

phase1:

تحلیل و گزارش فاز «تحلیل مسئله، طراحی مدل و آماده‌سازی داده‌ها» بر اساس نوت‌بوک `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:

۱. جنسیت (Gender)
۲. دسته‌بندی اصلی (Master Category)
۳. زیرمجموعه (Sub Category)
۴. رنگ پایه (Base Colour)
۵. فصل (Season)
۶. 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 متادیتا (`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:

- **توزیع کلاس‌ها:** توزیع دسته‌بندی‌های مختلف (مانند subCategory) بررسی شده است. نتایج نشان می‌دهد که برخی کلاس‌ها (مانند Topwear) بسیار پرتکرار 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 Charcoal، Grey Melange و Steel همگی به برچسب کلی Gray تغییر یافته‌اند. این کار تعداد کلاس‌های رنگ را به ۲۰ کلاس استاندارد کاهش داده است.
- **کدگذاری (Encoding):**
 - استفاده از LabelEncoder برای تبدیل برچسب‌های متنی (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 تنسورها با میانگین و انحراف معیار استاندارد ایمیجنت (Mean=[0.485, 0.456, 0.406] و Std=[0.229, 0.224, 0.225]).

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.
 - نوت‌بوک امکان انتخاب بین چندین معماری مشهور را فراهم کرده است:
 - ResNet50: مناسب برای استخراج ویژگی‌های استاندارد.
 - EfficientNet_B5: مناسب برای دقت بالاتر با پارامترهای بهینه.
 - (ViT_B_16 (Vision Transformer): برای استفاده از مکانیزم توجه (Attention) روی تصاویر.
 - ConvNeXt_Tiny: یک معماری مدرن CNN.
- **سرهای خروجی (Heads):** خروجی بدنه اصلی (Features) به ۶ لایه تمام‌متصل (Fully Connected) جداگانه 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 تسک‌ها مشترک است، در حالی که سرهای جداگانه امکان یادگیری ویژگی‌های اختصاصی هر تسک را فراهم می‌کنند. این روش از نظر محاسباتی بسیار بهینه‌تر از آموزش ۶ مدل جداگانه 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 تسک 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 تسک به صورت جداگانه 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 ویژگی‌هایی 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 (Accuracy و F1-Macro روی داده‌ی Test در همه‌ی وظایف) انتخاب شده است.

مدل نهایی: ConvNeXt-Tiny

علت انتخاب:

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

استراتژی آموزش و بهبودها (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 (Backbone محافظه‌کارانه‌تر به‌روزرسانی 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

- **Weight Decay** برای کاهش Overfitting

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

در نسخه‌ی **Baseline** (بدون بهبودها):

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

ذخیره Checkpoint

در طول آموزش مدل‌ها، دو نوع checkpoint ذخیره می‌شود:

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

- **Last**: مدل آخرین epoch

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

نمودارها (Graphs)

معیارهای ارزیابی و نمودارها

متریک‌های محاسبه‌شده

برای هر task:

- Accuracy

- F1-Macro

- Precision

- Recall

نمودارهای پروژه

در نوت‌بوک شما این نمودارها تولید می‌شوند:

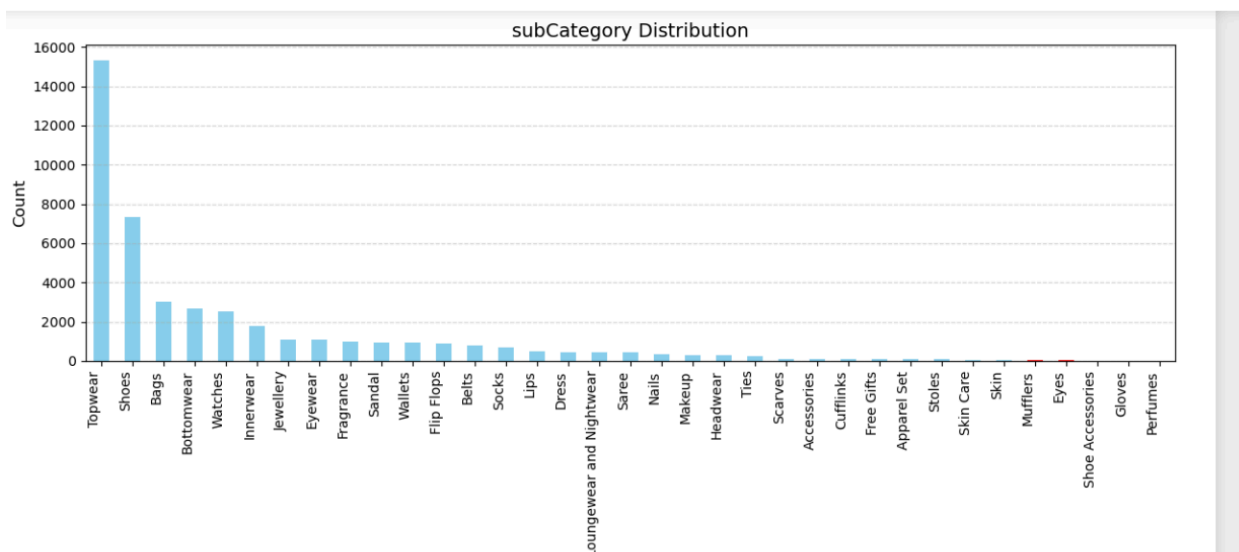
1. **EDA**: نمودار توزیع subCategory

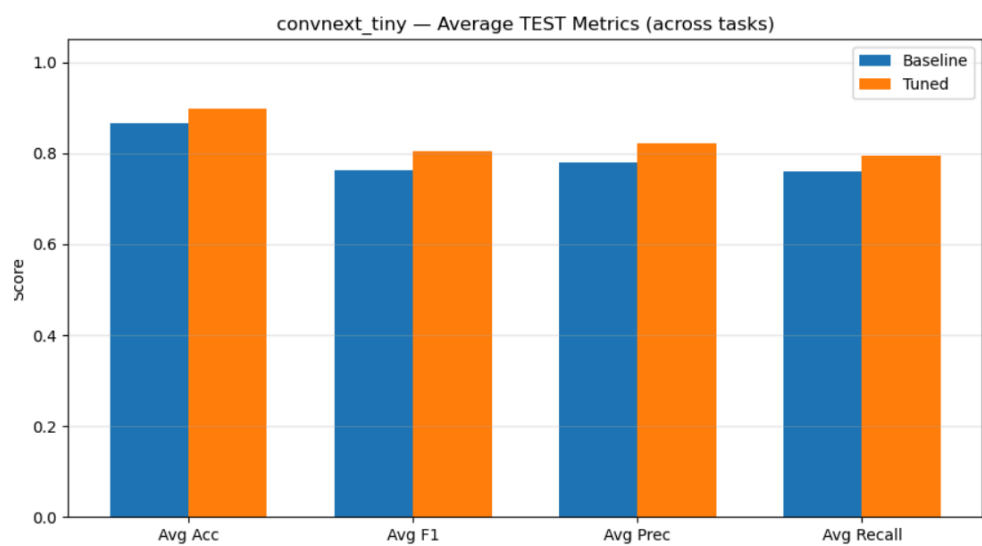
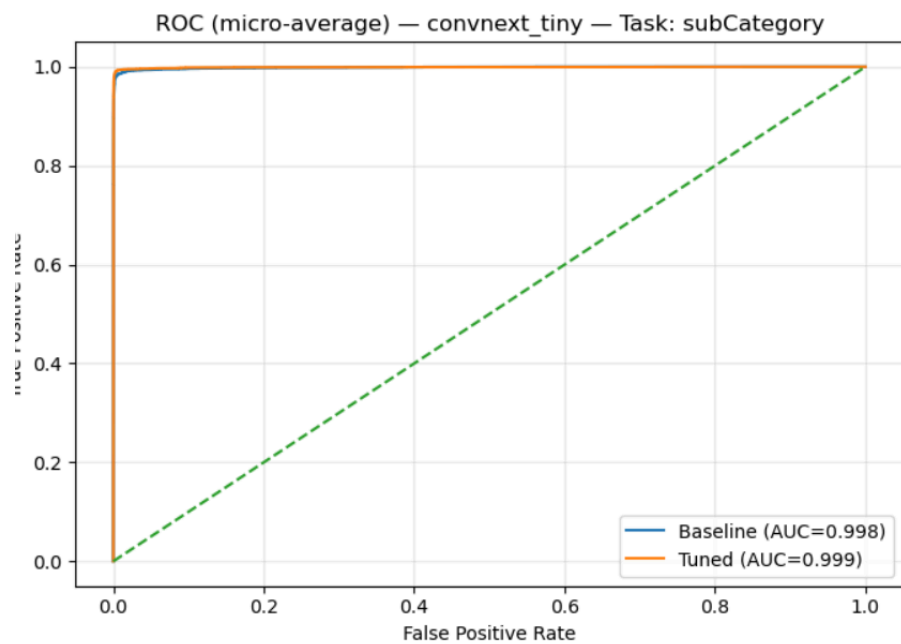
2. **نمودار مقایسه مدل‌ها**: Bar chart برای Accuracy و F1 برای هر task و هر مدل

3. **خلاصه عملکرد کلی**: Bar chart میانگین Accuracy و میانگین F1 روی همه task‌ها

4. **Ablation plot**: فقط Bar chart میانگین متریک‌های تست (Avg Acc / Avg F1 / Avg Prec / Avg Recall) برای baseline vs tuned

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





شکل ۱) نمودار توزیع EDA subCategory

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

- چند کلاس پرتکرار مثل **Topwear** و **Shoes** 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 پرتکرار 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:

- گرادیان‌ها 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 (به‌خصوص subCategory و articleType) در حالت کلی دشوارتر از masterCategory هستند: چون تعداد کلاس‌ها بیشتر است و توزیع آن‌ها نامتوازن‌تر.

شکل ۲) ROC Curve micro-average برای subCategory – 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 0.998 و برای Tuned حدود 0.999 گزارش شده است.

تفسیر

1. AUC نزدیک به 1 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 Ablation (میانگین متریک‌های تست) – Baseline vs Tuned

در این نمودار چهار معیار میانگین روی همه تسک‌ها مقایسه شده است:

• Avg Accuracy

• Avg F1

• Avg Precision

• Avg Recall

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

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

یعنی روش‌های بهبود (مثل $\text{AdamW} + \text{weight decay} + \text{OneCycleLR} + \text{differential LR}$ + gradient clipping) 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	Test Loss (کمتر بهتر)
------	------	------	------	-----------------------

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 خطا (Error Analysis)

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

1. **چالش رنگ‌ها (BaseColour):** دقت تشخیص 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 خطا می‌شود. همچنین کلاس‌های ترکیبی مثل "Multi" چالش‌برانگیز هستند.
2. **عدم توازن داده‌ها:** کلاس‌هایی که نمونه‌های کمتری دارند (مثل رنگ‌های خاص یا زیردسته‌هایی مثل "Perfumes") دقت پایین‌تری دارند، که در نمودارهای توزیع اولیه نیز پیش‌بینی می‌شد.

دمو:

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 پروژه:

- با رویکرد **Multi-Task Learning** توانستیم ۶ 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 تسک‌ها بالاترین امتیاز را کسب کرد.
- **Ablation Study** نشان می‌دهد استفاده از بهبودهایی مثل AdamW، weight decay، OneCycleLR و gradient clipping به شکل قابل اندازه‌گیری باعث بهتر شدن میانگین متریک‌های تست می‌شود.
- ذخیره checkpoint باعث می‌شود مدل نهایی قابل بازیابی و قابل ارائه باشد.
- دموهای طبقه‌بندی و captioning، پروژه را از حالت صرفاً عددی خارج کرده و قابلیت ارائه عملی را بالا می‌برد.