Unsupervised Anomaly Detection on X-Ray Images

Jeyasri Subramanian - Subarna Chowdhury Soma - Pranav Lodha

Abstract

AI (ML) and Deep Learning (DL) for medical services is an exceptionally dynamic region of examination in both the scholarly world and industry these days. ML and DL are promising in manners that they help specialists/analysts in finding new remedies for sicknesses that are presently serious, or they can increase doctors constantly to perform quicker and better.

Be that as it may, ML and DL have a solid desire for information, particularly if there should arise an occurrence of clinical information. Irregular clinical cases are typically a lot more uncommon than ordinary cases, so it is generally intensely slanted toward ordinary cases (negative examples). It is very tedious to gather a sensible sum for all cases. For directed ML/DL approach, the measure of information gathered should be named by qualified doctors/specialists also. So, it's over the top expensive to make names for clinical datasets.

A DL model that can perform moderately well—not really outflanks specialists, however, doesn't need a lot of adjusted information and excellent names can acquire gigantic points of interest terms of cost and speed to any medical care framework. In our project we will focus on exploring how to find anomalies within X-rays.

Introduction

With the significant amount of time required to label and audit the dataset for healthcare purposes, Machine learning and Deep Learning lose their effectiveness. To help solve this issue we take a look at Generative Adversarial Networks (GAN).

To produce reasonable examples in a multi-modular circulation of pictures, GAN's Generator and Discriminator more likely than not adapted progressively the significant level highlights of information, so it can recognize genuine and engineered tests.

All the more significantly, the names to prepare the Discriminator come for nothing (this example is genuine or counterfeit), so the educated highlights would be incredibly helpful when preparing names are difficult to get.

Be that as it may, can these take in highlights from GANs be utilized for something different, for example, solo peculiarity location? GAN can absolutely cool stuffs, yet would it be able to do helpful stuff as well?

This will allow us to put more focus on making technology much more available and help make it easier to detect anomalies in X-ray images. With this information we can then speed up the process of identifying issues and potentially saving more people's lives.

In next part, we will investigate utilizing these educated highlights to assemble a solo peculiarity discovery model on X-beam pictures.

Literature Survey

The following papers will help us solve our problem statement or unsupervised anomaly detection on X-Ray images.

- GANS (Generative Adversarial Network)
- Knowledge transfer on what a GAN is and how each GAN differs.
- https://en.wikipedia.org/wiki/Generative_adversarial_network

Problem Statement

Problem Statement

In our project we plan on using Generative Adversrial Network (GAN). A GAN is a class of AI systems, where Two neural organizations challenge with one another in a game (as a lose-lose situation, where one specialist's benefit is another specialist's misfortune). For our project we will be using two GANs, AlphaGAN and AnoGAN in-order to detect anomalies in the X-ray images.

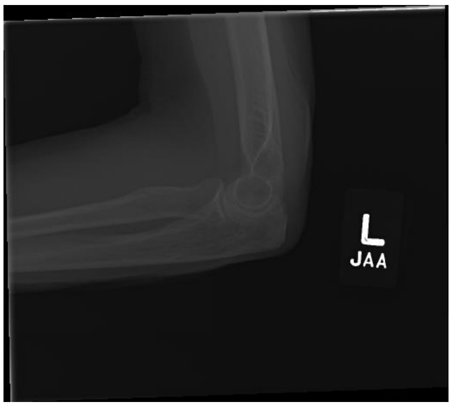
- AlphaGAN: The generator of this network is a convolutional encoder-decoder network that is trained both with help of the ground-truth alphas as well as the adversarial loss from the discriminator, and the discriminator is a patchGAN Discriminator.
- AnoGAN: the firstly proposed method using GAN for anomaly detection. The generator of GAN is trained to produce patches and fit the data distribution. Based on the second loss, the generator takes not only the information to fool the discriminator but the rich information of the feature representation

Dataset

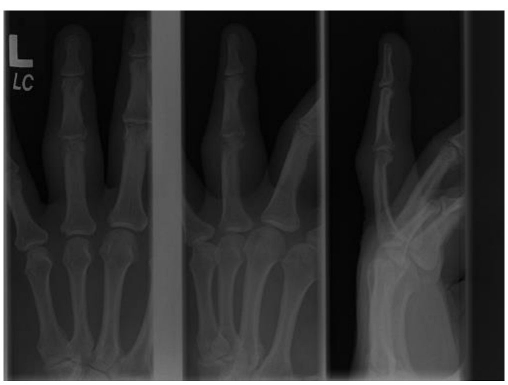
For our project we will be utilizing the musculoskeletal radiographs (MURA). This is a large dataset of X-rays consisting of 14,863 studies from 12,173 patients, with a total of 40,561 multiview radiographic images. Each belongs to one of seven standard upper extremity radiographic study types

https://stanfordmlgroup.github.io/competitions/mura/

- Elbow
    - Positive: 2006
    - Negative: 2925



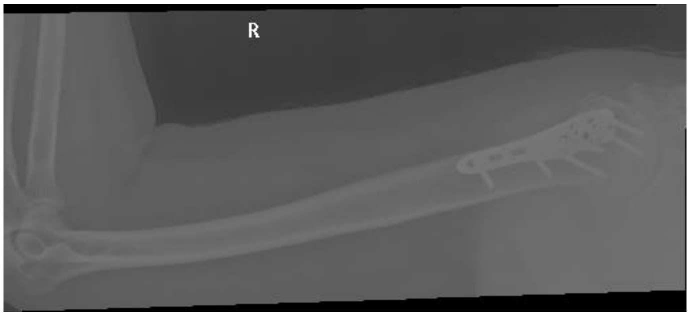
- Finger
    - Positive: 1968
    - Negative: 3138



- Forearm
    - Positive: 661
    - Negative: 1164

- Hand
  - Positive: 1484
  - Negative: 4059

- Humerus
  - Positive: 599
  - Negative 673



- Shoulder
    - Positive: 3987
    - Negative: 5765

- Wrist
    - Positive: 3987
    - Negative: 21935

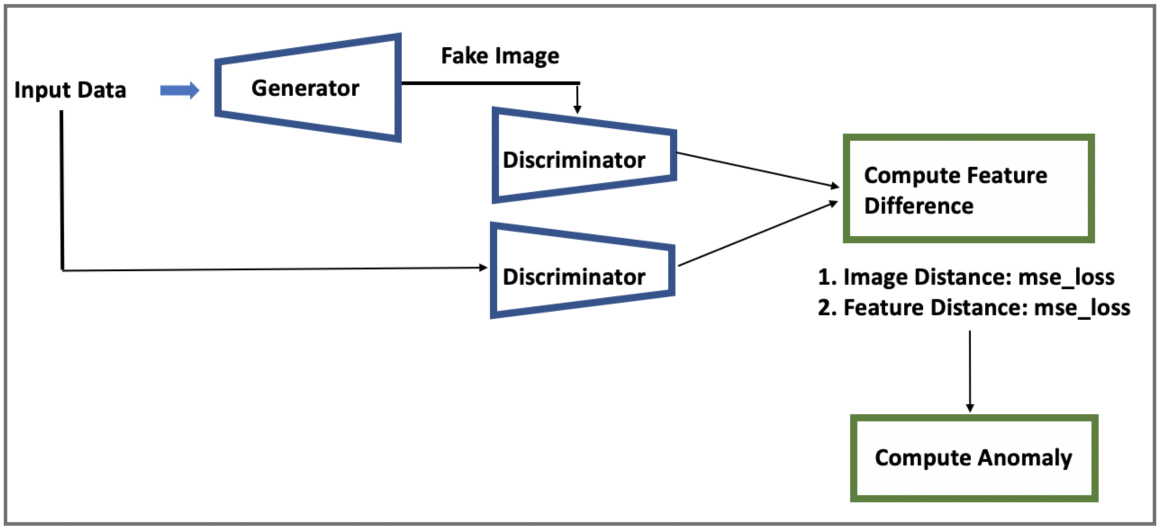
This dataset contains over four gigabytes worth of x-ray images and provides an very strong foundation in-order for us to train our model.

Methodology

Our basic strategy to tackle the problem will be to utilize an Generative Adversarial Network (GAN) in-order to achieve our goal. A GAN is the perfect approach due to its two components (Generator / Discriminator) competing neural networks.

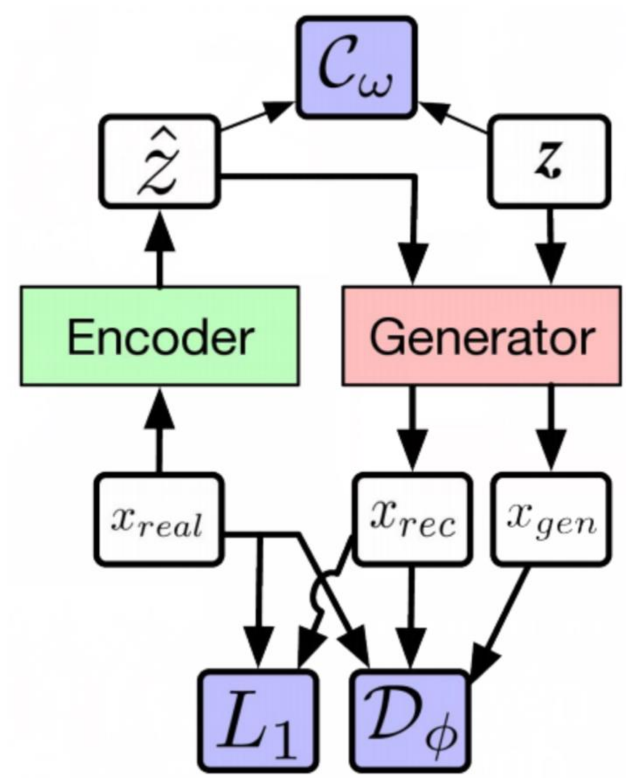
For this problem we will train a GAN to generate only normal (negative) x-ray images. Then when we are predicting the anomaly, we will use the GANs to reconstruct the input images of both normal and abnormal images (positive). After that we will compute the reconstruction, feature matching and discrimination losses.

We will initially begin with the strategy below so that the generator outputs random image samples from random latent vectors, and our discriminator classifies real and fake samples



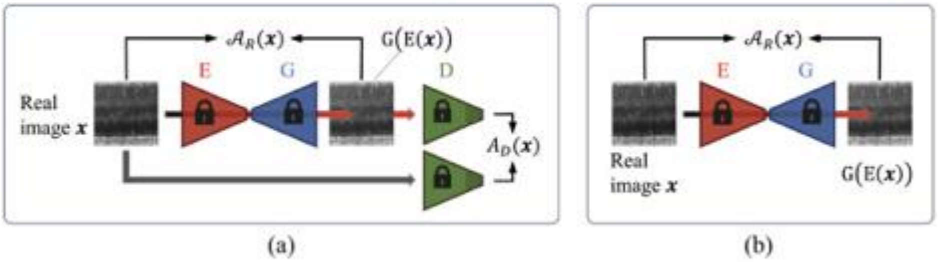
AlphaGAN

We will take the vanilla architecture an attempt to combine the auto-encoder with the GAN architecture. This will be our Alpha-GAN architecture.



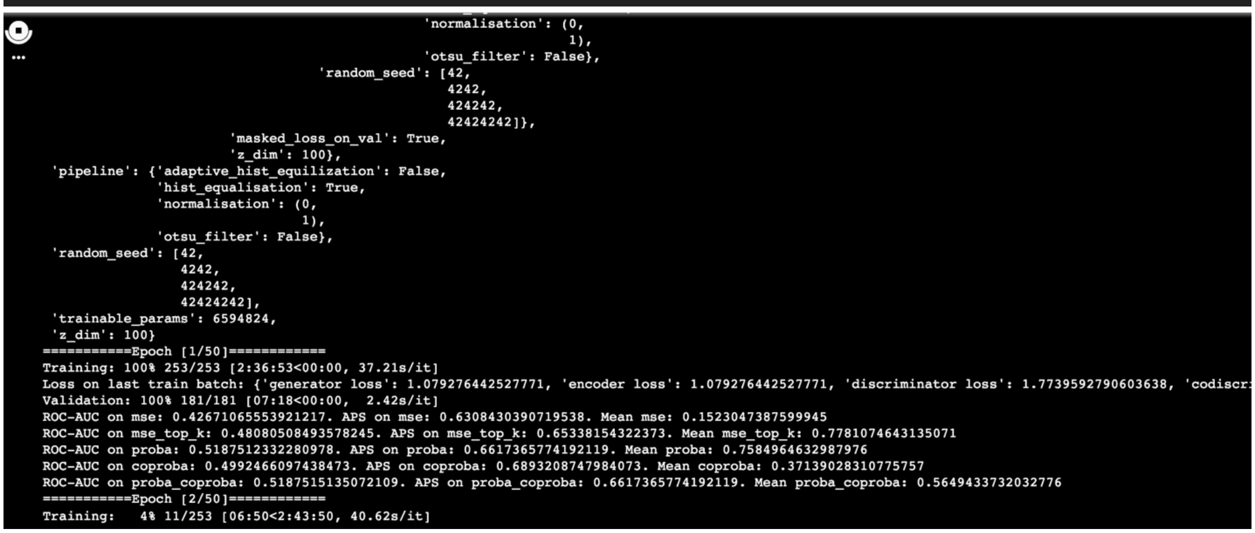
AnoGAN

Similarly to AlphaGAN we also compared our results with AnoGAN. Below is the architecture for the model.



Google Cloud Platform Training

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to euphoric-diode-296917.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
pranav_lodha@cloudshell:~ (euphoric-diode-296917)$ export PROJECT_ID=euphoric-diode-296917
pranav_lodha@cloudshell:~ (euphoric-diode-296917)$ gcloud config set project $PROJECT_ID
Updated property [core/project].
pranav_lodha@cloudshell:~ (euphoric-diode-296917)$
```



```
                                                  'normalisation': (0,
                                                                    1),
                                                  'otsu_filter': False},
                                      'random_seed': [42,
                                                      4242,
                                                      424242,
                                                      42424242]},
                           'masked_loss_on_val': True,
                           'z_dim': 100},
     'pipeline': {'adaptive_hist_equilization': False,
                  'hist_equalisation': True,
                  'normalisation': (0,
                                    1),
                  'otsu_filter': False},
     'random_seed': [42,
                     4242,
                     424242,
                     42424242],
     'trainable_params': 6594824,
     'z_dim': 100}
    ===========Epoch [1/50]============
    Training: 100% 253/253 [2:36:53<00:00, 37.21s/it]
    Loss on last train batch: {'generator loss': 1.079276442527771, 'encoder loss': 1.079276442527771, 'discriminator loss': 1.7739592790603638, 'codiscr
    Validation: 100% 181/181 [07:18<00:00,  2.42s/it]
    ROC-AUC on mse: 0.42671065553921217. APS on mse: 0.6308430390719538. Mean mse: 0.1523047387599945
    ROC-AUC on mse_top_k: 0.48080508493578245. APS on mse_top_k: 0.65338154322373. Mean mse_top_k: 0.7781074643135071
    ROC-AUC on proba: 0.5187512332280978. APS on proba: 0.6617365774192119. Mean proba: 0.7584964632987976
    ROC-AUC on coproba: 0.4992466097438473. APS on coproba: 0.6893208747984073. Mean coproba: 0.37139028310775757
    ROC-AUC on proba_coproba: 0.5187515135072109. APS on proba_coproba: 0.6617365774192119. Mean proba_coproba: 0.5649433732032776
    ===========Epoch [2/50]============
    Training:   4% 11/253 [06:50<2:43:50, 40.62s/it]
```

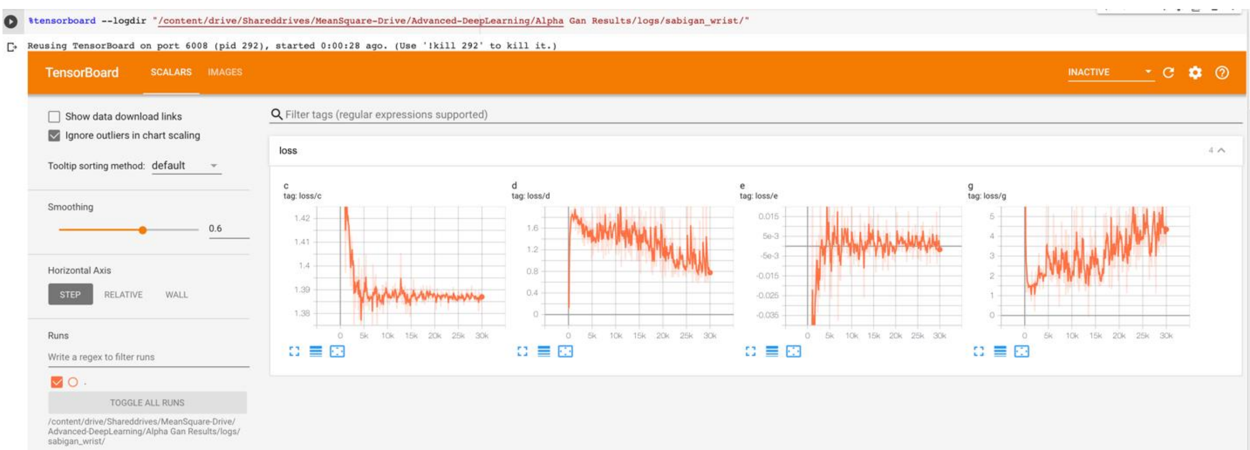
Generated Model

▼ logs	Today at 4:40 PM	--	Folder
▼ sabigan_wrist	Nov 21, 2020 at 12:32 PM	--	Folder
events.out.tfevents.1605990764.c3f1388974c4	Nov 21, 2020 at 7:12 PM	94.1 MB	Document
▼ models	Nov 21, 2020 at 8:10 PM	--	Folder
▼ sabigan_wrist	Nov 21, 2020 at 7:10 PM	--	Folder
2300_C.pth	Nov 21, 2020 at 1:03 PM	43 KB	Document
2300_D.pth	Nov 21, 2020 at 1:03 PM	12.8 MB	Document
2300_E.pth	Nov 21, 2020 at 1:03 PM	21.1 MB	Document
2300_G.pth	Nov 21, 2020 at 1:03 PM	15.4 MB	Document
4600_C.pth	Nov 21, 2020 at 1:33 PM	43 KB	Document
4600_D.pth	Nov 21, 2020 at 1:33 PM	12.8 MB	Document
4600_E.pth	Nov 21, 2020 at 1:33 PM	21.1 MB	Document
4600_G.pth	Nov 21, 2020 at 1:33 PM	15.4 MB	Document
6900_C.pth	Nov 21, 2020 at 2:04 PM	43 KB	Document
6900_D.pth	Nov 21, 2020 at 2:04 PM	12.8 MB	Document
6900_E.pth	Nov 21, 2020 at 2:04 PM	21.1 MB	Document

Evaluations

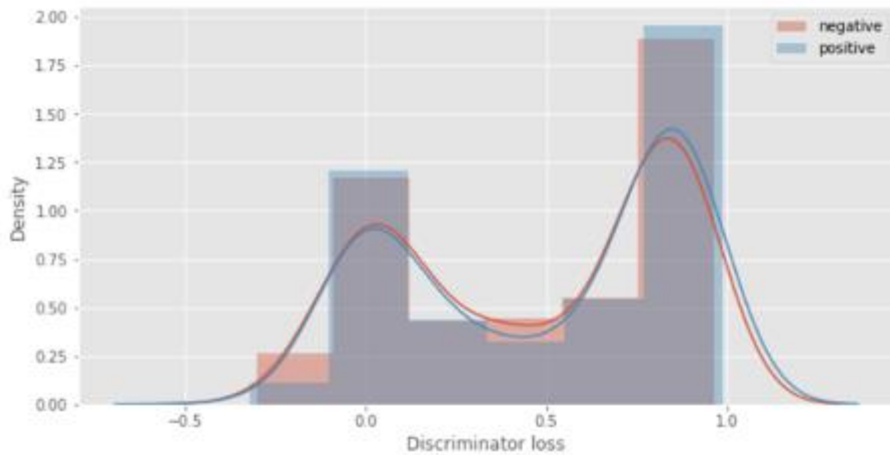
AlphaGan

Tensorboard



Alpha-GAN, we plot the histograms of the L1/L2 misfortunes and elevated level highlights misfortunes. For this situation, the recreation misfortunes show a distinction among negative and positive examples. This can be credited to alpha-GAN's capacity to remake picture saw previously.

The ROC bends show that while different highlights follow the askew standard intently, L1 and L2 misfortunes show noticeably deviation with the zone under the ROC bend of 0.65.



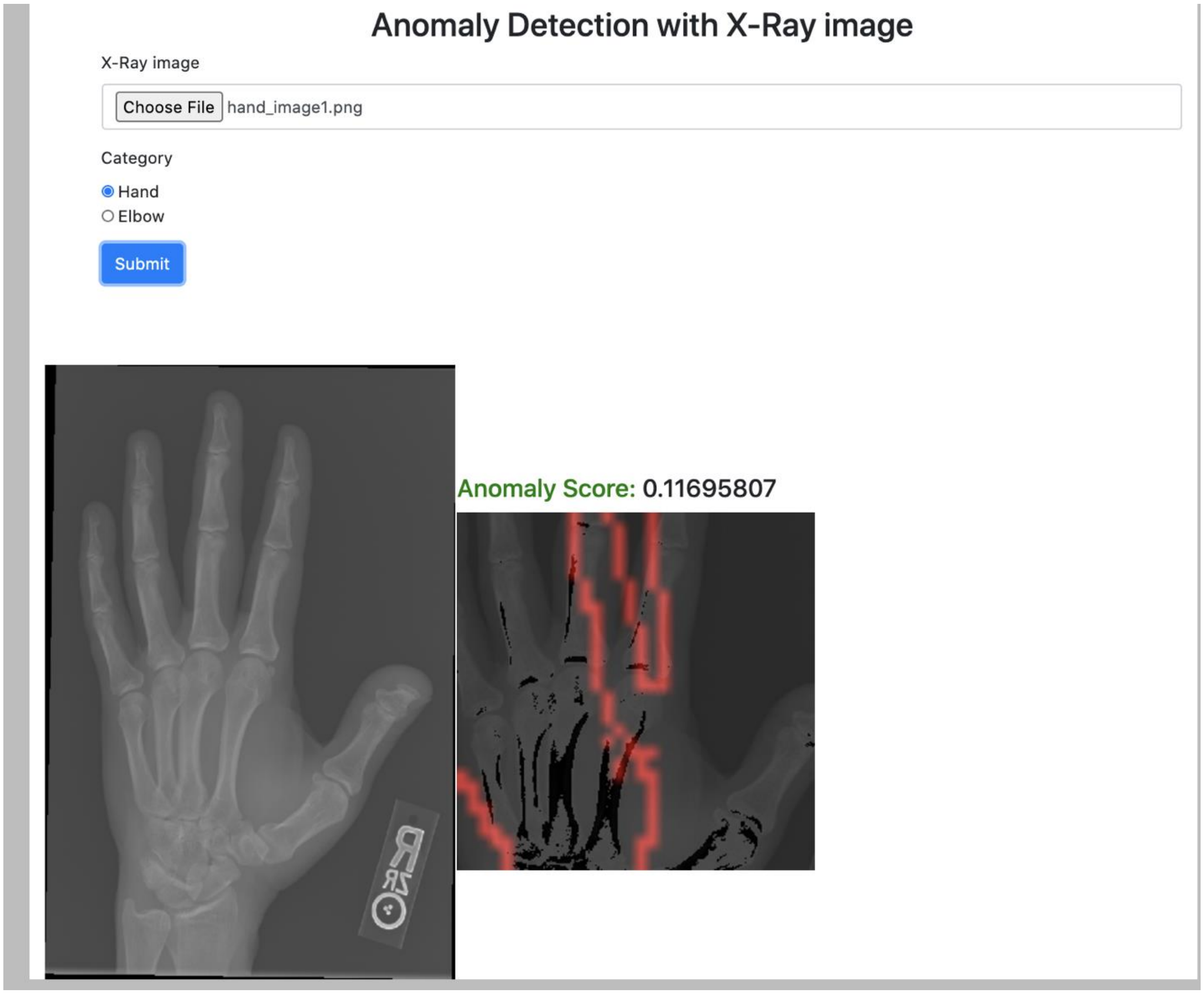
AnoGAN

Metrics for AnoGAN

No.	Category	Roc- AUC
1	ELBOW	0.88
2	FINGER	0.92
3	FOREARM	0.93
4	HAND	0.78
5	HUMERUS	0.91
6	SHOULDER	0.89
7	WRIST	0.97

Web-App

To put our project together we built a web-app. Below is a screenshot of the web-app



Here we load an x-ray image into our webapp and then the webapp attempts to detect any abnormalities in the x-ray.

Summary

Despite the fact that we cannot affirm the speculation of utilizing GAN's solo highlights to recognize abnormalities, it has given us a few bits of knowledge on the inner activity of GANs. This is as yet a fascinating methodology and we will see more research toward this path later on. We have investigated two GANs with Mura dataset; From the analysis, we have identified AnoGAN is performing better compared to AlphaGAN. Many researchers are studying this GAN specific anomaly detection recent times. This study would resolve problems in areas like Health care, fake detection and so on.