

## گزارش پروژه پایانی یادگیری عمیق

غزل بخشنده ۹۸۵۲۲۱۵۷

افشین زنگنه ۹۸۵۲۱۲۴۳

### لینک نوتبوک پروژه

<https://colab.research.google.com/drive/1aq1TVFg0mC6U5emJNXchzIxsPd04dorX?usp=sharing>

\* در این پروژه در بخش‌هایی از کد اولیه نوشته شده توسط نویسندگان مقاله شماره ۱ استفاده شده است:

<https://github.com/shervinmin/DeepCovid>

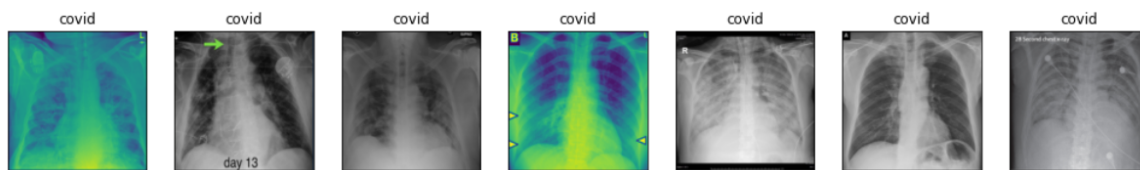
### خواندن داده‌ها

برای خواندن داده‌ها، ابتدا دادگان را روی گوگل درایو آپلود کردیم و سپس آن را روی کولب mount و داده‌ها را استخراج کردیم (بخش load data در نوت‌بوک). همانطور که مشاهده می‌شود در مجموع ۵۱۸۴ داده داریم که ۲۰۸۴ تا برای آموزش و ۳۱۰۰ تا برای تست مدل می‌باشد و همچنین ۸۴ تا از داده‌های آموزشی و ۱۰۰ تا از داده‌های تست، متعلق به کلاس دارای بیماری کووید هستند.

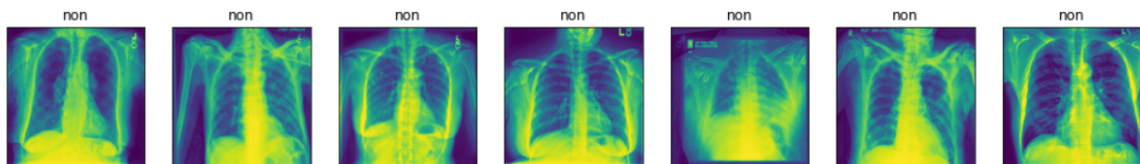
### داده افزایی

برای داده افزایی از transforms در pytorch استفاده می‌کنیم. ابتدا با استفاده از transforms داده‌ها را resize می‌کنیم. سپس ۶ نوع تبدیل دیگر روی دیتای بخش covid انجام داده و ۲ نوع تبدیل هم روی داده‌های بخش non covid انجام می‌دهیم. دلیل اینکه روی دیتای non covid تبدیل‌های کمتری انجام شده تعداد بیشتر آن‌ها نسبت به داده‌های covid است. در واقع با این کار تعداد آن‌ها نسبت به هم متعادل‌تر می‌شود.

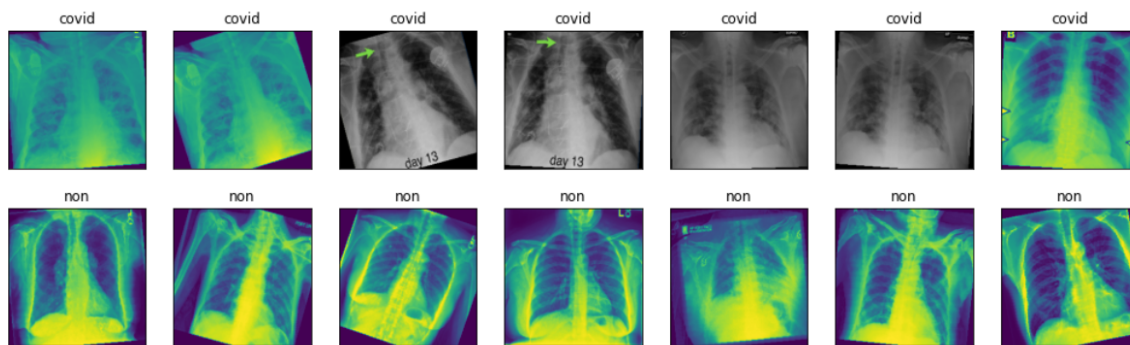
بخشی از دیتای covid بعد از resize:



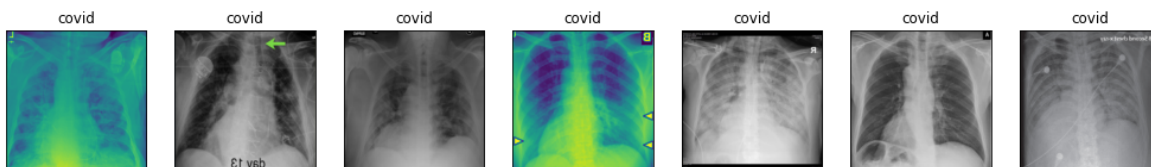
بخشی از دیتای non covid بعد از resize:



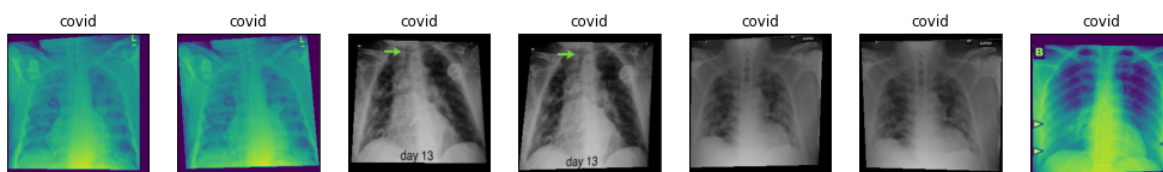
اعمال تبدیل اول:



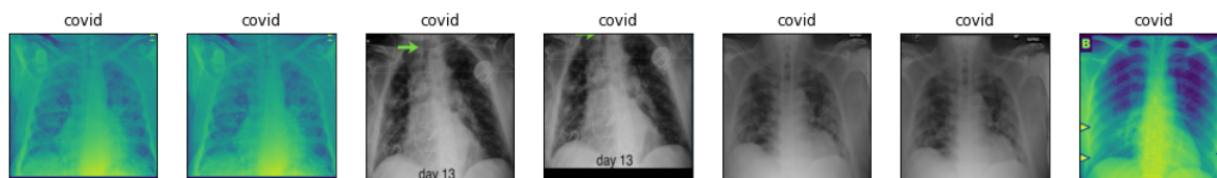
اعمال تبدیل دوم:



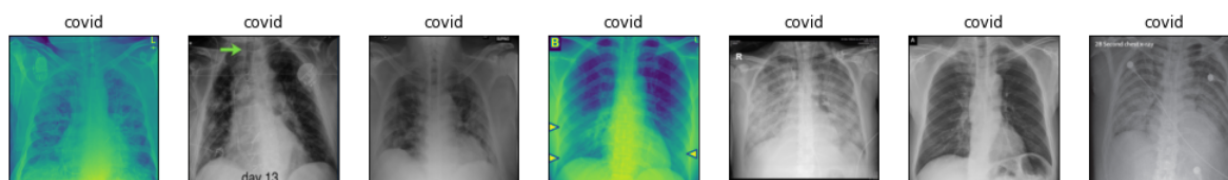
اعمال تبدیل سوم:



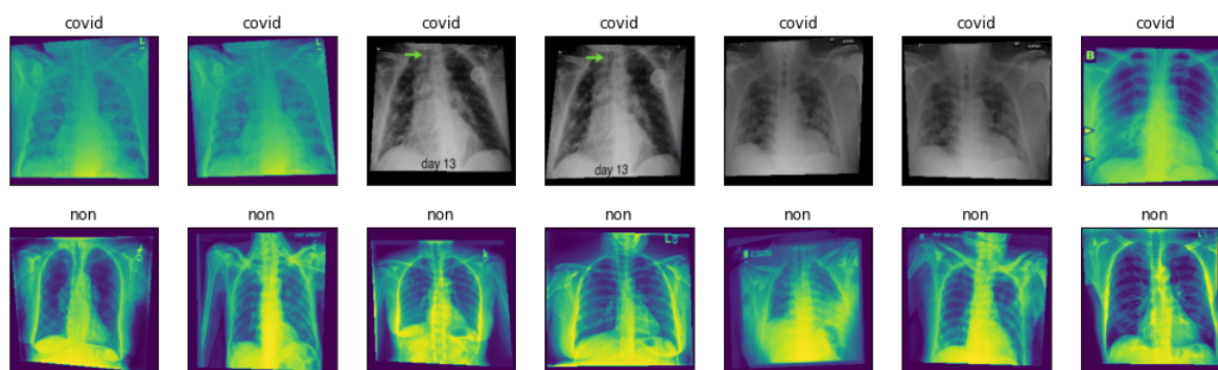
### اعمال تبدیل چهارم:



### اعمال تبدیل پنجم:



### اعمال تبدیل ششم:



## ساختن و آموزش مدل

در این بخش از مدل پیش‌آموزته‌ی SqueezeNet که روی مسئله ۱۰۰۰ کلاسه ImageNet آموزش داده شده، استفاده می‌کنیم و تنها لایه classifier آن را برای ۲ کلاس تغییر می‌دهیم و پارامترهای سایر بخش‌های مدل را فریز می‌کنیم.

```

model_conv = torchvision.models.squeezenet1_0(pretrained=True)
for param in model_conv.parameters():
    param.requires_grad = False

# change classifier number of classes to from 1000 to 2
model_conv.classifier[1] = nn.Conv2d(512, 2, kernel_size=(1,1))

model_conv = model_conv.to(device)
criterion = nn.CrossEntropyLoss()

# Observe that only parameters of final layer are being optimized as
# opposed to before.
optimizer_conv = optim.SGD(model_conv.classifier.parameters(), lr= 0.001, momentum= 0.9)

# Decay LR by a factor of 0.1 every 7 epochs
exp_lr_scheduler = lr_scheduler.StepLR(optimizer_conv, step_size=7, gamma=0.1)

if __name__ == "__main__":
    model_conv, train_acc, valid_acc = train_model(model_conv, criterion, optimizer_conv,
                                                    exp_lr_scheduler, batch_size, num_epochs=num_epochs)
    model_conv.eval()
    torch.save(model_conv, './squeezenet.pt')

```

همانطور که در کد نیز مشاهده می‌شود، از روش کاهش نرخ یادگیری به مرور زمان نیز در آموزش مدل استفاده کرده‌ایم و به تعداد ۳۰ epoch و با batch\_size ۶۴ مدل را آموزش می‌دهیم. در نهایت مدل به دست آمده را ذخیره می‌کنیم (بخش build and train model در نوت‌بوک).

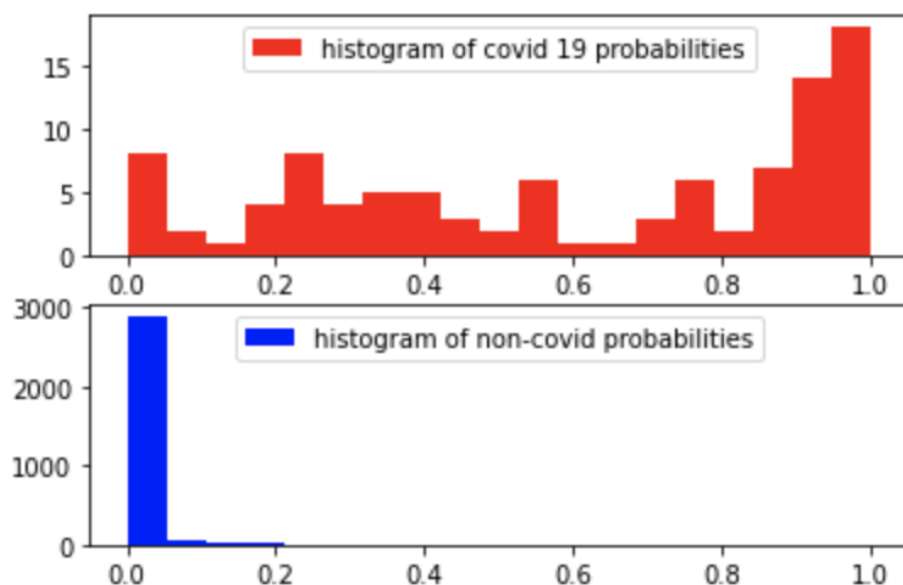
## ارزیابی مدل

ابتدا با استفاده از قطعه کد زیر، احتمال‌های به دست آمده برای دادگان را به ازای دو کلاس دارای کووید و بدون کووید محاسبه می‌کنیم (بخش evaluate model در نوت‌بوک):

```

bins = np.linspace(0, 1, 20)
plt.subplot(2, 1, 1)
plt.hist(positive_covid_probabilities, bins, color= 'red', histtype = 'bar', label='histogram of covid 19 probabilities')
plt.legend(loc='upper center')
plt.subplot(2, 1, 2)
plt.hist(negative_covid_probabilities, bins, color= 'blue', label='histogram of non-covid probabilities')
plt.legend(loc='upper center')

```



سپس با استفاده از مقادیر مختلف آستانه، مقادیر sensitivity و specificity را به دست می‌آوریم که این اطلاعات در جدول زیر آمده است:

threshold	sensitivity	specificity
0.05	0.92	0.964
0.1	0.91	0.982
0.2	0.87	0.993
0.3	0.74	0.996
0.4	0.65	0.997
0.5	0.6	0.9976

\* با توجه به مقاله، تعریف به صورت زیر است:

$$\text{Sensitivity} = \frac{\text{\#Images correctly predicted as COVID-19}}{\text{\#Total COVID-19 Images}},$$

$$\text{Specificity} = \frac{\text{\#Images correctly predicted as Non-COVID}}{\text{\#Total Non-COVID Images}}.$$