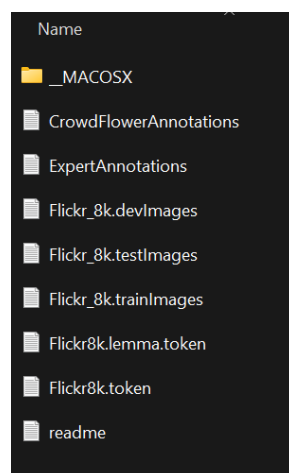


## Image Caption generator with CNN & LSTM

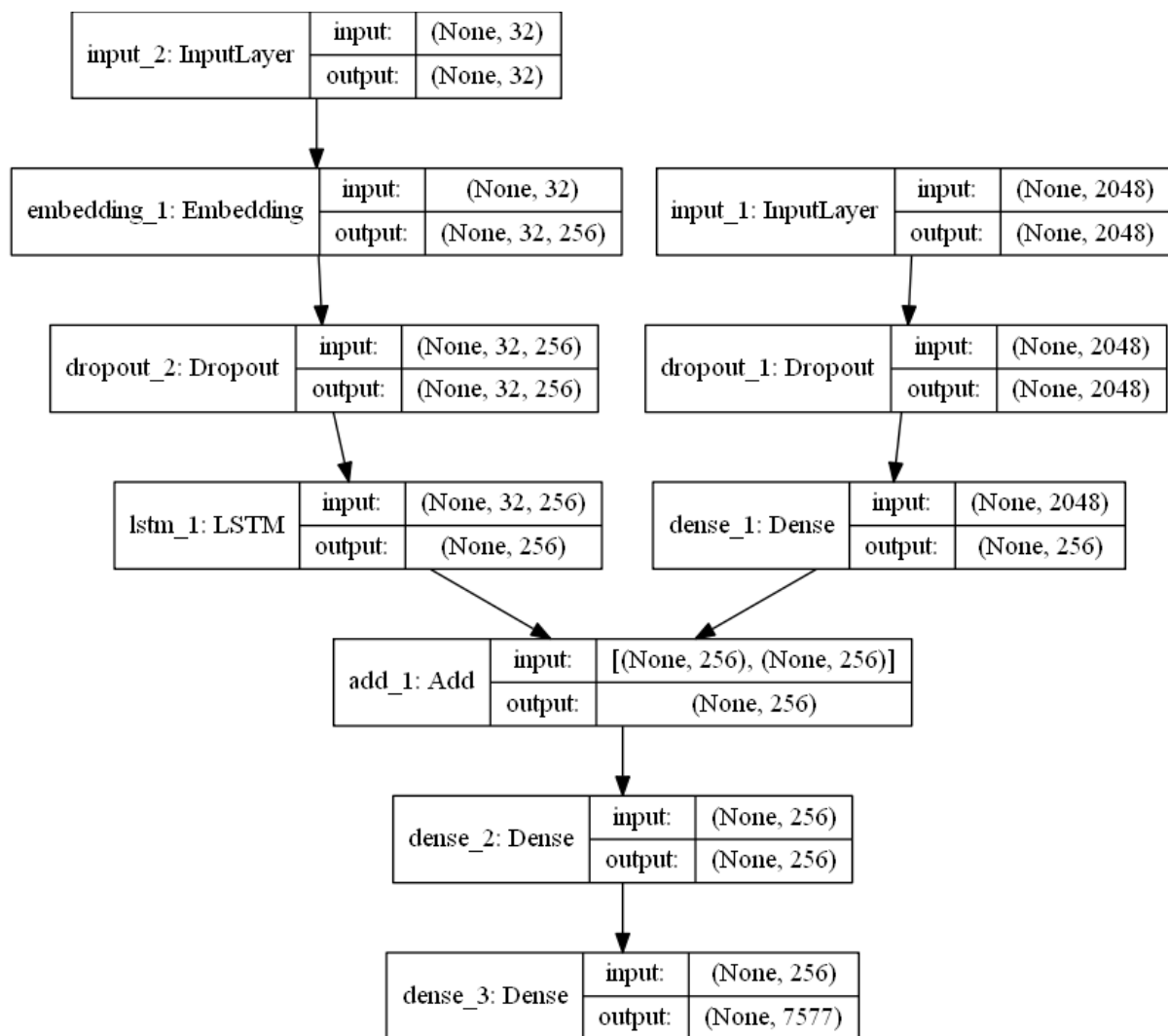
دیتا:

[Flickr8k Dataset](#) •  
[Flickr8k text](#) •



```
1305564994_00513f9a5b.jpg#0 A man in street racer armor be examine the tire of another racer 's motorbike .
1305564994_00513f9a5b.jpg#1 Two racer drive a white bike down a road .
1305564994_00513f9a5b.jpg#2 Two motorist be ride along on their vehicle that be oddly design and color .
1305564994_00513f9a5b.jpg#3 Two person be in a small race car drive by a green hill .
1305564994_00513f9a5b.jpg#4 Two person in race uniform in a street car .
1351764581_4d4fb1b40f.jpg#0 A firefighter extinguish a fire under the hood of a car .
1351764581_4d4fb1b40f.jpg#1 a fireman spray water into the hood of small white car on a jack
1351764581_4d4fb1b40f.jpg#2 A fireman spray inside the open hood of small white car , on a jack .
1351764581_4d4fb1b40f.jpg#3 A fireman use a firehose on a car engine that be up on a carjack .
1351764581_4d4fb1b40f.jpg#4 Firefighter use water to extinguish a car that be on fire .
1358089136_976e3d2e30.jpg#0 A boy sand surf down a hill
1358089136_976e3d2e30.jpg#1 A man be attempt to surf down a hill make of sand on a sunny day .
1358089136_976e3d2e30.jpg#2 A man be slide down a huge sand dune on a sunny day .
1358089136_976e3d2e30.jpg#3 A man be surf down a hill of sand .
1358089136_976e3d2e30.jpg#4 A young man in short and t-shirt be snowboard under a bright blue sky .
1362128028_8422d53dc4.jpg#0 kid play in a blue tub full of water outside
1362128028_8422d53dc4.jpg#1 On a hot day , three small kid sit in a big container fill with water .
1362128028_8422d53dc4.jpg#2 Little kid sit outdoors in a small tub of water .
1362128028_8422d53dc4.jpg#3 Three child squeeze into a plastic tub fill with water and play .
1362128028_8422d53dc4.jpg#4 Three little boy take a bath in a rubber bin on the grass .
1383698008_8ac53ed7ec.jpg#0 A man be snowboard over a structure on a snowy hill .
1383698008_8ac53ed7ec.jpg#1 A snowboarder jump through the air on a snowy hill .
1383698008_8ac53ed7ec.jpg#2 a snowboarder wear green pants do a trick on a high bench
1383698008_8ac53ed7ec.jpg#3 Someone in yellow pants be on a ramp over the snow .
1383698008_8ac53ed7ec.jpg#4 A man be perform a trick on a snowboard high in the air .
1468103286_96a6e07029.jpg#0 A Baseball batter raise his arm .
1468103286_96a6e07029.jpg#1 A baseball player from New York wait to bat during a game .
1468103286_96a6e07029.jpg#2 A baseball player in a Yankee uniform be hold a bat in one hand
1468103286_96a6e07029.jpg#3 A New York Yankee hold up a bat .
1468103286_96a6e07029.jpg#4 New York Yankee warm up .
1478268555_7e301fc510.jpg#0 A bare backed climber be attach to the rock face on a pink safety rope .
1478268555_7e301fc510.jpg#1 a lone rock climber in a harness climb a huge rock wall .
1478268555_7e301fc510.jpg#2 A man be rock crawl up a rock wall
1478268555_7e301fc510.jpg#3 A man wear no shirt climb a steep cliff .
1478268555_7e301fc510.jpg#4 man with no shirt on climb rock
161669933_3e7d8c7e2c.jpg#0 A competitive motorcycle racer prepare to make a left turn along a paved road .
161669933_3e7d8c7e2c.jpg#1 A man ride a motorcycle with the number 5 on it .
```

## معماری استفاده شده:



## ۱. ایمپورت کتابخانه‌ها

```
import string
import numpy as np
from PIL import Image
import os
from pickle import dump, load

from keras.applications.xception import Xception, preprocess_input
from keras.preprocessing.image import load_img, img_to_array
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.utils import to_categorical
from keras.layers.merge import add
from keras.models import Model, load_model
from keras.layers import Input, Dense, LSTM, Embedding, Dropout
from tqdm import tqdm_notebook as tqdm
tqdm().pandas()
```

## ۲. توابع پایه برای خواندن و آماده‌سازی توصیفات (Captions)

### ۲/۱. تابع load\_doc

```
def load_doc(filename):
    """
    فایل متنی را باز کرده و محتوای آن را به صورت یک رشته برمی‌گرداند
    """
    # TODO: فایل را باز کنید و محتوایش را برگردانید.
    #       کنید return را (text) در نهایت رشته
    pass
```

## ۲/۲. تابع `all_img_captions`

```
def all_img_captions(filename):  
    """  
    فایل کپشن‌ها را می‌گیرد و دیکشنری‌ای برمی‌گرداند  
    که کلید نام عکس و مقدار یک لیست از کپشن‌های مربوط به آن عکس است  
    """  
  
    # TODO:  
    # 1) کل فایل کپشن را بخوانید، با load_doc کمک تابع (1)  
    # 2) هر خط از فایل را جدا کنید تا کپشن‌های هر عکس را تفکیک کنید (2)  
    # 3) یک دیکشنری بسازید که کلیدش نام عکس و مقدارش لیستی از کپشن‌های آن باشد (3)  
    pass
```

## ۲/۳. تابع `cleaning_text`

```
def cleaning_text(captions):  
    """  
    کارهای زیر را انجام دهید  
    """  
  
    # TODO  
    # 1) حذف یا جایگزینی برخی علائم نگارشی (1)  
    # 2) تبدیل همه کلمات به حروف کوچک (2)  
    # 3) حذف کلمات خیلی کوتاه یا کلماتی که حاوی عدد هستند (3)  
    # 4) ایجاد خروجی نهایی تمیزشده (4)  
    pass
```

## ۲/۴. تابع `text_vocabulary`

```
def text_vocabulary(descriptions):  
    """  
    از کلمات منحصریه‌فردی که در همه کپشن‌ها وجود دارد، برمی‌گرداند (set) یک مجموعه  
    """  
    vocab = set()  
  
    # TODO
```

## ۲/۵. تابع `save_descriptions`

```
def save_descriptions(descriptions, filename):  
    """  
    را که حاوی کپشن‌های تمیزشده است descriptions دیکشنری  
    در فایلی متنی ذخیره می‌کند. هر خط  
    """  
    # TODO  
    pass
```

## ۳. آماده‌سازی دیتاست و پیش‌پردازش

```
dataset_text = ""  
dataset_images = ""  
  
filename = dataset_text + "/" + "Flickr8k.token.txt"  
# برای بدست آوردن دیکشنری all_img_captions فراخوانی تابع  
descriptions = all_img_captions(filename)  
print("Length of descriptions =", len(descriptions))  
  
# تمیز کردن کپشن‌ها  
clean_descriptions = cleaning_text(descriptions)  
  
# ساختن دایره لغت  
vocabulary = text_vocabulary(clean_descriptions)  
print("Length of vocabulary =", len(vocabulary))  
  
# descriptions.txt ذخیره کپشن‌های تمیز شده در فایل  
save_descriptions(clean_descriptions, "descriptions.txt")
```

```
1000268201_693b08cb0e.jpg child in pink dress is climbing up set of stairs in an entry way
1000268201_693b08cb0e.jpg girl going into wooden building
1000268201_693b08cb0e.jpg little girl climbing into wooden playhouse
1000268201_693b08cb0e.jpg little girl climbing the stairs to her playhouse
1000268201_693b08cb0e.jpg little girl in pink dress going into wooden cabin
1001773457_577c3a7d70.jpg black dog and spotted dog are fighting
1001773457_577c3a7d70.jpg black dog and tricolored dog playing with each other on the road
1001773457_577c3a7d70.jpg black dog and white dog with brown spots are staring at each other in the street
1001773457_577c3a7d70.jpg two dogs of different breeds looking at each other on the road
1001773457_577c3a7d70.jpg two dogs on pavement moving toward each other
1002674143_1b742ab4b8.jpg little girl covered in paint sits in front of painted rainbow with her hands in bowl
1002674143_1b742ab4b8.jpg little girl is sitting in front of large painted rainbow
1002674143_1b742ab4b8.jpg small girl in the grass plays with fingerpaints in front of white canvas with rainbow on it
1002674143_1b742ab4b8.jpg there is girl with pigtails sitting in front of rainbow painting
1002674143_1b742ab4b8.jpg young girl with pigtails painting outside in the grass
1003163366_44323f5815.jpg man lays on bench while his dog sits by him
1003163366_44323f5815.jpg man lays on the bench to which white dog is also tied
1003163366_44323f5815.jpg man sleeping on bench outside with white and black dog sitting next to him
1003163366_44323f5815.jpg shirtless man lies on park bench with his dog
1003163366_44323f5815.jpg man laying on bench holding leash of dog sitting on ground
1007129816_e794419615.jpg man in an orange hat starring at something
1007129816_e794419615.jpg man wears an orange hat and glasses
1007129816_e794419615.jpg man with gauges and glasses is wearing blitz hat
1007129816_e794419615.jpg man with glasses is wearing beer can crocheted hat
1007129816_e794419615.jpg the man with pierced ears is wearing glasses and an orange hat
1007320043_627395c3d8.jpg child playing on rope net
1007320043_627395c3d8.jpg little girl climbing on red roping
1007320043_627395c3d8.jpg little girl in pink climbs rope bridge at the park
1007320043_627395c3d8.jpg small child grips onto the red ropes at the playground
1007320043_627395c3d8.jpg the small child climbs on red ropes on playground
1009434119_febe49276a.jpg black and white dog is running in grassy garden surrounded by white fence
1009434119_febe49276a.jpg black and white dog is running through the grass
1009434119_febe49276a.jpg boston terrier is running in the grass
1009434119_febe49276a.jpg boston terrier is running on lush green grass in front of white fence
1009434119_febe49276a.jpg dog runs on the green grass near wooden fence
1012212859_01547e3f17.jpg dog shakes its head near the shore red ball next to it
1012212859_01547e3f17.jpg white dog shakes on the edge of beach with an orange ball
```

#### ۴. استخراج ویژگی از تصاویر (با مدل Xception)

```
def extract_features(directory):
    """
    برای استخراج ویژگی‌های تصویری استفاده می‌کند Xception از مدل
    ویژگی هر عکس به شکل یک بردار 2048 تایی خواهد بود
    """
    # TODO:
    # 1) بدون درنظر گرفتن لایه آخر تعریف کنید مدل Xception
    # 2) کرده، نرمال‌سازی کنید و به مدل دهید تا ویژگی آن را استخراج کند resize برای هر تصویر در پوشه، آن را
    # 3) را صدا بزنید و خروجی را در دیکشنری ذخیره کنید predict(...) مدل.
    pass

# TODO:
# 1) dataset_images روی extract_features فراخوانی تابع
# 2) pickle.dump(...) در فایل features.p کنید
# 3) دوباره بارگذاری کنید pickle.load(...) بعداً با
```

۵. آماده‌سازی داده‌ها برای آموزش

۵/۱. توابع کمکی برای بارگذاری داده‌های آموزشی

```
def load_photos(filename):  
    """  
    هستند را (Flickr_8k.trainImages.txt) نام عکس‌هایی که در فایل  
    در قالب یک لیست برمی‌گرداند.  
    """  
    # TODO  
    pass  
  
def load_clean_descriptions(filename, photos):  
    """  
    را می‌خواند و تنها کپشن‌های مربوط به descriptions.txt فایل  
    را نگه می‌دارد photos عکس‌های موجود در لیست  
    به ابتدا و انتهای کپشن اضافه می‌کند <end> و <start> همچنین  
    """  
    # TODO  
    pass  
  
def load_features(photos):  
    """  
    تمامی ویژگی‌های تصاویر را بارگذاری می‌کند features.p از فایل  
    هستند برمی‌گرداند photos و تنها ویژگی‌های عکس‌هایی که در لیست  
    """  
    # TODO  
    pass
```

```
filename = dataset_text + "/" + "Flickr_8k.trainImages.txt"  
  
train_imgs = load_photos(filename)  
train_descriptions = load_clean_descriptions("descriptions.txt", train_imgs)  
train_features = load_features(train_imgs)
```

## ۵/۲. ساخت Tokenizer

```
def dict_to_list(descriptions):  
    """  
    .دیکشنری حاوی کپشن‌ها را به یک لیست ساده از جملات تبدیل می‌کند.  
    """  
    # TODO  
    pass  
  
def create_tokenizer(descriptions):  
    """  
    1) همه کپشن‌ها را در یک لیست جمع کنید  
    2) بسازید و روی آن لیست فیت کنید Tokenizer یک شیء  
    """  
    # TODO  
    pass  
  
# TODO:  
# 1) tokenizer از روی train_descriptions ایجاد  
# 2) tokenizer در فایل pickle.dump(...) با ذخیره  
# 3) vocab_size = len(tokenizer.word_index) + 1 تعریف
```

## ۵/۳. محاسبه بیشترین طول توصیف

```
def max_length(descriptions):  
    """  
    .بالاترین تعداد کلمه در بین همه کپشن‌ها را برمی‌گرداند  
    """  
    # TODO  
    pass  
  
# TODO:  
# 1) max_length روی descriptions صدا زدن تابع  
# 2) نمایش نتیجه
```



## ۶. طراحی معماری مدل

### ۶/۱. آماده‌سازی داده برای مدل (تابع Generator)

```
def data_generator(descriptions, features, tokenizer, max_length):  
    """  
    برای آموزش تولید می‌کند (X1,X2,y) جنریتور داده که در هر فراخوانی، تعدادی نمونه  
    """  
    # TODO:  
    # راهنما:  
    # while 1: در یک حلقه  
    #     descriptions.items(): روی for در حلقه  
    #         feature = features[key][0]  
    #         X1, X2, y تبدیل به create_sequences(...) سپس با  
    #         yield [[X1, X2], y] در نهایت  
    pass  
  
def create_sequences(tokenizer, max_length, desc_list, feature):  
    """  
    . اندیس کلمات تبدیل می‌کند sequence هر کپشن را به  
    . با حرکت از کلمه اول تا آخر، جفت (ورودی، خروجی) می‌سازد  
    ورودی: (ویژگی تصویر + توالی کلمات تاکنون)  
    خروجی: کلمه بعدی  
    """  
    # TODO  
    pass
```

### ۶/۲. ساخت خود مدل (تابع define\_model)

```
def define_model(vocab_size, max_length):  
    """  
    1) CNN مسیر (ویژگی‌های تصویر) (2048) -> Dropout -> Dense(256)  
    2) LSTM مسیر: (max_length) ورودی توالی کلمات -> Embedding(vocab_size, 256) ->  
    Dropout -> LSTM(256)  
    3) ادغام دو مسیر: add() -> Dense(256, activation='relu') -> Dense(vocab_size, softmax)  
    4) کامپایل مدل  
    """  
    # TODO  
    pass
```

## ۷. آموزش مدل

```
# TODO:
# 1) define_model ساخت مدل با
# 2) epochs = 10 و steps = len(train_descriptions) تعیین
# 3) epochs: به تعداد for در حلقه
#     generator = data_generator(...)
#     model.fit_generator(...)
#     model.save("models/model_" + str(i) + ".h5") ذخیره مدل با
```

## ۸. کد تست مدل (تولید کپشن برای یک عکس جدید)

```
import argparse

ap = argparse.ArgumentParser()
ap.add_argument('-i', '--image', required=True, help="Image Path")
args = vars(ap.parse_args())
img_path = args['image']

def extract_features(filename, model):
    """
    ویژگی 2048 تایی استخراج می‌کند. Xception مشابه مرحله آموزش: یک تصویر را گرفته و با مدل
    """
    # TODO
    pass

def word_for_id(integer, tokenizer):
    """
    اندیس عددی را به کلمه مربوطه برمی‌گرداند.
    None اگر اندیسی پیدا نشد خروجی
    """
    # TODO
    pass

def generate_desc(model, tokenizer, photo, max_length):
    """
    آغاز می‌کند. مدل در هر گام کلمه بعدی را پیش‌بینی می‌کند 'start' با کلمه
    محدود شود max_length برسد یا به 'end' تا جایی که به
    """
    # TODO
    pass

# TODO:
# 1) max_length (همان مقداری که در تمرین بدست آوردید) را تعیین کنید
# 2) tokenizer.p از فایل pickle.load(...) را با
# 3) load_model(...) مدل نهایی را با
# 4) Xception مدل (بدون top) استخراج کنید
# 5) generate_desc با مدل یک کپشن تولید کنید و نتیجه را چاپ کنید
```