به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین سوم

نام و نام خانوادگی	حسام اسداله‌زاده – مسعود طهماسبی
شماره دانشجویی	810198346 – 810198429
تاریخ ارسال گزارش	۱۴۰۱.۰۹.۲۲

فهرست

شکل‌ها

جدول‌ها

پاسخ 1. آشنایی با یادگیری انتقالی (Transfer Learning)

رقم آخر شماره دانشجویی نفر اول = 9

رقم آخر شماره دانشجویی نفر دوم = 6

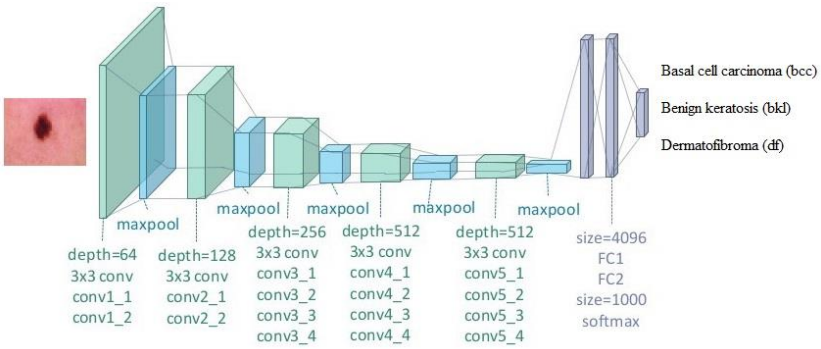
$$6 + 9 \equiv 3 \pmod 4$$

Skin Cancer Classification Model Based on VGG19 and Transfer Learning

۱-۱- گزارش و خلاصه‌ی مقاله

سرطان پوست یک مشکل سلامتی نگران کننده است که تعداد آن به صورت سالانه افزایش می‌یابد. تشخیص و طبقه‌بندی نوع سرطان مشکل‌ساز است، به‌ویژه از آنجایی که بیماران باید چندین مرحله تشخیص (diagnosis) را در دوره‌های زمانی طولانی انجام دهند، که مانع از درمان زودهنگام و موجب کاهش شانس بهبودی می‌شود. با کمک پردازش تصویر دیجیتال، می‌توان ویژگی‌هایی را برای شناسایی سرطان پوست و انواع مختلف آن استخراج کرد. شبکه‌های عصبی کانولوشنال (CNN) اخیراً به‌عنوان استخراج‌کننده ویژگی‌های مستقل قدرتمند ظاهر شده‌اند و پتانسیل بالایی برای دستیابی به دقت بالایی در تشخیص سرطان پوست دارند. در این مقاله، دو نوع سرطان علاوه بر یک نوع غیر سرطانی برگرفته از مجموعه داده انسان در برابر ماشین (HAM10000) با استفاده از مدل CNN بر اساس VGG19 و تکنیک یادگیری انتقالی طبقه‌بندی می‌شوند. استراتژی آموزش با محاسبه دقت و loss کلی شبکه توضیح، آزمایش و ارزیابی شده است.

مجموعه داده HAM10000 برای این مطالعه انتخاب شده است که حاوی تصاویری از انواع مختلف سرطان پوست است. دو کلاس از این مجموعه داده انتخاب شده است. درماتوفیبروما (DF) و کارسینوم سلول بازال (BCC)، علاوه بر یک ضایعه‌ی خوش خیم کراتوز مانند نوع غیر سرطانی (BKL) انتخاب شده است. با این حال، به دلیل رایج تر بودن BKL نسبت به دو مورد دیگر، عدم تعادل در مجموعه داده وجود دارد. عدم تعادل می تواند بر روند train تاثیر منفی بگذارد و به طور بالقوه باعث بایاس شود. این نیز نوعی بایاس داده است که در آن یک نمونه بیشتر از بقیه نمایش داده می شود. بنابراین، از data augmentation برای افزایش داده‌های کلاس‌های DF و BCC استفاده می‌شود. روش‌های augmentation شامل برش، تغییر مقیاس، تنظیم کنتراست و روشنایی، چرخش افقی، چرخش عمودی و ترکیبی از این روش‌ها است. پس از augmentation ، هر نوع سرطان پوست دارای 1000 نمونه در مجموعه داده است و اندازه نهایی مجموعه داده 3000 است. اندازه تصاویر در مجموعه داده به 64×64 تغییر یافته است.



شکل**1** . معماری شبکه **VGG19**

VGG19 برای اولین بار در مقاله‌ی:

“Very Deep Convolutional Networks for Large-Scale Image Recognition”

توسعه یافت که نسخه‌ی پیشرفته VGG16 است. VGG19 یک CNN عمیق است که از چندین لایه کانولوشن و لایه‌های max pooling تشکیل شده است که به عنوان استخراج‌کننده ویژگی شناخته می‌شوند. حداقل یک لایه fully connected به دنبال این لایه ها قرار می‌گیرد که به عنوان طبقه‌بندی کننده شناخته می‌شود. اندازه و تعداد لایه‌های کانولوشن و fully connected به عنوان یک انتخاب طراحی تعیین شده توسط معمار CNN در نظر گرفته می‌شود. ساختار کلی VGG19 در شکل1 نشان داده شده است. لایه ورودی به اندازه 64 × 64 تنظیم شده است و لایه خروجی با تابع فعال‌سازی softmax جایگزین شده است که احتمال تعلق به یکی از سه نوع سرطان را نمایش می‌دهد. قابلیت استخراج مستقل ویژگی‌های VGG19، یافتن ویژگی‌هایی را که هر نوع سرطان را متمایز می‌کند، بدون نیاز به صرف زمان برای بررسی دستی، آسان می‌کند. این مدل VGG19 برای طبقه‌بندی سرطان پوست مورد استفاده قرار گرفته است. با استفاده از یک VGG19 از پیش آموزش دیده با پارامترهای دقیق، آموزش انتقالی (TL) اعمال شده است. ٪80 از مجموعه داده برای train شبکه استفاده شده و بقیه برای test استفاده شده است. از 2400 تصویر آموزشی، ٪20 برای validation برای ارزیابی عملکرد شبکه پس از هر epoch استفاده شده. این شبکه بیش از 100 epoch و batch size=50 آموزش داده شده و نرخ یادگیری 0.01 برای تابع بهینه‌ساز Adam انتخاب شده است. پس از 100 epoch، پارامترهای مدل با بهترین عملکرد انتخاب و با تصاویر آزمایشی به منظور ارزیابی عملکرد کلی شبکه مورد استفاده قرار گرفته است.

۱-۲- معماری شبکه و مزایا و معایب آن

این شبکه دارای 5 مرحله کانولوشنی هستند که به شکل زیر ویژگی‌ها را استخراج می‌کنند:

Stage 1: conv-conv-pool

Stage 2: conv-conv-pool

Stage 3: conv-conv-pool

Stage 4: conv-conv-conv-conv-pool

Stage 5: conv-conv-conv-conv-pool

کل کانولوشن‌های موجود در شبکه‌ی VGG19 با اندازه‌ی فیلتر 3×3 و stride=1 و pad=1 هستند و لایه‌های max pooling با اندازه 2×2 و stride=2 هستند.

از **مزایای** این مدل نسبت به مدل‌های مشابه قبلی آن است که از انباشت کردن تعدادی لایه‌ی کانولوشن با فیلترهای کوچکتر استفاده کرده تا به receptive field مشابه با فیلترهای بزرگتر (با هزینه‌ی محاسباتی کمتر) برسد. از دیگر مزایای این شبکه، در دسترس بودن وزن‌های train شده‌ی این شبکه است که امکان استفاده از این شبکه برای transfer learning را فراهم می‌کند. همچنین این شبکه به علت **عمق بالایی** که دارد، می‌تواند به عنوان یک استخراج‌کننده‌ی ویژگی برای طبقه‌بندی با دقت بالا عمل کند.

شکل 2. **VGG19**

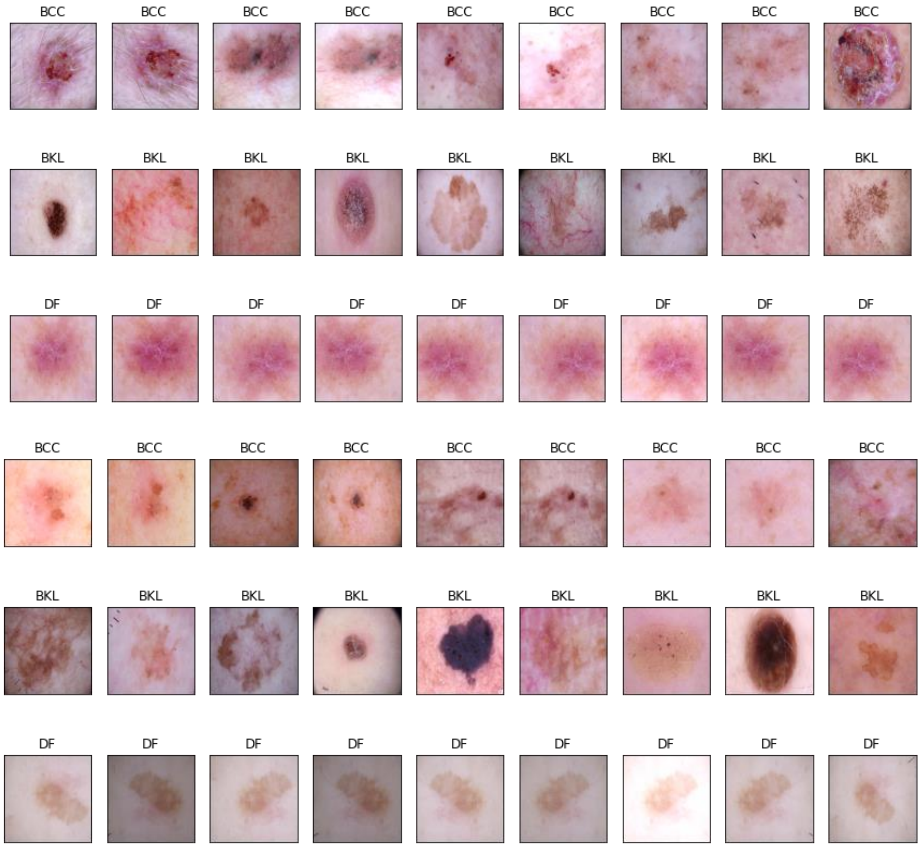
همچنین از معایب این مدل می‌توان به تعداد بسیار زیاد پارامترهای این شبکه و عمق بسیار زیاد این شبکه است که آموزش این مدل (از پایه) را سخت‌تر می‌کند. یافتن یک نقطه بهینه تابع loss برای این شبکه (با تعداد پارامتر زیاد) نیازمند داشتن تعداد داده‌های بسیار زیادی نیز می‌باشد که ممکن است در برخی مسائل جمع‌آوری داده هزینه‌بر یا وقت‌گیر باشد و فرآیند آموزش را سخت‌تر کند که معمولا برای حل این موضوع از تکنیک data augmentation استفاده می‌شود.

در این مقاله از روش‌های crop، scale، contrast adjustment، brightness adjustment، horizontal flip، vertical flip و ترکیب این روش‌ها استفاده شده. در پیاده‌سازی، ما از کتابخانه albumentation برای انجام این عملیات و augment کردن داده استفاده کردیم:

```python
import albumentations as A
import cv2
from matplotlib.image import imread
from matplotlib import pyplot as plt

transform = A.Compose([
    A.RandomCrop(width = 450, height = 450),
    A.RandomScale(scale_limit = 0.5),
    A.RandomBrightnessContrast(p =0.5),
    A.HorizontalFlip(p = 0.5),
    A.VerticalFlip(p = 0.5)
])
```

شکل**3** . کتابخانه‌ی **Albumentation** برای **data augmentation**

همانطور که قبلا بحث شد، تعداد داده‌های کلاس‌های انتخاب شده توسط مقاله متفاوت هستند که این موضوع باعث ایجاد بایاس در فرآیند آموزش مدل می‌شود. در نتیجه نیاز به انجام augmentation داریم تا این موضوع را حل کنیم.



شکل**4** . داده‌های ورودی

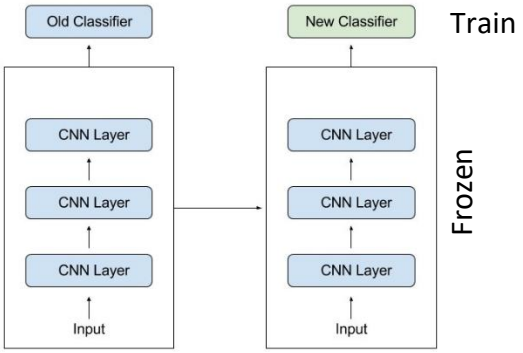
همانطور که در تصویر مشخص است، به دلیل کم بودن تعداد داده‌های کلاس DF از augmentation استفاده شده و نسخه‌های متفاوتی از یک تصویر ایجاد شده است.

۱-۳- قابلیت تشخیص شبکه

این شبکه روی مجموعه داده ImageNet آموزش داده شده است که دارای تصاویری از 1000 کلاس مختلف می‌باشد. این مدل با 144 میلیون پارامتر دارای دقت top-1 حدود 74.5% و دقت top-5 حدود 91% می‌باشد. از جمله کلاس‌های تصاویر موجود در این مجموعه داده می‌توان موارد زیر را نام برد:

- Pizza, pizza pie
- Strawberry
- Mushroom
- Jellyfish
- Persian Cat
- Guitar
- ...

حال اگر عکسی داخل دسته‌های موجود در ImageNet نباشد یعنی طبیعتا این مدل قادر به تشخیص آن نخواهد بود. در نتیجه باید از روش‌های جایگزین مانند Transfer Learning استفاده کنیم. یادگیری انتقالی (Transfer Learning) به معنای استفاده از یک مدل از پیش آموزش دیده در یک کاربرد جدید است. این مبحث، امروزه در یادگیری عمیق بسیار مورد توجه است، زیرا امکان آموزش شبکه‌های عصبی عمیق را با داده‌های نسبتا کمی فراهم می‌کند. هدف از یادگیری انتقالی در واقع این است که از دانشی که در یک مسئله به دست آمده (یعنی توانایی استخراج ویژگی آن) برای بهبود تعمیم‌پذیری در مسئله‌ای دیگر استفاده شود. یعنی به جای شروع پروسه آموزش از صفر، از الگوهای بدست آمده در مسئله‌ای مشابه استفاده شود.



شکل **5** . **Transfer Learning**

۱-۴- لود دیتاست و کار کردن با آن

```
1 ! kaggle datasets download -d umangjpatel/ham10000-imagenet-style-dataset

Downloading ham10000-imagenet-style-dataset.zip to /content
100% 2.58G/2.58G [01:30<00:00, 36.5MB/s]
```

شکل 6. دانلود دیتاست از **kaggle**

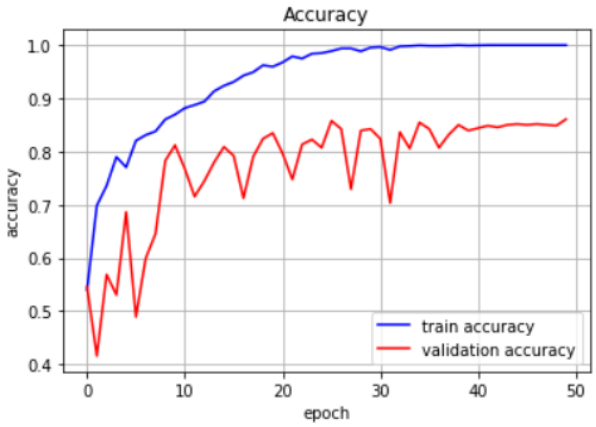
```
image_generator = ImageDataGenerator(rescale = 1./255., validation_split=0.2)

train = image_generator.flow_from_directory(batch_size=50,
                                            directory='/content/ham10000-imagenet-style-dataset-transformed',
                                            shuffle=True,
                                            subset="training",
                                            target_size=(64, 64),
                                            class_mode='categorical',
                                            classes = ["bcc", "bkl", "df"],
                                            seed = 42)
```

شکل 7. استفاده از **ImageDataGenerator** و متد **flow_from_directory**

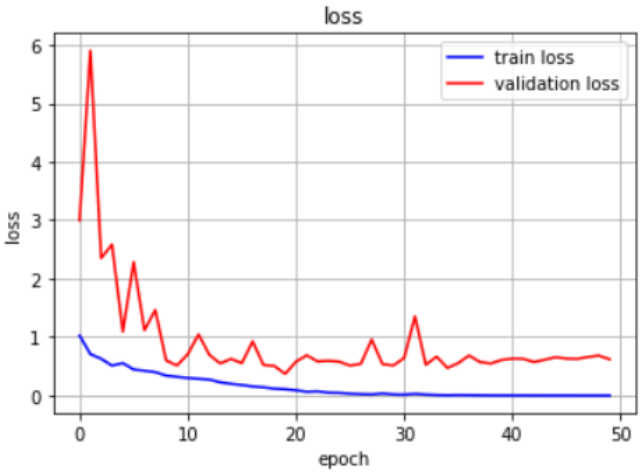
۱-۵- پیاده‌سازی شبکه و گزارش نتایج

```
base_model = VGG19(input_shape=(64,64,3), weights='imagenet', include_top=False, pooling = max)
model=Sequential()
model.add(base_model)
model.add(Dropout(0.2))
model.add(Flatten())
model.add(tf.keras.layers.BatchNormalization())

model.add(Dense(3,kernel_initializer='he_uniform'))
model.add(Activation('softmax'))

optimizer = optimizers.Adam(learning_rate=0.01,epsilon = 1e-7)

model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
```

شکل 8. پیاده سازی مدل بر اساس **VGG19**



شکل 9. دقت شبکه



شکل 10. **Loss** شبکه

```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred,target_names=["bcc", "bkl", "df"]))
```

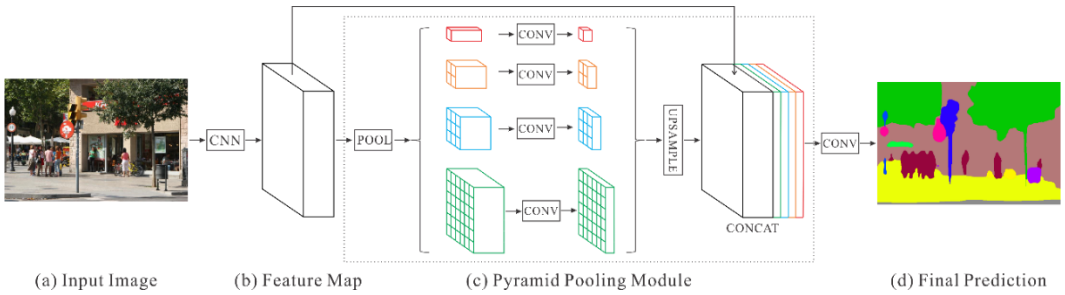
	precision	recall	f1-score	support
bcc	0.80	0.80	0.80	205
bkl	0.91	0.96	0.94	219
df	0.87	0.82	0.84	230
accuracy			0.86	654
macro avg	0.86	0.86	0.86	654
weighted avg	0.86	0.86	0.86	654

شکل 11. خلاصه عملکرد شبکه

پاسخ ۲ – آشنایی با تشخیص چهره مسدود شده

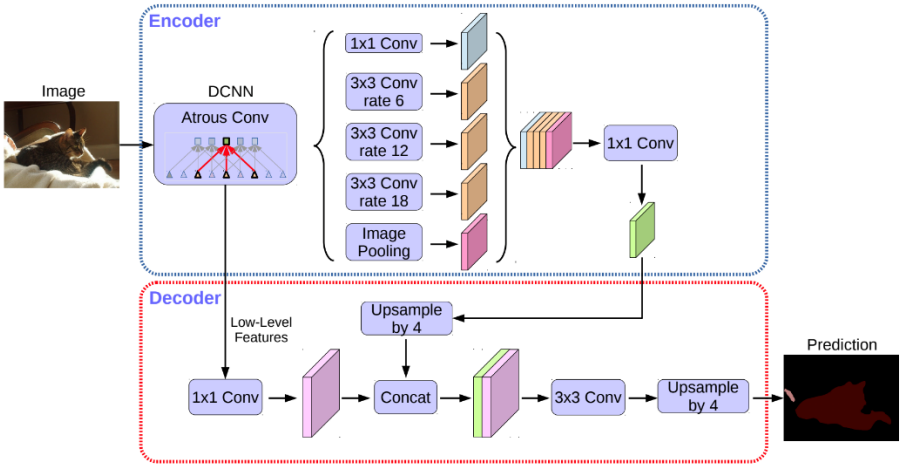
۲-۱- خلاصه‌ی ساختار شبکه

در این مقاله از ۳ شبکه مختلف برای segmentation استفاده شده. شبکه‌های مبتنی بر CNN از جمله PSPNet و DeepLabv3+ با پایه‌ی ResNet-101 و شبکه‌ی ترنسفورمر SegFormer با پایه‌ی MIT-B5 در این مقاله استفاده شده است.



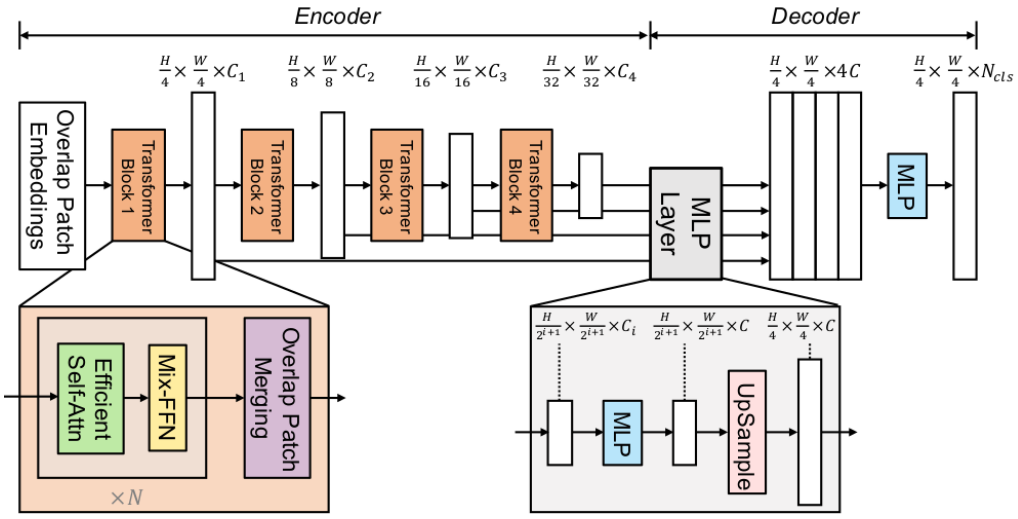
شکل 12. شبکه‌ی **PSPNet**

در این معماری با توجه به یک تصویر ورودی (a)، ابتدا از CNN برای بدست آوردن و استخراج ویژگی (b) استفاده می کنیم، سپس خروجی آخرین لایه کانولوشن وارد یک ماژول تجزیه هرمی برای به دست آوردن نمایش‌های مختلف زیرمنطقه‌های مختلف می‌شود، سپس لایه‌های upsampling و concatenation برای تشکیل لایه‌ها اعمال می‌شود. ویژگی‌های نهایی، که اطلاعات زمینه محلی (local) و کلی (global) را در (c) حمل می‌کنند، وارد یک لایه کانولوشن می‌شود تا پیش‌بینی نهایی هر پیکسل (d) به دست آید و segmentation تصویر انجام شود.



شکل**13** . شبکه‌ی **DeepLabv3+**

شبکه‌ی DeepLabv3+ یک شبکه‌ی fully-convolutional است و یک ساختار encoder-decoder دارد. شبکه‌ی Encoder با استفاده از dilated convolution با rate های مختلف سعی در استخراج ویژگی‌های contextual دارد. در واقع در این شبکه از Dilated Spatial Pyramid Pooling استفاده شده که field-of-view یا receptive field شبکه را (بدون افزایش غیرمنطقی تعداد پارامترها و مقدار stride) افزایش می‌دهد. همچنین شبکه‌ی Decoder با استفاده از ویژگی‌های استخراج شده از مدل پایه و مدل پیشنهادی، سعی در پیش‌بینی پیکسل‌ها و انجام semantic segmentation دارد.



شکل 14. شبکه‌ی **SegFormer**

شبکه‌ی SegFormer از یک Encoder سلسله مراتبی برای استخراج ویژگی های کلی و جزئی تشکیل شده است، و یک Decoder سبک وزن All-MLP برای ترکیب مستقیم این ویژگی‌های چندسطحی و پیش‌بینی کلاس تقسیم‌بندی معنایی برای هر پیکسل استفاده شده است.

۲-۲- تفاوت بین Occlusionهای مختلف

تقسیم‌بندی تصاویر مربوط به صورت، در تسک‌های مختلفی مانند face recognition، face-swapping و facial reconstruction استفاده می‌شود و آموزش مدل‌های با دقت بالا برای segmentation بین صورت و پس‌زمینه دارای اهمیت ویژه‌ای می‌باشد. مجموعه‌های داده مختلف برای آموزش مدل‌ها وجود دارند که از روش‌های مختلفی برای Occlusion تصاویر صورت استفاده می‌کنند. در واقعیت تصاویر صورت انسان می‌توانند با اشیاء مختلفی مانند عینک، ماسک، دست و ... پوشش داده شوند که باید تصاویر اینگونه Occlusionها در آموزش مدل‌های real-world، استفاده شوند تا دقت و robustness کافی داشته باشند.

چند روش مختلف برای ایجاد Occlusion روی تصاویر صورت وجود دارد. روش اول استفاده از تصاویر واقعی است که دارای پوشش یا ماسک یا هرگونه شیء اضافه جلوی تصویر صورت باشد. طبیعتا

استخراج یا ساختن مجموعه داده‌ای که واقعا چنین باشد هزینه‌بر و زمانبر خواهد بود. روش دوم استفاده از متدهای data augmentation بر روی تصاویر واقعی و بدون occlusion صورت و اضافه کردن دستی یک occlusion به صورت تصادفی (random) روی آن‌هاست. این روش با اینکه روشی ساده و قابل پیاده‌سازی است ولی تصاویر تولید شده با استفاده از این روش با واقعیت فاصله دارد. به طور مثال اضافه کردن یک مستطیل (یا هر شکل چندوجهی با رنگ‌های مختلف) به تصویر صورت یا اضافه کردن قسمت‌هایی از برخی تصاویر تصادفی (مانند تصاویر مجموعه داده DTD)، در واقعیت اتفاق نمی‌افتند و مدل با استفاده از برخی داده‌های غیرواقعی آموزش داده می‌شود.

روش سوم استفاده از روش‌های augmentation واقعی و طبیعی‌تر است. به طور مثال به جای اضافه کردن برخی اشکال نامربوط، می‌توانیم تصاویری مانند عینک یا ماسک یا دست را به تصویر صورت اضافه کنیم. البته نکته‌ای که وجود دارد آن است که به طور مثال قرار گرفتن دستی با رنگ پوست تیره جلوی صورت یک شخص با رنگ پوست روشن، اتفاقی نادر است و معمولا رنگ اعضای بدن تصویر با یکدیگر شباهت دارند و این روش هم جای بهبود دارد.

در نهایت می‌توان گفت که augmentation های natural دقت شبکه بالاتری نسبت به روش‌های دیگر از جمله augmentation های تصادفی و یا غیرطبیعی دارند و استفاده از روش درست برای دستیابی به generalization مناسب در کاربردهای real-world اهمیت بالایی دارد.

۲-۳- کلاس‌بندی کردن داده‌ها

تقسیم کردن داده‌های موجود در مجموعه داده و کلاس‌بندی کردن آنان از اهمیت ویژه‌ای برخوردار است. روش پیش‌گرفته شده در مقاله به صورت زیر است:

ابتدا مجموعه داده‌ی CelebAMask-HQ (که دارای ۳۰۰۰۰ عکس مختلف صورت است) به دو قسمت occluded و non-occluded تقسیم می‌شود. کلاس occluded شامل تصاویری است که شیءای جلوی تصویر صورت را گرفته یا با آن هم‌پوشانی داشته باشد. کلاس non-occluded نیز شامل تصاویر مربوط به صورت است که هیچ‌گونه occlusionای روی بخش صورت تصاویر وجود ندارد.

جدول 1. کلاس‌بندی داده‌های **CelebAMask-HQ**

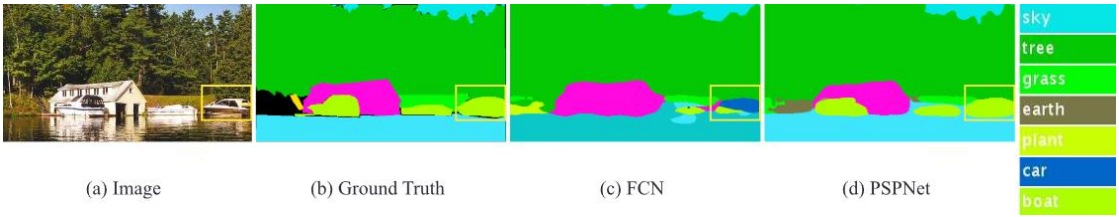
Category	Quantity
CelebAMask-HQ-WO (Train)	24603
CelebAMask-HQ-WO (Test)	716
CelebAMask-HQ-O	4597
Excluded Images	86
WO - Without occlusion | O - Occluded	

۲-۴- تفاوت intensity چهره‌ها با Occlusionهای مصنوعی

اگر intensity چهره‌ها با occlusionهای مصنوعی متفاوت باشد، این به معنای آن است که داده حالت غیرطبیعی دارد و احتمالا در دنیای واقعی اتفاق نمی‌افتد. پس باید از مدلی استفاده کنیم که اولا بیشتر از ویژگی‌های محلی به ویژگی‌های کلی تصویر توجه کند و ثانیا قابلیت تعمیم و generalization بیشتری داشته باشد. بنابراین علاوه بر PSPNet و DeepLabv3+ می‌توان از شبکه‌های ParseNet و MaskLab از مدل‌های مطالعه شده در درس استفاده کرد. زیرا این شبکه‌ها با میانگین‌گیری از feature map های هر لایه قابلیت تشخیص ویژگی‌های global را دارند.

۲-۵- مقایسه PSPNet و DeepLabv3+

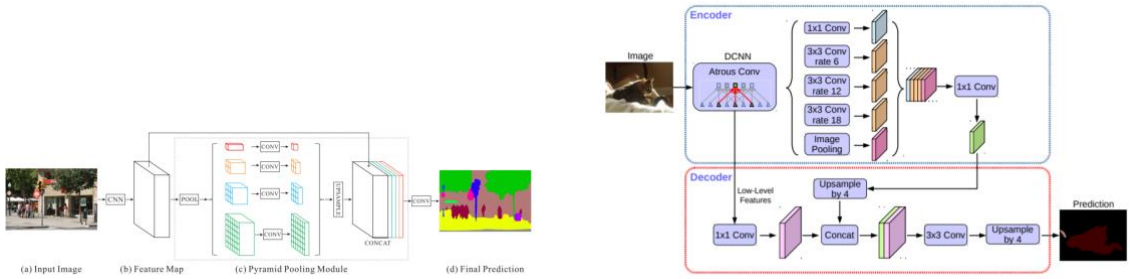
اکثر شبکه‌های مربوط به Semantic Segmentation از یک ساختار Encoder-Decoder استفاده می‌کنند که در این ساختار وظیفه‌ی انکودر استخراج ویژگی از تصویر و وظیفه‌ی دیکودر پیش‌بینی دسته‌ی پیکسل مربوطه می‌باشد. شبکه‌ی PSPNet از dilated convolution و pyramid pooling استفاده کرده و نسخه‌ی بهبودیافته‌ی شبکه‌ی Fully Convolutional Network (FCN) است. این شبکه از ساختار و ویژگی‌های کلی یا Global تصویر برای پیش‌بینی‌های محلی برای تصاویر استفاده می‌کند که منجر به افزایش دقت و کارایی شبکه می‌شود (مدل‌های FCN قادر به تشخیص context کلی تصویر و ویژگی‌های Global نبودند).



شکل 15. مقایسه‌ی **FCN** و **PSPNet**

معمولا روش‌های مبتنی بر شبکه‌های عصبی کانولوشن نیازمند یک تصویر ورودی با اندازه ثابت هستند. این محدودیت ممکن است باعث کاهش دقت تشخیص برای تصاویری با اندازه دلخواه شود. به منظور حذف این محدودیت از یک معماری شبکه عصبی کانولوشن معمولی استفاده می‌شود با این تفاوت که لایه pooling آخر با یک لایه spatial pyramid pooling جایگزین می‌شود. این لایه قادر به استخراج نمادها (تصاویر) با اندازه ثابت از تصاویر (یا نواحی) دلخواه است. این روش باعث ایجاد یک راه حل قابل انعطاف برای مدیریت مقیاس‌ها, اندازه‌ها و aspect ratio های مختلف می‌شود که می‌توان از آن در هر ساختار و معماری CNNی استفاده کرده و کارایی آن را افزایش داد.

همانطور که اشاره شد شبکه PSPNet، اطلاعات زمینه global را استخراج و تجمیع کرد و کیفیت بخش‌بندی را بدون استفاده از روش‌های post-processing پرهزینه مانند CRF بهبود بخشید. DeepLabv3 تلاش کرد تا با استفاده از نرخ‌های اتساع (dilation) چندگانه در ماژول‌های پشت هم قرارگرفته (cascaded) و DSPP که از dilated convolution استفاده می‌کردند، زمینه چند مقیاسی (multi-scale context) را استخراج کند. علاوه بر این، DeepLabv3+ از معماری مشابه DeepLabv3 استفاده کرد، اما استفاده از یک شبکه Decoder را برای **بهبود دقت تقسیم‌بندی در اطراف لبه‌ها** پیشنهاد کرد. در DeepLabv3+، لایه‌های کانولوشن قابل تفکیک عمقی[1] هم در ماژول DSPP و هم در شبکه Decoder مورد استفاده قرار گرفتند و طبق گزارش‌ها عملکرد محاسباتی را بهبود بخشیدند.



شکل 16. مقایسه‌ی شبکه‌های **DeepLabv3+** و **PSPNet**

Method	Structure	Backbone	LRP	Loss
FCN-32s	-	VGG-16	fixed	Cross Entropy
FCN-16s	Multi-Scale	VGG-16	fixed	Cross Entropy
FCN-8s	Multi-Scale	VGG-16	fixed	Cross Entropy
U-Net	Encoder-Decoder	VGG-16	step	Cross Entropy
SegNet	Encoder-Decoder	VGG-16	step	Cross Entropy
DeepLab	Multi-Scale	VGG-16	poly	Cross Entropy
DeepLab	Multi-Scale	ResNet-101	poly	Cross Entropy
PSPNet	Multi-Scale	VGG-16	poly	Cross Entropy
PSPNet	Multi-Scale	ResNet-101	poly	Auxiliary Loss

شکل 17. مقایسه‌ی ساختار و **Loss** های شبکه‌های مختلف

[1] depthwise separable convolutional layers

پاسخ ۳ – تشخیص بلادرنگ اشیاء (YOLOv6)

۳-۱- نحوه شخصی‌سازی یک مجموعه داده جدید روی YOLOv6

برای این کار ابتدا باید داده‌ها در پوشه‌بندی مناسب قرار داده شوند. تصاویر باید در پوشه‌ی images و برچسب‌ها در پوشه‌ی labels قرار داده شوند. همچنین یک فایل data.yaml باید در کنار این پوشه‌ها قرار بگیرد تا configuration داده‌ها را در خود داشته باشد. محتوای این فایل مانند شکل زیر است:

```
train: ./images/train
val: ./images/valid
test: ./images/test
nc: 13
names: ['bishop', 'black-bishop', 'black-king', 'black-knight', 'black-pawn', 'black-queen', 'black-rook', 'white-bishop', 'white-king', 'white-knight', 'white-pawn', 'white-queen', 'white-rook']
```

همانطور که مشخص است در پوشه‌ی images نیز ۳ پوشه‌ی train، validation و test وجود دارد که برچسب معادل آنها نیز در همین پوشه‌ها در labels قرار گرفته است.

شکل 18. یکی از تصاویر موجود در مجموعه‌ی تست

برچسب این داده نیز به صورت زیر است:

```
7 0.5120192307692 0.0829326923076 0.04567307692307 0.1334134615384
```

که نماینده مرکز سگمنت (مختصات مرکز) و طول و عرض مستطیل Bounding Box و کلاس مربوط به این سگمنت می‌باشد.

شکل 19. تصویر **segment** شده بر اساس مدل آموزش دیده

برای آموزش دادن این مدل نیز باید از وزن‌های pre-trained مربوط به yolov6s.pt استفاده کرده و برای fine-tune کردن آن از دستور زیر استفاده می‌کنیم:

```
1 !python tools/train.py --batch 32 --conf configs/yolov6s_finetune.py --data-path /content/drive/MyDrive/Q3/data.yaml --device 0 --epochs 50

Using 1 GPU for training...
```

شکل 20. دستور مربوط به **fine-tune** مدل **YOLOv6**

```
!python tools/infer.py --weights runs/train/exp/weights/best_ckpt.pt --source /content/drive/MyDrive/Q3/images/test/ --yaml /content/drive/MyDrive/Q3/data.yaml --device 0
```

شکل 21. دستور مربوط به **infer** کردن از مدل **fine-tune** شده

شکل 22. یکی از عکس‌های سگمنت شده توسط مدل

همچنین فایل‌های مربوط به تصاویر سگمنت شده در کنار فایل‌ها در ایلرن آپلود شده است.