



## هدف پروژه

در این پروژه از شما می خواهیم 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 دیتاست های 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 PyTorch می‌توانید استفاده کنید.
- در تمام پروژه به فاز Preprocess توجه داشته باشید.
- در تمام بخش‌های پروژه Tuning Hyperparameter انجام دهید.
- برای تمام مدل‌ها و آموزش‌های جدید در تمام فازها، نمودارهای یادگیری، Matrix Confusion و منحنی Curve ROC را رسم و در داکيومنت تحلیل کنید.
- برای تمام مدل‌ها در تمام فازها، Recall Precision و F1-Score را برای داده‌های تست محاسبه و در داکيومنت بررسی کنید.
- در تمام فازهای آموزش مدل، تعدادی از تصاویر تست را به همراه لیبل واقعی و پیش‌بینی‌شده مصورسازی (Visualize) کرده و عملکرد مدل را تحلیل کنید.

## دیتاست

• CIFAR-10:

- مجموعه داده استاندارد با 60,000 تصویر رنگی 32×32 پیکسلی در 10 کلاس مختلف
- تقسیم‌بندی داده‌ها:
  - \* آموزش (Train): 45,000 تصویر (90% از داده‌های اصلی آموزش)
  - \* اعتبارسنجی (Validation): 5,000 تصویر (10% از داده‌های اصلی آموزش)
  - \* تست (Test): 10,000 تصویر (مجموعه‌ی جداگانه)

## فاز اول: Model Designing

در این فاز باید یک شبکه عصبی طراحی کنید و بر روی دیتاست Cifar10 آموزش دهید. توجه کنید یک شبکه خیلی بزرگ طراحی نکنید و حداکثر 1,000,000 (یک میلیون) 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: Prevent Overfitting

مدل شما در فاز قبل به احتمال خیلی زیاد overfit می‌شوند. در این فاز باید از تمام روش‌ها و متدهایی که برای جلوگیری از overfit یاد گرفته‌اید استفاده کنید (تعدادی روش جدید هم بررسی کنید و در کد خود استفاده کنید). برای Augmentation Data 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.

در زمان تحویل پروژه لازم است به روش هایی که استفاده میکنید تسلط داشته باشید.

- برای این مدل حتماً 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. (تمامی آزمایشات در کد باشد و نتایج هم در داکيومنت بررسی شده باشد).
- بهترین مدل را حتماً برای فازهای بعد ذخیره کنید.

## فاز سوم: Transfer Learning

در این فاز از شما می‌خواهیم تا از بهترین مدلی که در فاز قبل آموزش داده‌اید برای آموزش مجدد بر روی دیتاست های جدید استفاده کنید.

- **caltech-101** - یک دیتاست شامل 101 کلاس مختلف است و کلاس‌ها تنوع بیشتری نسبت به دیتاست اصلی دارند.

۱. 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.
  ۲. 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.

• **Classification Horse Cat, Dog** - این دیتاست از 3 کلاس سگ و گربه و اسب تشکیل شده است. این دیتاست شامل داده های کمتری از دیتاست اصلی می باشد.

۱. برای این دیتاست نیز مانند دیتاست قبل یک بار لایه آخر مدل خود را بر اساس کلاس های جدید تغییر دهید و مدل را مجدد آموزش دهید و دفعه بعد یک بلاک آن را از freeze در بیاورید و مدل را آموزش دهید.

- حتماً از overfit شدن مدل خود جلوگیری کنید.

• **استفاده از مدل های معروف** - اکنون می خواهیم از مدل های معروف pretrained استفاده کنیم و آن ها را بر روی هر دو دیتاست FineTune کنیم.

۱. برای تمام مدل ها، hyperparameter tuning انجام دهید و به یک مقدار برای پارامترها بسنده نکنید.

۲. حتما روی هر دو دیتاست داده شده FineTune کنید.

۳. تمام نتایج را در داکيومنت خود بررسی کنید.

۴. در این بخش می خواهیم از سه مدل EfficientNetB0 و MobileNetV2 ResNet50 استفاده کنیم. برای این سه مدل لایه آخر را عوض کنید ولی تعداد لایه های قابل آموزش، دست خودتان است (از 10 لایه بیشتر نشود). در نهایت هم مدل ها را تست و موارد خواسته شده در ابتدای پروژه را انجام دهید. از overfit شدن مدل ها 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.
- در هنگام تحویل لازم است به مواردی که در پروژه استفاده کرده اید تسلط داشته باشید
- برای دیتاست‌های فاز سوم از نسبت 70% آموزش، 10% اعتبارسنجی و 20% تست استفاده کنید.
- 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 گروه‌ها تقلب محسوب می‌شود 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 می‌توانید با [ARHPA](#) و [Aalireza\\_s](#) و [Kourosh\\_Hsz](#) در ارتباط باشید یا در گروه درسی مطرح کنید.

موفق باشید؛  
تیم حل تمرین