به نام خدا

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

درس سیستم‌های هوشمند

پروژه پایانی

بینایی ماشین

بهمن ماه 1401

فهرست مطالب

قسمت اول: بازسازی تصویر از نقشه‌های ویژگی

همان طور که میدانیم در یادگیری ماشین، در هر مساله، بازنمایی هایی از داده ها مناسب است که ویژگی‌های زائد از آنها حذف و اطلاعات و ویژگی های مفید برای استنتاج از داده ها حفظ شده باشد. در بسیاری از مسائل، اثبات شده است که شبکه های عصبی کانولوشنی میتوانند بازنمایی‌هایی با شرط گفته شده را انجام دهند. در این مساله با پیاده‌سازی یک مدل ساده صحت این امر را متوجه خواهید شد و میبینید که کدام ویژگی ها از یک تصویر توسط مدل کانولوشنی حذف و کدام ویژگی ها حفظ میشوند.

بخش الف)

در این بخش لازم است مدل هایی را برای بازسازی تصویر از نقشه های ویژگی استخراج شده از لایه های مختلف یک شبکه کانولوشنی آموزش دهید. مبنای این سوال این مقاله است. در این مقاله نویسندگان ساختار مدل AlexNet که بر روی دیتاست ImageNet برای مساله طبقه بندی تصاویر آموزش دیده است را در نظر میگیرند که ساختار آن در جدول ۳ از مقاله آمده است. سپس با استفاده از بردار های ویژگی خروجی لایه های مختلف آن، قصد دارند تا تصاویر اولیه را بازسازی کنند.

در این تمرین به جای استفاده از دیتاست ImageNet از دیتاست Tiny ImageNet استفاده میکنید. از میان ۲۰۰ کلاس موجود در دیتاست ۲۰ کلاس با به صورت تصادفی انتخاب کنید و شناسه آنها را در گزارش خود بیاورید. از هر کلاس ۵۰ تصویر را به عنوان داده ارزیابی و باقی را برای آموزش مدل استفاده کنید. دقت کنید که مدل را با نمونه گیری یکنواخت از تمام کلاس ها آموزش دهید (هم برای آموزش هم ارزیابی). برای صرفه جویی در مصرف اینترنت از دستور زیر برای دانلود دیتاست در colab استفاده کنید.

```
wget http://cs231n.stanford.edu/tiny-imagenet-200.zip
```

شما لازم است تا از میان ۸ لایه مختلف AlexNet مدل هایی را برای بازسازی تصویر از خروجی سه لایه conv2, conv5, fc8, اموزش دهید. برای خروجی لای های کانولوشنی از ساختار معرفی شده در جدول ۷ و برای خروجی لایه های خطی از ساختار معرفی شده در جدول ۸ بهره بگیرید (چهار مدل مجزا آموزش دهید). برای کاهش محاسبات و حافظه عمق شبکه ها کانولوشنی را به نصف تقلیل دهید. دقت کنید که وزن های شبکه AlexNet را در زمان آموزش مدل های کدگشا ثابت در نظر بگیرید و آموزش ندهید. از مدل از پیش آموزش دیده AlexNet در پکیج tochvision استفاده کنید. جزئیات و هایپرپارامترهای[1] آموزش مدل ها را از بخش ۲,۲ بگیرید و در صورت نیاز آنها را تغییر دهید. همانطور که در مقاله ذکر شده است از تابع هزینه L2 برای آموزش استفاده کنید.

---

[1] Hyper parameter

بخش ب)

نمودار هزینه بر اساس تکرار را برای دادگان اموزش و ارزیابی در طول آموزش رسم کنید و آنها را بررسی کنید.

بخش ج)

۵ کلاس از ۲۰ کلاس موجود را انتخاب کنید و از هر کدام ۱ تصویر از دادگان آموزش و ۱ تصویر از دادگان ارزیابی را انتخاب کنید و خود تصویر ها را به همراه تصاویر بازسازی شده آنها، از خروجی هر سه لایه را، رسم کنید.

بخش د)

با توجه به تصاویر بخش ب، ویژگی هایی که توسط یک شبکه عصبی کانولوشنی در هر لایه حذف یا حفظ میشوند را تحلیل کنید. در این تحلیل لایه های مختلف را هم با هم بررسی کنید و نتیجه بگیرید. میتوانید با مطالعه مقاله تحلیل های خود را قوی تر و علمی تر کنید.

قسمت دوم: تشخیص و شناسایی اشیا در ویدیو

یکی از کاربردی ترین وظایف بینایی ماشین[۱] در شناسایی اشیاء[۲] می‌باشد که توسط مدل‌های مدرن[۳] YOLO[۴] و SSD[5] به نتایج بسیار خوبی رسیده است. (در این لینک می‌توانید ویدیو مختصری از آخرین مدل YOLO منتشر شده (YOLOv8) را مشاهده کنید). در این قسمت قصد داریم که به کمک مدل YOLOv3، به شناسایی اشیاء در یک ویدیو دلخواه ورودی که در فایل پروژه به اسم "yolo_test.mp4" قرار دارد، بپردازیم. هدف این بخش از پروژه آشنایی شما با سیر تکاملی مدل های بینایی ماشین در طی زمان، کار با کتابخانه "open cv" و همچنین نحوه پیاده سازی YOLO به کمک این کتابخانه می‌باشد.

الف) تاریخچه

اگر به تاریخچه بینایی ماشین نگاهی بندازیم، تا سال 2005 دو الگوریتم مختص به شناسایی چهره[۶] معرفی شده بود (الگوریتم Violo-Jones که در سال 2001 و الگوریتم HOG[7] در سال 2005 مطرح شد). پس از این دو الگوریتم، با آمدن شبکه‌های کانولوشنی[۸]، انقلاب اساسی در این زمینه با معرفی مدل های R-CNNs[۹]، Fast-CNNs و Faster-CNNs در سال 2015 بوجود آمد که امکان شناسایی اشیاء که تنها منحصر به چهره نبود را فراهم آورد. در نهایت در سال 2016 اولین ورژن[۱۰] YOLO معرفی شد.

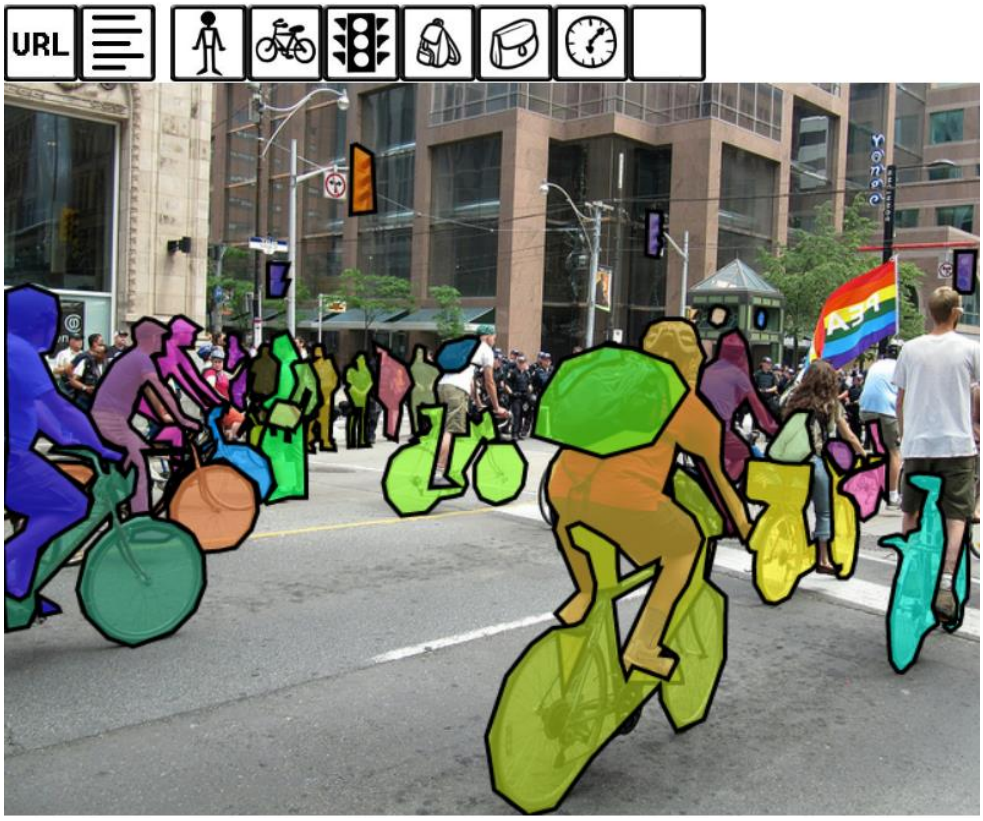
1. در مورد نحوه کار و ایده پشت هر یک از الگوریتم ها توضیح مختصری ارائه دهید و انگیزه ایجاد هر مدل/الگوریتم را نسبت به مدل/الگوریتم قبلی شرح دهید.
2. از بین دو الگوریتم Violo-Jones یا HOG یکی را به دلخواه انتخاب کرده و بر روی یک تصویر دلخواه از اینترنت نتایج را به کمک کتابخانه "open cv" بدست آورید.

Machine Vision [1]
Object detection [2]
State of the art [3]
You look only once [4]
Single-shot multi box detector [5]
Face detection [6]
Histogram of Oriented Gradients [7]
Convolution Neural Networks [8]
Region based Convolution Neural Networks [9]
Version [10]

ب) پیاده سازی YOLO

1) آشنایی با مدل

در این بخش به پیاده سازی YOLOv3 می‌پردازیم. YOLOv3 بر روی مجموعه داده "COCO" آموزش داده شده است که دارای بیش از 30,000 تصویر می‌باشد و 80 کلاس مختلف از اشیاء را پوشش می‌دهد.



نمونه‌ای از یک تصویر برچسب خورده در مجموعه داده "COCO"

همچنین از طریق این لینک میتوانید به مدل‌های مختلف YOLOv3 که بر روی دیتاست "Coco" آموزش داده شده اند، دسترسی داشته باشید. برای کاربرد ما، مدل YOLOv3-320 مناسب می‌باشد (به این معنا که تصویر ورودی مدل، به 320 پیکسل در جهت طول و 320 پیکسل در جهت عرض اسکیل می‌شود). برای این منظور دو فایل مربوط به وزن‌های مدل و اطلاعات معماری مدل را در زیر ستون‌های "Weights" و "Cfg" دانلود کنید.

2) درک تئوری مسئله

مهم ترین قسمت قبل از پیاده سازی YOLO در کد، فهم اجزاء مختلف پیاده سازی می‌باشد که تا حدی نیاز به درک تئوری دارد. با پاسخ دادن به سوالات زیر می‌توانید به درک بهتری از الگوریتم نهفته YOLO قبل پیاده سازی آن داشته باشید (لینک مقالات YOLO و YOLOv3):

1.2) مفهوم Grid cell را تحقیق کنید و بررسی کنید که سیاست YOLO در هر Grid cell به چه صورتی می‌باشد ؟ (هر Grid cell چند Bounding box و چند کلاس را پیش بینی می‌کند ؟)

2.2) مفهوم Bounding box را توضیح دهید. برای شناسایی یک bounding box به چه پارامترهایی نیاز است ؟

3.2) خروجی پیش بینی شده توسط YOLO شامل چه پارامترهایی می‌باشد ؟ هر کدام را به مختصر توضیح دهید. اندازه Tensor خروجی پیش بینی شده را به صورت رابطه‌ای از پارامتر های بدست آمده ذکر کنید.

4.2) معیار Confidence را چگونه می‌توان بدست آورد ؟ این معیار دقیقا چه چیزی را نشان می‌دهد ؟

5.2) تابع خطا[1] در YOLO به چه صورتی تعریف می‌شود و از چه پارامترهایی تاثیر می‌پذیرد ؟

6.2) مفهوم Non-Max-suppression را توضیح دهید؟

7.2) اگر در یک grid cell بیش از یک Object وجود داشته باشد، چه راهکاری را باید اتخاذ کرد ؟ بردار خروجی پیش بینی شده چه تغییری خواهد کرد ؟

3) مراحل پیاده سازی

فرض کنید که می‌خواهیم تنها سه اشیاء ماشین (Car)، انسان (Person) و اتوبوس (Bus) را در درون ویدیوی دلخواه ورودی شناسایی کنیم. به کمک فایل "COCO_Class_Codes" در فایل پروژه می‌توانید اندیس متناظر هر کدام از اشیاء ذکر شده را پیدا کنید.

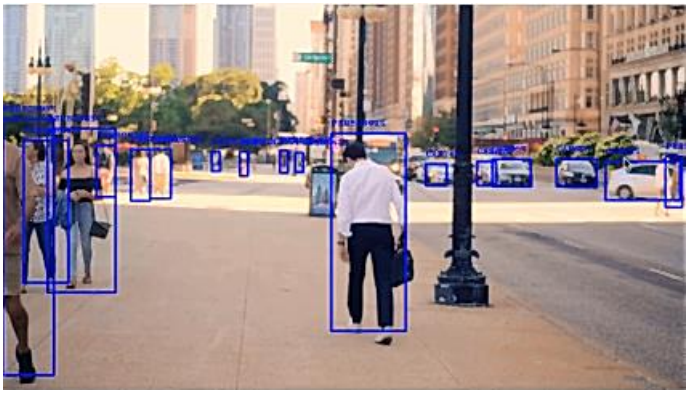
a.3) برای آنکه بتوان شناسایی اشیاء را بر روی یک ویدیو پیاده سازی کرد، نیاز است که ویدیو را به فریم‌های[2] متوالی تقسیم کنیم و در هر فریم وظیفه[3] شناسایی اشیاء را انجام دهیم. دستورهای کار با ویدیو در قسمت دستور های کاربردی open cv آورده شده است.

Loss function [1]
Frame [2]
Task [3]

b.3) در هر فریم نیاز است که به کمک دستور forward خروجی‌های پیش بینی شده را استخراج کرد. تابعی بنام "find_object" بنویسید که با گرفتن خروجی پیش بینی شده در هر فریم، پارامترهای هر فریم را (مختصات اشیاء، مقادیر confidence و ...) برگرداند.

c.3) تابعی بنام "show_detected_object" بنویسید که با گرفتن مقادیر پارامترهای بدست آمده از قسمت b.3، اشیاء شناسایی شده به همراه لیبل آنها (ماشین، انسان یا اتوبوس) را بر روی ویدیو ورودی نشان دهد. ویدیو جدیدی حاوی اشیاء شناسایی شده ذخیره کنید و در پوشه‌ی آپلودی قرار دهید.

نمونه‌ای از یک فریم خروجی در زیر نشان داده شده است:



4) دستورهای کاربردی open cv

cv2.dnn.NMSBoxes: Applying Non-Max-Suppression
cv2.rectangle: Drawing a rectangle over an area
cv2.putText: draw text over an area
cv2.VideoCapture: Reading a desired video
cv2.VideoWriter: Write a video
.read(): Read a video
cv2.dnn.readNetFromDarknet: Loading pre-trained Neural Net using weight and cfgs files
. setPreferableTarget: setting CPU or GPU
. blobFromImage: Converting to blob format
. setInput(blob): setting the input of your NN as blob format
. getLayerNames(): Get all imported layers in YOLO
. getUnconnectedOutLayers(): Get indexes of Unconnected output layers
. waitKey: Waite between each frame
. release(): Release a video

نکات تحویل:

- **مهلت تحویل این تمرین 16 بهمن میباشد.**
- **انتظار می‌رود تمام اعضای گروه بر انواع مفاهیم، راه‌کار ها، روش های پیشنهادی و نتایج تسلط کامل داشته باشند.**
- **برای انجام این تمرین تنها مجاز به استفاده از زبان برنامه نویسی پایتون هستید. در سوالاتی که ازشما خواسته شده یک الگوریتم را پیاده سازی کنید مجاز به استفاده از توابع آماده نمی باشید مگر اینکه در صورت سوال مجاز بودن استفاده از این توابع یا کتابخانه ها صریح ذکر شده باشد.**
- **کدهای مربوط به هربخش می بایست در پوشه ای با نام Codes در کنار گزارش کار شما موجود باشد. این کدها باید خوانا و به صورت مرتبط نام گذاری شده باشند، لذا توضیحات لازم را به صورت یادداشت[1] در کدهای خود قرار دهید.**
- **لطفا تمامی نکات و مفروضاتی که برای پیاده سازی ها و محاسبات خود در نظر می گیرید را در گزارش ذکر کنید. همچنین رعایت موازین نگارشی در گزارش توجه ویژه ای داشته باشید (بطور مثال استفاده از زیرنویس برای تصاویر و بالانویس برای جداول).**
- **برای پروژه هر گروه علاوه بر گزارش کتبی ملزم به ارائه گزارشی در قالب ارائه علمی خواهد بود، این ارائه از اهمیت ویژه ای در مراحل نمره دهی برخوردار است و تمام افراد گروه باید به تمام مباحث پروژه اشراف داشته باشند و به سوالات مطرح شده در ارائه پاسخ دهند.**
- **لطفا گزارش، فایل کدها و سایر ضمائم مورد نیاز را با ترتیب نام گذاری زیر در صفحه درس در سامانه بارگذاری کنید.**

FinalProject_[StudentNumber(s)].zip

- **در صورت وجود هر گونه ابهام یا مشکل لطفا به مسئولان پروژه ایمیل بزنید.**

قسمت اول (ati.mollabagher@ut.ac.ir , mr.teymoorian@ut.ac.ir)

قسمت دوم (sh.vassef@ut.ac.ir)

[1] Comment