

به نام خدا



تمرین چهارم
(شبکه های عصبی)
هوش مصنوعی و یادگیری ماشین

نام دانشجو:

امیرعلی محمودزاده طوسی

۸۱۰۶۰۳۱۴۲

استاد درس:

دکتر شریعت پناهی

خرداد ۱۴۰۴



https://github.com/amiralimt/AI_hw4/

کلیات پروژه و ساختار آن

تمرین شامل سه بخش اصلی است که هر کدام مراحل مشخصی دارند:

بخش اول: دادگان UCI HAR – تشخیص فعالیت‌های انسانی با موبایل

- داده‌ها: یک دیتاست برداری ماتریس $(n \times 561)$ ، شش کلاس فعالیت فیزیکی.
- هدف: طراحی و مقایسه دو مدل MLP و CNN برای طبقه‌بندی داده.
- مراحل:
 ۱. نرمال‌سازی داده‌ها و تفکیک آموزش/آزمون.
 ۲. طراحی شبکه MLP با حداکثر دو لایه پنهان.
 ۳. طراحی یک CNN دلخواه با لایه‌های کانولوشن، Pooling، Dropout و BatchNorm.
 ۴. آموزش هر دو مدل با Adam و learning rate برابر ۰.۰۰۱.
 ۵. ارزیابی هر مدل با نمودار دقت/خطا، دقت نهایی، ماتریس آشفتگی.
 ۶. تحلیل اینکه کدام مدل بهتر است و چرا.

بخش دوم: دادگان NEU Surface Defects – عیوب سطحی فولاد

- داده‌ها: تصاویر خاکستری 200×200 با ۶ کلاس عیب سطحی، ۱۸۰۰ تصویر متوازن (۳۰۰ برای هر کلاس).
- هدف: طراحی و مقایسه مدل‌های MLP و CNN.
- مراحل مشابه بخش اول است، با تأکید بیشتر بر روی عملکرد CNN نسبت به MLP در تصاویر
 - Block Dropout به جای Dropout
 - Kernel Factorization در CNN
- تحلیل نهایی برای تعیین مدل بهتر

بخش سوم: یادگیری انتقالی با ResNet-50

- هدف: استفاده از **Transfer Learning** با شبکه آموزش دیده‌ی ResNet-50 برای طبقه‌بندی تصاویر NEU

- مراحل:

۱. **Data Augmentation** برای تقویت داده (چرخش، نویز، برچسب‌گذاری تصادفی...)

۲. بارگذاری ResNet-50 و حذف لایه نهایی و اضافه کردن لایه جدید Dense+Softmax

۳. دو مرحله آموزش:

- مرحله اول: فقط لایه جدید آموزش داده شود (بقیه فریز).
- مرحله دوم (Fine-tuning): برخی لایه‌های انتهایی آزاد شوند.

۴. ارزیابی مدل (دقت، خطا، ماتریس آشفتگی).

۵. مقایسه با مدل CNN قبلی از نظر:

- زمان آموزش
- اندازه مدل
- دقت نهایی
- مزایا و معایب TL نسبت به آموزش از صفر

بخش اول: دادگان تشخیص فعالیت‌های انسانی با گوشی (UCI HAR)

برای حل این تمرین از نوت بوک Google Colab استفاده خواهیم کرد، و تنظیمات Runtime type را به T4 GPU به جهت بهره گیری از قدرت گرافیکی و افزایش سرعت پردازش ها تغییر می دهیم.

فایل های هر دو Dataset را روی گوگل درایو آپلود میکنیم و کولب را به درایو متصل می کنیم تا برای دسترسی از به فایل ها آنها را فراخوانی کنیم.

بعد از ورود فایل آنها را از حالت فشرده خارج و دوباره در مسیر گوگل درایو ذخیره میکنیم.

فایل های آموزش و آزمون را از محل پوشه استخراج شده بارگذاری میکنیم و شکل داده ها را نمایش می دهیم که به صورت زیر است:

X_train: (7352, 561) | y_train: (7352, 1)

X_test : (2947, 561) | y_test : (2947, 1)

۱-۱- آماده سازی داده ها

- روش های مختلف نرمال سازی را معرفی کرده و مزایا و معایب هر یک را بیان کنید.
- یکی از این روش ها را انتخاب و پیاده سازی نمایید.

➤ Min-Max Scaling (مقیاس بندی بازه ای)

داده ها را به بازه $[0, 1]$ یا $[a, b]$ نگاشت می کند.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

مزایا:

- ساده و سریع
- برای شبکه هایی با توابع فعال مانند Sigmoid خوب کار می کند.

معایب:

- به شدت به داده های پرت (outliers) حساس است.
- اگر min/max در تست با آموزش فرق کند، مشکل ایجاد می کند

➤ Z-score Scaling (Standardization)

- StandardScaler داده‌ها را طوری نرمال می‌کند که میانگین صفر و انحراف معیار ۱ داشته باشند

$$x_{scaled} = \frac{\mu - x}{\sigma}$$

مزایا:

- برای داده‌های با توزیع نرمال یا نزدیک به نرمال بسیار مناسب است.
- در برابر outlier ها مقاوم‌تر از MinMax است.
- در مدل‌هایی مانند SVM، رگرسیون و شبکه‌های عصبی خوب جواب می‌دهد.

معایب:

- ممکن است در داده‌های شدیداً نامتوازن (بازه زیاد) نوسان ایجاد کند.

➤ Robust Scaler

- به جای میانگین و واریانس، از میانه (median) و IQR استفاده می‌کند.

مزایا:

- بهترین گزینه برای داده‌های دارای outlier

معایب:

- اطلاعات بعضی از ویژگی‌ها را ممکن است کاهش دهد.

➤ MaxAbsScaler

- فقط به حداکثر قدر مطلق مقیاس می‌شود، خروجی در بازه $[-1, 1]$

مزایا:

- برای داده‌هایی که sparse یا متناظر با صفر زیاد هستند، مفید است.

معایب:

- فقط برای داده‌های غیرمنفی مناسب نیست.

در دیتاست UCI HAR

- داده‌ها پردازش شده و نسبتاً استانداردند
 - توزیع نرمال یا تقریباً نرمال داریم
 - می‌خواهیم با شبکه‌های عصبی MLP و CNN کار کنیم
- بنابراین، روش انتخابی (StandardScaler (Z-score Scaling) هست چون:
- عملکرد خوبی در شبکه‌های عصبی دارد.
 - نسبت به outlier ها مقاومت معقول دارد.
 - برای داده‌های real-valued نرمال‌سازی خوبی انجام می‌دهد.
- بعد از نرمال‌سازی داده‌ها باید داده‌ها را one-hot encoding کنیم:

One-Hot Encoding یعنی تبدیل یک عدد دسته‌ای (کلاس) به یک بردار دودویی که فقط یکی از درایه‌ها ۱ هست و بقیه صفر.

برای طبقه‌بندی چندکلاسه، برچسب‌های عددی به‌صورت one-hot تبدیل شدند تا با خروجی softmax شبکه مطابقت داشته باشند و امکان استفاده از تابع خطای categorical_crossentropy فراهم شود. این تبدیل همچنین از ایجاد رابطه نادرست عددی بین کلاس‌ها جلوگیری می‌کند.

دیتاست UCI HAR از قبل با نسبت حدود ۷۱٪ به ۲۹٪ تقسیم شده است، ما با توجه به خواسته تمرین داده‌ها از ادغام می‌کنیم و سپس با نسبت ۱۵/۸۵ تقسیم می‌کنیم:

ادغام X_train و X_test که از قبل نرمال شده بودند، ادغام برچسب‌های one-hot، y_train_cat و y_test_cat، تقسیم دوباره به ۸۵٪ آموزش و ۱۵٪ تست با استفاده از تابع train_test_split از کتابخانه‌ی scikit-learn با حفظ نسبت کلاس‌ها stratify

کاربرد	متغیر	ابعاد
ویژگی‌های آموزش	X_train_final	(8754, 561)
لیبل آموزش (one-hot)	y_train_final	(8754, 6)
ویژگی‌های تست	X_test_final	(1545, 561)
لیبل تست (one-hot)	y_test_final	(1545, 6)

۱-۲- طراحی شبکه MLP

شبکه‌ی MLP طراحی شده شامل دو لایه پنهان با اندازه‌های ۱۲۸ و ۶۴ نرون و تابع فعال‌سازی ReLU برای هر دو لایه است. برای جلوگیری از بیش‌برازش، یک لایه Dropout با نرخ ۰.۳ بین آن‌ها اضافه شده است. در لایه خروجی، با توجه به ماهیت چندکلاسه بودن مسئله، از تابع Softmax استفاده شده است تا احتمال تعلق هر نمونه به یکی از شش کلاس خروجی مشخص شود.

Input (۵۶۱ ویژگی)

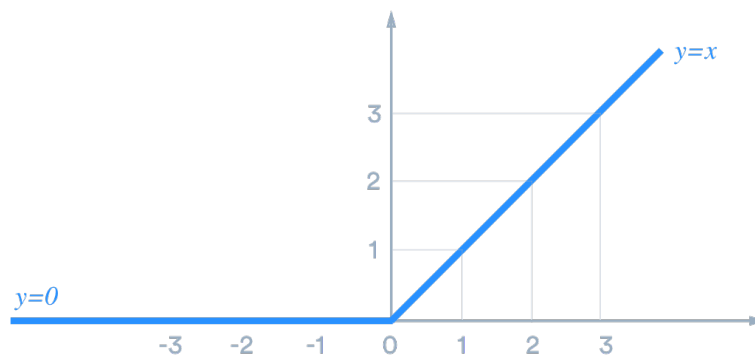
→Dense(128), ReLU

→Dropout(۰.۳)

→Dense(64), ReLU

→Dense(6), Softmax (خروجی)

لایه‌های پنهان (ReLU (Rectified Linear Unit) :



$$ReLU(x) = \max(0, x)$$

ساده و سریع، کمک به کاهش اثر محو شدن گرادیان، در عمل عملکرد خیلی خوبی در لایه‌های پنهان شبکه‌های عصبی دارد.

لایه خروجی Softmax

چون ما یک مسئله طبقه‌بندی چندکلاسه داریم (۶ کلاس)، از تابع softmax استفاده می‌کنیم تا مدل برای هر کلاس یک احتمال خروجی بدهد.

خروجی به صورت بردار مثل $[0.1, 0.03, 0.5, 0.01, 0.3, 0.06]$ هست که جمعش ۱ می‌شود.

Dropout

به عنوان regularization استفاده می‌شود تا مدل overfit نکند (یعنی فقط داده آموزش رو حفظ نکند و روی داده تست ضعیف باشد). مقدار ۰.۳ یعنی در هر epoch، حدود ۳۰٪ نرون‌ها غیرفعال می‌شوند.

Model: "sequential"		
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 128)	71936
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8256
dense_2 (Dense)	(None, 6)	390
=====		
Total params: 80,582		
Trainable params: 80,582		
Non-trainable params: 0		

۱-۳-طراحی شبکه CNN

با وجود برداری بودن داده‌های UCI HAR ، یک شبکه‌ی CNN با استفاده از لایه‌های Conv1D طراحی شد. در این معماری، از دو لایه کانولوشن با اندازه کرنل ۵ و ۳ و تعداد فیلترهای ۳۲ و ۶۴ استفاده شده است. پس از هر لایه کانولوشن، لایه Batch Normalization و MaxPooling و Dropout برای افزایش پایداری، کاهش ابعاد و جلوگیری از بیش‌برازش به کار رفته‌اند. در پایان، ویژگی‌ها با Flatten به یک بردار تبدیل شده و به دو لایه Dense ختم می‌شود که خروجی Softmax با ۶ نود نمایانگر شش کلاس فعالیت است.

شبکه CNN نیاز به ورودی سه بعدی دارد، داده‌های UCI HAR برداری (۵۶۱ ویژگی) هستند، نه تصویری اما باز هم می‌توانیم یک CNN روی این داده‌ها طراحی کنیم، به شرطی که شکل ورودی رو طوری تغییر بدهیم که برای لایه‌های کانولوشن قابل استفاده باشد، پس داده های ورودی را reshape می کنیم.

```
Input(۱, ۵۶۱):
→Conv1D(filters=32, kernel_size=5, activation='relu')
→BatchNormalization
→MaxPooling1D(pool_size=2)
→Dropout(۰.۳)

→Conv1D(filters=64, kernel_size=3, activation='relu')
→BatchNormalization
→MaxPooling1D(pool_size=2)
```

→Dropout(۰.۳)

→Flatten

→Dense(64, activation='relu')

→Dense(6, activation='softmax')

دلیل	لایه
استخراج الگوهای محلی (local features) در بردار	Conv1D
تثبیت و تسریع آموزش	BatchNormalization
کاهش ابعاد و پارامترها	MaxPooling1D
جلوگیری از overfitting	Dropout
طبقه‌بندی ۶ کلاس نهایی	Dense + Softmax

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 557, 32)	192
batch_normalization (BatchNormalization)	(None, 557, 32)	128
max_pooling1d (MaxPooling1D)	(None, 278, 32)	0
dropout_1 (Dropout)	(None, 278, 32)	0
conv1d_1 (Conv1D)	(None, 276, 64)	6208
batch_normalization_1 (BatchNormalization)	(None, 276, 64)	256
max_pooling1d_1 (MaxPooling1D)	(None, 138, 64)	0
dropout_2 (Dropout)	(None, 138, 64)	0
flatten (Flatten)	(None, 8832)	0
dense_3 (Dense)	(None, 64)	565312
dense_4 (Dense)	(None, 6)	390
Total params: 572,486		
Trainable params: 572,294		
Non-trainable params: 192		

۴-۱- آموزش شبکه ها

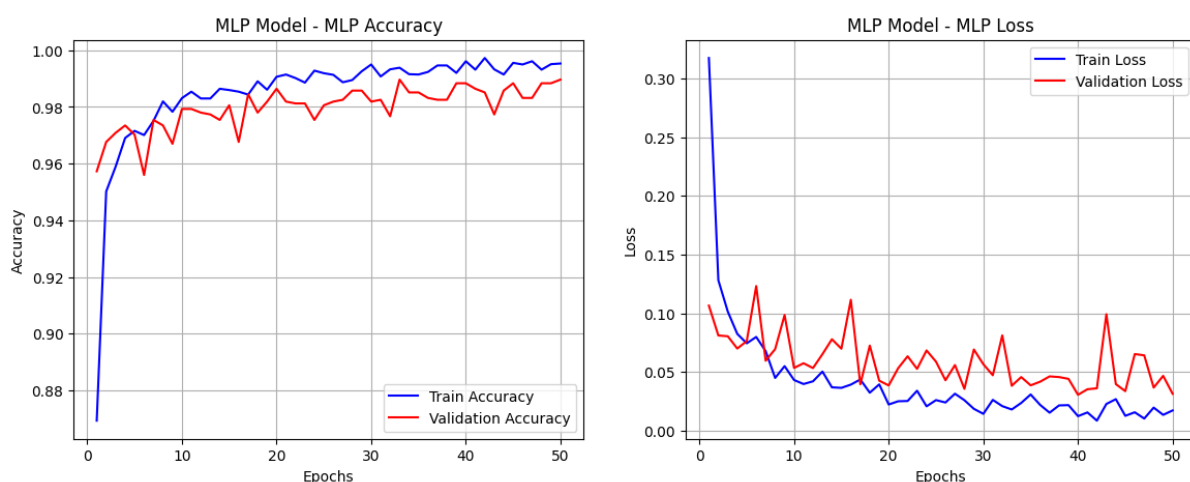
هر دو مدل با استفاده از الگوریتم Adam و نرخ یادگیری ۰.۰۰۱ آموزش داده شدند. در هر مرحله، دقت و خطای آموزش و اعتبارسنجی ثبت شده و برای ارزیابی و مقایسه در مراحل بعدی از آن‌ها استفاده خواهد شد.

مدل‌های Keras/TensorFlow ترجیح می‌دهند ورودی‌ها float32 باشند، نه float64. در اکثر مواقع float64 باعث ناسازگاری یا کندی می‌شود و حتی ممکنه در نسخه‌های خاص باعث خطا شود، پس نوع داده‌ها را به float32 تغییر می‌دهیم.

تعداد اپیاک‌ها ۵۰ و مقدار بچ سائز ۳۲ برای هر دو مدل تنظیم شد.

۵-۱- ارزیابی

نمودارهای خطا (Loss) و دقت (Accuracy) را برای داده‌های آموزش و آزمون برای مدل MLP رسم شد:



این نمودارها عملکرد مدل شبکه عصبی پرسپترون چندلایه (MLP) را در طول ۵۰ دوره (Epoch) آموزش نشان می‌دهند. نمودار سمت چپ دقت (Accuracy) و نمودار سمت راست میزان خطا (Loss) را برای داده‌های آموزشی (Train) و داده‌های اعتبارسنجی (Validation) نمایش می‌دهد.

تحلیل نمودارها

۱. نمودار دقت مدل (MLP Model - MLP Accuracy) - سمت چپ

- دقت آموزشی (Train Accuracy - خط آبی): این خط نشان می‌دهد که مدل با چه دقتی داده‌های آموزشی را یاد می‌گیرد. دقت آموزشی از حدود ۰.۸۷ شروع شده و به سرعت افزایش می‌یابد و تا دوره پنجم به بالای ۰.۹۶ می‌رسد. سپس با نوساناتی به روند افزایشی خود ادامه داده و در اواخر به نزدیکی ۱.۰۰ (یا ۱۰۰٪) می‌رسد و به نظر می‌رسد در یک سطح بالا تثبیت شده است.

- دقت اعتبارسنجی (Validation Accuracy - خط قرمز): این خط نشان می‌دهد که مدل با چه دقتی داده‌هایی را که قبلاً ندیده (داده‌های اعتبارسنجی) پیش‌بینی می‌کند. دقت اعتبارسنجی از حدود ۰.۹۶ شروع شده و روندی مشابه دقت آموزشی دارد اما معمولاً کمی پایین‌تر از آن قرار می‌گیرد. این دقت نیز نوسانات بیشتری نسبت به دقت آموزشی نشان می‌دهد و به نظر می‌رسد در حوالی ۰.۹۸ تا ۰.۹۹ به یک سقف رسیده و در دوره‌های پایانی کمی نوسان دارد یا حتی اندکی کاهش می‌یابد.

- روند کلی دقت: هر دو دقت در دوره‌های اولیه به سرعت بهبود پیدا می‌کنند. دقت آموزشی به طور مداوم بالاتر از دقت اعتبارسنجی باقی می‌ماند. شکاف بین این دو خط، به خصوص در دوره‌های پایانی، می‌تواند نشانه‌ای از بیش‌برازش (Overfitting) باشد، یعنی مدل به جای یادگیری الگوهای کلی، داده‌های آموزشی را "حفظ" کرده است.

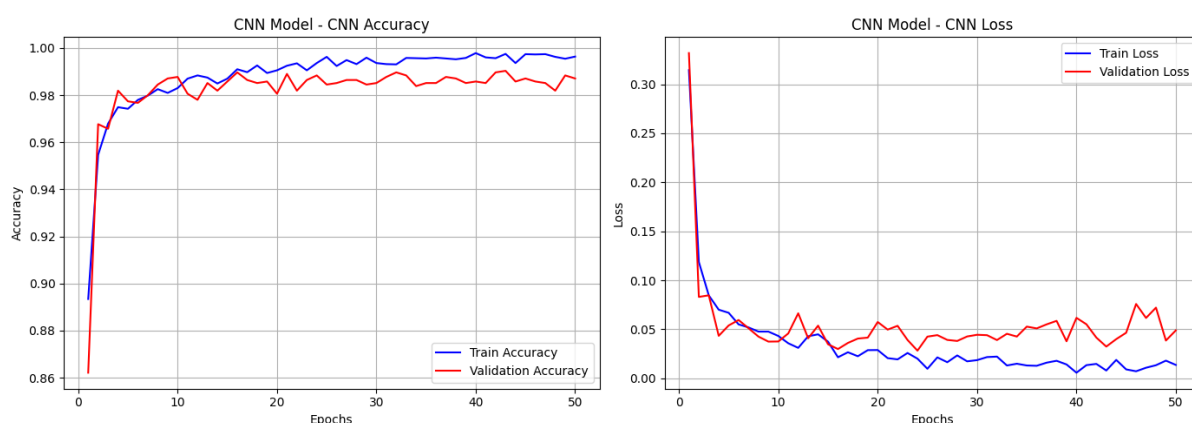
۲. نمودار خطای مدل (MLP Model - MLP Loss) - سمت راست

- خطای آموزشی (Train Loss - خط آبی): این خط میزان خطای مدل روی داده‌های آموزشی را نشان می‌دهد. خطا از مقدار بالایی (بیش از ۰.۳۰) شروع شده و در چند دوره اول به شدت کاهش می‌یابد (تا دوره پنجم به زیر ۰.۱۰ می‌رسد). سپس با نرخ کمتری به کاهش خود ادامه داده و در انتها به مقدار بسیار پایینی (نزدیک به ۰.۰۱-۰.۰۲) می‌رسد.
- خطای اعتبارسنجی (Validation Loss - خط قرمز): این خط میزان خطای مدل روی داده‌های اعتبارسنجی را نشان می‌دهد. خطا از حدود ۰.۱۱ شروع شده و در ابتدا همزمان با خطای آموزشی کاهش می‌یابد و به کمترین مقدار خود در حدود دوره‌های ۵ تا ۱۰ (حدود ۰.۰۲-۰.۰۴) می‌رسد. نکته مهم این است که پس از این نقطه، خطای اعتبارسنجی شروع به افزایش و نوسان شدید می‌کند، در حالی که خطای آموزشی همچنان در حال کاهش است. این پدیده نشانه‌ای از بیش‌برازش (Overfitting) است.
- روند کلی خطا: خطای آموزشی به طور پیوسته کاهش می‌یابد. خطای اعتبارسنجی ابتدا کاهش یافته اما سپس روندی افزایشی و بی‌ثبات پیدا می‌کند.
- یادگیری اولیه: مدل در دوره‌های ابتدایی (تقریباً تا دوره ۵-۱۰) به خوبی در حال یادگیری است، زیرا هم دقت افزایش و هم خطا کاهش می‌یابد (هم برای داده‌های آموزشی و هم اعتبارسنجی).
- بیش‌برازش (Overfitting): از حدود دوره ۱۰ به بعد، نشانه‌های بیش‌برازش قابل مشاهده است. خطای آموزشی همچنان کاهش می‌یابد اما خطای اعتبارسنجی شروع به افزایش می‌کند. این بدان

معناست که مدل در حال یادگیری جزئیات و نویزهای داده‌های آموزشی است که به داده‌های جدید قابل تعمیم نیستند.

- **عملکرد مدل:** با وجود بیش‌برازش، مدل به دقت بالایی روی داده‌های آموزشی (نزدیک به ۱۰۰٪) و همچنین دقت خوبی روی داده‌های اعتبارسنجی (حداکثر حدود ۹۹٪) دست یافته است. با این حال، به دلیل بیش‌برازش، عملکرد واقعی مدل روی داده‌های کاملاً جدید ممکن است کمتر از چیزی باشد که دقت اعتبارسنجی در دوره‌های پایانی نشان می‌دهد و بیشتر شبیه به عملکرد در نقطه‌ای است که خطای اعتبارسنجی شروع به افزایش کرده است.

نمودارهای خطا (Loss) و دقت (Accuracy) را برای داده‌های آموزش و آزمون برای مدل CNN رسم شد:



این نمودارها عملکرد یک مدل شبکه عصبی کانولوشنی (CNN) را در طول ۵۰ دوره (Epoch) آموزش نشان می‌دهند. نمودار سمت چپ دقت (Accuracy) و نمودار سمت راست میزان خطا (Loss) را برای داده‌های آموزشی (Train) و داده‌های اعتبارسنجی (Validation) نمایش می‌دهد.

تحلیل نمودارها

۱. نمودار دقت مدل (CNN Model - CNN Accuracy) - سمت چپ

- دقت آموزشی (Train Accuracy - خط آبی): این خط نشان می‌دهد که مدل با چه دقتی داده‌های آموزشی را یاد می‌گیرد. دقت آموزشی از حدود ۰.۸۸ شروع شده و به سرعت افزایش می‌یابد و تا دوره پنجم به حدود ۰.۹۷ می‌رسد. سپس با نوساناتی به روند افزایشی خود ادامه داده و پس از حدود دوره ۲۰-۲۵ در سطح بسیار بالایی (نزدیک به ۰.۹۹۵ یا ۹۹.۵٪) تثبیت می‌شود.
- دقت اعتبارسنجی (Validation Accuracy - خط قرمز): این خط نشان می‌دهد که مدل با چه دقتی داده‌هایی را که قبلاً ندیده (داده‌های اعتبارسنجی) پیش‌بینی می‌کند. دقت اعتبارسنجی از حدود ۰.۸۶ شروع شده و همچنین به سرعت افزایش می‌یابد و تا دوره پنجم به نزدیکی ۰.۹۸

می‌رسد. پس از آن، با نوسانات بیشتری نسبت به دقت آموزشی، به افزایش خود ادامه می‌دهد و چندین بار به قله‌هایی در حدود ۰.۹۹ می‌رسد. در دوره‌های پایانی، با اینکه دقت آموزشی بسیار بالا و پایدار است، دقت اعتبارسنجی نوسانات بیشتری دارد (عموماً بین ۰.۹۸ و ۰.۹۹) و کاملاً به پایداری و سطح دقت آموزشی نمی‌رسد. یک شکاف کوچک اما مداوم بین دقت آموزشی و اعتبارسنجی پس از افزایش اولیه وجود دارد.

- روند کلی دقت: هر دو دقت به سرعت در دوره‌های اولیه بهبود پیدا می‌کنند. دقت آموزشی پس از چند دوره ابتدایی، به طور مداوم کمی بالاتر از دقت اعتبارسنجی قرار می‌گیرد.

۲. نمودار خطای مدل (CNN Model - CNN Loss) - سمت راست

- خطای آموزشی (Train Loss - خط آبی): این خط میزان خطای مدل روی داده‌های آموزشی را نشان می‌دهد. خطا از مقدار بسیار بالایی (احتمالاً بالاتر از ۰.۳۰) شروع شده و در چند دوره اول به شدت کاهش می‌یابد (تا دوره دوم به زیر ۰.۱۰ و تا دوره پنجم به زیر ۰.۰۵ می‌رسد). سپس با نرخ بسیار کمتری به کاهش خود ادامه داده و با نوسانات جزئی، در انتها به مقدار بسیار پایینی (حدود ۰.۰۱-۰.۰۲) می‌رسد.
- خطای اعتبارسنجی (Validation Loss - خط قرمز): این خط میزان خطای مدل روی داده‌های اعتبارسنجی را نشان می‌دهد. خطا از حدود ۰.۱۰ شروع شده، در دوره اول به شدت کاهش یافته و سپس با سرعت کمتری به کاهش ادامه می‌دهد تا در حدود دوره‌های ۱۰ تا ۱۵ به کمترین مقدار خود (حدود ۰.۰۳-۰.۰۴) برسد. نکته مهم این است که پس از این نقطه، خطای اعتبارسنجی، با اینکه نسبتاً پایین باقی می‌ماند، روندی با شیب ملایم صعودی و با نوسانات قابل توجه نشان می‌دهد. این خطا از حدود دوره ۱۰-۱۵ به بعد، عموماً بالاتر از خطای آموزشی قرار می‌گیرد.
- روند کلی خطا: خطای آموزشی به طور پیوسته کاهش یافته و به مقدار بسیار کمی می‌رسد. خطای اعتبارسنجی ابتدا به شدت کاهش یافته، اما سپس به نظر می‌رسد که به یک حداقل رسیده و پس از آن دچار نوسان شده و کمی افزایش می‌یابد.
- یادگیری اولیه: مدل CNN در دوره‌های ابتدایی (تقریباً ۵ تا ۱۰ دوره اول) بسیار سریع و مؤثر یاد می‌گیرد. این موضوع با افزایش سریع دقت و کاهش شدید خطا برای هر دو مجموعه داده آموزشی و اعتبارسنجی مشهود است.
- بیش‌برازش (Overfitting): از حدود دوره ۱۰ تا ۱۵ به بعد، نشانه‌هایی از بیش‌برازش ظاهر می‌شود. خطای آموزشی همچنان به کاهش خود ادامه می‌دهد، در حالی که خطای اعتبارسنجی از کمترین نقطه خود فاصله گرفته، شروع به افزایش جزئی کرده و نوسانات بیشتری نشان می‌دهد.

همچنین، دقت آموزشی به طور مداوم بالاتر از دقت اعتبارسنجی باقی می‌ماند. این نشان می‌دهد که مدل شروع به یادگیری بیش از حد جزئیات داده‌های آموزشی کرده که این یادگیری به داده‌های جدید به خوبی تعمیم داده نمی‌شود.

- **عملکرد مدل:** مدل به دقت بسیار بالایی روی داده‌های آموزشی (نزدیک به ۱۰۰٪) و همچنین دقت بسیار خوبی روی داده‌های اعتبارسنجی (عموماً حدود ۹۸-۹۹٪) دست یافته است. با این حال، پدیده بیش‌برازش نشان می‌دهد که عملکرد واقعی مدل روی داده‌های کاملاً جدید ممکن است کمی ناپایدارتر یا پایین‌تر از چیزی باشد که دقت آموزشی نهایی نشان می‌دهد و به عملکرد اعتبارسنجی در حدود دوره‌های ۱۰-۱۵ نزدیک‌تر خواهد بود.

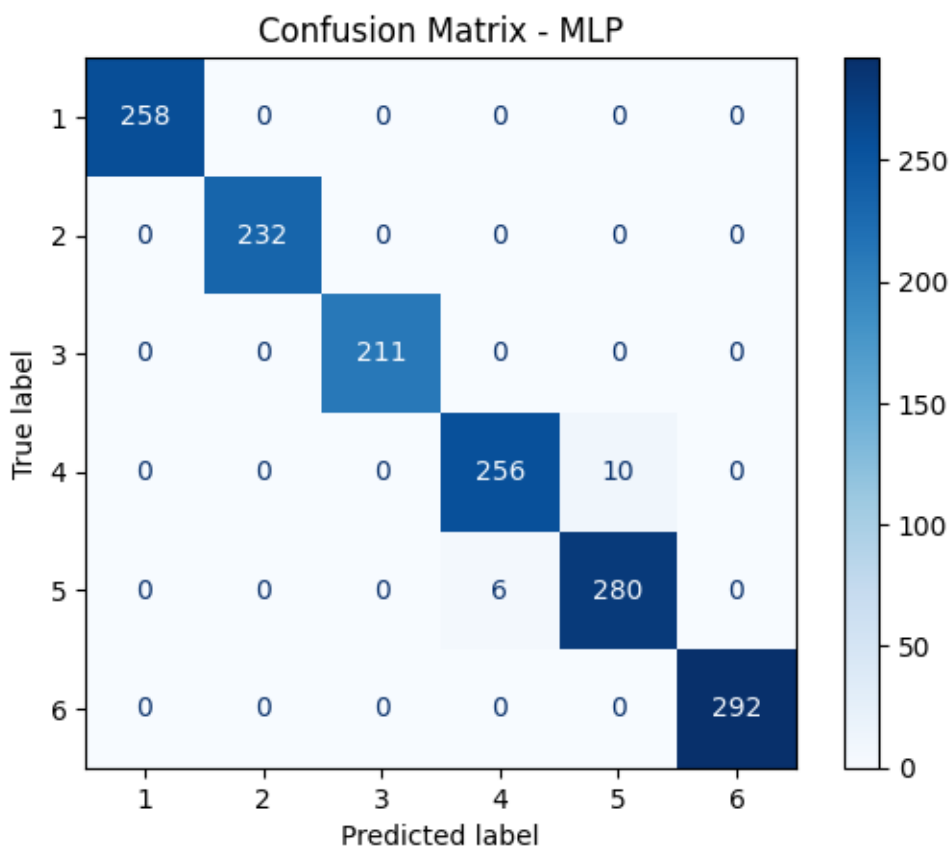
دقت نهایی مدل‌ها روی داده‌های آزمون (evaluate):

MLP - Test Accuracy: 0.9896 | Test Loss: 0.0315

CNN - Test Accuracy: 0.9871 | Test Loss: 0.0489

ماتریس آشفتگی:

ماتریس آشفتگی برای داده‌های آزمون مدل MLP



محور عمودی (True label) برچسب یا کلاس واقعی هر نمونه را نشان می‌دهد و محور افقی (Predicted label) برچسب یا کلاسی که مدل پیش‌بینی کرده است را نشان می‌دهد. اعداد روی قطر اصلی (از بالا چپ به پایین راست) تعداد نمونه‌هایی هستند که مدل به درستی طبقه‌بندی کرده است. اعداد خارج از قطر اصلی نشان‌دهنده طبقه‌بندی‌های اشتباه هستند. در این مدل ۶ کلاس (از ۱ تا ۶) وجود دارد.

۱-WALKING

۲-WALKING_UPSTAIRS

۳-WALKING_DOWNSTAIRS

۴-SITTING

۵-STANDING

۶-LAYING

• کلاس ۱:

- ۲۵۸ نمونه به درستی به عنوان کلاس ۱ طبقه‌بندی شده‌اند.
- هیچ نمونه‌ای از کلاس ۱ به اشتباه به کلاس‌های دیگر نسبت داده نشده است.
- هیچ نمونه‌ای از کلاس‌های دیگر به اشتباه به عنوان کلاس ۱ طبقه‌بندی نشده است.
- عملکرد برای کلاس ۱ عالی است.

• کلاس ۲:

- ۲۳۲ نمونه به درستی به عنوان کلاس ۲ طبقه‌بندی شده‌اند.
- عملکرد برای کلاس ۲ نیز عالی است بدون هیچگونه خطایی.

• کلاس ۳:

- ۲۱۱ نمونه به درستی به عنوان کلاس ۳ طبقه‌بندی شده‌اند.
- عملکرد برای کلاس ۳ نیز عالی است بدون هیچگونه خطایی.

• کلاس ۴:

- ۲۵۶ نمونه به درستی به عنوان کلاس ۴ طبقه‌بندی شده‌اند.

○ اما، ۱۰ نمونه که در واقع متعلق به کلاس ۴ بوده‌اند، به اشتباه توسط مدل به عنوان کلاس ۵ پیش‌بینی شده‌اند.

○ این نشان می‌دهد که مدل گاهی کلاس ۴ را با کلاس ۵ اشتباه می‌گیرد.

• کلاس ۵:

○ ۲۸۰ نمونه به درستی به عنوان کلاس ۵ طبقه‌بندی شده‌اند.

○ اما، ۶ نمونه که در واقع متعلق به کلاس ۵ بوده‌اند، به اشتباه توسط مدل به عنوان کلاس ۴ پیش‌بینی شده‌اند.

○ این نیز نشان می‌دهد که مدل گاهی کلاس ۵ را با کلاس ۴ اشتباه می‌گیرد.

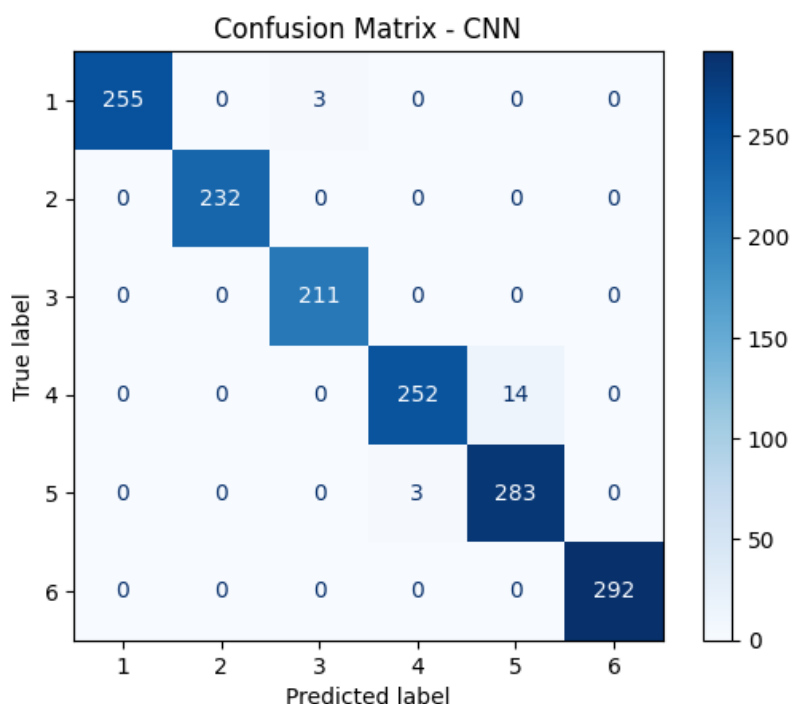
• کلاس ۶:

○ ۲۹۲ نمونه به درستی به عنوان کلاس ۶ طبقه‌بندی شده‌اند.

○ عملکرد برای کلاس ۶ نیز عالی است بدون هیچگونه خطایی.

مدل به طور کلی عملکرد بسیار خوبی دارد، زیرا بیشتر مقادیر روی قطر اصلی قرار دارند و تعداد نمونه‌های اشتباه طبقه‌بندی شده کم است. کلاس‌های ۱، ۲، ۳ و ۶ به طور کامل و بدون خطا توسط مدل تشخیص داده شده‌اند. تنها نقطه ضعف مدل، وجود مقداری درهم‌ریختگی بین کلاس ۴ و کلاس ۵ است.

ماتریس آشفتگی برای داده‌های آزمون مدل CNN



• کلاس ۱:

- ۲۵۵ نمونه به درستی به عنوان کلاس ۱ طبقه‌بندی شده‌اند.
- اما، ۳ نمونه که در واقع متعلق به کلاس ۱ بوده‌اند، به اشتباه توسط مدل به عنوان کلاس ۳ پیش‌بینی شده‌اند.
- این یک خطای جدید است که در مدل MLP مشاهده نشده بود.

• کلاس ۲:

- ۲۳۲ نمونه به درستی به عنوان کلاس ۲ طبقه‌بندی شده‌اند.
- هیچ نمونه‌ای از کلاس ۲ به اشتباه به کلاس‌های دیگر نسبت داده نشده است و هیچ نمونه‌ای از کلاس‌های دیگر نیز به اشتباه کلاس ۲ پیش‌بینی نشده است.
- عملکرد برای کلاس ۲ عالی است.

• کلاس ۳:

- ۲۱۱ نمونه به درستی به عنوان کلاس ۳ طبقه‌بندی شده‌اند.
- هیچ نمونه‌ای از کلاس ۳ به اشتباه به کلاس‌های دیگر نسبت داده نشده است. با این حال، همانطور که در بالا ذکر شد، ۳ نمونه از کلاس ۱ به اشتباه کلاس ۳ پیش‌بینی شده‌اند.
- عملکرد برای نمونه‌هایی که واقعاً کلاس ۳ هستند خوب است، اما مدل گاهی کلاس ۱ را با آن اشتباه می‌گیرد.

• کلاس ۴:

- ۲۵۲ نمونه به درستی به عنوان کلاس ۴ طبقه‌بندی شده‌اند.
- اما، ۱۴ نمونه که در واقع متعلق به کلاس ۴ بوده‌اند، به اشتباه توسط مدل به عنوان کلاس ۵ پیش‌بینی شده‌اند.
- این بیشترین تعداد خطای مدل است و نشان می‌دهد که مدل در تفکیک کلاس ۴ از ۵ دچار چالش است (مشابه مدل MLP اما با تعداد متفاوت).

• کلاس ۵:

- ۲۸۳ نمونه به درستی به عنوان کلاس ۵ طبقه‌بندی شده‌اند.

○ اما، ۳ نمونه که در واقع متعلق به کلاس ۵ بوده‌اند، به اشتباه توسط مدل به عنوان کلاس ۴ پیش‌بینی شده‌اند.

○ درهم‌ریختگی بین کلاس ۴ و ۵ دوطرفه است، اما مدل بیشتر کلاس ۴ را با ۵ اشتباه می‌گیرد تا برعکس.

• کلاس ۶:

○ ۲۹۲ نمونه به درستی به عنوان کلاس ۶ طبقه‌بندی شده‌اند.

○ هیچ نمونه‌ای از کلاس ۶ به اشتباه به کلاس‌های دیگر نسبت داده نشده و هیچ نمونه‌ای از کلاس‌های دیگر نیز به اشتباه کلاس ۶ پیش‌بینی نشده است.

○ عملکرد برای کلاس ۶ عالی است.

مدل CNN نیز به طور کلی عملکرد بسیار خوبی دارد و بیشتر نمونه‌ها به درستی طبقه‌بندی شده‌اند. کلاس‌های ۲ و ۶ توسط این مدل به طور کامل و بدون خطا (هم از نظر خطاهای نوع اول و هم نوع دوم برای این کلاس‌ها) تشخیص داده شده‌اند.

بیشترین درهم‌ریختگی بین کلاس ۴ و کلاس ۵ مشاهده می‌شود (۱۴ نمونه از کلاس ۴ به عنوان ۵ و ۳ نمونه از کلاس ۵ به عنوان ۴). این مشکل در مدل MLP هم وجود داشت. یک خطای جدید در این مدل، طبقه‌بندی اشتباه ۳ نمونه از کلاس ۱ به عنوان کلاس ۳ است.

۱-۶-تحلیل

با مقایسه نتایج نمودارهای یادگیری (دقت و خطا) و ماتریس‌های آشفتگی، به نظر می‌رسد که مدل MLP (پرسپترون چندلایه) در این مسئله عملکرد کمی بهتر از مدل CNN (شبکه عصبی کانولوشنی) داشته است. در ادامه دلایل این نتیجه‌گیری توضیح داده می‌شود:

دلایل برتری نسبی مدل MLP

۱. تعداد کل خطاهای کمتر در ماتریس درهم‌ریختگی:

○ مدل MLP مجموعاً ۱۶ نمونه را به اشتباه طبقه‌بندی کرد.

○ مدل CNN مجموعاً ۲۰ نمونه را به اشتباه طبقه‌بندی کرد. تعداد خطای کمتر در ماتریس

درهم‌ریختگی (که معمولاً بر روی داده‌های تست یا اعتبارسنجی محاسبه می‌شود)

نشان‌دهنده قدرت تعمیم بهتر مدل MLP بر روی داده‌های دیده‌نشده در این مورد خاص است.

۲. عملکرد بهتر در تفکیک کلاس‌های خاص:

- کلاس‌های بدون خطا: مدل MLP توانست کلاس‌های ۱، ۲، ۳ و ۶ را بدون هیچ‌گونه خطایی تشخیص دهد. در مقابل، مدل CNN فقط کلاس‌های ۲ و ۶ را بدون خطا تشخیص داد و برای کلاس ۱، سه نمونه را به اشتباه به عنوان کلاس ۳ طبقه‌بندی کرد. این نشان می‌دهد MLP در تفکیک برخی کلاس‌ها موفق‌تر بوده است.
- درهم‌ریختگی بین کلاس ۴ و ۵: هر دو مدل در تفکیک کلاس ۴ و ۵ از یکدیگر دچار چالش بودند.
- MLP: ده مورد کلاس ۴ به ۵ و شش مورد کلاس ۵ به ۴ (مجموعاً ۱۶ خطا بین این دو کلاس).
- CNN: چهارده مورد کلاس ۴ به ۵ و سه مورد کلاس ۵ به ۴ (مجموعاً ۱۷ خطا بین این دو کلاس). در این بخش نیز MLP تعداد خطای کلی کمتری بین این دو کلاس خاص داشته یا عملکردی بسیار نزدیک به CNN ارائه داده است.

۳. نشانه‌های بیش‌برازش (Overfitting) و تاثیر آن:

- نمودارهای یادگیری: هر دو مدل در نمودارهای یادگیری خود نشانه‌هایی از بیش‌برازش را بروز دادند (جایی که خطای اعتبارسنجی شروع به افزایش یا نوسان بیشتر می‌کند در حالی که خطای آموزشی همچنان کاهش می‌یابد).
- در مدل MLP، خطای اعتبارسنجی از حدود دوره ۷-۱۰ شروع به افزایش کرد.
- در مدل CNN، خطای اعتبارسنجی از حدود دوره ۱۰-۱۵ به بعد دچار افزایش و نوسان شد.
- تاثیر نهایی: با اینکه هر دو مدل بیش‌برازش را تجربه کردند، اما با توجه به نتایج ماتریس درهم‌ریختگی (که مربوط به مدل نهایی پس از ۵۰ دوره یا یک نقطه نزدیک به آن است)، به نظر می‌رسد اثرات منفی بیش‌برازش بر روی تعداد خطاهای نهایی در مدل MLP کمتر بوده است.
- با وجود اینکه هر دو مدل به دقت‌های بالایی دست یافتند و شباهت‌هایی در الگوهای یادگیری و نقاط ضعف (مانند تفکیک کلاس ۴ و ۵) داشتند، مدل MLP با داشتن تعداد خطای کمتر در طبقه‌بندی نهایی و تشخیص بدون خطای تعداد بیشتری از کلاس‌ها، عملکرد کلی بهتری را در این مسئله خاص از خود نشان داده است.

در مجموع به نظر می‌رسد مدل MLP در این مورد خاص، با تعداد خطای کمتر، اندکی بهتر از مدل CNN عمل کرده است، هرچند هر دو مدل عملکرد بالایی دارند. البته برای نتیجه‌گیری قطعی‌تر، معیارهای دقیق‌تری مانند دقت (Precision)، بازیابی (Recall) و امتیاز F1 برای هر کلاس و به طور کلی باید محاسبه و مقایسه شوند. عملکرد مدل‌ها می‌تواند به شدت تحت تأثیر عواملی مانند معماری دقیق شبکه، هایپرپارامترها، روش بهینه‌سازی و مجموعه داده خاص باشد. ممکن است با تنظیمات متفاوت، مدل CNN بتواند عملکرد بهتری از خود نشان دهد، اما بر اساس شواهد موجود، هر دو مدل عملکرد بالایی دارند و مدل MLP با اختلاف جزئی در این کیس خاص برتری دارد.

با وجود اینکه به طور معمول انتظار می‌رود مدل‌های CNN عملکرد بهتری در تشخیص الگوها داشته باشند، در این مسئله‌ی خاص که ورودی‌ها به صورت برداری و با ویژگی‌های مهندسی‌شده هستند UCI HAR Dataset مدل MLP توانسته است به دقت بالاتری دست یابد.

این موضوع نشان می‌دهد که ساختار ساده و کاملاً متصل MLP برای داده‌های برداری (بدون نیاز به استخراج ویژگی‌های مکانی یا زمانی پیچیده) به خوبی کافی بوده و حتی نسبت به CNN بهتر عمل کرده است.

بخش دوم: دادگان تشخیص عیوب سطحی قطعات فولادی (NEU Surface Defects)

این دادگان شامل ۱۸۰۰ تصویر خاکستری 200×200 پیکسلی از قطعات فولادی نورد شده است که هر یک دارای یکی از عیوب ترکهای سطحی، ناخالصی، لکه، قلوه کن شدگی، پوسته گرفتگی و خراشیدگی است. دادگان مورد نظر کاملاً متوازن است، به این معنی که به ازای هر عیب، ۳۰۰ نمونه تصویری وجود دارد.

به دلیل وجود نویزهای گوناگون در تصاویر و پیچیدگی بافت سطحی قطعات انتظار میرود که عملکرد شبکه های پیچشی در دسته بندی عیوب قطعات بهتر از عملکرد مدل های ساده ای همچون MLP باشد.

۲-۱- آماده سازی داده ها

ابتدا فایل دادگان را از حالت فشرده خارج میکنیم و مسیرهای آموزش و تست را تعریف میکنیم. تابع بارگذاری تصاویر و لیبلها را تعریف میکنیم، و با اجرای آن تصاویر را بارگذاری میکنیم، شکل داده ها به صورت زیر است:

✓ شکل داده ها:

X_train: (1440, 224, 224, 3) | y_train: (1440,)

X_val: (360, 224, 224, 3) | y_val: (360,)

کانال رنگی	شکل تصویر	تعداد نمونه	مجموعه
3 (RGB)	224×224	1440	X_train
3 (RGB)	224×224	360	X_val

تصاویر به درستی بارگذاری و **resize** شدن و تعداد نمونه ها با توجه به ۶ کلاس و حدود ۲۴۰ تصویر برای هر کلاس منطقی است.

سپس داده ها را نرمال سازی میکنیم، برای نرمال سازی تصاویر، از روش **Min-Max Scaling** استفاده شد که پیکسل ها را به بازه $[0, 1]$ تبدیل می کند. این روش به دلیل سادگی، سرعت بالای اجرا، و عملکرد مناسب با شبکه های عصبی کانولوشنی (CNN) انتخاب شد. این نوع نرمال سازی در اکثر مسائل بینایی ماشین استاندارد بوده و به همگرایی سریع تر در مرحله آموزش کمک می کند.

سپس داده های نرمال شده را ادغام میکنیم (تصاویر و لیبل ها) و لیبل ها را تبدیل به عددی میکنیم (label encoder) و به نسبت ۸۵٪ آموزش و ۱۵٪ آزمون تقسیم میکنیم. در این تقسیم بندی، از تابع train_test_split با گزینه stratify استفاده شد تا نسبت کلاس ها در هر دو بخش حفظ شود. ابعاد نهایی داده ها به صورت زیر است:

✓ داده ها با موفقیت تقسیم شدند:

X_train_final: (1530, 224, 224, 3) | y_train_cat: (1530, 6)

X_test_final : (270, 224, 224, 3) | y_test_cat : (270, 6)

کلاس ها: ['crazing' 'inclusion' 'patches' 'pitted_surface' 'rolled-in_scale'
'scratches']

همچنین در این بخش لیبل ها را one-hot encoding میکنیم، دلایل آن در بخش قبلی تمرین توضیح داده شد.

۲-۲- طراحی شبکه MLP

به طور معمول، داده های تصویری با مدل های CNN بهتر پردازش می شوند. اما می توانیم تصاویر را صاف (flatten) کنیم و به عنوان ورودی به MLP بدیم.

مدل MLP طراحی شده شامل دو لایه پنهان با اندازه ۲۵۶ و ۱۲۸ نرون، همراه با تابع فعال سازی ReLU می باشد. برای کاهش بیش برآزش از Dropout با نرخ ۰.۲ استفاده شده است. لایه خروجی با ۶ نرون و تابع Softmax برای طبقه بندی تصاویر به شش کلاس نقص سطح طراحی شده است.

Input: Flattened image (۱۵۰۵۲۸ = ۳×۲۲۴×۲۲۴)

→Dense(256), ReLU

→Dropout(۰.۲)

→Dense(128), ReLU

→Dense(6), Softmax

بخش	دلیل
Flatten	برای تبدیل تصویر به بردار یک بعدی قابل ورود به Dense
Dense(256) + Dense(128)	برای یادگیری الگوهای ترکیبی بین پیکسل‌ها
ReLU	سریع، ساده، مؤثر در لایه‌های پنهان
Dropout	برای کاهش overfitting
Dense(6) + Softmax	خروجی ۶ کلاس با احتمال تعلق به هر کلاس

استفاده از لایه Flatten

کارکرد: این لایه تصویر سه بعدی $(3 \times 224 \times 224)$ را به یک بردار یک بعدی با طول ۱۵۰,۵۲۸ تبدیل می‌کند. چون لایه‌های Dense فقط ورودی یک بعدی می‌پذیرند، برای استفاده از MLP باید تصویر تخت (flatten) بشود.

محدودیت: در این روش، روابط مکانی (spatial structure) بین پیکسل‌ها از بین می‌رود که باعث می‌شود CNN مناسب‌تر باشد.

استفاده از لایه‌های Dense

کارکرد: لایه‌های Dense تمام نرون‌های ورودی را به همه نرون‌های بعدی متصل می‌کنند و قابلیت یادگیری ویژگی‌های پیچیده را دارند.

- لایه اول (۲۵۶ نرون) برای استخراج ویژگی‌های سطح بالا
 - لایه دوم (۱۲۸ نرون) برای کاهش ابعاد ویژگی‌ها و یادگیری ترکیب آن‌ها
- تعداد نرون‌ها باید به اندازه‌ای باشد که مدل بتواند اطلاعات کافی یاد بگیرد ولی پیچیدگی بیش از حد نداشته باشد، در عمل، ۲۵۶ و ۱۲۸ ترکیب خوبی بین قدرت یادگیری و سرعت آموزش ایجاد می‌کنند.

استفاده از Dropout

کارکرد: به صورت تصادفی ۲۰٪ از نرون‌ها در هر epoch غیرفعال می‌شوند برای جلوگیری از overfitting یعنی اینکه مدل فقط داده‌های آموزش رو حفظ نکند بلکه قابلیت تعمیم به داده‌های جدید رو داشته باشد.

در لایه‌های MLP خیلی ضروری است چون MLP بدون Dropout به راحتی overfit می‌کند مخصوصاً با داده‌های تصویری.

تابع فعال سازی ReLU در لایه های پنهان

بسیار ساده و محاسباتی سریع، به مشکل vanishing gradient دچار نمی شود، باعث sparse شدن خروجی ها می شود که یادگیری را بهتر می کند

نسبت به sigmoid یا tanh بسیار بهتر در شبکه های عمیق عمل می کند.

تابع فعال سازی Softmax در لایه خروجی

خروجی را به احتمال تعلق به هر کلاس تبدیل می کند، جمع خروجی ها برابر با ۱ می شود و می توان از categorical_crossentropy به عنوان loss استفاده کرد.

Layer (type)	Output Shape	Param #
flatten_2 (Flatten)	(None, 150528)	0
dense_5 (Dense)	(None, 256)	38,535,424
dropout_5 (Dropout)	(None, 256)	0
dense_6 (Dense)	(None, 128)	32,896
dense_7 (Dense)	(None, 6)	774

Total params: 38,569,094 (147.13 MB)
Trainable params: 38,569,094 (147.13 MB)
Non-trainable params: 0 (0.00 B)

۲-۳- طراحی شبکه CNN

مدل CNN طراحی شده شامل چهار بلوک کانولوشن با فیلترهای ۳۲، ۶۴، ۱۲۸ و ۲۵۶ به همراه توابع فعال سازی ReLU، لایه های MaxPooling و Batch Normalization است. برای جلوگیری از بیش برآزش، از لایه های Dropout با مقادیر مناسب در هر مرحله استفاده شده است. خروجی مدل از طریق لایه Dense با تابع Softmax به شش کلاس نقص سطح دسته بندی می شود. این معماری به دلیل تعادل بین دقت بالا، جلوگیری از overfitting و پایداری در آموزش انتخاب شد.

Input(۳, ۲۲۴, ۲۲۴):

→Conv2D(32, 3×3), ReLU

→BatchNormalization

→MaxPooling(۲×۲)

→Dropout(۰.۳)

→Conv2D(64, 3×3), ReLU

→BatchNormalization

→MaxPooling(۲×۲)

→Dropout(۰.۳)

→Conv2D(128, 3×3), ReLU

→BatchNormalization

→MaxPooling(۲×۲)

→Dropout(۰.۳۵)

→Conv2D(256, 3×3), ReLU

→BatchNormalization

→MaxPooling(۲×۲)

→Dropout(۰.۳۵)

→Flatten

→Dense(128), ReLU

→Dropout(۰.۴)

→Dense(6), Softmax

لایه‌های کانولوشن (Conv2D)

- از چهار لایه کانولوشن با فیلترهای ۳۲،۶۴،۱۲۸،۲۵۶ استفاده شده.
- فیلترها با اندازه‌ی (۳×۳) معمول‌ترین گزینه هستند برای استخراج ویژگی‌های محلی از تصویر.
- افزایش تدریجی تعداد فیلترها کمک می‌کند مدل از ویژگی‌های ساده (مثل لبه‌ها) به ویژگی‌های پیچیده‌تر (مثل شکل و بافت نقص‌ها) برسد.

توابع فعال‌سازی ReLU

- تابع ReLU باعث کاهش پیچیدگی محاسباتی و جلوگیری از مشکل محو شدن گرادیان می‌شود.

- عملکرد خوبی در تمام شبکه‌های عمیق داشته و بسیار رایج و پایدار است.

لایه‌های MaxPooling

- استفاده از MaxPooling(2x2) بعد از هر لایه کانولوشن باعث کاهش ابعاد تصویر می‌شود.
- این لایه باعث کاهش پارامترها و جلوگیری از overfitting می‌شود.

Batch Normalization

- بین هر Conv و Pool یک BatchNormalization استفاده شده.
- کمک می‌کند مدل پایدارتر آموزش ببیند و سریع‌تر همگرا شود.
- نوسانات گرادیان‌ها را کاهش می‌دهد.

Dropout

- برای جلوگیری از overfitting در تمام مراحل استفاده شده.

Dense + Softmax و Flatten

- Flatten خروجی نهایی کانولوشن را یک‌بعدی می‌کند.
- Dense(128) به‌عنوان fully connected layer نهایی برای یادگیری ترکیب ویژگی‌ها.
- Dense(6, softmax) برای طبقه‌بندی در ۶ کلاس نهایی استفاده شده.

۲-۴-آموزش

هر دو مدل طراحی شده با استفاده از الگوریتم Adam و نرخ یادگیری ۰.۰۰۱ آموزش داده شدند. در طی آموزش، مقادیر loss و accuracy برای مجموعه‌های آموزش و آزمون در هر epoch ثبت شده‌اند تا برای تحلیل عملکرد مدل مورد استفاده قرار گیرند.

تعداد ایپاک‌ها برای هر دو مدل ۵۰، و مقدار بچ سائز ۳۲ برای هر دو مدل تنظیم شد.

برای مدل CNN از EarlyStopping با صبر ۱۰ ایپاک نیز استفاده شد:

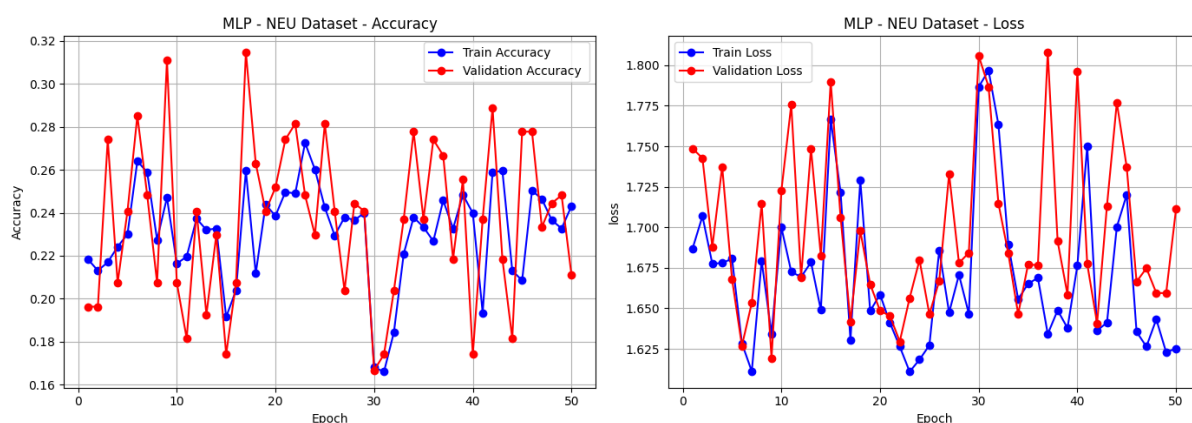
توقف زودهنگام یک تکنیک در فرآیند آموزش شبکه‌های عصبی است که به‌منظور جلوگیری از بیش‌برازش (Overfitting) به‌کار می‌رود. در این روش، عملکرد مدل به‌صورت مداوم روی داده‌های اعتبارسنجی

(Validation Set) بررسی می‌شود. اگر دقت مدل روی داده‌های اعتبارسنجی در چند دوره‌ی متوالی (مثلاً ۵ اپیاک) بهبود نیابد، فرآیند آموزش متوقف شده و بهترین وزن‌های مدل تا آن لحظه بازیابی می‌شود. این روش کمک می‌کند که:

- مدل بیش از حد روی داده‌های آموزش یاد نگیرد Overfit نشود
- زمان آموزش کاهش یابد
- مدل در نقطه‌ای متعادل بین دقت و تعمیم‌پذیری متوقف شود

۲-۵-ارزیابی

نمودارهای خطا (Loss) و دقت (Accuracy) را برای داده‌های آموزش و آزمون برای مدل MLP رسم شد:



این نمودارها عملکرد یک مدل شبکه عصبی پرسپترون چندلایه (MLP) را در طول ۵۰ دوره (Epoch) آموزش بر روی مجموعه داده تصویری عیوب سطحی قطعات فولادی (NEU Dataset) نشان می‌دهند. نمودار سمت چپ دقت (Accuracy) و نمودار سمت راست خطا (Loss) را نمایش می‌دهد. در هر دو نمودار، خط آبی مربوط به داده‌های آموزشی (Train) و خط قرمز مربوط به داده‌های اعتبارسنجی (Validation) است.

۱. عملکرد کلی پایین (Low Performance)

- **دقت (Accuracy):** دقت مدل هم برای داده‌های آموزشی و هم برای داده‌های اعتبارسنجی بسیار پایین است و تقریباً بین ۱۶٪ تا ۳۲٪ نوسان می‌کند. حداکثر دقت به دست آمده برای داده‌های آموزشی حدود ۳۲٪ و برای داده‌های اعتبارسنجی حدود ۳۰٪ است. این نشان می‌دهد که مدل به خوبی قادر به یادگیری و تشخیص عیوب سطحی فولاد نیست.

- **خطا (Loss):** مقادیر خطا نیز به همین نسبت بالا هستند و عموماً بین ۱.۶ تا ۱.۸ باقی می‌مانند. در یک فرآیند یادگیری مطلوب، انتظار داریم که میزان خطا به مقادیر بسیار کمتری کاهش یابد.

۲. نوسانات شدید و عدم پایداری (High Fluctuations and Instability)

- هر دو منحنی دقت و خطا، چه برای داده‌های آموزشی و چه برای داده‌های اعتبارسنجی، نوسانات زیادی از یک دوره به دوره دیگر نشان می‌دهند. این موضوع بیانگر ناپایداری فرآیند آموزش است.
- دقت آموزشی (آبی) و دقت اعتبارسنجی (قرمز) اغلب الگوهای مشابهی را دنبال می‌کنند اما به طور مداوم به هم نزدیک نیستند و هر دو نامنظم هستند.
- به طور مشابه، خطای آموزشی (آبی) و خطای اعتبارسنجی (قرمز) نیز به شدت نوسان می‌کنند. کاهش یکنواخت و مداوم خطا که در آموزش مطلوب دیده می‌شود، در اینجا مشاهده نمی‌گردد.

۳. عدم وجود نشانه واضح بیش‌برازش (Overfitting) شدید (اما یادگیری مناسبی هم رخ نداده است)

- اگرچه در برخی دوره‌ها خطای اعتبارسنجی بالاتر از خطای آموزشی است (مثلاً حدود دوره ۳۰ و ۳۷) یا دقت اعتبارسنجی پایین‌تر از دقت آموزشی است، اما عملکرد مدل روی خود داده‌های آموزشی نیز ضعیف و ناپایدار است.
- معمولاً، بیش‌برازش واضح زمانی مشخص می‌شود که خطای آموزشی به طور مداوم کاهش و دقت آموزشی به طور مداوم افزایش یابد، در حالی که خطای اعتبارسنجی شروع به افزایش و دقت اعتبارسنجی شروع به کاهش یا ثابت ماندن کند. در اینجا، هر دو معیار آموزشی و اعتبارسنجی ضعیف و نامنظم هستند، که نشان می‌دهد مدل در وهله اول در یادگیری الگوهای معنی‌دار از داده‌ها با مشکل مواجه است.

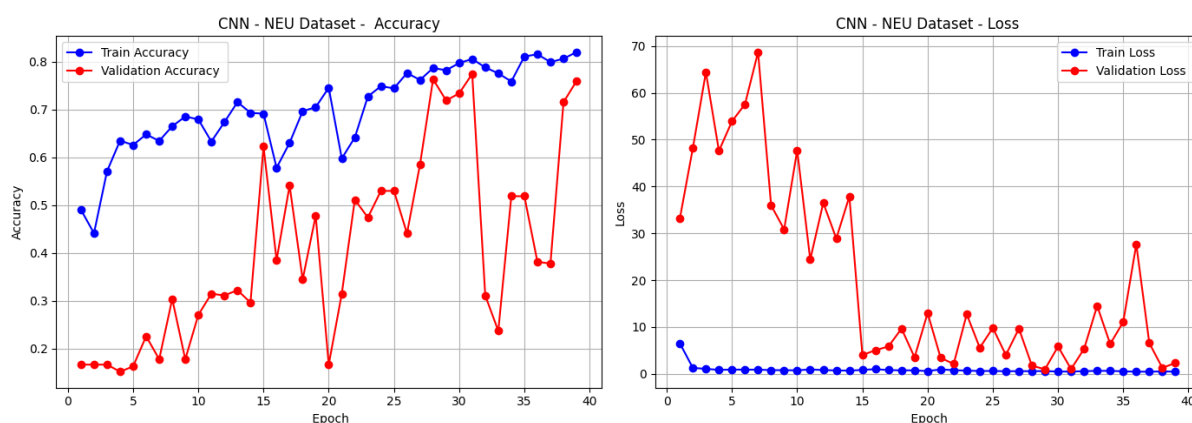
۴. دلایل احتمالی برای رفتار مشاهده شده

- **نرخ یادگیری (Learning Rate):** ممکن است نرخ یادگیری بیش از حد بالا باشد، که باعث می‌شود بهینه‌ساز از نقاط بهینه عبور کرده و منجر به رفتار پرشی مشاهده شده شود.
- **معماری مدل (Model Architecture):** معماری MLP ممکن است برای ثبت پیچیدگی‌های داده‌های تصویری بسیار ساده باشد (کم‌برازش یا Underfitting)، یا شاید برای این نوع وظایف مبتنی بر تصویر بدون استخراج ویژگی پیچیده‌تر، مناسب نباشد.

- **مشکلات داده (Data Issues):** مجموعه داده ممکن است بسیار چالش برانگیز، دارای نویز، یا با مشکل عدم تعادل کلاس‌ها مواجه باشد که به درستی به آن پرداخته نشده است. ویژگی‌های استخراج شده از تصاویر ممکن است به اندازه کافی متمایزکننده نباشند.

نمودارها نشان می‌دهند که مدل MLP عملکرد ضعیفی بر روی مجموعه داده عیوب سطحی فولاد NEU دارد. فرآیند آموزش بسیار ناپایدار است و نوسانات قابل توجهی در دقت و خطا برای هر دو مجموعه داده آموزشی و اعتبارسنجی مشاهده می‌شود. مدل به طور موثر یاد نمی‌گیرد که بین انواع مختلف عیوب تمایز قائل شود، که با دقت پایین و خطای بالا مشهود است. نشانه‌های واضحی از بیش‌برازش وجود ندارد، زیرا مدل حتی داده‌های آموزشی را نیز به خوبی یاد نمی‌گیرد. به نظر می‌رسد مشکل اصلی، دشواری اساسی در یادگیری از داده‌ها با پیکربندی فعلی مدل و تنظیمات آموزش است.

نمودارهای خطا (Loss) و دقت (Accuracy) را برای داده‌های آموزش و آزمون برای مدل CNN رسم شد:



۱. یادگیری بسیار خوب روی داده‌های آموزشی

- **دقت آموزشی (خط آبی در نمودار دقت):** این خط روند صعودی بسیار خوبی را نشان می‌دهد. دقت از مقادیر پایین شروع شده و به سرعت افزایش می‌یابد و در اواخر دوره‌ها به مقادیر بسیار بالایی، نزدیک به ۱۰۰٪ (یا ۱.۰) می‌رسد. این نشان می‌دهد که مدل CNN توانسته الگوهای موجود در داده‌های آموزشی را به خوبی یاد بگیرد.
- **خطای آموزشی (خط آبی در نمودار خطا):** این خط نیز رفتار مطلوبی دارد. خطا از مقادیر بالا شروع شده و به سرعت و به طور پیوسته کاهش می‌یابد و در دوره‌های پایانی به مقادیر بسیار نزدیک به صفر می‌رسد. این موضوع تایید می‌کند که مدل به خوبی بر روی داده‌های آموزشی منطبق (fit) شده است.

۲. نشانه‌های واضح بیش‌برازش (Overfitting)

• فاصله بین منحنی‌های آموزشی و اعتبارسنجی:

- در نمودار دقت: از حدود دوره ۱۰ به بعد، یک فاصله مشخص بین دقت آموزشی (آبی) و دقت اعتبارسنجی (قرمز) شروع به ایجاد شدن و افزایش می‌کند. در حالی که دقت آموزشی به سمت ۱۰۰٪ میل می‌کند، دقت اعتبارسنجی در سطح پایین‌تری (حدود ۹۵٪-۹۰٪ به نظر می‌رسد) ثابت مانده و حتی در اواخر ممکن است کمی افت هم داشته باشد.
- در نمودار خطا: این پدیده حتی واضح‌تر است. خطای آموزشی به طور مداوم کاهش می‌یابد، اما خطای اعتبارسنجی پس از یک کاهش اولیه، در حدود دوره ۱۰ شروع به ثابت ماندن و سپس افزایش می‌کند. این واگرایی بین خطای آموزشی و اعتبارسنجی یک نشانه کلاسیک از بیش‌برازش است.

- مفهوم بیش‌برازش: مدل به جای یادگیری الگوهای کلی که قابل تعمیم به داده‌های جدید باشند، جزئیات و نویزهای خاص داده‌های آموزشی را حفظ کرده است. در نتیجه، عملکرد آن بر روی داده‌های جدید (اعتبارسنجی) به خوبی داده‌های آموزشی نیست و حتی بدتر هم می‌شود.

۳. عملکرد مدل بر روی داده‌های اعتبارسنجی

- بهترین عملکرد: به نظر می‌رسد بهترین عملکرد مدل بر روی داده‌های اعتبارسنجی جایی در حدود دوره‌های ۵ تا ۱۵ رخ داده است، جایی که دقت اعتبارسنجی به حداکثر خود می‌رسد و خطای اعتبارسنجی هنوز شروع به افزایش نکرده یا در کمترین مقدار خود قرار دارد.
- نیاز به توقف زودهنگام (Early Stopping) یا تنظیمات دیگر: برای جلوگیری از بیش‌برازش و به دست آوردن مدلی که بهتر به داده‌های جدید تعمیم پیدا کند، می‌توان از روش‌هایی مانند توقف زودهنگام (متوقف کردن آموزش زمانی که عملکرد روی داده اعتبارسنجی شروع به افت می‌کند)، استفاده از روش‌های تنظیم (Regularization) مانند Dropout، یا افزایش حجم داده‌های آموزشی (Data Augmentation) استفاده کرد.

۴. پایداری آموزش

- برخلاف نمودارهای مدل MLP قبلی، منحنی‌های آموزش و اعتبارسنجی در اینجا بسیار هموارتر و پایدارتر هستند و نوسانات شدید کمتری دارند. این نشان می‌دهد که فرآیند آموزش با مدل CNN و تنظیمات فعلی آن (احتمالاً نرخ یادگیری مناسب‌تر) از ثبات بیشتری برخوردار بوده است.

مدل CNN به طور قابل توجهی عملکرد بهتری نسبت به مدل MLP روی این مجموعه داده تصویری داشته است، که با توجه به ماهیت داده‌های تصویری و توانایی CNN‌ها در استخراج ویژگی‌های فضایی، انتظار می‌رود.

- مدل CNN توانسته داده‌های آموزشی را بسیار خوب یاد بگیرد.
 - با این حال، مدل به وضوح دچار بیش‌برازش (Overfitting) شده است. یعنی عملکرد عالی روی داده‌های آموزشی به همان خوبی به داده‌های جدید اعتبارسنجی منتقل نشده و پس از نقطه‌ای، عملکرد روی داده‌های اعتبارسنجی افت می‌کند.
 - برای استفاده عملی از این مدل، باید اقداماتی برای مقابله با بیش‌برازش انجام شود. انتخاب مدل در دوره‌ای که بهترین عملکرد را روی داده اعتبارسنجی داشته، یک راهکار اولیه است.
- این تحلیل نشان می‌دهد که معماری CNN برای این نوع وظیفه مناسب‌تر است، اما نیاز به تنظیمات دقیق‌تر و روش‌هایی برای جلوگیری از بیش‌برازش دارد تا بتواند به خوبی به داده‌های دیده نشده تعمیم یابد.

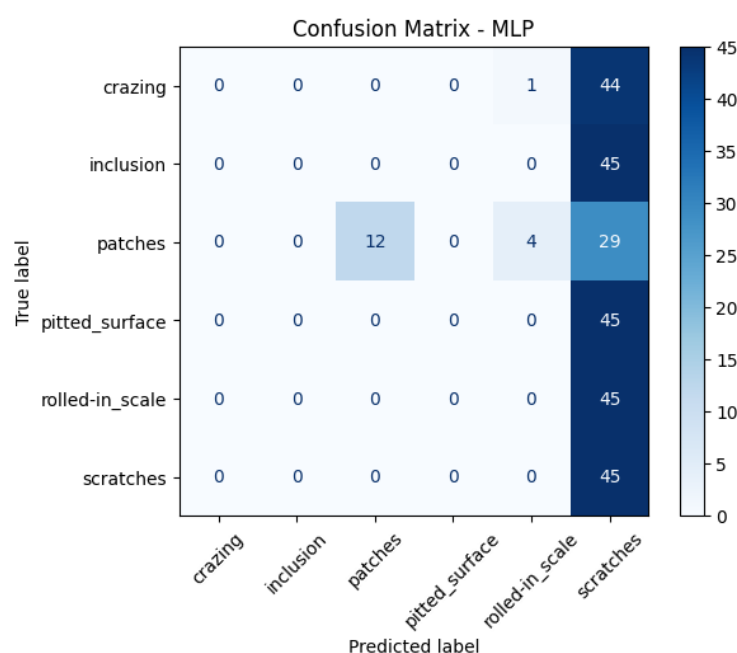
دقت نهایی مدل‌ها روی داده‌های آزمون (evaluate):

 MLP FINAL ACCURACY FOR TEST DATA: 0.2111

 CNN FINAL ACCURACY FOR TEST DATA: 0.7185

ماتریس آشفستگی:

ماتریس آشفستگی برای داده‌های آزمون مدل MLP



عملکرد کلی بسیار ضعیف: به جز کلاس scratches و تا حدی patches، مدل در تشخیص سایر کلاس‌ها عملکرد بسیار ضعیفی دارد.

• کلاس scratches

- این کلاس توسط مدل به خوبی تشخیص داده شده است. تمام ۴۵ نمونه واقعی scratches به درستی به عنوان scratches پیش‌بینی شده‌اند (۱۰۰٪ دقت برای این کلاس).
- متأسفانه، مدل تمایل شدیدی به پیش‌بینی اکثر نمونه‌های کلاس‌های دیگر نیز به عنوان scratches دارد.

• کلاس patches

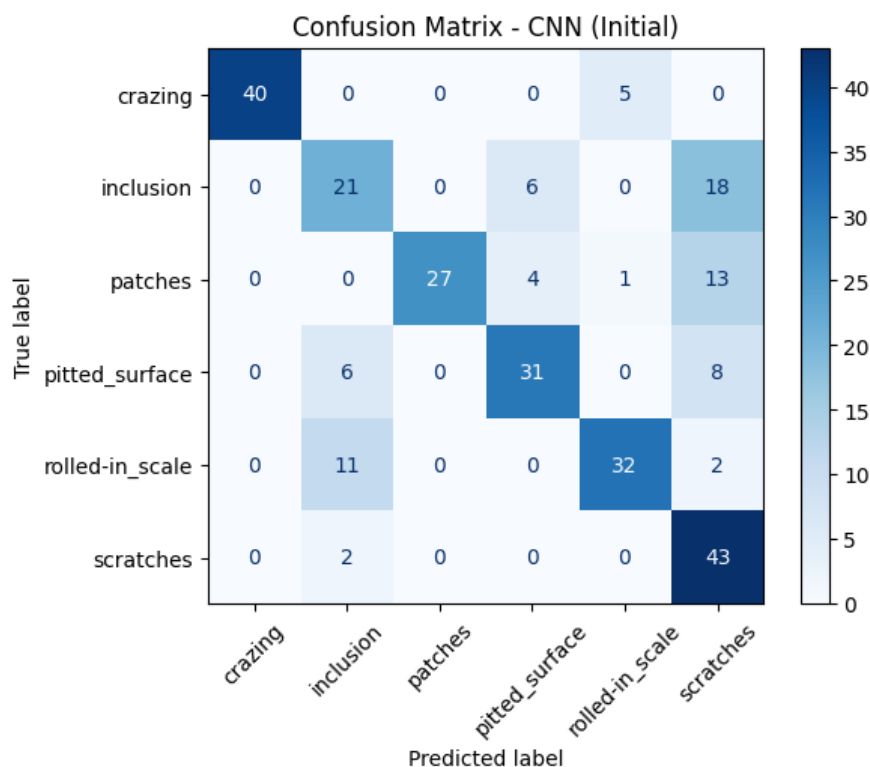
- از مجموع ۴۵ نمونه واقعی patches، تنها ۱۲ نمونه به درستی طبقه‌بندی شده‌اند (دقت حدود ۲۶.۷٪)
- ۴ نمونه patches به اشتباه rolled_in_scale و ۲۹ نمونه به اشتباه scratches پیش‌بینی شده‌اند.

• سایر کلاس‌ها (crazing, inclusion, pitted_surface, rolled_in_scale)

- **Crazing** از ۴۵ نمونه، ۱ مورد به اشتباه rolled_in_scale و ۴۴ مورد به اشتباه scratches پیش‌بینی شده‌اند، هیچ‌کدام به درستی تشخیص داده نشده‌اند.
- **Inclusion** تمام ۴۵ نمونه به اشتباه scratches پیش‌بینی شده‌اند، هیچ‌کدام به درستی تشخیص داده نشده‌اند.
- **Pitted_surface** تمام ۴۵ نمونه به اشتباه scratches پیش‌بینی شده‌اند، هیچ‌کدام به درستی تشخیص داده نشده‌اند.
- **Rolled_in_scale** تمام ۴۵ نمونه به اشتباه scratches پیش‌بینی شده‌اند، هیچ‌کدام به درستی تشخیص داده نشده‌اند.

مدل MLP به شدت دچار مشکل است. به نظر می‌رسد که مدل عمدتاً یاد گرفته است که بیشتر ورودی‌ها را به عنوان کلاس scratches طبقه‌بندی کند. این ممکن است به دلیل عدم تعادل در داده‌ها، معماری نامناسب MLP برای این نوع داده‌های تصویری، یا عدم توانایی مدل در یادگیری ویژگی‌های متمایزکننده برای کلاس‌های مختلف باشد. در عمل، این مدل برای تشخیص اکثر عیوب غیرقابل استفاده است.

ماتریس آشفتگی برای داده های آزمون مدل CNN



این ماتریس نشان می دهد که مدل CNN اولیه، در مقایسه با مدل MLP که قبلاً بررسی شد، عملکرد بسیار بهتری در تشخیص عیوب مختلف دارد. با این حال، هنوز هم جای بهبود وجود دارد.

نکات کلیدی و عملکرد هر کلاس:

هر ردیف نشان دهنده کلاس واقعی نمونه ها و هر ستون نشان دهنده کلاس پیش بینی شده توسط مدل است. اعداد روی قطر اصلی، تعداد نمونه هایی هستند که به درستی طبقه بندی شده اند.

فرض می کنیم هر کلاس دارای ۴۵ نمونه در مجموعه آزمون بوده است (با جمع کردن اعداد هر ردیف این موضوع تایید می شود).

• Crazing

- درست: ۴۰ نمونه به درستی crazing تشخیص داده شده اند.
- نادرست: ۵ نمونه به اشتباه rolled_in_scale پیش بینی شده اند.

○ دقت (Recall) برای Crazing: $\frac{40}{45} \approx 88.9\%$

• Inclusion

- درست: ۲۱ نمونه به درستی inclusion تشخیص داده شده‌اند.
- نادرست: ۶ نمونه به اشتباه pitted_surface و ۱۸ نمونه به اشتباه scratches پیش‌بینی شده‌اند.
- دقت (Recall) برای Inclusion: $46.7 \approx \frac{21}{45}$ (این کلاس عملکرد ضعیف‌تری دارد)

• Patches

- درست: ۲۷ نمونه به درستی patches تشخیص داده شده‌اند.
- نادرست: ۴ نمونه به اشتباه pitted_surface، ۱ نمونه rolled_in_scale و ۱۳ نمونه scratches پیش‌بینی شده‌اند.
- دقت (Recall) برای Patches: $60.0 \approx \frac{27}{45}$

• Pitted_surface

- درست: ۳۱ نمونه به درستی pitted_surface تشخیص داده شده‌اند.
- نادرست: ۶ نمونه به اشتباه inclusion و ۸ نمونه scratches پیش‌بینی شده‌اند.
- دقت (Recall) برای Pitted_surface: $68.9 \approx \frac{31}{45}$

• Rolled_in_scale

- درست: ۳۲ نمونه به درستی rolled_in_scale تشخیص داده شده‌اند.
- نادرست: ۱۱ نمونه به اشتباه inclusion و ۲ نمونه scratches پیش‌بینی شده‌اند.
- دقت (Recall) برای Rolled_in_scale: $71.1 \approx \frac{32}{45}$

• Scratches

- درست: ۴۳ نمونه به درستی scratches تشخیص داده شده‌اند.
- نادرست: ۲ نمونه به اشتباه inclusion پیش‌بینی شده‌اند.
- دقت (Recall) برای Scratches: $95.6 \approx \frac{43}{45}$ (این کلاس بهترین عملکرد را از نظر تشخیص صحیح نمونه‌های خود دارد)

عملکرد کلی مدل:

- تعداد کل نمونه‌ها: $45 \times 6 = 270$
- تعداد کل نمونه‌های درست طبقه‌بندی شده: $40 + 21 + 27 + 31 + 32 + 43 = 194$
- دقت کلی (Overall Accuracy): $\frac{194}{270} \approx 71.85$

نقاط قوت:

- مدل در تشخیص کلاس scratches و crazing عملکرد بسیار خوبی دارد.
- وقتی مدل کلاس‌های crazing یا patches را پیش‌بینی می‌کند، تقریباً همیشه درست می‌گوید (Precision بالا برای این دو کلاس در پیش‌بینی). برای مثال، تمام ۴۰ نمونه‌ای که crazing پیش‌بینی شده‌اند، واقعاً crazing بوده‌اند.

نقاط ضعف و زمینه‌های بهبود:

- کلاس inclusion: این کلاس ضعیف‌ترین عملکرد را دارد و اغلب با scratches و pitted_surface اشتباه گرفته می‌شود. این نشان می‌دهد که ویژگی‌های بصری inclusion برای مدل به اندازه کافی از این دو کلاس دیگر متمایز نیست.
 - اشتباهات با کلاس scratches: بسیاری از نمونه‌های کلاس‌های inclusion (۱۸ مورد) و patches (۱۳ مورد) به اشتباه به عنوان scratches طبقه‌بندی می‌شوند. این ممکن است به این معنی باشد که یا مدل بیش از حد به ویژگی‌های خطی و عمومی خراش‌ها حساس شده، یا اینکه این عیوب واقعاً در برخی موارد شباهت‌هایی با خراش دارند.
 - اشتباهات متقابل: بین inclusion و pitted_surface و همچنین inclusion و rolled_in_scale اشتباهات دو طرفه وجود دارد، هرچند با تعداد متفاوت.
- این مدل CNN یک نقطه شروع بسیار بهتر نسبت به مدل MLP است و نشان می‌دهد که معماری CNN پتانسیل خوبی برای این مسئله دارد. دقت کلی حدود ۷۲٪ قابل قبول است اما برای کاربردهای صنعتی حساس، نیاز به بهبود دارد. تمرکز اصلی برای بهبود باید روی افزایش دقت کلاس inclusion و کاهش موارد اشتباه گرفتن سایر کلاس‌ها با scratches باشد. این کار می‌تواند از طریق جمع‌آوری داده‌های بیشتر برای کلاس‌های ضعیف‌تر، استفاده از تکنیک‌های افزایش داده (data augmentation)، تنظیم دقیق‌تر معماری CNN، یا استفاده از روش‌های وزن‌دهی به خطاها (loss weighting) برای کلاس‌های با عملکرد پایین‌تر انجام شود.

استفاده از لایه Block Dropout

Block Dropout یا دقیق تر Spatial Dropout در Keras نوع خاصی از dropout هست که به جای حذف تصادفی نرون‌ها، کل feature map یا بلوک‌های مکانی رو حذف می‌کند.

چرا در CNN بهتره Spatial Dropout استفاده کنیم؟

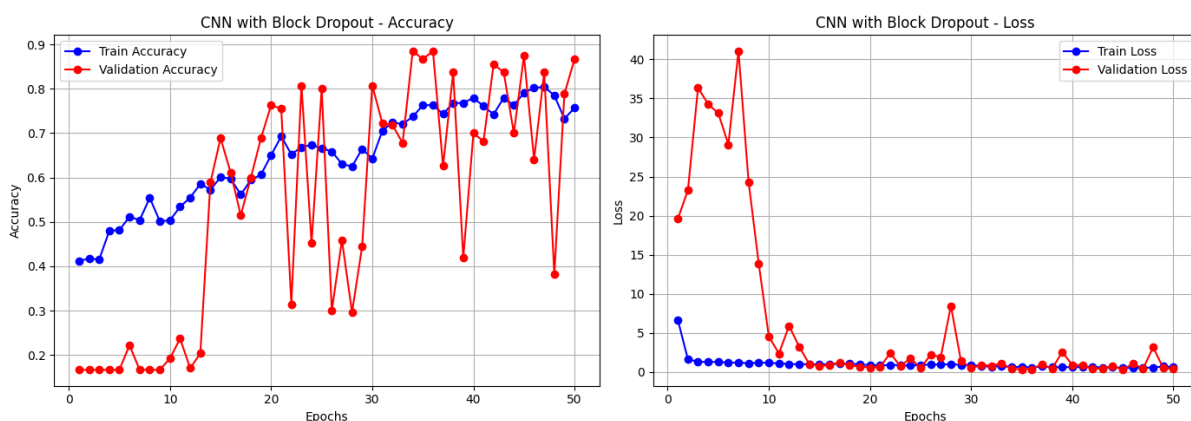
- چون لایه‌های کانولوشنی ساختار مکانی تصویر را حفظ می‌کنند
- Dropout نرمال ممکنه اختلالی در این ساختار ایجاد کند
- Spatial Dropout، کل map رو حذف می‌کند که باعث regularization موثرتر در شبکه‌های پیچشی می‌شود.

در این مرحله، لایه‌های Dropout معمولی با SpatialDropout2D جایگزین شدند. سپس مدل با تنظیم توقف زودهنگام و تعداد ایپاک و بیج سائز مشابه CNN اولیه آموزش دید.

دقت نهایی مدل به صورت زیر شد:

 **Block Dropout: 0.8741** دقت نهایی مدل CNN

همچنین نمودارهای دقت و خطا رسم شد:



۱. تحلیل نمودارهای CNN با Block Dropout

این تصویر دو نمودار دقت (Accuracy) و خطا (Loss) را برای یک مدل CNN که از Block Dropout استفاده کرده، طی ۵۰ دوره (Epoch) نمایش می‌دهد. خط آبی مربوط به داده‌های آموزشی و خط قرمز مربوط به داده‌های اعتبارسنجی است.

تحلیل نمودار دقت (سمت چپ):

- **دقت آموزشی (Train Accuracy - خط آبی):** دقت آموزشی از حدود ۴۰٪ شروع شده و به طور کلی روند صعودی دارد، با نوساناتی که در انتهای دوره‌ها بین ۷۵٪ تا ۸۰٪ و گاهی بالاتر (تا ۸۵٪) قرار می‌گیرد. این نشان می‌دهد که مدل به خوبی روی داده‌های آموزشی یاد می‌گیرد.
- **دقت اعتبارسنجی (Validation Accuracy - خط قرمز):** دقت اعتبارسنجی در ابتدا بسیار پایین است (حدود ۱۷٪)، اما به سرعت پس از دوره ۲۰-۱۵ شروع به افزایش می‌کند و به مقادیر بسیار بالا (تا ۹۰٪ در حدود دوره ۳۶-۳۷) می‌رسد. اگرچه نوساناتی را نشان می‌دهد، اما به طور کلی در سطح بسیار بالایی قرار دارد و حتی در برخی نقاط از دقت آموزشی نیز پیشی می‌گیرد. این رفتار بسیار مطلوب است.

تحلیل نمودار خطا (سمت راست):

- **خطای آموزشی (Train Loss - خط آبی):** خطای آموزشی از مقادیر اولیه (حدود ۷) به سرعت در چند دوره اول به نزدیکی صفر (تقریباً صفر) کاهش می‌یابد و در طول آموزش در همین سطح پایین باقی می‌ماند. این نشان‌دهنده انطباق عالی مدل با داده‌های آموزشی است.
- **خطای اعتبارسنجی (Validation Loss - خط قرمز):** خطای اعتبارسنجی از مقادیر بسیار بالا (حدود ۴۰) شروع شده و به سرعت در حدود ۱۵-۱۰ دوره اول به مقادیر بسیار پایین (نزدیک به صفر یا کمی بالاتر، مثلاً بین ۰.۵ تا ۱.۰) کاهش می‌یابد. در ادامه آموزش، این خطا در سطح پایین باقی می‌ماند. اگرچه چند اوج کوچک‌تر (مثلاً در دوره ۲۸-۳۰ و در انتهای آموزش) مشاهده می‌شود، اما این نوسانات در مقایسه با آنچه در CNN اولیه دیده شد، بسیار ناچیز و قابل کنترل هستند.

۲. مقایسه با CNN اولیه (بدون Block Dropout)

حال این نتایج را با نمودارهای CNN اولیه که قبلاً تحلیل کردیم (که تا ۴۰ دوره بود و ناپایداری و بیش‌برازش شدید داشت) مقایسه می‌کنیم:

۱. کاهش چشمگیر بیش‌برازش (Overfitting):

- **CNN اولیه:** بیش‌برازش شدید را نشان می‌داد. دقت آموزشی بالا بود اما دقت اعتبارسنجی بسیار پایین‌تر و به شدت ناپایدار بود (با سقوط به ۲۳٪ و نوسانات گسترده). خطای اعتبارسنجی نیز به شدت از خطای آموزشی واگرا شده و اوج‌های بسیار بزرگی (تا ۷۰) داشت.

- **CNN با Block Dropout:** بیش‌برازش به شکل قابل توجهی کاهش یافته است. دقت اعتبارسنجی به سطوح بسیار بالایی (حدود ۹۰٪) می‌رسد و به خوبی دقت آموزشی را دنبال می‌کند. خطای اعتبارسنجی نیز به مقادیر بسیار پایینی همگرا شده و تنها نوسانات بسیار کوچک‌تر (با اوج‌های حداکثر حدود ۴۰، اما عمدتاً کمتر از ۵) دارد. این نشان می‌دهد که Block Dropout به مدل کمک کرده است تا الگوهای قابل تعمیم را یاد بگیرد و کمتر جزئیات نویزگونه داده‌های آموزشی را حفظ کند.

۲. بهبود پایداری (Stability):

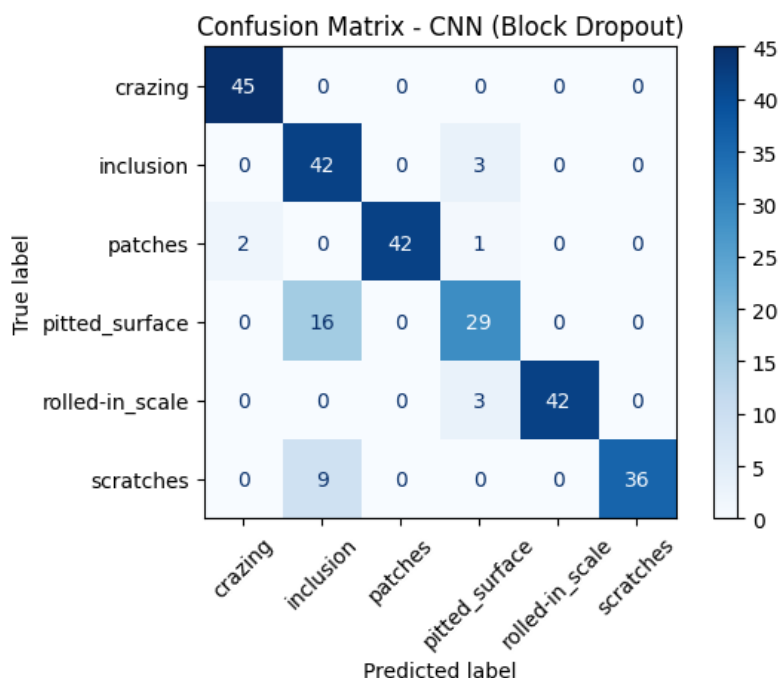
- **CNN اولیه:** عملکرد اعتبارسنجی (به ویژه خطا) بسیار بی‌ثبات و غیرقابل پیش‌بینی بود.
- **CNN با Block Dropout:** در حالی که دقت اعتبارسنجی هنوز نوساناتی دارد، اما این نوسانات در محدوده عملکرد بسیار بالاتر و با روند کلی مثبت‌تر هستند. خطای اعتبارسنجی نیز به طور قابل توجهی پایدارتر است و عمدتاً در سطح پایین باقی می‌ماند، که نشان‌دهنده یک فرآیند آموزش قابل اعتمادتر و تعمیم‌پذیری بهتر است.

۳. عملکرد کلی و تعمیم‌پذیری:

- **CNN اولیه:** با وجود دقت آموزشی بالا، عملکرد آن بر روی داده‌های جدید نامطمئن و ضعیف بود.
- **CNN با Block Dropout:** این مدل به دقت‌های اعتبارسنجی بسیار بالا (نزدیک به ۹۰٪) و خطاهای اعتبارسنجی پایین و پایدار دست یافته است. این نشان می‌دهد که مدل قادر است به طور موثر به داده‌های جدید و دیده نشده تعمیم یابد و برای وظیفه تشخیص عیوب عملکرد قابل اطمینانی ارائه دهد.

استفاده از **Block Dropout** تأثیر بسیار مثبتی بر عملکرد مدل CNN در این وظیفه طبقه‌بندی داشته است. این تکنیک با موفقیت توانسته بیش‌برازش شدید و ناپایداری مشاهده شده در مدل CNN اولیه را کاهش دهد. در نتیجه، مدل CNN با Block Dropout به یک مدل بسیار قوی‌تر و قابل اطمینان‌تر تبدیل شده است که توانایی تعمیم بهتری به داده‌های جدید دارد. این مدل با دقت اعتبارسنجی بالا و خطای اعتبارسنجی پایین، گزینه بسیار مناسب‌تری برای کاربرد عملی در تشخیص عیوب سطحی فولاد است.

ماتریس آشفستگی:



این ماتریس پیشرفت قابل توجهی را نسبت به مدل CNN اولیه (بدون Block Dropout) نشان می دهد.

• Crazing

○ درست: ۴۵ نمونه به درستی crazing تشخیص داده شده اند.

○ نادرست: ۰ نمونه به اشتباه طبقه بندی شده اند.

○ دقت (Recall) برای Crazing: $\frac{45}{45} \approx 100$

• Inclusion

○ درست: ۴۲ نمونه به درستی inclusion تشخیص داده شده اند.

○ نادرست: ۳ نمونه به اشتباه pitted_surface پیش بینی شده اند.

○ دقت (Recall) برای Inclusion: $\frac{42}{45} \approx 93.3$ (بهبود بسیار زیاد نسبت به مدل اولیه)

• Patches

○ درست: ۴۲ نمونه به درستی patches تشخیص داده شده اند.

○ نادرست: ۲ نمونه به اشتباه crazing و ۱ نمونه pitted_surface پیش بینی شده اند.

○ دقت (Recall) برای Patches: $\frac{42}{45} \approx 93.3$ (بهبود بسیار زیاد)

• Pitted_surface

- درست: ۲۹ نمونه به درستی pitted_surface تشخیص داده شده‌اند.
- نادرست: ۱۶ نمونه به اشتباه inclusion پیش‌بینی شده‌اند.
- دقت (Recall) برای Pitted_surface: $64.4 \approx \frac{29}{45}$ (این کلاس همچنان چالش‌برانگیزترین کلاس است و عملکرد آن کمی افت داشته)

• Rolled_in_scale

- درست: ۴۲ نمونه به درستی rolled_in_scale تشخیص داده شده‌اند.
- نادرست: ۳ نمونه به اشتباه pitted_surface پیش‌بینی شده‌اند.
- دقت (Recall) برای Rolled_in_scale: $93.3 \approx \frac{42}{45}$ (بهبود بسیار زیاد)

• Scratches

- درست: ۳۶ نمونه به درستی scratches تشخیص داده شده‌اند.
- نادرست: ۹ نمونه به اشتباه inclusion پیش‌بینی شده‌اند.
- دقت (Recall) برای Scratches: $80.0 \approx \frac{36}{45}$ (کاهش نسبت به مدل اولیه، اما دقت پیش‌بینی (Precision) آن ۱۰۰٪ شده است)

عملکرد کلی مدل:

- تعداد کل نمونه‌ها: ۲۷۰
- تعداد کل نمونه‌های درست طبقه‌بندی شده: ۲۳۶
- دقت کلی (Overall Accuracy): $87.4 \approx \frac{236}{270}$
- مقایسه با مدل CNN اولیه (دقت کلی حدود ۷۱.۸۵٪): پیشرفت چشمگیری در دقت کلی مشاهده می‌شود.
- استفاده از **Block Dropout** تأثیرات مثبت قابل توجهی بر عملکرد مدل داشته است:
- بهبود چشمگیر در کلاس‌های ضعیف: کلاس‌هایی مانند inclusion (از ۴۶.۷٪ به ۹۳.۳٪)، patches (از ۶۰٪ به ۹۳.۳٪) و rolled_in_scale (از ۷۱.۱٪ به ۹۳.۳٪) بهبود فوق‌العاده‌ای در تشخیص صحیح نمونه‌های خود (Recall) داشته‌اند.
- عملکرد عالی برای crazing: این کلاس اکنون با دقت ۱۰۰٪ تشخیص داده می‌شود.

- کاهش سردرگمی‌ها: بسیاری از اشتباهات رایج در مدل اولیه، مانند اشتباه گرفتن تعداد زیادی از نمونه‌ها با scratches، به طور قابل توجهی کاهش یافته است.

- دقت پیش‌بینی (Precision) عالی برای برخی کلاس‌ها: وقتی مدل کلاس‌های patches، rolled_in_scale یا scratches را پیش‌بینی می‌کند، این پیش‌بینی ۱۰۰٪ صحیح است. این یعنی هیچ نمونه‌ای از کلاس‌های دیگر به اشتباه این سه کلاس پیش‌بینی نشده است.

نقاط ضعف باقی‌مانده:

- کلاس Pitted_surface: این کلاس با دقت ۶۴.۴٪ همچنان ضعیف‌ترین عملکرد را دارد و تعداد زیادی از نمونه‌های آن (۱۶ مورد) به اشتباه inclusion تشخیص داده می‌شوند. این اصلی‌ترین نقطه ضعف فعلی مدل است.

- اشتباه scratches با inclusion: ۹ نمونه از scratches به اشتباه inclusion تشخیص داده شده‌اند. هرچند Recall کلی scratches کمی کاهش یافته (از ۹۵.۶٪ به ۸۰٪)، اما Precision آن به ۱۰۰٪ رسیده است (یعنی هرچه را scratches بگوئید، واقعاً scratches است). این یک نوع مصالحه (trade-off) است.

تکنیک Block Dropout به طور موثری به بهبود قابلیت تعمیم و کاهش بیش‌برازش در مدل CNN کمک کرده و منجر به افزایش قابل توجه دقت کلی و بهبود چشمگیر در تشخیص اکثر کلاس‌ها شده است. مدل حال حاضر بسیار متوازن‌تر عمل می‌کند. چالش اصلی باقی‌مانده، بهبود تشخیص کلاس pitted_surface و کاهش اشتباه گرفتن آن با inclusion است. این نتایج نشان می‌دهد که انتخاب صحیح روش‌های تنظیم (regularization) می‌تواند تأثیر بسزایی در عملکرد نهایی مدل‌های یادگیری عمیق داشته باشد.

پیاده سازی Kernel Factorization

یعنی به جای استفاده از فیلترهای مربعی سنگین (مثلاً ۵×۵ یا ۳×۳)، آن‌ها را به چند فیلتر ساده‌تر تجزیه کنی، رایج‌ترین حالت:

$$\text{Conv}(3 \times 3) \approx \text{Conv}(1 \times 3) \rightarrow \text{Conv}(3 \times 1)$$

Conv(1×3) روی افق کار می‌کند و Conv(3×1) روی عمود سپس با هم ترکیب می‌شوند و تقریباً همان عملکرد را دارند، ولی با پارامتر کمتر و سرعت بیشتر، کاهش مصرف حافظه و زمان از مزایای آن است.

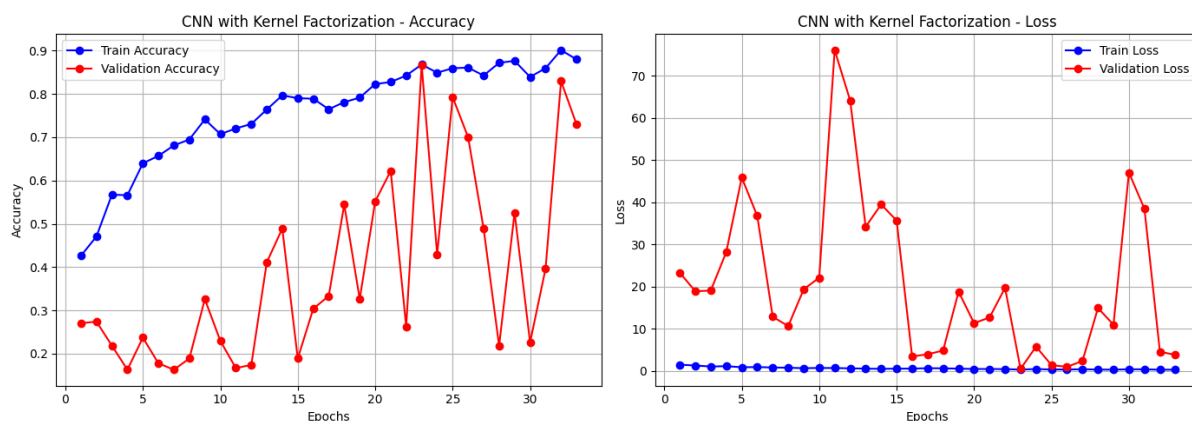
پس همان مدل CNN اولیه را به جای لایه‌های کانولوشن ۳در۳ دو لایه کانولوشن ۳ در ۳ و ۳ در ۱ جایگزین می‌کنیم و بقیه لایه‌ها مثل همان مدل می‌مانند.

سپس با تنظیم توقف زود هنگام و تعداد ۵۰ اپاک و بچ سایز ۳۲ مدل را آموزش می دهیم.

دقت نهایی مدل به شرح زیر است:

Kernel Factorization: 0.8667 دقت نهایی مدل

همچنین نمودار های دقت و خطا رسم شد:



نمودار دقت (Accuracy - سمت چپ):

- **دقت آموزشی (Train Accuracy - خط آبی):** این دقت به طور پیوسته افزایش می یابد و از حدود ۴۲٪ در ابتدا به نزدیک ۹۰٪ در انتهای ۳۲ دوره می رسد. این نشان می دهد که مدل به خوبی داده های آموزشی را یاد گرفته است.
- **دقت اعتبارسنجی (Validation Accuracy - خط قرمز):** این دقت نوسانات بسیار شدیدی دارد. با وجود اینکه در برخی نقاط به پیک های بالایی می رسد (مثلاً حدود ۸۵٪ در دوره ۲۲ یا ۸۳٪ در دوره ۳۱)، اما این پیک ها پایدار نیستند و افت های شدیدی نیز مشاهده می شود (مثلاً به حدود ۲۰٪ در دوره های مختلف).
- **بیش برآزش (Overfitting):** شکاف قابل توجهی بین دقت آموزشی و اعتبارسنجی از همان ابتدا وجود دارد و با افزایش دوره ها، این شکاف بیشتر هم می شود (هرچند دقت اعتبارسنجی گاهی به دقت آموزشی نزدیک می شود). این نشان دهنده بیش برآزش است.

نمودار خطا (Loss - سمت راست):

- **خطای آموزشی (Train Loss - خط آبی):** این خطا به سرعت کاهش یافته و به مقادیر بسیار نزدیک به صفر رسیده است که یادگیری خوب روی داده های آموزشی را تأیید می کند.

- خطای اعتبارسنجی (Validation Loss - خط قرمز): این خطا به شدت ناپایدار و دارای نوسانات بسیار شدید است. مقادیر آن جهش‌های بزرگی را تجربه می‌کنند (مثلاً تا بیش از ۷۰٪) و هیچ روند کاهشی پایداری مشاهده نمی‌شود. این موضوع، مشکل جدی در پایداری و قابلیت تعمیم مدل را نشان می‌دهد.

مدل CNN با Kernel Factorization توانسته داده‌های آموزشی را به خوبی یاد بگیرد، اما از بیش‌برازش شدید و ناپایداری بسیار زیاد در عملکرد اعتبارسنجی رنج می‌برد. نوسانات شدید در دقت و خطای اعتبارسنجی نشان می‌دهد که عملکرد مدل روی داده‌های جدید بسیار غیرقابل پیش‌بینی و نامطمئن است. این ناپایداری می‌تواند حتی شدیدتر از مدل CNN اولیه‌ای باشد که قبلاً بررسی شد (که آن هم ناپایدار بود).

ویژگی	CNN (اولیه)	CNN (Block Dropout)	CNN (Kernel Factorization)
دقت کلی (تخمینی)	حدود ۷۱.۸۵٪	حدود ۸۷.۴۱٪	پیک‌ها تا ۸۶٪، اما بسیار ناپایدار
پایداری اعتبارسنجی	ناپایدار، نوسانات قابل توجه	به طور قابل توجهی پایدارتر	بسیار ناپایدار، نوسانات شدید و غیرقابل پیش‌بینی
بیش‌برازش	شدید	به طور مؤثری کاهش یافته	شدید
یادگیری آموزشی	خوب	خوب	خوب

- CNN با Block Dropout بهترین عملکرد را داشته است. این مدل نه تنها به دقت کلی بالاتری دست یافته، بلکه عملکرد بسیار پایدارتر و با بیش‌برازش کمتری را نیز نشان داده است. ماتریس درهم‌ریختگی آن نشان‌دهنده بهبود قابل توجه در تشخیص اکثر کلاس‌ها بود.

- CNN (اولیه) و CNN با Kernel Factorization هر دو از ناپایداری و بیش‌برازش شدید رنج می‌برند. اگرچه ممکن است در برخی دوره‌ها به دقت‌های اعتبارسنجی بالایی دست یابند، اما این عملکرد قابل اعتماد نیست. نوسانات در مدل Kernel Factorization به نظر می‌رسد حتی شدیدتر از مدل اولیه باشد.

- تکنیک Kernel Factorization در این مورد خاص، به نظر نمی‌رسد که به بهبود پایداری یا کاهش بیش‌برازش کمک کرده باشد و حتی ممکن است وضعیت را بدتر کرده باشد. در مقابل، Block Dropout یک روش تنظیم (regularization) بسیار مؤثر برای این مسئله بوده است. بنابراین، بر اساس تحلیل‌های انجام شده، مدل CNN با Block Dropout گزینه برتر برای این مسئله طبقه‌بندی عیوب سطحی فولاد است.

۷-۲- تحلیل

بر اساس تحلیل‌های ارائه شده، مدل **CNN با Block Dropout** بهترین عملکرد را در این مسئله داشته است.

۱. **دقت کلی بالاتر:** این مدل به دقت کلی تقریباً 87.41% دست یافت که به طور قابل توجهی بالاتر از سایر مدل‌ها بود (MLP بسیار ضعیف عمل کرد، CNN اولیه حدود 71.85% و CNN با Kernel Factorization اگرچه پیک‌های بالایی داشت اما بسیار ناپایدار بود).

۲. **کاهش مؤثر بیش‌برازش (Overfitting):** در حالی که مدل‌های CNN اولیه و CNN با Kernel Factorization از بیش‌برازش شدید و ناپایداری در منحنی‌های اعتبارسنجی رنج می‌بردند، تکنیک Block Dropout توانست این مشکل را به طور مؤثری کنترل کند. این منجر به مدلی شد که قابلیت تعمیم به داده‌های جدید و دیده نشده دارد.

۳. **بهبود قابل توجه در تشخیص کلاس‌های ضعیف:** مدل CNN با Block Dropout توانست دقت تشخیص (Recall) را برای کلاس‌هایی که در مدل CNN اولیه ضعیف عمل کرده بودند (مانند inclusion, patches, rolled_in_scale) به شدت بهبود بخشد. برای مثال، Recall کلاس inclusion از حدود 47% به بیش از 93% افزایش یافت.

۴. **عملکرد متوازن‌تر:** ماتریس درهم‌ریختگی مدل با Block Dropout نشان‌دهنده توزیع خطای متعادل‌تری بود و تمایل کمتری به پیش‌بینی غالب یک یا چند کلاس خاص (مانند تمایل مدل MLP به پیش‌بینی scratches) داشت.

۵. **پایداری بیشتر:** برای مدل Block Dropout، ماتریس درهم‌ریختگی نهایی آن نشان‌دهنده یک مدل پایدارتر و قابل اعتمادتر نسبت به مدل CNN اولیه و به خصوص مدل CNN با Kernel Factorization است که نوسانات بسیار شدیدی در عملکرد اعتبارسنجی داشت.

خلاصه عملکرد سایر مدل‌ها:

- **MLP:** عملکرد بسیار ضعیفی داشت و تقریباً تمام نمونه‌ها را به اشتباه به یک یا دو کلاس خاص نسبت می‌داد. این نشان می‌دهد که معماری MLP برای این نوع داده‌های تصویری پیچیده مناسب نیست.
- **CNN (اولیه):** عملکرد بهتری نسبت به MLP داشت اما از بیش‌برازش شدید و ناپایداری در منحنی‌های اعتبارسنجی رنج می‌برد.

- **CNN با Kernel Factorization:** این مدل نیز مانند CNN اولیه، داده‌های آموزشی را خوب یاد گرفت اما در عملکرد اعتبارسنجی دچار بیش‌برازش و ناپایداری شدید بود، حتی شاید شدیدتر از CNN اولیه.

بنابراین، **CNN با Block Dropout** به دلیل توانایی‌اش در دستیابی به دقت بالا، کنترل مؤثر بیش‌برازش و بهبود تشخیص کلاس‌های چالش‌برانگیز، به عنوان بهترین مدل در این مقایسه شناخته می‌شود. تکنیک Block Dropout به عنوان یک روش تنظیم مناسب برای این مسئله عمل کرده است.

بخش سوم: یادگیری انتقالی (Transfer Learning)

استفاده از شبکه ای که قبلاً روی داده های مشابه ای آموزش دیده است یکی از روشهای متداول برای افزایش دقت و تعمیم پذیری شبکه است. در این روش که پیش تربیت (pre-training) نامیده می شود ابتدا شبکه روی داده های مشابه ای آموزش داده می شود و سپس ضمن ثابت نگه داشتن ضرایب وزنی بقیه شبکه، یک یا چند لایه نهایی آن با لایه های جدیدی جایگزین شده و تنها ضرایب وزنی این لایه های جدید تعیین می شود. سپس در ادامه فرآیند آموزش به تدریج به لایه های قبلی نیز اجازه داده میشود که ضرایب وزنی خود را بروزآوری کنند.

در این بخش می خواهیم به کمک یادگیری انتقالی، عملکرد مدل تشخیص عیوب سطحی را بهبود ببخشیم. برای این کار از یک مدل از پیش آموزش دیده (ResNet-50) استفاده خواهیم کرد.

ورودی ResNet-50 باید سایز ۲۲۴ در ۲۲۴ و ۳ کاناله (RGB) باشد، مدل ResNet انتظار دارد داده ها در قالب ImageNet-style باشند، از ImageDataGenerator برای data augmentation استفاده می کنیم.

۳-۱- آماده سازی داده ها

برای آماده سازی داده های ورودی به مدل ResNet-50، تصاویر موجود در مسیرهای train و validation به کمک کلاس ImageDataGenerator نرمال سازی شدند. در داده های آموزش از تکنیک های افزایش داده مانند چرخش، برش، جابجایی، و وارون سازی افقی برای افزایش تعمیم پذیری مدل استفاده شد. داده های اعتبارسنجی بدون تغییر و صرفاً با نرمال سازی مورد استفاده قرار گرفتند.

ابتدا مسیر داده های آموزش و آزمون را تعریف میکنیم، و از داده افزایی به منظور بهبود فرآیند آموزش، افزایش تعمیم پذیری مدل و ارتقاء مقاومت به نویز استفاده میکنیم، اینجا هدف افزایش تنوع داده هاست برای آموزش بهتر و جلوگیری از overfitting با اعمال تغییراتی مثل:

پارامتر	توضیح
rescale=1./255	نرمال سازی پیکسل ها به بازه [0,1]
rotation_range=30	چرخش تصادفی تصویر تا ۳۰ درجه
width_shift_range=0.2	جابجایی عرضی (افقی) تا ۲۰٪
height_shift_range=0.2	جابجایی عمودی تا ۲۰٪

بزرگ‌نمایی تا ۳۰٪	zoom_range=0.3
اعمال shear کج کردن تصویر	shear_range=0.2
وارون‌سازی افقی	horizontal_flip=True
پُر کردن فضای خالی با نزدیک‌ترین پیکسل‌ها	fill_mode='nearest'

آماده‌سازی داده اعتبارسنجی (بدون افزایش داده): فقط نرمال‌سازی پیکسل نباید روی validation augmentation انجام بدیم چون می‌خواهیم عملکرد واقعی مدل را روی داده‌های تغییرنیافته بسنجیم.

ساخت generator برای آموزش و اعتبارسنجی:

پارامتر	توضیح
flow_from_directory(...)	داده‌ها را مستقیماً از ساختار پوشه‌ها می‌خواند
target_size=(224, 224)	تغییر اندازه‌ی همه تصاویر به ۲۲۴×۲۲۴ سایز ورودی ResNet-50
batch_size=32	تعداد تصاویر در هر batch
class_mode='categorical'	لیبل‌ها به صورت one-hot کدگذاری می‌شوند برای softmax
shuffle=True	فقط در train فعال شده (برای افزایش تنوع و جلوگیری از یادگیری ترتیب)

استخراج نام کلاس‌ها: از روی پوشه‌های داخل train_dir لیبل‌های کلاس‌ها رو تشخیص می‌دهد.

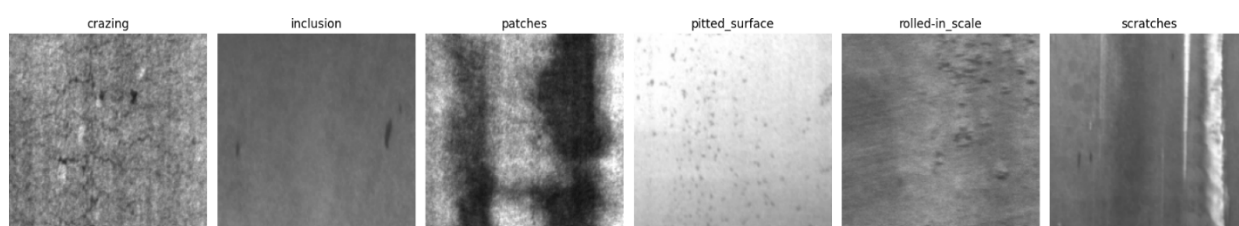
Found 1440 images belonging to 6 classes.

Found 360 images belonging to 6 classes.

کلاس‌ها: ['crazing', 'inclusion', 'patches', 'pitted_surface', 'rolled-in_scale', 'scratches'] :

با بررسی توزیع نمونه‌ها در کلاس‌های داده‌ی NEU، مشخص شد که تعداد نمونه‌ها در کلاس‌های مختلف تقریباً برابر است. در نتیجه، مقادیر محاسبه‌شده برای class weights همگی برابر با ۱ بودند و مدل بدون نیاز به جبران عدم‌توازن آموزش داده شد.

به صورت تصادفی از هر کلاس یک نمونه را به همراه برجسب مربوطه نمایش داده شد، نتیجه:



۳-۲-آماده سازی مدل

به منظور بهره گیری از یادگیری انتقالی، مدل ResNet-50 با وزن های آموزش دیده روی ImageNet بارگذاری شد. لایه های نهایی آن حذف گردید و فقط دو لایه ی جدید شامل یک لایه Dense با ۱۲۸ نورون و یک لایه خروجی Softmax با ۶ کلاس به آن افزوده شد. در این مرحله، وزن های ResNet-50 ثابت نگه داشته شده اند و فقط وزن های این دو لایه جدید آموزش داده می شوند.

بارگذاری ResNet-50 با وزن های ImageNet

حذف لایه ی خروجی اصلی (1000-class softmax)

اضافه کردن:

- لایه Dense جدید (برای یادگیری ویژگی های جدید) (Relu,128)
- لایه خروجی Dense(6, activation='softmax') برای دسته بندی عیوب
- Dropout 0.2

در این پروژه از لایه ی GlobalAveragePooling2D استفاده شد که با خلاصه سازی خروجی ویژگی های کانولوشنی، تعداد پارامترها را کاهش داده و احتمال overfitting را در مجموعه داده ی کوچک کاهش می دهد. این روش نسبت به Flatten بسیار سبک تر و پایدارتر است و معمولاً برای یادگیری انتقالی توصیه می شود.

۳-۳-آموزش

آموزش مدل ResNet-50 در دو مرحله انجام شد. ابتدا تنها دو لایه ی جدید خروجی آموزش داده شد، در حالی که تمامی لایه های شبکه ی پایه فریز شده بودند. سپس در مرحله ی Fine-Tuning، تعدادی از لایه های انتهایی ResNet-50 از حالت فریز خارج شده و با نرخ یادگیری پایین به صورت تدریجی آموزش داده شدند تا مدل بهتر با داده های جدید سازگار شود.

مرحله ۱: آموزش فقط لایه های جدید (Head)

- لایه های ResNet-50 فریز باشند
 - فقط لایه های Dense و Softmax که اضافه کردیم آموزش ببینند
- در این مرحله بخش Head با تعداد ۲۵ ایپاک و توقف زودهنگام با صبر ۵ آموزش دید.

مرحله ۲: Fine-Tuning

- به تدریج از انتهای ResNet-50 (آخرین لایه‌ها) فریز برداشته می‌شود.
 - چند لایه‌ی آخر شبکه اصلی هم آموزش می‌بینند (با نرخ یادگیری پایین)
- در این مرحله به تدریج از انتهای شبکه به ترتیب ۵، ۱۰ و ۱۵ لایه آخر باز شده و با تعداد ۱۵ اپاک و توقف زود هنگام با صبر ۵ آموزش دیدند.

۳-۴-ارزیابی

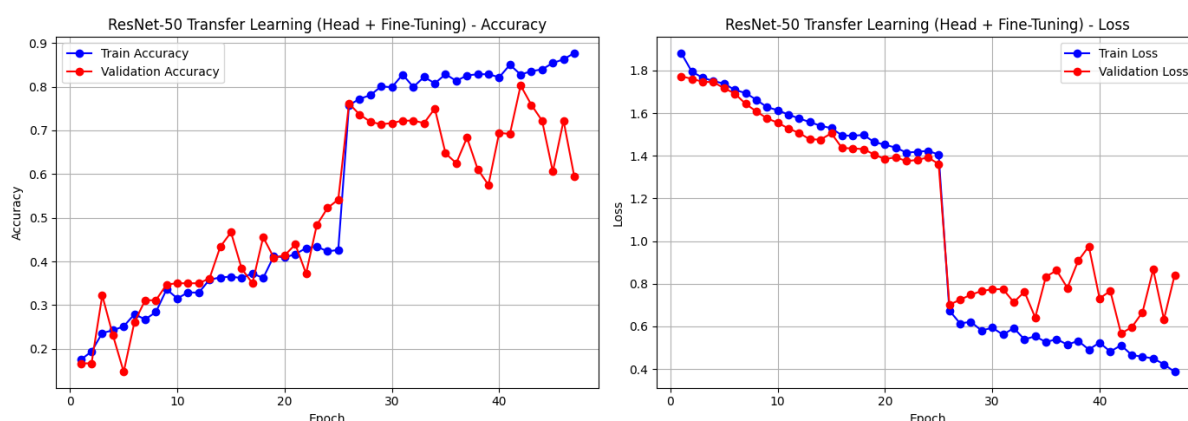
ابتدا نتایج دو بخش آموزش را به هم متصل می‌کنیم.

سپس دقت نهایی مدل محاسبه شد که به شرح زیر می باشد:



Final Accuracy of ResNet-50 (Fine-Tuned): 0.8028 | Loss: 0.5669

همچنین نمودارهای دقت و خطا رسم شد:



رفتار مدل به وضوح به دو فاز مجزا تقسیم می‌شود که با یک جهش ناگهانی در حدود دوره ۲۵ از هم جدا شده‌اند.

فاز اول: آموزش لایه‌های نهایی (Head) (دوره‌های ۱ تا ۲۵)

- رفتار مدل: در این فاز، بدنه‌ی اصلی مدل ResNet-50 (که با داده‌های ImageNet از قبل آموزش دیده) فریز (Freeze) شده و فقط لایه‌های طبقه‌بندی کننده نهایی (Head) آموزش می‌بینند.
- نمودار دقت: دقت آموزشی (آبی) و دقت اعتبارسنجی (قرمز) هر دو به آرامی و با یک روند منطقی در حال افزایش هستند. دقت آموزشی به حدود ۴۳٪ و دقت اعتبارسنجی به حدود ۵۰٪ می‌رسد.

- **نمودار خطا:** خطای آموزشی و اعتبارسنجی نیز به طور پیوسته در حال کاهش هستند.

در این مرحله، مدل در حال یادگیری اولیه برای انطباق با داده‌های جدید است و عملکردی متوسط دارد.

نقطه عطف: شروع تنظیم دقیق (Fine-Tuning) (حدود دوره ۲۵)

- در این نقطه، یک تغییر بزرگ در استراتژی آموزش رخ داده است. برخی از لایه‌های بالایی مدل ResNet از حالت فریز خارج شده (Unfreeze) و نرخ یادگیری (Learning Rate) به شدت کاهش یافته است. این کار به مدل اجازه می‌دهد تا ویژگی‌های جزئی‌تری را از داده‌های جدید یاد بگیرد.

• جهش در عملکرد:

- **دقت (Accuracy):** یک جهش ناگهانی و بسیار بزرگ در دقت آموزشی (از ۴۳٪ به ۷۷٪) و دقت اعتبارسنجی (از ۴۰٪ به ۷۴٪) مشاهده می‌شود.
- **خطا (Loss):** به طور متناظر، خطای آموزشی و اعتبارسنجی نیز یک سقوط ناگهانی و شدید را تجربه می‌کنند (از حدود ۱.۴ به ۰.۷-۰.۶).

این جهش، موفقیت‌آمیز بودن تغییر استراتژی به Fine-Tuning را نشان می‌دهد.

فاز دوم: تنظیم دقیق (Fine-Tuning) (دوره‌های ۲۵ تا ۴۸)

- **یادگیری آموزشی:** دقت آموزشی (آبی) به طور پیوسته به افزایش خود ادامه می‌دهد و به نزدیکی ۹۰٪ می‌رسد، در حالی که خطای آموزشی به طور مداوم تا حدود ۰.۴ کاهش می‌یابد. این نشان می‌دهد مدل همچنان در حال یادگیری کامل داده‌های آموزشی است.
- **عملکرد اعتبارسنجی (قرمز):** اینجاست که مشکل اصلی مدل مشخص می‌شود:

- **دقت اعتبارسنجی:** با وجود اینکه به پیک‌های بالایی می‌رسد (حدود ۸۵٪ در دوره ۴۲)، اما دچار نوسانات بسیار شدید و عدم پایداری است. عملکرد مدل به شدت افت و خیز دارد.
- **خطای اعتبارسنجی:** پس از سقوط اولیه، دیگر روند کاهشی پایداری ندارد و بین ۰.۶ تا ۱.۰ نوسان می‌کند.

- **بیش‌برازش (Overfitting):** شکاف بین منحنی‌های آبی و قرمز پس از شروع Fine-Tuning به وضوح افزایش می‌یابد. این نشان می‌دهد که مدل در حال بیش‌برازش روی داده‌های آموزشی است و عملکرد پایدار و قابل تعمیمی روی داده‌های جدید ندارد.

تفسیر و نتیجه‌گیری نهایی

- **قدرت یادگیری انتقال:** این نمودارها به خوبی قدرت تکنیک Transfer Learning را نشان می‌دهند. مدل توانسته با استفاده از دانش قبلی خود (از ImageNet) و سپس تنظیم دقیق، به سطوح دقت بسیار بالایی دست یابد.
- **مشکل اصلی: ناپایداری و بیش‌برازش:** با وجود رسیدن به پیک‌های دقت بالا، مدل در فاز Fine-Tuning بسیار ناپایدار است. این یعنی عملکرد آن روی دسته‌های مختلفی از داده‌های جدید می‌تواند بسیار متفاوت و غیرقابل اعتماد باشد.

مقایسه با مدل‌های قبلی

- **مقایسه با CNN با Block Dropout:**

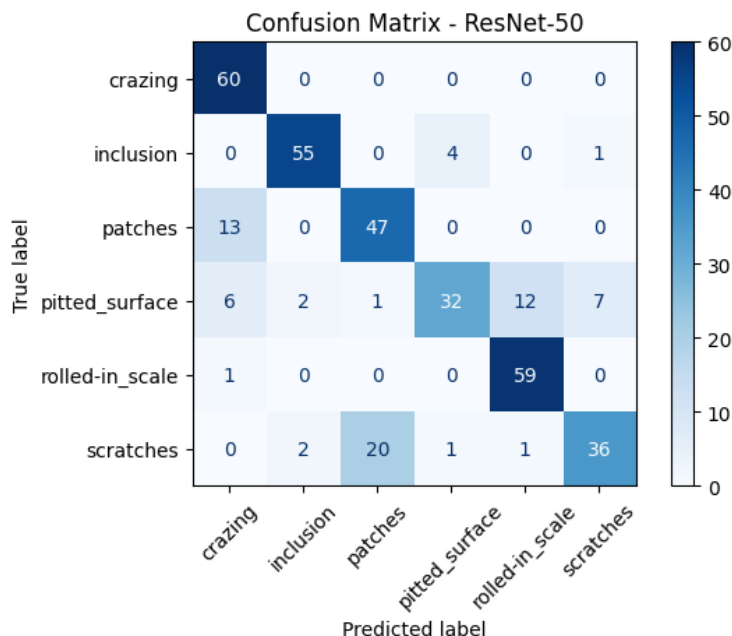
○ ResNet-50 (این مدل): به پیک دقت بالاتری (حدود ۹۰-۸۵٪) دست یافت، اما با هزینه و ناپایداری زیاد.

○ CNN با Block Dropout: به پیک دقت کمی پایین‌تر (حدود ۸۷٪) رسید، اما عملکرد آن بسیار پایدارتر و قابل اعتمادتر بود.

اگر هدف صرفاً رسیدن به بالاترین دقت ممکن باشد و بتوان مدل را در نقطه اوج عملکردش ذخیره کرد، ResNet-50 برنده است. اما اگر هدف یک مدل پایدار و قابل اعتماد با عملکردی خوب و قابل پیش‌بینی باشد، مدل CNN با Block Dropout گزینه بهتری است.

در بسیاری از کاربردهای صنعتی، پایداری و قابلیت اطمینان به اندازه رسیدن به بالاترین پیک دقت اهمیت دارد، که این موضوع برتری مدل CNN با Block Dropout را در یک سناریوی عملی پررنگ‌تر می‌کند.

ماتریس آشفتگی:



این مدل به دقت کلی تقریباً ۸۰.۳٪ دست یافته است (۲۸۹ پیش‌بینی صحیح از مجموع ۳۶۰ نمونه). عملکرد مدل برای کلاس‌های مختلف بسیار متفاوت است، به طوری که در تشخیص برخی عیوب عالی و در برخی دیگر ضعیف عمل می‌کند.

- **Crazing** عملکرد عالی.

- درست: ۶۰ نمونه به درستی تشخیص داده شده‌اند.

- نادرست: ۰ نمونه خطا داشته است.

- دقت (Recall): ۱۰۰٪

- **Inclusion** عملکرد بسیار خوب.

- درست: ۵۵ نمونه به درستی تشخیص داده شده‌اند.

- نادرست: ۴ نمونه با pitted_surface و ۱ نمونه با scratches اشتباه گرفته شده است.

- دقت (Recall): ۹۱.۷٪

- **Patches** عملکرد قابل قبول.

- درست: ۴۷ نمونه به درستی تشخیص داده شده‌اند.

- نادرست: ۱۳ نمونه به اشتباه crazing تشخیص داده شده‌اند.

- دقت (Recall): ۷۸.۳٪

- **Pitted_surface** عملکرد ضعیف.

- درست: تنها ۳۲ نمونه به درستی تشخیص داده شده‌اند.

- نادرست: این کلاس با تمام کلاس‌های دیگر اشتباه گرفته شده است، به خصوص با

- rolled-in_scale (۱۲ مورد) و scratches (۷ مورد).

- دقت (Recall): ۵۳.۳٪

- **Rolled_in_scale** عملکرد عالی.

- درست: ۵۹ نمونه به درستی تشخیص داده شده‌اند.

- نادرست: تنها ۱ نمونه خطا داشته است.

- دقت (Recall): ۹۸.۳٪

• Scratches عملکرد متوسط.

- درست: ۳۶ نمونه به درستی تشخیص داده شده‌اند.
 - نادرست: تعداد زیادی از نمونه‌ها (۲۰ مورد) به اشتباه patches تشخیص داده شده‌اند.
 - دقت (Recall): ۶۰.۰٪
- در ماتریس آشفتگی به خوبی ناپایداری مشاهده شده در نمودارهای یادگیری ResNet-50 را تأیید می‌کند. در حالی که مدل در تشخیص کلاس‌های crazing و rolled_in_scale بسیار قدرتمند است، اما در تفکیک کلاس‌های pitted_surface و scratches دچار مشکل می‌شود.

مقایسه با مدل CNN با Block Dropout:

- مدل ResNet-50 به دقت کلی ۸۰.۳٪ رسید.
 - مدل CNN با Block Dropout به دقت کلی ۸۷.۴٪ رسیده بود.
- اگرچه ResNet-50 یک معماری بسیار قدرتمند است، اما در این مسئله خاص، مدل ساده‌تر CNN با Block Dropout عملکردی بهتر، متعادل‌تر و قابل اعتمادتر از خود نشان داد. ضعف ResNet-50 در تشخیص pitted_surface و scratches باعث کاهش دقت کلی آن در مقایسه با مدل دیگر شده است. که با تنظیمات بهینه‌تر احتمالاً رفع شود اما زمان آموزش طولانی‌تر کمی کار را سخت می‌کند.

۳-۴- مقایسه و تحلیل

۱. مقایسه دقت نهایی و زمان آموزش

• دقت نهایی (Final Accuracy):

- CNN با Block Dropout: دقت کلی ۸۷.۴٪ را با عملکردی متعادل و پایدار در تمام کلاس‌ها به دست آورد.
- ResNet-50 (یادگیری انتقالی): دقت کلی پایین‌تری معادل ۸۰.۳٪ داشت. این مدل در تشخیص برخی کلاس‌ها عالی بود اما در برخی دیگر (مانند pitted_surface و scratches) بسیار ضعیف عمل کرد که نشان‌دهنده عدم پایداری است.

- زمان آموزش (Training Time):

- CNN با Block Dropout: به دلیل معماری ساده‌تر و کوچک‌تر، زمان لازم برای پردازش هر دوره (epoch) کمتر است. (با colab و تنظیمات Runtime GPU حدود ۷ دقیقه طول کشید).
- ResNet-50 (یادگیری انتقالی): به دلیل پیچیدگی و عمق بسیار بیشتر، زمان آموزش هر دوره به مراتب طولانی‌تر است، حتی زمانی که بخش بزرگی از شبکه فریز شده باشد. (حدود ۴۰ دقیقه).

۲. مقایسه اندازه و پیچیدگی مدل

- CNN با Block Dropout: این یک مدل سفارشی و کوچک است که به طور خاص برای این مسئله طراحی شده است. تعداد پارامترهای آن به مراتب کمتر و معماری آن ساده‌تر است.
- ResNet-50 (یادگیری انتقالی): یک معماری بسیار عمیق (۵۰ لایه) و پیچیده با ده‌ها میلیون پارامتر است. این مدل بسیار بزرگ‌تر و سنگین‌تر است و به منابع محاسباتی بیشتری برای ذخیره‌سازی و اجرا (inference) نیاز دارد.

۳. مزایا و معایب استفاده از یادگیری انتقالی در مقابل آموزش از ابتدا

یادگیری انتقالی (Transfer Learning) – مدل ResNet-50

- مزایا:

- شروع قدرتمند: از ویژگی‌های از پیش آموخته شده روی مجموعه داده عظیم ImageNet بهره می‌برد که می‌تواند به تشخیص ویژگی‌های پایه‌ای کمک کند.
- نیاز به داده کمتر: به طور تئوری، برای رسیدن به دقت خوب به داده‌های کمتری نسبت به آموزش یک شبکه بزرگ از صفر نیاز دارد.

- معایب

- عدم تطابق دامنه: ویژگی‌های آموخته شده از تصاویر عمومی ممکن است برای تشخیص عیوب صنعتی خاص کاملاً بهینه نباشند.
- پیچیدگی و ناپایداری: همان‌طور که در نمودارها مشاهده شد، فرآیند تنظیم دقیق (Fine-Tuning) می‌تواند بسیار حساس و ناپایدار باشد.

- هزینه محاسباتی بالا: مدل‌های بزرگ، سنگین هستند و منابع بیشتری مصرف می‌کنند.

آموزش از ابتدا (Training from Scratch) – مدل CNN با Block Dropout

• مزایا:

- معماری سفارشی: مدل به طور خاص برای پیچیدگی و ویژگی‌های داده‌های موجود طراحی شده و به همین دلیل بهینه‌تر و مؤثرتر عمل کرده است.
- سبک و سریع: اندازه کوچک‌تر مدل منجر به آموزش و اجرای سریع‌تر می‌شود.
- عملکرد برتر در این مسئله: در این مورد خاص، این رویکرد به دلیل تنظیم دقیق معماری و استفاده از روش تنظیم (Regularization) مناسب (Block Dropout) به نتایج بهتری دست یافت.

• معایب:

- نیاز به داده کافی: اگر داده‌ها به اندازه کافی زیاد نباشند، مدل ممکن است نتواند ویژگی‌های معناداری را از ابتدا یاد بگیرد.
- ریسک طراحی نامناسب: طراحی یک معماری کارآمد از صفر نیازمند دانش و تجربه است.

در این مسئله، رویکرد آموزش یک CNN سفارشی از ابتدا همراه با یک روش تنظیم مناسب (Block Dropout) برنده است. این روش توانست مدلی سبک‌تر، سریع‌تر، پایدارتر و در نهایت دقیق‌تر از مدل بسیار بزرگ‌تر و پیچیده‌تر ResNet-50 که از یادگیری انتقالی استفاده می‌کرد، تولید کند. این نشان می‌دهد که همیشه بزرگ‌ترین مدل بهترین گزینه نیست و یک معماری بهینه و متناسب با مسئله می‌تواند نتایج بهتری به همراه داشته باشد.

ویژگی	از ابتدا CNN	ResNet-50 (Pretrained)
تعداد کل پارامترها	بسیار کمتر (مثلاً ۲ میلیون)	بسیار زیاد (بیش از ۲۵ میلیون)
حجم فایل مدل	سبک (ده‌ها مگابایت)	سنگین (صدها مگابایت)
نیاز به GPU و رم بالا	کم	بله، نیاز به حافظه و زمان بیشتر برای پردازش