



## هدف پروژه

در این پروژه از شما می‌خواهیم تا با استفاده از دیتاست مورد نظر یک شبکه عصبی پیچشی (CNN) آموزش دهید و سپس با استفاده از آن، مدل خود را با استفاده از Transfer learning و FineTuning بر روی دیتاست‌های دیگری که اشاره خواهد شد، ارزیابی کنید و از مدل خود برای کاربردهای متفاوت استفاده کنید.

نکات قابل توجه:

- حتما داکيومنت داشته باشید.
- از هر دو فریمورک tensorflow(keras) و torch می‌توانید استفاده کنید.
- در تمام پروژه به فاز preprocess توجه داشته باشید.
- در تمام بخش‌های پروژه hyperparameter tuning انجام دهید.
- برای تمام مدل‌ها و آموزش‌های جدید در تمام فازها، نمودارهای یادگیری و ماتریس گمراهی و ROC\_Curve را رسم کنید و آن‌ها را در داکيومنت خود تحلیل کنید.
- برای تمام مدل‌ها در تمام فازها، برای دیتای تست Precision و Recall و F1\_score را محاسبه کنید و آن‌ها را در داکيومنت خود بررسی کنید.
- در تمام فازهایی که مدلی آموزش داده می‌شود؛ تعدادی از تصاویر تست را به همراه لیبل واقعی و لیبل پیش‌بینی شده آن‌ها visualize کنید و مدل خود را بررسی کنید.

## دیتاست Animals10

این دیتاست شامل 10 کلاس از 10 نوع حیوان متفاوت بوده. 70 درصد دیتاست را برای train، 10 درصد برای validation و 20 درصد را برای تست در نظر بگیرید.

## فاز اول: Model Designing

در این فاز باید یک شبکه عصبی طراحی کنید و بر روی دیتاست Animals10 آموزش دهید. توجه کنید یک شبکه خیلی بزرگ طراحی نکنید و حداکثر 1,400,000 (یک میلیون و 400 هزار) پارامتر داشته باشد.

- سعی کنید با توجه به تعداد پارامترها و تعداد دیتا به بهترین دقت برسید. (هم داده‌های آموزشی و هم داده‌های تست)

- در این فاز نیاز نیست از overfit شدن مدل‌تان جلوگیری کنید. صرفاً یک شبکه عصبی ایجاد کنید و آن را آموزش دهید.

## فاز دوم: Parameter Limitation

در این فاز باید سعی کنید تا بدون تغییر ساختار کلی مدل خود، یعنی لایه‌ها و بلاک‌های Conv، پارامترهای شبکه خود را به حداکثر 400,000 برسانید و مجدداً این مدل جدید را بر روی دیتاست Animals10 آموزش دهید. مجدداً در این فاز هم هیچ تلاشی برای جلوگیری از overfit نکنید. و در نهایت مدل خود را تست کنید.

- سعی کنید با توجه به تعداد پارامترها و تعداد دیتا به بهترین دقت برسید.

- تعدادی از تصاویر تست را به همراه لیبل واقعی و لیبل پیش‌بینی شده آن‌ها visualize کنید و مدل خود را بررسی کنید.

## فاز سوم: Overfitting Prevent

مدل‌های شما در دو فاز قبل به احتمال خیلی زیاد overfit می‌شود. در این فاز باید از تمام روش‌ها و متودهایی که برای جلوگیری از overfit یاد گرفته‌اید استفاده کنید (تعدادی روش جدید هم بررسی کنید و در کد خود استفاده کنید).

برای DataAugmentation، چندین روش در کنار هم امتحان کنید و فقط به یک متود بسنده نکنید. سعی کنید چندین روش و متود دیگر برای جلوگیری از overfit شدن استفاده کنید تا مدل شما به بهترین دقت برسد. و در نهایت مدل خود را تست کنید.

## در این فاز باید به دقت بالایی دست یابید.

- توجه کنید که این فاز را باید برای هر دو مدل در فاز اول و دوم انجام دهید. (می‌توانید از متودهای یکسانی استفاده کنید ولی حتماً هر دو مدل را مجدد آموزش دهید)

- برای این مدل حتما hyperparameter tuning انجام دهید و برای پارامترهایی مثل optimizer، چندین optimizer را تست کنید و نتایج را در داکيومنت خود بررسی کنید. همچنین از پاک کردن کدهای مربوط به آموزش مدل با پارامترهای مختلف خودداری کنید. (تمام آزمایشات در کد باشد و نتایج هم در داکيومنت بررسی شده باشد)

- بهترین مدل را حتما برای فازهای بعد ذخیره کنید.

### فاز چهارم: Transfer Learning

در این فاز از شما میخواهیم تا از بهترین مدلی که در فاز قبل آموزش داده‌اید برای آموزش مجدد بر روی دیتاست جدیدی ولی با دیتای کمتر استفاده کنید.

#### • دیتاست یک Dog Sentiment Classification

- این دیتاست شامل 10 کلاس از حالت‌ها و احساسات مختلف سگ‌ها است. همچنین شامل داده‌های بسیار کمتری نسبت به دیتاست اصلی است.

1.1 همانطور که با فرایند Transfer Learning آشنا هستید، ابتدا تمام لایه‌های مدل را freeze کنید و فقط لایه آخر را که عوض می‌کنید؛ آموزش دهید. تمام موارد خواسته شده در ابتدای پروژه را برای این مدل و مدل‌های بعدی هم انجام دهید.

1.2 پس از آن مدل اولیه در این فاز را مجدد لود کرده و این بار یک بلاک آن را از freeze در بیاورید و مدل را آموزش دهید.

\* از overfit شدن مدل‌ها جلوگیری کنید.

\* تمام نتایج را در داکيومنت خود بررسی کنید. دلایل بالا یا پایین بودن دقت خود را بررسی کنید.

## • دیتاست دو Dog, Cat, Horse Classification

- این دیتاست از 3 کلاس سگ و گربه و اسب تشکیل شده است. این دیتاست نیز شامل داده‌های کمتری از دیتاست اصلی می‌باشد.

1.1 برای این دیتاست فقط لایه آخر مدل خود را بر اساس کلاس‌های جدید تغییر دهید و مدل را مجدد آموزش دهید. تمام موارد خواسته شده در ابتدای پروژه را برای این مدل و مدل‌های بعدی هم انجام دهید.  
\* حتما از overfit شدن مدل خود جلوگیری کنید.

1.2 اکنون می‌خواهیم از مدل‌های معروف pretrained استفاده کنیم و آن‌ها را بر روی دیتاست خود FineTune کنیم. در \* تمام مدل‌ها، hyperparameter tuning انجام دهید و به یک مقدار برای پارامترها بسنده نکنید.

1.3 در این بخش می‌خواهیم از سه مدل ResNet50 و MobileNetV2 و EfficientNetB0 استفاده کنیم. برای این سه مدل لایه آخر را عوض کنید ولی تعداد لایه‌های قابل آموزش، دست خودتان است (از 10 لایه بیشتر نشود). در نهایت هم مدل‌ها را تست و موارد خواسته شده در ابتدای پروژه را انجام دهید. از overfit شدن مدل‌ها جلوگیری کنید و به بهترین دقت برسید.

## نکات تکمیلی

- حتما موارد ذکر شده در اول صورت پروژه را رعایت کنید. در غیر این صورت از شما نمره کسر خواهد شد.
- علاوه بر سورس کد پروژه، فایل مستندات نیز باید آپلود شود.
- نام اعضای گروه در فایل مستندات ذکر شود و فقط یکی از اعضا پروژه را آپلود کند.
- هر گونه شباهت نامتعارف بین کد شما و کد سایر گروه ها تقلب محسوب می شود و نمره ای برای این پروژه دریافت نخواهید کرد.
- در صورت نوشتن داکيومنت تمیز (برای مثال با LATEX) نمره اضافه برای شما در نظر گرفته خواهد شد.
- فایل شامل سورس کد پروژه و مستندات را در قالب فایل zip و با نام شماره دانشجویی خود ذخیره و ارسال نمایید.
- در صورت داشتن هرگونه سوال می توانید با [Yosef\\_Rezazadeh81](#) و [Kourosh\\_Hsz](#) و [ipedram\\_vr](#) در ارتباط باشید یا در گروه درسی مطرح کنید.

موفق باشید؛  
تیم حل تمرین