به نام خدا

مبانی بینایی کامپیوتر (دکتر مهدی سیفی پور)

طراح : امیر نداف فهمیده

تمرین سوم

نیم‌سال دوم **۱۴۰۵-۱۴۰۴**

مقدمه

در سال‌های اخیر، مدل‌های از پیش‌آموزش‌دیده (Pre-trained Models) به یکی از ارکان اصلی بینایی ماشین تبدیل شده‌اند. این مدل‌ها با بهره‌گیری از میلیون‌ها پارامتر و آموزش روی مجموعه‌داده‌های عظیم، توانایی چشمگیری در استخراج ویژگی‌های بصری عمومی دارند. نکته قابل توجه آن است که بسیاری از این مدل‌ها، حتی بدون هیچ‌گونه آموزش مجدد (Fine-tuning)، قادرند روی تصاویر کاملاً جدید و در کاربردهای متنوع، عملکرد قابل قبولی از خود نشان دهند، به شرط آنکه پیش‌پردازش صحیح و متناسب با معماری آن‌ها صورت گیرد.

در این پروژه، از همین قابلیت بهره می‌بریم: شما با بارگذاری وزن‌های از پیش‌آموزش‌دیده و اعمال دقیق مراحل پیش‌پردازش، وظایف کلیدی و متفاوت در بینایی ماشین را روی تصاویر واقعی پیاده‌سازی خواهید کرد.

هدف اصلی این تمرین، فراتر از پیاده‌سازی فنی، درک عمیق رفتار این مدل‌ها در شرایط واقعی است: پیش‌پردازش‌های مختلف (نظیر محو کردن، تغییر ابعاد، تبدیل فضای رنگی) چگونه بر خروجی تأثیر می‌گذارند؟ مدل‌ها در برابر چرخش، نویز، تغییر روشنایی و مات‌شدگی چه واکنشی نشان می‌دهند؟ کجا موفق می‌شوند و کجا دچار خطا می‌گردند؟ و در نهایت، چگونه می‌توان از این ابزارها در یک خط لوله عملی و ترکیبی بهره گرفت.

بخش اول: تشخیص چهره‌ی گربه

در این بخش از تمرین با یکی از روش‌های کلاسیک و سریع تشخیص اشیا آشنا می‌شوید یعنی آشکارساز Haar Cascade. مدل از پیش‌آموزش‌دیده‌ای برای تشخیص چهره‌ی گربه در اختیار شماست.

شما باید آن را بارگذاری کرده، فرآیند پیش‌پردازش مناسب را بیاموزید و سپس با دستکاری تصاویر ورودی تأثیر این تغییرات را بر خروجی مدل تحلیل کنید.

شما فایل مدل haarcascade_frontalcatface_extended.xml را به همراه چند تصویر از گربه‌ها در اختیار دارید. مراحل زیر را برای انجام این بخش دنبال کنید.

۱. ابتدا تصاویر ورودی را بخوانید و نمایش دهید.
۲. آشکارساز Haar را با بارگذاری کنید. برای این کار می‌توانید از cv2.CascadeClassifier استفاده کنید.
۳. تابع detectMultiScale را با پارامترهای scaleFactor=1.1, minNeighbors=5, minSize=(50,50) فراخوانی کرده و مستطیل‌های تشخیص را روی تصویر بکشید. برای هر گربه‌ی پیداشده، متن Cat را با فونت و مکان مناسب بنویسید. (برای اینکه تشخیص دقیق تر باشد میتوان از محو کردن (blur) تصویر هم استفاده کنید.)
۴. بعد از اطمینان از صحت عملکرد مدل یک بار تصاویر را به صورت خاکستری (grayscale) و یک بار به صورت رنگی به مدل بدهید و نتایج را تحلیل کنید.
۵. برای تست مقاومت مدل نسبت به چرخش، این بار تصاویر را ۹۰ درجه بچرخانید (rotate) و به مدل بدهید سپس خروجی را تحلیل کنید.

بخش دوم: تشخیص چهره، سن و جنسیت

در این بخش با سه مدل از پیش‌آموزش‌دیده کار می‌کنید. یک مدل تشخیص چهره (Face Detection)، یک مدل تخمین سن (Age Prediction) و یک مدل تشخیص جنسیت (Gender Prediction). فایل‌های وزن و ساختار شبکه (pb. و pbtxt.) در اختیار شماست. باید خودتان آن‌ها را بارگذاری کرده و پیش‌پردازش‌های دقیق که تأثیر زیادی بر عملکرد دارند را پیاده‌سازی کنید و سپس با دستکاری هدفمند این پیش‌پردازش‌ها، اثر آن‌ها را تحلیل نمایید.

فایل های زیر در اختیار شما قرار داده شده است.

- وزن های opencv_face_detector.pbtxt و opencv_face_detector_uint8.pb (مدل تشخیص چهره)
- وزن های age_deploy.prototxt و age_net.caffemodel (مدل تخمین سن)
- وزن هایgender_deploy.prototxt و gender_net.caffemodel (مدل تشخیص جنسیت)
- به‌همراه چند تصویر تست.

مراحل انجام این بخش را به صورت زیر دنبال کنید.

۱. هر سه مدل را با cv2.dnn.readNet بارگذاری کنید.
۲. لیست‌های زیر را عیناً تعریف کنید:

```
ageList = ["(0-2)", "(4-6)", "(8-12)", "(15-20)", "(25-32)", "(38-43)", "(48-53)", "(60-100)"]
genderList = ["Male", "Female"]
MODEL_MEAN_VALUES = (78.4263377603, 87.7689143744, 114.895847746)
```

۳. تابع highlightFace را پیاده‌سازی کنید به طوری که:
- با cv2.dnn.blobFromImage یک blob با scale=1.0 ، size=(300,300) ، mean=(104,117,123) و swapRB=True بسازد. پس از تنظیم ورودی و اجرای forward، تشخیص‌هایی با اطمینان بالای conf_threshold=0.7 را نگه دارد و کادر آن‌ها را روی تصویر رسم کند.

۴. تابع detect_face_age_gender را پیاده‌سازی کنید به طوری که:
- ابتدا با highlightFace چهره‌ها را بیابد.
- برای هر چهره، ناحیه صورت را با padding=20 پیکسل جدا کند.
- یک blob جدید با size=(227,227)، mean=MODEL_MEAN_VALUES و swapRB=False بسازد و به‌ترتیب به مدل‌های جنسیت و سن بدهد. برچسب "Gender, Age" را بالای کادر رسم کند.

۵. تصاویر را به مدل بدهید و عملکرد مدل را بررسی کنید.

۶. بعد از تست مدل، تمامی آزمایش‌های زیر را روی تمامی تصاویر انجام دهید و نتایج را تحلیل کنید:

الف) چرخش ۹۰ درجه: تصویر را ۹۰ درجه بچرخانید و خروجی تشخیص چهره و سن/جنسیت را بررسی کنید.

ب) تغییر روشنایی: پیش از blob اول، روشنایی تصویر را با ضرب در ۰٫۵ و سپس ۱٫۸ تغییر دهید. خروجی را بررسی کنید.

ج) محو کردن عمدی صورت: پیش از ارسال به مدل سن/جنسیت، ناحیه صورت را با فیلتر گوسین ۵۱×۵۱ محو کنید. آیا سن و جنسیت هنوز درست تشخیص داده می‌شود؟

۷. در آخر یک اسکریپت بنویسید که یک ویدئوی ورودی را فریم به فریم پردازش کرده و روی هر فریم تشخیص چهره، سن و جنسیت را اجرا کند و ویدئوی خروجی را همراه با نمایش FPS تولید کند.

بخش سوم: رنگی‌سازی تصاویر

در این بخش با یک شبکه عصبی پیچشی (CNN) از پیش‌آموزش‌دیده کار می‌کنید که تصاویر خاکستری (grayscale) را به نسخه رنگی طبیعی تبدیل می‌کند.

فایل وزن‌های مدل (colorize_autoencoder_first1000.keras) و چند تصویر از دسته‌های طبیعت و ساختمان برای شما قرار داده شده است.

توضیح معماری مدل در ادامه آورده شده است ولی پیاده‌سازی آن با شماست.

مدل یک Autoencoder کانولوشنی با ساختار Encoder-Decoder است:

- ورودی: تک‌کاناله با ابعاد ۲۵۶×۲۵۶ (مقادیر ۰ تا ۱۰۰، مربوط به کانال L در فضای رنگ Lab)
- خروجی: دو کاناله (کانال‌های a و b در فضای Lab) با تابع فعال‌سازی tanh در لایه نهایی
- فشرده‌سازی با MaxPooling2D و بازسازی با UpSampling2D انجام می‌شود.

۱. ابتدا مدل را با دستور زیر لود کنید و معماری آن را مشاهده کنید.

```
model = tf.keras.models.load_model(MODEL_PATH)
model.summary()
```

۲. برای پیش پردازش باید توجه داشته باشید که هدف مدل این است که تصاویر خاکستری را به تصاویر رنگی تبدیل کند اما برای مقایسه، شما تصاویر رنگی اصلی را در اختیار دارید. پس ابتدا تمامی تصاویر را خاکستری کنید، سپس چون مدل تنها کانال روشنایی (L) از فضای رنگ Lab را می‌پذیرد نیاز دارید که تصاویر را به این فضای رنگ انتقال دهید. برای تبدیل یک تصویر خاکستری به این فرمت، دو روش اصلی وجود دارد:

- روش اول: تصویر تک‌کاناله خاکستری را سه بار در بعد کانال تکرار کنید تا یک تصویر سه‌کاناله RGB بدست آید. سپس با استفاده از تابع rgb2lab (از کتابخانه skimage.color) آن را به فضای رنگ Lab ببرید و تنها کانال L (روشنایی) را استخراج کنید.
- روش دوم: تصویر خاکستری را با استفاده از cv2.imread و پرچم cv2.IMREAD_COLOR بارگذاری کنید (OpenCV به‌طور خودکار آن را سه‌کاناله می‌کند)، سپس مشابه روش اول، تصویر را به کمک rgb2lab به فضای Lab تبدیل کرده و کانال L را جدا کنید.

شما باید هر دو روش را روی چهار تصویر پیاده‌سازی کرده و کانال L را با ابعاد 256×256×1 آماده کنید.

۳. برای تست مدل، کانال L را به مدل بدهید و خروجی مدل (دو کانال a و b) را دریافت کنید. توجه کنید که مقادیر a و b در خروجی مدل در بازه ۱- تا ۱ قرار دارند. کانال L اصلی را با a و b پیش‌بینی‌شده ترکیب کرده و به فضای RGB برگردانید. تصویر رنگی نهایی را نمایش دهید.
برای هر تصویر، خروجی روش اول و دوم را کنار یکدیگر قرار داده و تحلیل کنید کدام روش خروجی طبیعی‌تری تولید می‌کند و چرا؟

۴. برای سنجش پایداری مدل، آزمایش‌های زیر را فقط با روش اول (تکرار سه‌باره کانال خاکستری) انجام دهید.

الف) چرخش تصویر(Rotation): تصویر ورودی را ۴۵ درجه بچرخانید، سپس مراحل پیش‌پردازش (تبدیل به Lab و استخراج L) را انجام داده و به مدل بدهید. در نهایت خروجی رنگی را مجدداً ۴۵ درجه در خلاف جهت بچرخانید تا با تصویر اصلی هم‌راستا شود و آن را با خروجی حالت بدون چرخش مقایسه کنید.

ب) نویز گوسی (Gaussian Noise): پس از استخراج کانال L و پیش از نرمال‌سازی (مقادیر ۰ تا ۱۰۰)، نویز گوسی با انحراف معیار ۰.۱ اضافه کنید. با توجه به خروجی‌ها کدام نواحی تصویر بیشتر تحت تأثیر قرار گرفته‌اند؟

ج) تغییر روشنایی (Brightness): کانال L را در ۰.۷ ضرب کرده و به مدل بدهید. همین کار را با ضرب در ۱.۳ (افزایش روشنایی) تکرار کنید. با توجه به خروجی‌ها آیا رنگ‌های خروجی متناسب با تغییر روشنایی تغییر می‌کنند؟

نکات نهایی

۱. ددلاین تمرین ۱ خرداد ۱۴۰۵ است.
۲. تمرین دارای ارائه می‌باشد و باید آمادگی ارائه آن را داشته باشید.
۳. فایل کد، فایل HTML آن، ویدیو خروجی بخش دوم و گزارش‌کار را در پوشه‌ای به نام CV-HW3-std#1-std#2 در سایت آپلود کنید. نیازی به آپلود وزن‌ها نیست.
۴. نوشتن گزارش‌کار الزامی است و می‌تواند در Jupyter Notebook نیز نوشته شود.
۵. در صورت پیدا شدن موارد تقلب، نمره تمرین صفر شده و همچنین نمره منفی برای تمارین بعدی لحاظ می‌شود.
۶. شما مجاز به استفاده از کتابخانه‌های رایج پایتون مانند OpenCV هستید.
۷. دقت کنید که تمامی نتایج موارد خواسته‌شده باید در گزارش‌کار یا فایل HTML و یا در فایل گزارش آورده شده باشد.

موفق باشید