

phase1:

تحلیل و گزارش فاز «تحلیل مسئله، طراحی مدل و آماده‌سازی داده‌ها» بر اساس نوت‌بوک Multi-Task fashion_multitasknet.ipynb به شرح زیر است. این پروژه یک رویکرد یادگیری چندمنظوره (Multi-Task Learning) را برای طبقه‌بندی محصولات مد و پوشاک اتخاذ کرده است.

۱. تعریف مسئله (Problem Definition)

مسئله اصلی در این پروژه، طبقه‌بندی چندگانه و چندخروجی (Multi-Output Classification) محصولات مد است. به جای اینکه برای هر ویژگی محصول (مانند رنگ، جنسیت، نوع لباس و ...) یک مدل جداگانه ساخته شود، یک مدل واحد طراحی شده است که با دریافت تصویر محصول، به طور همزمان چندین ویژگی آن را پیش‌بینی می‌کند.

- **اهداف دقیق:** پیش‌بینی همزمان ۶ ویژگی برای هر تصویر:
 1. جنسیت (Gender)
 2. دسته‌بندی اصلی (Master Category)
 3. زیرمجموعه (Sub Category)
 4. رنگ پایه (Base Colour)
 5. فصل (Season)
 6. نوع کالا (Article Type)

هدف

۲. انتخاب دیتاست (Dataset Selection)

دیتاست مورد استفاده، مجموعه‌ای از تصاویر محصولات مد (Fashion Product Images) است که شامل پوشه تصاویر و یک فایل متادادا (styles.csv) می‌باشد.

- این دیتاست ساختار یافته است و هر سطر در فایل CSV متناظر با یک تصویر است که ویژگی‌های آن را توصیف می‌کند.

۳. تحلیل داده‌ها و (Exploratory Data Analysis (EDA))

در این نوت‌بوک، تحلیل‌های آماری برای درک توزیع داده‌ها انجام شده است:

- **توزيع کلاس‌ها:** توزیع دسته‌بندی‌های مختلف (مانند subCategory) بررسی شده است. نتایج نشان می‌دهد که برخی کلاس‌ها (مانند Topwear) بسیار پر تکرار و برخی دیگر (مانند Perfumes) بسیار کمیاب هستند.
- **شناسایی داده‌های پرت:** کلاس‌هایی که تعداد نمونه‌های آن‌ها بسیار کم است (مثلاً کمتر از ۵۰ نمونه)، شناسایی شده‌اند تا از ایجاد نویز در مدل جلوگیری شود.

۴. پیش‌پردازش داده‌ها (Data Preprocessing)

مراحل پیش‌پردازش به صورت جامع و سیستماتیک پیاده‌سازی شده است:

- **پاکسازی داده‌ها (Cleaning):**
 - حذف ردیف‌هایی که مقادیر خالی (Null) در ستون‌های هدف دارند.
 - حذف ردیف‌هایی که فایل تصویر آن‌ها در پوشه تصاویر موجود نیست.
 - **فیلتر کردن کلاس‌های کمیاب:** حذف کلاس‌هایی که تعداد نمونه‌های آن‌ها کمتر از یک آستانه مشخص (مثلاً ۵۰) است تا تعادل نسبی در آموزش برقرار شود.
- **مهندسی ویژگی (Feature Engineering):**
 - ساده‌سازی رنگ‌ها: برای کاهش پیچیدگی، رنگ‌های مشابه ادغام شده‌اند. برای مثال، رنگ‌های کلاس‌های رنگ را به ۲۰ کلاس استاندارد کاهش داده است.
- **کدگذاری (Encoding):**
 - استفاده از LabelEncoder برای تبدیل برچسب‌های متغیر (مانند "Red", "Men") به اعداد صحیح برای هر ۶ ستون هدف به صورت جداگانه.
- **تقسیم داده‌ها (Splitting):**
 - داده‌ها به سه دسته تقسیم شده‌اند: آموزش (Train)، اعتبارسنجی (Validation) و تست (Test). از روش stratify استفاده شده تا توزیع کلاس‌ها در هر سه دسته یکسان باقی بماند.
- **آماده‌سازی تصویر (Augmentation & Normalization):**
 - تغییر اندازه تصاویر به ۲۲۴x۲۲۴ پیکسل.
 - نرمال‌سازی تنзорها با میانگین و انحراف معیار استاندارد ایمیجن‌نت (Imagenet). ([Std=[0.229, 0.224, 0.225] و [0.406

۵. انتخاب مدل و معماری (Model Architecture)

معماری انتخاب شده یک شبکه عصبی چند وظیفه‌ای (Multi-Task Neural Network) است.

- **بدنه اصلی (Backbone)**: از مدل‌های پیش‌آموزش‌دیده (Pre-trained) قدرتمند استفاده شده است.
نوت‌بوك امکان انتخاب بین چندین معماری مشهور را فراهم کرده است:
 - ResNet50
 - EfficientNet_B5
 - ViT_B_16 (Vision Transformer)
 - ConvNeXt_Tiny
- **سرهای خروجی (Heads)**: خروجی بدنه اصلی (Features) به ۶ لایه تمام‌متصل (Fully Connected) جدأگانه وارد می‌شود که هر کدام مسئول پیش‌بینی یکی از ویژگی‌ها (جنسیت، رنگ، فصل و ...) هستند.
- **توجیه علمی**: استفاده از بدنه مشترک باعث می‌شود مدل الگوهای بصری پایه‌ای (مانند بافت و شکل کلی) را یاد بگیرد که بین همه تسكها مشترک است، در حالی که سرهای جدأگانه امکان یادگیری ویژگی‌های اختصاصی هر تسك را فراهم می‌کنند. این روش از نظر محاسباتی بسیار بهینه‌تر از آموزش ۶ مدل جدأگانه است.

۶. طراحی آزمایش‌ها و معیار ارزیابی (Evaluation Plan)

- **تابع هزینه (Loss Function)**: از CrossEntropyLoss برای هر یک از ۶ خروجی استفاده شده است. هزینه نهایی مدل، مجموع هزینه‌های این ۶ تسك است.
- **بهینه‌ساز (Optimizer)**: استفاده از AdamW با نرخ یادگیری متفاوت برای بدنه اصلی (نرخ کمتر برای حفظ دانش قبلی) و سرهای خروجی (نرخ بیشتر برای یادگیری سریع‌تر).
- **معیارهای ارزیابی (Metrics)**: برای هر تسك به صورت جدأگانه دو معیار محاسبه می‌شود:
 1. **دقت (Accuracy)**: درصد پیش‌بینی‌های کاملاً صحیح.
 2. **امتیاز F1-Macro**: برای سنجش عملکرد مدل در کلاس‌های نامتوازن (بسیار مهم برای ویژگی‌های مثل رنگ یا نوع کالا که توزیع نامتوازن دارند).
- **روند آزمایش**: مدل‌های مختلف (ResNet, ViT, EfficientNet) به نوبت آموزش داده شده و نتایج آن‌ها روی داده‌های تست مقایسه می‌شود.

این ساختار نشان‌دهنده یک پروژه کامل و استاندارد در حوزه بینایی ماشین است که تمامی مراحل چرخه حیات داده (Data Lifecycle) را پوشش می‌دهد.

phase2:

مدل نهایی انتخاب شده

مدل نهایی بر اساس نتایج مقایسه‌ی مدل‌ها (F1-Macro و Accuracy) در همه‌ی Test روی داده‌ی وظایف انتخاب شده است.

ConvNeXt-Tiny: مدل نهایی

علت انتخاب:

- بیشترین/بهترین میانگین Avg F1-Macro و Avg Accuracy در میان وظایف
- پایداری بهتر در آموزش و همگرایی مناسب
- عملکرد متوازن روی تمام وظایف (نه فقط یک وظیفه).

(Hyperparameter Tuning + Regularization)

در نسخه‌ی Tuned (بهبود یافته) از موارد زیر استفاده شده است:

بهبودهای مربوط به Hyperparameter / Training Strategy

: Differential Learning Rates •

نرخ یادگیری Backbone کمتر و برای Head‌ها بیشتر در نظر گرفته شده است (محافظه‌کارانه‌تر به روزرسانی می‌شود).

: OneCycleLR Scheduler •

تغییر نرخ یادگیری در طول آموزش برای همگرایی بهتر و جلوگیری از گیر کردن در مینیمم‌های ضعیف.

: Optimizer: AdamW •

به دلیل رفتار بهتر در تنظیم وزن‌ها و سازگاری با Weight Decay.

Regularization / Stabilization

: Overfitting کاهش Weight Decay •

برای جلوگیری از جهش گرادیان و پایداری آموزش **Gradient Clipping** •

در نسخه **Baseline** (بدون بهبودها):

Gradient Clipping و Weight Decay، بدون Scheduler، بدون Adam ساده، بدون Weight Decay •

ذخیره Checkpoint

در طول آموزش مدل‌ها، دو نوع checkpoint ذخیره می‌شود:

`val_loss`: بهترین مدل بر اساس کمینه شدن **Best** •

epoch: مدل آخرین **Last** •

این کار برای بازتولید نتایج و جلوگیری از از دست رفتن بهترین وزن‌ها ضروری است

نمودارها (Graphs)

معیارهای ارزیابی و نمودارها

متريک‌های محاسبه شده

:task برای هر

Accuracy •

F1-Macro •

Precision •

Recall •

نمودارهای پروژه

در نوبت بوك شما اين نمودارها توليد مي شوند:

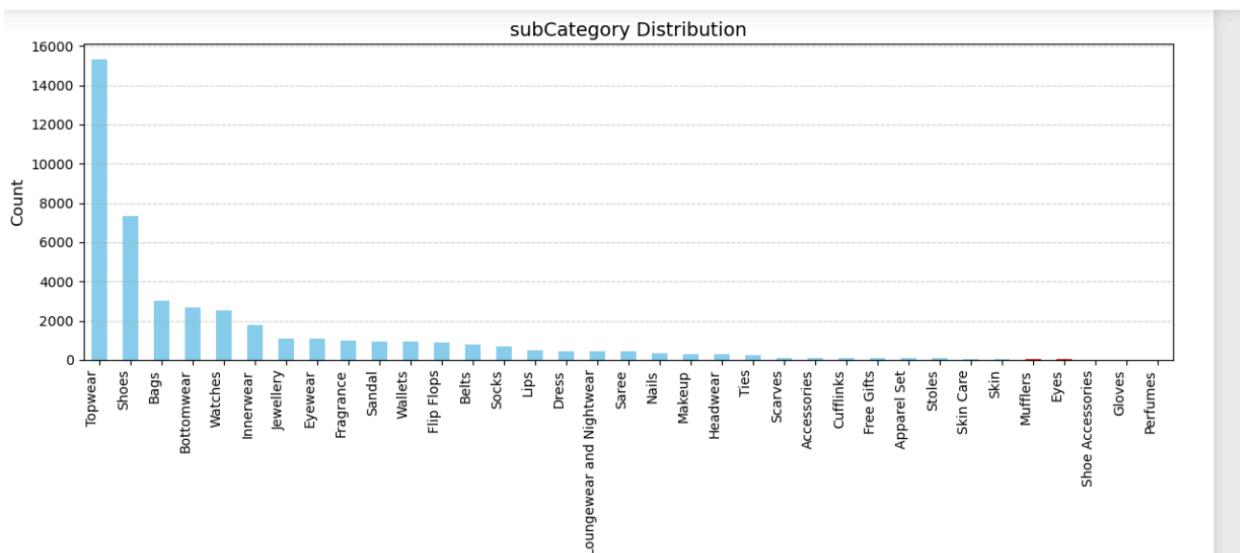
1. EDA: نمودار توزيع subCategory

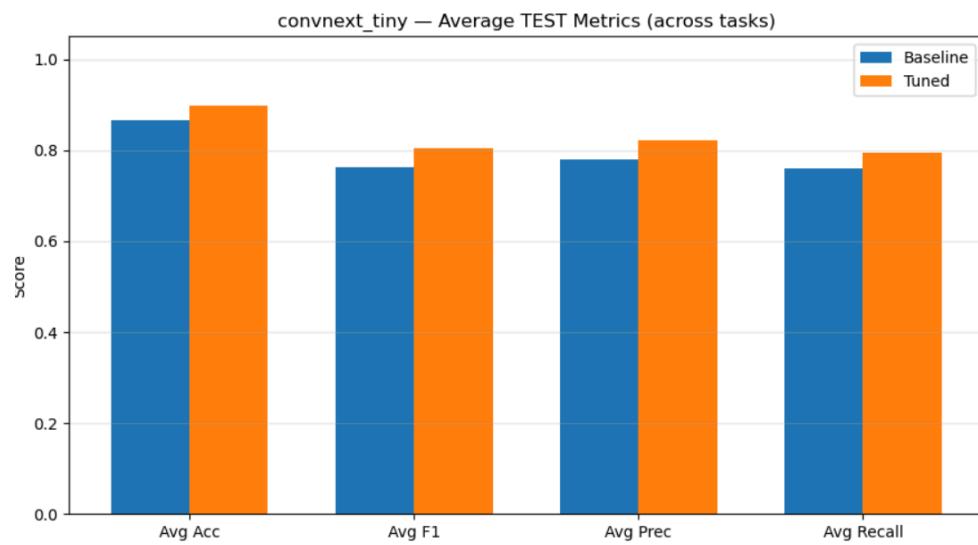
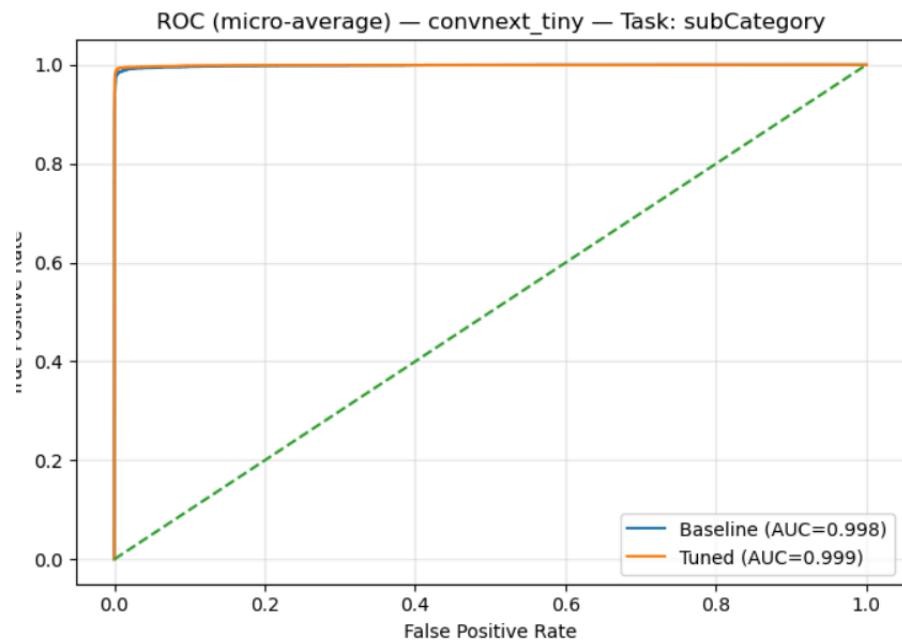
2. Bar chart: Accuracy و F1 برای هر task و هر مدل

3. Bar chart: خلاصه عملکرد کلی: میانگین Accuracy و میانگین F1 روی همه task ها

4. Ablation plot: فقط میانگین متريک‌های تست Avg Acc / Avg F1 / Avg Prec / Avg با task برای baseline vs tuned (Recall)

5. ROC Curve: برای هر task یک ROC (خطی) با AUC برای tuned (micro-average) و baseline در حالت multiclass





شکل ۱) نمودار توزیع subCategory EDA

این نمودار نشان می‌دهد که توزیع کلاس‌های زیردسته (subCategory) به شدت نامتوازن است:

- چند کلاس پرترکار مثل Shoes و Topwear تعداد نمونه بسیار بالایی دارند (میله‌های بسیار بلند در سمت چپ).
- در مقابل، تعداد زیادی کلاس در (میله‌های کوتاه در سمت راست) که بعضی از آنها نمونه‌های بسیار کمی دارند (حتی نزدیک به صفر).

تفسیر و اهمیت

1. اثر عدم توازن روی یادگیری مدل:
وقتی چند کلاس خیلی پرترکار باشند، مدل به شکل طبیعی تمایل پیدا می‌کند روی همان‌ها خوب عمل کند و روی کلاس‌های کم‌نمونه ضعیف شود. این دقیقاً همان جایی است که معیارهایی مثل F1-Macro اهمیت پیدا می‌کنند (چون به کلاس‌های کم‌نمونه هم وزن می‌دهد).

2. توجیه علمی حذف کلاس‌های نادر (Filtering):
وجود کلاس‌های بسیار کم‌نمونه باعث می‌شود:

- گردایان‌ها نویزی شوند،
 - مدل «الگوی پایدار» از کلاس نادر یاد نگیرد،
 - و معیارهای Macro (مثل F1-Macro) افت کند.
- بنابراین اقدام شما برای اعمال threshold (مثلاً 50) کاملاً منطقی و استاندارد است.

این نمودار توضیح می‌دهد چرا برخی وظایف (به خصوص subCategory و articleType) در حالت کلی دشوارتر از masterCategory هستند: چون تعداد کلاس‌ها بیشتر است و توزیع آنها نامتوازن‌تر.

– **subCategory** برای ROC Curve micro-average (۲) شکل و مقایسه Tuned و Baseline

در نمودار ROC مربوط به **subCategory** مشاهده می‌شود:

- هر دو منحنی بسیار نزدیک به گوشه بالا-چپ هستند.
- مقدار AUC برای Baseline حدود ۰.۹۹۸ و برای Tuned حدود ۰.۹۹۹ گزارش شده است.

تفسیر

۱. AUC نزدیک به ۱ یعنی تفکیک‌پذیری بسیار بالا یعنی مدل در سطح احتمال‌ها (score/probability) می‌تواند نمونه‌های صحیح را تقریباً همیشه بالاتر از نمونه‌های غلط رتبه‌بندی کند.

شکل ۳) نمودار Baseline vs – (میانگین متريک‌های تست) – Tuned

در اين نمودار چهار معيار ميانگين روی همه تسکها مقایسه شده است:

Avg Accuracy •

Avg F1 •

Avg Precision •

Avg Recall •

و در هر معیار دو ستون وجود دارد: **Tuned** و **Baseline**.

1. بهبود **Tuned** در هر چهار معیار قابل مشاهده است

یعنی روش‌های بهبود (مثل AdamW + weight decay + OneCycleLR + differential LR) فقط روی یک معیار اثر نگذاشته‌اند، بلکه کیفیت کلی مدل را بهتر کرده‌اند.

2. بهبود **F1/Recall** مهم‌تر از صرفاً **Accuracy** است

چون دیتاست نامتوازن است، اگر فقط Accuracy بهتر شود ممکن است صرفاً بهبود در کلاس‌های پرتکرار باشد. اما بهتر شدن **F1** و **Recall** نشان می‌دهد:

◦ مدل روی کلاس‌های ضعیفتر هم بهتر شده،

◦ و تعداد **false negative** ها کاهش یافته (**Recall** بالاتر).

جدول مقایسه مدل ها

EfficientNet-B 5	ViT-B/16	ResNet50	ConvNeXt Tiny	ویژگی / مدل
91.05%	92.54%	93.43%	94.10%	Gender Acc
98.90%	99.22%	99.41%	99.63%	MasterCategory Acc
95.45%	96.80%	97.21%	97.51%	SubCategory Acc
68.26%	72.54%	72.59%	75.61%	BaseColour Acc
75.15%	76.34%	78.47%	81.69%	Season Acc
85.77%	88.97%	88.99%	91.03%	ArticleType Acc

3.25	2.56	2.67	2.48	(کمتر بهتر) Test Loss
------	------	------	-------------	------------------------------

همانطور که از نتایج برمی‌آید، مدل ConvNeXtTiny بهترین عملکرد کلی را با کمترین میزان Loss از خود نشان داده است.

- **MasterCategory** با دقت حدود 99% برای همه مدل‌ها
(چون دسته‌بندی کلی و coarse است؛ کلاس‌ها کمتر و مرزها واضح‌تر است.)

سخت‌ترین تسک‌ها:

- **BaseColour** کمترین دقت را دارد (%68 تا %76)
(چون رنگ به نور، سایه، کیفیت عکس، رنگ‌های نزدیک و حتی پس‌زمینه حساس است.)
- **Season** نیز از تسک‌های متوسط رو به سخت است (%75 تا %82)
(زیرا فصل بیشتر "مفهومی" است و ممکن است از روی تصویر تنها به‌طور مستقیم قابل تشخیص نباشد.)

تحلیل خطای خروجی (Error Analysis)

با بررسی نتایج خروجی، نکات زیر در مورد خطاهای مدل قابل مشاهده است:

1. **چالش رنگ‌ها (BaseColour):** دقیق تشخصیص رنگ (حدود ۷۵٪) نسبت به سایر ویژگی‌ها (مثل دسته‌بندی اصلی با ۹۹٪) به مراتب کمتر است.
 - علت: وجود رنگ‌های مشابه (مثل آبی تیره و مشکی، یا کرم و سفید) و تاثیر نورپردازی در تصاویر باعث این خطای شود. همچنین کلاس‌های ترکیبی مثل "Multi" چالش‌برانگیز هستند.
2. **عدم توازن داده‌ها:** کلاس‌هایی که نمونه‌های کمتری دارند (مثل رنگ‌های خاص یا زیردسته‌هایی مثل "Perfumes") دقیق پایین‌تری دارند، که در نمودارهای توزیع اولیه نیز پیش‌بینی می‌شد.

:90

gender: Men (0.98)
masterCategory: Apparel (1.00)
subCategory: Topwear (1.00)
baseColour: Blue (0.98)
season: Summer (0.82)
articleType: Tshirts (0.99)



gender: Men (0.66)
masterCategory: Accessories (1.00)
subCategory: Eyewear (1.00)
baseColour: Black (0.30)
season: Winter (0.99)
articleType: Sunglasses (1.00)



12) نتیجه‌گیری

در این پژوهه:

- با رویکرد **Multi-Task Learning** ویژگی اصلی محصولات مد را همزمان از تصویر پیش‌بینی کنیم.
- پیش‌پردازش‌های هدفمند (حذف کلاس‌های نادر، ساده‌سازی رنگ) داده را تمیزتر و آموزش را پایدارتر کرده است.
- مقایسه چند معماری نشان می‌دهد انتخاب backbone نقش مهمی در کیفیت دارد.
 - معماری **ConvNeXt Tiny** بهترین تعادل را بین دقت و کارایی نشان داد و در تمامی تسكّع‌ها بالاترین امتیاز را کسب کرد.
- AdamW، weight decay، OneCycleLR نشان می‌دهد استفاده از بهبودهایی مثل **Ablation Study** و gradient clipping به شکل قابل اندازه‌گیری باعث بهتر شدن میانگین متريک‌های تست می‌شود.
- ذخیره checkpoint باعث می‌شود مدل نهایی قابل بازیابی و قابل ارائه باشد.
- دموهای طبقه‌بندی و captioning، پژوهه را از حالت صرفاً عددی خارج کرده و قابلیت ارائه عملی را بالا می‌برد.