تمرین ۳ شبکه های عصبی – مازندرانیان - ۸۳۰۴۰۲۰۶۶

سوال اول:

هر ساله شبکه هایی با ساختار جدید روی دادگان imagenet معرفی میشود که بهترین نتایج روز را بهبود می دهد. از جمله آنها شبکه های Vision Transformers (ViT), Swin Transformers, EfficientNet هستند. ساختار این شبکه ها و ویژگی های آن ها را توضیح دهید.

ViT (Vision Transformer):

- **ایده اصلی ViT :** به جای استفاده از عملیات هم‌لول (convolution) که در شبکه‌های عصبی معمولی برای پردازش تصویر استفاده می‌شود، از مکانیزم توجه (attention) الهام گرفته از ترانسفورمرها استفاده می‌کند.
- **نحوه کار :** تصاویر به پچ‌های کوچک تقسیم می‌شوند و هر پچ به یک توکن تبدیل می‌شود. سپس این توکن‌ها به یک ترانسفورمر داده می‌شوند تا روابط بین آن‌ها را مدل‌سازی کند.
- **مزایا ViT :** توانایی مدل‌سازی روابط پیچیده بین قسمت‌های مختلف تصویر را دارد و در بسیاری از وظایف پردازش تصویر نتایج بسیار خوبی کسب کرده است.

Swin Transformer:

- **توسعه بر اساس ViT:** Swin Transformer یک توسعه بر روی ViT است که برای بهبود کارایی و تطبیق‌پذیری آن طراحی شده است.
- **ویژگی‌ها :**
  - **سلسله مراتبی Swin Transformer :**از یک ساختار سلسله مراتبی استفاده می‌کند که به آن اجازه می‌دهد ویژگی‌های در مقیاس‌های مختلف را مدل‌سازی کند.
  - **توجه محلی Swin Transformer :**از مکانیزم توجه محلی استفاده می‌کند که به آن اجازه می‌دهد بر روی نواحی کوچکتر از تصویر تمرکز کند و محاسبات را کاهش دهد.
- **مزایا Swin Transformer :**در مقایسه با ViT کارآمدتر است و در وظایف مختلف پردازش تصویر، مانند تشخیص شیء و تقسیم‌بندی تصویر، نتایج بسیار خوبی کسب کرده است.

EfficientNet:

- **هدف EfficientNet :**با هدف طراحی شبکه‌های عصبی کارآمد و با عملکرد بالا توسعه یافته است.
- **روش EfficientNet :**با استفاده از یک روش جستجوی معماری، بهترین ترکیب از عرض، عمق و رزولوشن را برای شبکه پیدا می‌کند.

- **مزایا EfficientNet :**در مقایسه با سایر شبکه‌های عصبی، با پارامترهای کمتر و محاسبات کمتر، عملکرد بسیار بهتری دارد.

تفاوت‌های کلیدی:

- **ViT:** بر پایه توجه جهانی، مناسب برای داده‌های بزرگ و پیچیده.
- **Swin Transformer:** ترکیبی از توجه محلی و جهانی، کارآمدتر و تطبیق‌پذیرتر.
- **EfficientNet:** تمرکز بر کارایی و عملکرد بالا، با استفاده از جستجوی معماری.

سوال دوم:

یکی از روش های تخمین عدم قطعیت تصمیم شبکه های عمیق استفاده از dropout در فاز تست است. روش ارائه شده در مقاله زیر را برای تخمین نایقینی تصمیم شبکه به طور کلی بیان کنید.

Dropout Injection at Test Time for Post Hoc Uncertainty Quantification in Neural Networks

در شبکه‌های عصبی عمیق، به ویژه در کاربردهایی که تصمیم‌گیری بر اساس آن‌ها حیاتی است، تخمین عدم قطعیت در پیش‌بینی‌ها بسیار مهم است. یکی از روش‌های رایج برای این کار، استفاده از تکنیک Dropout است Dropout .به طور معمول در فاز آموزش برای جلوگیری از بیش‌برازسازی استفاده می‌شود، اما در این مقاله، نویسندگان پیشنهاد می‌کنند که می‌توان از Dropout در فاز تست نیز برای تخمین عدم قطعیت استفاده کرد.

ایده اصلی:

در این روش، در فاز تست، به جای استفاده از یک مدل ثابت (که همه نورون‌ها فعال هستند)، مدل چندین بار با پیکربندی‌های مختلف Dropoutاجرا می‌شود. هر بار، برخی از نورون‌ها به طور تصادفی حذف می‌شوند و پیش‌بینی متفاوتی تولید می‌شود. با تکرار این فرآیند چندین بار، می‌توانیم یک توزیع از پیش‌بینی‌ها ایجاد کنیم. پراکندگی این توزیع می‌تواند به عنوان یک معیار از عدم قطعیت در پیش‌بینی در نظر گرفته شود.

روش کار:

۱. **آموزش مدل :**مدل شبکه عصبی با استفاده از تکنیک Dropout معمولی آموزش داده می‌شود.

۲. **تست با Dropout:** در فاز تست، برای هر نمونه ورودی، مدل چندین بار اجرا می‌شود. در هر اجرا، یک ماسک Dropout به طور تصادفی ایجاد می‌شود و نورون‌هایی که در این ماسک قرار دارند، از شبکه حذف می‌شوند.

۳. **محاسبه توزیع پیش‌بینی‌ها :** برای هر نمونه ورودی، خروجی‌های مختلفی از مدل به دست می‌آید. این خروجی‌ها یک توزیع احتمال را تشکیل می‌دهند.

٤. **تخمین عدم قطعیت :** پراکندگی این توزیع، به عنوان مثال، واریانس یا آنتروپی، می‌تواند به عنوان یک معیار از عدم قطعیت در پیش‌بینی در نظر گرفته شود. هرچه پراکندگی بیشتر باشد، عدم قطعیت نیز بیشتر است.

مزایای این روش:

- **سادگی :** این روش به سادگی به مدل‌های موجود اضافه می‌شود و نیاز به تغییرات عمده در معماری شبکه ندارد.
- **کارایی :** این روش نسبتاً سریع است و می‌تواند به صورت آنلاین برای تخمین عدم قطعیت استفاده شود.
- **تفسیرپذیری :** پراکندگی توزیع پیش‌بینی‌ها به طور مستقیم با مفهوم عدم قطعیت مرتبط است و تفسیر آن آسان است.

محدودیت‌ها:

- **وابستگی به نرخ Dropout:** انتخاب نرخ Dropout مناسب برای تخمین عدم قطعیت مهم است و ممکن است نیاز به تنظیم دقیق داشته باشد.
- **تخمین عدم قطعیت اپستیمی :** این روش عمدتاً برای تخمین عدم قطعیت اپستیمی (ناشی از عدم قطعیت در پارامترهای مدل) مناسب است و ممکن است برای تخمین عدم قطعیت الپستیمی (ناشی از نویز در داده‌ها) کافی نباشد.

جمع‌بندی:

استفاده از Dropout در فاز تست یک روش ساده و موثر برای تخمین عدم قطعیت در شبکه‌های عصبی عمیق است. این روش به ما کمک می‌کند تا به تصمیمات شبکه اعتماد بیشتری داشته باشیم و در کاربردهایی که خطای پیش‌بینی می‌تواند عواقب جدی داشته باشد، از آن استفاده کنیم.

سوال سوم:

نشان دهید اگر ورودی شبکه عصبی پیچشی (CNN)، یک تصویر I با ابعاد $H\times W$ باشد، آنگاه با اعمال فیلتر پیچشی (Convolution) $k\times k$ ، ابعاد تصویر خروجی $H'\times W'$ به صورت زیر بدست می‌آیند:

$$\begin{cases} 1+\dfrac{W-k+2p}{s}=W' \\ 1+\dfrac{H-k+2p}{s}=H' \end{cases}$$

که در آن، p نماد $padding$ (لایه‌گذاری) و s نماد $stride$ (گام) است.

1. Input matrix: $\text{Input} = \begin{bmatrix} 1 & 2 & 3 & 0 \\ 4 & 5 & 6 & 1 \\ 7 & 8 & 9 & 2 \\ 1 & 2 & 3 & 4 \end{bmatrix}$ (4x4 matrix)
2. Kernel (filter): $\text{Kernel} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ (2x2 matrix)
3. Stride: $\text{Stride} = 2$
4. Padding: $\text{Padding} = 1$

$$\text{Padded Input} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 \\ 0 & 4 & 5 & 6 & 1 \\ 0 & 7 & 8 & 9 & 2 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix}$$

$$\text{Output} = \begin{bmatrix} -1 & -3 & 0 \\ -4 & -4 & -4 \\ -7 & -4 & -4 \end{bmatrix}$$

Step-by-step Calculation:

1. Position (0, 0): Sub-matrix:

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Convolution:

$$(0 \cdot 1) + (0 \cdot 0) + (0 \cdot 0) + (1 \cdot -1) = -1$$

2. Position (0, 1): Sub-matrix:

$$\begin{bmatrix} 0 & 0 \\ 2 & 3 \end{bmatrix}$$

Convolution:

$$(0 \cdot 1) + (0 \cdot 0) + (2 \cdot 0) + (3 \cdot -1) = -3$$

3. Position (0, 2): Sub-matrix:

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Convolution:

$$(0 \cdot 1) + (0 \cdot 0) + (0 \cdot 0) + (0 \cdot -1) = 0$$

4. Position (1, 0): Sub-matrix:

$$\begin{bmatrix} 0 & 1 \\ 0 & 4 \end{bmatrix}$$

Convolution:

$$(0 \cdot 1) + (1 \cdot 0) + (0 \cdot 0) + (4 \cdot -1) = -4$$

5. Position (1, 1): Sub-matrix:

$$\begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$$

Convolution:

$$(1 \cdot 1) + (2 \cdot 0) + (4 \cdot 0) + (5 \cdot -1) = -4$$

6. Position (1, 2): Sub-matrix:

$$\begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix}$$

Convolution:

$$(2 \cdot 1) + (3 \cdot 0) + (5 \cdot 0) + (6 \cdot -1) = -4$$

7. Position (2, 0): Sub-matrix:

$$\begin{bmatrix} 0 & 4 \\ 0 & 7 \end{bmatrix}$$

Convolution:

$$(0 \cdot 1) + (4 \cdot 0) + (0 \cdot 0) + (7 \cdot -1) = -7$$

8. Position (2, 1): Sub-matrix:

$$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$$

Convolution:

$$(4 \cdot 1) + (5 \cdot 0) + (7 \cdot 0) + (8 \cdot -1) = -4$$

9. Position (2, 2): Sub-matrix:

$$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$$

Convolution:

$$(5 \cdot 1) + (6 \cdot 0) + (8 \cdot 0) + (9 \cdot -1) = -4$$

Step 2: Determine Output Dimensions

The output dimensions are calculated using the formula:

$$\text{Output size} = \left\lfloor \frac{\text{Input size} + 2 \times \text{Padding} - \text{Kernel size}}{\text{Stride}} \right\rfloor + 1$$

For this example:

$$\text{Output height} = \text{Output width} = \left\lfloor \frac{4 + 2 \times 1 - 2}{2} \right\rfloor + 1 = 3$$

The output will be a 3×3 matrix.

سوال چهارم:

به سوالات مفهومی زیر پاسخ دهید:

الف) در یک شبکه عصبی پیچشی (CNN) چه تفاوت‌هایی بین لایه‌های پیچش (Convolutional Layers) و لایه‌های تماما متصل (Fully Connected Layers) وجود دارد؟

ب) در شبکه عصبی پیچشی (CNN) با استفاده از روش پس‌انتشار (Back-Propagation) فرمول اصلاح وزن لایه ی کانولوشنی را بدست آورید.

ج) چگونه فیلترها یا هسته‌ها در لایه‌های پیچش (Convolutional Layers) به شناسایی ویژگی‌های مختلف تصویر کمک می‌کنند؟

د) چه معایب و محدودیت‌هایی در شبکه‌های عصبی پیچشی (CNN) وجود داشت که باعث شد مدل‌هایی مانند R-CNN و Fast R-CNN توسعه یابند؟

ه) در شبکه‌های عصبی باقی‌مانده (ResNet)، مفهوم اتصالات باقی‌مانده (Residual Connections) چیست؟ چگونه باعث بهبود آموزش شبکه‌های عمیق می‌شود؟

و) فرض کنید مجبور هستیم در یک مساله با تعداد اندک داده از شبکه عمیق استفاده کنیم. چه روش های regularization ی را برای حوزه ورودی، شبکه و خروجی پیشنهاد می کنید.

الف) لایه‌های پیچشی با استفاده از فیلترهایی کوچک، ویژگی‌های محلی تصاویر را استخراج می‌کنند. این فیلترها روی تصویر حرکت کرده و با بخش کوچکی از آن ضرب نقطه‌ای می‌شوند. این عمل باعث می‌شود که شبکه بتواند ویژگی‌هایی مانند لبه‌ها، گوشه‌ها و بافت‌ها را در تصویر تشخیص دهد. یکی از ویژگی‌های مهم لایه‌های پیچشی، اشتراک‌گذاری وزن‌ها است که باعث کاهش تعداد پارامترهای شبکه و در نتیجه کاهش احتمال بیش‌برازسازی می‌شود. از سوی دیگر، لایه‌های تماماً متصل به تمام نورون‌های لایه قبلی متصل هستند. این لایه‌ها وظیفه ترکیب ویژگی‌های استخراج شده توسط لایه‌های پیچشی و تولید خروجی نهایی را بر عهده دارند. برای مثال، در یک شبکه تشخیص اشیاء، لایه تماماً متصل می‌تواند احتمال تعلق یک تصویر به هر کلاس را محاسبه کند.

ب)

فرمول به‌روزرسانی وزن:

$$W = W - \eta * dL/dW$$

برای محاسبه dL/dW (گرادیان خطا نسبت به وزن‌ها)، از قاعده زنجیره‌ای در حساب دیفرانسیل استفاده می‌کنیم:

$$dL/dW = dL/dY * dY/dW$$

dL/dY: همانطور که گفته شد، این گرادیان از لایه بعدی به لایه کانولوشنی منتقل می‌شود .

dY/dW: گرادیان خروجی لایه نسبت به وزن‌ها است و با استفاده از عملیات کانولوشن روی گرادیان خطا محاسبه می‌شود.

ج) فیلترها یا هسته‌ها در لایه‌های پیچشی نقش بسیار مهمی در شناسایی ویژگی‌های مختلف تصاویر دارند. به عبارت ساده‌تر، این فیلترها مانند الگوهایی هستند که در تصویر جستجو می‌شوند. وقتی یک فیلتر با بخشی از تصویر همخوانی پیدا می‌کند، به معنای آن است که آن بخش از تصویر حاوی ویژگی خاصی است که فیلتر برای تشخیص آن طراحی شده است.

نحوه عملکرد:

۱. **حرکت فیلتر روی تصویر :**فیلتر به صورت اسلایدی روی تصویر حرکت می‌کند و در هر موقعیت، با ناحیه‌ای از تصویر که به اندازه خود فیلتر است، ضرب نقطه‌ای می‌شود.

۲. **محاسبه خروجی :**نتیجه ضرب نقطه‌ای به عنوان یک عدد در خروجی قرار می‌گیرد که نشان‌دهنده میزان شباهت بین فیلتر و آن ناحیه از تصویر است.

۳. **تشکیل feature map:** با تکرار این عمل برای تمام موقعیت‌های ممکن فیلتر روی تصویر، یک feature map تولید می‌شود که هر پیکسل آن نشان‌دهنده وجود یک ویژگی خاص در آن ناحیه از تصویر است.

انواع ویژگی‌های قابل تشخیص توسط فیلترها:

- **لبه‌ها :**فیلترهایی با گرادیان بالا در یک جهت خاص، لبه‌های عمودی یا افقی را تشخیص می‌دهند.
- **گوشه‌ها :**فیلترهایی با گرادیان بالا در دو جهت عمود بر هم، گوشه‌ها را تشخیص می‌دهند.
- **تکرار الگوها :**فیلترهایی با الگوهای خاص، تکرار الگوها در تصویر را شناسایی می‌کنند.
- **رنگ‌ها :**فیلترهایی که حساسیت بالایی به کانال‌های رنگی دارند، تغییرات رنگ در تصویر را تشخیص می‌دهند.

- **تکسچر :**فیلترهایی با الگوهای پیچیده، تکسچرهای مختلف را تشخیص می‌دهند.

اهمیت استفاده از چندین فیلتر:

- هر فیلتر یک ویژگی خاص را تشخیص می‌دهد.
- با استفاده از چندین فیلتر، می‌توان ویژگی‌های مختلفی را از تصویر استخراج کرد.
- لایه‌های بعدی شبکه، این ویژگی‌ها را ترکیب کرده و ویژگی‌های سطح بالاتر را ایجاد می‌کنند.

د)

معایب و محدودیت‌های شبکه‌های عصبی پیچشی (CNN) و ضرورت توسعه مدل‌هایی مانند Fast R-CNN وR-CNN

شبکه‌های عصبی پیچشی (CNN) به عنوان ابزاری قدرتمند در حوزه پردازش تصویر شناخته می‌شوند، اما این شبکه‌ها نیز مانند هر مدل دیگری، با محدودیت‌هایی همراه هستند که محققان را بر آن داشت تا مدل‌های کارآمدتری را توسعه دهند.

محدودیت‌های اصلی CNN های اولیه عبارتند از:

- **کُند بودن در تشخیص اشیاء:** CNN های اولیه برای تشخیص اشیاء در تصاویر، نیاز به پردازش تصویر کامل داشتند و این امر باعث کندی آن‌ها می‌شد، خصوصاً در تصاویر با ابعاد بالا.
- **مشکل در تشخیص اشیاء کوچک و متعدد :** این مدل‌ها در تشخیص اشیاء کوچک و متعدد در یک تصویر، عملکرد خوبی نداشتند.
- **عدم توانایی در تشخیص اشیاء با نسبت ابعاد مختلف:** CNN های اولیه در تشخیص اشیایی که نسبت ابعاد متفاوتی داشتند، با مشکل مواجه بودند.

توسعه مدل‌هایی مانند Fast R-CNN و R-CNN :

برای رفع این محدودیت‌ها، مدل‌هایی مانند R-CNN و Fast R-CNN توسعه داده شدند. این مدل‌ها بر اساس ایده ترکیب تشخیص اشیاء با طبقه‌بندی تصاویر عمل می‌کنند. در این مدل‌ها، ابتدا مناطق احتمالی وجود اشیاء در تصویر شناسایی شده و سپس این مناطق به یک شبکه عصبی پیچشی داده می‌شوند تا طبقه‌بندی شوند.

مزایای مدل‌های R-CNN و:Fast R-CNN

- **سرعت بالاتر :** این مدل‌ها با تمرکز بر مناطق خاصی از تصویر، سرعت تشخیص اشیاء را به طور قابل توجهی افزایش می‌دهند.
- **دقت بالاتر :** با استفاده از مکانیزم‌های پیشنهادی منطقه، این مدل‌ها قادر به تشخیص اشیاء کوچک و متعدد با دقت بیشتری هستند.
- **انعطاف‌پذیری بیشتر :**این مدل‌ها می‌توانند اشیاء با نسبت ابعاد مختلف را تشخیص دهند.

مقایسه R-CNN و:Fast R-CNN

- **R-CNN:** در این مدل، از الگوریتم انتخاب منطقه (Selective Search) برای تولید مناطق پیشنهادی استفاده می‌شود. سپس هر یک از این مناطق به یک شبکه عصبی پیچشی داده می‌شود تا طبقه‌بندی شود. این روش اگرچه دقت بالایی دارد اما بسیار کند است.
- **Fast R-CNN:** این مدل با بهبود معماریR-CNN ، سرعت پردازش را به طور قابل توجهی افزایش داده است. درFast R-CNN ، از یک شبکه عصبی پیچشی برای استخراج ویژگی‌های کل تصویر استفاده می‌شود و سپس از این ویژگی‌ها برای طبقه‌بندی مناطق پیشنهادی استفاده می‌شود.

در نهایت:

مدل‌های R-CNN و Fast R-CNN نشان دادند که با ترکیب ایده‌های مختلف می‌توان بر محدودیت‌های CNN های اولیه غلبه کرد و سیستم‌های تشخیص اشیاء کارآمدتری را توسعه داد. این مدل‌ها پایه و اساس بسیاری از مدل‌های تشخیص اشیاء مدرن مانندFaster R-CNN، Mask R-CNNو YOLO قرار گرفته‌اند.

موارد دیگری که می‌توان به آن اشاره کرد:

- **محدودیت‌های محاسباتی :**آموزش و اجرای CNN های عمیق نیاز به سخت‌افزار قدرتمندی دارد.
- **مشکل بیش‌برازسازی :**مدل‌های پیچیده CNN ممکن است به داده‌های آموزشی بیش از حد برازش شوند و در روی داده‌های تست عملکرد خوبی نداشته باشند.
- **عدم تفسیرپذیری :**تصمیم‌گیری‌های یک شبکه عصبی پیچشی معمولاً برای انسان قابل درک نیست.

ه) **اتصالات باقیمانده (Residual Connections)** یکی از نوآوری‌های اصلی در معماری شبکه‌های عصبی باقی‌مانده (ResNet) است. این اتصالات، به صورت مستقیم خروجی لایه‌های ابتدایی را به لایه‌های بعدی مرتبط می‌کنند و به شبکه اجازه می‌دهند تا اطلاعات را به صورت مستقیم انتقال دهد.

چرا اتصالات باقیمانده اهمیت دارند؟

مشکل اصلی در شبکه‌های عصبی عمیق، پدیده ناپدید شدن گرادیان است. با افزایش عمق شبکه، گرادیان‌ها در حین انتشار به عقب به شدت کاهش می‌یابند و در نتیجه، لایه‌های اولیه شبکه به خوبی آموزش نمی‌بینند. اتصالات باقیمانده این مشکل را با ایجاد مسیرهای کوتاه‌تری برای انتشار گرادیان‌ها برطرف می‌کنند.

چگونه اتصالات باقیمانده کار می‌کنند؟

در یک بلوک باقی‌مانده، خروجی لایه‌های قبلی به صورت مستقیم به ورودی لایه بعدی اضافه می‌شود. به عبارت دیگر، خروجی یک بلوک باقی‌مانده برابر است با:

خروجی بلوک = تابع فعال‌سازی (خروجی لایه‌های داخلی + ورودی بلوک)

این بدان معناست که شبکه می‌تواند به سادگی ورودی بلوک را به عنوان خروجی انتخاب کند، بدون اینکه مجبور باشد ویژگی‌های جدیدی را بیاموزد. این امر به شبکه اجازه می‌دهد تا به راحتی بر روی یادگیری ویژگی‌های پیچیده‌تر تمرکز کند.

مزایای اتصالات باقیمانده:

- **تسهیل آموزش شبکه‌های عمیق :**با ایجاد مسیرهای کوتاه برای انتشار گرادیان‌ها، اتصالات باقیمانده به شبکه اجازه می‌دهند تا به راحتی آموزش ببیند و از مشکل ناپدید شدن گرادیان جلوگیری کند.
- **بهبود عملکرد :** شبکه‌های ResNet معمولاً دقت بالاتری نسبت به شبکه‌های عصبی عمیق سنتی دارند، به خصوص در مسائل پیچیده مانند تشخیص تصویر با مجموعه داده‌های بزرگ.
- **سادگی پیاده‌سازی :**اضافه کردن اتصالات باقیمانده به یک معماری شبکه عصبی موجود بسیار ساده است.

و) **روش‌های منظم‌سازی در شبکه‌های عمیق با داده‌های اندک**

هنگامی که با تعداد داده‌های آموزشی محدودی روبه‌رو هستیم، استفاده از روش‌های منظم‌سازی برای جلوگیری از بیش‌برازسازی (Overfitting)و بهبود تعمیم‌پذیری مدل از اهمیت بالایی برخوردار است. در ادامه، برخی از روش‌های منظم‌سازی موثر برای حوزه‌های ورودی، شبکه و خروجی در شبکه‌های عمیق معرفی می‌شوند:

منظم‌سازی در حوزه ورودی:

- **افزودن نویز به داده‌ها :**با افزودن نویز به داده‌های ورودی، شبکه مجبور می‌شود ویژگی‌های اصلی داده‌ها را یاد بگیرد و به نویز حساس نباشد.
- **آگمنتاسیون داده‌ها :**با اعمال تغییراتی کوچک مانند چرخش، برش، تغییر اندازه و تغییر روشنایی بر روی تصاویر، تعداد داده‌های آموزشی را افزایش داده و تنوع آن‌ها را بهبود می‌بخشیم.

منظم‌سازی در حوزه شبکه:

- **L1 و L2 Regularization:** با افزودن جریمه‌هایی به تابع هزینه، از پیچیدگی بیش از حد مدل جلوگیری می‌کنیم L1 .باعث ایجاد وزن‌های پراکنده‌تر و L2 باعث کاهش مقدار مطلق وزن‌ها می‌شود.
- **Dropout:** با غیرفعال کردن تصادفی برخی از نورون‌ها در طول آموزش، از همبستگی بیش از حد بین نورون‌ها جلوگیری کرده و مدل را قوی‌تر می‌کنیم.
- **Early Stopping:** آموزش مدل را زمانی متوقف می‌کنیم که خطای اعتبارسنجی شروع به افزایش کند تا از بیش‌برازسازی جلوگیری کنیم.
- **BatchNormlization:** با نرمالایز کردن ورودی هر لایه، آموزش را تسریع کرده و به تعمیم‌پذیری بهتر مدل کمک می‌کند.

منظم‌سازی در حوزه خروجی:

- **:Label Smoothing**با نرم کردن برچسب‌ها، از اطمینان بیش از حد مدل به یک برچسب خاص جلوگیری می‌کنیم و به مدل اجازه می‌دهیم تا برچسب‌های متعددی را در نظر بگیرد.

سایر روش‌ها:

- **استفاده از مدل‌های از پیش آموزش دیده :(Transfer Learning)** با استفاده از مدل‌های از پیش آموزش دیده روی داده‌های بزرگ، می‌توانیم از دانش موجود استفاده کرده و مدل خود را با داده‌های محدود آموزش دهیم.
- **کاهش ابعاد داده‌ها :**با استفاده از روش‌هایی مانند PCA یا t-SNE می‌توانیم ابعاد داده‌ها را کاهش داده و از پیچیدگی مدل بکاهیم.
- **طراحی معماری‌های ساده‌تر :**با استفاده از شبکه‌های ساده‌تر با تعداد پارامتر کمتر، می‌توانیم از بیش‌برازسازی جلوگیری کنیم.

انتخاب روش مناسب منظم‌سازی به عوامل مختلفی بستگی دارد، از جمله:

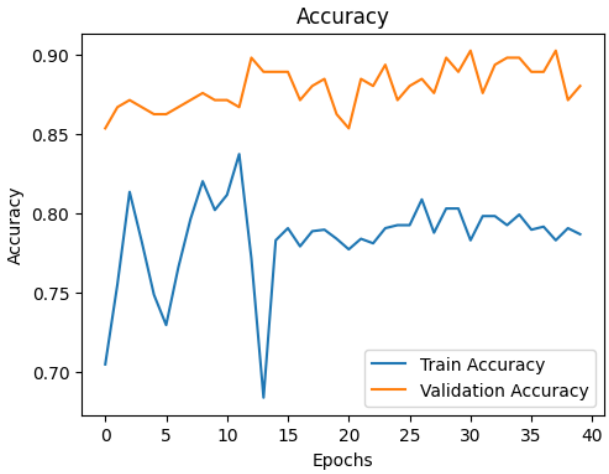
- **نوع داده‌ها:** نوع داده‌ها (تصویر، متن، صوت) و ویژگی‌های آن‌ها در انتخاب روش منظم‌سازی موثر است.
- **معماری شبکه:** معماری شبکه و تعداد لایه‌ها و پارامترها بر انتخاب روش‌های منظم‌سازی تاثیرگذار است.
- **حجم داده‌ها:** هرچه حجم داده‌ها کمتر باشد، نیاز به استفاده از روش‌های منظم‌سازی قوی‌تر احساس می‌شود.

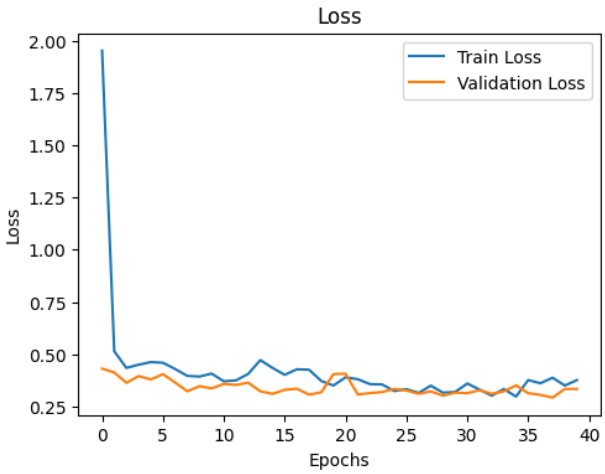
سوال پنجم:

استفاده از شبکه‌های عصبی باقی‌مانده ResNet، به عنوان یک مدل از پیش‌آموزش‌داده‌شده (Pre-Trained)، برای طبقه‌بندی تصاویر در دیتاست COVID-19 یک روش کارآمد است. هدف این است که از قابلیت انتقال یادگیری (Transfer Learning) استفاده کرده و لایه‌های مدل را منجمد (Freeze) کنید تا تنها لایه‌های انتهایی آموزش داده شوند.

دیتاست:

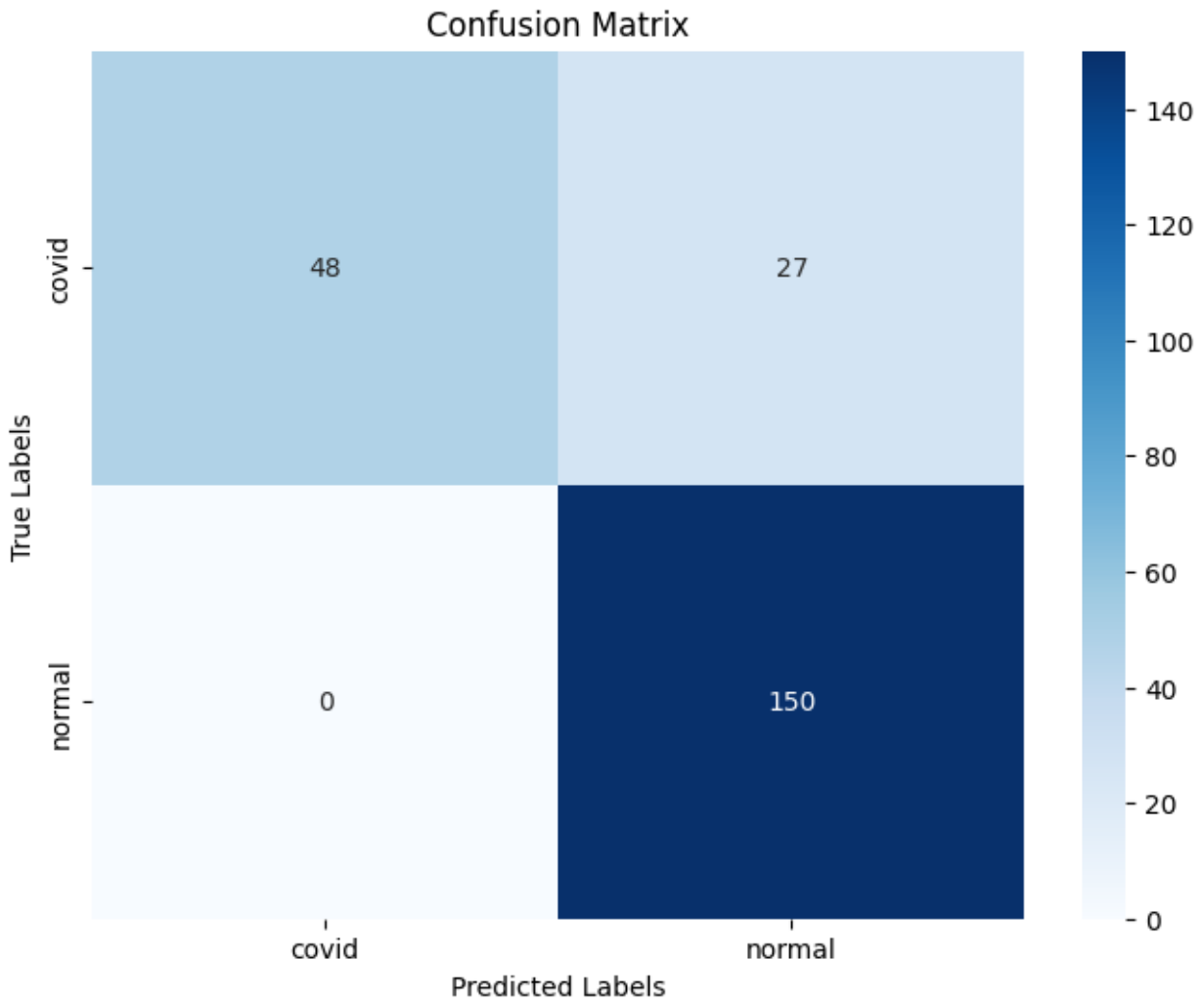
از دیتاست موجود در COVID-19 X-Ray and CT-Scan Image Dataset استفاده کنید. این دیتاست شامل تصاویر X-Ray و CT-Scan از بیماران با شرایط مختلف است که به کلاس‌های COVID-19 و Normal (یا سایر دسته‌ها) تقسیم می‌شود.





Classification Report:

	precision	recall	f1-score	support
covid	1.00	0.64	0.78	75
normal	0.85	1.00	0.92	150
accuracy			0.88	225
macro avg	0.92	0.82	0.85	225
weighted avg	0.90	0.88	0.87	225



منجمد کردن مدل (freezing) به معنای ثابت نگه‌داشتن وزن‌های لایه‌های مشخصی از مدل است تا در طول فرآیند آموزش به‌روزرسانی نشوند. این کار معمولاً زمانی انجام می‌شود که بخواهید از یک مدل از پیش‌آموزش‌دیده به‌عنوان بخشی از یک مدل جدید استفاده کنید. با این روش، ویژگی‌های عمومی که قبلاً آموخته شده‌اند حفظ می‌شوند و فقط بخش‌های جدید مدل آموزش می‌بینند. این کار باعث کاهش زمان آموزش و جلوگیری از بیش‌برازش در لایه‌های اولیه می‌شود، اما ممکن است توانایی مدل در یادگیری جزئیات خاص مسئله محدود شود.

برای بهبود عملکرد مدل، چند راهکار مفید وجود دارد:

۱. **افزایش تعداد دوره‌های آموزش:** این کار به مدل زمان بیشتری برای یادگیری الگوهای داده می‌دهد، اما باید مراقب بیش‌برازش بود.

۲. **تنظیم نرخ یادگیری:** کاهش نرخ یادگیری می‌تواند به مدل کمک کند که به‌تدریج به حداقل‌های بهینه نزدیک شود. استفاده از تکنیک‌هایی مانند کاهش نرخ یادگیری پویا نیز مؤثر است.

۳. **افزایش حجم داده‌ها:** استفاده از داده‌های بیشتر یا تکنیک‌هایی مثل افزایش داده (data augmentation) می‌تواند تنوع ورودی‌ها را افزایش داده و از بیش‌برازش جلوگیری کند.

۴. **استفاده از مدل‌های پیچیده‌تر یا پیش‌آموزش‌دیده:** انتخاب مدل‌های بزرگ‌تر یا انتقال یادگیری از مدل‌های پیش‌آموزش‌دیده می‌تواند نتایج را بهبود بخشد.

۵. **تنظیم معماری مدل:** اضافه کردن لایه‌ها، تغییر توابع فعال‌سازی، یا استفاده از تکنیک‌هایی مانند dropout برای کاهش بیش‌برازش نیز مفید است.

برای حل مشکل عدم توازن کلاس‌ها و سوگیری شبکه به سمت کلاس‌های پرتکرار، می‌توان از راهکارهای زیر استفاده کرد:

۱. **در بخش داده‌ها:** از تکنیک‌های افزایش داده (data augmentation) برای کلاس‌های کم‌نمونه یا کاهش داده برای کلاس‌های پرتکرار استفاده کنید. همچنین می‌توانید از روش‌های oversampling مانند SMOTE یا undersampling بهره ببرید.

۲. **در بخش وزن‌دهی به کلاس‌ها:** در تابع خطای شبکه، وزن‌های متفاوتی برای کلاس‌ها اختصاص دهید تا کلاس‌های کم‌نمونه تأثیر بیشتری در به‌روزرسانی وزن‌ها داشته باشند.

۳. **استفاده از تکنیک‌های یادگیری تطبیقی:** روش‌هایی مانند focal loss که تمرکز بیشتری بر نمونه‌های دشوار یا کم‌نمونه دارند، می‌توانند مفید باشند.

۴. **استفاده از معیارهای ارزیابی مناسب:** به جای دقت کلی(accuracy) ، از معیارهایی مانند F1-score، precision، و recall برای ارزیابی مدل استفاده کنید تا عملکرد در کلاس‌های کم‌نمونه بهتر بررسی شود.

۵. **افزودن داده مصنوعی:** تولید داده‌های مصنوعی برای کلاس‌های کم‌نمونه از طریق مدل‌هایی مثل GAN یا روش‌های دیگر می‌تواند کمک‌کننده باشد.

ResNet50 یک شبکه عصبی عمیق است که برای یادگیری ویژگی‌ها و طبقه‌بندی تصاویر استفاده می‌شود و از ساختار **شبکه‌های باقی‌مانده (Residual Networks)** بهره می‌برد. مشخصات معماری آن عبارت‌اند از:

۱. **تعداد لایه‌ها** ResNet50 :شامل ۵۰ لایه با وزن قابل یادگیری است که شامل کانولوشن، نرمال‌سازی دسته‌ای (Batch Normalization)، و توابع فعال‌سازی ReLU است.

۲. **بلوک‌های باقی‌مانده :(Residual Blocks)** از دو نوع بلوک استفاده می‌کند:

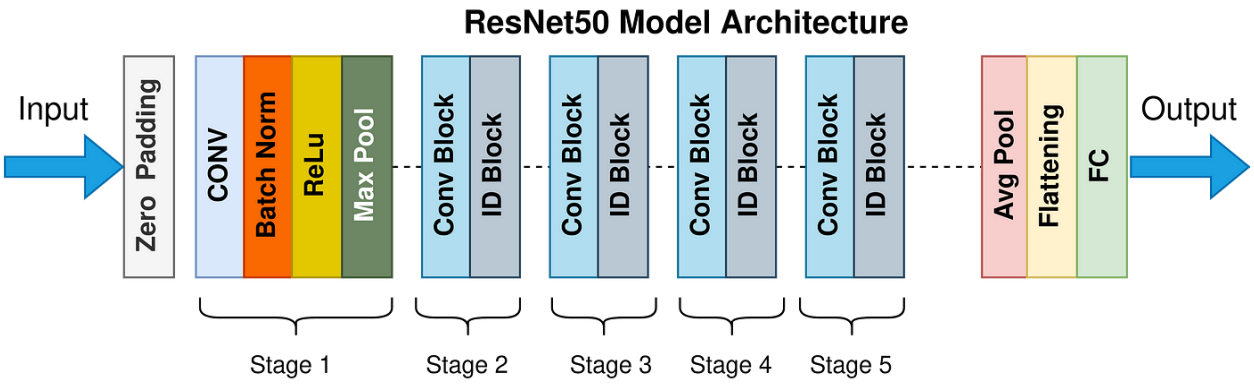
- **بلوک‌های ساده :(Basic Block)** برای معماری‌های کوچکتر.
- **بلوک‌های :Bottleneck** برای ResNet50 و معماری‌های بزرگتر که شامل سه لایه در هر بلوک است.

۳. **ساختار کلی** : شبکه از یک لایه کانولوشن اولیه، به‌دنبال آن ۴ گروه از بلوک‌های Bottleneck با تعداد لایه‌های مختلف (۳، ۴، ۶، ۳)، و در نهایت یک لایه کاملاً متصل (fully connected) تشکیل شده است.

۴. **میان‌برها (Shortcuts):** ارتباطات مستقیم بین ورودی و خروجی هر بلوک باقی‌مانده برای جلوگیری از ناپدید شدن گرادیان.

۵. **ابعاد ورودی و خروجی** : ورودی معمولاً تصاویر 224×224 و خروجی ۱۰۰۰ کلاس برای ImageNet است.

این معماری به دلیل بلوک‌های باقی‌مانده، از ناپدید شدن گرادیان جلوگیری کرده و امکان آموزش مدل‌های عمیق‌تر را فراهم می‌کند.



در مدل پیاده سازی شده در این سوال دقت مدل روی عکس های normal مقداری کم است که با روش های گفته شده میتوان آن را بهبود داد.

سوال ششم: (امتیازی)

تشخیص اشیاء، یک روش مبتنی بر بینایی کامپیوتر در حوزه هوش مصنوعی است که برای تعیین نوع و مکان اشیا در تصاویر یا فیلم ها مورد استفاده قرار می گیرد. از جمله چالش‌های مهم در این حوزه، حضور چند شئ مختلف در تصویر و ضرورت تشخیص مکان دقیق آن‌ها است.

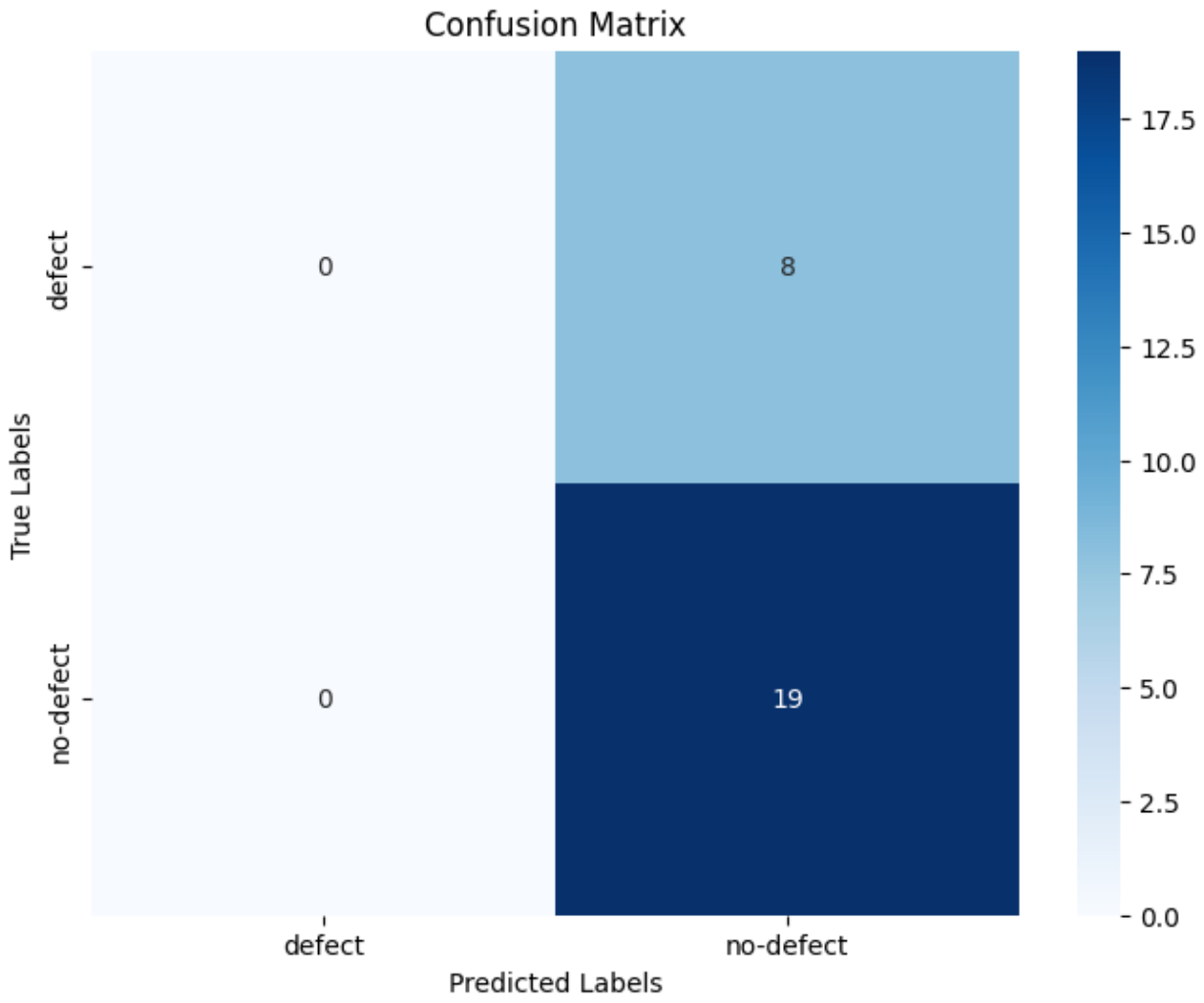
در این سوال بایستی به بررسی تشخیص عیوب جوش مبتنی بر شبکه‌های پیچشی (CNNs) بپردازید و از دو مدل معروف در این حوزه به نام‌های YOLOv3 و Faster-RCNN استفاده کنید.

دیتاست:

از دیتاست موجود در پیوست استفاده کنید. این دیتاست شامل تصاویر X-Ray از جوشکاری با شرایط مختلف است که بایستی عیوب جوشکاری بر روی آنها مشخص گردد.

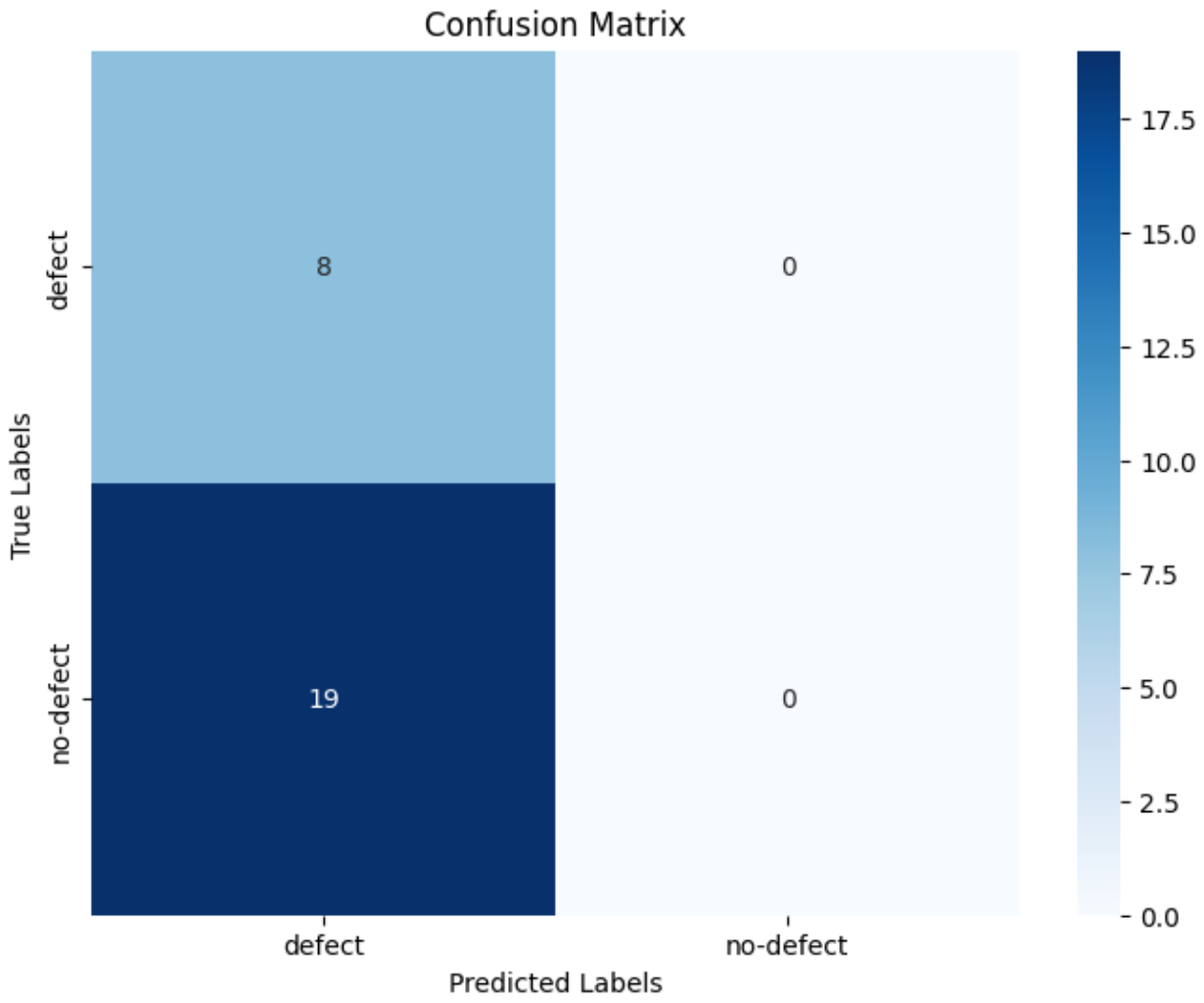
گزارش نهایی باید شامل موارد زیر باشد:

- مشخصات معماری مدل (YOLOv3 و Faster-RCNN).
- تنظیمات هایپرپارامترها (مانند تابع هزینه، بهینه‌ساز، و نرخ یادگیری).
- نمودارهای Loss و Accuracy برای داده‌های آموزش و اعتبارسنجی.
- ارزیابی نهایی شامل معیارهای Accuracy، Precision، Recall، F1-Score و ماتریس آشفتگی.
- تحلیل کامل از عملکرد و نتیجه‌ی نهایی مدل.

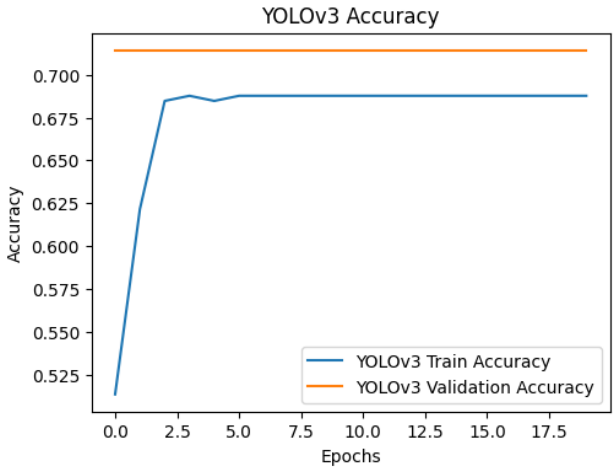


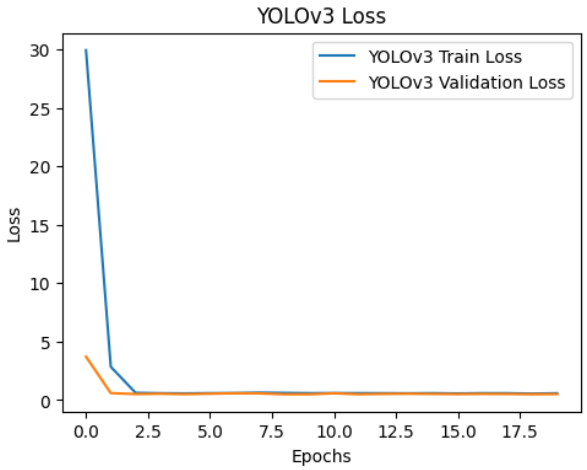
Classification Report:

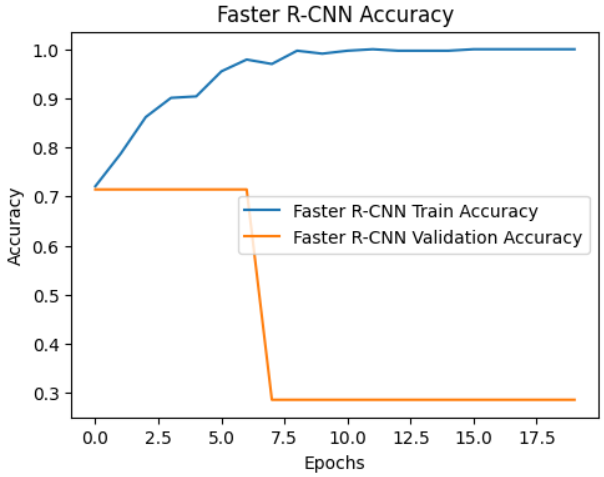
	precision	recall	f1-score	support
defect	0.00	0.00	0.00	8
no-defect	0.70	1.00	0.83	19
accuracy			0.70	27
macro avg	0.35	0.50	0.41	27
weighted avg	0.50	0.70	0.58	27

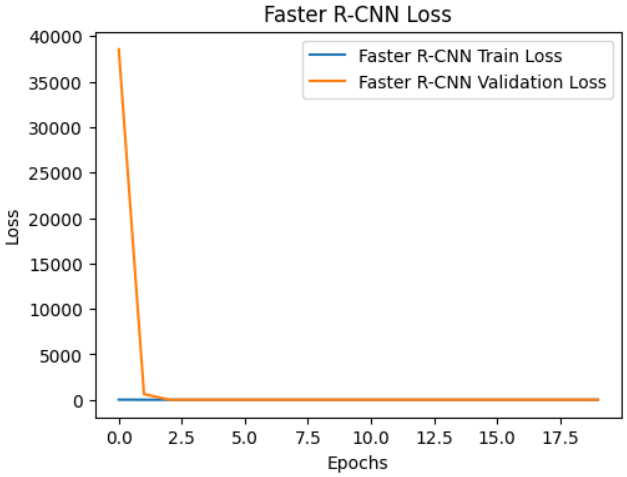



```
Classification Report:
              precision    recall  f1-score   support

      defect       0.30      1.00      0.46         8
   no-defect       0.00      0.00      0.00        19

    accuracy                           0.30        27
   macro avg       0.15      0.50      0.23        27
weighted avg       0.09      0.30      0.14        27
```



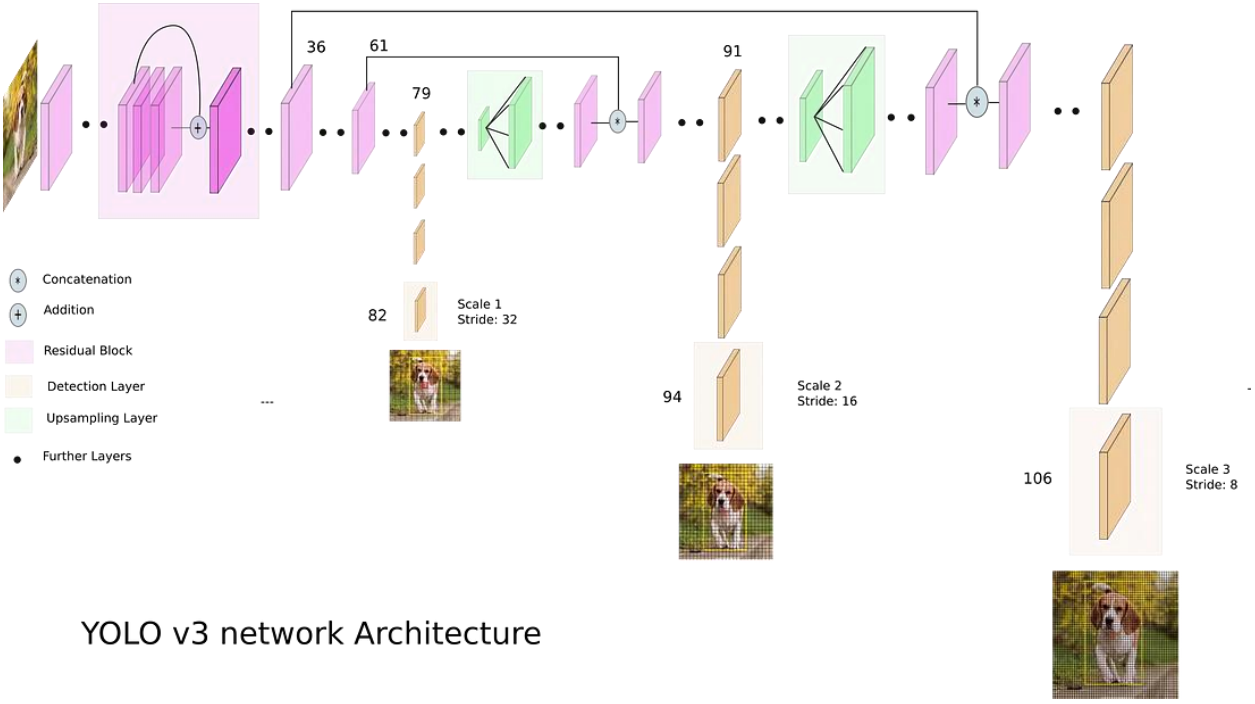




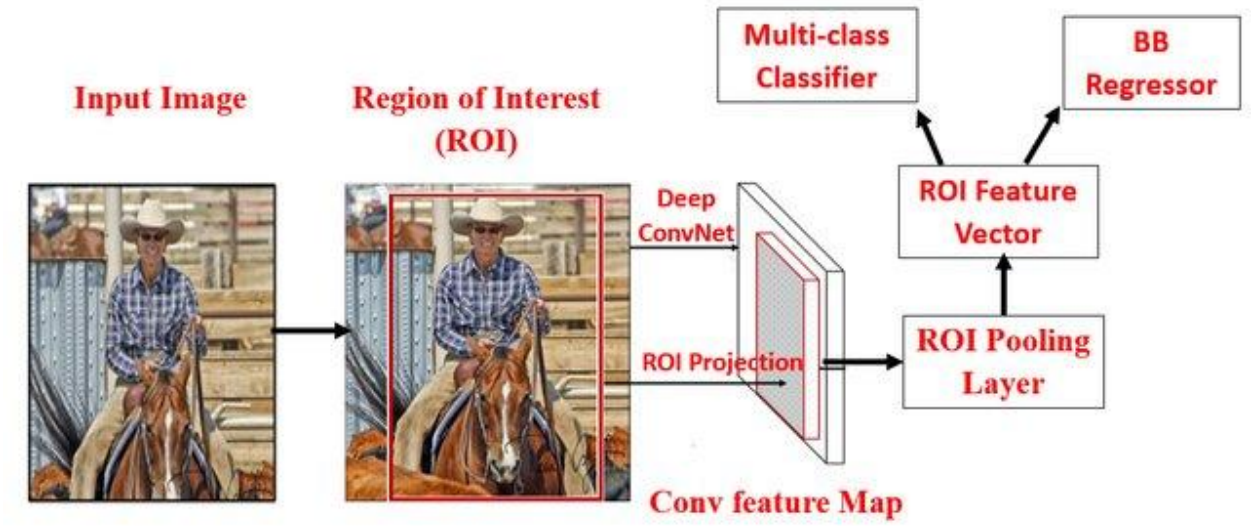


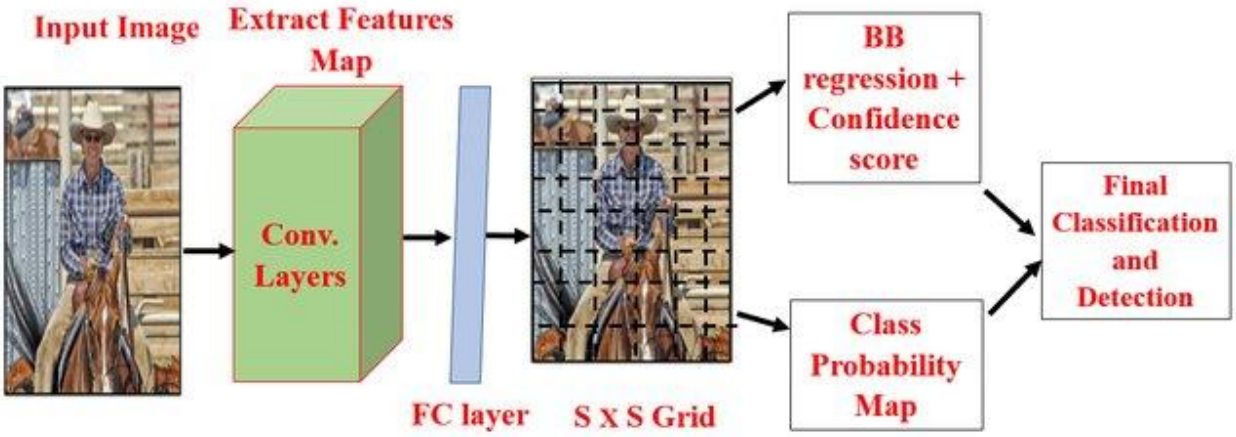

```
yolov3_model = build_yolov3_model((224, 224, 3))
yolov3_model.compile(
    optimizer=Adam(learning_rate=0.001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

faster_rcnn_model = build_faster_rcnn_model((224, 224, 3))
faster_rcnn_model.compile(
    optimizer=Adam(learning_rate=0.001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

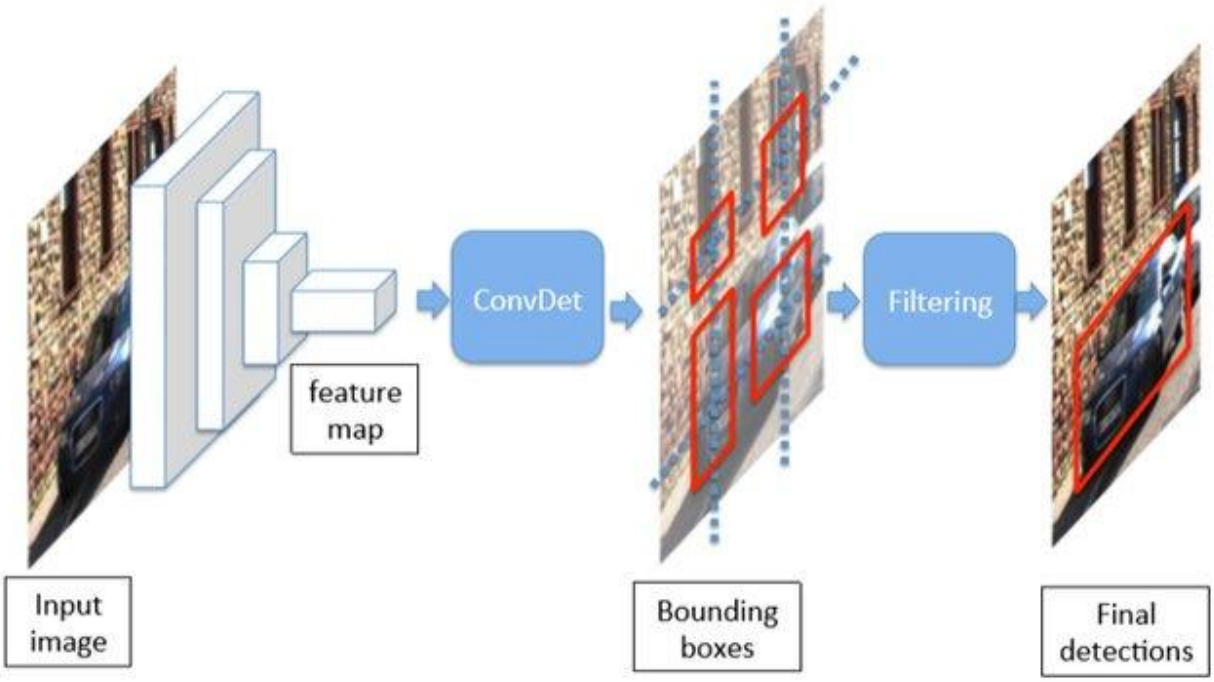
YOLO (You Only Look Once)و RCNN (Region-Based Convolutional Neural Network) دو روش محبوب برای شناسایی اشیاء در تصاویر هستند YOLO .یک مدل end-to-end است که کل تصویر را به یک شبکه عصبی وارد کرده و با تقسیم تصویر به شبکه‌های کوچک، به صورت همزمان مختصات جعبه‌های اطراف اشیاء و برچسب‌های کلاس را پیش‌بینی می‌کند. این روش به دلیل سرعت بالا و کارایی مناسب در کاربردهای بلادرنگ مثل ویدئو بسیار محبوب است. در مقابل، RCNNابتدا با استفاده از روش‌های منطقه‌ای (Selective Search) بخش‌های پیشنهادی تصویر را پیدا کرده و سپس این بخش‌ها را به یک شبکه عصبی برای تشخیص و طبقه‌بندی ارسال می‌کند RCNN .دقت بالایی دارد، اما به دلیل چند مرحله‌ای بودن، نسبت به YOLO کندتر است. نسخه‌های بهبود یافته مانند Fast RCNN و Faster RCNN تلاش کرده‌اند این محدودیت‌های سرعت را کاهش دهند.



YOLO v3 network Architecture




Input Image
Extract Features Map
Conv. Layers
FC layer
S X S Grid
BB regression + Confidence score
Class Probability Map
Final Classification and Detection


feature map
ConvDet
Filtering
Input image
Bounding boxes
Final detections