



دانشگاه آزاد اسلامی واحد تهران جنوب

پروژه درس بینایی ماشین

مقطع کارشناسی ارشد رشته مهندسی پزشکی گرایش بیوالکتریک

دکتر مهدی اسلامی

تقسیم‌بندی تصاویر پزشکی (داخل سینه)

نگارنده:

نیما ولدبیگی

خرداد ۱۴۰۳

۱- مجموعه داده

مدل ما بر روی مجموعه داده INbreast آموزش داده خواهد شد که بر اساس ۱۲۵ مورد کل شامل ۴۱۵ تصویر ماموگرافی است که تقریباً ۹۵ مورد از زنان مبتلا به هر دو سینه (چهار تصویر در هر مورد) و ۲۵ مورد از بیماران ماستکتومی (دو تصویر) است. مجموعه داده InBreast شامل تصاویر ماموگرافی به شکل تصاویر DICOM است که حاشیه نویسی سطح پیکسل هر تصویر مربوطه در یک فایل XML موجود است.

۲- استفاده از پردازنده گوگل

از آنجایی که قرار است از google colab برای آموزش مدل خود استفاده کنیم، نمی‌توانیم تمام داده‌های خود را ذخیره کنیم و پیش پردازش داده‌های خود را در محیط موقتی google colabs انجام دهیم زیرا پس از مدت زمان مشخصی پاک می‌شود. برای اینکه داده‌هایمان پایدار بماند و از درایو گوگل به‌عنوان سیستم فایل دائمی خود استفاده کنیم، زیرا فضای ذخیره‌سازی و ساختار دایرکتوری مورد نیاز برای سازماندهی پروژه را در اختیار ما قرار می‌دهد. ما قصد داریم از خطوط کد زیر برای اتصال درایو گوگل خود با این نوت بوک colab استفاده کنیم.

```
from google.colab import drive
drive.mount('/content/drive')
```

همچنین به دلیل اینکه حجم داده‌ها به نسبت زیاد است از خود فضای ذخیره سازی Kaggle استفاده خواهیم کرد.

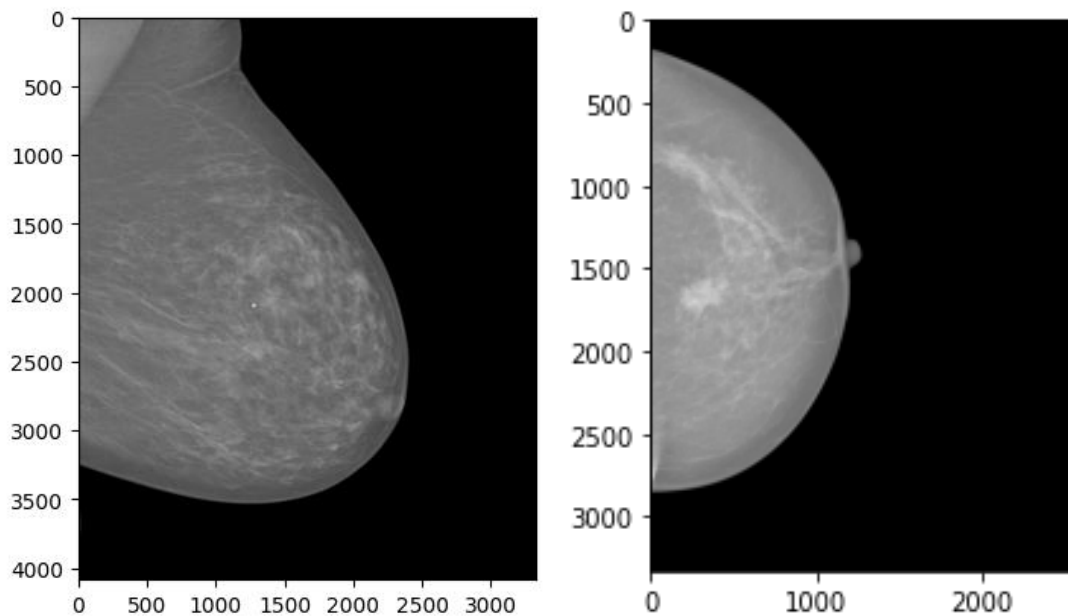
```
import opendatasets
opendatasets.download("https://www.kaggle.com/datasets/ramanathansp20/inbreast-dataset")
```

سپس کتاب خانه های مورد نیاز که در گوگل کولب موجود نیستن را نصب خواهیم کرد.

۳- تبدیل تصاویر DICOM به PNG

مجموعه داده Inbreast تصاویر ماموگرافی را در قالب استاندارد تصویربرداری دیجیتال و ارتباطات در پزشکی (DICOM) همراه با حاشیه نویسی سطح پیکسل در قالب XML ارائه می‌دهد.

تصاویر DICOM بسیار سنگین هستند و آموزش آن‌ها دشوار است، بنابراین ابتدا باید تصاویر dicom را به فرمت png تبدیل کنیم تا بتوانیم مدل خود را به راحتی بر روی آن تصاویر png پردازش و آموزش دهیم. بعد از تبدیل تصاویر dicom به png آن‌ها را در پوشه دیگری با دایرکتوری داده داخلی به نام ALL PNGS ذخیره می‌کنیم. در شکل زیر نمونه‌ای از تصاویر سینه نمایش داده شده است.



شکل ۱ نمونه تصویر از مجموعه داده

حالا از کد زیر استفاده می‌کنیم تا ببینیم چند بعد/اندازه عکس داریم. این اطلاعات بعداً برای ما مفید خواهد بود.

```
file_sizes[]
```

```
for filename in os.listdir('/content/drive/MyDrive/BCD/Inbreast_Dataset/AllPNG/');
```

```
    if filename.endswith(".png"):
```

```
        img = cv2.imread('/content/drive/MyDrive/BCD/Inbreast_Dataset/AllPNG/'+filename)
```

```
        file_sizes.append(img.shape)
```

```
dim = set(file_sizes)
```

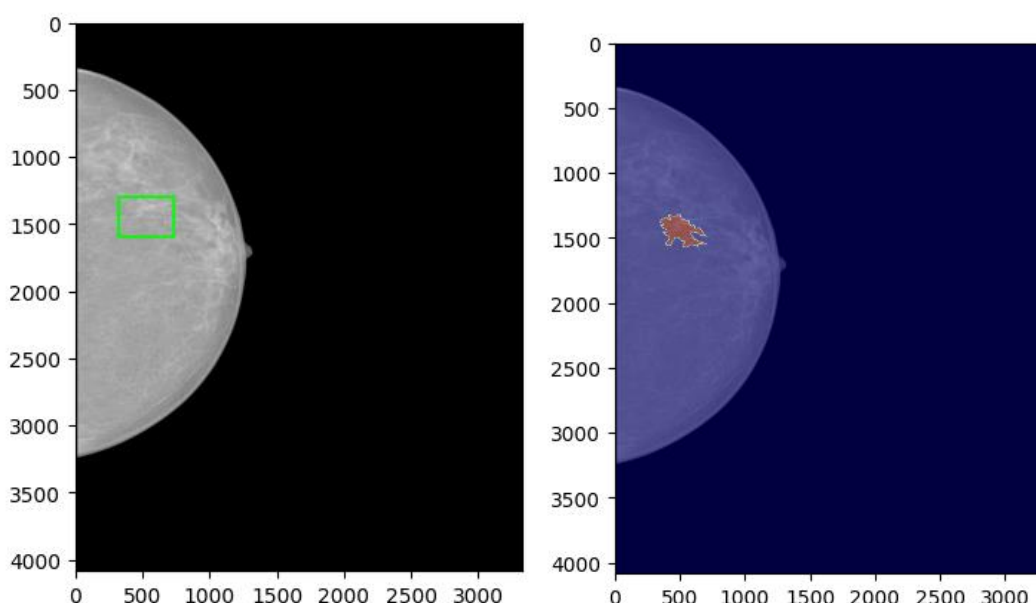
```
print(dim)
```

از آنجایی که اکنون تصاویر خود را آماده کرده‌ایم، باید راهی برای استخراج ناحیه انبوه علائق (ROI)^۲ پیدا کنیم، زیرا می‌خواهیم که مختصات جعبه محدودکننده ROI را در حین آموزش ارائه کنیم، فقط باید تابعی بنویسید که حاشیه نویسی سطح پیکسل ارائه شده با مجموعه داده Inbreast را در قالب XML استخراج می‌کند و یک ماسک باینری برای هر ROI جرم موجود در یک ماموگرافی خاص برمی‌گرداند.

ما از تابع زیر برای استخراج مقادیر پیکسل ROI و ایجاد یک ماسک باینری برای هر یک از جرم‌های موجود استفاده می‌کنیم.

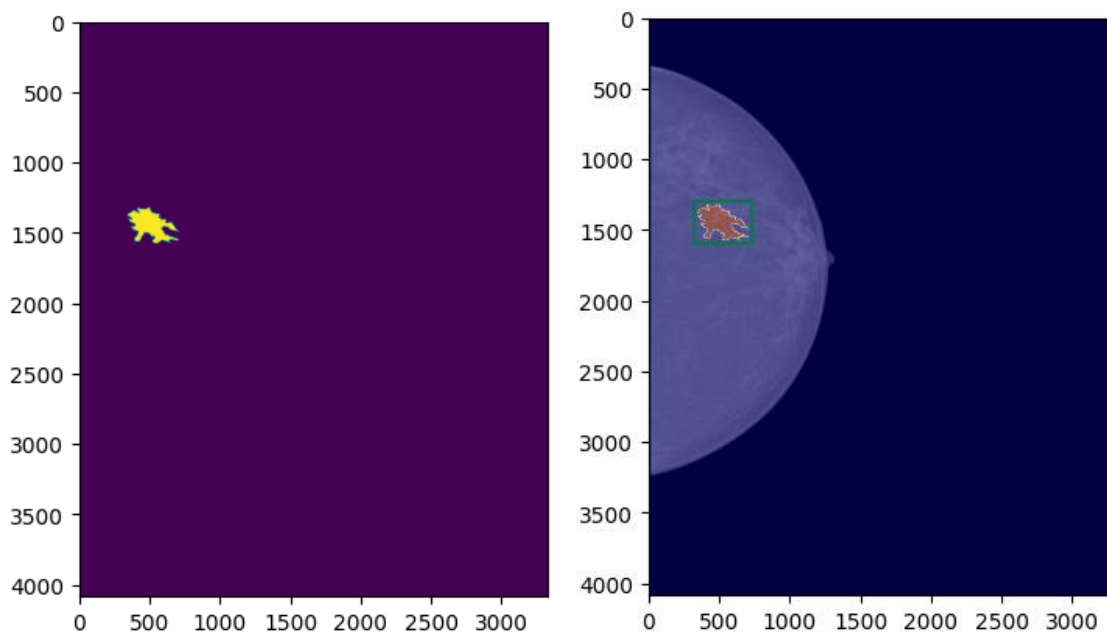
ما از plistlib برای دسترسی به فایل‌های حاشیه نویسی XML و عبور از آن استفاده می‌کنیم و از numpy برای تولید یک ماسک باینری استفاده می‌کنیم.

نمونه‌ای از تصویر ماسک مربوطه در تصویر زیر نمایش داده شده است.



شکل ۲ نمونه تصویر ماسک از مجموعه داده.

^۲Region Of Interests (ROI's)



شکل ۳ نمونه ای از ماسک به همراه جعبه محدود کننده.

۴- پیش پردازش

مراحل انجام شده در این بخش از نوت بوک اجباری نیست. با این حال، آنها برای تولید نتایج رضایت بخش ضروری هستند.

کد زیر برای استخراج پروفایل پستان از هر تصویر ماموگرافی در پوشه AllPNG استفاده می شود. از پروفایلها برای استخراج ماسک های تصویر در فرمت png استفاده می شود.

```
def padimages(image,file_name, ratio):

    [length, width] = np.shape(image)
    if length/width>ratio:#1024/800
        print("This image needs padding.")
        add_wid = round(length*(1/ratio)-width)
        pad = np.zeros((length,add_wid))
        pad = pad.astype(image.dtype)
        if '_R_' in file_name:
            # pad on the left
            pad_image = np.concatenate((pad,image),axis=1)
```

```

        else:

            pad_image = np.concatenate((image,pad),axis=1)

        return pad_image

#from skimage import data, color, io, img_as_float

image_path = "/content/inbreast-dataset/INbreast Release 1.0/AllPNG/"

annotation_path = '/content/inbreast-dataset/INbreast Release 1.0/AllXML/'

save_image_path = "/preprocessed_image/"

if not os.path.exists(save_image_path):

    os.mkdir(save_image_path)

save_mask_path = "/preprocessed_mask/"

if not os.path.exists(save_mask_path):

    os.mkdir(save_mask_path)

file_names = os.listdir(image_path)

file_names = sorted(file_names)

for i in range(0,len(file_names)):

    print(file_names[i])

    mammo = io.imread(image_path+file_names[i],0)

    xml_file_path = annotation_path + file_names[i].replace('png', 'xml')

    if not os.path.exists(xml_file_path):

        print(f"XML file not found for {file_names[i]}. Skipping this image.")

        continue

    lesion_mask = load_inbreast_mask(xml_file_path, mammo.shape[0:2])

    if np.max(lesion_mask)>=0:

#   Extract the breast profile and crop the mammogram, breast mask and the lesion mask

#   Normalize the image into 16-bit

        breast_preprocess = Preprocess.extract_breast_profile(mammo,lesion_mask,1)

        mammo = breast_preprocess.image

        breast_mask = breast_preprocess.mask

        lesion_mask =breast_preprocess.lesion_mask

```

```

print ('Number of lesions: '+str(np.max(np.unique(label(lesion_mask)))))

# pad the image, to ensure the aspect ratio is 1:1
pad_mammo = padimages(mammo,file_names[i],1)

# save the preprocessed image
io.imsave(save_image_path + file_names[i],pad_mammo)

# if the image has more than 1 lesion, then seperate them into different masks and number them.

labelim = label(lesion_mask)

if np.max(labelim)>0:

# if there is at least 1 lesion.

    for l in range(1,np.max(labelim+1)):

        l_mask = np.zeros(np.shape(labelim))

        l_mask = l_mask.astype(lesion_mask.dtype)

        l_mask [labelim==l] = 255

        num_nonzero = np.where(l_mask>0)

        num_nonzero = len(num_nonzero[0])

        if num_nonzero>15:

            print('A valid mask')

#         Pad the mask in the same way as padding the image

        pad_l_mask = padimages(l_mask,file_names[i],1)

        io.imsave(save_mask_path+file_names[i][:4]+str(l)+'png',pad_l_mask)

    else:

        print('Has a tiny piece of noise that is not valid for training!')

else:# if there is no lesion

    pad_lesion_mask = padimages(lesion_mask,file_names[i],1)

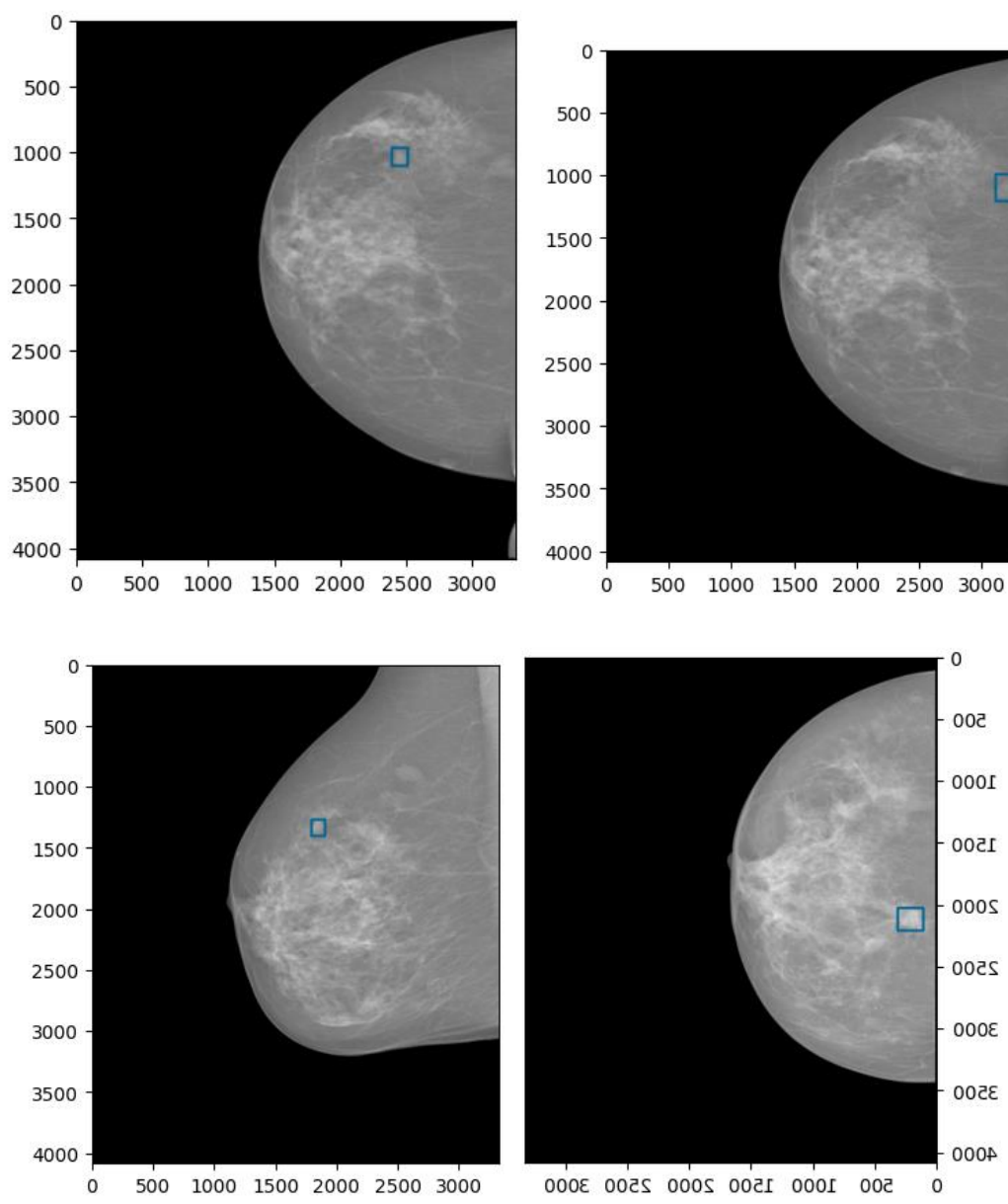
    io.imsave(save_mask_path+file_names[i][:4]+str(0)+'png',pad_lesion_mask)

stop = timeit.default_timer()

print('RunTime per image: ', (stop - start)/ len(file_names))

```

ناحیه سیاه اضافی بریده شده و تصاویر ماموگرافی به ۱۶ بیت نرمال می شوند. سپس تصاویر به گونه‌ای قرار می‌گیرند که اسکن‌های تمام سینه‌ها را به سمت چپ تصویر منتقل می‌کنند در حالی که به طور همزمان تصاویر مربعی ایجاد می‌شوند.



شکل ۴۵ نمونه‌ای از تصاویر به همراه جعبه محدود کننده.

این یک کار محاسباتی سنگین است، به همین دلیل ماموگرافی‌ها تا ۲۵ درصد اندازه اصلی خود نمونه برداری شده‌اند. نسبت تصویر حفظ می‌شود و ماسک‌های ماموگرافی به طور مناسب مقیاس بندی می‌شوند.