An Investigation Into Improving Bone Fracture Medical Imaging Classifiers Through An Application of Generative Adversarial Networks

Jacob Platin
jplatin@seas.upenn.edu

Advisor
Paul A. Yushkevich, Ph.D
pauly2@mail.med.upenn.edu

Abstract

Recent developments with Generative Adversarial Networks (GANs) have enabled the creation of novel, photorealistic images from an existing dataset. This is of particular use in the medical imaging community, where real training data is scarce and synthetic data could be of great use. Specifically, classifiers, which rely on large datasets to function well, are now being trained to detect a variety of ailments from x-ray images. Improving the accuracy of these classifiers could have major positive impacts on the workflows of radiologists and patient outcomes. This paper shows that state-of-the-art GANs can successfully produce realistic novel x-ray images. However, augmenting an existing dataset was not found to increase model performance on an existing presence-of-fracture classifier. Future work should attempt to remedy the methodology and other concerns present, particularly given the promising GAN output.

Introduction

As computing power has been democratized and academic research has progressed, there have been massive technological advances in the field of machine learning in the past 20 years (Kazeminia et. al, 2019, p.2-3). In particular, computer vision has seen a boon, with deep-learning models such as AlexNet, VGG-16, and Faster R-CNN demonstrating impressive performance in scene understanding, image segmentation, and object detection/classification (Chahal et al., 2021, p. 1-6). One area where computer vision has immense promise is in the medical imaging community (Kazeminia

et al., 2019, p. 3). Being able to read and understand an x-ray (or other medical image) is a crucial portion of evaluating the condition of patients, since a study regarded as normal eliminates the need for further medical intervention. Moreover, according to recent estimates, musculoskeletal conditions worldwide affect more than 1.6 billion people (United States Bone and Joint Initiative, 2020), and musculoskeletal conditions are a primary cause of both long-term pain and disability (Woolf & Pfleger, 2003). In fact, over 30 million trips to the emergency room are attributable to musculoskeletal conditions (Woolf & Pfleger, 2003).

The existing labeling process (i.e. manual study by a radiologist) is not only time-consuming and costly, but also potentially susceptible to error (Kazeminia et al., 2019, p. 3-4). Computer vision-based anomaly classifiers may be able to assuage these problems by acting as both a safeguard and another set of eyes in reading these medical images. If the radiologist and model agree, then it is likely the diagnosis is correct, and if not, the radiologist can further examine the image in order to ensure their conclusion. There is also the remote possibility that these models could outperform radiologists, but this seems highly unlikely due to their still blackbox nature. Given that these models can produce a result in a matter of seconds, they may greatly speed up the general radiology workflow, removing the need for intensive manual intervention.

A major hurdle in improving these classifiers is the lack of widely-available training data (Skandarani et al., 2021, p. 2; Kazeminia et al., 2019, p. 3-4). Between the cost of x-rays, concerns over patient privacy, and a lack of interest from hospitals and associated medical groups, finding high-quality training data is quite difficult (Skandarani et al., 2021, p. 3). Recent advances in Generative Adversarial Networks (GANs), however, may offer a solution to this problem. GANs use an existing dataset to generate novel, photo-realistic images (Goodfellow, 2014, p. 6). These novel images can, in turn, be added to the existing dataset in order to provide more data to train a given model. The aim is that this hybrid, generated dataset will further improve the accuracy of the classifier.

This paper attempts to first generate novel images from the RSNA Bone Age dataset, which contains only normal (non-diseased) hand x-rays. These x-rays are then used to augment the MURA dataset (for only the normal class for the hand study type) in a variety of experiments. For each of these experiments, an existing MURA classifier is trained, in order to determine whether GANs can indirectly improve classifier accuracy via dataset augmentation. Due to time and resource constraints, this paper is only concerned with the hand study type, and future work can attempt to extrapolate this approach to all of the study types.

Datasets and Existing Classifiers

Mura Dataset

Rajpurkar et al. (2018) created the MURA dataset and associated challenge in December of 2017. The MURA dataset, of musculoskeletal, upper-extremity radiographs, contains 40,561 images from 14,863 studies (p. 1). Each study is composed of potentially many views (images) and each view belongs to one of the following types: finger, forearm, elbow, humerus, wrist, hand or shoulder (p. 3). Board-certified radiologists from the Stanford hospital labeled each study as normal or abnormal (p. 1, 3).

The dataset was split into a training dataset -- 11,184 patients, 13,457 studies, and 36,808 views -- a validation dataset -- 783 patients, 1,199 studies, and 3,197 views -- and a test dataset -- 206 patients, 207 studies, and 556 views (Rajpurkar et al., 2018, p. 3). There was no overlap in patients between any of the sets, and, for the novel test set, additional labels were collected from board-certified Stanford radiologists (Rajpurkar et al., 2018, p. 3). The images vary in size, but all are approximately 400x512. A summary of the data is given in Table 1. Examples of normal and abnormal images from the (hand) dataset can be found in Figure 1 and Figure 2, respectively.

Study	**Train**		**Validation**		**Total**
	Normal	**Abnormal**	**Normal**	**Abnormal**	
Elbow	1094	660	92	66	1912
Finger	1280	655	92	83	2110
Hand	1497	521	101	66	2185
Humerus	321	271	68	67	727
Forearm	590	287	69	64	1010
Shoulder	1364	1457	99	95	3015
Wrist	2134	1326	140	97	3697
Total Number of Studies	8280	5177	661	538	14656

Table 1: a summary of the studies within the MURA dataset (Rajpurkar et al., 2018, p. 3).

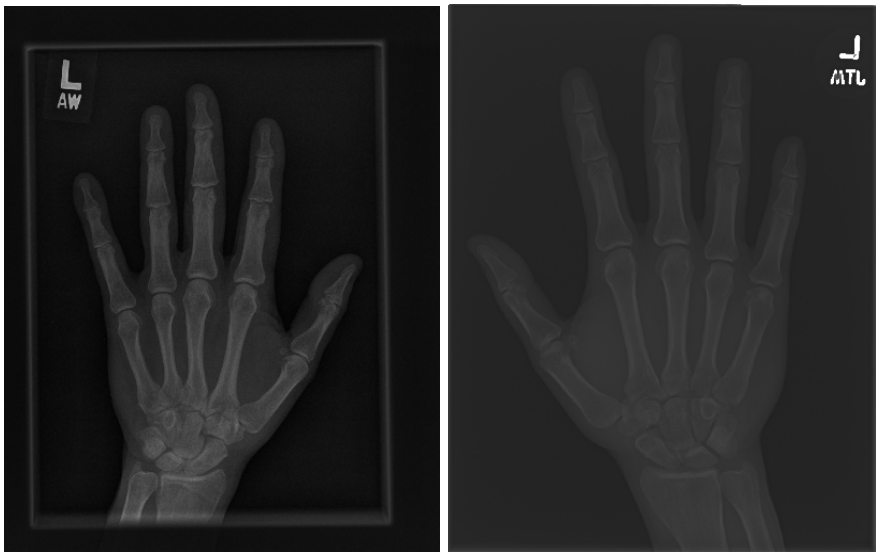

Figure 1: two example (normal) hand x-rays from the MURA dataset

Figure 2: two example (abnormal) hand x-rays from the MURA dataset

As mentioned previously, due to resource and time constraints, this paper is only concerned with the hand study type. A summary table of the hand-only dataset (in terms of *number of images*) is given below:

Train		Validation		Total
Normal	**Abnormal**	**Normal**	**Abnormal**	
4059	1484	271	189	6003

Table 2: a summary of the hand-only images from the MURA dataset (Rajpurkar et al., 2018, p. 3).

Mura Model Architecture

Rajpurkar et al. (2018) used the MURA dataset to train a deep model in an attempt to classify whether the images were abnormal. The input of the model is one or more images for a given study, and the output is the probability of abnormality (p. 2). If the study contains more than one image, then the overall probability of abnormality is computed by taking the average of the probabilities of each image (p. 2). If this average is greater than 0.5, then the model will predict an abnormal study (p. 2). The model is a 169-layer **convolutional neural network** that uses the Dense Convolutional Network architecture (p. 3-4; Huang, 2018). In this architecture, every layer is connected to every other layer in a typical fully-connected manner, but the given network was altered by replacing the final **fully-connected layer** with a layer that has just one output, so as to account for the binary nature of the problem (Rajpurkar et al., 2018, p. 4; Huang, 2018).

Mura Model Training and Evaluation

The authors utilized the pre-trained weights from a 2009 model by Deng et al. The network was then trained using the **Adam optimizer** with default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and using **minibatches** of size 8 (Rajpurkar et al., 2018, p. 4). A **learning rate** of 1e-4 was used as an initial value. Each time the validation loss plateaued, it was decayed by a factor of 10 (Rajpurkar et al., 2018, p. 4). In addition, the authors normalized each image to have a standard deviation and mean that matched the images found in the training dataset (ImageNet) (Rajpurkar et al., 2018, p. 4). The images were then scaled to 320 × 320, and the data was augmented during training by applying various inversions and rotations (Rajpurkar et al., 2018, p. 4).

In order to evaluate the model, a test dataset was created of 207 studies, with six labels being provided from board-certified radiologists. In order to create a gold-standard baseline, the researchers randomly chose 3 of the radiologists' labels and defined the ground truth as whichever label occurred most frequently (Rajpurkar et al., p. 4) The radiologists and model were then compared via the Cohen's kappa statistic (κ), which

quantifies how close the model and radiologists were with respect to the gold standard (Rajpurkar et al., p.4).

Rajpurkar et al. (2018) found that, for the finger and wrist study types, their model performed similar to the performance of the best radiologist (p. 5). However, model performance was worse (in terms of abnormality detection) on the forearm, elbow, hand, shoulder, and humerus study types (p. 5). In order to improve on their results, the researchers made the MURA dataset free to the public and created a public challenge to beat their Kappa value of 0.705.

RSNA Bone Age Dataset

The RSNA Bone Age Dataset was created as a result of the 2017 Radiological Society of North America (RSNA) annual meeting. The dataset was paired with an associated challenge in which contestants were tasked with correctly inferring the age of a child given just an x-ray (Halabi, 2018). The dataset contains 12,611 x-rays of normal (i.e. non-diseased) hands and was created by combining smaller datasets from Stanford University, the University of Colorado and the University of California - Los Angeles (Halabi, 2018). A selection of images taken from this dataset are shown below:

Figure 3: 8 example hand x-rays from the RSNA dataset

GANs

Overview

Generative Adversarial Networks (GANs) were proposed by Goodfellow et. al (2014) in their seminal 2014 work of the same name. GANs are composed of two models: a generative model that captures the data distribution (of the training data), and a discriminative model that estimates the probability that a sample came from the training set and not the generative model. During training, the goal for the generative model is to maximize the probability of fooling the discriminator. By the end of the training process, the discriminator can be discarded, and the generator can be used to create novel samples from the data distribution. For this paper's use case, this would entail generating novel x-rays, which can be modeled as a distribution (p. 2-5).

GANs engage in a minimax-style game between a generator, which tries to maximize an objective function, and a discriminator, which tries to minimize the objective function (Goodfellow, 2014, p. 1). The original GAN proposed by Goodfellow et al. (2014) can thus be formulated as the following problem (p. 3):

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Skandarani et al. (2021) concisely summarizes this formulation:

> G(z) is the generator network with parameters θ_G. It is fed with a random variable $z \sim p_z$ sampled from a given prior distribution that G tries to map to $x \sim p_{data}$. To achieve this, another network D (aka the discriminator) with parameters θ_D is trained to differentiate between real samples $x \sim p_{data}$ from the given dataset and fake samples $\widehat{x} \sim p_{\theta_G}(x|z)$ produced by the generator. In doing so, the generator is pushed to gradually produce more and more realistic samples with the goal of making the discriminator misclassify them as real (p. 4).

Improvements in the GAN Architecture

Although GANs work in theory, they have proven difficult to train. Proper convergence relies on correct hyper-parameter tuning (so as to avoid the issue of vanishing or exploding gradients) (Skandarani et al., 2021, p. 5). Of greater importance is that basic GANs tend to suffer from mode collapse, in which the GAN only produces samples of a particular class or feature of the data distribution (Skandarani et al., 2021, p. 5). In order to combat many of the issues surrounding the original GAN architecture, Mirza & Osindero (2014) developed conditional GAN (cGAN) architecture, which explicitly includes additional information (conditions) such as specified labels when generating images. Mirza and Osindero showed that these cGANs greatly improved the stability of training as well as generating the desired features in the output images (p. 1).

A slight tweak on cGANs are ACGANs (Auxiliary Classifier GANs), which modify the discriminator to predict a particular image's label (as opposed to receiving it as an input) (Odena et al., 2016, p. 3). ACGANs exhibit a much more stable training process, generate higher quality output images, and learn to create output that is strictly independent of target label (Odena et al., 2016, p. 3). Another major development was Karras et al's novel training methodology that progressively grows GANs (Karras, 2018). In this paradigm, "both the generator and discriminator grow progressively: that is,

starting from a low resolution, more, new layers that capture increasingly fine detail are added as training progresses" (Karras, 2018, p. 1). This process both speeds up and stabilizes training, in addition to producing images that, even now, are of amazing quality (Karras, 2018, p. 5-7). The author's do note that the results are still far away from being truly photo-realistic, and that certain fine features, such as the quality for certain objects to be straight or curved, can be missed (Karras, 2018, p. 9).

DCGANs

DCGAN's, as described by Radford et al. (2016), are the first GANs to utilize **convolution** and **convolutional layers**, rather than the solely **fully-connected layers** utilized previously. **Convolutional layers** have shown great promise in the supervised realm of machine learning (e.g. object recognition), and so it is quite natural to extend their usage to unsupervised use cases, such as GANs (p. 1).

The outlined generator ingests a 100x1 noise vector and, through applying **strided 2D-convolutional transpose layers** (in tandem with a **batch norm layer** and a **ReLU** activation layer), outputs a 64x64x3 vector (or 3-channel image of size 64) (Radford et al., 2016, p. 3-4). The **TanH** function is lately used in order to return the generator's output to the given input range (Radford et al., 2016, p. 3-4). The **convolutional-transpose** layer can be thought of simply as a type of reverse **convolution**, which upsamples the data and interpolates to fill in coarse portions (Radford et al., 2016, p. 2). A major contribution of this paper was the use of **batch norm** functions immediately after the **2D-convolutional transpose layers** (Radford et al., 2016, p. 3). The authors note that this helps maintain the flow of gradients during training, a major issue in GANs (Radford et al., 2016, p. 3).

The discriminator ingests a 3x64x64 RGB image and using a variety of **(strided) convolution layers**, **batch norm layers**, and **LeakyReLU** activation layers, outputs a probability (via the **Sigmoid** activation function) that the given input is real (i.e. is drawn from the true data distribution) (Radford et al., 2016, p. 3-4). An important contribution of this paper was its use of **strided convolution** as opposed to **pooling** for downsampling. This enables the model to learn its own pooling function, which can improve results (Radford et al., 2016, p. 3). As discussed above, **batch norm** and **Leaky ReLU** also help maintain proper backpropagation flow, which stabilizes training (Radford et al., 2016, p. 3).

DCGANs are likely the most widely-used GAN in present work, and show immense improvement in the quality of the images generated, as well as training stability (Skandarani et al., 2021, p. 6).

Relevant Work

GANs and Medical Imaging

A majority of the work in applying GANs to medical imaging has centered around **CT** and **MR**-based image segmentation and image translation (e.g. converting **CT scans** to **PET** scans). Dai et al. (2017) implemented a **Fully-Connected Network** (FCN) based GAN that, in tandem with a FCN-based segmentation network, was able to create both heart and lung segmentation images from chest X-rays. Islam & Zhang (2020) were able to successfully generate synthetic brain **PET** images for different stages of Alzheimer's disease using a DCGAN. Cirillo et al. (2020) utilized a combination of a Res-Net/U-Net generator and a PatchGAN discriminator in order to segment brain tumors in 3D **MRI**. Mahapatra et al. (2019) used the cGAN architecture in their proposal that generated novel chest x-rays (given both segmentation masks and images) in order to improve a segmentation model. Sundaram & Hulkund (2021) used a cGAN to generate novel chest x-rays in order to augment the public CheXpert data set. They found that existing classifiers performed better on the augmented (as opposed to non-augmented) dataset. The authors based much of their GAN architecture on previous work developed by Han et al. (2019), who attempted to use a federated learning approach

Moradi et al. (2018) attempted to improve cardiovascular abnormality detectors by generating both normal and abnormal chest x-rays from an existing dataset using a cGAN. Interestingly, the authors asked a radiologist to attempt to distinguish between the real and fake images, resulting in a surprisingly low accuracy of 66.7%. The GAN-generated data showed only modest improvement over traditional data augmentation techniques. Salehinejad et al. (2018) obtained similar results using a DCGAN and also training their classifier on a mix of the real and synthetic data.

A major recent development in the usage of GANs to create novel x-rays is in attempting to improve models for COVID-19 detection. Motamed et al. (2020) used a DCGAN generator and 4-layer CNN discriminator in order to develop a semi-supervised approach for diagnosing COVID-19 from chest x-rays. Menon et al. (2020) employed transfer learning from weights trained on a pneumonia classifier, in addition to fine-tuning from a small COVID19 dataset. Waheed et al. (2020) implement an ACGAN that augmented an existing COVID chest x-ray dataset in order to improve the accuracy of an existing classifier, when compared to the same classifier being trained solely on non-synthetic data. Karbhari et al. (2021) also utilized an ACGAN and found similar results.

Data Augmentation As An Effective Approach to Improving Classifiers

One of the primary motivations for the use of GANs in the medical imaging community (and elsewhere) is due to a lack of available data (Skandarani et al., 2021, p. 2; Kazeminia et al., 2019, p. 3-4). Labeling data is not only expensive, but also requires expert knowledge (Kazeminia et al., 2019, p. 3-4). In order to best train deep-learning models, a large amount of high-quality data is needed. The primary method in which this is done, with the aid of GANs, is through combining synthetic (GAN-generated) data with real data, an approach known as data (or dataset) augmentation.

In their comprehensive work, Bowles et al. (2018) augmented datasets by choosing how many samples of the original dataset were kept and how many synthetic samples were appended (p. 5-7). They then compared the performance of a variety of **MR** and **CT** segmentation models on the augmented datasets and the original datasets and found that synthetic data tended to yield a modest improvement in classifier performance (p. 7-8). They additionally found that the synthetic data had the greatest positive effect when the classifier used a relatively small amount of the real data -- for example, if the classifier only had access to 10% of the real data, appending synthetic data had a marked improvement (p. 7-8). This trend seemed to disappear, however, when the available data became too low, suggesting that the GAN could not be trained thoroughly (p. 11).

Sundaram & Hulkund (2021) similarly found that classifiers trained on augmented datasets performed best when the amount of available data was low (below 50%) (p. 3-4). After the 50% threshold, however, the classifiers began to perform worse with the augmented dataset (p. 3-4).

Karbhari et al. (2021) utilized two different types of datasets: one that used original images only and another that used half synthetic, half real images (p. 12). They found that their COVID-19 classifiers' accuracies improved by around 0.2 to 1.5% when using the hybrid dataset (p. 13). **AUC** measures improved only by around 0.2% to 0.7% (p. 13). It also be noted that most of the classifiers already achieved quite high accuracies and **AUC** values on the original data (all greater than 99%) (p. 13). Again, the dataset was relatively small at 1816 samples.

Salehinejad et al. (2018) only compared their classifier trained on the augmented dataset and non-augmented dataset, and found massive improvements in performance (around 22%) (p. 4). Waheed et al. obtained similar, yet more modest results, with accuracy increasing 10%, sensitivity increasing 21%, and specificity increasing 2% (p. 5-6). Again, it should be noted that the original dataset was fairly small (only 1,124 images). Madani et al. also tested their classifiers only on non-augmented and

synthetically-augmented datasets and found an even less significant accuracy increase of 2.26% (p. 7).

Methods

GAN Architecture

This paper chooses to implement a DCGAN, due to this architecture's proven record across a variety of tasks as well as its stability in training (Skandarani et al., 2021, p. 6).

Discriminator

The discriminator architecture takes as input a 1x128x128 image for a given number of batches. It then applies 5 consecutive layers of **convolution** (and occasionally **batch norm**) followed by **LeakyReLU**. The penultimate layer is a single **convolutional layer**, which is followed by a **Sigmoid** layer. The resulting output is a 1x1x1 boolean for a given number of batches that specifies whether the given input is from the real dataset or from the GAN. This is effectively the same architecture as Reford et al.'s original proposal (2016), but scaled to handle an input of size 128x128. A summary of each layer of the model, including the layer type, output shape, and number of parameters is given below in Figure 4 for a batch size of 100.

Layer Type	Output Shape	Number of Parameters
Input	[100, 1, 128, 128]	--
Conv2d	[100, 32, 64, 64]	512
LeakyReLU	[100, 32, 64, 64]	--
Conv2d	[100, 64, 32, 32]	32,768
LeakyReLU	[100, 64, 32, 32]	--
Conv2d	[100, 128, 16, 16]	131,072
BatchNorm2d	[100, 128, 16, 16]	256
LeakyReLU	[100, 128, 16, 16]	--
Conv2d	[100, 256, 8, 8]	524,288
BatchNorm2d	[100, 256, 8, 8]	512

LeakyReLU	[100, 256, 8, 8]	--
Conv2d	[100, 512, 4, 4]	2,097,1
BatchNorm2d	[100, 512, 4, 4]	1,024
LeakyReLU	[100, 512, 4, 4]	--
Conv2d	[100, 1, 1, 1]	8,192
Sigmoid	[100, 1, 1, 1]	--
Total		2,795,776

Figure 4: Proposed discriminator architecture

Generator

The generator takes as input a vector of size 100x1x1 for a given number of batches. This is followed by 5 layers of **2D transposed convolution**, **batch norm**, and **ReLU**. The final output layer is a single **2D transposed convolution** followed by a **Tanh** activation. Again, this is effectively the same architecture as Reford et al.'s (2016) original proposal, but scaled to output an image of size 128x128. A summary of each layer of the model, including the layer type, output shape, and number of parameters is given below in Figure 5 for a batch size of 100.

Layer	**Output Shape**	**Number of Parameters**
Input	[100, 100, 1,1]	--
ConvTranspose2d	[100, 1024, 4, 4]	1,638,400
BatchNorm2d	[100, 1024, 4, 4]	2,048
ReLU	[100, 1024, 4, 4]	--
ConvTranspose2d	[100, 512, 8, 8]	8,388,608
BatchNorm2d	[100, 512, 8, 8]	1,024
ReLU	[100, 512, 8, 8]	--
ConvTranspose2d	[100, 256, 16, 16]	2,097,152
BatchNorm2d	[100, 256, 16, 16]	512
ReLU	[100, 256, 16, 16]	--
ConvTranspose2d	[100, 128, 32, 32]	524,288

BatchNorm2d	[100, 128, 32, 32]	256
ReLU	[100, 128, 32, 32]	--
ConvTranspose2d	[100, 64, 64, 64]	131,072
BatchNorm2d	[100, 64, 64, 64]	128
ReLU	[100, 64, 64, 64]	--
ConvTranspose2d	[100, 1, 128, 128]	1,024
Tanh	[100, 1, 128, 128]	--
Total		12,784,512

Figure 5: Proposed generator architecture

GAN Training Procedure

PyTorch was used to implement and train the DCGAN. The RSNA images were ingested, resized to 128x128, converted to grayscale, and then finally normalized. Training was undertaken on a NVIDIA Quadro RTX 5000 GPU and took approximately 3 hours. Binary Cross Entropy Loss was used as the loss function for the discriminator due to the binary nature of the problem as well as its use in other similar papers (Sundaram & Hulkund, 2021, p. 3). Binary Cross Entropy Loss is a specific implementation of cross-entropy loss which associates a numeric error when a given label does not match the predicted label in a binary classification task (Rajpurkar et al., 2018, p. 4). The GAN was trained for 100 epochs with a batch size of 100, and the Adam Optimizer was used for both the discriminator and generator with **learning rate** 3.4e-4, β_1 = 0.5 and β_2 = 0.999.

GAN hyperparameters are difficult to choose because of the simultaneous training of the discriminator and generator. Thus, improvements in one model generally come at the expense of the other model (i.e. zero-sum). These particular values were chosen after visually observing convergence in generated images at each epoch. The remaining training procedure, including gradient calculation, is identical to Radford et al.'s original proposal.

MURA Classifier

An open-source implementation (Agrahari, 2020) of Rajpurkar et al's original network was used and can be found at https://github.com/pyaf/DenseNet-MURA-PyTorch. In order to decrease computing requirements, images were scaled to 224x224 before training (as opposed to 320x320 in Rajpurkar et al.'s original work).

Of note is that, although Rajpurkar et al.'s MURA classifier does not necessarily obtain the state-of-the-art Kappa value on the MURA challenge, due to code availability and method credibility, this paper will focus solely on improving the results from this classifier. The results obtained, however, should hopefully generalize to the other models, given that the proposed method is model-agnostic.

Results

Experimental Setup

Synthetic Dataset Generation

Using the proposed generator, a training dataset of 10,000 synthetic images was generated.

MURA Classifier Training Procedure and Data Augmentation

In order to train the MURA classifier using the newly-generated synthetic data, a modified experiment schedule was taken from Bowles et al. (2018). The classifier was trained on a random sample of 90% of the real MURA hand data, in addition to increments of 0% (as a baseline), 10%, 50%, and 100% synthetic data. *The percentage of synthetic data is measured relative to the number of normal images in the training dataset.* Because the MURA dataset has no specified test set, the so-called validation set was used as the de-facto test set and 10% of the explicit training data was held-out for validation (hence the 90% mentioned previously). Thus, in the 10% added synthetic data experiment, for example, 365 (10% of 3,653) images were added to the training dataset. The augmenting synthetic training data was randomly sampled from the 10,000-image synthetic dataset.

Train		Validation		Test		Total
Normal	**Abnormal**	**Normal**	**Abnormal**	**Normal**	**Abnormal**	
3653	1336	406	148	271	189	6003

Table 3: Breakdown of training-validation-test split of the utilized dataset (Rajpurkar et al., 2018, p. 3)

A summary of the experiments run, including the total (real and synthetic) number of images in the normal class of the training dataset is given below. This metric is relevant because the RSNA dataset did not contain abnormal studies, and therefore, only the normal class of the original MURA training dataset can be augmented.

Percent-Added Synthetic Data	Number of Synthetic Images Added	Total Number of Images in the Normal Class of the Augmented Training Dataset
0	0	3653
10	365	4018
50	1827	5480
100	3653	7306

Table 4: Summary of the experiments conducted

GAN Output

Figure 6 displays a typical output of the GAN.

Figure 6: Example outputs from the proposed DCGAN

Fréchet Distance of Inception (FID)

The Fréchet Distance of Inception (FID) score, proposed by Heusel et. al (2018), is a metric used to evaluate how well a GAN is performing, based on how similar the generated and real images are. The FID score calculates the distance between the feature vectors of real and fake images, and thus, lower FID scores represent higher quality images. More specifically, the FID is the squared Wasserstein metric between two multidimensional Gaussian distributions: the distribution of the feature vectors from the GAN and the distribution of feature vectors from the real images (Panaretos & Zemel, 2019, p. 1-4). The feature vectors are approximated by using the activations of the pool3 layer from the Inception-v3 image recognition model. FID is given by:

$$\mathrm{FID} = ||\mu_r - \mu_g||^2 + \mathrm{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}),$$
$$X_r \sim \mathcal{N}(\mu_r, \Sigma_r) \quad X_g \sim \mathcal{N}(\mu_g, \Sigma_g)$$

X_r and X_g are the "2048-dimensional activations of the Inception-v3 pool3 layer for the real and generated samples," respectively (p. 6).

In order to calculate the FID score between the GAN-generated dataset and the original RSNA dataset, an open source port of the official implementation of FID -- PyTorch-FID -- was used (Seitzer, 2020). The calculated FID score was 166.755. For reference, a DCGAN trained and tested on the CIFAR-10 dataset reported an FID score of 35.6; a Progressively-Growing GAN had an FID score of 7.3 on the CELEBA-HQ dataset; and StyleGAN2, another state-of-the-art GAN architecture, obtained an FID score on the FFHQ dataset of 2.84 (Shmelkov et. al., 2020, p. 10; Karras et. al, 2018, p. 19; Karras et al., 2020, p. 4).

Experimental Results

Each of the classifiers delineated in the experimental schedule were tested on the test (i.e. MURA validation) set. The reported accuracy is an average of 5 independent training-test runs.

Percent-Added Synthetic Data	Accuracy
0	0.831
10	0.826
50	0.8
100	0.775

Table 5: Experimental results

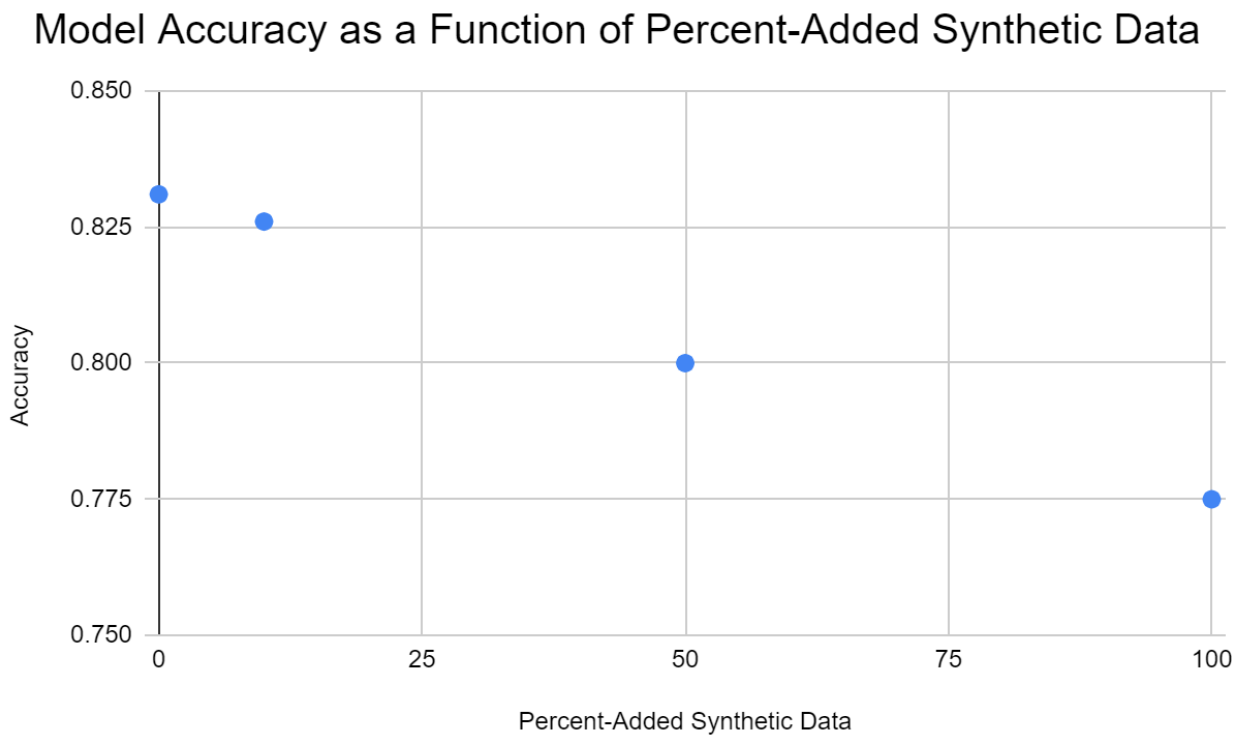

Figure 7: Reported model accuracy as a function of percent-added synthetic data

Discussion

GAN Output

In general, the output of the proposed GAN visually matched the RSNA dataset. As shown below in Figure 8, the GAN produced outputs that can clearly be identified as x-rays and, further, as accurately depicting hands. Moreover, the x-rays are generally anatomically correct and, at least to the untrained eye, appear almost indistinguishable from the original RSNA dataset. A clear separation between the forearm bones, wrist, hand bones, and finger bones can be seen. It should also be noted that certain secondary x-ray features, such as lettering and general smears were also accurately reproduced, which, although is not an important feature for normality classification, demonstrates how accurately the GAN is capturing the input training data. Similarly, the GAN performed well at producing images of varying exposures (e.g. general lightness or darkness), which was seen within the input dataset.

After choosing a random sample of 20 x-rays 20 times, approximately 90% of the generated images fall into the so-called visually accurate category.

Figure 8: visually accurate outputs from the proposed DCGAN

Approximately 10% of the time, the GAN exhibited some noticeable issues. These issues primarily fall into a few major categories shown below in Figure 9.

Figure 9: visually inaccurate outputs from the proposed DCGAN

For each of the following issues, it should be noted that the described characteristics were *not* observed in the training RSNA dataset. For example, all of the images in this dataset exhibit five proportionate fingers. A primary issue of the GAN output is that fingers were occasionally missing. A similar issue is that fingers appeared disproportionately long or short, especially relative to the palm size. On rare occasions, the output would be nearly completely visually dissimilar to the training dataset, with no clear forearm or finger bone separation. An extreme example of this is in the second row, third column, where the image can hardly be recognized as a hand. The GAN also had difficulty distinguishing between the foreground x-ray and the black background, as shown in the second row, first column.

One major, yet very rarely observed, concern was odd background presentation, seen in the second row, fourth column. Some of the malformed images also exhibited issues in capturing minute detail, with fingers often not exhibiting bone separation (see first row, third column). Moreover, as seen in the images above, the separation between the patient's hand and the background displayed a bit of blurriness that was not observed often in the training data. In general, however, the GAN's inability to capture fine detail was not a concern and clear finger and forearm bone separation and skin-bone separation was present in the vast majority of the images.

Given the short time frame of this project and the fact that GANs are extremely hard to train, the output of the proposed GAN was quite acceptable. The impact of malformed output on the classifier is discussed below, but it should be again noted that the vast majority (approximately 90%) of the data was visually acceptable. Given that the author is not a radiologist, however, it is difficult, if not impossible, to ascertain whether the

GAN output is truly anatomically accurate. Future work should attempt to recruit a radiologist to further examine the data and determine its worth.

Classifier Data Augmentation

The experimental results of synthetic data augmentation are found above in Table 5 and Figure 7. As a baseline, Rajpurkar et al. (2018, p. 5) report an accuracy (κ) of 0.851 on the hand study type. With no data augmentation, the model achieved a slightly lower performance of 0.831, but this is within the reported confidence interval of (0.830, 0.871). With 10-percent added synthetic data, model accuracy dropped by approximately 0.005 to 0.826. With 50-percent added synthetic data, model accuracy further dropped by 0.031 to 0.800. Lastly, in the 100-percent added synthetic data trial, model accuracy fell by 0.056 to 0.775.

It is quite clear that data augmentation was not effective and that adding an increasing amount of synthetic data lowered model performance. Given the visually accurate nature of the training data, this was surprising, but it is possible that the malformed outputs from the GAN looked similar to abnormal x-rays and confounded the classifier. Moreover, as discussed below, the GAN generated examples from a dataset that was not MURA, and this dataset mixing also likely had a negative effect. Lastly, the GAN-generated images were of lower resolution than the MURA images, and important details were likely lost during this downsampling.

Methodology Concerns

Class Imbalance

A major issue with the methodology presented is that the synthetic data only contained images of the normal class. The original MURA dataset is already imbalanced, with 3,653 normal instances and 1,336 abnormal instances and adding synthetic data exacerbated this issue. Class imbalance has been shown to negatively affect object classification models, and future work should be focused on attempting to generate both normal and abnormal instances in order to reduce the pre-existing class imbalance issue (Phan & Yamamoto, 2020, p.3). This would change the scope of the paper, however, as the RSNA dataset does not contain abnormal studies, though the original MURA dataset could be utilized.

Dataset Confusion

A concern about this work is that using images from a different dataset is problematic because the datasets may not be that similar. While this is true, the datasets appeared to the untrained eye as being quite similar, and the FID metric -- using Seitzer (2020) -- between the RNSA dataset and the MURA dataset was found to be 123.32.

In addition, medical imaging datasets are relatively rare, and while this paper was not successful, it is logical to believe that utilizing multiple, disparate datasets to train classifiers may improve outcomes. In fact, an argument can be made that other datasets may contain more, different information that can make the existing classifier more robust on the original dataset.

Paper-Specific Downsampling

The output of the proposed GAN was images of size 128x128, but the MURA classifier used images of size 224x224. This upsampling is generally bad practice due to resolution loss as well as introducing more fundamental differences between the higher-quality MURA images (all approximately 400x512) and the lower-quality GAN images (128x128). Since fine detail is of great importance in x-rays, upsampling is particularly detrimental, but it was believed that enough information could be preserved to overcome this issue. Moreover, if successful, this paper may have shown that lower-quality images could have beneficial effects regarding classifier performance, but this was not observed. It is difficult to quantify exactly what detail was lost with this resolution decrease, but it is likely that it played a major role in decreasing classifier accuracy.

Unfortunately, downsampling was needed due to enormous computational burdens imposed by generating higher quality images from the GAN, with training likely taking on the order of days. With more advanced GAN architectures, training would have taken even longer; regardless, future work should look to address this issue.

General Downsampling

Another concern about using GANs for generating medical images is their general requirement of downsampling the medical images. Downsampling is of particular concern because even the smallest of details in medical images can convey some important meaning in diagnosing (or not diagnosing) the patient's condition. The existing literature has shown, however, that classifiers trained on downsampled data are still highly performant, although higher quality images would provide classifiers with more information to consider when making a decision. For now, even with these

concerns, computational and time limitations dictate the necessity of downsampling. Future work can look to avoid this issue.

Overall, the methodology concerns presented are troubling and were likely the primary reason as to why this paper was not successful. Fortunately, many of the issues described can be assuaged given proper resources and time.

Conclusion and Future Work

The GAN, as described in the paper, proved to be largely capable of the desired, but data augmentation was found to be detrimental to MURA classifier accuracy. Although a variety of methodology concerns have already been discussed, the most prominent issues are: the inability to generate abnormal images (resulting in a greater training dataset imbalance), the unknown medical accuracy and worth of the GAN output, image downsampling, and dataset confusion. Fortunately, all of these issues can be addressed in future work. Moreover, past applications of GANs in this field show great promise, so the results of this paper, which was time and resource constrained, should be taken more lightly.

Given the critical nature of medicine, GANs, however, are far from the elixir to fix all lacking datasets, and they have a long way to go in order to secure the trust of clinicians. Most GANs still rely on deep learning-based architectures, which are not well understood. Training GANs is both extremely compute-intensive (which may limit their widespread adoption) and difficult, with problems such as mode collapse and arduous hyperparameter tuning particularly plaguing the medical imaging domain (Skandarani et al., 2021, p. 5, 8).

The highly specialized nature of medical imaging also greatly negatively impacts the usefulness of GANs. It is difficult to evaluate how accurate and useful the output of a medical imaging GAN is. In regular images, it is usually quite easy to visually and quantitatively understand how well the GAN is performing. However, the specialized and fine-grained nature of medical imaging renders this quite difficult, particularly given that medical images can have many, challenging-to-distinguish modes. In addition, each detail on a medical image usually has a specific interpretation, but this cannot be said for the output of GANs. Lastly, and likely most importantly, the images generated by GANs do not have a defined medical worth. That is, because they have no associated patient or study, it is difficult to determine whether the synthetic images have any true medical meaning, outside of potentially improving classifiers.

In general, the GAN model presented here, and those presented in much of the state-of-the-art medical imaging literature, can be thought of as black boxes. Thus, as long as they can improve the accuracy of classifiers and/or other indirect methods, they have worth, even if the images themselves are difficult or impossible to interpret. Generally, this is a questionable tactic, as feeding in mediocre data will almost universally lead to mediocre results, particularly if the process and data at hand are not well-understood. In this sense, GANs can be best used in medical imaging scenarios as supplements to radiologists' toolkits.

Outside of medical imaging, the images generated by GANs can also demonstrate fundamental issues in their construction. Particularly, GANs struggle with presenting symmetry accurately. In their article for the New York Times, Hill & White (2020) show how eye glasses, ears, and earrings often appear mismatched in synthetic images trained on human faces. However, in other features, such as eyes and mouths, symmetry is preserved to an unrealistic extent. This may be of concern in medical imaging, where bodies often deviate from perfect symmetry and thus, the images generated by GANs may not be realistic. Hill & White also demonstrate how GANs can produce random artifacts and surreal backgrounds, though this is likely less of an issue for medical images, where the backgrounds tend to be monotone.

Another issue with GANs, as discussed above, is that they often take as input and subsequently output downsampled images. Future work should attempt to mitigate this problem by achieving similar fidelity to traditionally-captured medical images. This is difficult because of the computation and space requirements needed, but, with more efficient GAN architectures, this can hopefully be achieved. In addition, future work could attempt to utilize more recent GAN architectures, such as progressively-growing GANs, in order to mitigate some of the issues seen in the generated images.

Further research off of this paper should also utilize better-performing and/or different models than just Rajpurkar et al's. Recent work by Ananda et al. (2021) has demonstrated the possibility of using models besides Rajpurkar et al. for successful medical classification. Moreover, the top submissions on the MURA challenge could be used, assuming high-quality code can be found.

Acknowledgements

I want to especially thank my advisor, Dr. Paul Yushkevich, as well as Pulkit Khandelwal and Dr. Norman Badler for their guidance throughout the development and completion of my thesis. I would also like to thank Long Xie, Gaylord

Holder, and the PICSL group at the University of Pennsylvania for their help in onboarding my work onto a GPU cluster.

Glossary

Adam Optimizer: a common optimizer algorithm that can be used in place of stochastic gradient descent in order to update the weights of a neural network (Kingma & Ba, 2017). Adam is one of the most commonly used optimizer algorithms for computer vision and GANs. Compared to stochastic gradient descent, Adam requires less memory and is more computationally efficient (Kingma & Ba, 2017).

AUROC (AUC) Values: refers to the area under the ROC curve (Rajpurkar et al., 2018, p. 5-6). The ROC curve plots the true positive rate against the false positive rate of a classifier (Rajpurkar et al., 2018, p. 5-6). Thus, a higher AUROC (AUC) value implies a classifier that has a smaller false positive rate, and a value of 1 would imply that the classifier never outputs a false positive (in regards to true positives).

Batch Norm(alization): a layer in a CNN that normalizes the input (i.e. near-0 mean and near-1 standard deviation) (Ioffe & Szegedy, 2015). While not well understood, batch norm increases neural network performance and stability (Ioffe & Szegedy, 2015).

(Strided) Convolutional Layers: the basic building block of CNNs, which applies the convolutional operation to a given input (O'Shea & Nash, 2015, p. 5-7). In the convolutional layer, a kernel (filter) slides across the input data (such as an image) and performs element wise multiplication, which effectively sums up the input into a single output (O'Shea & Nash, 2015, p. 5-7). In this sense, convolutional layers both reduce the input dimensionality and extract features by attempting to learn the weights associated with the filters (O'Shea & Nash, 2015, p. 5-7). The stride of a convolution is the distance between each successive location where the kernel is applied (O'Shea, 2015, p. 5-7). Thus, a larger stride results in more dimensionality reduction.

Convolutional Neural Network (CNN): a type of (deep) neural network that utilizes convolutional layers (see above) in order to extract features from a given input (generally images) (O'Shea & Nash, 2015, p. 2-5). CNNs work by passing the input of one convolutional layer to another (almost always buffered by pooling (see below), normalization, or activation layers) (O'Shea & Nash, 2015, p. 2-5).

(Strided) Convolutional Transpose Layer: a type of reverse convolution, which upsamples and interpolates the data to fill in coarse portions (Radford et al., 2016, p. 2). See Radford et al. (2016, p. 2) for more information.
CT, PET, and MR Images: Positron Emission Tomography (PET) scans use radioactive dye in order to identify proper organ and tissue function. Computed Tomography (CT) scans are similar to an x-ray, but capture greater three-dimensional details of organs, bones, and tissues. Magnetic Resonance (MR) imaging utilizes magnets and radio waves in order to create a detailed image of internal organs. For more information, please consult Liu et al. (2017)

Fully-Connected Network/Layers: a neural network in which each neuron in one layer is connected to every neuron in the next layer (O'Shea & Nash, 2015, p. 4). A layer in which each neuron is connected to every neuron in the previous and next layers is referred to as a fully-connected layer (O'Shea & Nash, 2015, p. 8).

Learning Rate: in gradient descent and other optimization algorithms (e.g. Adam), this is the amount by which the weights are updated during an iteration of training (Wu et al., 2019, p.1). Often, learning rates are decayed during training, which tends to improve training optimization (Wu et al., 2019, p. 2).

Mini-Batch Gradient Descent (Minibatches): a form of gradient descent in which small batches of the training data are used to compute the model's loss and update the model's weights (Li, 2020, p. 1-2). These small batches are known as minibatches, and mini-batch gradient descent is generally far more computationally efficient than traditional gradient descent, which utilizes every example from the training dataset (Li, 2020, p. 1-2).

Pooling/Pooling Layer: pooling refers to the operation of downsampling a given input by outputting a summary statistic for a chosen region (O'Shea & Nash, 2015, p. 5). Two common summary statistics are the max and average of a given area (O'Shea & Nash, 2015, p. 5). A pooling layer in a CNN takes a two-dimensional input and, by applying a summary statistic, outputs a lower resolution version of the input (O'Shea & Nash, 2015, p. 5).

ReLU and LeakyReLU functions: a piecewise nonlinear activation function (in neural networks) that outputs the given input if it is greater than 0 and 0 otherwise. ReLU is among the most commonly used activation functions given its ease of training and increased performance gains. Leaky ReLU is effectively the same function, but outputs a small gradient in place of 0 (e.g. 0.01x) so as to avoid issues with sparse or extremely

small gradients. These scenarios are common in the GANs, and thus Leaky ReLU is one of the most common GAN activation functions (Nwankpa, 2018, p. 8-9).

Sigmoid: a non-linear activation function (in neural networks) that takes any real value as input and outputs values in the range of 0 to 1. Due to its biological basis and historical use, the sigmoid activation function is among the most commonly used activation functions in neural networks (Nwankpa, 2018, p. 5).

Tanh function: a non-linear activation function (in neural networks) that utilizes the hyperbolic tangent function, which takes any real valued input and outputs values in the range of -1 to 1. TanH is ideal for hidden layers, where it can act to “center” the data (Nwankpa, 2018, p. 7).

References

Agrahari, R. (2020). DenseNet-MURA-PyTorch: DenseNet on MURA Dataset using PyTorch. https://github.com/pyaf/DenseNet-MURA-PyTorch.

Ananda, A., Ngan, K., Karabağ, C., Ter-Sarkisov, A., Alonso, E., Reyes-Aldasoro, C. (2021). Classification and Visualisation of Normal and Abnormal Radiographs; A Comparison between Eleven Convolutional Neural Network Architectures. Sensors, 21(16).

Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., Dickie, D.A., Hernández, M.V., Wardlaw, J., Rueckert, D. (2018). GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks. https://arxiv.org/pdf/1810.10863.pdf.

Chahal, K.S., Dey, K. (2018). A Survey of Modern Object Detection Literature using Deep Learning. https://arxiv.org/pdf/1808.07256.pdf.

Cirillo M.D., Abramian, D., Eklund, A. (2020). Vox2Vox: 3D-GAN for Brain Tumour Segmentation. https://arxiv.org/pdf/2003.13653.pdf.

Dai, W., Doyle, J., Liang, X., Zhang, H., Dong, N., Li, Y., Xing, E. (2017). SCAN: Structure Correcting Adversarial Network for Organ Segmentation in Chest X-rays. https://arxiv.org/pdf/1703.08770.pdf.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F. (2009). ImageNet: a Large-Scale Hierarchical Image Database. In IEEE Conference on Computer Vision and Pattern Recognition. 248-255.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). Generative Adversarial Networks. https://arxiv.org/pdf/1406.2661.pdf.

Halabi S., Prevedello L., Kalpathy-Cramer, J. (2018). The RSNA Pediatric Bone Age Machine Learning Challenge.

Han, T., Nebelung, S., Haarburger, C., Horst, N., Reinartz, S., Merhof, D., Truhn, D. (2019). Breaking Medical Data Sharing Boundaries by Employing Artificial Radiographs. bioRxiv. doi:10.1101/841619

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S. (2018). GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. https://arxiv.org/pdf/1706.08500.pdf.

Hill, K., White, J. (2019). Designed to Deceive: Do These People Look Real to You? https://www.nytimes.com/interactive/2020/11/21/science/artificial-intelligence-fake-people-faces.html.

Huang, G., Liu, Z., van der Maaten,L., Weinberger, K. (2018). Densely Connected Convolutional Networks. https://arxiv.org/pdf/1608.06993.pdf

Ioffe, S., Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

Islam, J., Zhang, Y. (2020). GAN-based synthetic brain PET image generation. Brain informatics, 7(1), 3. https://doi.org/10.1186/s40708-020-00104-2. https://pubmed.ncbi.nlm.nih.gov/32232602.

Karbhari, Y., Basu, A., Geem, Z. W., Han, G.T., Sarkar, R. (2021). Generation of Synthetic Chest X-ray Images and Detection of COVID-19: A Deep Learning Based Approach. Diagnostics, 11(5).

Karras, T., Aila, T., Laine, S., Lehtinen, J. (2018). Progressive Growing of GANs for Improved Quality, Stability, and Variation. https://arxiv.org/pdf/1710.10196.pdf.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T. (2020). Analyzing and Improving the Image Quality of StyleGAN. https://arxiv.org/pdf/1912.04958v2.pdf.

Kazeminia, S., Baur, C., Kuijper, A., van Ginneken, B., Navab, N., Albarqouni, S., Mukhopadhyay, A. (2019). GANs for Medical Image Analysis.

Kingma, D., Ba, J. (2017). Adam: A Method for Stochastic Optimization. https://arxiv.org/pdf/1412.6980.pdf.

Li, X., Pan, D., Li, X., Zhu, D. (2020). Improve SGD Training via Aligning Mini-batches. https://arxiv.org/pdf/2002.09917.pdf.

Liu, B., Gao, S., Li, S. (2017). A Comprehensive Comparison of CT, MRI, Positron Emission Tomography or Positron Emission Tomography/CT, and Diffusion Weighted Imaging-MRI for Detecting the Lymph Nodes Metastases in Patients with Cervical Cancer: A Meta-Analysis Based on 67 Studies. https://pubmed.ncbi.nlm.nih.gov/28183074.

Mahapatra, D., Bozorgtabar, B., Thiran, J., Reyes, M. (2019). Efficient Active Learning for Image Classification and Segmentation using a Sample Selection and Conditional Generative Adversarial Network. https://arxiv.org/pdf/1806.05473.pdf.

Menon, S., Galita, J., Chapman, D., Gangopadhyay, A., Mangalagiri, J., Nguyen, P., Yesha, Y., Yelena, Y., Saboury, B., Morris, M. (2020). Generating Realistic COVID19 X-rays with a Mean Teacher + Transfer Learning GAN. https://arxiv.org/pdf/2009.12478.pdf.

Mirza, M., Osindero, M. (2014). Conditional Generative Adversarial Nets. https://arxiv.org/pdf/1411.1784.pdf.

Moradi, M., Madani, A., Karargyris, A., Syeda-Mahmood, T. (2018). Chest X-Ray Generation and Data Augmentation for Cardiovascular Abnormality Classification.

Motamed, S., Rogalla, P., Khalvati, F. (2020). RANDGAN: Randomized Generative Adversarial Network for Detection of COVID-19 in Chest X-ray. https://arxiv.org/abs/2010.06418

Nwankpa, C., Ijomah, W., Gachagan, A., Marshall, S. (2018). Activation Functions: Comparison of trends in Practice and Research for Deep Learning.

Odena, A., Olah, C., Shlens, J. (2017). Conditional Image Synthesis With Auxiliary Classifier GANs. https://arxiv.org/pdf/1610.09585.pdf.

O'Shea, K, Nash, R. (2015). An Introduction to Convolutional Neural Networks. https://arxiv.org/pdf/1511.08458.pdf.

Panaretos, V., Zemel, Y. (2019). Statistical Aspects of Wasserstein Distances. Annual Review of Statistics and Its Application, 6(1), 405–431. https://arxiv.org/pdf/1806.05500.pdf.

Phan, T.H., Yamamoto, K. (2020). Resolving Class Imbalance in Object Detection with Weighted Cross Entropy Losses. https://arxiv.org/ftp/arxiv/papers/2006/2006.01413.pdf.

Radford, A., Metz, L., Chintala, S. (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. https://arxiv.org/pdf/1511.06434.pdf.

Rajpurkar, P., Irvin, J., Bagul, A., Ding, D., Duan, T., Mehta, H., Yang, B., Zhu, K., Laird, D., Ball, R., Langlotz, C., Shpanskaya, K., Lungren, M., Ng, A. (2018). MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs. https://arxiv.org/pdf/1712.06957.pdf.

Salehinejad, H., Valaee, S., Dowdell, T., Colak, E., Barfett, J. (2018). Generalization of Deep Neural Networks for Chest Pathology Classification in X-Rays Using Generative Adversarial Networks. https://arxiv.org/pdf/1712.01636.pdf.

Seitzer, M. (2020). pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid.

Shmelkov, K., Schmid, C., Alahari, K. (2018). How good is my GAN?. https://arxiv.org/pdf/1807.09499.pdf.

Skandarani, Y., Jodoin, P., Lalande, A. (2021). GANs for Medical Image Synthesis: An Empirical Study. https://arxiv.org/pdf/2105.05318.pdf.

Sundaram, S.,Hulkund, N. (2021). GAN-based Data Augmentation for Chest X-ray Classification. https://arxiv.org/pdf/2107.02970.pdf.

United States Bone and Joint Initiative. (2020). The Burden of Musculoskeletal Diseases in the United States (BMUS), Fourth Edition.

Waheed, A., Goyal, M., Gupta, D., Khanna, A., Al-Turjman, F., Pinheiro, P. (2020). CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection. IEEE Access, 8, 91916–91923. https://arxiv.org/ftp/arxiv/papers/2103/2103.05094.pdf.

Woolf, A., Pfleger, B. (2003). Burden of major musculoskeletal conditions. Bulletin of the World Health Organization, 81(9):646–656.

Wu, Y., Liu, L., Bae, J., Chow, K.H., Iyengar, A., Pu, C., Wei, W., Yu, L., Zhang, Q. (2019). Demystifying Learning Rate Policies for High Accuracy Training of Deep Neural Networks. https://arxiv.org/pdf/1908.06477.pdf.