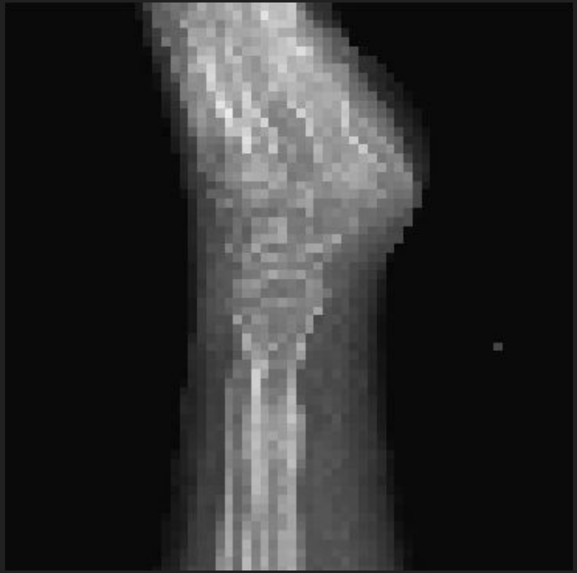
Leveraging GANs for Improved Medical Image Classification

Abdel Rahman Ibrahim
Adil Bukhari



Problem Statement

The availability of labeled training data plays a crucial role in model performance. Collecting this data at scale can be expensive and time consuming.

One option is to augment the data to increase the size of the dataset.

- ➢ Risk of overfitting and data leakage
- ➢ Limited scope of transformations
- ➢ Little variability in details and patterns

Proposal

We propose leveraging Generative Adversarial Networks to generate synthetic data as a means to increase the size and diversity of a given dataset.

By generating data samples that resemble the true data distribution, we can use it to augment the original dataset.

This will address the aforementioned limitations of traditional data augmentation, while potentially introducing variability into the training process.

Methods

Utilizing a dataset of fractured and unfractured bone X-Ray images, we will train a GAN to generate synthetic data that closely resembles the real images from the dataset.

~10,500 images of positive and negative samples of the bone fractures data will be generated.

The usability of the synthetic data will be evaluated across multiple experiments using a Convolutional Neural Network to classify the data.

Architecture - Generative Adversarial Network

Model: "generator"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 8192)	1,056,768
reshape (Reshape)	(None, 8, 8, 128)	0
conv2d_transpose (Conv2DTranspose)	(None, 16, 16, 128)	262,272
leaky_re_lu_3 (LeakyReLU)	(None, 16, 16, 128)	0
conv2d_transpose_1 (Conv2DTranspose)	(None, 32, 32, 256)	524,544
leaky_re_lu_4 (LeakyReLU)	(None, 32, 32, 256)	0
conv2d_transpose_2 (Conv2DTranspose)	(None, 64, 64, 512)	2,097,664
leaky_re_lu_5 (LeakyReLU)	(None, 64, 64, 512)	0
conv2d_3 (Conv2D)	(None, 64, 64, 3)	38,403

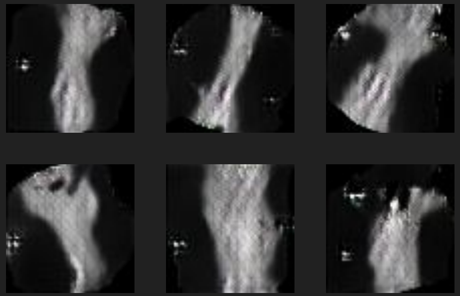
Total params: 3,979,651 (15.18 MB)
Trainable params: 3,979,651 (15.18 MB)
Non-trainable params: 0 (0.00 B)

Model: "discriminator"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 64)	3,136
leaky_re_lu (LeakyReLU)	(None, 32, 32, 64)	0
conv2d_1 (Conv2D)	(None, 16, 16, 128)	131,200
leaky_re_lu_1 (LeakyReLU)	(None, 16, 16, 128)	0
conv2d_2 (Conv2D)	(None, 8, 8, 128)	262,272
leaky_re_lu_2 (LeakyReLU)	(None, 8, 8, 128)	0
flatten (Flatten)	(None, 8192)	0
dropout (Dropout)	(None, 8192)	0
dense (Dense)	(None, 1)	8,193

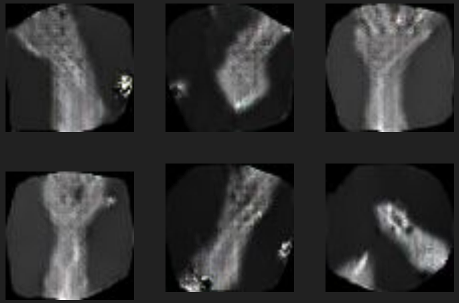
Total params: 404,801 (1.54 MB)
Trainable params: 404,801 (1.54 MB)
Non-trainable params: 0 (0.00 B)

GAN Results - 100 Epochs



Fractured

10,500 Generated Images



Unfractured

10,500 Generated Images

Experiments

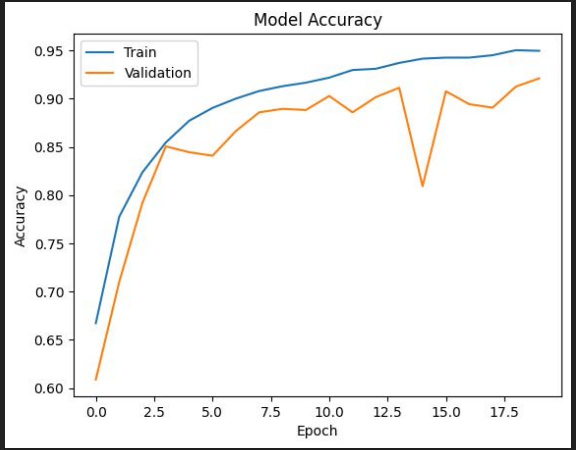
We conduct four experiments to examine the impact of synthetic data on generalizability performance of the classifier.

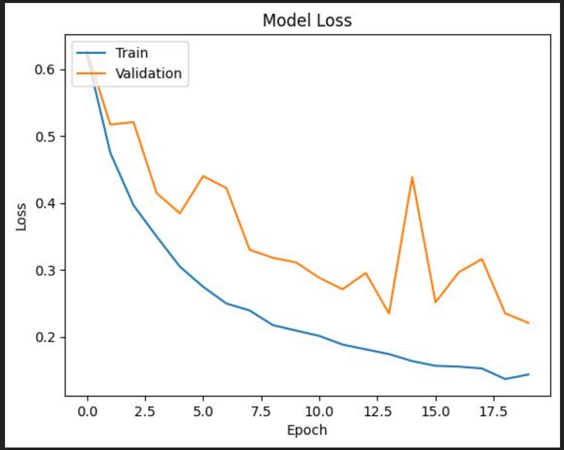
- ➢ Train CNN with original dataset
- ➢ Train CNN with augmented dataset
- ➢ Train CNN with synthetically-augmented dataset
- ➢ Train CNN on original dataset after warm-starting it with synthetic data

Architecture - Convolutional Neural Network

```
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_14 (Conv2D)          (None, 21, 21, 32)        320

 batch_normalization_14 (Ba  (None, 21, 21, 32)        128
 tchNormalization)

 conv2d_15 (Conv2D)          (None, 7, 7, 16)          4624

 max_pooling2d_7 (MaxPoolin  (None, 2, 2, 16)          0
 g2D)

 batch_normalization_15 (Ba  (None, 2, 2, 16)          64
 tchNormalization)

 dropout_14 (Dropout)        (None, 2, 2, 16)          0

 flatten_7 (Flatten)         (None, 64)                0

 dense_14 (Dense)            (None, 128)               8320

 dropout_15 (Dropout)        (None, 128)               0

 dense_15 (Dense)            (None, 1)                 129

=================================================================
Total params: 13585 (53.07 KB)
Trainable params: 13489 (52.69 KB)
Non-trainable params: 96 (384.00 Byte)
```

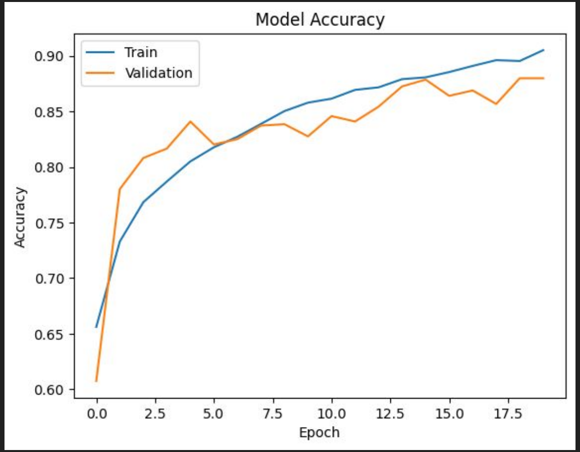
Results - Original Dataset

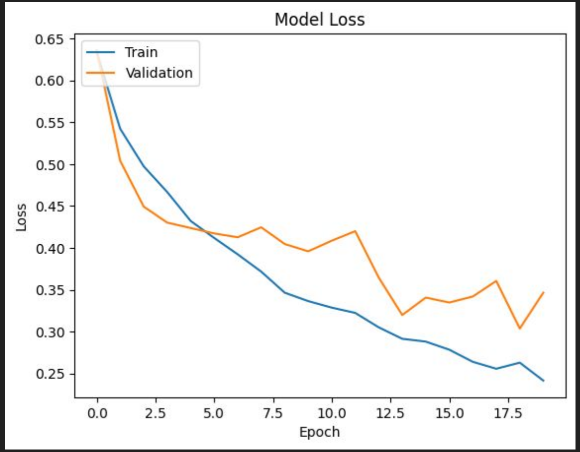




92% after 20 Epochs

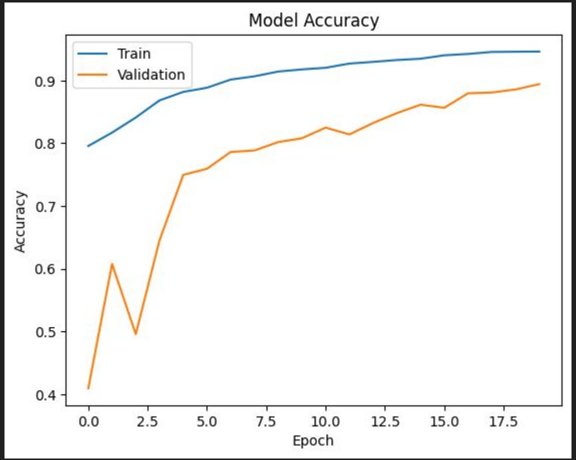
Results - Augmented Data (Original)

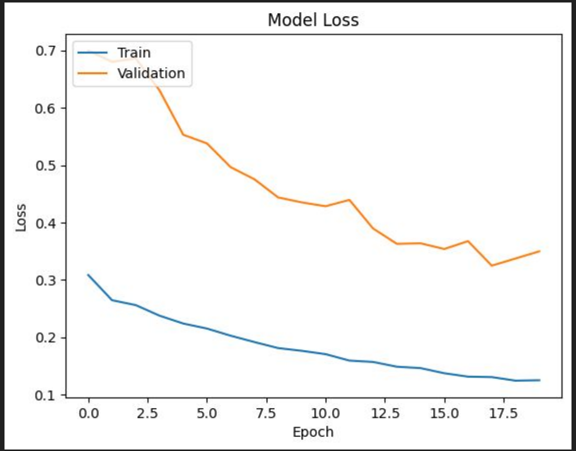




87% after 20 Epochs

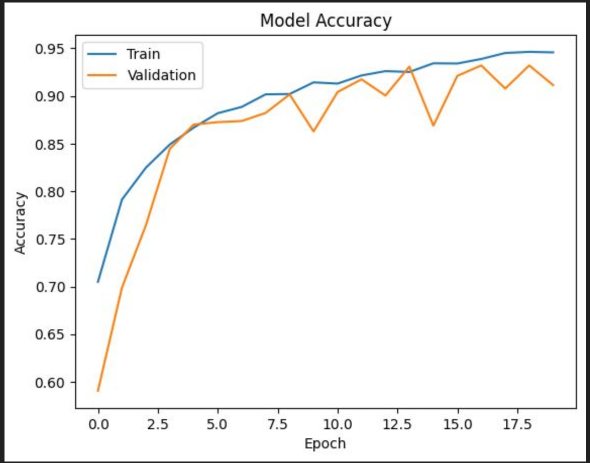
Results - Augmented Data (Synthetic + Original)

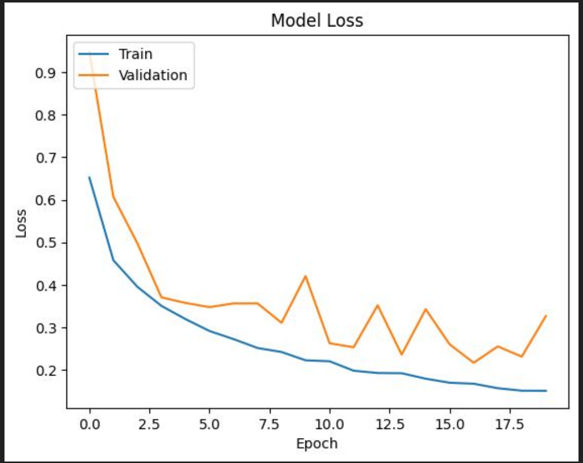




91% after 20 Epochs

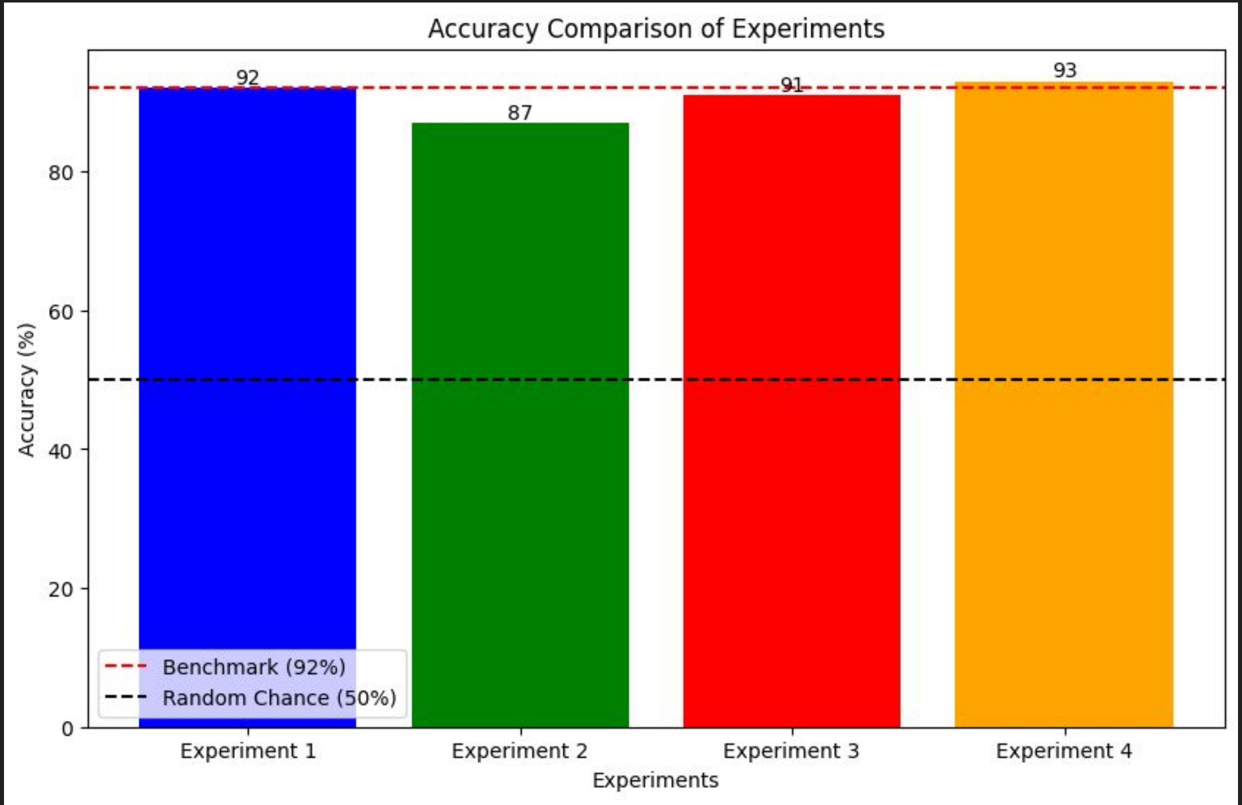
Results - Original Dataset with Synthetic Warm Start





93% after 20 Epochs

Results - Comparison of Accuracies



Evaluation

- ➢ Synthetically-augmented data was more viable in our experiment than traditional augmentation methods.
  - We believe that, given a more advanced GAN, this can be improved. Ex: CycleGAN, StyleGAN, etc.
- ➢ Using weights obtained from the synthetic data training, however, yielded an improvement in accuracy and loss.
  - The model was likely able to pick up features from the synthetic data and avoided needing to start from scratch.