پاسخ سوال ۱):

آ)

$$\#p = N \times ((k \times k) \times M) + N = NMk^2 + N$$

N : number of filters and each filter has a bias.

چون padding به صورت same است، ابعاد feature map با ابعاد ورودی برابر و برابر است با: $H \times W$

هر خروجی از عملیات کانولوشن فیلتر با تصویر ورودی بدست می‌آید، در نتیجه تعداد $H \times W \times M \times k^2$ ضرب به ازای هر فیلتر داریم؛ پس در کل تعداد عملیات ضرب برابر است با:

$$\#op = N \times [(H \times W) \times M \times k^2]$$

ب)

Input: $3 \times 128 \times 128$
$p = 2, k = 5, s = 2$ for all layers
Layer: 1 2 3
Filters: 64 128 256
Layer_1:

$$w_{output} = \frac{128-5+4}{2} + 1 = 64 \rightarrow Output_{size} = (64 \times 64 \times 64),$$
$$\#p = 64(1 + 5^2 \times 3) = 4864,$$
$$\#op = 64 \times 64 \times 25 \times 3 \times 64 = 19660800$$

Layer_2:

$$w_{output} = \frac{64-5+4}{2} + 1 = 32 \rightarrow Output_{size} = (128 \times 32 \times 32),$$
$$\#p = 128(1 + 5^2 \times 64) = 204928,$$
$$\#op = 32 \times 32 \times 25 \times 128 \times 64 = 209715200$$

Layer_3:

$$w_{output} = \frac{32-5+4}{2} + 1 = 16 \rightarrow Output_{size} = (256 \times 16 \times 16),$$
$$\#p = 256(1 + 5^2 \times 128) = 819456,$$
$$\#op = 256 \times 128 \times 16 \times 16 \times 25 = 209715200$$

برای Receptive Field از فرمول زیر استفاده میکنم:

$$r_0 = \Sigma_{l=1}^{L}(k_l - 1)\prod_{i=1}^{l-1} s_i + 1$$
$$r_0 = (5-1) + 4 \times 2 + 4 \times 4 + 1 = 29$$

این به این معنی است که هر پیکسل از feature map در لایه سوم، به ۲۹ پیکسل از تصویر ورودی نگاه میکند.

ج)

محاسبات را به دو قسمت تقسیم میکنم:

قسمت اول، روی هر کانال ورودی یک فیلتر قرار میدهیم؛ پس تعداد پارامتر ها برابر است با:

$$M + k^2 \times M = M(k^2 + 1)$$

و تعداد عملیات های ضرب موردنیاز برابر است با:

$$H \times W \times k^2 \times M = HWk^2M$$

قسمت دوم، به تعداد کانال خروجی، فیلتر های ۱بعدی قرار میدهیم؛ پس تعداد پارامتر ها برابر است با:

$$N + N \times (1^2 \times M) = N(1 + M)$$

و تعداد عملیات های ضرب موردنیاز برابر است با:

$$H \times W \times N \times M = HWNM$$

پس تعداد کل ضرایب برابر است با:

$$\#p = M(k^2 + 1) + N(1 + M)$$

و تعداد کل عملیات های ضرب برابر است با:

$$\#op = k^2HWM + HWNM$$

مشخصا تعداد پارامتر و تعداد عملیات های ضرب مورد نیاز این قسمت خیلی کمتر از قسمت آ) است.

Input: $3 \times 128 \times 128$
$p = 2, k = 5, s = 2$ for all layers
Layers: 1 2 3
Filters: 64 128 256
Layer_1: $M = 3, N = 64, H = W = 128, k = 5$

$$w_{output} = \frac{128-5+4}{2} + 1 = 64 \rightarrow Output_{size} = (64 \times 64 \times 64),$$
$$\#p = (25 + 1) \times 3 + (3 + 1) \times 64 = 334,$$
$$\#op = 128^2 \times 25 \times 3 + 64 \times 3 \times 128^2 = 3156528$$

Layer_2: $M = 64, N = 128, H = W = 64, k = 5$

$$w_{output} = \frac{64-5+4}{2} + 1 = 32 \rightarrow Output_{size} = (128 \times 32 \times 32),$$
$$\#p = 9984,$$
$$\#op = 40108032$$

Layer_3: $M = 128, N = 256, H = W = 32, k = 5$

$$w_{output} = \frac{32-5+4}{2} + 1 = 16 \rightarrow Output_{size} = (256 \times 16 \times 16),$$
$$\#p = 36352,$$
$$\#op = 36831232$$

د)

تعداد پارامتر های لایه FC به صورت زیر محاسبه میشود:

$$16 \times 16 \times 256 \times 200 + 200 = 13107400$$

نبست پارامتر های FC به بقیه	۱۳۱۰۷٤۰۰	تعداد ضرایب لایه FC
٪ ۹۲.۷	۱۰۲۹۲٤۸	تعداد ضرایب معماری آ
٪ ۹۹.٦	٤٦٦۷۰	تعداد ضرایب معماری ج

برای کاهش تعداد ضرایب در لایه FC، میتوان تعداد feature map های لایه قبل از FC را کاهش داد. برای این کار میتوان قبل از لایه FC از لایه GAP یا لایه GMP برای تبدیل feature map، به یک عدد به ازای هر کانال استفاده کرد. اگر در اینجا این لایه را استفاده کنیم، تعداد پارامتر ها به قرار زیر است:

نبست پارامتر های FC به بقیه	۵۱٤۰۰	تعداد ضرایب لایه FC
٪ ٤.۷۵	۱۰۲۹۲٤۸	تعداد ضرایب معماری آ
٪ ۵۲.٤۱	٤٦٦۷۰	تعداد ضرایب معماری ج

پاسخ سوال ۲)

الف) تفاوت اصلی dense connections در DensNet و residual connections در ResNet:

مفهوم residual connections در Resnet:

در ResNet (Residual Network)، از اتصالات باقی‌مانده (Residual Connections) استفاده می‌شود. ایده این است که خروجی یک لایه به ورودی لایه‌ای در جلوتر اضافه شود. این رویکرد با استفاده از فرمول زیر مدل می‌شود:

$$x + F(x, \{W_i\}) = y$$

در اینجا $F(x, \{w_i\})$ نشان‌دهنده عملیات لایه‌های میانی است (مثل کانولوشن یا غیرخطی‌سازی)، و x ورودی اولیه است. در واقع ResNet تلاش میکند با کاهش پیچیدگی یادگیری شبکه، امکان یادگیری تغییرات کوچک (Residuals) را فراهم کند. در واقع، به‌جای یادگیری کل نگاشت، شبکه فقط تفاوت‌ها (Residuals) را یاد می‌گیرد.

ویژگی های کلیدی ResNet:

- کمک به یادگیری مؤثر در شبکه‌های بسیار عمیق.
- کاهش مشکل ناپدید شدن گرادیان (Vanishing Gradient) از طریق مسیرهای مستقیم برای گرادیان.
- استفاده از Skip Connections برای ترکیب اطلاعات.

مفهوم dense connections در DensNet:

در DenseNe**t** (Densely Connected Network)، ایده اصلی این است که هر لایه با تمام تعدادی(نه لزوما یک لایه) قبلی خود ارتباط دارد. به عبارتی، ویژگی‌های استخراج‌شده از هر لایه به تعدادی (حداکثر۱۶) از لایه‌های بعدی منتقل می‌شود. برخلاف ResNet در این حالت اطلاعات لایه های قبلی به هم متصل میشوند؛ بنابراین باید ابعاد خروجی لایه ها به درستی انتخاب شود. این اتصالات با فرمول زیر توصیف می‌شوند:

$$\mathrm{H_l}\left([\mathrm{x_0}, \mathrm{x_1}, \ldots, \mathrm{x_{l-1}}]\right) = x_l$$

- در اینجا x_l ویژگی‌های خروجی لایه l است، و$[x_0, x_1, \ldots, x_{l-1}]$,نشان‌دهنده اتصال تمام خروجی‌های قبلی است.
- H_l عملیات یک لایه مثل کانولوشن یا تابع فعالسازی است.

ویژگی‌های کلیدی:DenseNet

- هر لایه ویژگی‌های لایه‌های قبلی را به ارث می‌برد و آن‌ها را به ویژگی‌های خود اضافه می‌کند.
- انتقال مستقیم اطلاعات :این ساختار باعث کاهش مشکلات اطلاعات از دست‌رفته و استفاده بهتر از ویژگی‌ها می‌شود.
- DenseNet نیاز به تعداد کمتری از پارامترها دارد، زیرا نیازی به بازآفرینی ویژگی‌ها نیست؛ لایه‌ها ویژگی‌های قبلی را دوباره استفاده می‌کنند. بنابراین علاوه بر اینکه باعث ایجاد مسیر هایی برای گرادیان میشود، میتوان با عمق کمتری نسبت به سایر شبکه ها به دقت مورد نظر دست یافت و به این ترتیب میتواند برای مشکل گرادیان ناپدید شونده موثر باشد.

تفاوت‌های کلیدی		
DenseNet	ResNet	ویژگی
استفاده از اتصالات متراکم	استفاده از اتصال باقی‌مانده	نوع اتصال
انتقال و ترکیب مستقیم تمام ویژگی‌های قبلی	یادگیری تفاوت‌ها	هدف اصلی اتصال
از طریق ترکیب مستقیم ویژگی‌ها	از طریق مسیرهای باقی‌مانده	کاهش مشکل گرادیان ناپدید شونده
کمتر، زیرا ویژگی‌ها بین لایه‌ها به اشتراک گذاشته می‌شوند	بیشتر، زیرا هر لایه ویژگی‌های خاص خود را تولید می‌کند	تعداد پارامترها
اطلاعات از تعدادی از لایه‌های قبلی استفاده می‌شود	اطلاعات فقط از یک لایه قبلی استفاده می‌شود	باز استفاده اطلاعات

مزایا و معایب

ResNet

- **مزایا:**
  - بهبود یادگیری در شبکه‌های بسیار عمیق.
  - طراحی ساده و موثر برای معماری‌های پیچیده.
- **معایب:**
  - استفاده ناکافی از اطلاعات ویژگی‌های لایه‌های قبلی.

DenseNet

- **مزایا:**
  - بازاستفاده مؤثر اطلاعات.
  - نیاز به تعداد کمتری از پارامترها.
  - بهبود انتشار اطلاعات و گرادیان.
- **معایب:**
  - هزینه محاسباتی بالاتر به دلیل ترکیب مداوم ویژگی‌ها.

به طور کلی ResNet و DenseNet هر دو برای رفع مشکلات شبکه‌های عمیق مانند کاهش گرادیان طراحی شده‌اند. ResNet با یادگیری تفاوت‌ها (residuals) و DenseNet با ترکیب ویژگی‌های تعدادی از لایه‌ها، روش‌های متفاوتی برای بهبود یادگیری ارائه می‌دهند.DenseNet به دلیل بهره‌برداری بهتر از اطلاعات، در بسیاری از موارد عملکرد بهتری نسبت به ResNet دارد، اما ممکن است نیاز به محاسبات بیشتری داشته باشد.

ب)DenseNet چگونه مشکل گرادیان ناپدیدشونده را حل میکند و مزیت محاسباتی آن چیست؟

۱. **اتصال مستقیم تمامی لایه‌ها به یکدیگر:**

- درDenseNet ، هر لایه به تمام لایه‌های قبلی متصل است. به عبارت دیگر، ویژگی‌های استخراج‌شده از لایه‌های قبلی مستقیماً به عنوان ورودی به لایه‌های بعدی منتقل می‌شوند.
- این اتصال مستقیم باعث می‌شود که گرادیان‌ها بدون کاهش، از لایه‌های انتهایی به لایه‌های اولیه منتقل شوند.

۲. **مسیرهای متعدد برای انتشار گرادیان:**

- از آنجا که هر لایه به تمام لایه‌های قبلی متصل است، مسیرهای متعددی برای عبور گرادیان وجود دارد. این مسیرها باعث می‌شوند که گرادیان‌ها از لایه‌های انتهایی به لایه‌های اولیه بدون از دست رفتن اطلاعات منتقل شوند.
- به این ترتیب، حتی در شبکه‌های عمیق، اطلاعات و گرادیان به‌صورت کامل به لایه‌های ابتدایی می‌رسند.

۳. **افزایش استفاده از ویژگی‌ها:**

- درDenseNet ، ویژگی‌های هر لایه به صورت مستقیم به لایه‌های بعدی اضافه می‌شوند. این امر باعث می‌شود که اطلاعات اولیه حفظ شوند و نیازی به محاسبه دوباره آن‌ها نباشد.
- این بازاستفاده از ویژگی‌ها همچنین به بهبود یادگیری کمک می‌کند و مشکل از دست دادن اطلاعات در شبکه‌های عمیق را کاهش می‌دهد.

۴. **حذف نیاز به بازآفرینی اطلاعات:**

- برخلاف ResNet که فقط از اطلاعات لایه قبلی استفاده می‌کند، DenseNet اطلاعات را از تمام لایه‌های قبلی می‌گیرد. این روش تضمین می‌کند که گرادیان‌ها حتی در لایه‌های عمیق شبکه به اندازه کافی قوی باقی بمانند.

برتری DenseNet در کاهشVanishing Gradient

- **اتصالات مستقیم و پیوسته** DenseNet **:**از ساختاری بهره می‌برد که مسیرهای مستقیم و کوتاه برای انتقال گرادیان فراهم می‌کند.
- **جریان مؤثر اطلاعات :**اطلاعات ویژگی‌ها و گرادیان‌ها به طور مستقیم و مداوم در طول شبکه جریان پیدا می‌کنند، و این باعث می‌شود که اطلاعات در شبکه "گم" نشوند.
- **تعداد پارامترهای کمتر :**این طراحی به کاهش تعداد پارامترها کمک می‌کند، زیرا نیازی به محاسبه مجدد اطلاعات نیست و این امر باعث کاهش پیچیدگی محاسباتی می‌شود.

ج) چه زمان استفاده از DnseNet در مسائل عملی توصیه میشود؟

۱. **مسائل با داده‌های پیچیده و چندلایه:(High-Dimensional Data)**

- DenseNet در مشکلاتی که داده‌ها دارای ویژگی‌های پیچیده و چندسطحی هستند (مانند تصاویر با جزئیات زیاد) عملکرد خوبی دارد.
- به دلیل بازاستفاده از ویژگی‌های هر لایه، این معماری اطلاعات بیشتری را در هر مرحله پردازش می‌کند.

۲. **مسائل با محدودیت منابع محاسباتی:**

- DenseNet با به اشتراک‌گذاری ویژگی‌ها بین لایه‌ها تعداد پارامترها را کاهش می‌دهد. این باعث می‌شود که در مقایسه با شبکه‌هایی مانند ResNet که به تعداد بیشتری پارامتر نیاز دارند، DenseNet سبک‌تر باشد.

۳. **مسائل با عمق شبکه زیاد:**

- زمانی که شبکه نیاز به لایه‌های زیادی برای استخراج ویژگی‌های پیچیده دارد، DenseNet با جلوگیری از **ناپدید شدن گرادیان (Vanishing Gradient)** و انتقال اطلاعات در سراسر شبکه مناسب است.

۴. **مسائل نیازمند انتقال دانش از چندین لایه قبلی:**

- DenseNet برای مسائل با داده‌های چندبعدی و وابسته به ترکیب ویژگی‌ها مانند داده‌های چندحسی یا multimodal مناسب است، زیرا ویژگی‌های تمامی لایه‌های قبلی به لایه‌های بعدی منتقل می‌شود.

کاربرد: تشخیص بیماری از تصاویر پزشکی(Medical Image Diagnosis)

یکی از کاربردهای عملی و مهم DenseNet در حوزه **تصویربرداری پزشکی** است، به‌ویژه برای تشخیص بیماری‌ها از تصاویر رادیولوژی یا سی‌تی‌اسکن. برای مثال، در مسئله **تشخیص سرطان ریه از تصاویر سی‌تی‌اسکن**، DenseNet بسیار مؤثر است.

چرا DenseNet مناسب است؟

- **اطلاعات غنی‌تر از ویژگی‌ها :**در تصاویر پزشکی، جزئیات بسیار ظریف و مهمی وجود دارند که نیازمند ترکیب اطلاعات از چندین سطح (لایه) هستند.DenseNet به دلیل بازاستفاده از اطلاعات تمام لایه‌های قبلی، می‌تواند این ویژگی‌ها را بهتر استخراج کند.
- **پایداری گرادیان :**هنگام پردازش تصاویر پیچیده، نیاز است که اطلاعات در طول لایه‌ها منتقل شود. DenseNet این انتقال اطلاعات را تضمین می‌کند.
- **تعداد پارامترهای کمتر :**مدل‌های پردازش تصاویر پزشکی معمولاً روی سخت‌افزارهایی با محدودیت منابع اجرا می‌شوند (مانند بیمارستان‌ها). DenseNet به دلیل کاهش تعداد پارامترها، برای این شرایط ایده‌آل است.

(Case Study):

مسئله :تشخیص **رتینوپاتی دیابتی** از تصاویر چشم.

در این مسئله، مدل باید میزان آسیب به شبکیه را از تصاویر دقیق شناسایی کند. DenseNet توانسته است به دلیل توانایی در استخراج ویژگی‌های دقیق و عمیق، به دقت بالایی در این حوزه دست یابد.

نتایج:

- استفاده از DenseNet در این پروژه باعث بهبود دقت تشخیص شده است.
- زمان آموزش به دلیل کاهش تعداد پارامترها کمتر بوده است.
- بهبود انتقال گرادیان باعث یادگیری بهتر شبکه در مقایسه با معماری‌های دیگر شده است.

د) ارائه معماری مناسب برای پردازش داده های چند modality با استفاده DenseNet:

برای پردازش داده‌های ورودی متشکل از چند modality مانند تصویر و متن با استفاده ازDenseNet ، دو ساختار زیر را پیشنهاد میدهم:

ساختار پیشنهادی اول:

۱. **مدل تصویر:(DenseNet)**

- از DenseNet به عنوان مدل اصلی برای پردازش داده‌های تصویری استفاده می‌شود. این مدل به دلیل ساختار خاص خود که شامل اتصالات کوتاه‌مدت (skip connections) است، قادر به استخراج ویژگی‌های عمیق و دقیق از تصاویر می‌باشد.

۲. **مدل متن (LSTM یا Transformer)**

- برای پردازش متن، می‌توان از مدل‌هایی مانند LSTM یا Transformer استفاده کرد. این مدل‌ها توانایی بالایی در درک توالی و روابط معنایی در متن دارند.

۳. **ادغام ویژگی‌ها:**

- پس از استخراج ویژگی‌ها از هر دو مد، این ویژگی‌ها باید ادغام شوند. این کار می‌تواند با استفاده از روش‌هایی مانندconcatenation ، summationیا attention انجام شود.

۴. **لایه‌های:Fully Connected**

- بعد از ادغام ویژگی‌ها، می‌توان از چند لایه Fully Connected برای پردازش نهایی و پیش‌بینی استفاده کرد.

۵. **خروجی نهایی:**

- خروجی نهایی می‌تواند یک کلاس‌بندی، رگرسیون یا هر نوع پیش‌بینی دیگری باشد که بسته به وظیفه خاص متفاوت است.

کارایی DenseNet: **DenseNet** به دلیل استفاده از اتصالات بین لایه‌ها، می‌تواند اطلاعات را به طور مؤثری بین لایه‌ها منتقل کند و از مشکلاتی مانند ناپدید شدن گرادیان جلوگیری کند.

مدل متن :انتخاب LSTM یا Transformer به نیازهای خاص و نوع داده‌های متنی بستگی دارد LSTM .برای داده‌های توالی مناسب است، در حالی که Transformer به پردازش موازی کمک می‌کند و در بسیاری از وظایف NLP عملکرد بهتری دارد.

ادغام ویژگی‌ها :ادغام ویژگی‌ها از هر دو modality به مدل این امکان را می‌دهد که از اطلاعات چندگانه برای یادگیری بهتر بهره‌برداری کند.

ساختار پیشنهادی:

به دلیل محدودیت‌های موجود، نمی‌توانم تصویر را رسم کنم، اما می‌توانم توصیف کنم:

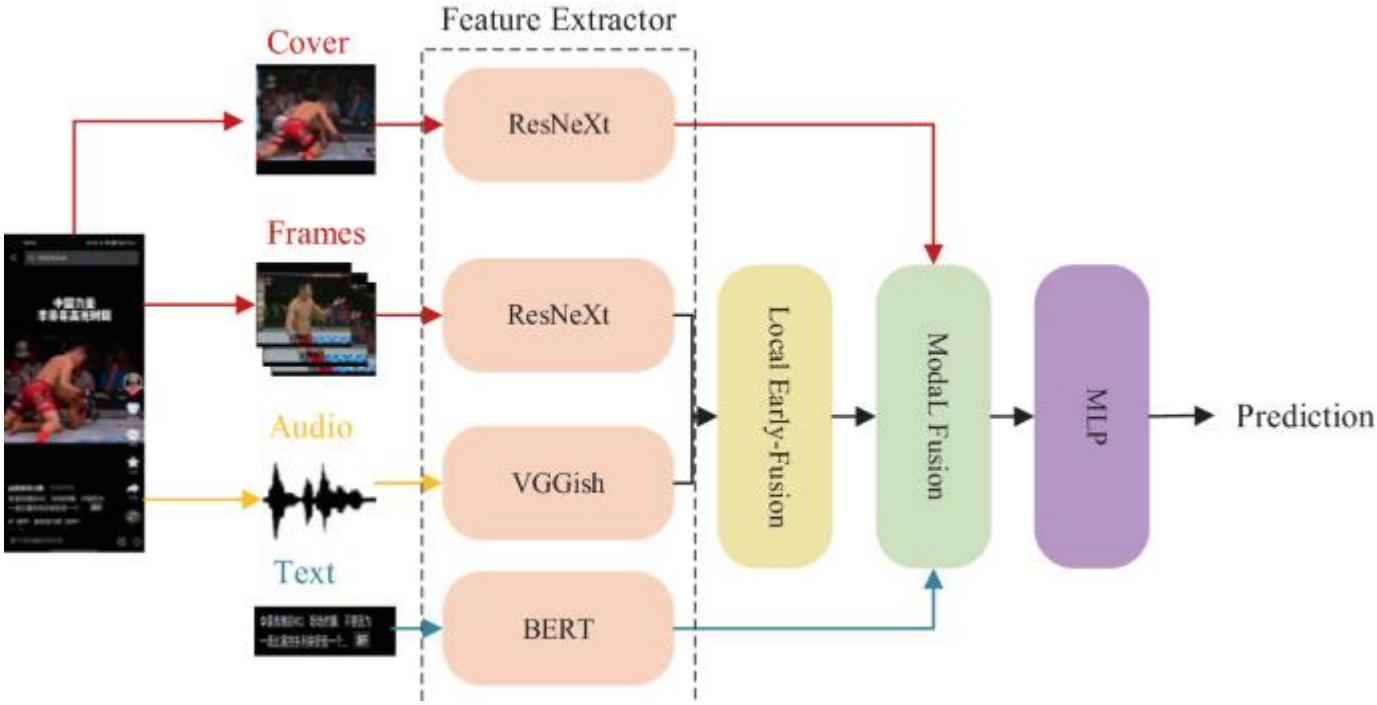
۱. **ورودی تصویر** → DenseNet →ویژگی‌های تصویری

۲. **ورودی متن** → LSTM/Transformer →ویژگی‌های متنی

۳. **ادغام ویژگی‌ها** → Fully Connected Layers →خروجی نهایی

این ساختار امکان استفاده همزمان از اطلاعات بصری و متنی را فراهم می‌آورد و می‌تواند به بهبود دقت در وظایف مختلف یادگیری ماشین کمک کند.

مانند ساختار زیر:



ساختار پیشنهادی دوم:

معماری شبکه در مقاله "Multimodal DenseNet" به شرح زیر است:

معماری:DenseNet

- شبکه اصلی بر پایه DenseNet طراحی شده است، که از لایه‌های متراکم (dense layers) استفاده می‌کند. در این معماری، هر لایه به ورودی لایه‌های قبلی متصل است و این اتصالات به کاهش مشکل گرادیان‌های ناپدید vanishing gradients کمک می‌کند.

مدل چندمودالیتی:

- **دو کانال ورودی** :یکی برای داده‌های تصویری (RGB) و دیگری برای داده‌های عمق (Depth) یا تصاویر نوار باند (Narrow Band Imaging).
- در ابتدا، این دو مدالیت به طور جداگانه از طریق لایه‌های DenseNet پردازش می‌شوند.

ادغام ویژگی‌ها:

- ویژگی‌های استخراج شده از دو کانال در لایه‌های پایانی ادغام می‌شوند. این ادغام در لایه‌های مختلف و به شیوه‌ای تدریجی انجام می‌شود تا اطلاعات هر دو منبع به خوبی ترکیب شوند.

لایه‌های انتقال:

- بین بلاک‌هایDense ، لایه‌های انتقال (Transition Layers) وجود دارند که شامل Batch Normalization، ReLU، و میانگین‌گیری (Average Pooling) هستند.

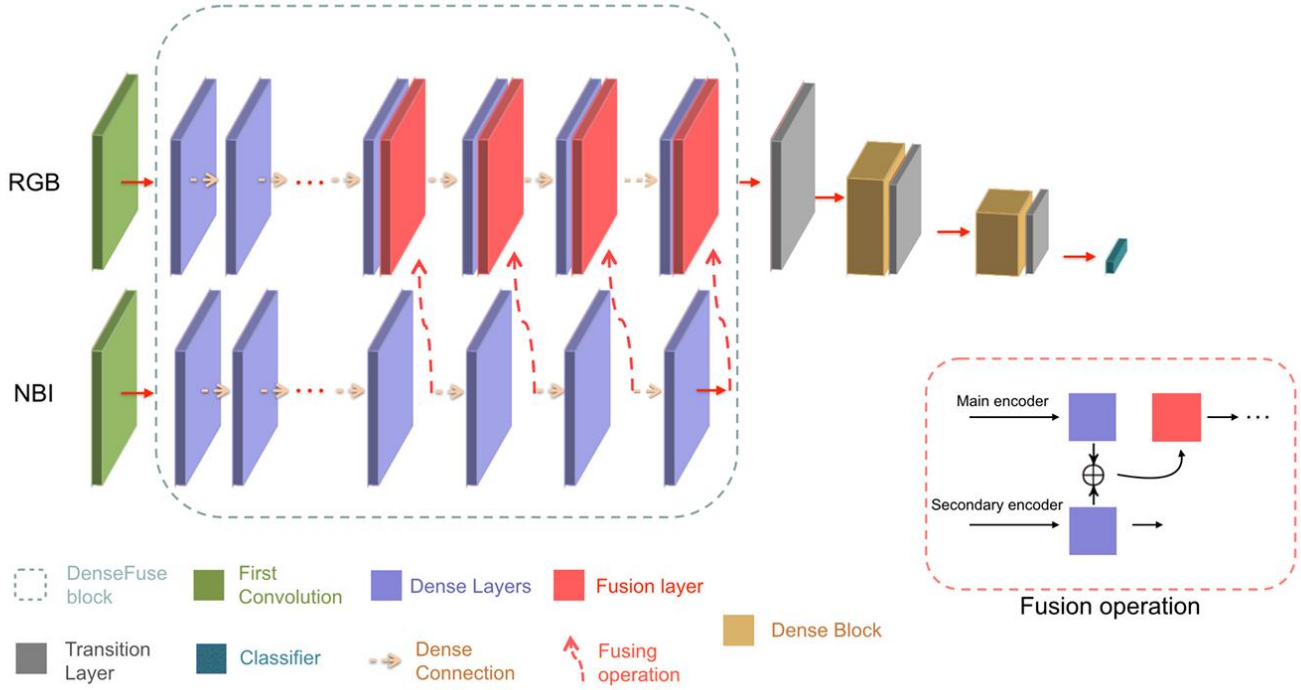
خروجی نهایی:

- در نهایت، داده‌ها از طریق یک لایه Fully Connected و Softmax برای طبقه‌بندی نهایی عبور می‌کنند.

نقاط قوت معماری:

- **فراهم کردن انعطاف‌پذیری** در ترکیب اطلاعات از چند مدالیت.
- **حل مشکل گرادیان‌های ناپدید** به دلیل اتصالات متراکم.
- **عملکرد بهتر در مقایسه با تکنیک‌های دیگر** در طبقه‌بندی و شناسایی.

این معماری به طور خاص برای چالش‌هایی مانند شناسایی و طبقه‌بندی پولیپ‌ها و شناسایی نقاط مرجع در اندوسکوپی طراحی شده است.



کاربرد عملی

این معماری در مسائل **چندوجهی** مانند **توصیف تصویر(Image Captioning)** ، **تشخیص احساس از ویدئو و متن (Sentiment Analysis)**، یا **تشخیص بیماری بر اساس تصاویر و گزارش‌های پزشکی متنی** کاربرد دارد.

مثال :در پزشکی، می‌توان از این معماری برای تحلیل تصاویر سی‌تی‌اسکن و ترکیب آن با گزارش‌های متنی پزشکان برای تشخیص دقیق‌تر بیماری‌ها استفاده کرد.

پاسخ سوال ۳)

آ) دلیل اتصالات بین بخش Decoder و Encoder در شبکه U-Net:

دلیل وجود اتصالاتSkip

- **حفظ اطلاعات مکانی دقیق:**
  - در طی فرآیند فشرده‌سازی(Encoding) ، ویژگی‌های سطح پایین که شامل جزئیات مکانی هستند، به دلیل عملیات‌هایی مانند Pooling از دست می‌روند.
  - اتصالات Skip این اطلاعات را از لایه‌های کدگذار گرفته و مستقیماً به لایه‌های متناظر در بخش کدگشا منتقل می‌کنند تا این جزئیات در بازسازی تصویر حفظ شوند.
- **ترکیب اطلاعات سطح بالا و پایین:**
  - لایه‌های Decoder اطلاعات زمینه (Context) را در سطح بالا استخراج می‌کنند، در حالی که لایه‌های Encoder نیازمند ترکیب این اطلاعات با جزئیات محلی (Local Details) هستند. اتصالات Skip این امکان را فراهم می‌کنند.

تأثیر اتصالاتSkip

- **افزایش دقت تقسیم‌بندی:**
  - ترکیب اطلاعات جزئیاتی و زمینه‌ای باعث می‌شود که مرزهای اشیاء در تصویر به‌دقت بیشتری شناسایی شوند، به‌ویژه در مسائل تقسیم‌بندی دقیق مانند تصاویر زیست‌پزشکی.
- **بهبود یادگیری شبکه:**
  - اتصالات Skip جریان گرادیان را از بخش کدگشا به کدگذار تسهیل می‌کنند. این امر باعث می‌شود مشکل ناپدید شدن گرادیان (Vanishing Gradient) کاهش یابد و شبکه بتواند یادگیری بهتری داشته باشد.
- **کاهش نیاز به محاسبات اضافی:**
  - به جای اینکه شبکه دوباره اطلاعات از دست‌رفته را بازآفرینی کند، اتصالات Skip این اطلاعات را مستقیماً از لایه‌های قبلی بازیابی می‌کنند.

مثال از تأثیر عملی

- در مقاله، نشان داده شده است که این اتصالات باعث می‌شوند U-Net بتواند ساختارهای کوچک و پیچیده مانند غشاهای سلولی یا مرزهای سلولی را با دقت بالاتری تقسیم‌بندی کند. به‌ویژه در تصاویری مانند داده‌های میکروسکوپی، این جزئیات برای تحلیل نتایج حیاتی هستند.

اتصالات Skip یکی از ویژگی‌های کلیدی U-Net هستند که امکان ترکیب اطلاعات دقیق محلی با ویژگی‌های زمینه‌ای را فراهم می‌کنند. این اتصالات منجر به تقسیم‌بندی دقیق‌تر، یادگیری مؤثرتر و بهبود جریان اطلاعات در طول شبکه می‌شوند.

ب) چگونگی انجام Random Deformation:

در مقاله، تکنیک **تغییر شکل تصادفی (Random Deformation)** برای افزایش داده‌های آموزشی استفاده شده است. این تکنیک با اعمال تغییرات تصادفی به تصاویر آموزشی اولیه، شبکه را به مقاومت در برابر تغییرات واقعی موجود در داده‌ها مجهز می‌کند. مراحل انجام این تکنیک به شرح زیر است.

شبکه نقاط جابجایی(Displacement Grid)

ایجاد شبکه:

- یک شبکه‌ی مربعی با ابعاد 3×3 روی تصویر آموزشی تعریف می‌شود. این نقاط به عنوان مراکز اصلی برای ایجاد تغییرات عمل می‌کنند.

جابجایی تصادفی نقاط:

- برای هر نقطه از این شبکه، جابجایی مختصاتی dx, dy به‌صورت تصادفی از یک **توزیع گاوسی** نمونه برداری می‌شود.
- انحراف معیار این توزیع معمولاً مقدار ثابتی مانند ۱۰ **پیکسل** تنظیم می‌شود. این مقدار تضمین می‌کند که تغییرات خیلی زیاد نباشند و ساختار کلی تصویر حفظ شود.

اعمال تغییرات به تصویر

درون‌یابی:(Interpolation)

- برای محاسبه جابجایی هر پیکسل تصویر، جابجایی‌های نقاط شبکه با استفاده از **درون‌یابی مکعبی** در کل تصویر تعمیم داده می‌شود. این درون‌یابی باعث می‌شود تغییرات به صورت پیوسته و طبیعی اعمال شوند.

اعمال تغییر شکل:(Deformation)

- با استفاده از جابجایی‌های محاسبه‌شده، هر پیکسل تصویر به مختصات جدید خود منتقل می‌شود.
- این انتقال شامل تغییراتی مانند کشیدگی(Stretching) ، فشردگی(Compression) ، و انحراف (Shearing) است که تصویر را تغییر می‌دهند، اما روابط فضایی بین بخش‌های مختلف تصویر حفظ می‌شود.

کنترل تغییرات

حفظ انسجام تصویر:

- تکنیک به گونه‌ای طراحی شده که **جزئیات محلی تصویر حفظ شوند**. برای مثال، در تصاویر زیست‌پزشکی مانند سلول‌ها، این روش تضمین می‌کند که مرزهای سلول یا ساختارهای حساس از بین نروند.

تنظیم شدت تغییرات:

- پارامترهایی مانند انحراف معیار توزیع گاوسی یا تعداد نقاط شبکه می‌توانند شدت تغییرات را کنترل کنند. برای تصاویر زیست‌پزشکی، این پارامترها طوری انتخاب می‌شوند که تغییرات مشابه تغییرات طبیعی بافت‌ها باشند.

چرا این روش مؤثر است؟

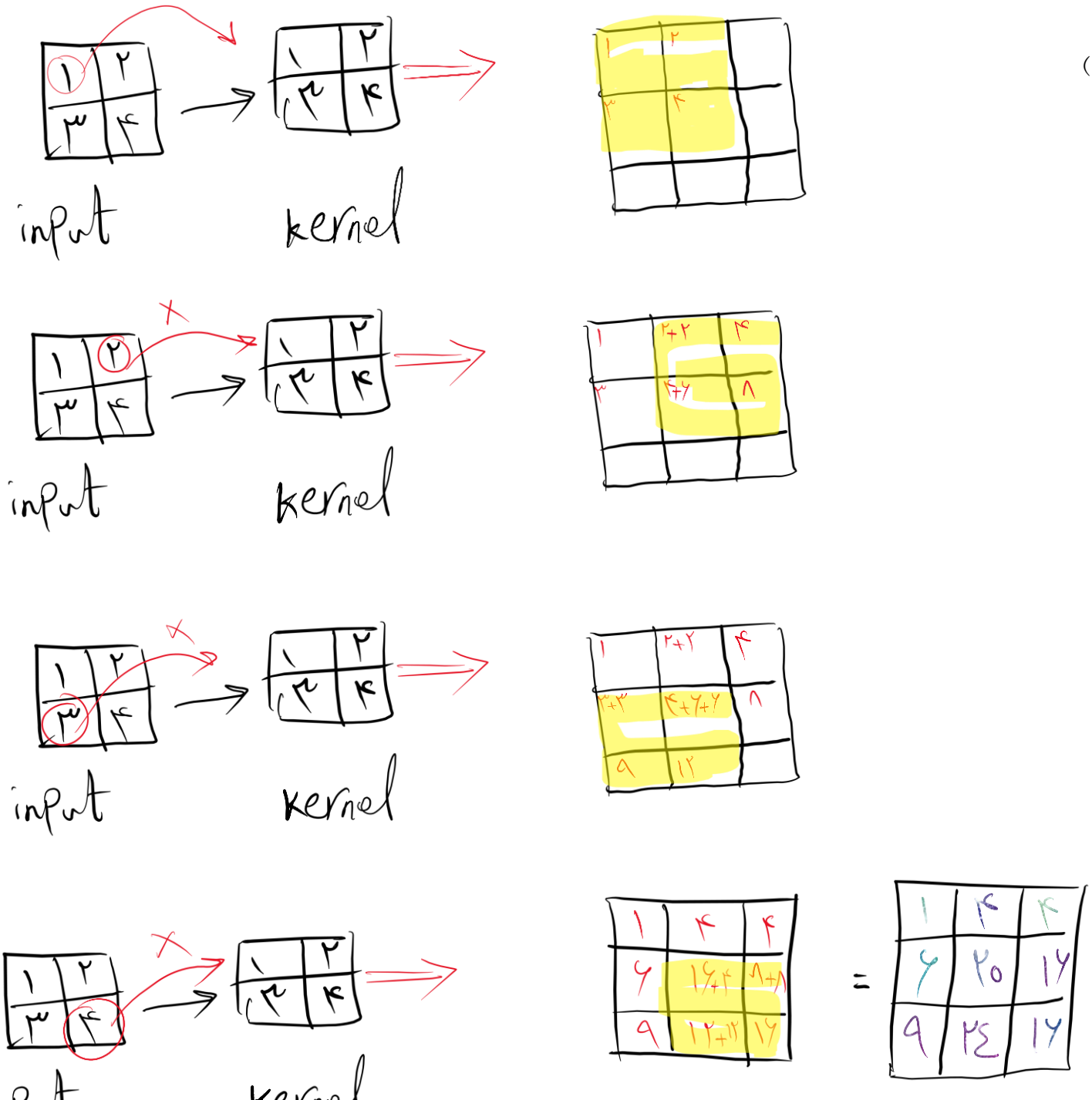
۱. **افزایش مقاومت شبکه به تغییرات :**شبکه یاد می‌گیرد که به تغییرات طبیعی مانند انحرافات یا تغییر شکل‌های بافتی حساس نباشد.

۲. **تولید داده‌های متنوع از یک مجموعه کوچک :**با اعمال تغییرات مختلف، تصاویر بیشتری تولید می‌شود که به مدل اجازه می‌دهد بهتر تعمیم دهد.

۳. **شبیه‌سازی تغییرات واقعی :**بسیاری از تغییرات ایجادشده توسط این تکنیک شبیه تغییراتی هستند که در داده‌های واقعی زیست‌پزشکی رخ می‌دهند، مانند تغییر شکل بافت‌ها.

تکنیک Random Deformation با اعمال تغییرات ظریف و کنترل‌شده به تصاویر آموزشی، یکی از کلیدهای موفقیت U-Net در یادگیری از داده‌های محدود است. این روش باعث می‌شود که مدل بتواند تصاویر پیچیده‌تر و تغییرات طبیعی را بهتر شناسایی کند.

د)



پاسخ سوال ٤)

آ)

YOLOv1

In YOLOv1, the output of the network is a tensor of shape $S \times S \times (B \times 5 + C)$, where:

- S: Grid size ($S = 7$)
- B: Number of bounding boxes per grid cell ($B = 2$ in YOLOv1).
- 5: Represents x, y, w, h and confidence for each bounding box.
- C: Number of classes($C = 80$).

Thus, the depth (number of channels) in YOLOv1's output is calculated as:

$$B \times 5 + C = 90$$

YOLOv3

In YOLOv3, the output predictions are made at three scales, and each scale predicts a tensor of shape $N \times N \times [B \times (5 + C)]$ where:

- N: Feature map size at each scale (depending on the scale).
- B: Number of anchor boxes per grid cell (B=3B = 3B=3 in YOLOv3).
- 5: Represents x, y, w, h and objectness score for each anchor box.
- C: Number of classes ($C = 80$).

At each scale, the depth of the output tensor is:

$$B \times (5 + C) = 255$$

مقایسه و دلیل تفاوت YOLOv1 و YOLOv3:

YOLOv1: خروجی در هر سلول تنها شامل ۲ جعبه محدودکننده است و از یک توزیع احتمال شرطی (Softmax) برای پیش‌بینی کلاس‌ها استفاده می‌کند، بنابراین عمق خروجی کمتر است (۹۰ کانال).

YOLOv3: به جایSoftmax ، از طبقه‌بندی Multi-label Classification استفاده می‌کند که از طبقه‌بندی مستقل برای هر کلاس بهره می‌برد. همچنین از ۳ جعبه پیش‌فرض (Anchor Boxes) برای هر سلول استفاده می‌کند. این تغییرات باعث افزایش عمق خروجی به ۲۵۵ کانال می‌شود.

در کل، تفاوت در تعداد Bounding Boxes و تغییر در نحوه مدل‌سازی کلاس‌ها از Softmax به Logistic Regression دلیل افزایش عمق خروجی در YOLOv3 است.

- استفاده از ۳ Anchor Box به جای ۲ Bounding Box در هر سلول.
- استفاده از طبقه‌بندی‌کننده‌های لجستیک مستقل به‌جای یک خروجی softmax واحد.
- معرفی پیش‌بینی‌های چند مقیاسی در YOLOv3 برای بهبود تشخیص در اشیاء با اندازه‌های مختلف.

ب) چه راهکاری در YOLOv3 برای غلبه به مشکل همپوشانی برچسب ها ارائه شده است؟

در YOLOv3، برای مدیریت برچسب‌های همپوشانی و امکان عدم تعلق اشیا به یک کلاس داده واحد، استفاده از طبقه‌بندی‌کننده‌های لجستیک مستقل برای پیش‌بینی کلاس است، برخلاف فعال‌سازی softmax که در نسخه‌های قبلی مانند YOLOv1 استفاده می‌شد.

YOLOv3 تابع فعال سازی softmax را با طبقه بندی کننده های لجستیک مستقل برای هر کلاس جایگزین می کند. هر پیش‌بینی کلاس به عنوان یک مسئله طبقه‌بندی باینری در نظر گرفته می‌شود، جایی که شبکه احتمال حضور هر کلاس را مستقل از سایرین پیش‌بینی می‌کند. این به این معنی است که چندین کلاس را می توان برای یک شی پیش بینی کرد که امکان طبقه بندی Multi-label Classification را فراهم می کند.

این رویکرد برای مجموعه‌های داده با برچسب‌های همپوشانی خوب عمل می‌کند (به عنوان مثال، یک شی را می‌توان هم به عنوان "شخص" و هم "ورزشکار" برچسب‌گذاری کرد) و از فرض دقیق انحصار متقابل بین کلاس ها، که در تابع فعال سازی softmax ذاتی است، اجتناب می کند.

برای پیش‌بینی‌های کلاس، YOLOv3 از Binary Cross-Entropy Loss به جای Categorical Cross-Entropy Loss استفاده می‌کند. این تضمین می کند که پیش بینی های هر کلاس به طور مستقل بررسی می شوند.

بنابراین YOLOv3 مشکل همپوشانی برچسب‌ها و کلاس‌های غیر انحصاری را توسط:

- استفاده از طبقه بندی کننده های لجستیک مستقل به جای softmax برای پیش بینی کلاس.
- امکان طبقه بندی Multi-label Classification، که در آن یک شی می تواند به چند کلاس به طور همزمان تعلق داشته باشد.
- آموزش Binary Cross-Entropy Loss برای بهینه سازی پیش بینی هر کلاس به طور مستقل.

این رویکرد انعطاف پذیری YOLOv3 را افزایش می دهد و آن را برای مجموعه داده های پیچیده با برچسب های همپوشانی یا سلسله مراتبی مناسب تر می کند.

ج) الگوریتم های استفاده شده در YOLO برای جلوگیری از تشخیص تکراری و چندگانه اشیاء

در مقالات YOLO، Non-Maximum Suppression (NMS) برای جلوگیری از شناسایی مکرر و چندگانه یک شی استفاده می شود.

نحوه عملکرد NMS:

- **آستانه اعتماد:** ابتدا، الگوریتم bounding box ها با امتیازات اطمینان پایین (شیئ بودن) را فیلتر می‌کند تا تعداد نامزدهای در نظر گرفته شده را کاهش دهد.
- **مرتب سازی بر اساس اطمینان:** bounding box های باقیمانده بر اساس امتیاز اطمینان آنها به ترتیب نزولی مرتب می شوند.
- **محاسبه IoU:** برای هر bounding box، الگوریتم IoU بین این کادر و تمام کادرهای دیگر را محاسبه می کند.
- **سرکوب جعبه های همپوشانی:** اگر IoU جعبه ای با کادر با امتیاز بالاتر از آستانه از پیش تعریف شده (مثلاً ۰.۵) فراتر رود، کادر با امتیاز پایین تر سرکوب می شود (یعنی از بررسی حذف می شود).
- **تشخیص های عدم همپوشانی خروجی:** پس از پردازش، تنها جعبه‌های مرزی با بالاترین اطمینان برای هر شی حفظ می‌شوند و پیش‌بینی‌های اضافی یا همپوشانی را حذف می‌کنند.

چرا NMS موثر است:

YOLO اغلب به دلیل همپوشانی سلول‌های شبکه یا Anchor Box چندگانه، چندین Bounding Box را برای یک شی پیش‌بینی می‌کند. NMS تضمین می کند که فقط مطمئن ترین پیش بینی برای هر شیء حفظ می شود. NMS از نظر محاسباتی کارآمد است.

استفاده از NMS در نسخه های YOLO:

YOLOv1: NMS به عنوان یک مرحله پس از پردازش برای حذف تشخیص های تکراری اعمال می شود.

YOLOv2 and YOLOv3: NMS همچنان مورد استفاده قرار می‌گیرد، اما با پیش‌بینی Bonding Box های پیشرفته، مانند Anchor Box و خروجی‌های چند مقیاسی، کارایی آن را در سرکوب تکرارها بهبود می‌بخشد.

Non-Maximum Suppression (NMS) الگوریتم کلیدی است که در YOLO برای حذف چندین تشخیص از یک شی استفاده می‌شود و تضمین می‌کند که فقط مطمئن‌ترین و غیر همپوشانی‌ترین پیش‌بینی‌ها حفظ می‌شوند.

د) چرا در YOLOv2 و YOLOv3 شبکه قابلیت مقیاس پذیری ورودی دارد؟

برخلاف YOLOv1، YOLOv2 و YOLOv3 دارای آموزش چند مقیاسی هستند که به شبکه‌ها اجازه می‌دهد بر روی تصاویر با اندازه‌های مختلف آموزش داده و ارزیابی شوند. این به دلیل تغییرات معماری و معرفی شده در YOLOv2 و در YOLOv3 انجام شده است.YOLOv1 به یک اندازه ورودی ثابت هم برای آموزش و هم برای استنتاج نیاز دارد.

این محدودیت به این دلیل به وجود می آید که لایه های کاملاً متصل نهایی به اندازه وابسته هستند و وزن آنها به طور خاص به ابعاد ثابت ورودی گره خورده است.

تغییرات در YOLOv2 و YOLOv3

YOLOv2 و YOLOv3 لایه‌های کاملاً متصل را حذف می‌کنند و کاملاً به عملیات کانولوشن و ادغام متکی هستند.

اندازه تانسور خروجی متناسب با اندازه تصویر ورودی می شود. برای پیش بینی ها، و تغییر اندازه ورودی منجر به تغییرات متناسب در شبکه خروجی می شود.

در طول آموزش، شبکه به طور تصادفی اندازه تصاویر ورودی را هر چند دسته تغییر می دهد. همچنین تغییر اندازه به ابعاد انتخاب شده از یک مجموعه از پیش تعریف شده انجام می شود. این مدل بر روی این وضوح های مختلف آموزش داده شده است، و آن را مجبور می کند تا پیش بینی های دقیق را در مقیاس های مختلف بیاموزد.

چرا آموزش چند مقیاسی امکان پذیر است؟ معماری کانولوشن، تصاویر ورودی را به صورت فضایی پردازش می‌کند و وزن‌ها در تمام مکان‌های فضایی به اشتراک گذاشته می‌شوند. این ویژگی به شبکه اجازه می دهد تا به اندازه های مختلف تصویر تعمیم یابد. از آنجایی که اندازه خروجی شبکه به طور خودکار با اندازه ورودی تنظیم می شود، نیازی به بازآموزی یا تغییر ساختار معماری برای وضوح های مختلف وجود ندارد.

مزایای آموزش چند مقیاسی

- **سازگاری با برنامه های مختلف:** شبکه می تواند به صورت پویا با منابع محاسباتی مختلف یا نیازهای کاربردی سازگار شود. اندازه ورودی کوچکتر: استنتاج سریعتر اما دقت کمی پایین تر، مناسب برای برنامه های بلادرنگ در دستگاه های با محدودیت منابع. اندازه ورودی بزرگتر: دقت بالاتر به قیمت استنتاج کندتر، مناسب برای تجزیه و تحلیل دقیق.
- **استحکام بهبود یافته**: آموزش در اندازه های مختلف تصویر، شبکه را در مقیاس تغییرات در تصاویر دنیای واقعی قوی می کند، مانند بزرگتر یا کوچکتر ظاهر شدن اشیا بسته به فاصله آنها از دوربین.
- **استفاده بهینه از منابع**: اندازه‌های کوچک‌تر تصویر در طول آموزش، تکرارها را سرعت می‌بخشد و در عین حال به شبکه اجازه می‌دهد تا وضوح‌ها را به خوبی تعمیم دهد.

YOLOv2 و YOLOv3 از طریق معماری کاملاً کانولوشن و آموزش چند مقیاسی به انعطاف پذیری اندازه دست می یابند. این تغییرات باعث می‌شود شبکه با وضوح‌های ورودی مختلف سازگار باشد، استحکام در تغییرات مقیاس را بهبود بخشد و بین سرعت و دقت در حین استنتاج تعادل ایجاد کند. این قابلیت برای کاربردهای عملی، از پردازنده‌های گرافیکی با کارایی بالا تا سیستم‌های تعبیه‌شده کم مصرف، حیاتی است.

ه) مشکلات و راهکار های مقابله با استفاده از Anchor Box در YOLOv2 و راه های مقابله با آن را بیان کنید.

مقاله YOLOv2 دو مشکل اصلی استفاده از Anchor Box را برجسته می‌کند و راه‌حل‌های خاصی برای رفع آنها ارائه می‌کند:

مشکل اول: **Hand-Picked Anchor Box Dimensions**

در روش های سنتی (به عنوان مثال، Faster R-CNN)، ابعاد Anchor Box به صورت دستی بر اساس قوانین اکتشافی یا شهود تعریف می شود و ممکن است توزیع واقعی اندازه‌ها و اشکال شی در مجموعه داده را نشان ندهند، و یادگیری پیش‌بینی‌های Bounding Box خوب را برای شبکه سخت‌تر می‌کنند.

راه حل: **Dimension Clusters**

YOLOv2 از خوشه‌بندی k-means بر روی ابعاد Bounding Box ها در مجموعه داده آموزشی استفاده می‌کند تا اولویت‌های بهتری برای Anchor Box ایجاد کند. به جای استفاده از فاصله استاندارد اقلیدسی در الگوریتم k-means، یک متریک فاصله سفارشی استفاده می شود:

$$d(box, centroid) = 1 - IoU(box, centroid)$$

این معیار اولویت‌بندی جعبه‌های پیشینی را تعیین می‌کند که (IoU) را با ground truth به حداکثر می‌رساند، و منجر به Anchor Box نمونه تر می‌شود. با استفاده از این روش، YOLOv2 ۵ یا ۹ Anchor پیشین را شناسایی می کند که تغییرپذیری در شکل و اندازه اشیا را بهتر نشان می دهد. در واقع با این کار شبکه با Anchor Box هایی شروع می‌شود که به واقعیت نزدیک‌تر هستند، پیچیدگی یادگیری را کاهش می‌دهند و عملکرد را بهبود می‌بخشند.

مشکل دوم: بی ثباتی در پیش بینی مکان جعبه

پیشبینی مختصات x, y Bouding Box ها به‌طور مستقیم به‌عنوان جابجایی مطلق منجر به بی‌ثباتی در طول تمرین، به‌ویژه در تکرارهای اولیه می‌شود. این بی‌ثباتی به این دلیل به وجود می‌آید که Anchor Box ها می‌توانند خودسرانه در تصویر جابه‌جا شوند و باعث ایجاد گرادیان‌های غیرقابل پیش‌بینی و هم‌گرایی کندتر شوند.

راه حل: پیش بینی موقعیت مکانی مستقیم

YOLOv2 مختصات Bounding Box را به عنوان جابجایی نسبی به سلول شبکه ای که Anchor Box در آن قرار دارد، پیش بینی می کند. یک تابع فعال سازی سیگموئید برای محدود کردن پیشبینی های x, y اعمال می شود تا مقادیر را بین ۰ و ۱ نگه دارد و اطمینان می‌دهد که کادر پیش بینی شده در محدوده سلول شبکه باقی می ماند.

$b_x = \sigma(t_x) + c_x, \qquad b_y = \sigma(t_y) + c_y$

b_x, b_y: Predicted box center coordinates.

t_x, t_y: Predicted offsets.

c_x, c_y: Top-left corner of the grid cell.

این رویکرد با محدود کردن مکان‌های جعبه برای قرار گرفتن در نزدیکی سلول شبکه مربوطه، آموزش را تثبیت می‌کند و بهینه‌سازی شبکه را آسان‌تر و قابل اطمینان‌تر می‌کند.

و) تفاوت کلیدی معماری YOLOv3:

معماری شبکه YOLOv3 چندین تفاوت و پیشرفت کلیدی را در مقایسه با YOLOv2 معرفی می‌کند و بر دقت، انعطاف‌پذیری و استحکام بهتر تمرکز دارد و در عین حال عملکرد بلادرنگ خود را حفظ می‌کند.

۱. شبکه زیرساختی

YOLOv2: از Darknet-19، یک شبکه عصبی کانولوشن سبک وزن با ۱۹ لایه کانولوشن و ۵ لایه Max-pooling استفاده می کند و برای سرعت و کارایی طراحی شده است اما فاقد قدرت شبکه های عمیق تر است.

YOLOv3: Darknet-53 را معرفی می کند که شبکه ای عمیق تر و قوی تر با ۵۳ لایه کانولوشن و با الهام از ResNet برای بهبود جریان گرادیان و همگرایی در طول آموزش، دارای Skip connection است که با حفظ عملکرد کارآمد، به دقت بالاتری دست می‌یابد.

۲. پیش بینی های چند مقیاسی

YOLOv2: پیش بینی ها را در یک مقیاس انجام می دهد.

YOLOv3:جعبه های مرزی را در سه مقیاس پیش بینی می کند: مقیاس درشت، مقیاس متوسط، مقیاس ریز این پیش‌بینی چند مقیاسی عملکرد را در طیف وسیعی از اندازه‌های شی افزایش می‌دهد.

۳. پیش بینی جعبه مرزی

YOLOv2: از Anchor Box ها با ۵ پیش بینی در هر سلول استفاده میکند.

YOLOv3: هنوز از Anchor Box ها استفاده می کند اما تعداد Anchor ها را به ۳ در هر مقیاس افزایش می دهد. یک تانسور خروجی جدید با اندازه $N \times N \times\times [3 \times (5 + C)]$ را معرفی می کند که C در هر مقیاس تعداد کلاس ها را نشان میدهد.

٤. پیش بینی کلاس

YOLOv2: با فرض انحصار متقابل بین کلاس‌ها، از فعال‌سازی softmax برای احتمالات کلاس استفاده می‌کند.

YOLOv3:Softmax را با طبقه‌بندی‌کننده‌های لجستیک مستقل برای طبقه‌بندی چند برچسبی جایگزین می‌کند. هر پیش‌بینی کلاس به‌عنوان یک مسأله طبقه‌بندی باینری در نظر گرفته می‌شود، که اشیا میتوانند همزمان به چند کلاس تعلق داشته باشند.

۵. تجمیع ویژگی

YOLOv2: دارای یک لایه عبوری برای ترکیب ویژگی های دانه ریز و درشت.

YOLOv3: از یک ساختار شبه هرمی (FPN) مانند استفاده می کند:

نمونه‌های بالا نقشه‌هایی را از مقیاس‌های درشت‌تر نشان می‌دهند و آن‌ها را با نقشه‌های ویژگی با دانه‌ریزتر ترکیب می‌کنند. این رویکرد غنای معنایی پیش بینی ها را در مقیاس های کوچکتر بهبود می بخشد.

٦. توابع فعال سازی

YOLOv2: در درجه اول از leaky ReLU برای فعال سازی استفاده می کند.

YOLOv3: leaky ReLU را حفظ می‌کند اما از فعال‌سازی خطی برای لایه‌های نهایی برای پیش‌بینی جابجایی‌های جعبه و احتمالات کلاس استفاده می‌کند.

۷. بهبود عملکرد

YOLOv3: اگرچه کمی کندتر از YOLOv2 است، اما به طور قابل توجهی دقیق تر است، به ویژه برای اشیاء کوچک و برچسب های همپوشانی و همچنین با استفاده از پیش بینی های چند مقیاسی و Darknet-53 سرعت و دقت را به طور موثرتری متعادل می کند. این تغییرات باعث می شود YOLOv3 همه کاره تر و دقیق تر شود و در عین حال قابلیت های تشخیص بلادرنگ آن را حفظ کند.