دانشگاه فردوسی مشهد

گروه مهندسی کامپیوتر

گزارش پروژه درس

مبانی هوش محاسباتی

Blood Cell Classification with CNNs

دسته‌بندی تصاویر گلبول‌های سفید خون با شبکه‌های عصبی کانولوشنی

نام و نام خانوادگی	شماره دانشجویی
امیرحسین ابوالفضلی	۴۰۲۲۲۶۲۰۳۵
آرمان بیجاری	۴۰۲۱۲۶۲۱۳۱

استاد: **دکتر فضل ارثی**

نیمسال دوم ۱۴۰۵–۱۴۰۴

فهرست مطالب

۱ مقدمه ۳

۲ مجموعه داده ۳
۱.۲ معرفی ۳
۲.۲ ساختار و توزیع داده ۳
۳.۲ تحلیل اکتشافی داده (EDA) ۳
۱.۳.۲ نمونه تصاویر از هر کلاس ۳
۲.۳.۲ توزیع کلاس‌ها ۴
۳.۳.۲ ابعاد تصاویر ۴
۴.۳.۲ توزیع شدت پیکسل ۴
۵.۳.۲ تصویر میانگین هر کلاس ۵

۳ پیش‌پردازش و افزایش داده ۵
۱.۳ تغییر اندازه و نرمال‌سازی ۵
۲.۳ تقسیم داده: آموزش و اعتبارسنجی ۶
۳.۳ ساخت Data Generator ۶
۴.۳ افزایش داده ۷
۵.۳ نمایش یک batch از داده‌های آموزشی ۷

۴ مدل CNN از پایه ۸
۱.۴ معماری ۸
۲.۴ تنظیمات آموزش ۹
۳.۴ نمودار آموزش ۹
۴.۴ ارزیابی روی مجموعه آزمون ۱۰
۵.۴ ماتریس درهم‌ریختگی ۱۰
۶.۴ منحنی ROC ۱۱
۷.۴ نمونه پیش‌بینی‌ها ۱۲
۸.۴ تحلیل نتایج ۱۲

۵ مدل CNN بهبودیافته ۱۳
۱.۵ تغییرات نسبت به مدل پایه ۱۳
۲.۵ تنظیم فراپارامترها (Hyperparameter Tuning) ۱۳
۳.۵ نمودارهای آموزش ۱۴
۴.۵ نتایج کامل مدل نهایی روی مجموعه‌ی آزمون ۱۴
۵.۵ مقایسه‌ی عددی با فاز دوم ۱۵
۶.۵ تحلیل ماتریس درهم‌ریختگی ۱۶
۷.۵ علت کاهش عملکرد نسبت به فاز دوم ۱۷

۶ تفسیر مدل **۱۷**
۱.۶ فیلترهای کانولوشن ۱۷
۲.۶ نقشه‌های فعال‌سازی ۱۸
۳.۶ Grad-CAM ۱۹
۴.۶ مقایسه‌ی پیش‌بینی‌های درست و اشتباه ۲۰

۷ یادگیری انتقالی با MobileNetV2 **۲۲**
۱.۷ تغییر لایه‌ی خروجی ۲۲
۲.۷ پیش‌پردازش ورودی ۲۲
۳.۷ پروتکل آموزش دو مرحله‌ای ۲۲
۱.۳.۷ مرحله‌ی اول — آموزش سر طبقه‌بند ۲۲
۲.۳.۷ مرحله‌ی دوم — Fine-tuning لایه‌های بالایی ۲۳
۴.۷ تنظیم ابرپارامترها ۲۳
۵.۷ نمودارهای آموزش ۲۳
۶.۷ ارزیابی روی مجموعه‌ی آزمون ۲۴
۱.۶.۷ ماتریس درهم‌ریختگی ۲۴
۲.۶.۷ منحنی‌های ROC ۲۵
۳.۶.۷ نمونه‌های پیش‌بینی ۲۵
۷.۷ مقایسه‌ی نهایی با فازهای قبل ۲۶
۸.۷ تحلیل نهایی ۲۷

۱. مقدمه

تشخیص نوع گلبول‌های سفید خون (White Blood Cells) یه کار مهمه که توی آزمایشگاه‌های پزشکی هر روز انجام می‌شه — و درواقع خیلی چیزا بهش وابسته‌ست، از تشخیص عفونت گرفته تا آلرژی و سرطان خون. روش سنتیش اینه که یه متخصص می‌شینه پشت میکروسکوپ و یه‌یه سلول‌ها رو بررسی می‌کنه، که هم وقت‌گیره، هم خطاپذیر — بخاطر همین جای خوبیه که یه مدل هوشمند بیاد این کار رو بکنه.

توی این پروژه از شبکه‌های عصبی کانولوشنی (CNN) استفاده می‌کنیم تا تصاویر میکروسکوپی گلبول‌های سفید رو به چهار کلاس اصلی دسته‌بندی کنیم. دو رویکرد بررسی می‌شه: اول آموزش یه CNN از صفر، بعد هم یادگیری انتقالی با مدل‌هایی که قبلاً روی ImageNet آموزش دیدن. پس هدف اینه که ببینیم کدوم رویکرد بهتر کار می‌کنه و چرا.

۲. مجموعه داده

۱.۲. معرفی

مجموعه داده‌ای که استفاده می‌کنیم Blood Cell Images نام داره، که Paul Timothy Mooney اون رو روی Kaggle منتشر کرده و تصاویرش اصلاً از موسسه ملی بهداشت آمریکا (NIH) گرفته شده. مجوزش هم CC0 (Public Domain) ه — یعنی کاملاً آزاده. تصاویر به فرمت JPEG رنگی با ابعاد 320×240 پیکسل ذخیره شدن و توی پوشه‌هایی با نام کلاس مربوطه سازماندهی شدن، که این ساختار مستقیماً با flow_from_directory کرس سازگاره.

۲.۲. ساختار و توزیع داده

بخش	تعداد کل	تعداد به ازای هر کلاس	کاربرد
TRAIN	9,957	$\approx$ 2,490	آموزش و اعتبارسنجی
TEST	2,487	$\approx$ 622	ارزیابی نهایی
TEST_SIMPLE	۷۱	متغیر	نمایش و دمو

جدول ۱: توزیع مجموعه داده در سه بخش

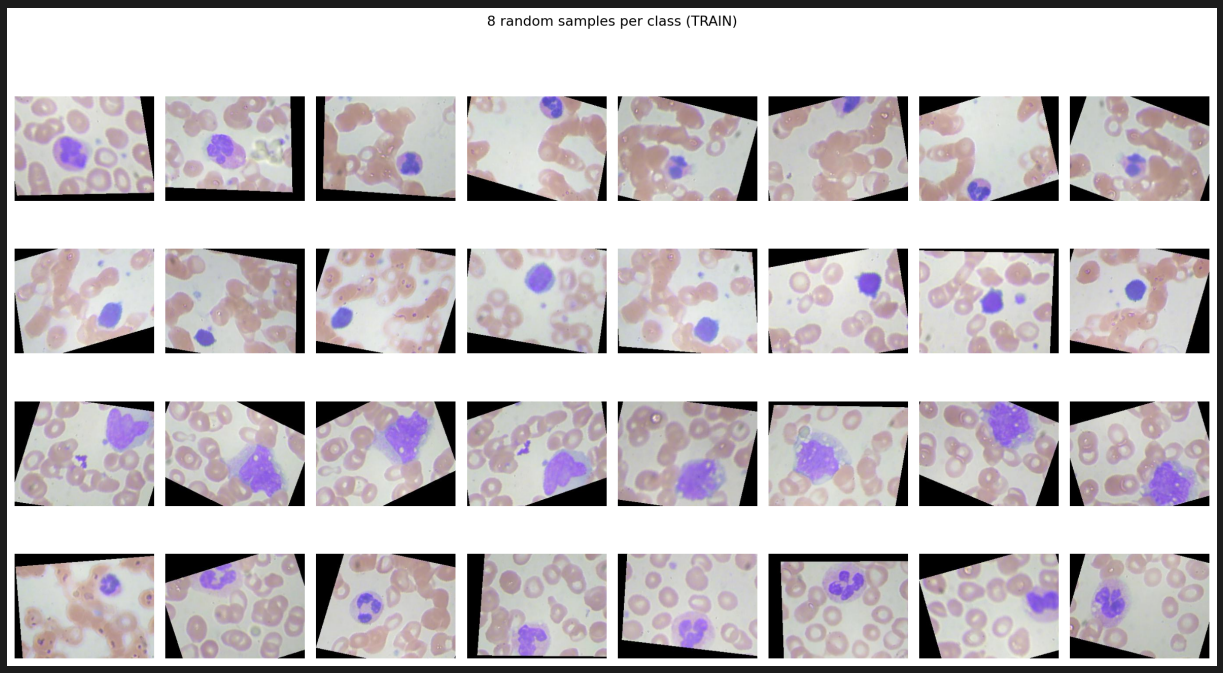
چهار کلاس هدف داریم که هرکدوم از نظر بصری یه سری ویژگی خاص دارن. Eosinophil با دانه‌های قرمز-نارنجی مشخصش راحت شناخته می‌شه و نقش اصلیش توی واکنش‌های آلرژیکه. Lymphocyte یه سلول گرده با هسته بزرگ و تیره که به سیستم ایمنی مرتبطه. Monocyte بزرگ‌ترین گلبول سفیده و هسته‌اش شکل کلیوی داره. Neutrophil هم شایع‌ترین گلبول سفیده — هسته چندلوبه بنفش‌رنگش کاملاً متمایزه. البته تشخیص بین این چهارتا برای مدل همیشه به یه سختیه، که جلوتر بیشتر می‌بینیم.

۳.۲. تحلیل اکتشافی داده (EDA)

۱.۳.۲. نمونه تصاویر از هر کلاس

شکل ۱ چند نمونه از هر چهار کلاس رو نشون می‌ده. تفاوت بصری بین کلاس‌ها معمولاً قابل مشاهده‌ست، ولی جایی که کار سخت می‌شه اینه که Monocyte و Lymphocyte بعضی وقتا خیلی شبیه هم به نظر می‌رسن —

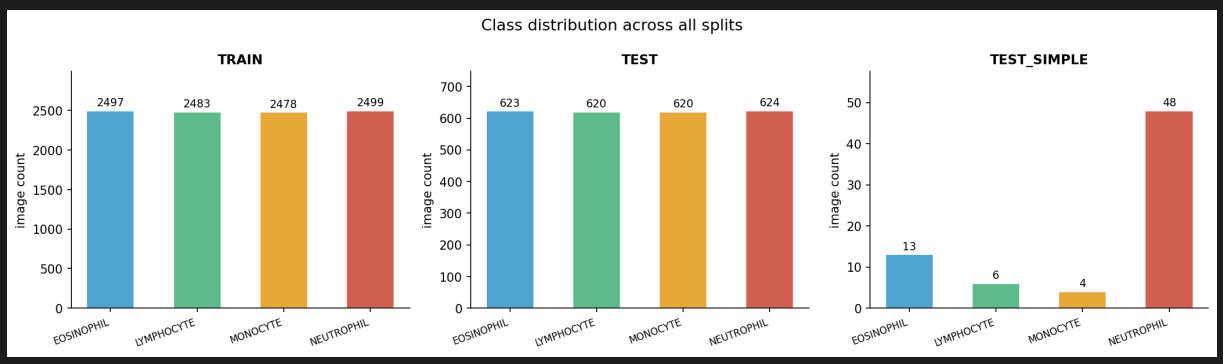
و این دقیقاً همون جاییه که مدل هم بیشتر اشتباه می‌کنه.



شکل ۱: ۸ نمونه تصادفی از هر کلاس در مجموعه آموزشی

۲.۳.۲. توزیع کلاس‌ها

شکل ۲ نشون می‌ده که توزیع کلاس‌ها توی هر سه بخش تقریباً متوازنه — EOSINOPHIL با 2,497 نمونه، LYMPHOCYTE با 2,483، MONOCYTE با 2,478 و NEUTROPHIL با 2,499 تا توی مجموعه آموزشی. خوبیش اینه که این تعادل یعنی نه نیازی به oversampling داریم، نه نیازی به وزن‌دهی کلاس — مدل از اول با یه دیتای منصفانه سر و کار داره.



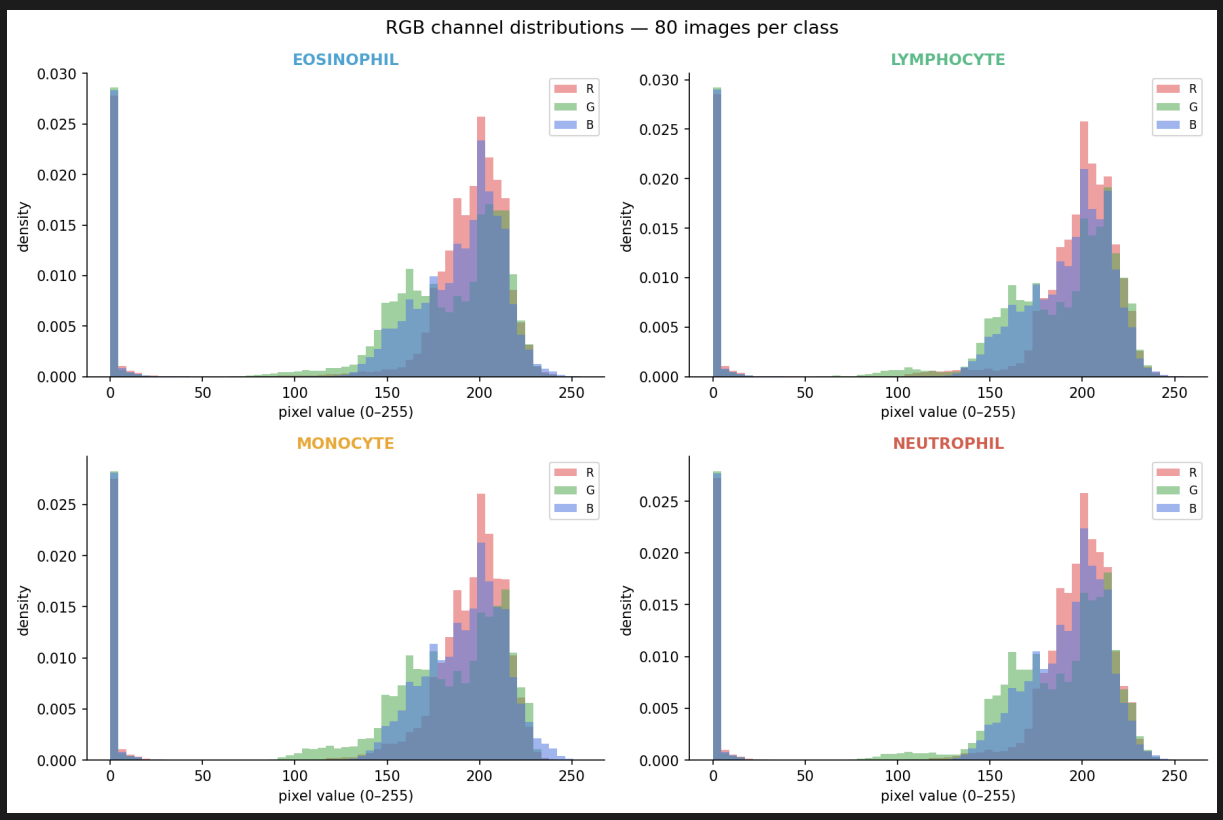
شکل ۲: تعداد تصاویر هر کلاس در سه بخش TRAIN، TEST و TEST_SIMPLE

۳.۳.۲. ابعاد تصاویر

بررسی 300 نمونه تصادفی از مجموعه آموزشی نشون داد که همه تصاویر دقیقاً 240×320 پیکسل و فرمت RGB دارن — هیچ ناهمگونی توی ابعاد نیست. توی مرحله آموزش این تصاویر به 224×224 پیکسل تغییر اندازه داده می‌شن که هم برای CNN سفارشی مناسبه، هم برای MobileNetV2.

۴.۳.۲. توزیع شدت پیکسل

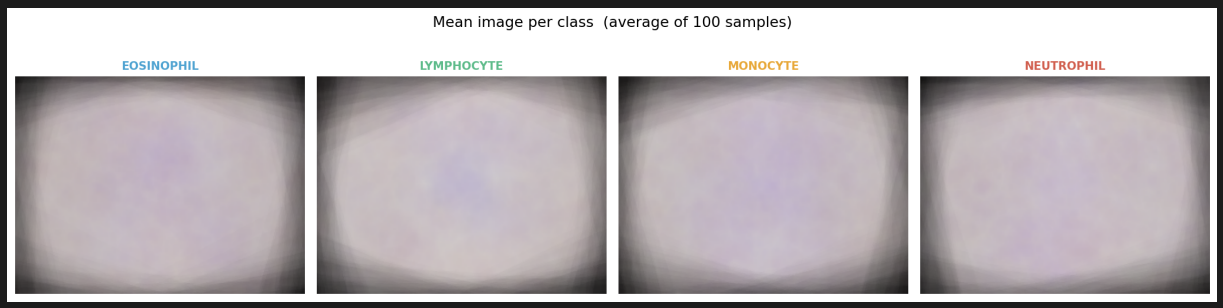
شکل ۳ توزیع سه کانال R، G، B رو برای هر کلاس نشون می‌ده. یه تفاوت جالب اینه که Eosinophil بخاطر رنگ‌آمیزی eosin یه غلبه قرمز-صورتی مشخص داره، در حالی که Neutrophil توزیع آبی بیشتری نشون می‌ده. درنتیجه رنگ به تنهایی می‌تونه یه سیگنال قوی برای تفکیک کلاس‌ها باشه، که این برای مدل یه مزیت حسابه.



شکل ۳: توزیع شدت کانال‌های RGB برای ۸۰ نمونه از هر کلاس

۵.۳.۲. تصویر میانگین هر کلاس

شکل ۴ میانگین 100 تصویر از هر کلاس رو نشون می‌ده. نکته مهم اینه که این تصاویر میانگین هنوزم تیز و قابل تشخیصن — انگار شکل کلی سلول توی همه نمونه‌های یه کلاس خیلی مشابهه. این یعنی هر کلاس از نظر ظاهری نسبتاً همگنه و مدل می‌تونه الگوهای پایدار یاد بگیره.



شکل ۴: تصویر میانگین هر کلاس (میانگین ۱۰۰ نمونه)

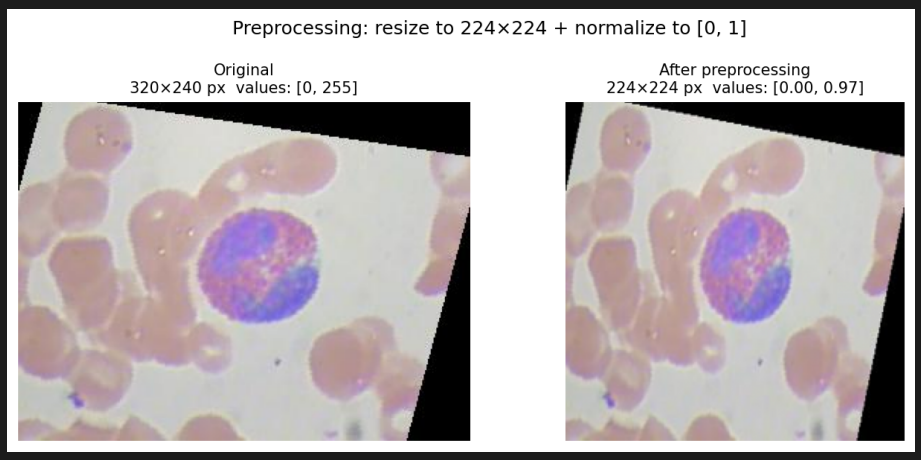
۳. پیش‌پردازش و افزایش داده

۱.۳. تغییر اندازه و نرمال‌سازی

همه تصاویر موقع بارگذاری به 224×224 پیکسل تغییر اندازه داده می‌شن. بعدش مقادیر پیکسل از بازه $[0, 255]$ به $[0, 1]$ نرمال می‌شن:

$$x_{\text{norm}} = \frac{x}{255}$$

حالا برای مدل‌های از پیش آموزش‌دیده یه فرق مهم وجود داره — به‌جای این تقسیم ساده، از تابع preprocess_input مخصوص همون مدل استفاده می‌شه که نرمال‌سازی رو متناسب با وزن‌های ImageNet اعمال می‌کنه. این یه نکته ظریفه که اگه رعایت نشه، مدل از پیش آموزش‌دیده خوب کار نمی‌کنه. شکل ۵ اثر این پیش‌پردازش رو نشون می‌ده.



شکل ۵: تصویر اصلی در مقابل تصویر پیش‌پردازش‌شده (تغییر اندازه به 224×224 و نرمال‌سازی)

۲.۳. تقسیم داده: آموزش و اعتبارسنجی

بخش TRAIN با نسبت 80/20 تقسیم می‌شه — یعنی 7,968 تصویر برای آموزش و 1,989 تصویر برای اعتبارسنجی. این تقسیم تصادفیه و با seed=42 انجام می‌شه که نتایج بازتولیدپذیر باشن.

بخش	تعداد تصاویر	درصد	کاربرد
آموزش (train)	7,968	80%	به‌روزرسانی وزن‌های مدل
اعتبارسنجی (validation)	1,989	20%	انتخاب بهترین checkpoint
آزمون (test)	2,487	—	ارزیابی نهایی مستقل

جدول ۲: تقسیم داده‌ها برای آموزش، اعتبارسنجی و آزمون

۳.۳. ساخت Data Generator

برای هر سه بخش یه generator جداگانه با ImageDataGenerator کرس ساخته می‌شه. generator آموزش شامل افزایش داده و shuffle هست، ولی generatorهای اعتبارسنجی و آزمون فقط نرمال‌سازی دارن و shuffle نمی‌شن.

چرا این تفاوت مهمه؟ خب، توی train اگه shuffle نباشه ممکنه یه batch کامل فقط از یه کلاس بشه که این گرادیان‌های به‌روزرسانی رو بی‌ثبات می‌کنه و آموزش ضعیفی به مدل می‌ده. از طرف دیگه، validation و test نباید shuffle بشن چون می‌خوایم نتایج ارزیابی هر بار دقیقاً یکسون باشن و بشه پیش‌بینی‌ها رو دقیقاً به تصاویر متناظرشون نسبت داد. درنتیجه این دو تا تنظیم مکمل همن، نه اشتباه.

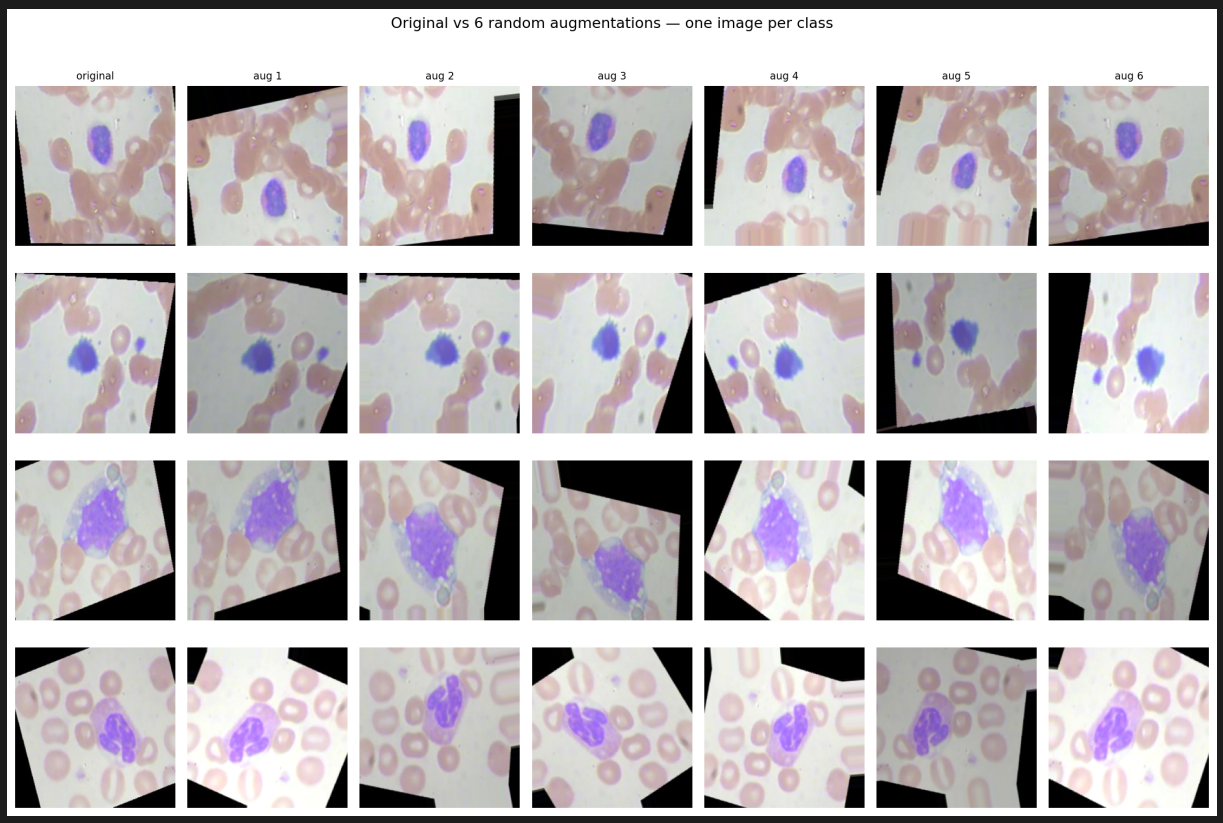
۴.۳. افزایش داده

افزایش داده (Data Augmentation) فقط روی مجموعه آموزشی اعمال می‌شه — این یه نکته اساسیه که نباید فراموش بشه. هر تبدیلی که انتخاب شده یه دلیل فیزیکی داره که از ماهیت تصاویر میکروسکوپی میاد:

تبدیل	مقدار	دلیل
چرخش	± 15	سلول‌ها در تمام جهات ظاهر می‌شوند
انعکاس افقی	فعال	تصویر آینه‌ای در میکروسکوپ معتبر است
انعکاس عمودی	فعال	جهت بالا/پایین در میکروسکوپ اهمیتی ندارد
جابجایی افقی/عمودی	$\pm 10\%$	سلول همیشه دقیقاً در مرکز نیست
زوم	$\pm 10\%$	تغییر بزرگنمایی میکروسکوپ
روشنایی	$\pm 20\%$	شدت رنگ‌آمیزی بین اسلایدها متفاوت است
حالت پر کردن	nearest	از ایجاد لبه‌های سیاه جلوگیری می‌کند

جدول ۳: تبدیل‌های افزایش داده و توجیه آن‌ها

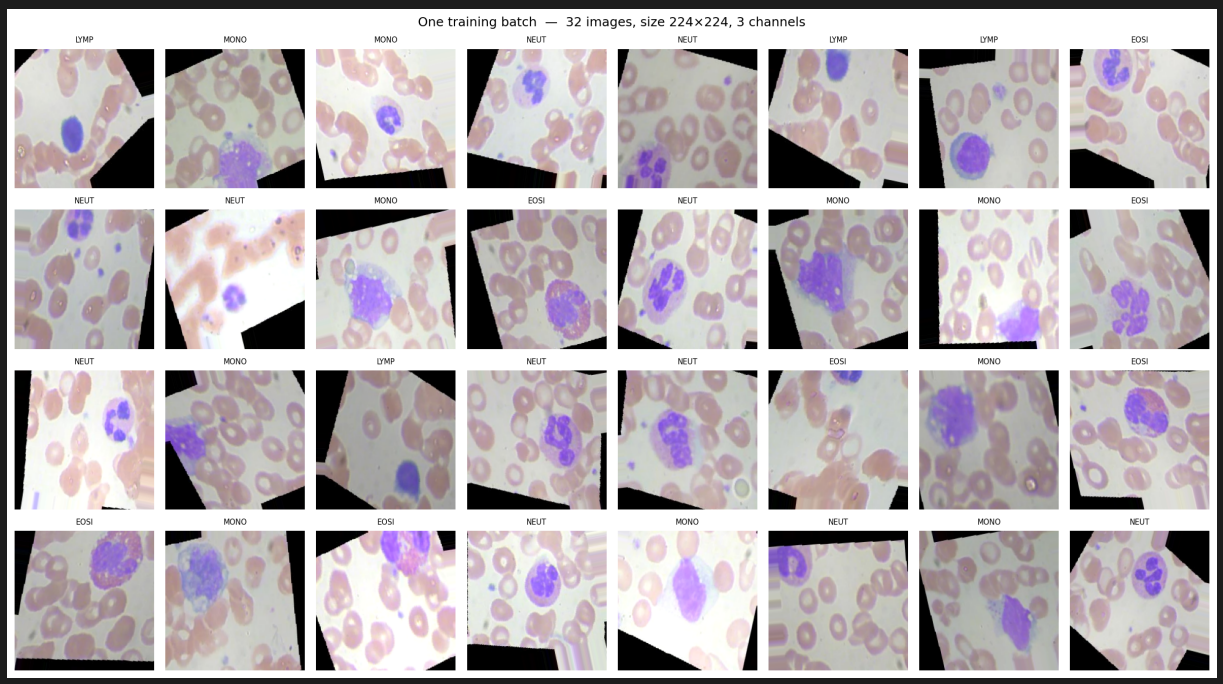
اینجوریه که مدل یاد می‌گیره به تغییرات غیرمعنادار حساس نباشه — انگار که هر بار یه نمونه جدید می‌بینه، در حالی که دیتاست اصلی همونه. شکل ۶ نمونه‌ای از تصویر اصلی در کنار ۶ نسخه افزوده‌شده رو برای هر کلاس نشون می‌ده.



شکل ۶: تصویر اصلی در مقابل ۶ نسخه افزوده‌شده برای یک نمونه از هر کلاس

۵.۳. نمایش یک batch از داده‌های آموزشی

شکل ۷ یه batch نمونه از generator آموزشی رو نشون می‌ده. هر batch شامل ۳۲ تصویر با شکل $(32, 224, 224, 3)$ و برچسب‌های one-hot با شکل $(32, 4)$ هست — پس ورودی و خروجی مدل دقیقاً همین ابعادن.



شکل ۷: یک batch از داده‌های آموزشی — ۳۲ تصویر 224×224 با برچسب کلاس

۴. مدل CNN از پایه

۱.۴ معماری

خب، اولین کاری که کردیم اینه که یه CNN ساده از صفر نوشتیم — بدون وزن از پیش آموزش‌دیده، بدون Dropout، بدون BatchNorm، بدون هیچ افزایش داده‌ای. هدف اینه که یه baseline داشته باشیم که بفهمیم فازهای بعدی چقدر بهترش کردن.

معماری شبکه چهار بلوک کانولوشنی داره که هر کدوم از یه Conv2D با ReLU و یه MaxPooling تشکیل شدن. بعد از این چهار بلوک، یه Flatten داریم و دو لایه‌ی Dense — اولی با ReLU و دومی با Softmax که چهار کلاس خروجی میده:

$$\text{ورودی}\ (128 \times 128 \times 3) \xrightarrow{\text{Conv۳۲+Pool}} \xrightarrow{\text{Conv۶۴+Pool}} \xrightarrow{\text{Conv۱۲۸+Pool}} \xrightarrow{\text{Conv۱۲۸+Pool}} \xrightarrow{\text{Flatten}} \xrightarrow{\text{Dense۶۴}} \xrightarrow{\text{Dense۴}}$$

لایه	نوع	شکل خروجی	تعداد پارامتر
ورودی	Input	$128 \times 128 \times 3$	–
بلوک ۱	Conv2D(32)	$128 \times 128 \times 32$	896
	MaxPool	$64 \times 64 \times 32$	–
بلوک ۲	Conv2D(64)	$64 \times 64 \times 64$	18,496
	MaxPool	$32 \times 32 \times 64$	–
بلوک ۳	Conv2D(128)	$32 \times 32 \times 128$	73,856
	MaxPool	$16 \times 16 \times 128$	–
بلوک ۴	Conv2D(128)	$16 \times 16 \times 128$	147,584
	MaxPool	$8 \times 8 \times 128$	–
Flatten	–	8192	–
Dense(64)	ReLU	64	524,352
Dense(4)	Softmax	4	260
جمع پارامترها			**765,444**

جدول ۴: معماری Baseline CNN — کمتر از یک میلیون پارامتر

درواقع کل شبکه 765,444 پارامتر قابل آموزش داره که خوشبختانه زیر سقف یک میلیون پارامتر می‌مونه.

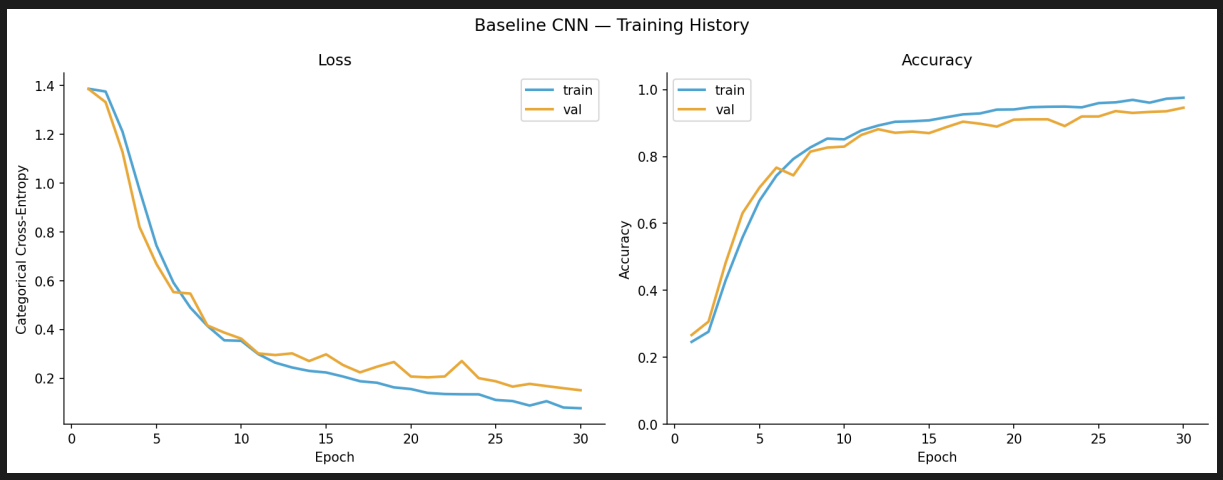
۲.۴. تنظیمات آموزش

تنظیم	مقدار
بهینه‌ساز	Adam
نرخ یادگیری	0.0001
تابع هزینه	Categorical Cross-Entropy
اندازه batch	32
تعداد epoch	30
اندازه تصویر	128×128 پیکسل
نرمال‌سازی	تقسیم بر 255
ذخیره بهترین مدل	ModelCheckpoint بر اساس val_accuracy

جدول ۵: تنظیمات آموزش Baseline CNN

۳.۴. نمودار آموزش

شکل ۸ روند loss و دقت رو در طول 30 epoch برای هر دو مجموعه‌ی آموزش و اعتبارسنجی نشون می‌ده.



شکل ۸: نمودار loss و دقت در طول آموزش — Baseline CNN

چیزی که توی این نمودار جالبه اینه که هر دو منحنی آموزش و اعتبارسنجی کاملاً با هم پیش رفتن و تا epoch آخر هم val accuracy داشت بهتر می‌شد — یعنی مدل هنوز داشت یاد می‌گرفت. دقت آموزشی در پایان به 97.55% رسید و دقت اعتبارسنجی به 94.57%، که اختلافشون فقط 2.98% هست. پس overfitting شدیدی اتفاق نیفتاده.

۴.۴. ارزیابی روی مجموعه آزمون

بهترین مدل ذخیره‌شده — که بر اساس val_accuracy انتخاب شده — روی مجموعه‌ی آزمون مستقل ارزیابی شد:

معیار	مقدار
دقت (Accuracy)	79.53%
دقت ماکرو (Precision)	81.78%
فراخوانی ماکرو (Recall)	79.55%
F1-score ماکرو	80.19%
AUC ماکرو	95.52%

جدول ۶: نتایج ارزیابی Baseline CNN روی مجموعه آزمون

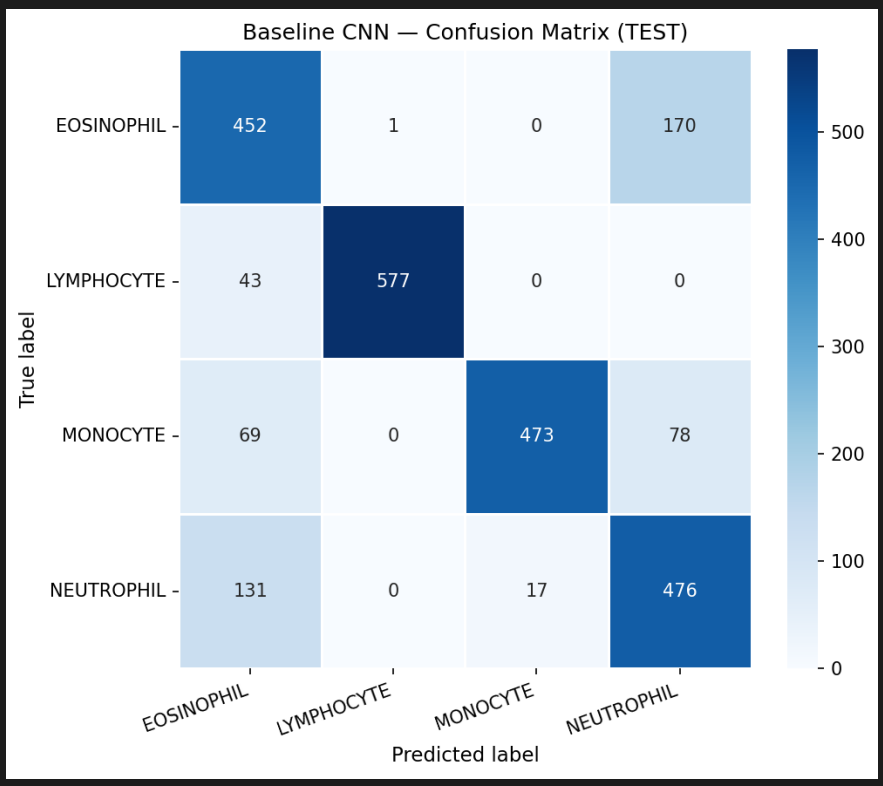
عملکرد مدل به تفکیک هر کلاس در جدول زیر آمده:

کلاس	Precision	Recall	F1	AUC
Eosinophil	0.65	0.73	0.69	0.9150
Lymphocyte	1.00	0.93	0.96	0.9996
Monocyte	0.97	0.76	0.85	0.9848
Neutrophil	0.66	0.76	0.71	0.9215

جدول ۷: عملکرد به تفکیک کلاس — Baseline CNN

۵.۴. ماتریس درهم‌ریختگی

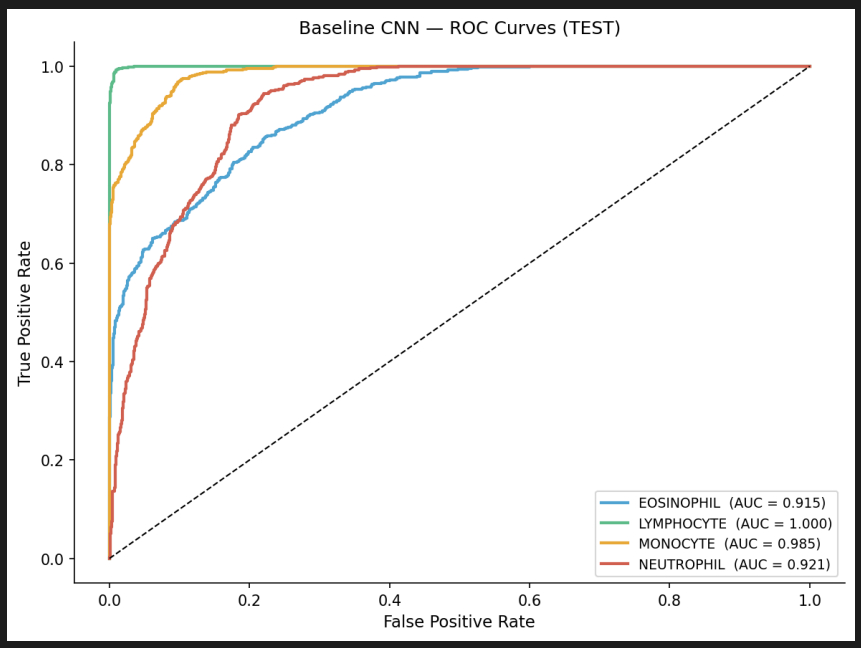
شکل ۹ ماتریس درهم‌ریختگی مدل روی مجموعه آزمون را نشان می‌دهد.



شکل ۹: ماتریس درهم‌ریختگی Baseline CNN روی مجموعه آزمون

۶.۴. منحنی ROC

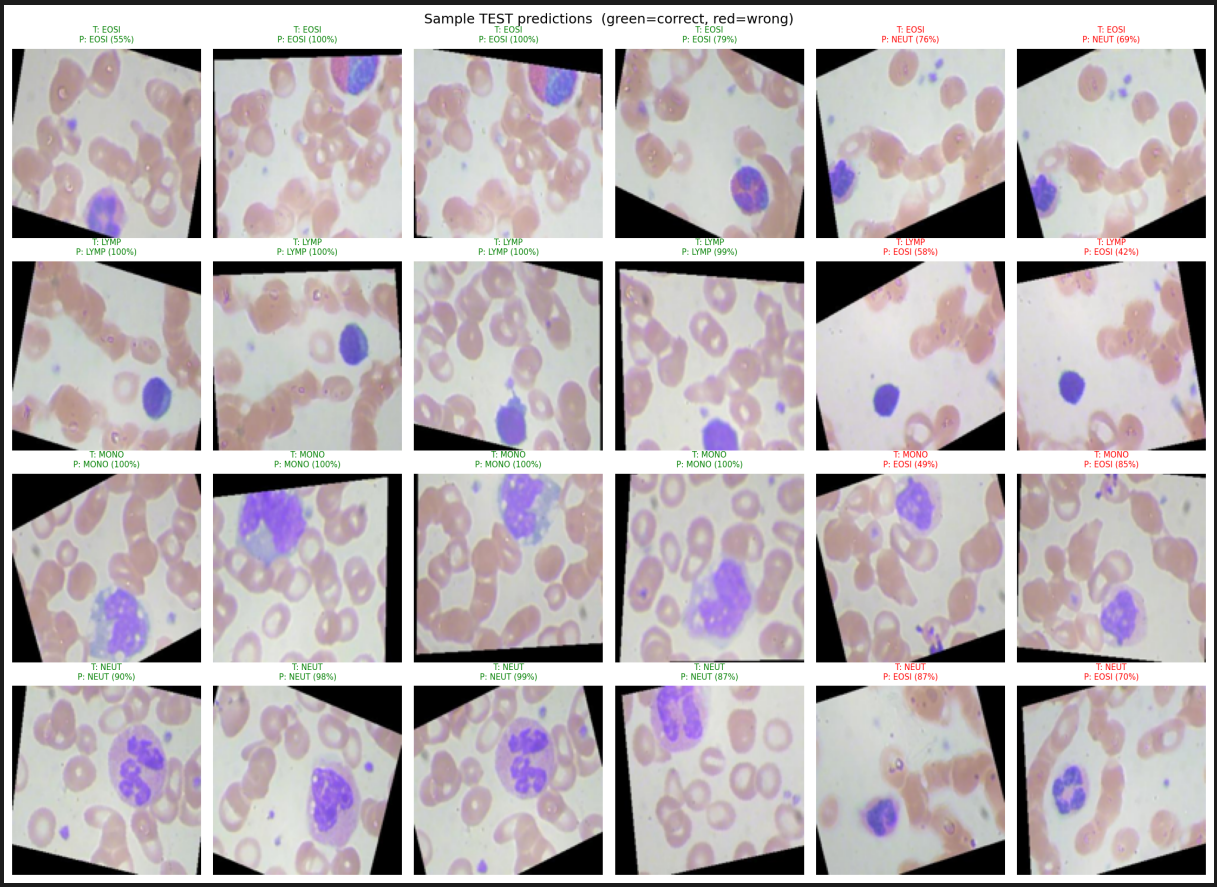
شکل ۱۰ منحنی‌های ROC به تفکیک هر کلاس رو نشون می‌ده. AUC مربوط به Lymphocyte تقریباً 1.00 هست که انگار مدل این کلاس رو با چشم‌بسته هم می‌شناسه — جداسازیش از بقیه کلاس‌ها کاملاً قطعیه.



شکل ۱۰: منحنی ROC برای چهار کلاس — Baseline CNN

۷.۴. نمونه پیش‌بینی‌ها

شکل ۱۱ چند نمونه از پیش‌بینی‌های مدل رو نشون می‌ده. عنوان سبز یعنی پیش‌بینی درست بوده، قرمز یعنی اشتباه.



شکل ۱۱: نمونه‌هایی از پیش‌بینی‌های Baseline CNN روی مجموعه آزمون

۸.۴. تحلیل نتایج

شکاف اعتبارسنجی–آزمون: یه چیزی که توی نتایج آدم رو یه لحظه نگران می‌کنه اینه که دقت اعتبارسنجی 94.57% بود ولی دقت آزمون افتاد به 79.53% — یعنی یه شکاف حدود 15%. ولی خب، این رو نباید سریع بچسبیم به overfitting، چون منحنی‌های آموزش و اعتبارسنجی کاملاً با هم رفتن و از هم جدا نشدن. دلیل احتمالی این شکاف اینه که ImageDataGenerator وقتی validation_split انجام می‌ده، آخرین 20% فایل‌ها رو بر اساس ترتیب الفبایی اسم فایل برمی‌داره — نه تصادفی. بخاطر همین، مجموعه اعتبارسنجی ممکنه توزیعش با توزیع واقعی مجموعه آزمون مستقل یکی نباشه و اعداد بهتری نشون بده.

بهترین و بدترین کلاس‌ها: Lymphocyte با F1 برابر 0.96 و AUC برابر 0.9996 قهرمان بلامنازع این مدله. دلیلش هم واضحه — این کلاس هسته‌ای بزرگ، گرد و تیره داره که از نظر بصری کاملاً متمایزه و مدل راحت تشخیصش می‌ده. در مقابل، Eosinophil و Neutrophil با F1 حدود 0.69 و 0.71 بدترین عملکرد رو داشتن. این دو تا هر دو گرانولوسیت هستن، هسته‌ی چندلوبه‌ای مشابه دارن، و تنها تفاوت جدیشون رنگ دانه‌های داخل سلوله — که در تصاویر 128×128 بخشی از این اطلاعات گم می‌شه. درنتیجه مدل بین این دو کلاس گیج می‌شه.

نتیجه‌گیری: این مدل پایه با 765,444 پارامتر و بدون هیچ تکنیک اضافه‌ای، به دقت آزمون 79.53% و AUC ماکرو 95.52% رسید. این اعداد حالا مبنای ما هستن — پس هر چیزی که توی فازهای بعدی با افزایش داده، BatchNorm، Dropout یا transfer learning بسازیم، باید از این‌ها بهتر باشه تا بشه گفت واقعاً پیشرفت داشتیم.

۵. مدل CNN بهبودیافته

۱.۵. تغییرات نسبت به مدل پایه

معماری فاز سوم دقیقاً همون ۳ بلوک کانولوشنی فاز دوم رو نگه داشت — یعنی 32، 64، و 128 فیلتر با یه لایه‌ی متراکم 256 نورونی — فقط سه چیز بهش اضافه شد که ببینیم چه اتفاقی می‌افته.

اول، Data Augmentation که پنج تبدیل تصادفی فقط موقع آموزش اعمال می‌شه: RandomFlip برای چرخش افقی و عمودی، RandomRotation تا 15 درجه، RandomZoom تا 10%، RandomTranslation تا 10% در هر جهت، و RandomBrightness با دامنه‌ی ±20%. ایده اینه که مدل به جزئیات بی‌ربط وابسته نشه.

دوم، Batch Normalization که بعد از هر لایه‌ی کانولوشن و قبل از تابع فعال‌سازی اضافه شد تا آمار ورودی هر لایه رو تنظیم کنه و همگرایی باثبات‌تری داشته باشیم. سوم هم Dropout با نرخ‌های مختلف بعد از لایه‌های متراکم که جلوی بیش‌برازش رو بگیره.

۲.۵. تنظیم فراپارامترها (Hyperparameter Tuning)

پنج آزمایش طراحی شد. تو همه‌شون از Adam با نرخ یادگیری 1e-4 و EarlyStopping بر اساس val_loss با صبر ۱۰ دوره استفاده شد. جدول ۸ تنظیمات و نتایج هر آزمایش رو نشون می‌ده.

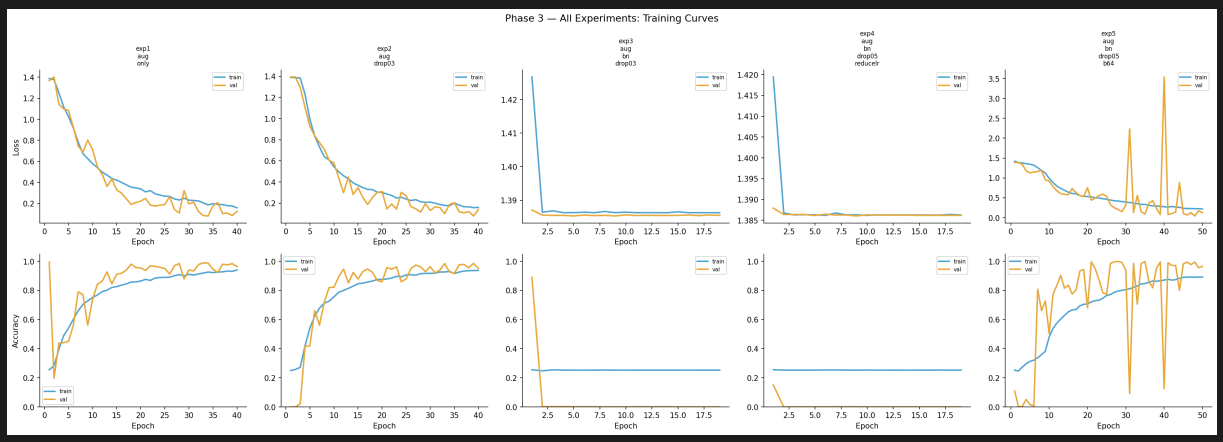
جدول ۸: آزمایش‌های فاز سوم و نتایج بهترین دوره

آزمایش	BN	Dropout	Batch	ReduceLR	دوره	Val Loss	دقت اعتبارسنجی
exp1 (تنها افزایش داده)	خیر	0.0	32	خیر	34	0.0816	98.79%
exp2 (افزایش داده + Dropout 0.3)	خیر	0.3	32	خیر	39	0.0788	98.59%
exp3 (Dropout 0.3 + BN)	بله	0.3	32	خیر	19	1.3853	0.05%
exp (ReduceLR + Dropout 0.5 + BN)	بله	0.5	32	بله	19	1.3860	0.05%
exp5 (batch=64 + Dropout 0.5 + BN)	بله	0.5	64	بله	48	0.0428	99.50%

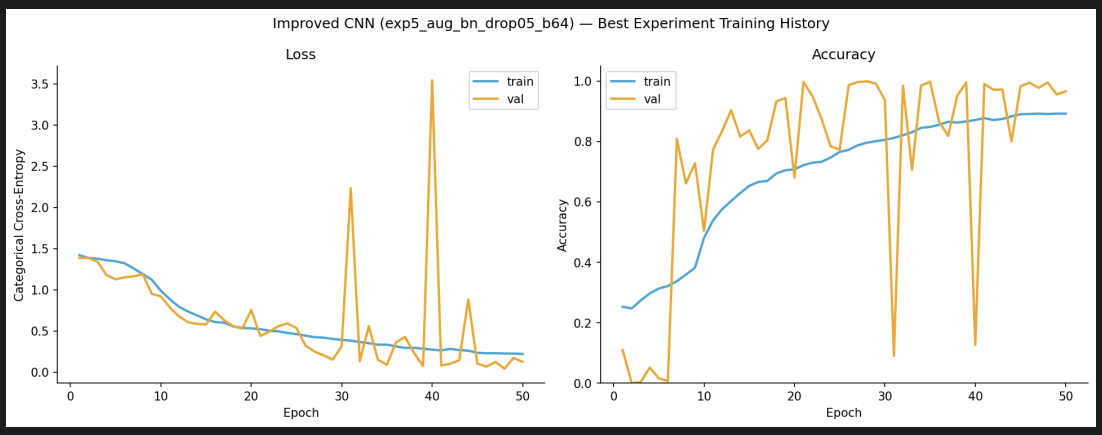
ناهنجاری exp3 و exp4 — شبکه‌ی مرده (Dead Network): خب، exp3 و exp4 از همون اول مُردن — نه استعاره، واقعاً. خطا از دوره‌ی اول روی 1.386 ≈ ln(4) قفل شد که انگار شبکه داره تصادفی حدس می‌زنه برای ۴ کلاس، و دقت اعتبارسنجی هم عملاً صفر بود. دلیلش اینه که BN با batch=32 توی TF 2.21 کار نمی‌کنه درست: وقتی دسته کوچیکه، آمار میانگین و واریانسی که BN محاسبه می‌کنه خیلی پرنویزه، و شبکه از همون ابتدا تو یه حالت تباهیده گیر می‌کنه که دیگه خروجی نداره. بخاطر همین وقتی توی exp5 اندازه‌ی دسته رو به 64 رسوندیم، مشکل حل شد.

۳.۵. نمودارهای آموزش

شکل ۱۲ منحنی‌های خطا و دقت همه‌ی آزمایش‌ها رو با هم نشون می‌ده — اون خطوط کاملاً تخت مربوط به exp3 و exp4 هستن که مشکل شبکه‌ی مرده رو خیلی واضح آشکار می‌کنن. شکل ۱۳ هم منحنی‌های بهترین مدل یعنی exp5 رو جداگانه نشون می‌ده.



شکل ۱۲: منحنی‌های آموزش و اعتبارسنجی همه‌ی آزمایش‌های فاز سوم



شکل ۱۳: منحنی‌های آموزش بهترین مدل (exp5) — کاهش یکنواخت val_loss تا دوره‌ی 48

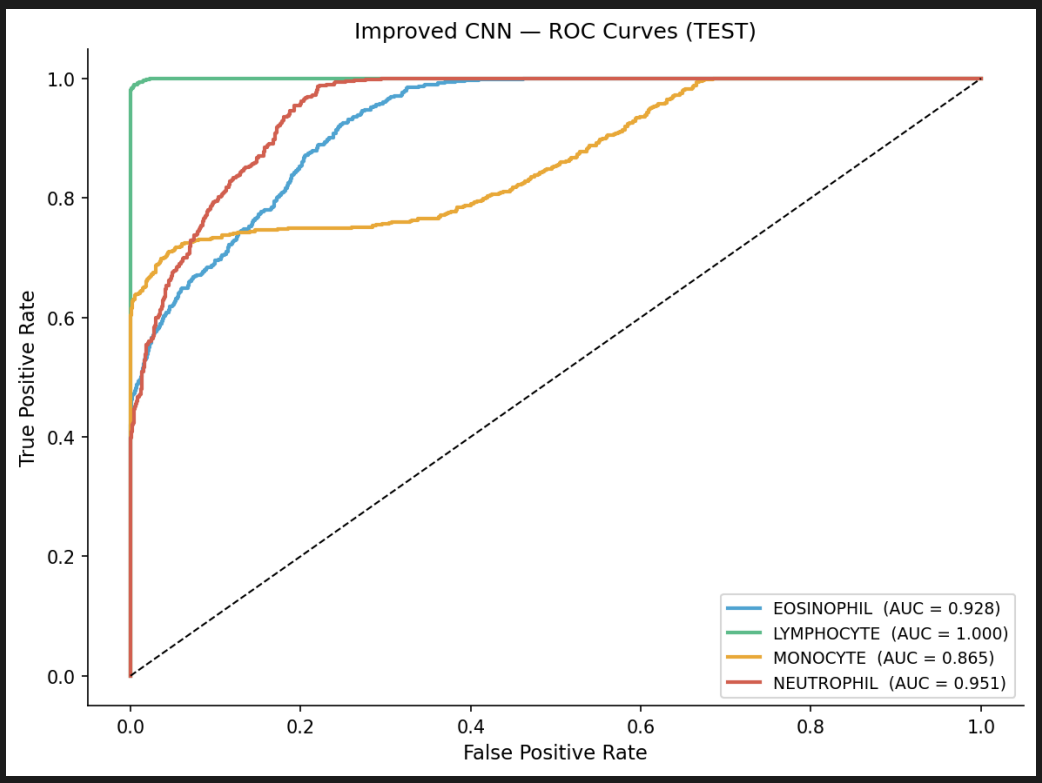
۴.۵. نتایج کامل مدل نهایی روی مجموعه‌ی آزمون

بهترین مدل یعنی exp5_aug_bn_drop05_b64 روی 2,487 تصویر مجموعه‌ی آزمون مستقل تست شد. جدول ۹ نتایج کامل رو نشون می‌ده.

جدول ۹: نتایج کامل مدل نهایی فاز سوم (exp5) روی مجموعه‌ی آزمون

کلاس	دقت (Precision)	بازخوانی (Recall)	F1	AUC
EOSINOPHIL	0.74	0.66	0.69	0.9277
LYMPHOCYTE	1.00	0.55	0.71	0.9998
MONOCYTE	0.79	0.72	0.76	0.8650
NEUTROPHIL	0.60	0.97	0.74	0.9509
میانگین ماکرو	0.78	0.73	0.73	0.9359

درواقع دقت کل روی آزمون 72.38% شد با خطای 1.4701 — که یه کم بعداً توضیح می‌دیم چرا این عدد اینقدر پایینه با وجود اینکه val acc خیلی بالا بود.



شکل ۱۴: منحنی‌های ROC مدل نهایی فاز سوم

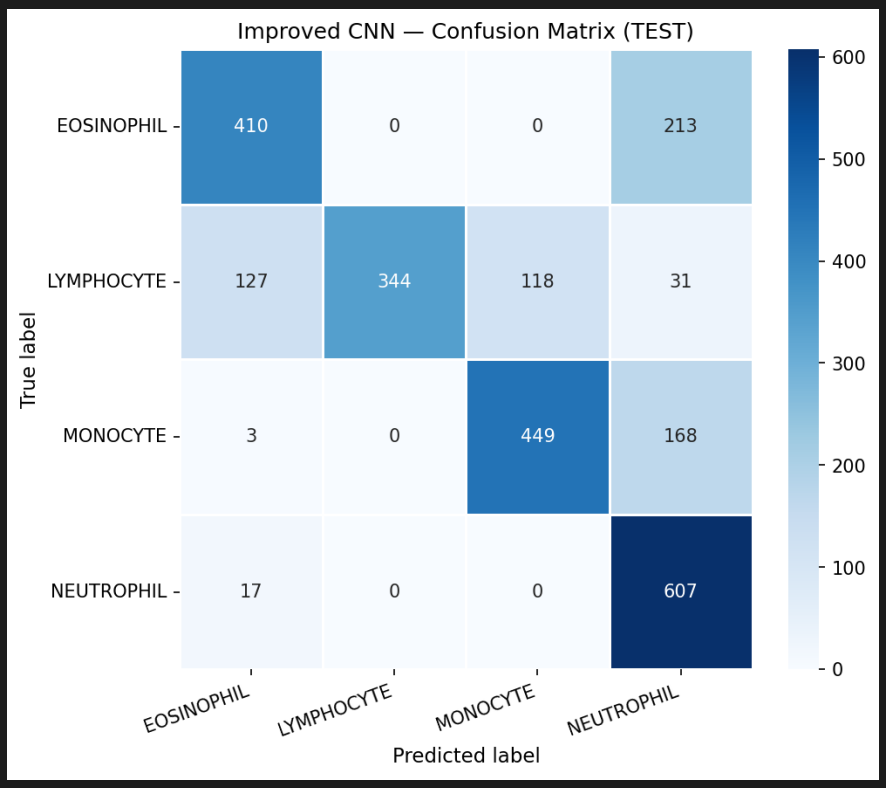
۵.۵. مقایسه‌ی عددی با فاز دوم

جدول ۱۰ مقایسه‌ی مستقیم بهترین مدل فاز سوم با مدل پایه‌ی فاز دوم رو نشون می‌ده.

جدول ۱۰: مقایسه‌ی مدل پایه (فاز دوم) با بهترین مدل فاز سوم

معیار	فاز دوم (مدل پایه)	فاز سوم (exp5)	تغییر
دقت آزمون	79.53%	72.38%	-7.15 pp
دقت اعتبارسنجی	94.57%	99.50%	+4.93 pp
F1 ماکرو	0.80	0.73	-0.07
AUC ماکرو	0.9552	0.9359	-0.0193

۶.۵. تحلیل ماتریس درهم‌ریختگی

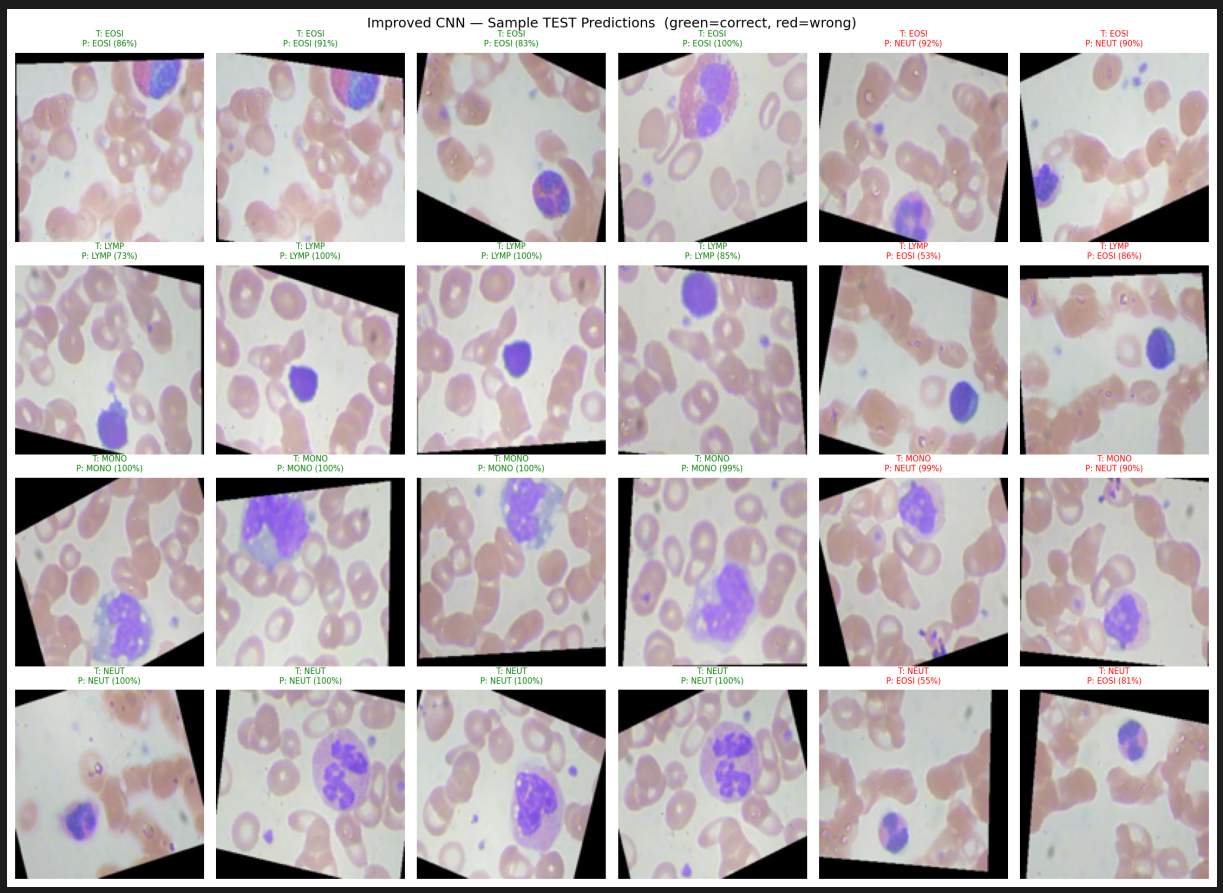


شکل ۱۵: ماتریس درهم‌ریختگی مدل نهایی فاز سوم روی مجموعه‌ی آزمون

یه نگاه به ماتریس درهم‌ریختگی چند چیز جالب نشون می‌ده. اول اینکه NEUTROPHIL بیش از حد تشخیص داده می‌شه — Recall=0.97 خوبه اما Precision=0.60 یعنی مدل نمونه‌های زیادی از کلاس‌های دیگه، خصوصاً LYMPHOCYTE و EOSINOPHIL، رو اشتباه به این کلاس نسبت می‌ده.

برعکسش رو توی LYMPHOCYTE می‌بینیم: Precision=1.00 یعنی هر چی مدل LYMPHOCYTE خونده درسته، ولی Recall=0.55 یعنی حدود 45% از نمونه‌های واقعی این کلاس از دست رفتن و احتمالاً به NEUTROPHIL نسبت داده شدن. EOSINOPHIL هم با Recall=0.66 وضع مشابهی داره — دلیلش اینه که Eosinophil و Neutrophil هر دو گرانولوسیت با هسته‌ی چندلوبه‌ای هستن و از نظر ظاهری شبیه همن، پس مدل گیج می‌شه. بهترین عملکرد بین کلاس‌های مشکل‌دار مال MONOCYTE بود که F1=0.76 گرفت.

۷.۵. نمونه پیش‌بینی‌های مدل



شکل ۱۶: نمونه‌ای از پیش‌بینی‌های مدل نهایی فاز سوم روی تصاویر آزمون — کادر سبز نشون‌دهنده‌ی پیش‌بینی درست و کادر قرمز نشون‌دهنده‌ی پیش‌بینی اشتباهه

شکل ؟؟ چند نمونه از پیش‌بینی‌های مدل رو نشون می‌ده. انگار می‌شه دید که اشتباهات اکثراً بین کلاس‌هایی هستن که از نظر مورفولوژی شبیه همن — دقیقاً همون چیزی که ماتریس درهم‌ریختگی هم نشون داد.

۸.۵. علت کاهش عملکرد نسبت به فاز دوم

یه چیز عجیب اینجاست: دقت اعتبارسنجی از 94.57% به 99.50% رفت بالا، ولی دقت آزمون از 79.53% افتاد به 72.38%. چطور ممکنه؟ دو تا دلیل اصلی داره.

اول، Distribution Shift — پوشه‌های TRAIN و TEST از نظر ویژگی‌های تصویری با هم فرق دارن: روشنایی، کنتراست، رنگ‌بندی پس‌زمینه. مدل فاز سوم با اون augmentation قوی‌تر خیلی خوب روی توزیع آموزشی جا افتاد، ولی بخاطر همین وقتی با توزیع متفاوت آزمون روبرو شد، تعمیم‌پذیریش کمتر شد. یعنی augmentation طراحی‌شده برای تنوع درون‌توزیعیه، نه پوشش تفاوت‌های بین‌توزیعی.

دوم، معیار اعتبارسنجی گمراه‌کننده‌ست. زیرمجموعه‌ی اعتبارسنجی از همون پوشه‌ی TRAIN جدا می‌شه، پس همون توزیع رو داره. حالا وقتی مدل روی اون 99.50% می‌گیره، این عدد خوش‌بینانه به نظر می‌رسه چون آزمون واقعی توزیع متفاوتی داره. پس val acc بالا به‌تنهایی تضمینی برای test acc بالا نیست، خصوصاً وقتی توزیع‌ها با هم فرق دارن.

درنتیجه، اگرچه BN، Dropout، و augmentation به آموزش کمک کردن و شبکه رو روی داده‌های خودش

بهتر کردن، مشکل اصلی — یعنی همون شکاف توزیع بین TRAIN و TEST — با این روش‌ها حل نشد. انتظار می‌ره مدل‌های ازپیش‌آموزش‌دیده در فازهای بعدی که نمایش‌های عمومی‌تری از تصویر یاد گرفتن، این شکاف رو کاهش بدن.

۶. تفسیر مدل

خب، یه سوال مهم اینه که مدل واقعاً به چی نگاه می‌کنه؟ برای جواب دادن به این سوال دو کار کردیم — اول رفتیم سراغ وزن فیلترهای کانولوشن، بعد از Grad-CAM استفاده کردیم که نشون بده موقع تشخیص هر سلول، مدل کجای تصویر رو می‌بینه. مدل مورد استفاده همون exp5 از فاز سومه که BN + Dropout=0.5 با batch=64 داشت و دقت آزمونش 72.38% بود.

۱.۶. فیلترهای کانولوشن

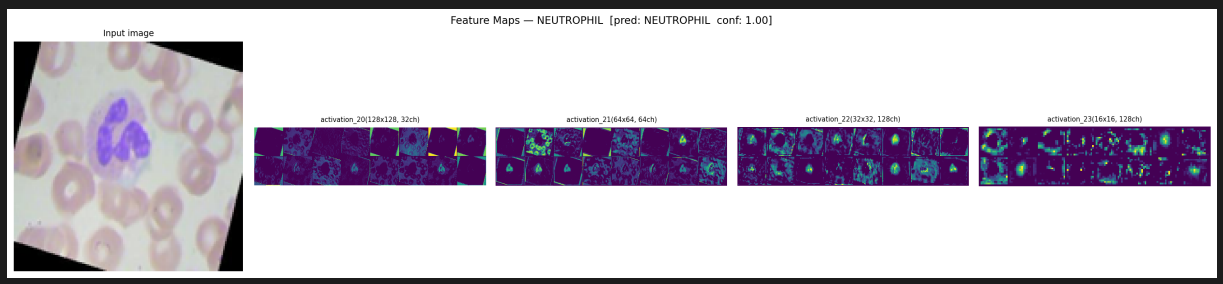
شکل ۱۶ وزن فیلترهای سه لایه رو نشون می‌ده: لایه‌ی اول (conv2d با 32 فیلتر)، لایه‌ی دوم (conv2d_1 با 64 فیلتر)، و لایه‌ی چهارم (conv2d_3 با 128 فیلتر).

شکل ۱۷: وزن فیلترهای سه لایه‌ی کانولوشن — از چپ: لایه‌ی اول، دوم، چهارم

در لایه‌ی اول هر فیلتر مستقیماً روی پیکسل‌های RGB کار می‌کنه، بخاطر همین وزن‌ها به شکل پچ‌های رنگی کوچیک دیده می‌شن. تنوع رنگیشون هم جالبه — از بنفش تا صورتی تا سبز — که انگار مدل از همون اول به طیف رنگی رنگ‌آمیزی H&E حساس شده. خب این منطقیه، چون همین رنگ‌ها در تصاویر خون بار اطلاعاتی دارن. در لایه‌های عمیق‌تر دیگه نمی‌شه فیلترها رو با رنگ تفسیر کرد، چون هر فیلتر روی ترکیبی از ده‌ها کانال قبلی عمل می‌کنه؛ اینجا بزرگی وزن‌ها رو خاکستری نشون دادیم. الگوهای تیره‌وروشنی که می‌بینی با فیلترهای لبه‌یاب و بافت‌یاب تطابق دارن.

۲.۶. نقشه‌های فعال‌سازی

برای هر کلاس یه نمونه‌ی درست با اطمینان بالا انتخاب کردیم و خروجی چهار لایه‌ی فعال‌سازی رو با هم مقایسه کردیم. شکل ۱۷ نتیجه رو برای کلاس NEUTROPHIL نشون می‌ده.



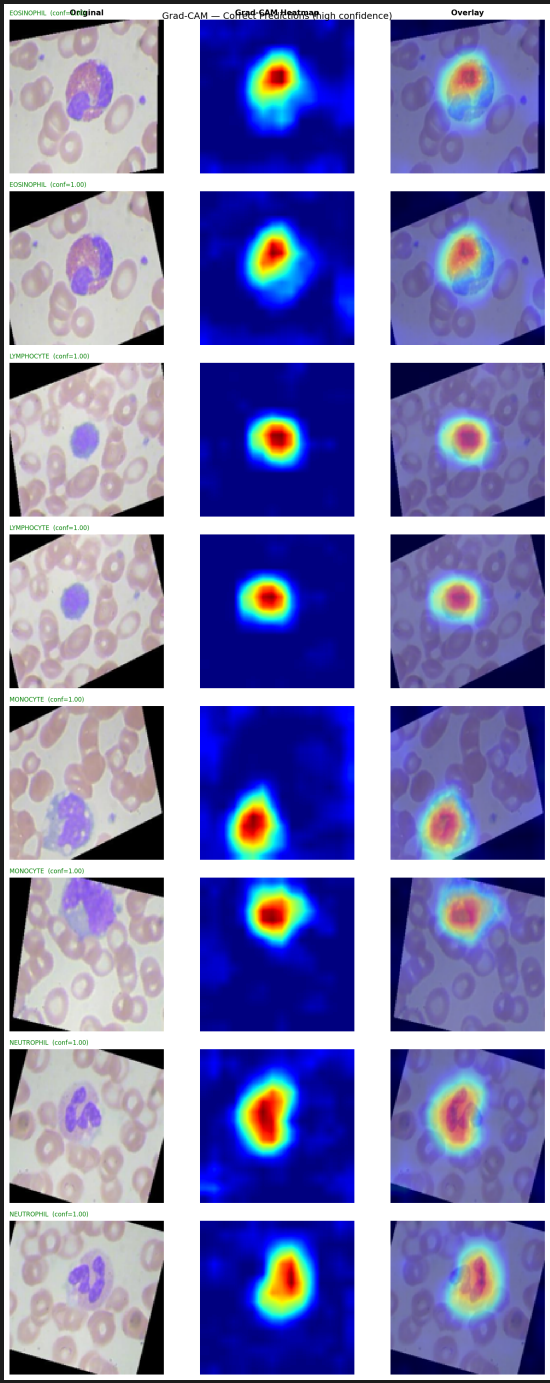
شکل ۱۸: نقشه‌های فعال‌سازی چهار مرحله برای یک نمونه‌ی NEUTROPHIL

در لایه‌ی اول (128×128) تقریباً همه‌ی نقشه‌ها فعالن و ساختار کلی تصویر — مرز سلول‌ها و رنگ پس‌زمینه — خوناست. با رفتن به لایه‌های بعدی، نقشه‌ها خلوت‌تر می‌شن و فقط نواحی خاص‌تری روشن می‌مونن. در لایه‌ی آخر (16×16) فقط چند نورون فعالن که به نواحی خیلی مشخصی مثل هسته‌ی سلول اشاره دارن. درواقع این رفتار دقیقاً همون چیزیه که انتظار داریم: هرچی عمیق‌تر بری، ویژگی انتزاعی‌تر و مکان‌یابی دقیق‌تر.

۳.۶. Grad-CAM

Grad-CAM از گرادیان امتیاز کلاس هدف نسبت به خروجی آخرین لایه‌ی کانولوشن استفاده می‌کنه تا نشون بده کدوم نواحی بیشترین سهم رو در اون تصمیم داشتن. نواحی قرمز و زرد در نقشه‌ی حرارتی جاهاییه که مدل بهشون توجه کرده.

شکل ۱۸ نتیجه رو برای هشت نمونه‌ی درست از مجموعه‌ی آزمون نشون می‌ده — دو نمونه از هر کلاس.



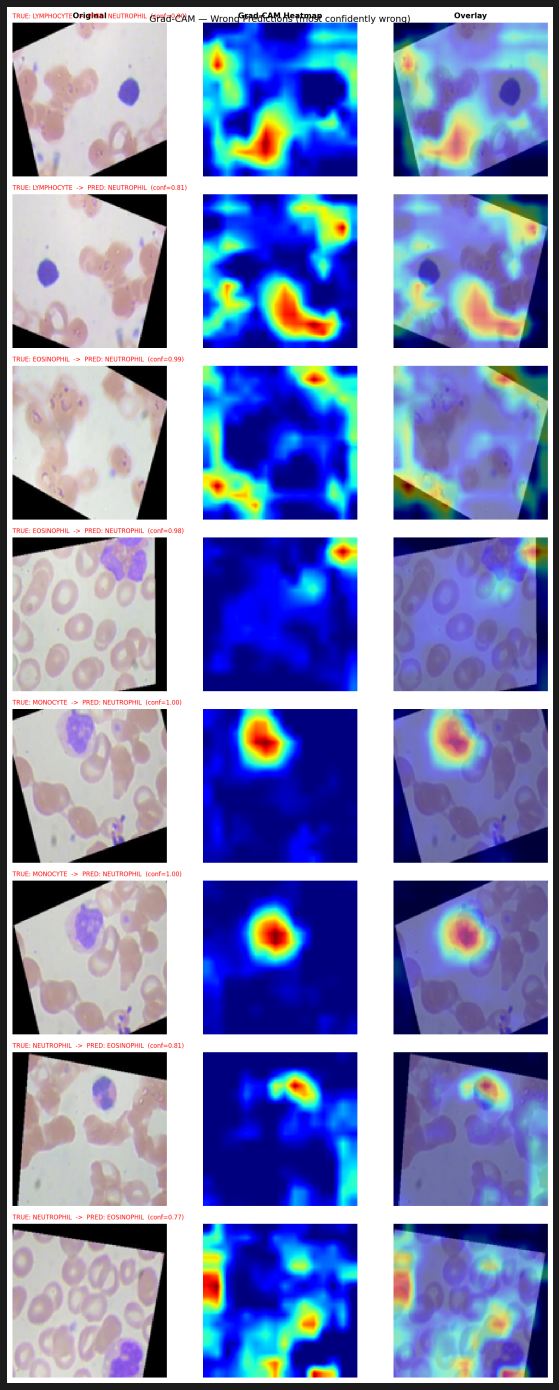
شکل ۱۹: Grad-CAM برای پیش‌بینی‌های درست — دو نمونه از هر کلاس

در همه‌ی این موارد نقطه‌ی داغ روی سلول اصلی قرار گرفته و نسبتاً فشرده‌ست. میانگین مساحت فعال (نواحی با heatmap > 0.5، روی 20 نمونه‌ی درست از هر کلاس) اینجوریه: EOSINOPHIL حدود 5.5%،

LYMPHOCYTE حدود 4.5%، MONOCYTE حدود 4.9%، و NEUTROPHIL حدود 5.8%. پس مدل تنها روی یه ناحیه‌ی کوچیک و مشخص از تصویر تمرکز می‌کنه، نه کل صحنه — که نشونه‌ی خوبیه.

۴.۶. مقایسه‌ی پیش‌بینی‌های درست و اشتباه

شکل ۱۹ همین تحلیل رو برای هشت نمونه‌ی اشتباه نشون می‌ده که مدل اونا رو با اطمینان بالا اشتباه تشخیص داده.

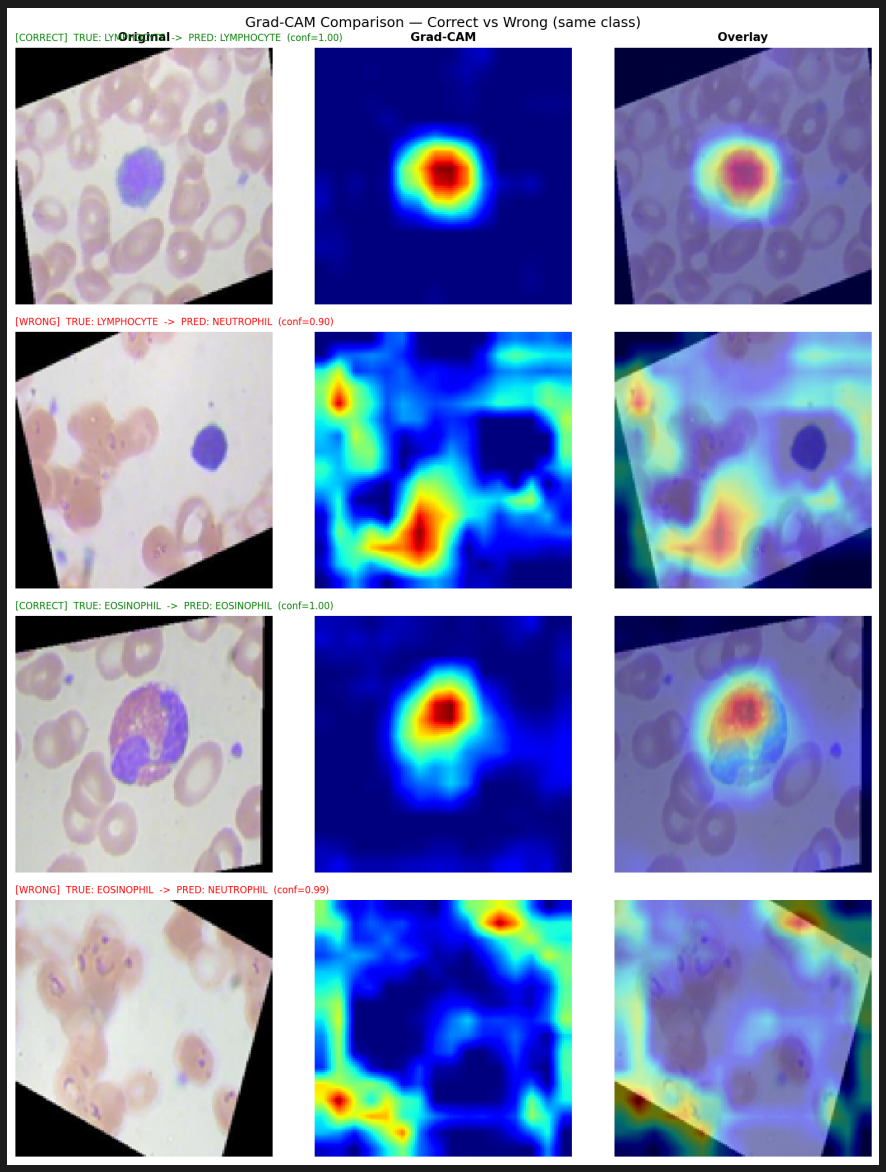


شکل ۲۰: Grad-CAM برای پیش‌بینی‌های اشتباه

تفاوت با نمونه‌های درست کاملاً مشهوده. نقشه‌ی حرارتی پراکنده‌ست و روی ناحیه‌ی مشخصی تمرکز نداره؛

در بعضی موارد مدل به پس‌زمینه یا سلول‌های قرمز اطراف توجه کرده نه به سلول اصلی. بخاطر همین وقتی مدل گیج می‌شه، به جای پیدا کردن ویژگی تمایزدهنده، به اطلاعات زمینه‌ای تکیه می‌کنه.

شکل ۲۰ این تفاوت رو برای دو کلاس به صورت مقایسه‌ای نشون می‌ده.



شکل ۲۱: مقایسه‌ی Grad-CAM برای پیش‌بینی درست و اشتباه از یک کلاس

در نمونه‌ی درست LYMPHOCYTE نقطه‌ی داغ دقیقاً روی هسته‌ی گرد و بزرگه — همون چیزیه که این سلول رو بصری متمایز می‌کنه. در نمونه‌ی اشتباه همون کلاس که به NEUTROPHIL تشخیص داده شده، توجه مدل پراکنده‌ست و به سلول‌های قرمز اطراف هم کشیده شده. برای EOSINOPHIL هم همین الگو تکرار می‌شه: نمونه‌ی درست روی هسته‌ی چندلوبه‌ای متمرکزه، نمونه‌ی اشتباه روی هیچ ناحیه‌ی مشخصی تمرکز نداره. درنتیجه این تحلیل دلیل Recall پایین LYMPHOCYTE (0.55) و EOSINOPHIL (0.66) در فاز سوم رو روشن می‌کنه: مدل در نمونه‌های دشوار نمی‌تونه ویژگی تمایزدهنده رو پیدا کنه.

۷. یادگیری انتقالی با MobileNetV2

در فازهای قبل بهترین دقت آزمون به 72.38% رسید. یه دلیل اصلیش اینه که توزیع مجموعه‌ی TRAIN با TEST جابجاست — ویژگی‌هایی که مدل از داده‌های آموزشی یاد گرفته با آنچه در آزمون می‌بینه کاملاً منطبق نیست. حالا مدل‌های ازپیش‌آموزش‌دیده روی ImageNet این مشکل رو کاهش می‌دن، چون از همون اول ویژگی‌های پایه‌ای قوی‌تری دارن که بهتر تعمیم می‌یابن.

برای این فاز از MobileNetV2 استفاده کردیم. خوبیش اینه که نسبت به گزینه‌های سنگین‌تر مثل ResNet50 یا EfficientNet پارامتر کمتری داره و روی GPUهای معمولی راحت‌تر آموزش می‌بینه؛ بدیش اینه که در مقابل اون مدل‌های بزرگ‌تر، سقف دقتش پایین‌تره — ولی برای این دیتاست کافیه.

۱.۷. تغییر لایه‌ی خروجی

MobileNetV2 در حالت استاندارد برای طبقه‌بندی 1000 کلاس ImageNet طراحی شده. برای تطبیق با مسئله‌ی ما که ۴ کلاسه، لایه‌ی top اصلی حذف شد (include_top=False) و یه سر طبقه‌بند سبک جایگزینش شد که از سه بخش تشکیل می‌شه. اول GlobalAveragePooling2D که خروجی سه‌بُعدی پشته‌ی کانولوشنی رو به یه بردار یک‌بُعدی تبدیل می‌کنه — در مقایسه با Flatten پارامتر کمتری داره و به موقعیت مکانی ویژگی‌ها حساسیت کمتری نشون می‌ده. بعدش یه لایه‌ی Dropout برای جلوگیری از overfitting سر طبقه‌بند. و در آخر Dense(4, softmax) که لایه‌ی خروجی با ۴ نورون برای چهار کلاس سلول خونیه.

یه نکته‌ی پیاده‌سازی مهم اینه که در forward pass، مدل پایه با آرگومان training=False فراخوانی می‌شه. این کار لایه‌های BatchNormalization درون MobileNetV2 رو در حالت استنتاج نگه می‌داره — یعنی از میانگین و واریانس تجمیعی استفاده می‌کنن نه از آمار دسته‌ی جاری. بدون این تنظیم، آموزش سر طبقه‌بند با پارامترهای BN فریز ناپایدار می‌شه.

۲.۷. پیش‌پردازش ورودی

MobileNetV2 انتظار داره پیکسل‌ها در بازه‌ی $[-1, +1]$ باشن، نه $[0, 1]$. بخاطر همین به جای تقسیم ساده بر 255، از تابع mobilenet_v2.preprocess_input استفاده شد که پیکسل‌ها رو از $[0, 255]$ به $[-1, +1]$ نگاشت می‌کنه. این تفاوت ظاهراً جزئیه ولی در عمل تأثیر قابل‌توجهی روی کیفیت آموزش داره — چون وزن‌های ImageNet با همین توزیع ورودی بهینه شدن و اگه این نگاشت رو نکنی، انگار داری با زبان اشتباه باهاشون حرف می‌زنی.

۳.۷. پروتکل آموزش دو مرحله‌ای

۱.۳.۷. مرحله‌ی اول — آموزش سر طبقه‌بند

در مرحله‌ی اول تمام لایه‌های MobileNetV2 منجمد (freeze) شدن و فقط لایه‌های سر طبقه‌بند آموزش دیدن. نرخ یادگیری 10^{-3} انتخاب شد — چون این لایه‌ها از صفر شروع می‌کنن و می‌شه با قدم‌های بزرگ‌تر آموزش داد. هدف اینه که سر طبقه‌بند اول یاد بگیره ویژگی‌های MobileNetV2 رو تفسیر کنه، بدون اینکه وزن‌های ازپیش‌آموزش‌دیده رو خراب کنه.

۲.۳.۷. مرحله‌ی دوم — Fine-tuning لایه‌های بالایی

پس از همگرایی مرحله‌ی اول، چند لایه‌ی آخر backbone از حالت منجمد خارج (unfreeze) شدن و آموزش با نرخ یادگیری خیلی کوچیک‌تر (10^{-5}) ادامه یافت. دلیل انتخاب این نرخ یادگیری کوچیک اینه که وزن‌های ImageNet در این لایه‌ها از قبل خوبن — فقط نیاز به تطبیق ظریف با دامنه‌ی تصاویر خون داریم. اگه نرخ یادگیری بالا باشه، این وزن‌ها خراب می‌شن و دقت به سطح تصادفی سقوط می‌کنه.

۴.۷. تنظیم ابرپارامترها

کلاً چهار آزمایش انجام شد — دو تا در مرحله‌ی اول با Dropout متفاوت، دو تا در مرحله‌ی دوم با تعداد لایه‌های unfreeze متفاوت.

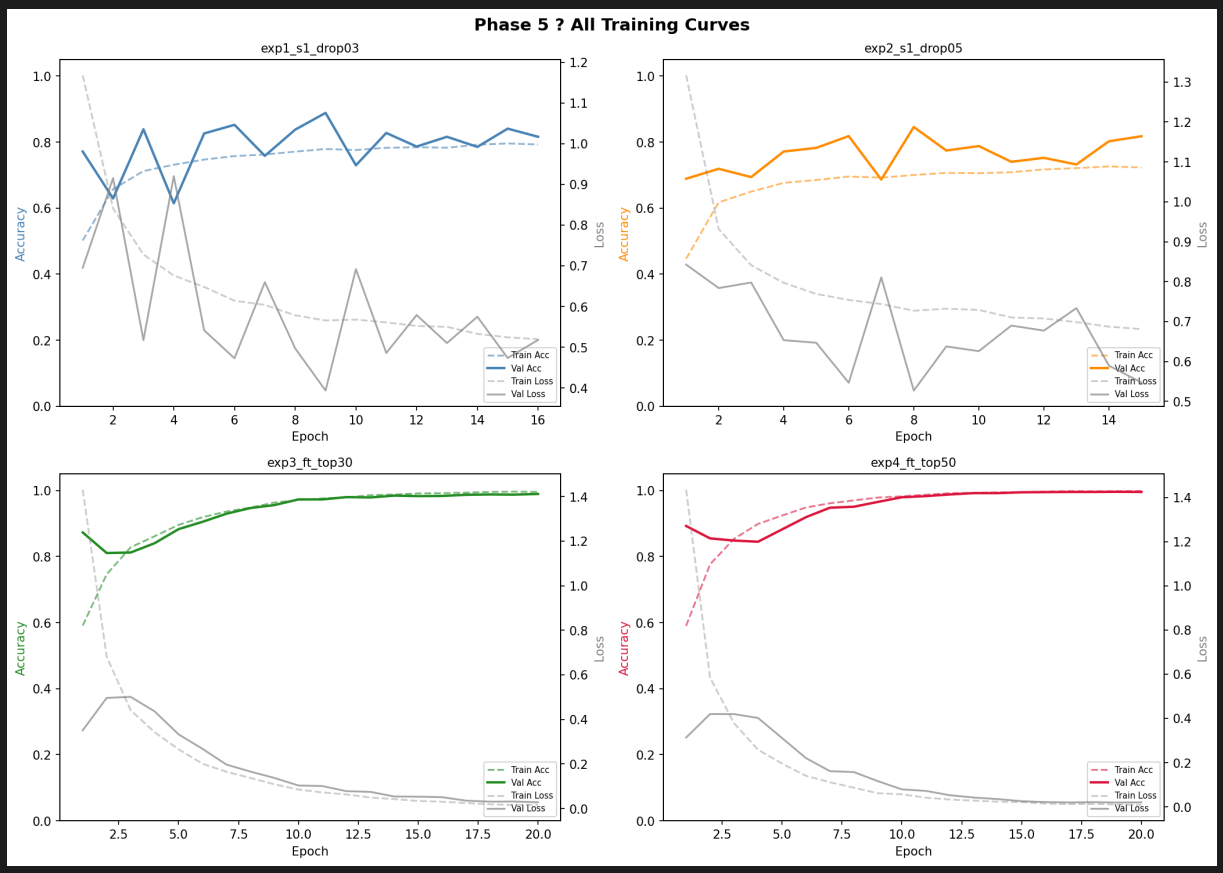
آزمایش	شرح	مرحله	Dropout	لایه‌های آزادشده	بهترین Val Acc
exp1	سر منجمد، Dropout سبک	1	0.3	—	88.85%
exp2	سر منجمد، Dropout قوی	1	0.5	—	84.58%
exp3	Fine-tune کم‌عمق	2	بهترین S1	30 لایه‌ی آخر	98.95%
exp4	Fine-tune عمیق‌تر	2	بهترین S1	50 لایه‌ی آخر	99.60%

جدول ۱۱: آزمایش‌های تنظیم ابرپارامتر — فاز پنجم

در هر دو مرحله از EarlyStopping با ویت 7 و ReduceLROnPlateau با فاکتور 0.5 و ویت 4 استفاده شد تا هم از آموزش بیش از حد جلوگیری بشه، هم نوسانات نرخ یادگیری کنترل بشه.

۵.۷. نمودارهای آموزش

شکل ۲۱ نمودارهای دقت و Loss رو برای هر چهار آزمایش نشون می‌ده.



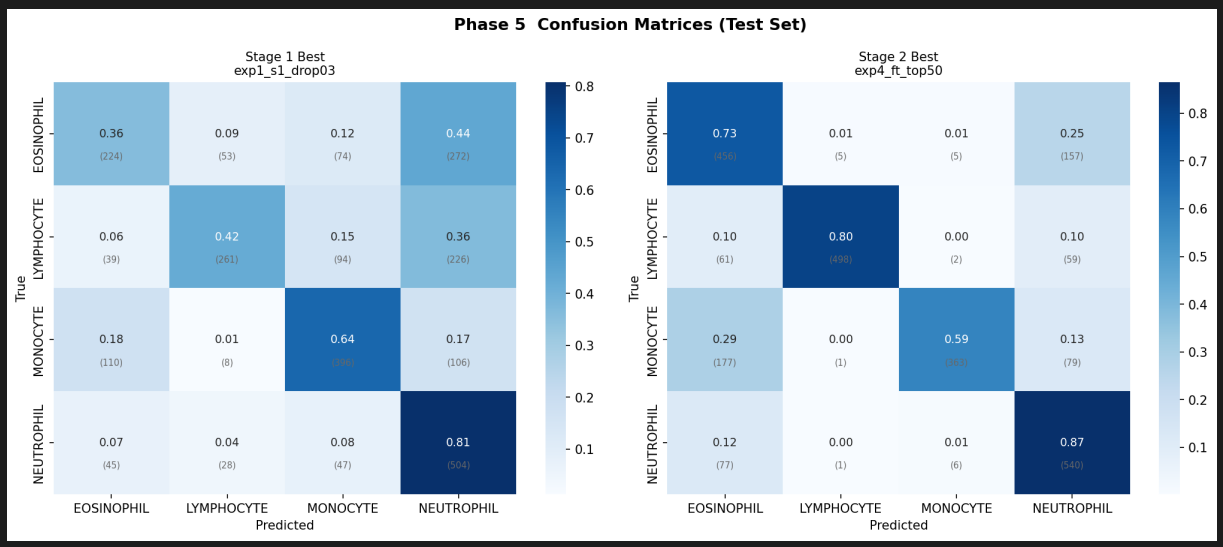
شکل ۲۲: نمودارهای آموزش چهار آزمایش — فاز پنجم

در آزمایش‌های مرحله‌ی اول (exp1 و exp2) دقت اعتبارسنجی نسبتاً سریع به یه سقف می‌رسه، چون تمام قدرت ویژگی‌سازی از ImageNet میاد و سر طبقه‌بند فقط باید یاد بگیره این ویژگی‌ها رو تفسیر کنه. در آزمایش‌های مرحله‌ی دوم (exp3 و exp4) آموزش از جایی که مرحله‌ی اول متوقف شده ادامه می‌یابه و با قدم‌های آرام‌تری بهبود پیدا می‌کنه — انگار داری یه مدل نیمه‌تمام رو با دقت بیشتر تنظیم می‌کنی.

۶.۷. ارزیابی روی مجموعه‌ی آزمون

۱.۶.۷. ماتریس درهم‌ریختگی

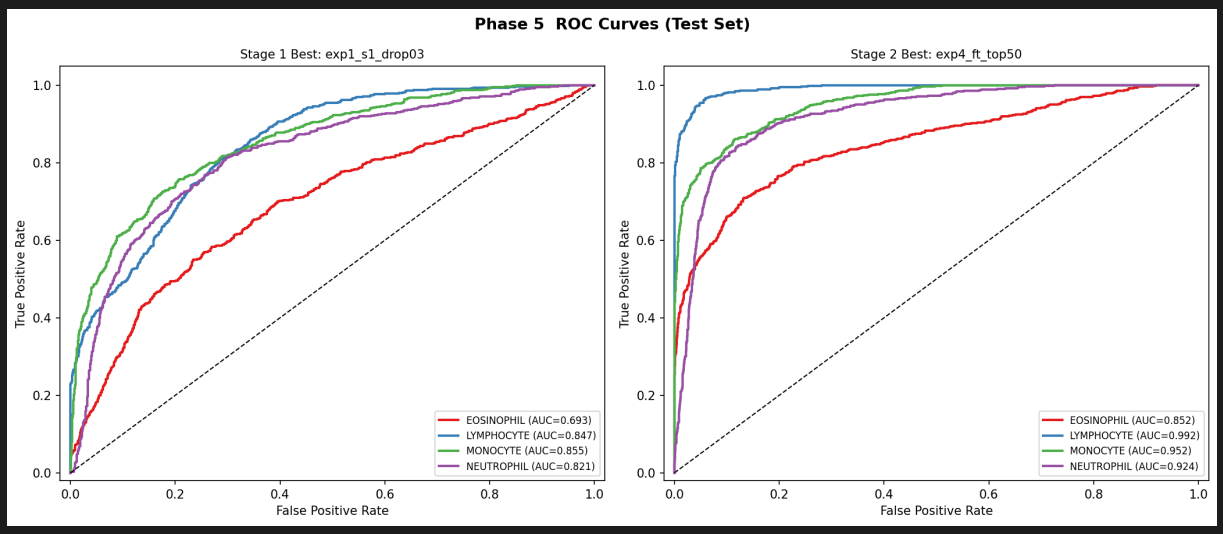
شکل ۲۲ ماتریس درهم‌ریختگی رو برای بهترین مدل مرحله‌ی اول و بهترین مدل مرحله‌ی دوم نشون می‌ده.



شکل ۲۳: ماتریس درهم‌ریختگی روی مجموعه‌ی آزمون — مرحله‌ی اول و دوم

۲.۶.۷. منحنی‌های ROC

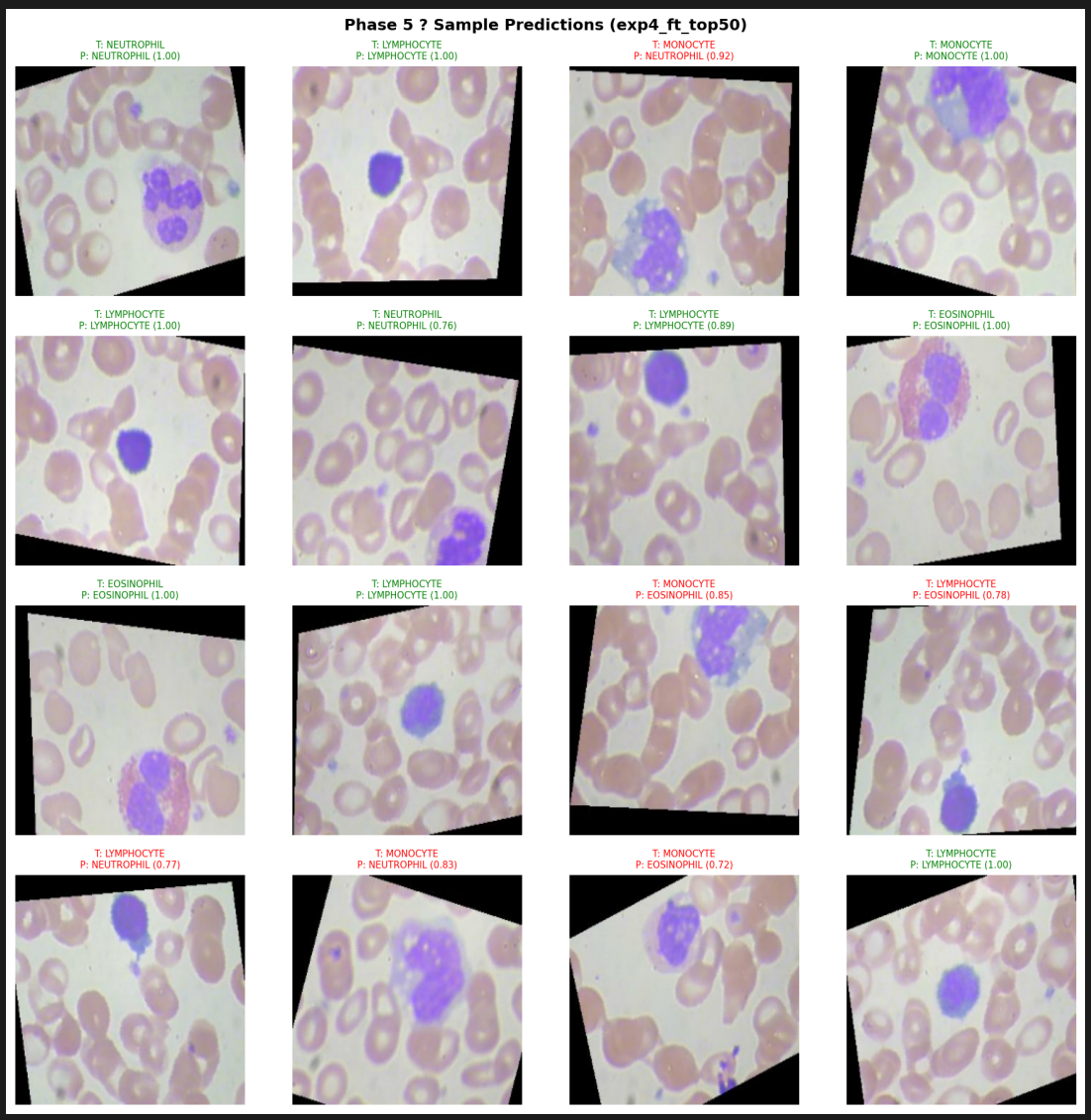
شکل ۲۳ منحنی‌های ROC رو به تفکیک کلاس نشون می‌ده. مساحت زیر منحنی (AUC) برای هر کلاس در داخل نمودار درج شده.



شکل ۲۴: منحنی‌های ROC روی مجموعه‌ی آزمون — مرحله‌ی اول و دوم

۳.۶.۷. نمونه‌های پیش‌بینی

شکل ۲۴ شانزده نمونه از مجموعه‌ی آزمون رو با برچسب واقعی و برچسب پیش‌بینی‌شده نشون می‌ده. پیش‌بینی‌های درست با کادر سبز و اشتباه‌ها با کادر قرمز مشخص شدن.



شکل ۲۵: نمونه‌های پیش‌بینی روی مجموعه‌ی آزمون — بهترین مدل فاز پنجم

۴.۶.۷. معیارهای کلاس به کلاس روی مجموعه‌ی آزمون

جدول ؟؟ نتایج دقیق‌تر رو برای بهترین مدل هر مرحله نشون می‌ده — exp1 از مرحله‌ی اول (دقت کل 55.69%) و exp4 از مرحله‌ی دوم (دقت کل 74.67%).

مدل	کلاس	Precision	Recall	F1	Support
	EOSINOPHIL	0.59	0.73	0.65	623
	LYMPHOCYTE	0.99	0.80	0.89	620
exp4 (مرحله‌ی دوم)	MONOCYTE	0.97	0.59	0.73	620
	NEUTROPHIL	0.65	0.87	0.74	624
	Macro avg	0.80	0.75	0.75	2,487

جدول ۱۲: معیارهای کلاس به کلاس روی مجموعه‌ی آزمون — بهترین مدل مرحله‌ی دوم (exp4)

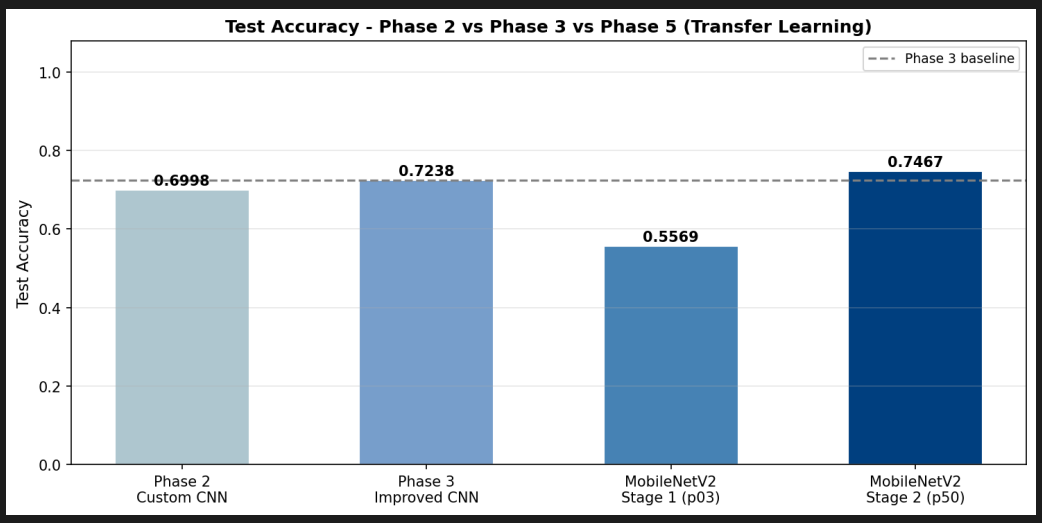
Fine-tuning عمیق‌تر (exp4) تونست دقت کل رو از 55.69% مرحله‌ی اول به 74.67% برسونه — که یه جهش قابل توجهه. بهترین کلاس LYMPHOCYTE با F1=0.89 و ضعیف‌ترین EOSINOPHIL با F1=0.65 بود.

۷.۷. مقایسه‌ی نهایی با فازهای قبل

جدول ۱۲ دقت آزمون تمام فازها رو کنار هم می‌ذاره و شکل ۲۵ همین مقایسه رو به صورت نمودار میله‌ای نشون می‌ده.

مدل	فاز	دقت آزمون
بهترین CNN سفارشی	فاز دوم	69.98%
بهترین CNN بهبودیافته	فاز سوم	72.38%
MobileNetV2 مرحله‌ی اول	فاز پنجم	55.69%
MobileNetV2 مرحله‌ی دوم	فاز پنجم	74.67%

جدول ۱۳: مقایسه‌ی دقت آزمون در تمام فازها



شکل ۲۶: مقایسه‌ی دقت آزمون — فاز دوم، سوم، و پنجم

۸.۷. تحلیل نهایی

نتایج این فاز تصویر جالب و در عین حال صادقانه‌ای از محدودیت‌های این دیتاست ارائه می‌ده.

شکاف اعتبارسنجی و آزمون. بزرگ‌ترین چیزی که این فاز آشکار کرد همون مشکلیه که از فاز سوم می‌شناختیمش: جابجایی توزیع میان TRAIN و TEST. exp1 در اعتبارسنجی به 88.85% رسید اما روی آزمون فقط 55.69% داد — یعنی مدل با وجود وزن‌های ImageNet، به همون دام overfitting به توزیع آموزشی افتاد. exp4 این شکاف رو کوچیک‌تر کرد: اعتبارسنجی 99.60%، آزمون 74.67%. شکاف هنوز وجود داره، که نشون می‌ده ریشه‌ی مشکل در خود دیتاسته، نه فقط در معماری مدل.

مقایسه با فازهای قبل. بهترین مدل این فاز (exp4 با دقت آزمون 74.67%) نسبت به بهترین مدل فاز سوم (72.38%) حدود 2.3 واحد درصد بهتر شد. این بهبود کوچیک‌تر از اون چیزیه که معمولاً از Transfer Learning انتظار می‌ره؛ ولی با توجه به جابجایی توزیع دیتاست قابل توضیحه — وزن‌های ImageNet کمک کردن ویژگی‌های بهتری استخراج بشه، اما مشکل اصلی که تفاوت توزیعه با هیچ معماری‌ای به‌تنهایی برطرف نمی‌شه.

تحلیل کلاس به کلاس. در بهترین مدل (exp4) کلاس LYMPHOCYTE با Precision بالای 0.99 بهترین عملکرد رو داشت — شاید چون این سلول ویژگی بصری مشخص‌تری (هسته‌ی گرد بزرگ) داره که MobileNetV2 به خوبی یادش گرفته. در مقابل، MONOCYTE با Recall پایین 0.59 مشکل‌دارترین کلاس بود: مدل اکثر MONOCYTEهای واقعی رو به کلاس‌های دیگه تشخیص داده، که احتمالاً به دلیل شباهت ظاهری این سلول با LYMPHOCYTE در نمونه‌های آزمونه.

نتیجه‌گیری کلی. از میان همه‌ی مدل‌های آزمایش‌شده در این پروژه، MobileNetV2 با Fine-tuning (exp4) بهترین دقت آزمون رو داشت. ولی بهبودش نسبت به فاز سوم اندک بود، چون مشکل اصلی این دیتاست — جابجایی توزیع بین مجموعه‌های آموزش و آزمون — یه مشکل ساختاریه که با انتخاب معماری بهتر به‌تنهایی حل نمی‌شه. پس برای حل اساسی‌تر این مشکل، روش‌هایی مثل domain adaptation یا جمع‌آوری داده‌ی آموزشی متنوع‌تر لازمه.