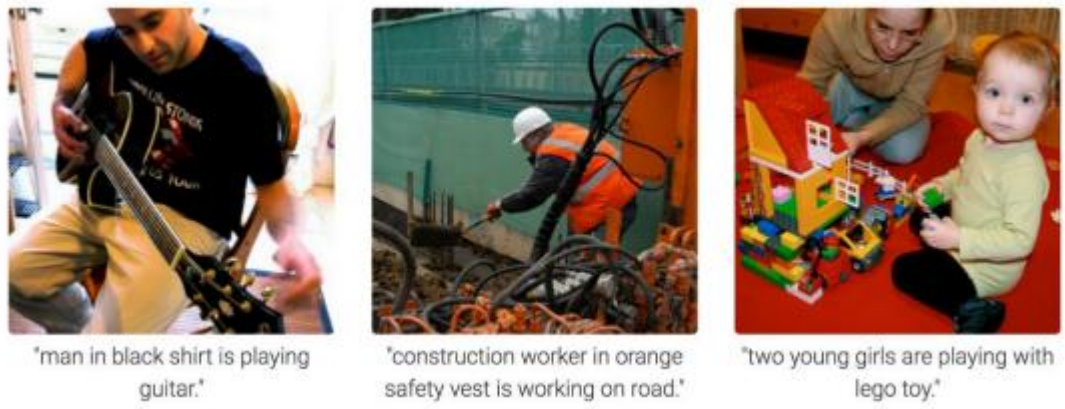
پرسش ۱. توصیف عکس[1]

یکی از حوزه‌های جذاب در یادگیری ماشین، توصیف یک عکس با یک جمله است. در واقع هدف ایجاد و آموزش مدلی است که بتواند یک تصویر را به عنوان ورودی بگیرد و در نهایت یک جمله در توصیف آن عکس در خروجی خود تولید کند. تصویر زیر نمونه‌ای از خروجی این شبکه را نشان می‌دهد.



شکل ۱. خروجی یک مدل آموزش دیده برای **Image Captioning**

حال در این تمرین قصد داریم یک مدل برای رسیدن به این هدف پیاده‌سازی نماییم. ساختار کلی این مدل‌ها به این صورت است که یک شبکه CNN جهت تولید ویژگی‌های تصاویر وجود دارد و در کنار آن روش‌های مختلفی برای Embedding جملات موجود است که در نهایت بردار ویژگی تصاویر و متن در کنار هم قرار گرفته و به عنوان ورودی یک شبکه بازگشتی اعمال می‌شود تا در نهایت جمله نهایی را تولید نماید. در ادامه بیشتر با بخش های مختلف آن آشنا خواهید شد. مقاله‌ای که شما در این بخش از تمرین میتوانید به آن رجوع کنید مقاله Image Captioning است که به پیوست هم برای شما قرار داده شده است.

[1] Image Captioning

۱-۱. مجموعه دادگان و پیش پردازش آنها

با مطالعه مقاله اشاره شده متوجه می‌شوید که سه مجموعه داده معرفی شده است. مجموعه دادگانی که باید شما در این تمرین استفاده کنید flickr8k است که مجموعه دادگانی با سایز کوچکتر در مقاله اشاره شده است. این مجموعه داده را می‌توانید از پیوند زیر دریافت کنید:

https://www.kaggle.com/datasets/adityajn105/flickr8k

این مجموعه از دو بخش به نام Image و Caption.txt تشکیل شده است که پوشه Image شامل ۸۰۹۱ تصویر و Caption.txt شامل ۴۰۴۵۵ جمله است که برای هر تصویر ۵ جمله مختلف توسط افراد مختلف جمع آوری شده است. در کنار هر جمله نام تصویر مورد نظر نیز آورده شده است.

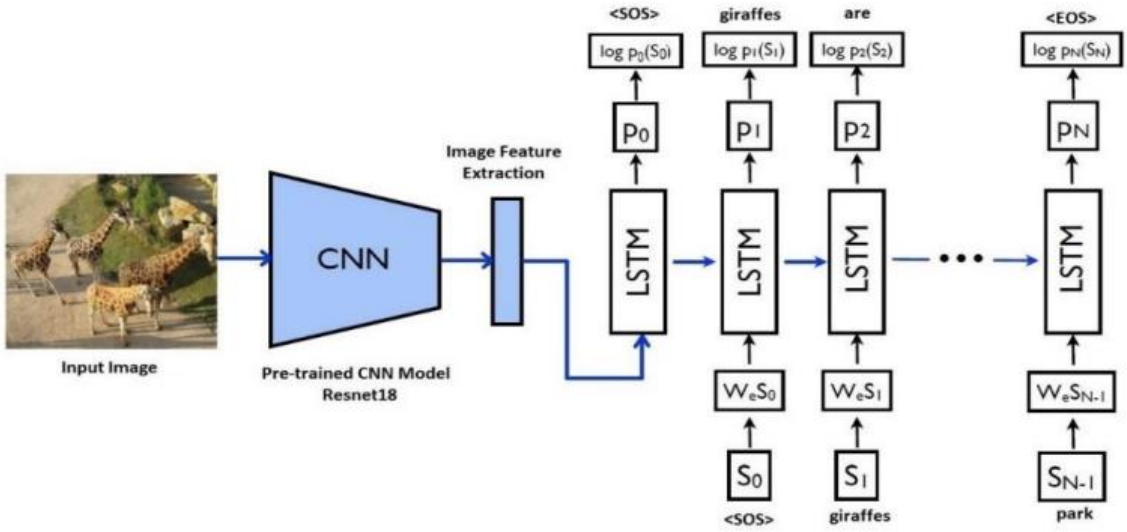
با آماده سازی تصاویر برای اعمال به شبکه‌های کانولوشنی پیش‌تر آشنا شدید. در اینجا جملات نیز باید پیش‌پردازش شوند تا به بردارهایی از اعداد تبدیل شوند. ما در اینجا برای سادگی پیشنهاد می‌کنیم که از لایه Embedding در پایتورچ استفاده کنید که نحوه کار با این لایه را در پیوند زیر مشاهده می‌کنید: (شما میتوانید از سایر روش ها هم به انتخاب خودتان بهره ببرید که نیاز هست که در گزارش خودتان به آن اشاره کنید)

https://pytorch.org/docs/stable/generated/torch.nn.Embedding.html

پارامتری به نام Embedding_dim در آن وجود دارد که می‌توانید آن را ۳۰۰ در نظر بگیرید که البته انتخاب آن در اختیار شما می‌باشد که در واقع این عدد مشخص می‌کند که برای هر کلمه یک بردار عددی با طول ۳۰۰ در نظر بگیرید. نکته که مهمی که در پیش پردازش داده ها باید توجه نمایید، این است که باید برای هر جمله از توکن‌های شروع و پایان جمله <SOS> و <EOS> استفاده نماییم. که توکن‌های خاصی می‌باشد که توسط خود شما تعریف می‌شوند. همچنین باید مجموعه لغات موجود در مجموعه دادگان خود را پردازش و به هر کدام از آنها یک Index نسبت دهید. بهتر است علامت های نگارشی از جملات حذف شوند. همچنین از آنجایی که جملات Captionها طولهای متفاوتی دارند باید طول آن ها باهم یکسان شوند، که این کار را با Padding مناسب می‌توانید انجام دهید که می‌توان یک طول مشخص ثابت را در نظر گرفت یا یکسان‌سازی را در هر mini batch انجام داد.

۱-۲. مدل شبکه

در شکل شماره ۲ مدل کلی مد نظر را مشاهده می‌کنید. همان‌طور که مشاهده می‌کنید، بخشی از مدل جهت استخراج ویژگی تصاویر مورد استفاده قرار می‌گیرد. در این مسئله ما قصد داریم از یک مدل از پیش آموزش داده شده Resnet18 استفاده نماییم. این مدل در کتابخانه پایتورچ قابل دسترس می‌باشد و از آخرین لایه شبکه کانولوشنی آن ویژگی‌های تصویر استخراج می‌شود که در نهایت نیاز است به یک لایه خطی جهت استخراج ویژگی های مورد نظر با ابعاد مناسب جهت ورود به شبکه بازگشتی، استفاده نمود.

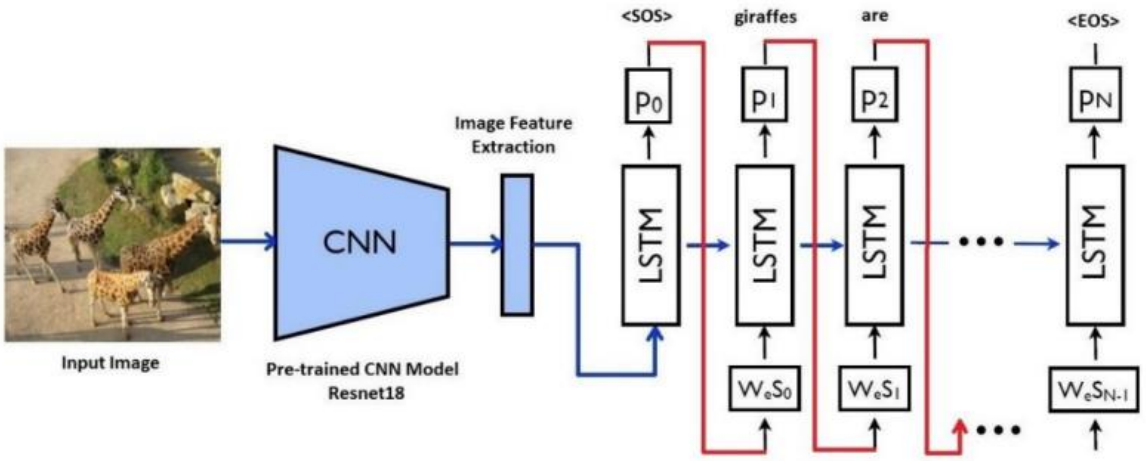


شکل ۲. تصویر مدل مورد بررسی در سوال اول

در این قسمت از یک لایه شبکه LSTM با تعداد ۲۵۶ لایه پنهان استفاده می نماییم و بردار های Embed شده جملات در کنار بردار تصویر به آن داده شده و خروجی آن به یک لایه خطی به سایز ورودی Hidden State و سایز خروجی تعداد کلمات موجود در مجموعه دادگان اعمال می‌شود و به این ترتیب به محاسبه خطا و پیش‌بینی مدل می‌پردازیم.

۱-۳. پیش‌بینی شبکه

بعد از آموزش شبکه، نیاز دارید تا شبکه را ارزیابی نمایید. جهت ارزیابی شبکه باید به صورتی که در شکل ۳ نشان داده شده از شبکه استفاده نماییم.



شکل ۳. نحوه استفاده از مدل در زمان تست جهت تولید جمله

همان‌طور که می‌دانیم در زمان تست شبکه آموزش داده شده، Caption وجود ندارد و ما باید برای یک تصویر Caption تولید نماییم. برای این منظور روش های مختلفی وجود دارد ولی ما در اینجا مدل بالا را پیشنهاد می‌دهیم. در یک تابع به عنوان ورودی، تصویر تست و مدل آموزش داده شده را جهت پیش‌بینی کلمات اعمال می‌کنیم. قطعه کد زیر الگوریتم این شبکه را نمایش داده‌است.

```
input_data = Trained_Model.CNN(image)
states = None  #(Hn, Cn)

for _ in range(max_length):
    hiddens, states = Trained_Model.lstm(input_data, states)

    output = Trained_Model.linear(hiddens)

    predicted_index = output.argmax()

    input_data = Trained_Model.Embedding(predicted_index)

    caption_prediction.append(predicted_index)

    if predicted_index.item() == "<EOS>":
        break
```

شکل ۴. الگوریتم بازگو کننده شبکه شکل ۳ جهت تولید جمله

در نهایت caption_prediction مجموعه indexهای کلمات می‌باشد که در نهایت به کمک دایره لغات موجود در مجموعه دادگان قابل تبدیل به کلمات می‌باشد. توجه داشته باشید که الگوریتم فوق فقط مراحل کار را نشان داده است و نیاز به بازنویسی درست، رعایت ابعاد تنسورها و غیره دارد که بر عهده شما می‌باشد. البته استفاده از هر شیوه دیگری جهت تست و تولید جملات بلامانع است.

۱-۴. پرسش‌ها

در این بخش به پرسش‌های زیر با توجه به بخش‌های پیش پردازش، مدل شبکه و پیش‌بینی شبکه برای هر پرسش پاسخ دهید:

۱. از یک مدل از پیش آموزش داده شده Resnet18 به عنوان شبکه CNN استفاده نمایید و به جز لایه خطی آخر تمامی لایه‌های آن را Freeze نمایید تا در عملیات بروزرسانی وزن‌ها شرکت نداشته باشند. سپس خروجی آن را در کنار بردارهای Embed شده جملات به یک لایه شبکه LSTM یک طرفه اعمال کرده و نمودار خطای آموزش و تست را در طول یادگیری گزارش نمایید. از تابع خطای CrossEntropy و تابع بهینه‌ساز Adam می‌توانید استفاده نمایید. بعد از فرآیند آموزش، ۳ عدد عکس از دادگان تست را جهت پیش‌بینی مدل، به آن اعمال کرده و خروجی آن را در گزارش کار خود ذکر نمایید. **(۵۰ نمره)**

(جزئیات بارم: پیش پردازش: ۱۰ نمره، مدل شبکه: ۱۰ نمره، پیش‌بینی شبکه: خروجی خطا: ۱۵ نمره و خروجی تصویر: ۱۵ نمره)

۲. با حفظ موارد گفته شده سؤال قبل تمامی لایه‌های شبکه Resnet18 را Unfreeze نمایید و مجددا موارد خواسته شده در سوال قبل را بررسی نمایید و نتایج بدست آمده را با سؤال قبل مقایسه کنید. **(۵۰ نمره)**

(جزئیات بارم: پیش پردازش: ۱۰ نمره، مدل شبکه: ۱۰ نمره، پیش‌بینی شبکه: خروجی خطا: ۱۵ نمره و خروجی تصویر: ۱۵ نمره)