پروژه پایانی درس مباحث پیشرفته هوش مصنوعی

موضوع پروژه:

Traffic Sign Recognition Using Deep Neural Networks

استاد درس: جناب دکتر حسن ختن‌لو

دانشجو: مهدی شیروانی

شماره دانشجویی: ۴۰۰۱۲۳۵۸۰۲۱

مهر ۱۴۰۴

مقدمه

تشخیص علائم راهنمایی و رانندگی یکی از مسائل مهم در حوزه بینایی ماشین و پردازش تصویر است. این مسئله نقش اساسی در سیستم‌های رانندگی خودکار و ایمنی جاده دارد. در این پروژه از داده‌های GTSRB و معماری‌های مختلف شبکه‌های عصبی عمیق برای شناسایی و دسته‌بندی علائم راهنمایی استفاده شده است.

هدف اصلی، بررسی عملکرد شبکه‌های مختلف و مقایسه دقت و کارایی آن‌ها است. برای این کار از شبکه‌های از پیش آموزش دیده و شبکه cnn,TinyResNet، MobileNet، DenseNet و EfficientNetB استفاده میکنیم.

برای اینکار از زبان پایتون و کتابخانه pytorch در محیط google colab استفاده می‌کنیم.

لینک کد: Google Colab Link

دیتاست

دیتاست German Traffic Sign Recognition Benchmark یکی از دیتاست‌های استاندارد در حوزه‌ی تشخیص علائم رانندگی است. این دیتاست شامل تصاویر رنگی از علائم ترافیکی است که در شرایط مختلف نوری، آب‌وهوایی و زاویه دید جمع‌آوری شده‌اند.

- تعداد کلاس‌ها: ۴۳ کلاس مختلف
- سایز تصاویر: به شکل یکنواخت به ۳۲×۳۲×۳ تبدیل شده‌اند.
- فایل‌های داده: هر فایل شامل دیکشنری با کلیدهای زیر است:
  - features: آرایه‌ای از تصاویر (N, ۳۲, ۳۲, ۳)
  - labels: برچسب عددی (۰ تا ۴۲)

سه فایل اصلی وجود دارد:

- train.p
- valid.p
- test.p

داده‌های train برای آموزش مدل، داده‌های valid برای اعتبار سنجی مدل و داده‌های test برای ارزیابی نهایی استفاده می‌شوند.

پیش‌پردازش داده‌ها

قبل از ورود داده‌ها به شبکه‌های عصبی، چند مرحله پیش‌پردازش داده انجام می‌دهیم.

1. نرمال‌سازی:
   میانگین و انحراف معیار تصاویر کل دیتاست را محاسبه می‌کنیم و هر تصویر را بر اساس این مقادیر نرمال‌سازی میکنیم.
2. Data Augmentation
   برای جلوگیری از overfitting و افزایش تنوع داده‌ها، از تکنیک‌های زیر استفاده کردیم:
   - چرخش تصادفی (تا ۱۵ درجه)
   - تغییرات کوچک در مقیاس
3. ساخت DataLoader
   داده‌ها به صورت batch های ۱۲۸تایی با ترتیب های مختلف بارگذاری می‌کنیم تا بتوان آموزش راحت‌تر و سریعتری داشته باشیم.

معماری شبکه‌ها

در این بخش معماری پنج شبکه استفاده شده را شرح می‌دهیم.

❄ SimpleCNN

این معماری از پیش آموزش دیده نیست و از صفر آن را ساخته ایم.

در این معماری از چندین لایه cnn و maxpooling و relu و dropout استفاده شده و همچنین در لایه آخر از یک طبقه‌بند ۴۳ تایی استفاده کردیم برای تشخیص کلاس هر تصویر.

❄TinyResNet

resNet به کمک skip connections می‌تواند مشکل gradient vanishing در شبکه‌های عمیق را حل کند. TinyResNet نسخه‌ای کوچک‌شده از ResNet34 است که شامل بلوک‌های residual کمتر است.

این مدل به این صورت است که یک convolution اولیه برای گرفتن ویژگی‌های پایه داریم و سه stage پشت سر هم که هر stage شامل چند بلوک residual. در هر stage تعداد کانال ها دو برابر میشه (۱۶ -> ۳۲ -> ۶۴).

در آخر یک global average pooling و یک fully connected layer برای دسته‌بندی ۴۳ کلاس‌ها استفاده می‌کنیم.این شبکه سریع‌تر آموزش می‌بیند ولی همچنان از مزیت skip connection استفاده می‌کند.

❄ MobileNetV2

MobileNetV2 برای اجرا روی دستگاه‌های سبک مانندموبایل طراحی شده است. ویژگی اصلی آن استفاده از:

- Depthwise Separable Convolutions برای کاهش محاسبات
- Inverted Residuals بلوک‌هایی با bottle neck

این معماری تعداد پارامترها را به شدت کاهش می‌دهد و در عین حال دقت بالایی دارد.

❄ DenseNet121

DenseNet ایده‌ی اتصال متراکم دارد. یعنی هر لایه خروجی خودش را به تمام لایه‌های بعدی منتقل می‌کند.این موضوع باعث استفاده‌ی بهینه از ویژگی‌ها و جلوگیری از یادگیری تکراری می‌شود.

DenseNet121 شامل ۴ بلوک Dense است و برای طبقه‌بندی از یک fully connected layer استفاده می‌کند.

* EfficientNet-B0

EfficientNet نوعی از شبکه‌های عصبی است که با روش Compound Scaling بهینه شده. این روش به جای افزایش صرفاً عمق یا عرض، هر سه (عمق، عرض، رزولوشن) را به شکل هماهنگ افزایش می‌دهد.

* EfficientNet-B0 نسخه پایه است و تعادل خوبی میان دقت و کارایی ایجاد می‌کند.

همه‌ی این شبکه‌ها بجز simpleCnn از پیش روی داده‌های مختلف زیادی آموزش داده شده‌اند.

نحوه‌ی استفاده ما به این صورت است که کل شبکه دوباره با داده‌های ما آموزش می‌بیند و وزن های با داده‌های جدید آپدیت می‌شوند. این به این منظور است که مدل دانش قبلی را دارد و دانش جدید را هم با داده‌های ما یاد می‌گیرد. درواقع ما از روش fine-tuning استفاده کرده‌ایم.

آموزش و پیاده‌سازی

حال برای آموزش مدل‌ها از پارامتر های زیر استفاده میکنیم.

Optimizer: Adam با learning rate = 0.001

Loss: CrossEntropyLoss

Batch size: 128

Epoch: 20

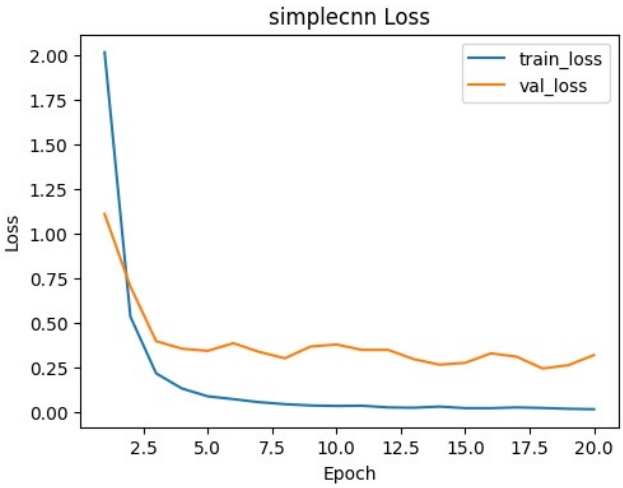
ما مدل را یکبار با داده‌های augment شده و یکبار با داده‌های بدون augment شده آموزش دادیم.

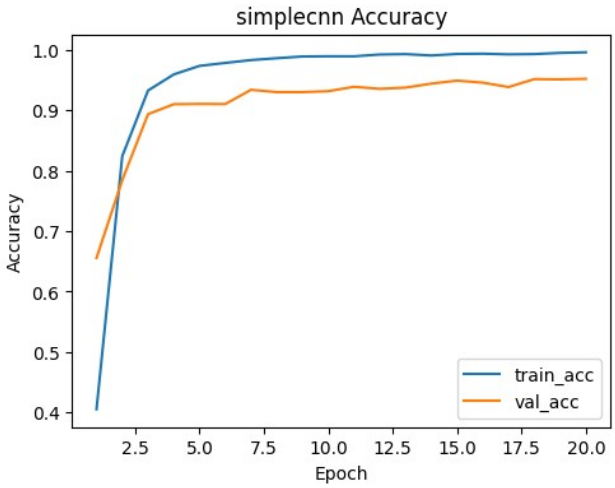
اگر مدل دقت بالاتری را با داده‌های valid ثبت کرد آن را ذخیره می‌کنیم. برای آموزش این مدل ها از کارت گرافیک تسلا مدل T4 استفاده شده است.

نتایج و ارزیابی

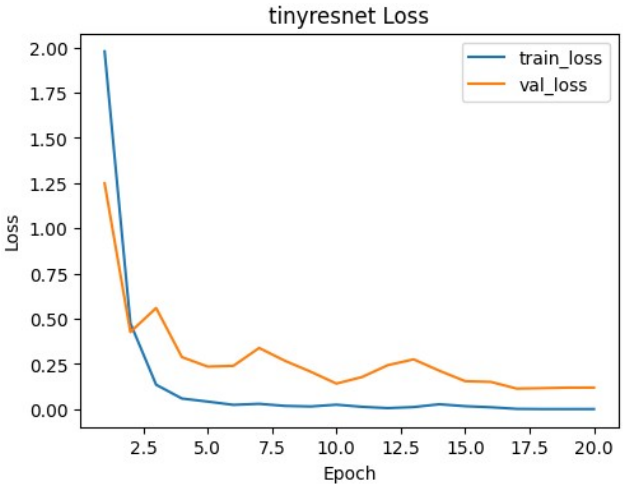
بعد از آموزش مدل ها، برای هر یک نمودار Loss و Accuracy در طول epochs ترسیم شده است. این نمودارها نشان می‌دهند که مدل‌ها به تدریج به دقت بالایی می‌رسند.

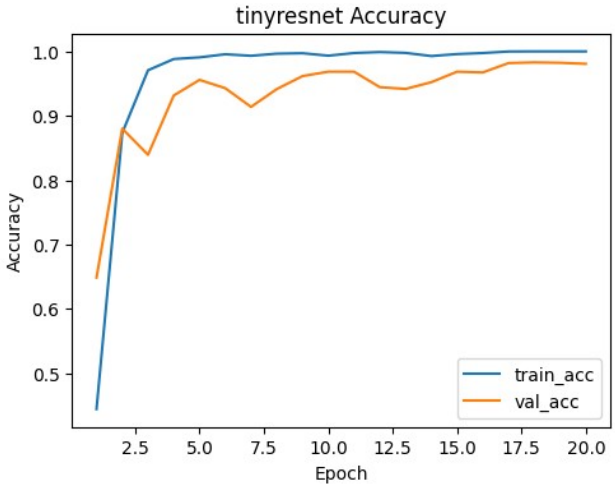
مدل simple Cnn با داده‌های aumented



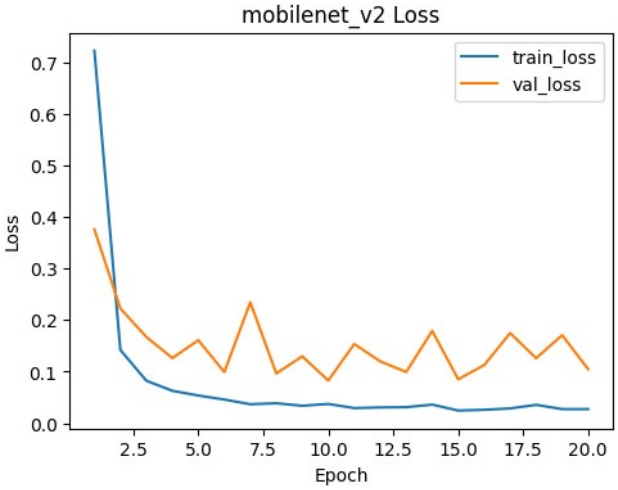


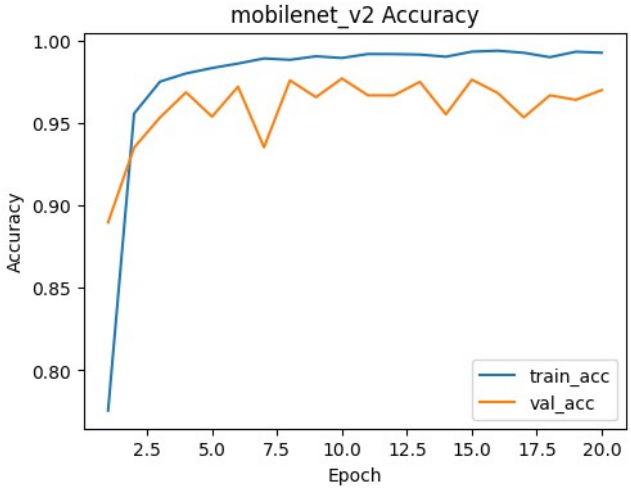
مدل tinyResNet با داده‌های augmented



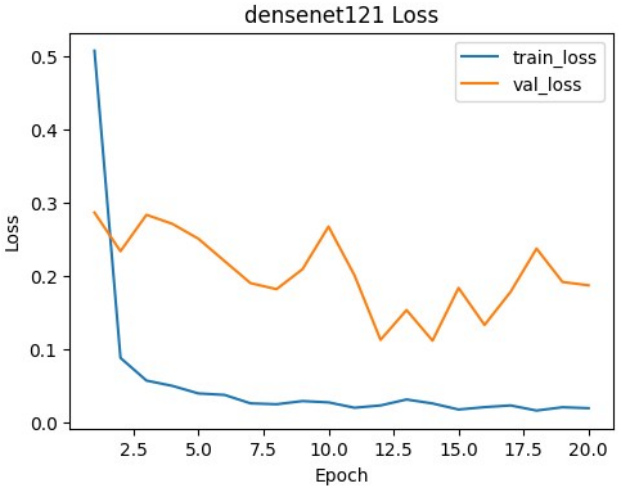


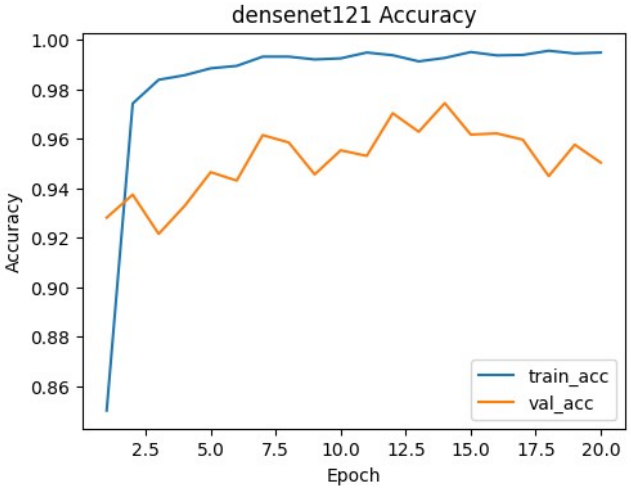
مدل mobileNet با داده‌های augmented



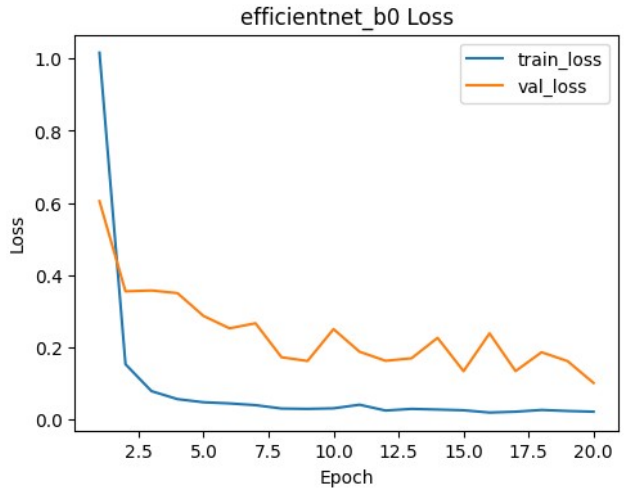


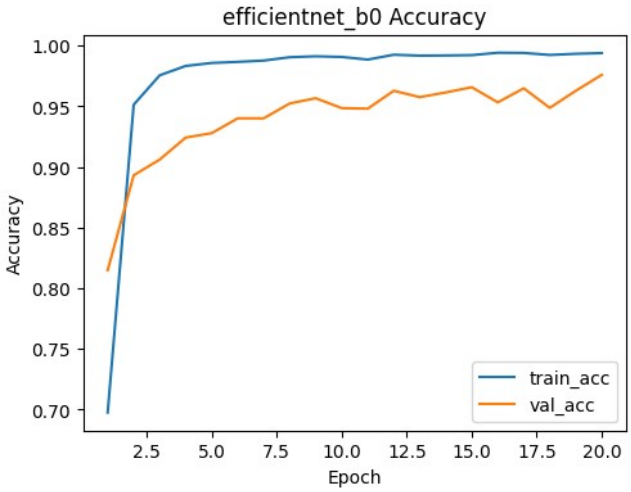
مدل denseNet با داده‌های augmented





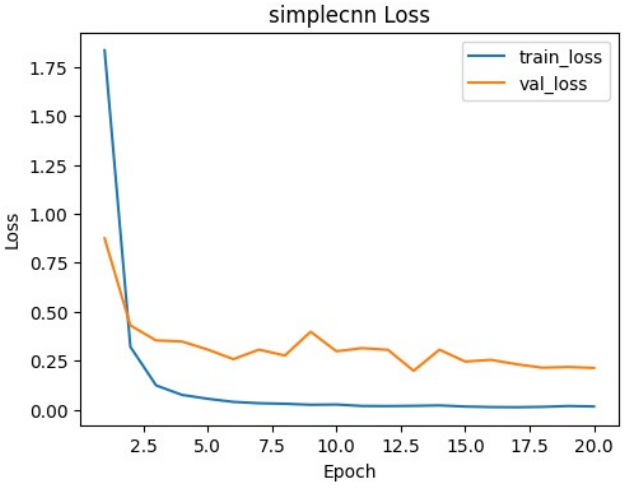
مدل efficient با داده‌های augmented

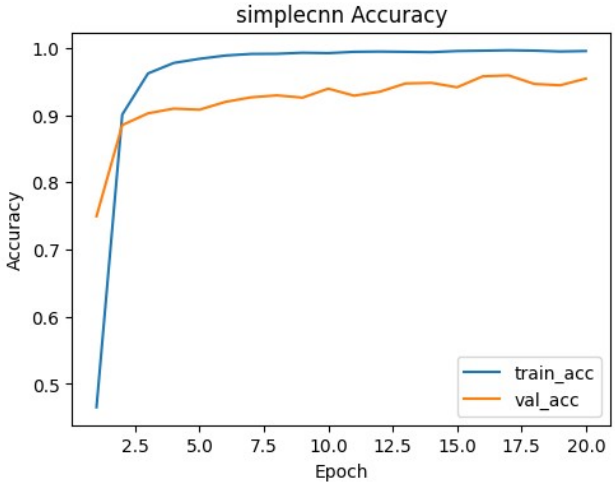




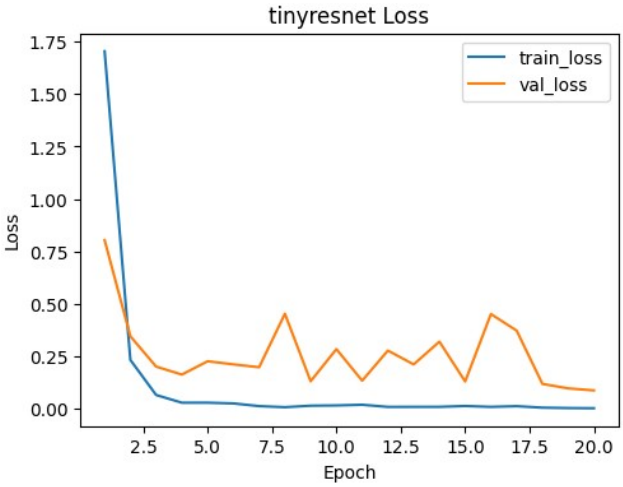
حال مدل ها را با داده‌های بدون augmentation ارزیابی می‌کنیم.

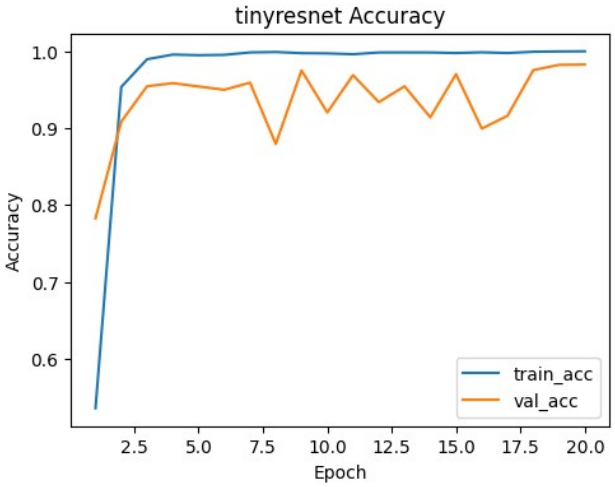
مدل simple Cnn با داده‌های بدون augmentation



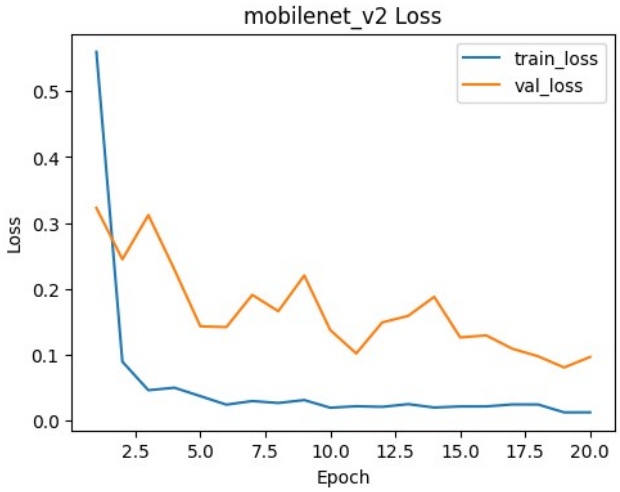


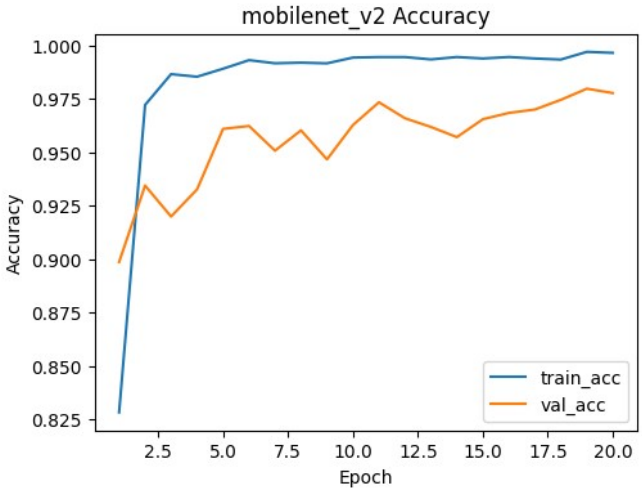
مدل tinyResNet با داده‌های بدون augmentation



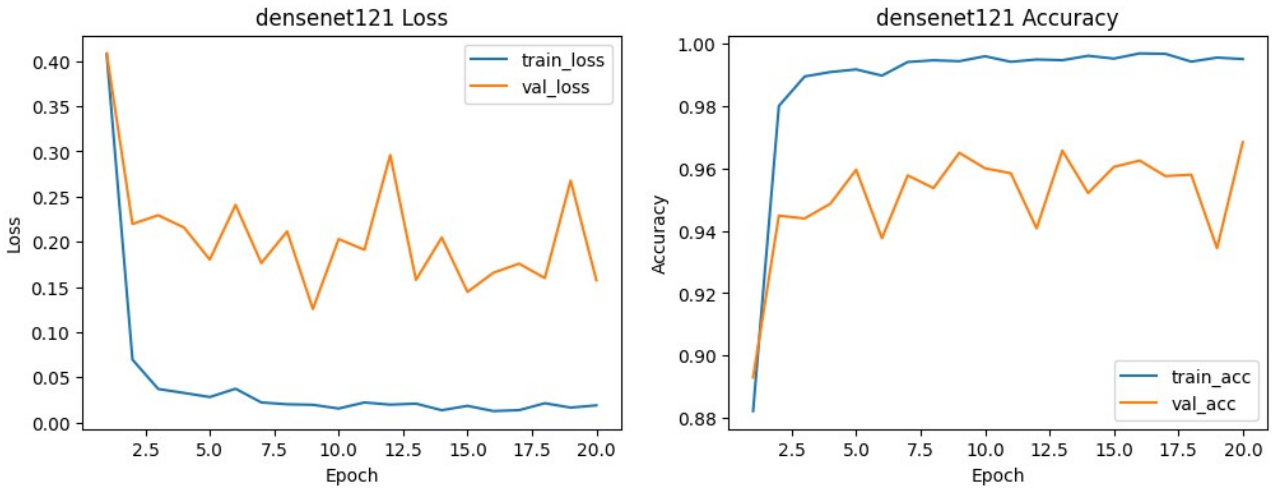


مدل mobileNet با داده‌های بدون augmentation

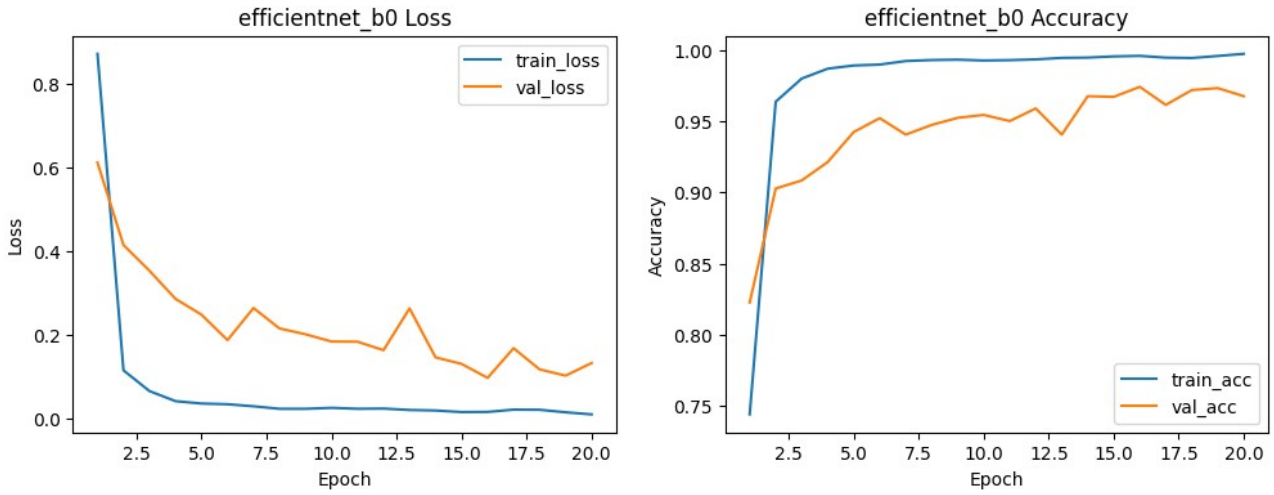




مدل denseNet با داده‌های بدون augmentation



مدل efficient با داده‌های بدون augmentation



مدل tinyResNet نسبت به سایر مدل ها عمل‌کرد بهتری داشته است.

در آخر هم با استفاده از روش ensemble طبقه‌بندی را با بالاترین دقت انجام می‌دهیم، به این صورت که ما داده ورودی را به هر کدام از مدل های بالا می‌دهیم و آن‌ها طبقه‌بندی را انتخاب می‌کنند و هر کدام یک درصدی از دقت پیش‌بینی را ارائه می‌دهند، در نهایت کلاسی که بالاتری نظر را میان همه داشته است انتخاب می‌شود.

Augmentation

```
Predicting probabilities for tinyresnet ...
Predicting probabilities for simplecnn ...
Predicting probabilities for mobilenet_v2 ...
Predicting probabilities for densenet121 ...
Predicting probabilities for efficientnet_b0 ...

Ensemble Soft Voting Accuracy on Test set: 0.9853
```

Not Augmentation

```
Predicting probabilities for tinyresnet ...
Predicting probabilities for simplecnn ...
Predicting probabilities for mobilenet_v2 ...
Predicting probabilities for densenet121 ...
Predicting probabilities for efficientnet_b0 ...

Ensemble Soft Voting Accuracy on Test set: 0.9863
```

نتیجه‌گیری

این نتیجه نشان می‌دهد که افزایش داده‌ها برای مدل هایی که از پیش آموزش دیده شده‌اند همیشه نمی‌تواند عمل‌کرد بهتری داشته باشد و بستگی به نوع مسأله و داده‌های موجود و همچنین پیچیدگی مدل دارد.

در این مسأله با توجه به اینکه داده‌های زیادی داشته‌ایم و مدل ها هم تقریباً پیچیده بوده‌اند، هر دو روش augmentation و not augmentation تقریباً دقت یکسانی دارند و افزایش داده خیلی تأثیر گذار نمی‌باشد.