Tutorial on Chest X-Ray Imaging Diagnosis

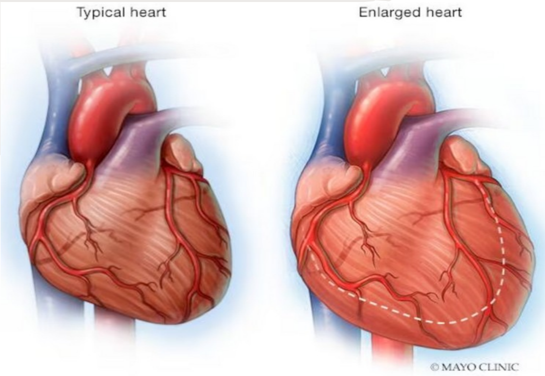
Introduction

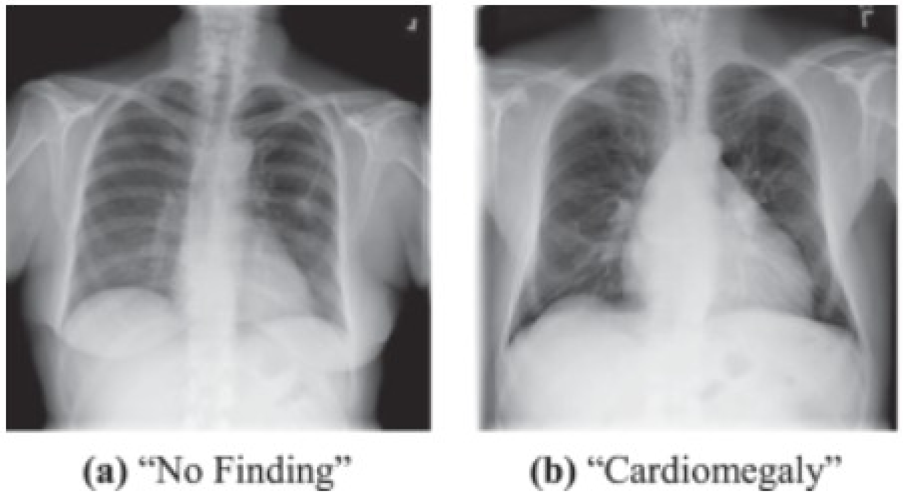
- In this tutorial, we will use ResNet to predict the disease label for Chest X-Ray images from NIH ChestX-Ray14 dataset and use GradCam to draw bounding boxes.
- The tutorial is divided into 3 parts.
  - Use an existing model (in this tutorial we use ResNet)
  - Finetune the model for a specific task (Binary classification: cardiomegaly)
  - Use GradCam for bounding box generation

Data

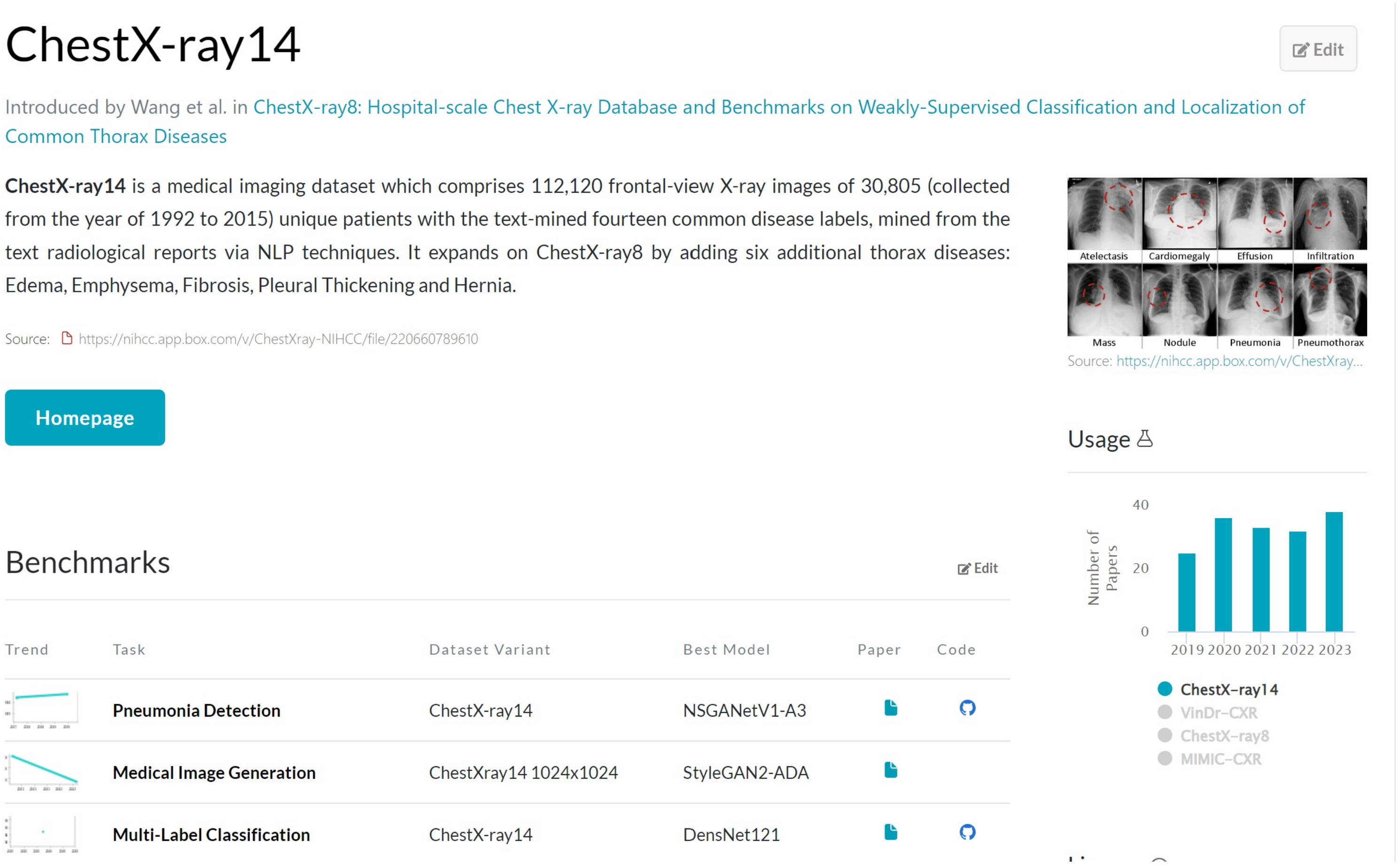
- NIH ChestX-ray14 dataset
    - https://www.kaggle.com/datasets/nih-chest-xrays/data
    - 112,120 frontal-view X-ray images of 30,805 unique patients
    - 14 thorax diseases

- Cardiomegaly and normal images is used for training and test
- 2,776 Cardiomegaly images in total
- 2,776 of 60,361 normal images is selected for data balance




(a) "No Finding"   (b) "Cardiomegaly"

Chest X-ray14



ChestX-ray14

Introduced by Wang et al. in ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases

ChestX-ray14 is a medical imaging dataset which comprises 112,120 frontal-view X-ray images of 30,805 (collected from the year of 1992 to 2015) unique patients with the text-mined fourteen common disease labels, mined from the text radiological reports via NLP techniques. It expands on ChestX-ray8 by adding six additional thorax diseases: Edema, Emphysema, Fibrosis, Pleural Thickening and Hernia.

Source: https://nihcc.app.box.com/v/ChestXray-NIHCC/file/220660789610

Source: https://nihcc.app.box.com/v/ChestXray...

Benchmarks

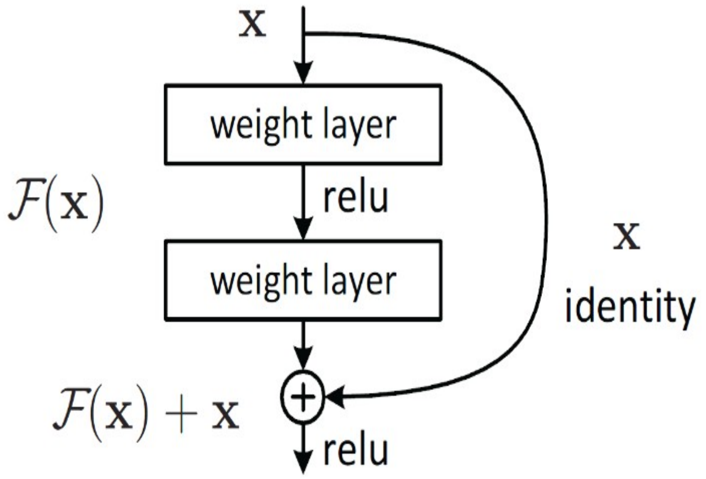
Trend	Task	Dataset Variant	Best Model	Paper	Code
	Pneumonia Detection	ChestX-ray14	NSGANetV1-A3		
	Medical Image Generation	ChestXray14 1024x1024	StyleGAN2-ADA		
	Multi-Label Classification	ChestX-ray14	DensNet121		

Usage

https://paperswithcode.com/dataset/chestx-ray14

Step1: Using an existing model (ResNet)

- ResNet, or Residual Networks, is a deep neural network architecture that was introduced to address the vanishing gradient problem observed in training of very deep convolutional neural networks.
- ResNet introduces skip connections, also known as residual connections (hence the name), that allow the output from one layer to bypass several intermediate layers and be added directly to the output of a later layer.



x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

https://paperswithcode.com/method/residual-block

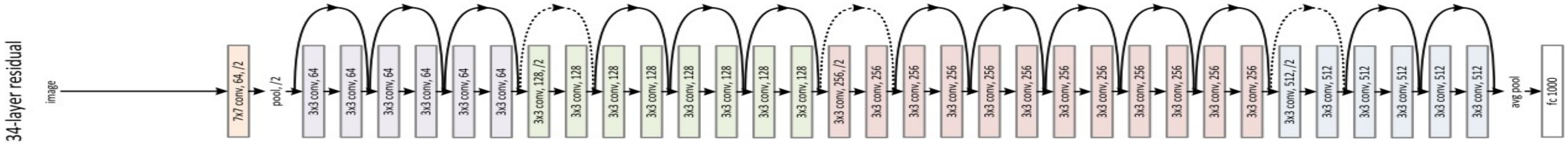


Image Source: https://betterprogramming.pub/a-deep-dive-into-residual-neural-networks-5d5b3ef60d21

ResNet Components

1. **Convolutional Layers (in red)**
   The first layer of the network is a convolutional layer that performs convolution on the input image. This is followed by batch normalization and a max-pooling layer that down samples the output of the convolutional layer.
2. **Residual Blocks (in orange)**
   Each residual block consists of two convolutional layers, each followed by a batch normalization layer and a rectified linear unit (ReLU) activation function.
3. **Fully Connected Layer (in green)**
   The final layer of the network is a fully connected layer that takes the output of the last residual block and maps it to the output classes. The number of neurons in the fully connected layer is equal to the number of output classes.

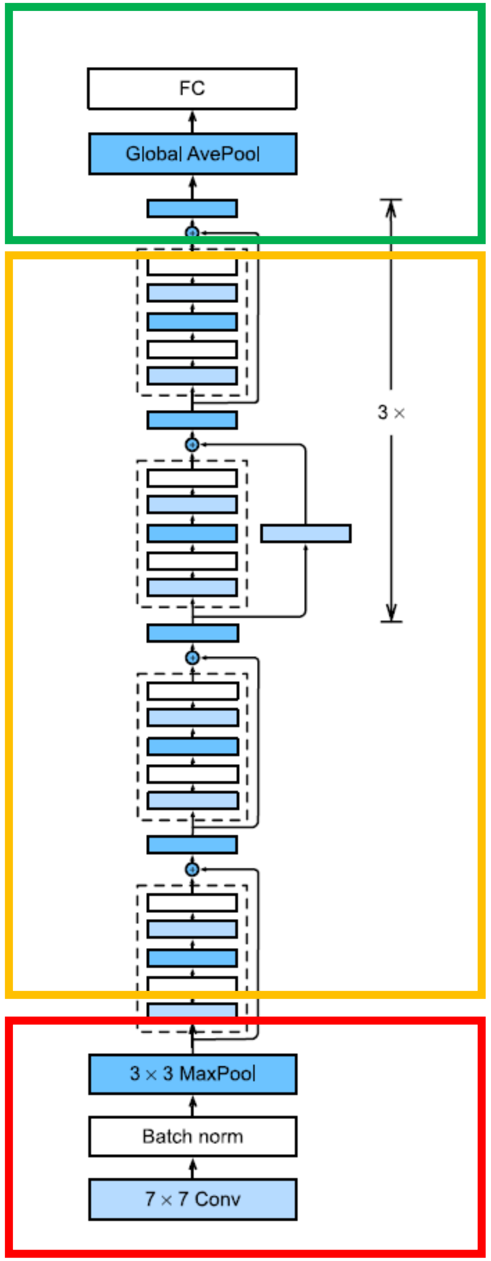
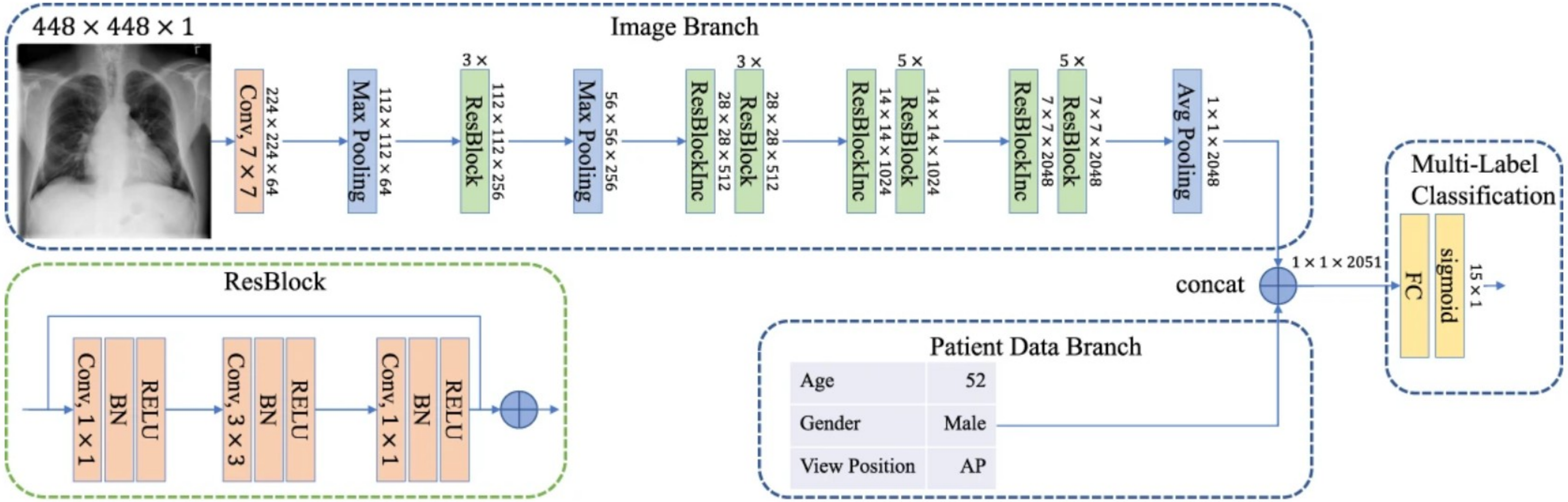


Image source: https://classic.d2l.ai/chapter_convolutional-modern/resnet.html#fig-resnet18

Step 2: Fine tuning ResNet50 for disease classification

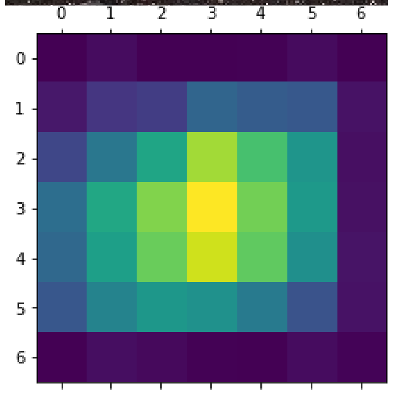


Age	52
Gender	Male
View Position	AP

https://www.nature.com/articles/s41598-019-42294-8

Step 3: GradCam for bounding box generation

- GradCam provides us with a way to investigate what parts of the image influenced the whole model’s decision.
- The intuition behind the algorithm is based upon the fact that the model must have seen some pixels (or regions of the image) and decided on what object is present in the image.
- By using the gradients from the last convolutional layer (as shown in second figure), Grad-CAM can highlight the regions of the image that most influenced the network's prediction.



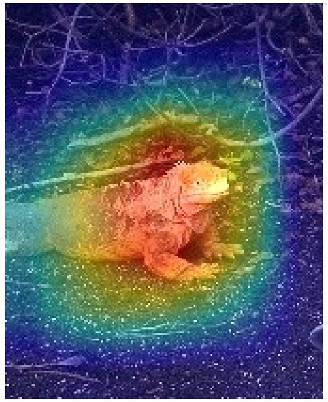
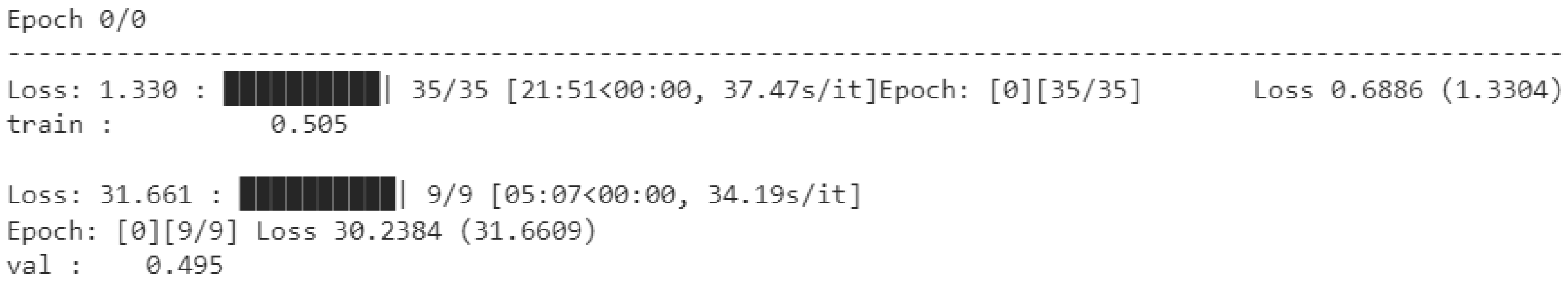


Image source: https://medium.com/@stepanulyanin/implementing-grad-cam-in-pytorch-ea0937c31e82

Evaluation metrics

- Since the model was trained on just one epoch, the overall accuracy of the model is not good.
- Training on higher epochs will result in better accuracy



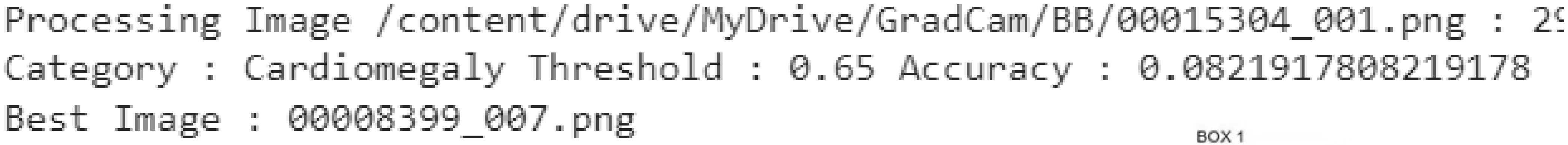
```
Epoch 0/0
----------------------------------------------------------------------------------------------------
Loss: 1.330 : ██████████| 35/35 [21:51<00:00, 37.47s/it]Epoch: [0][35/35]       Loss 0.6886 (1.3304)
train :          0.505

Loss: 31.661 : ██████████| 9/9 [05:07<00:00, 34.19s/it]
Epoch: [0][9/9] Loss 30.2384 (31.6609)
val :    0.495
```

- Here the training loss is 1.33 and training accuracy is 50.5%
- The evaluation loss is 31.661 and validation accuracy is 49.5%

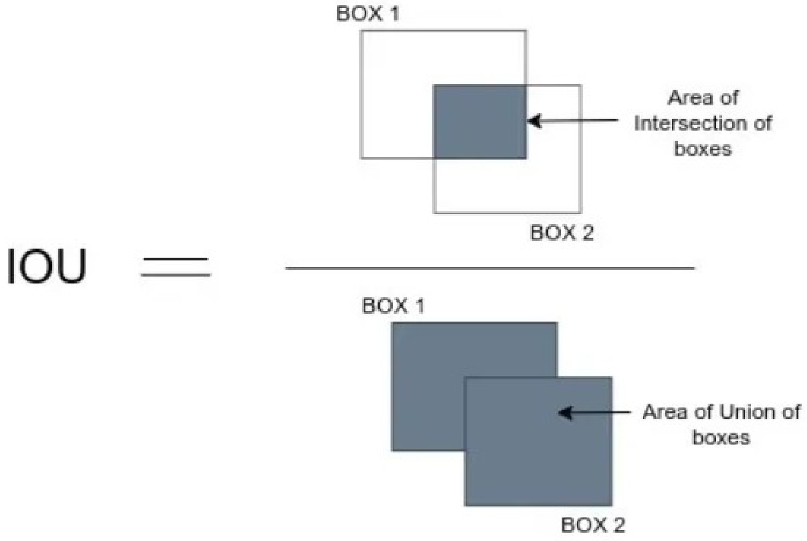
Evaluation Metrics

- The accuracy of Gradcam is based on Intersection Over Union (IOU). i.e how much overlapping is present between ground truth and model output



```
Processing Image /content/drive/MyDrive/GradCam/BB/00015304_001.png : 2
Category : Cardiomegaly Threshold : 0.65 Accuracy : 0.0821917808219178
Best Image : 00008399_007.png
```

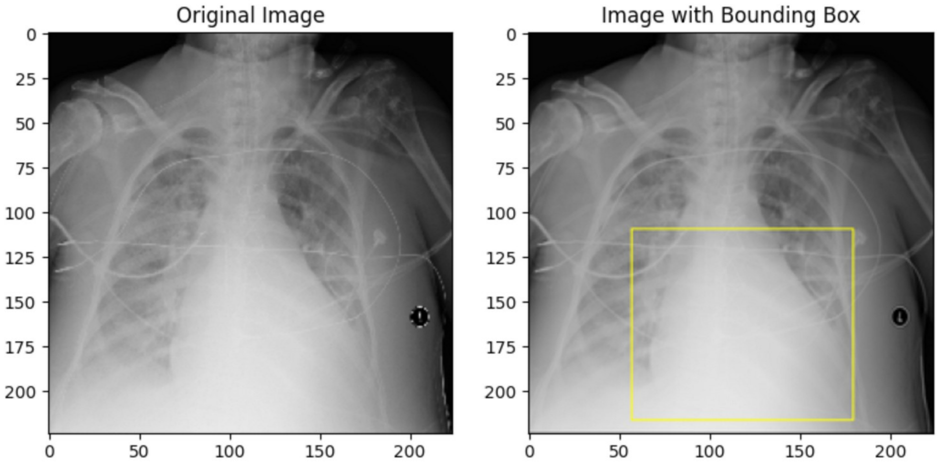
- Intersection Over Union (IoU)
  - Where one box is actual area, another is predicted area

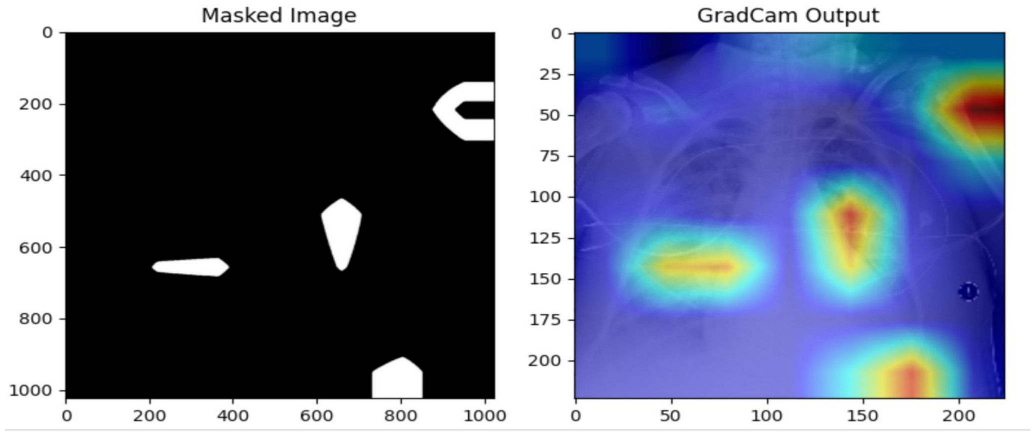


https://medium.com/analytics-vidhya/iou-intersection-over-union-705a39e7acef

Generated output of GradCam

- Once you open output folder, you will find 4 different images for one input image
    - gtimage_(name of original image).png: image with bounding box
    - mask_(name of original image).png: masked image
    - original__(name of original image).png: original image
    - pred_(name of original image).png: GradCam output





Generated output

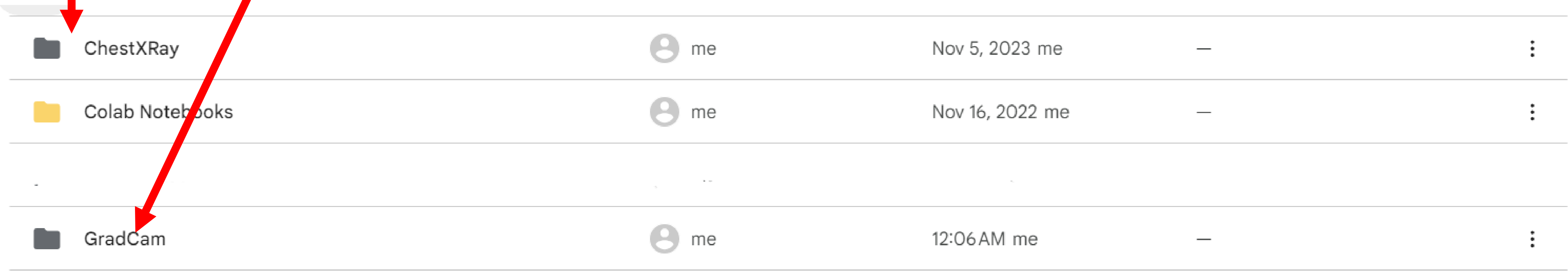
1. **Original Image:** This is the original input image without any modifications.
2. **gtimage:** This image is created by drawing a bounding box (in yellow) around the region of interest (ROI). The bounding box is typically obtained from the ground truth bounding box coordinates provided in the dataset.
3. **Mask Image:** This image is created by applying a mask to the original image. The mask is derived from the GradCAM or GradCAM++ output and represents regions of high significance for the predicted class. Only the regions with high values in the mask are retained in this image, while other areas are masked out.
4. **Pred Image:** This image visualizes the GradCAM or GradCAM++ output, highlighting the regions in the image that are most important for the model's prediction of the target class. This image typically shows regions of high activation in the neural network for the specific class.

Data storage

- The Google drive where the input and output data is stored looks like this

Contains a total of 5552 images used for fine tuning ResNet

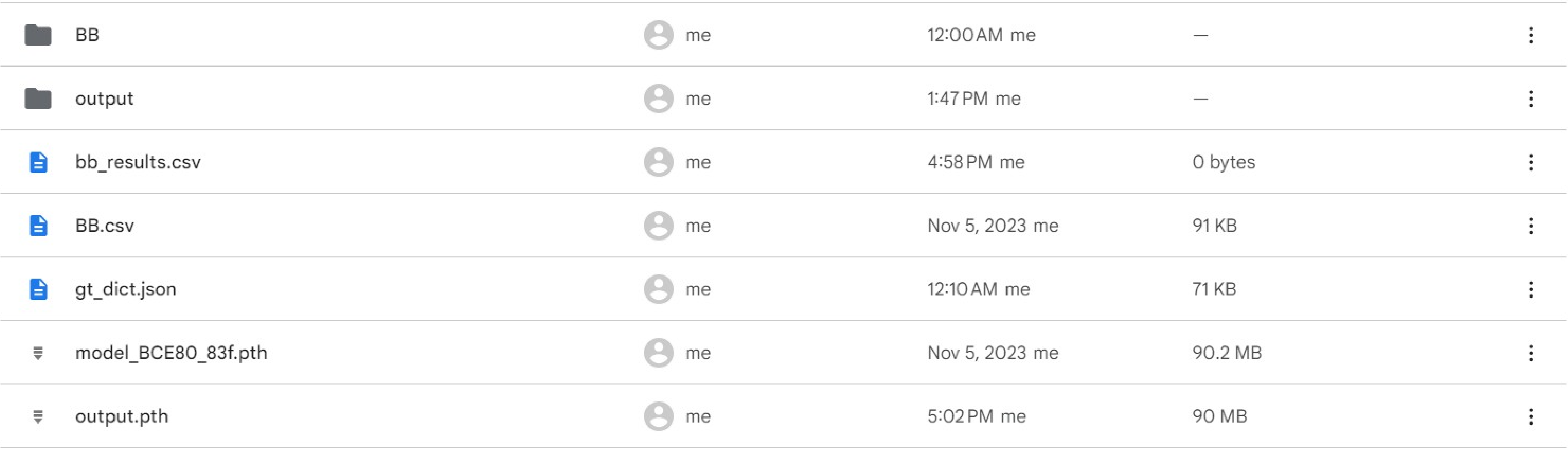
Contains all images that has Cardiomegaly and a json and BB.csv file that is used for generating bounding boxes



ChestXRay	me	Nov 5, 2023 me	—	⋮
Colab Notebooks	me	Nov 16, 2022 me	—	⋮
GradCam	me	12:06 AM me	—	⋮

Data Storage

- When you open GradCam folder, it looks like this
- BB folder contains all Cardiomelagy images (ground truth images)
- All generated files will be stored in output folder



BB	me	12:00 AM me	—
output	me	1:47 PM me	—
bb_results.csv	me	4:58 PM me	0 bytes
BB.csv	me	Nov 5, 2023 me	91 KB
gt_dict.json	me	12:10 AM me	71 KB
model_BCE80_83f.pth	me	Nov 5, 2023 me	90.2 MB
output.pth	me	5:02 PM me	90 MB

Model file

- Once finetuning is completed, a new model file named output.pth will be generated.
- Since this is obtained on just 1 epoch, the performance is not accurate. Hence use the ResNet model trained on 10 epochs named “Chest_cardi_001.pth” for GradCam which has a higher accuracy as it is trained on a larger number of epochs.
- bb_results.csv is a generated file that stores all the results such as location in the image which the model thinks is good for classification.