

نام و نام خانوادگی	ادیب رضائی - امیرمحمد خسروی
شماره دانشجویی	810198386 - 810198401
تاریخ ارسال گزارش	۱۴۰۲.۰۱.۱۷

	<p>به نام خدا</p> <p>دانشگاه تهران</p> <p>دانشکده مهندسی</p> <p>برق و کامپیوتر</p>	
<p>درس شبکه‌های عصبی و یادگیری عمیق</p> <p>تمرین دوم</p>		

فهرست

پاسخ 1. شبکه عصبی پیچشی کم عمق برای طبقه بندی تصاویر	1
پاسخ 2. طبقه‌بندی تصاویر اشعه ایکس قفسه سینه	1

شکل‌ها

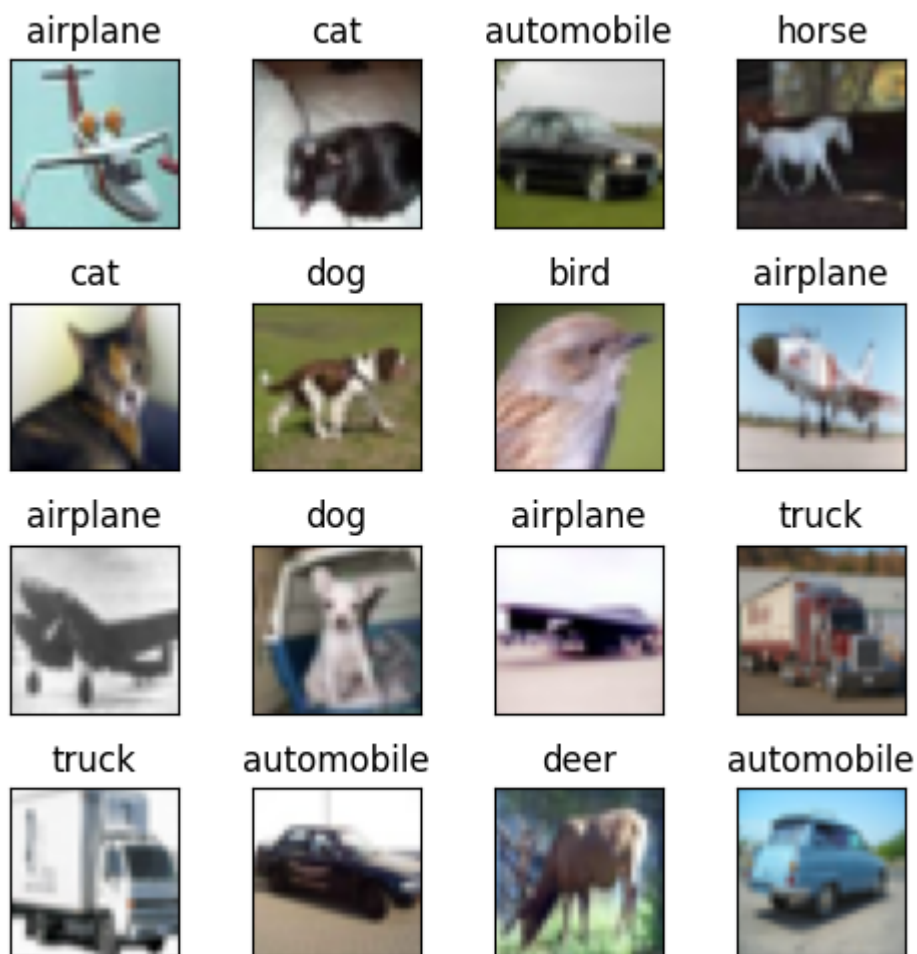
پاسخ 1. شبکه عصبی پیچشی کم عمق برای طبقه بندی تصاویر	1
آماده سازی و پیش پردازش داده ها . ۱-۱	1
پاسخ 2. طبقه بندی تصاویر اشعه ایکس قفسه سینه	1

جدول‌ها

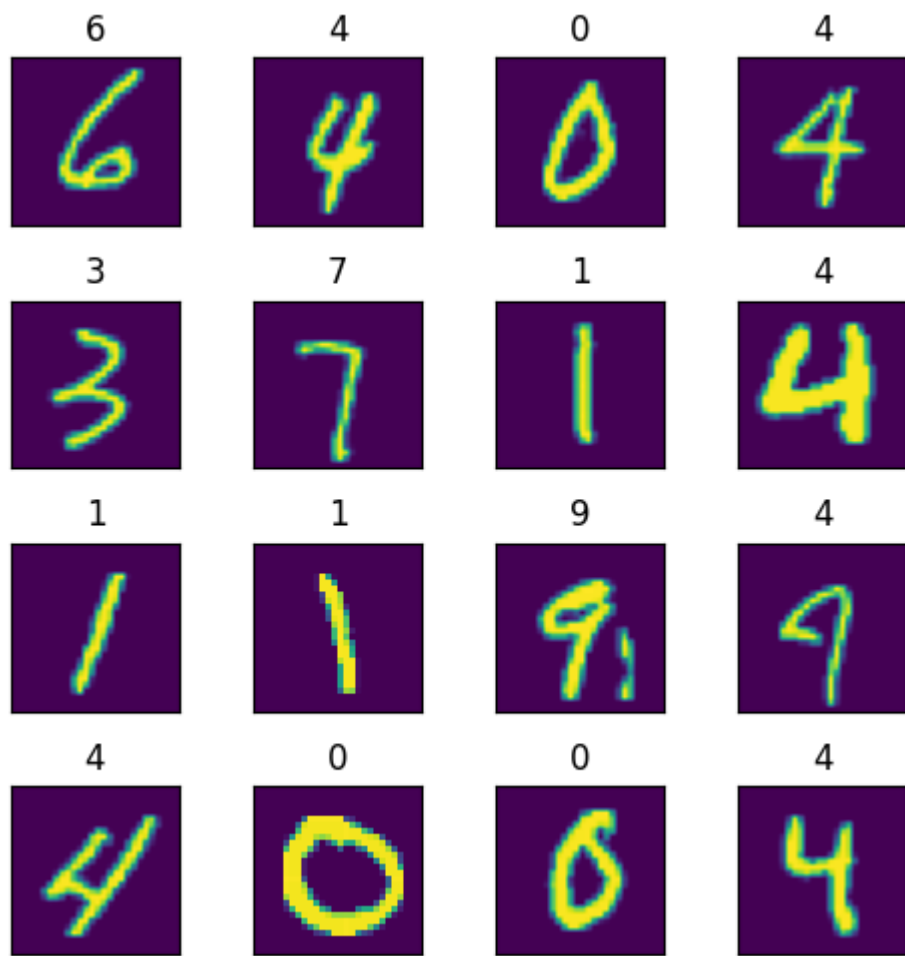
پاسخ 1. شبکه عصبی پیچشی کم عمق برای طبقه بندی تصاویر	1
--	---

پاسخ 1. شبکه عصبی پیچشی کم عمق برای طبقه بندی تصاویر

۱-۱. آماده سازی و پیش پردازش داده ها
ابتدا از داده ها سمپل گرفته و آنها را رویت میکنیم.



شکل ۱.۱. CIFAR10



شكل ١.٢. MNIST



شکل ۱.۳. FASHION MNIST

سپس ابعاد دیتاست ها را بررسی میکنیم. ابعاد دیتاست های Fashion Mnist و Mnist به صورت 28×28 است و ابعاد cifar10 به شکل $32 \times 32 \times 3$ است. برای اینکه مشکلی در عملیات train کردن پیش نیاید ابعاد Mnist و Fashion Mnist را به شکل $1 \times 28 \times 28$ در می آوریم. نتیجه به صورت شکل ۱.۴ میباشد.

```
print(f" cifar10 train shape {cifar10_x_train.shape } & test shape {cifar10_x_test.shape}")
print(f"mnist train shape {mnist_x_train.shape } & test shape {mnist_x_test.shape}")
print(f"fashion_mnist train shape {fashion_mnist_x_train.shape } & test shape {fashion_mnist_x_test.shape}")

cifar10 train shape (50000, 32, 32, 3) & test shape (10000, 32, 32, 3)
mnist train shape (60000, 28, 28, 1) & test shape (10000, 28, 28, 1)
fashion_mnist train shape (60000, 28, 28, 1) & test shape (10000, 28, 28, 1)
```

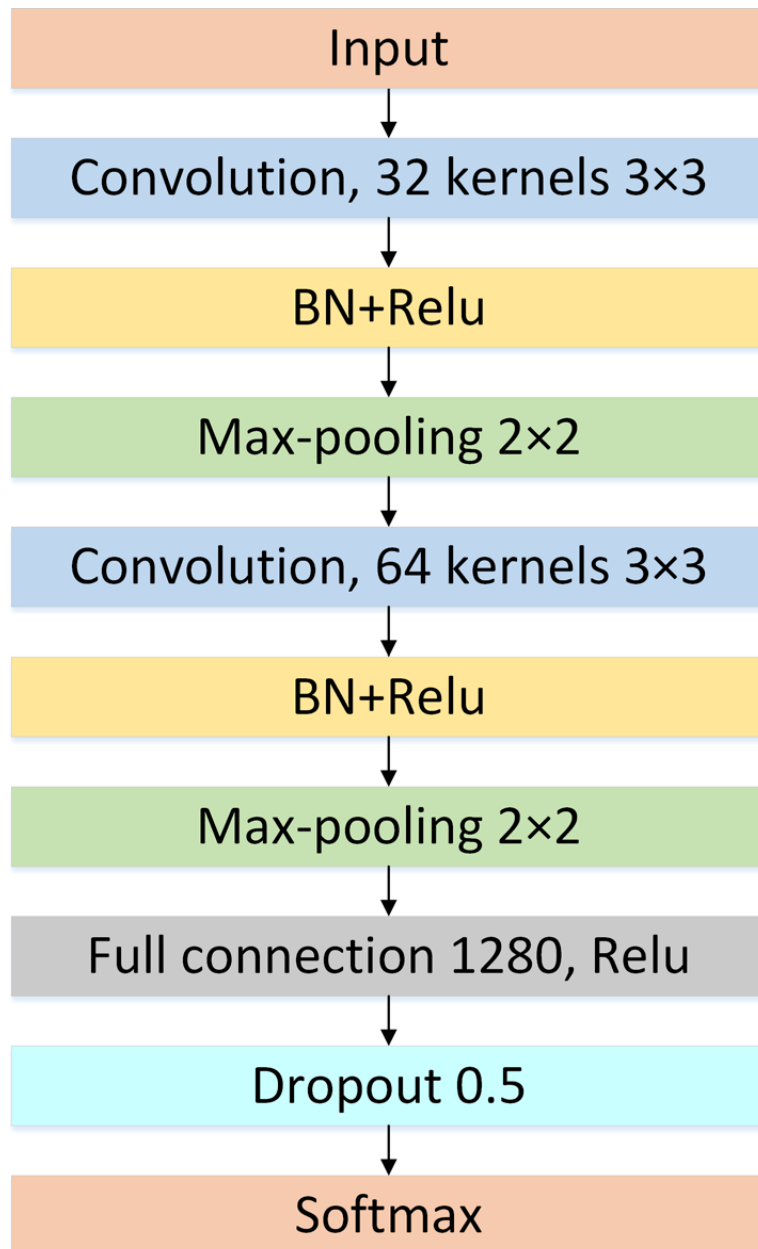
شکل ۱.۴. ابعاد هر دیتاست

برای پیش پردازش، داده های CIFAR10, MNIST, Fashion MNIST را از بازه $[0, 255]$ به بازه $[0, 1]$ می بریم. علت اینکار این است که گرادیان به شکل نرم تری به سمت نقاط اکسترمم فضای فیچر ها حرکت کند و برای همه فیچر ها با سرعت یکسانی حرکت کند. همچنین سرعت همگرایی به سمت مینیمم را هم افزایش میدهد.

همچنین لیبل های هر دسته را onehot میکنیم تا لیبل ها ترتیب نداشته باشند و پرفورمنس مدل بهبود یابد.

۱-۲. توضیح لایه های مختلف معماری شبکه

مدل SCNN شامل دو لایه Convolutional دو لایه max-pooling و یک لایه fully-connected و یک لایه softmax مطابق شکل ۱.۵ است.



شکل ۱.۵. معماری SCNNB

در ابتدا فیچر های اولیه توسط لایه کانولوشن با 32 فیلتر استخراج میشود و توسط Batch Normalization فیچر مپ آن نرمالایز میشود (میانگین و انحراف معیار). سپس لایه 2×2 max pooling استفاده میشود تا ابعاد داده را کم کند و محاسبات را سریع تر کند. در دولایه بعد همین مراحل طی میشود با این تفاوت که لایه کانولوشن این بار 64 فیلتر دارد. نتیجه این چهار لایه به لایه fully connect با تعداد نرون ۱۲۸۰ داده میشود و داده ها طبقه بندی میشوند. همچنین برای عمل regularization و عدم اورفیت از Dropout با سایز 0.5 استفاده شده تا در

هر مرحله بخشی از نوروں ها غیر فعال شده و generalization مدل بهبود یابد و overfit رخ ندهد. در آخر داده ها به لایه softmax با ۱۰ نوروں میروند.

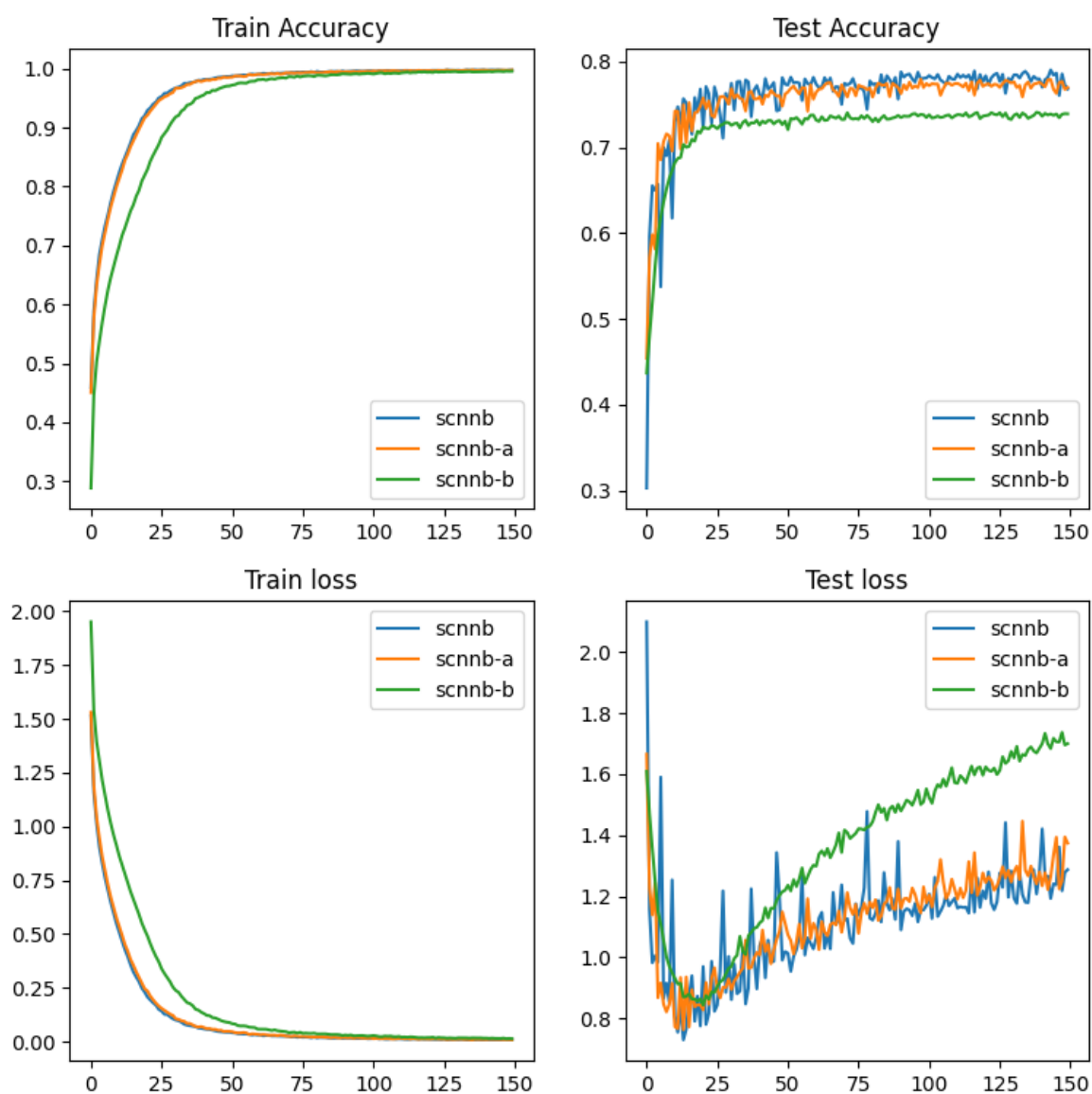
تمرکز اصلی این مقاله بر لایه BN یا Batch Normalization است. در این لایه فیچر مپ خروجی از لایه کانولوشن قبل از اینکه به لایه activation برود به ازای هر بچ نرمالایز میشود و در محدوده حساس به ورودی می افتد و در نتیجه اینکار احتمال gradient vanishing را کم میکند. gradient vanishing زمانی اتفاق می افتد که گرادیان آنقدر کم شود که شبکه در زمان آپدیت شدن وزن ها تغییری احساس نکند و دیگر آموزش داده نشود. در این مقاله برای اینکه نشان دهد این لایه تاثیر مهمی در آموزش شبکه دارد هر دیتاست را با سه مدل متفاوت آموزش میدهد که در بخش های بعدی نتیجه و تحلیل آن را خواهیم گفت.

۳-۱. پیاده سازی معماری

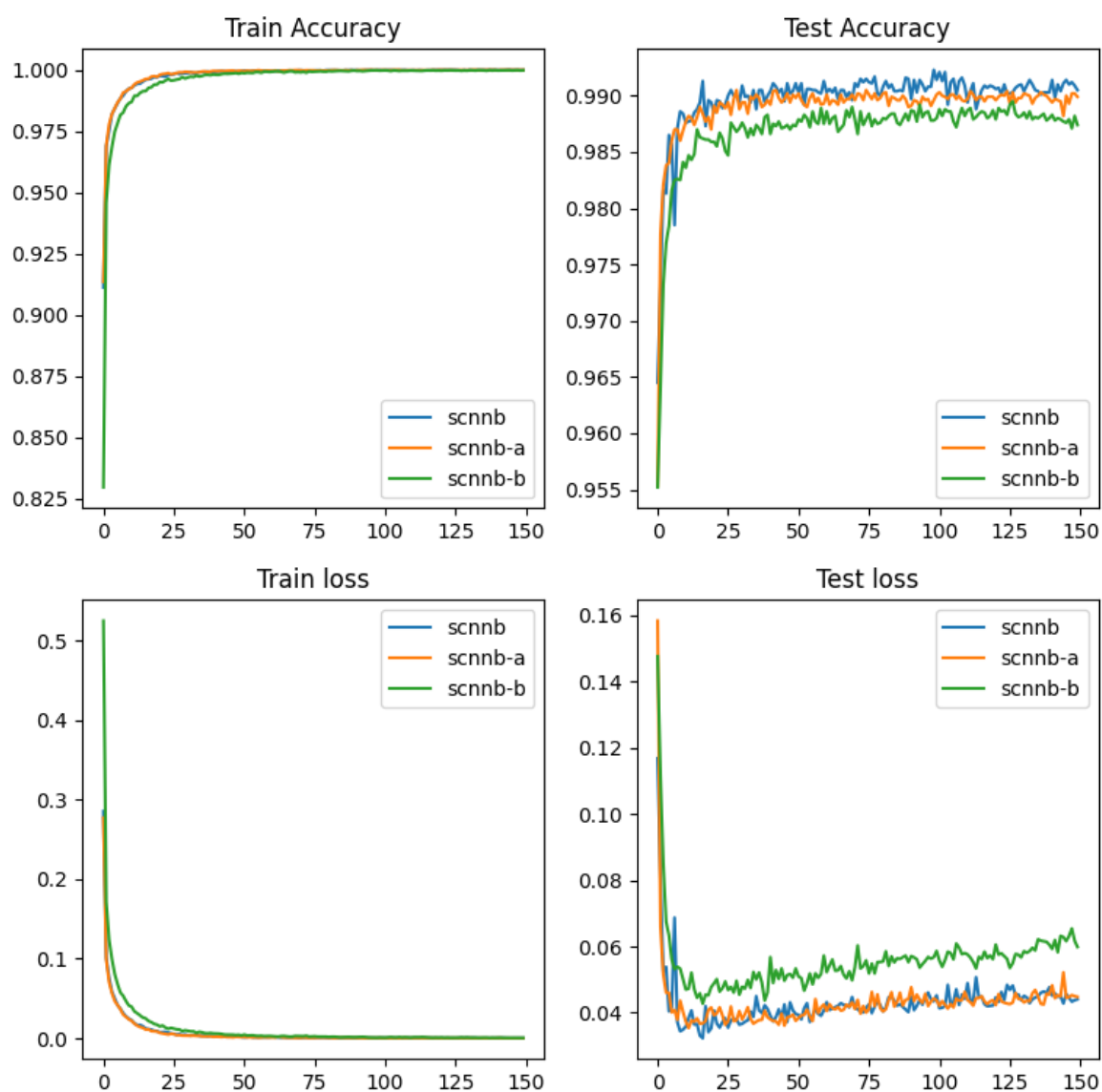
در بخش Define Model نوت بوک مدل های SCNNB, SCNNB-a, SCNNB-b به شکل تابع پیاده سازی شده اند. برای هایپر پارامتر ها، تعداد ایپاک را ۱۵۰ و Batch Size را برابر ۱۲۸ قرار دادیم. از اپتیمایزر SGD با learning rate = 0.02 و momentum=0.9 و weight decay = 0.000005 طبق مقاله استفاده کردیم. برای تابع loss در مقاله چیزی ذکر نشده است و ما از categorical_crossentropy استفاده کردیم. از آنجایی که categorical_crossentropy با لیبل های one hot کار میکند و ما نیز لیبل های خود را one hot کردیم گزینه مناسبی میباشد همچنین به طور کلی نتایج خوبی را نیز به استناد به منابع معتبر به همراه دارد.

۴-۱. نتایج پیاده سازی

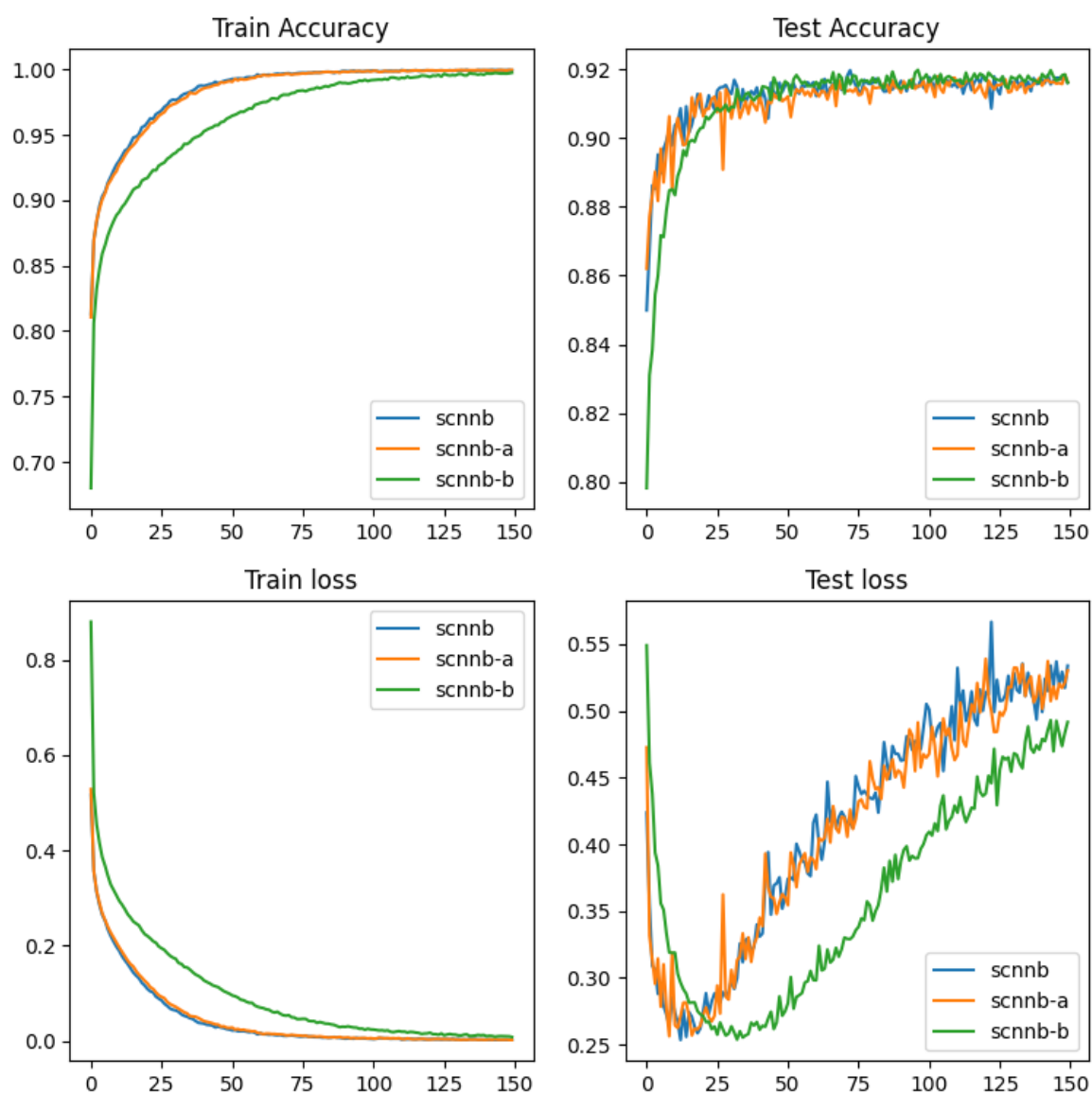
هر کدام از دیتاست های Fashion Mnist و Cifar10, Mnist را به سه مدل SCNNB, SCNNB-a و SCNNB-b می دهیم و نتایج را بررسی میکنیم. در مدل SCNNB معماری مطابق آنچه پیشتر توضیح داده شده است. در مدل SCNNB-a لایه اول BN برداشته شده اما لایه دوم آن تغییر نکرده. در مدل SCNNB-b هر دو لایه BN برداشته شده است. نتایج به صورت شکل های زیر میباشد.



شکل ۱.۶. نتایج Cifar10



شکل ۱.۷. نتایج Mnist



شکل ۱.۸. نتایج Fashion Mnist

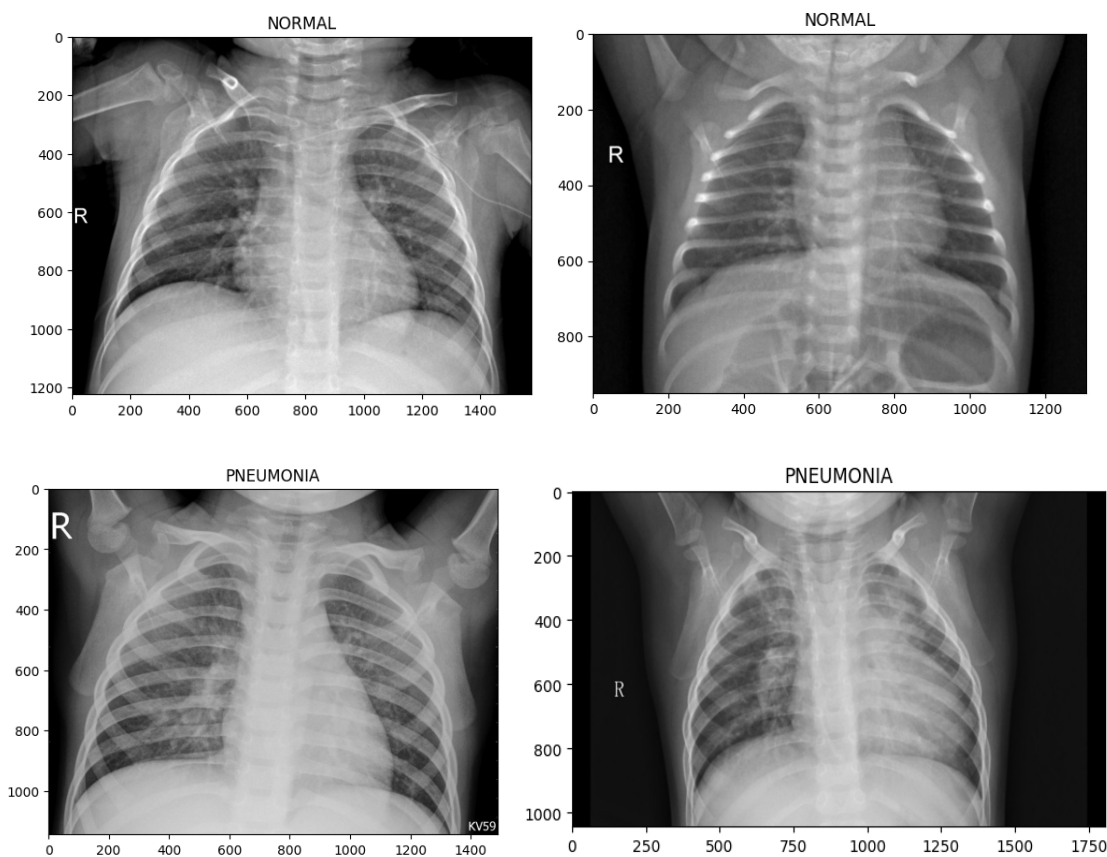
*	Cifar10	Mnist	Fashion Mnist
SCNNB ACC	76.79%	99.05%	91.61%
SCNNB-a ACC	76.97%	98.99%	91.61%
SCNNB-b ACC	73.90%	98.74%	91.63%

جدول ۱.۹. دقت هر دیتا ست بر روی دادگان ارزیابی

مشاهده میشود مدل های SCNNB و SCNNB-a بهتر از SCNNB-b عمل کرده اند. همچنین نمودار loss پس از کاهش افزایش یافته است که این نشان دهنده overfit کردن مدل روی داده train است. همچنین مدل های SCNNB و SCNNB-a با سرعت بیشتری نسبت به SCNNB-b به دقت بهینه رسیده اند بنابراین وجود لایه BN در مدل میتواند باعث شود فیچر ها بهتر و سریع تر یاد گرفته شوند.

پاسخ 2. طبقه‌بندی تصاویر اشعه ایکس قفسه سینه

2-1. آماده سازی و پیش پردازش داده ها
ابتدا از داده ها سمپل گرفته و آنها را رویت میکنیم.



شکل 2.1. نمونه تصویر

الف.

در فاز preprocess مقاله ابتدا تصاویر را resize می‌کند و سایز آنها را به $128 * 128$ تبدیل می‌کند.

سپس از data augmentation استفاده می‌کند. data augmentation فرایند تولید داده های جدید و مصنوعی برای آموزش مدل های یادگیری عمیق است که به دلایل مختلفی ممکن است در دسترس نباشد. بطور کلی، داده افزایی شامل تغییرات کوچکی در داده های اصلی می باشد تا

مجموعه داده ها بیشتر، متنوع تر، و قابل استفاده تر شود. این تغییرات شامل تغییر اندازه، چرخش، انعکاس، تغییر رنگ، افزودن نویز و... می شوند.

این روش می تواند کیفیت را بهبود و دقت مدل را افزایش دهد و برای جلوگیری از بیش پردازش و حفظ تعمیم پذیری مدل مفید است. همچنین، در مواقعی که حجم داده های موجود کم است، data augmentation می تواند به عنوان یک راه حل برای تولید مجموعه داده های بزرگتر و متنوع تر مورد استفاده قرار گیرد.

همچنین یکی از مراحل ذکر شده در مقاله استفاده از class weights برای حل مشکل class imbalance است. هنگامی که تعداد نمونه های هر کلاس در یک مجموعه داده بسیار متفاوت است، موجب ایجاد عدم تعادل کلاسی یا class imbalance می شود. این مشکل می تواند باعث کاهش دقت و کارایی مدل های یادگیری عمیق شود، زیرا مدل به سمت کلاس بیشتری که داده بیشتری دارد، جا به جا می شود و به کلاس کمتر توجه کمتری می کند.

برای حل مشکل عدم تعادل کلاسی، از روش وزن بندی کلاس یا class weights در آموزش مدل استفاده می شود. با این روش، وزن هر کلاس متناسب با تعداد نمونه های آن در داده ها تعیین می شود. به عنوان مثال، اگر یک کلاس دارای ۱۰۰ نمونه باشد و کلاس دیگر دارای ۱۰۰۰ نمونه باشد، وزن اولی باید ۱۰ برابر وزن دومی باشد تا مدل بتواند به خوبی از دو کلاس استفاده کند. در کل، با استفاده از class weights می توان عملکرد مدل را در مواجهه با مسئله عدم تعادل کلاسی بهبود بخشید و دقت و کارایی آن را افزایش داد.

ب.

ابتدا مجموعه دادگان را از سایت kaggle دانلود کرده و در Drive ذخیره می کنیم. سپس همه عکس ها را به یک پوشه منتقل کرده و مجدداً با درصدهای گفته شده split می کنیم (60% برای دادگان آموزش، ۲۰% برای دادگان ارزیابی و ۲۰% برای دادگان اعتبارسنجی). اکنون داده ها برای مراحل بعد آماده هستند.

2-2. توضیح لایه های مختلف معماری شبکه

مدل داده شده EfficientNet دارای لایه های زیراست:

1. global_average_pooling2d :

در این لایه، بعد از اجرای عملیات پیچش بر روی تصاویر و استخراج ویژگی‌های مختلف، تمام ویژگی‌های استخراج شده را با هم ترکیب می‌کنیم و به شکل یک بردار با اندازه ثابت تبدیل می‌کنیم. به این صورت که از هر یک از ویژگی‌های استخراج شده، میانگین مقادیر آن در هر کانال محاسبه می‌شود. سپس به جای این که این ویژگی‌ها را به یک شبکه کاملاً متصل (Fully Connected) وارد کنیم، مقدار این میانگین‌ها به عنوان خروجی این لایه در نظر گرفته می‌شود. این عملیات باعث کاهش تعداد پارامترها، افزایش سرعت یادگیری و جلوگیری از بیش‌برازشی در شبکه‌های پیچشی می‌شود.

2. Dense:

در این لایه، هر یک از ورودی‌های شبکه با همه وزن‌های لایه متصل می‌شود و خروجی به عنوان ورودی لایه بعدی ارائه می‌شود. به این صورت که تمام ورودی‌ها به یکدیگر وصل می‌شوند و در نهایت یک بردار خروجی با ابعاد مشخص تولید می‌شود.

در واقع، لایه Dense می‌تواند به عنوان یک لایه کاملاً متصل برای طبقه‌بندی، رگرسیون یا هر وظیفه‌ای که به دنبال حل آن هستیم، استفاده شود. این لایه در شبکه‌های عمیق، بعد از استفاده از لایه‌های پیچشی و ترکیبی، به عنوان لایه نهایی استفاده می‌شود.

3. لایه dropout:

این لایه در هر مرحله از آموزش، با احتمال مشخصی برخی از نورون‌ها را به صورت تصادفی غیرفعال می‌کند. این کار باعث کاهش پدیده‌ی برازش بیش‌ازحد (overfitting) در شبکه می‌شود.

دلیل انتخاب کردن EfficientNet:

یکی از مشکلاتی که در تشخیص ذات الیه با استفاده از تصاویر پزشکی، مانند اشعه ایکس، وجود دارد، حجم بالای داده‌های موجود است. برای مثال، ممکن است مجموعه داده شامل هزاران عکس از بیماران با شرایط مختلف باشد. بنابراین، شبکه‌های عصبی باید قادر باشند اطلاعات مفید را از داده‌های بسیار بزرگ استخراج کنند و با استفاده از این اطلاعات، بتوانند به طور دقیقی حالت بیماری را تشخیص دهند.

EfficientNet یک مدل شبکه عصبی پیشرفته است که برای پردازش تصاویر طراحی شده است و برای کاهش تعداد پارامترها و افزایش دقت و سرعت آموزش، از رویکردی به نام "Compound Scaling" استفاده می‌کند. به عبارت دیگر، با اعمال یک روش معقول و هوشمند برای تغییر ابعاد شبکه، این مدل می‌تواند با حفظ دقت بالا، تعداد پارامترها و سائز شبکه را کاهش دهد.

با توجه به مزایای این مدل، استفاده از EfficientNet برای تشخیص ذات الریه با استفاده از عکس‌های اشعه ایکس می‌تواند بهبود قابل توجهی در دقت و سرعت شناسایی بیماری داشته باشد. به علاوه، با توجه به پیچیدگی و ابعاد بالای داده‌های پزشکی، کاهش تعداد پارامترها و سائز شبکه می‌تواند به طور موثری از لحاظ حجم داده و زمان آموزش این مسئله را حل کند.

2-3. پیاده‌سازی شبکه

شبکه را با ویژگی گفته شده و Hyper-parameter های بهینه در مقاله پیاده‌سازی می‌کنیم.

2-4. نتایج پیاده‌سازی

الف.

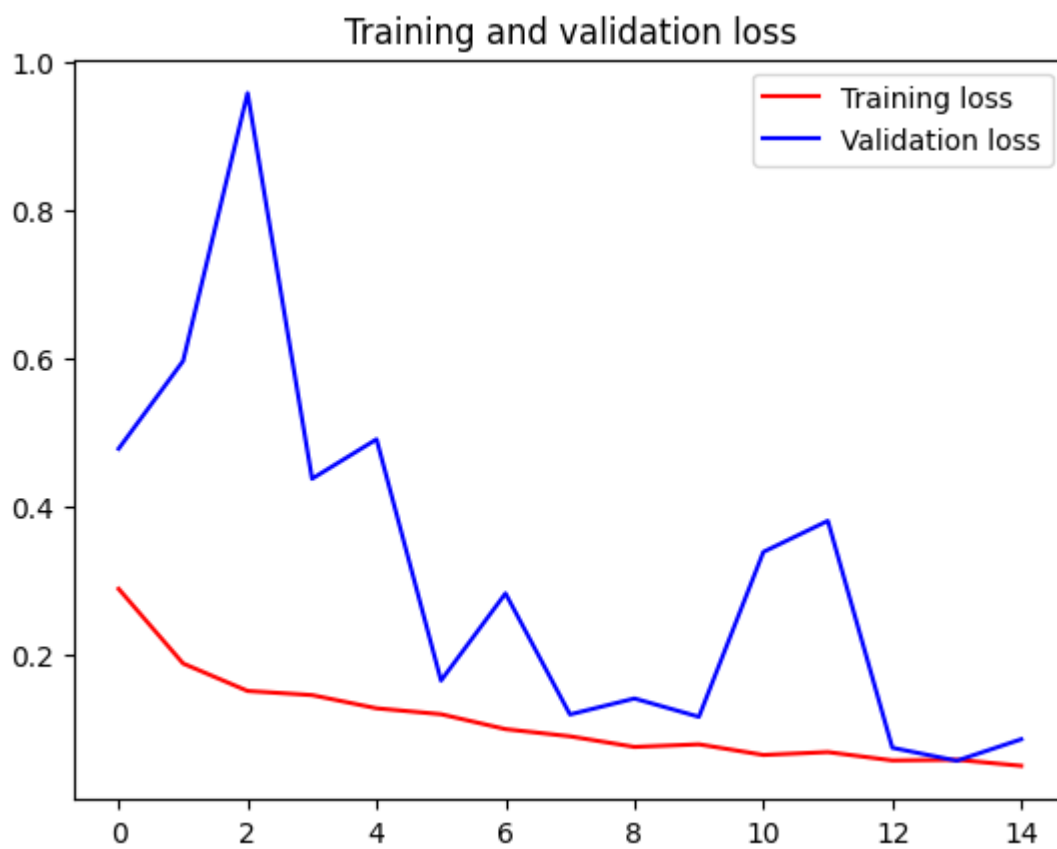
```
➡ ---- TRAIN ----  
Accuracy : 0.9828388690948486  
F1-score : 0.9947749376296997  
Precision : 0.9817532896995544  
---- VALIDATION ----  
Accuracy : 0.9711790680885315  
F1-score : 0.9927623867988586  
Precision : 0.9682353138923645
```

شکل 2.2. Acc, F1-score and Precision

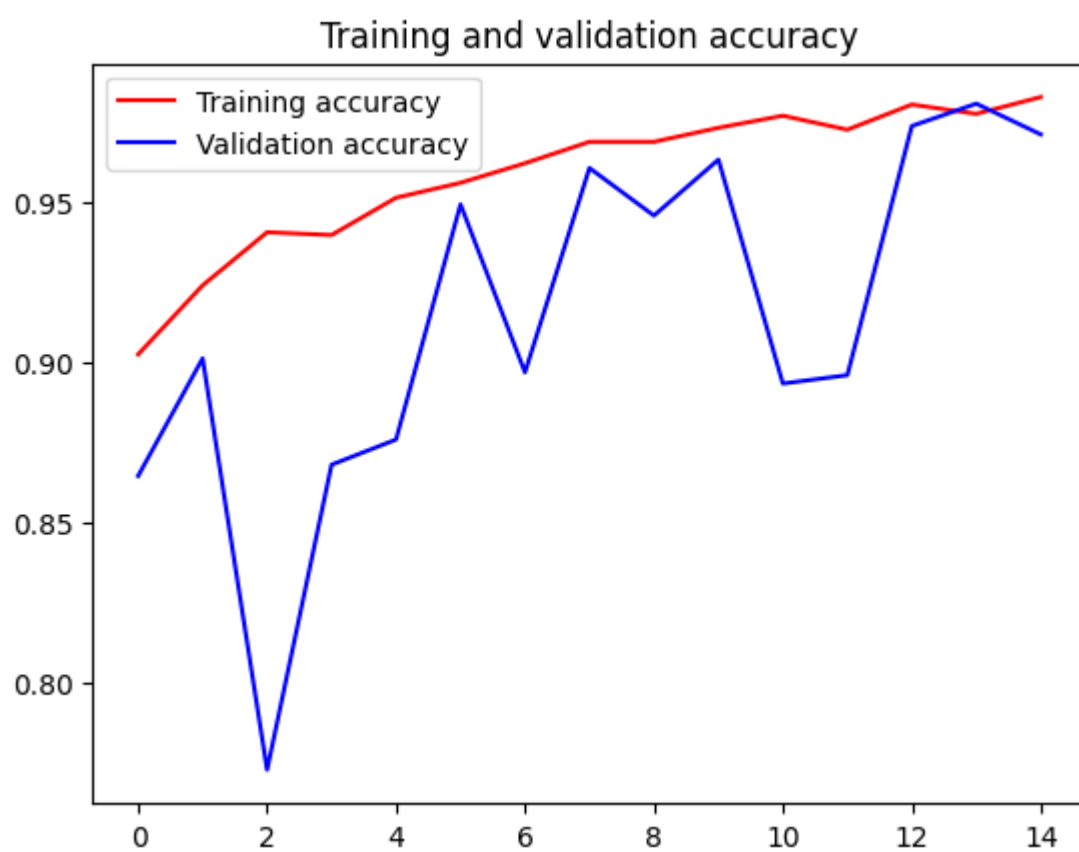
	precision	recall	f1-score	support
0	0.98	0.89	0.93	317
1	0.96	0.99	0.98	855
accuracy			0.96	1172
macro avg	0.97	0.94	0.95	1172
weighted avg	0.96	0.96	0.96	1172

شکل 2.3. تصویر دوم Acc, F1-score and Precision

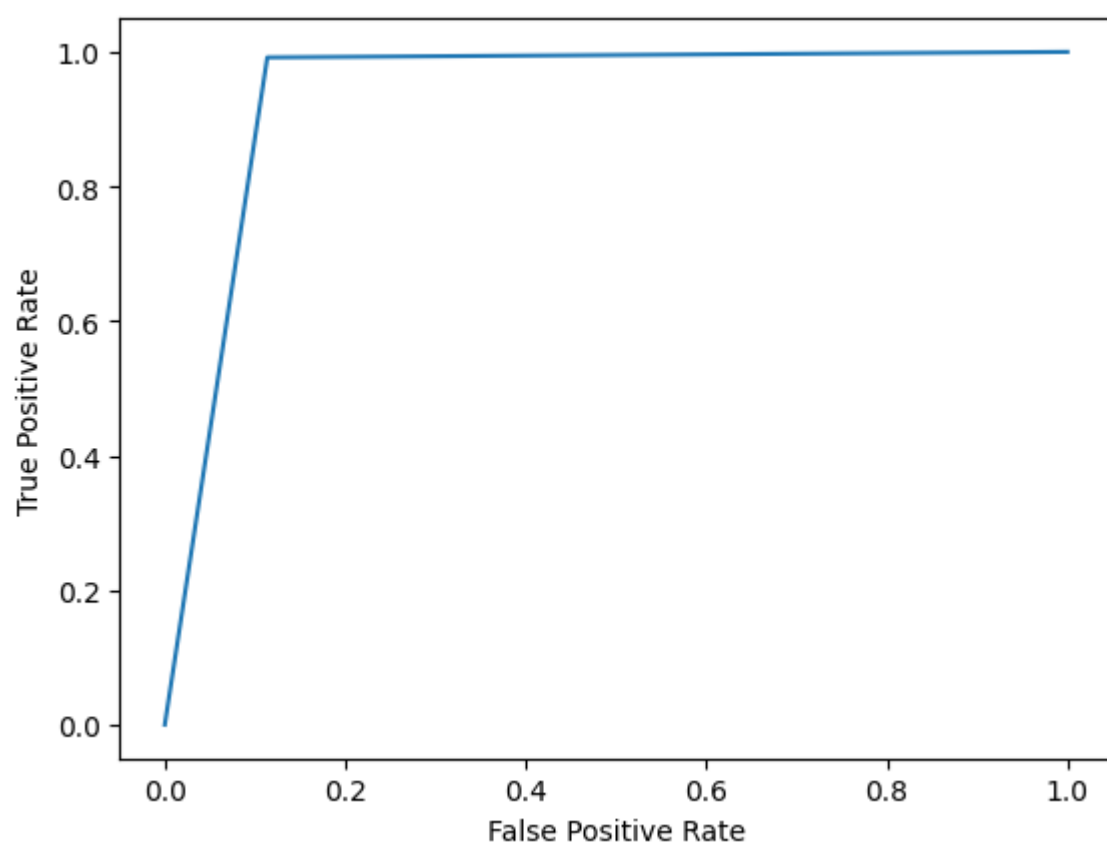
ب.



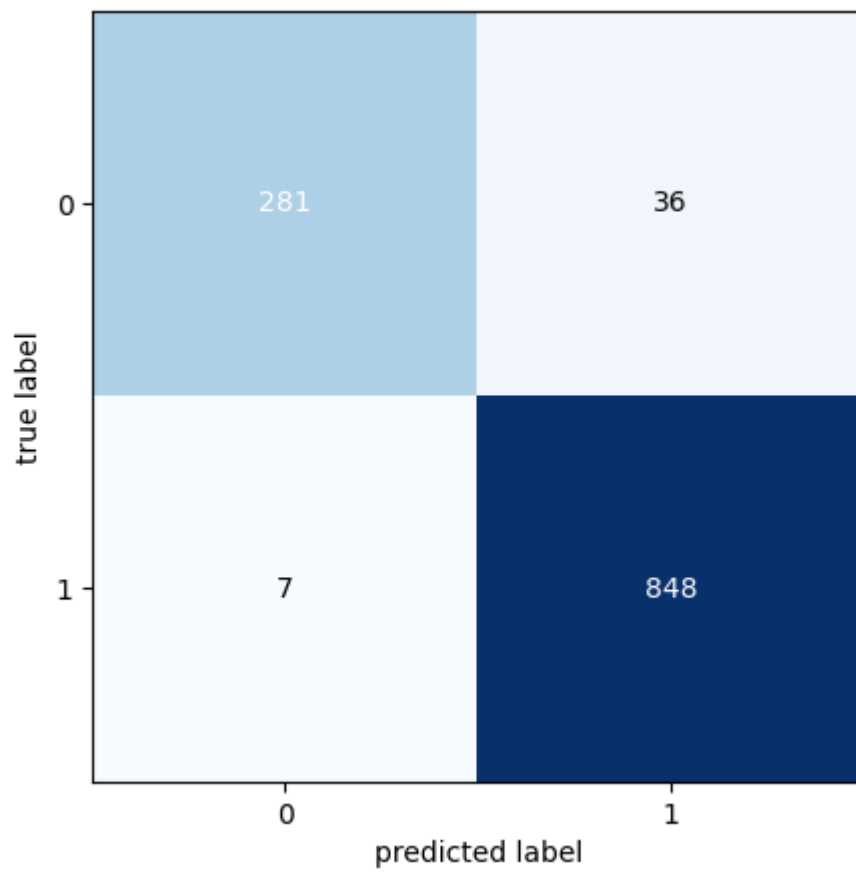
شکل 2.4. نمودار loss



شکل 2.5. نمودار accuracy



شکل 2.6. نمودار ROC curve



شکل 2.7. نمودار confusion matrix

ج.