regularization parameter applied to dX is $1\mathrm{e}{-7}$, and the number of iterations where we update X is 350.



(a) original **(b)** noise **(c)** gfilter **(d)** jitter **(e)** scale **(f)** powers **(g)** rotate **(h)** shear **(i)** flips

Figure 2: Mean Images; The mean image of each augmented set (b-i) is shown compared to the mean image of the original training set (a).

Results

To visualize the effect of each augmentation on the training set, in Figure 2, we present the mean image of the training set before augmentation as well as the mean images of each augmented training set. The mutual information between the mean images of each augmented training set and the mean image of the original data set are given in Table 1. These values enable us to explore the relationship between the mutual information of a mean image generated from an augmented training set and the training and validation accuracy of the CNN trained on that augmented set. In other words, we attempt to determine if the extent to which an augmentation captures information about the original training set affects the training and validation accuracies. Because the original training set is representative of the validation set, this effect gives us some intuition on how the model might generalize given some augmentation on the training set. Figure 3 shows how classification accuracies vary with the mutual information of the augmented training set's mean image.

augmentation type	MI	training acc.	validation acc.
Noise	2.27	0.625	0.660
Gaussian Filter	2.60	0.870	0.881
Jitter	2.59	0.832	0.813
Scale	2.67	0.887	0.874
Powers	2.33	0.661	0.737
Rotate	2.20	0.884	0.880
Shear	2.06	0.891	0.879
Flips	2.70	0.830	0.842

Table 1: Mutual Information between Mean Images + Average Accuracies

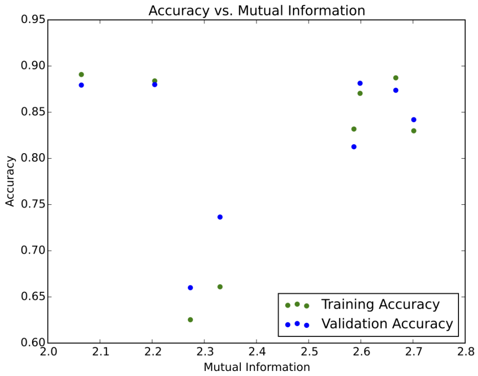
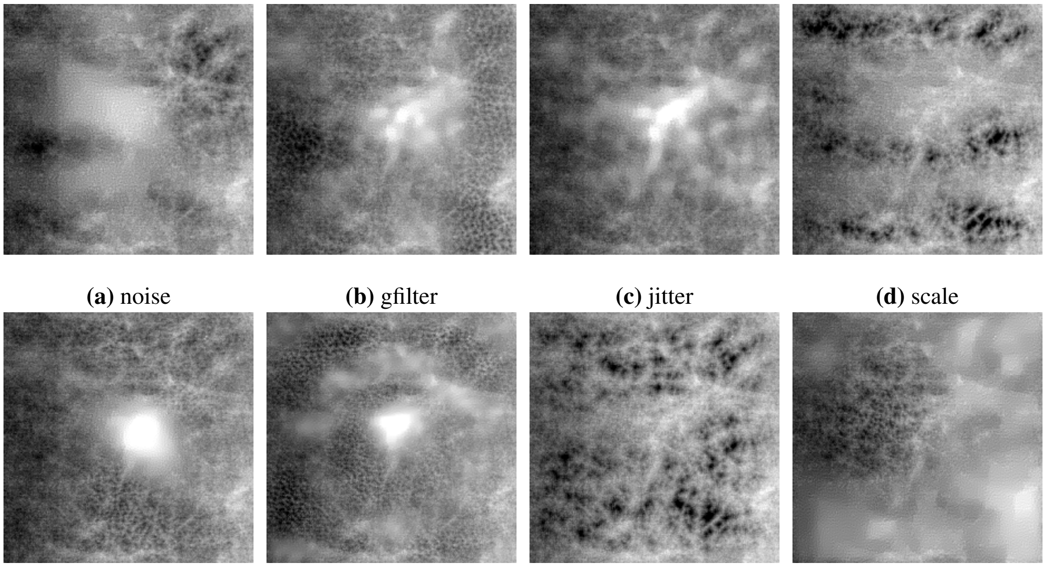


Figure 3: Accuracy vs. Mutual Information of Mean Images

Generally, we wish to see how each model performs by observing its classification accuracy over some number of epochs. The training/validation accuracy is computed simply by finding the number of samples classified correctly as mass or non-mass out of the total number of samples in the training/validation set. These accuracies are tracked over 2500 iterations (approximately 1.5 epochs), and the mean accuracies are shown in Table 1. Finally, we also assess the performance of each model by qualitatively analyzing the image generated from the CNN using the class visualization method described previously, where we iteratively alter an input image to make it look more like a ”mass”. These images are shown in Figure 4.



(a) noise **(b)** gfilter **(c)** jitter **(d)** scale

(e) powers **(f)** rotate **(g)** shear **(h)** flips

Figure 4: Visualizations; augmentation used has significant effect on the generated mass image. Artifacts that appear are correlated to the type of augmentation used.

Discussion

Looking at the results from Table 1 and Figure 3, we notice that aside from the accuracies for the shear and rotate augmentation sets (the cluster on the top left), a higher mutual information correlates to a higher classification accuracy. Specifically, the augmentations that resulted in mean images that had mutual informations of approximately 2.6-2.7 (flips, scale, jitter, gaussian filter) had accuracies of around 0.85 while those that resulted in mean images with mutual informations of around 2.3 (noise, powers) had much lower accuracies of around 0.65-0.70. Also, in general, the noise and powers augmentations, resulting in mutual information 2.27 and 2.33, respectively, have more variance in their validation accuracies over time compared to augmentations resulting in higher mutual information (plots not shown).

With regard to the shear and rotation augmentations, which result in mean images with relatively low mutual information but high classification accuracies, there is an intuitive explanation as to why this is the case. Namely, rotations and shears do retain many of the image statistics in the original training set, leading to a high classification accuracy, but artifacts in the augmented images may affect the mean image and lead to a lower mutual information. For example, if a mass appears on the edge of a breast, then the 500 by 500 crop of the original image will retain the 0 pixels that appear when rotating the 1000 by 1000 image. Generally, if the mass is near the center of the crop, as is the case for the majority of the training images, then the 0 pixels will be cropped away. These slight artifacts may decrease the overall mutual information, even though the high classification accuracies suggest that rotations and shears preserve mammography image statistics. To solidify the correlation between mutual information and classification accuracy, future work includes investigating other augmentation strategies and deriving a better representation to capture image statistics than the mean image.

Furthermore, we see that the augmentation strategy used has a significant effect on the type of mass that is generated, as shown by the images in Figure 4. In general, we see that the augmentation determines the type of artifact that predominates the mass generation. For example, the artifacts that appear in the generated image from the CNN trained on the rotations set has circular patterns. We see a similar phenomenon for the ’flips’ generated image, where we see several masses that look they have been flipped. These results suggest that using a combination of augmentations that have high mutual information might lead to an ensemble effect where the medical image statistics are holistically captured, leading to generated images with fewer artifacts. We hope to use these findings when applying generative deep learning techniques in our future work.

Conclusion

We show that the mutual information captures the basic image statistics of an augmented dataset and roughly correlates with the performance of a CNN trained on that augmented set. Overall, our work shows that augmentation strategy greatly affects discriminative performance but also drastically affects generative performance, suggesting a strong link between discriminative and generative learning.

References

1. Junghwan Cho, Kyewook Lee, Ellie Shin, Garry Choy, and Synho Do. Medical image deep learning with hospital pacs dataset. *arXiv preprint arXiv:1511.06348*, 2015.

2. Angel Alfonso Cruz-Roa, John Edison Arevalo Ovalle, Anant Madabhushi, and Fabio Augusto González Osorio. A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 403–410. Springer, 2013.

3. Rasool Fakoor, Faisal Ladhak, Azade Nazi, and Manfred Huber. Using deep learning to enhance cancer diagnosis and classification. In *Proceedings of the International Conference on Machine Learning*, 2013.

4. Martin A Tanner and Wing Hung Wong. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, 82(398):528–540, 1987.

5. Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014.

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

7. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

8. Rebecca Sawyer-Lee, Francisco Gimenez, Assaf Hoogi, and Daniel Rubin. Curated breast imaging subset of ddsm, 2016.

9. Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.

10. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

11. Daniel B Russakoff, Carlo Tomasi, Torsten Rohlfing, and Calvin R Maurer Jr. Image similarity using mutual information of regions. In *European Conference on Computer Vision*, pages 596–607. Springer, 2004.

12. Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.