Image Recognition Semantic Alignments for Image Captioning using Computer Vision and Natural Language Processing.

Final Report Presentation

Authors: Tejasri Surapaneni and Shinoj Kumar

Course Project Professor: Dr. Duoduo Liao

Course Name: Introduction to natural Language Processing

Course name and Section #: AIT 590-01

University Name: George Mason University

Date: 10th May 2020

Table of Contents

Title

Image Recognition Semantic Alignments for Image Captioning using Computer Vision and Natural Language Processing.

Abstract

Robots will eventually be part of every household and it is thus critical to enable algorithms to learn from and be guided by non-expert users. It is an easy problem for a human, but very challenging for a machine as it involves training to make it understand and then identify the content of an image and how to translate this understanding into natural language. The solution to this problem requires to create a computer vision model that is fast enough to analyse the images from the given dataset. We focus on the problem of image captioning in which the quality of the output can easily be judged by non-experts and then made corrections to any mis classification or mis identification and then re training the model accordingly. We first train a captioning model on a subset of images paired with human written captions. We then let the model describe new images and collect human feedback on the generated descriptions. The dataset that is been used in this project is an image dataset which is unstructured and not annotated, yet the descriptions to the image are given manually by our team as a part of this project.

There are several stunning algorithms to work on the image detection or recognition as it is one of the important aspect of computer vision due to increase of practical use. Image recognition is nothing but the ability of the algorithm to identify or recognise image based on the input image. This application is begun to be used in various fields. Some of the field are vehicle detection, security systems, web images and driverless cars. But, when it comes to the image captioning the models are still under stage of development and we are combining the process of both computer vision and then image captioning the results of the images in our dataset. We are choosing a neural captioning model which we will discuss and delineate more on this in the working system and also through the architectural design of it in the section of proposed system.

Introduction

Reinforcement learning has become a standard way of training artificial agents that interact with an environment. Several works explored the idea of incorporating humans in the learning process, in order to help the reinforcement learning agent to learn faster and accurately. In most cases, a human teacher observes the agent act in an environment and can give additional guidance to the learner. We aim to exploit natural language processing to guide an RL agent. While this is possible in limited domains, it can hardly scale to the real scenarios with large action spaces requiring versatile language feedback. Here our goal is to allow a non-expert human teacher to give feedback to an RL agent in the form of natural language. In order to overcome this challenge a method called image captioning has been evolved with many advanced technologies which helps us to identify the image and generate description based on the feedback given by the humans and also binding the descriptions which are fed manually to the model during the training stage.

Research plays a significant role in gathering information and this always helps to understand the better version of any concept. Research is done on what algorithms to choose for implementing the object detection, instance segmentation and localization of the object. Our team will be proceeding with research throughout the project as there are many new aspects to know and work with being a naïve for these datasets and algorithms.

Related Work

Several works incorporate human feedback to help an RL agent learn faster. A few attempts have been made to advise an RL agent using language pioneering work translated advice to a short program which was then implemented as a neural network. The units in this network represent Boolean concepts, which recognize whether the observed state satisfies the constraints given by the program. Several recent approaches trained the captioning model with policy gradients in order to directly optimize for the desired performance metrics. Our work differs from the recent efforts in conversation modelling or visual dialog using Reinforcement

Learning. There are several restrictions that can make methods unsuitable for image detection, such as computational constraints that impede scalability. While several captioning methods exist, we design our own which is phrase-based, allowing for natural guidance by a nonexpert. In order to overcome the difficulties we are focusing on achieving more accurate edge detection from a depth image and then modify the process using morphological operations. This model will yield more accurate edge detection from a depth image and will modify the process using morphological operations. In image recognition, various methods such as R-CNN, SPP-Net, Fast R-CNN and Faster R-CNN are used. These algorithms faced some problems due to processing time as it takes huge amount of time to train the model and real time image processing cannot be done.

Captioning represents a natural way of showing that the algorithm understands a photograph to a non-expert observer and due to its significance, this domain has received significant attention achieving the impressive performance on standard benchmarks. There are different models which aim at image captioning, but they lack in linguistic information and also focusing only on particular metrics rather all of them.

Objectives

The main objective of this project is that to train the model with the unstructured data with the image dataset we have. And then caption the images using the natural language processing techniques and detect the objects within the image using the computer vision. Putting all the objectives and goals of this project together

Generating textual descriptions for the images and then combine the breakthroughs from computer vision and natural language processing.

- Implementing a model which classifies the image and then describe a image which involves the feature of generating the textual descriptions of the contents of the image.
- Generating the textual descriptions for specific regions of the images in the dataset – annotating the images.
- Researching and exploring on different algorithms and ways to fulfil and furnish the goals we are aiming at.

- Implementing a neural network models for captioning through feature extraction and language modelling techniques.
- Create visualizations for the results we retrieved where the image along with a caption is depicted by the model.

The above are all the objectives which are aimed for our project.

Selected Dataset

Data plays a very important role in any analytics or experiment. Dataset we chose is an image dataset which is unlabelled and unstructured dataset. These Images are not confined to or limited to any specific domain related but are universal and candid where the images are related to multi-domain category. This raw dataset is first unsupervised and to make it conveniently compatible to implement the algorithms as it takes less computing power to process the pixels for the grey-scaled rather having coloured pixels. These images are given a description manually.

We use 6K images for training and 2K for testing. In particular, we randomly chose 2K validation and 4K test images from the official validation split. To collect feedback, we randomly chose 6K images from the training set, as well as all 2K images from our validation. In all experiments, we report the performance on our test set. For all the models we use a pre-trained network to extract image features. We use a word vocabulary size of 23,115.

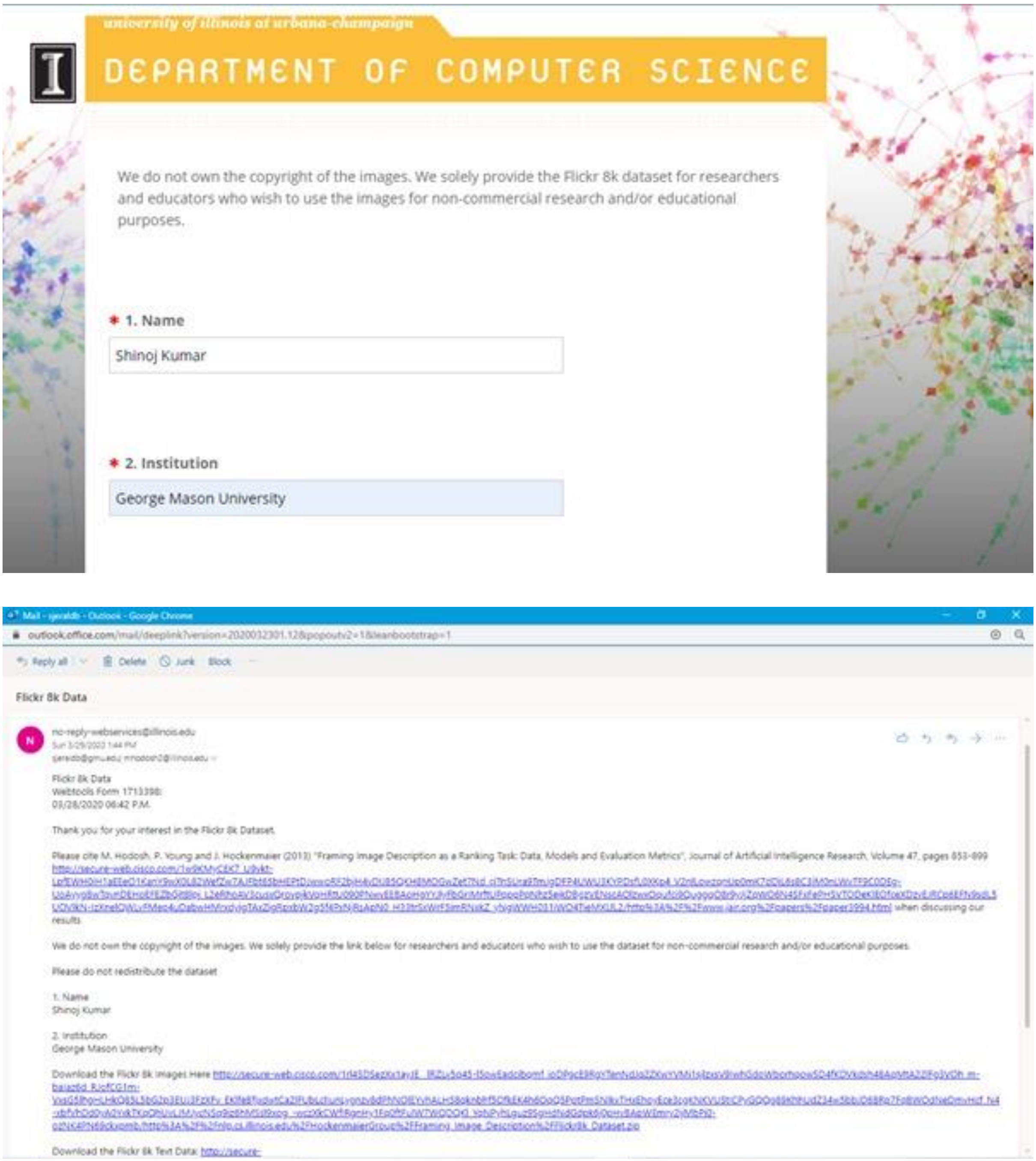


Figure 1: Showing the web page where the details has to be given to download the dataset and their response on registering for a research on this.

Proposed System

The proposed system includes choosing the best algorithm to implement the image recognition techniques which helps in identifying the image in the dataset with computer vision and we need to have a clear understanding of algorithm and its architecture. After continuous research and understanding of the dataset we have which is unstructured and we choose the fast object detection model called RL algorithm which is then used to train the model and then work on the image captioning on the images which are initially classified.

Our objective is to train and test the models with the dataset which consist of images which may take time and used to determine the resources to decrease the latency. Generating textual descriptions for images and the need to combine breakthroughs from computer vision and natural language processing. Describing an image is the problem of generating a human-readable textual description of an image, such as a photograph of an object or scene. Neural network models have come to dominate the field of automatic caption generation; this is primarily because the methods are demonstrating state-of-the-art results.

System Architecture

Explore, scrutinize and understand the image dataset and add the description to each image manually. Captioning represents a natural way of showing that our algorithm understands a photograph to a non-expert observer. Our framework consists of a new phrase-based captioning model that incorporates natural language feedback provided by a human who is a non-expertise.

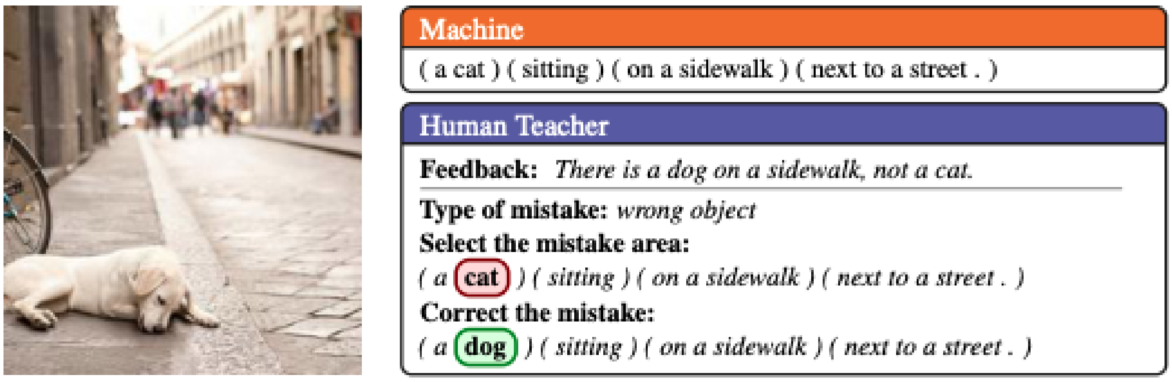


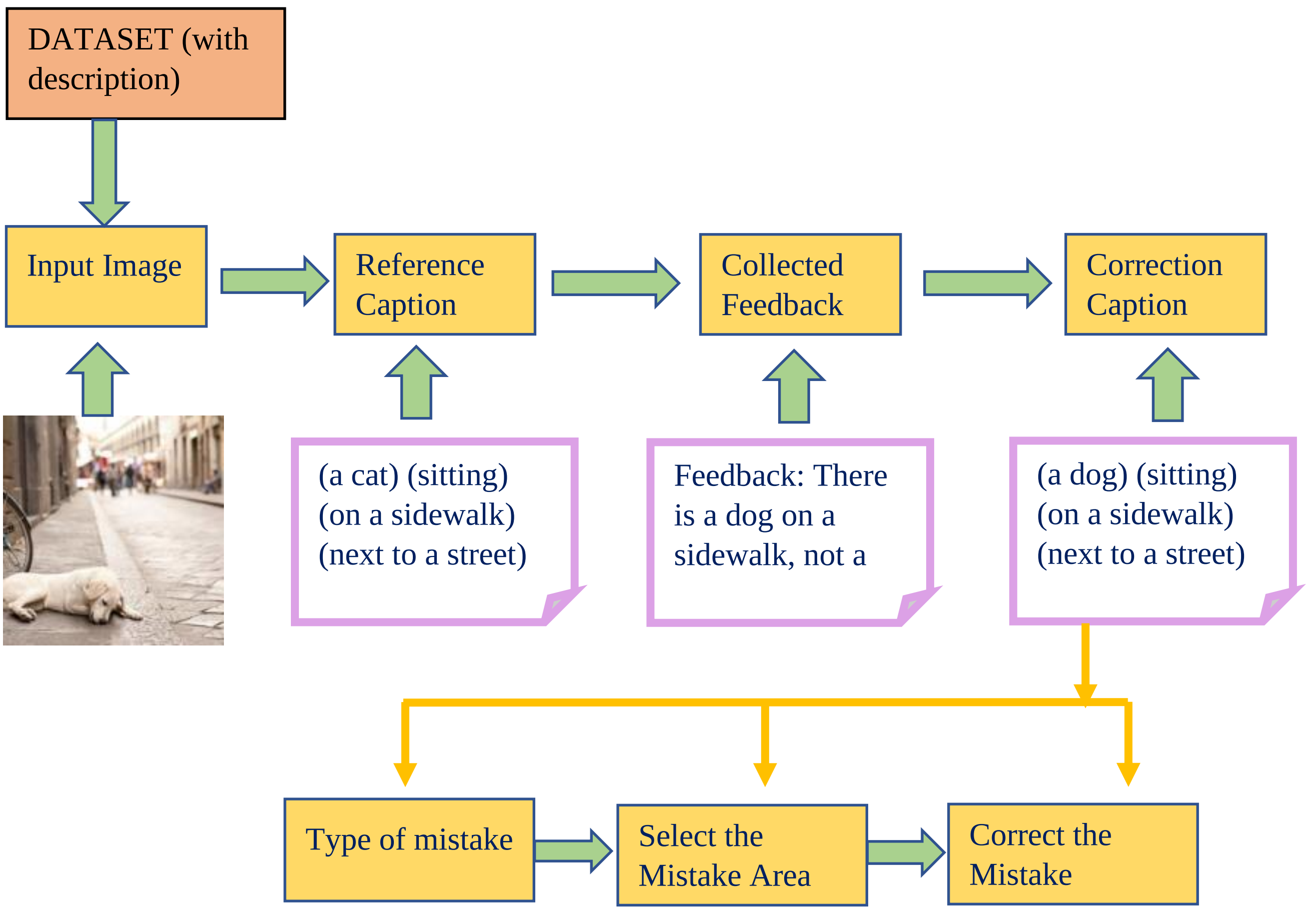
Figure 2: Depicting the model functioning.

Creating a model where it successfully classifies the image using the image recognition techniques and also we aim to predict phrases directly with our captioning model. We first describe our phrase-based captioner, then describe our feedback collection process, and finally propose how to exploit feedback as a guiding signal in policy gradient optimization. We only select captions which are not marked as either perfect or acceptable in the first round.

In the above figure 2 which is depicting the functioning of the model where it accepts the feedback from the human teacher in the form of natural language text and it generates the captions using the current snapshot of the model and then collect the feedback. The machine classifies the individual objects in the image given where it is shown in the picture it predicts or classifies the objects in the image as tuples and then the human teacher who is supposed to give inputs in the form of natural language text and the input is taken and then it corrects the mistakes in case of any wrong predictions made by the model and then model is trained with the corrections made.

While working with large datasets, using a binary file format for storage of data can have a significant impact on the performance of the import pipeline and as a consequence on the training time of the model. To read data efficiently it can be helpful to serialize the data and store it in a set of files that can each be read linearly. We need to specify the structure of the data before writing it to the file.

Conceptual Architecture



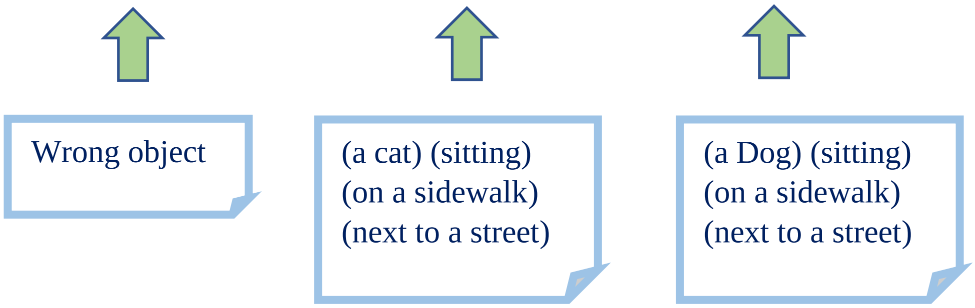


Figure 3: Conceptual System Architecture of the Proposed System

Data Analytics Approaches – NLP & Computer Vision

Summary

The main objective of this project is Image captioning for the images in the dataset through natural language processing. As there are several researches going on this technology we chose this as this is one of the challenging problem with the NLP in this fast paced world. This has many of the hybrid components involved within it, various steps to be implemented in various stages of the model to make this work accordingly. We are amalgamating the concepts of computer vision and image captioning to build a model where the caption of the image can be predicted by the model given the image from the dataset. Displaying both the manual feed to the model and the predicted ones.

Algorithms, Approaches and Accomplishments

Collecting the data: We haven't got the dataset usually from any known data sources like Kaggle/UCI etc.. but we got it from a research team from the University of Illinois where they granted the access to this dataset only for the research and academic purpose but not for any commercial use of it.

Understanding and pre-processing: We spend good amount of time in understanding the data like what the image dataset is consisting of and then how do we proceed with the data. For pre-processing we adjusted the scale of the image to the same size as (256, 256) as the inconsistency in the size may lead to more complications when the scale is changed during the model and for easy up - sampling and to ensure the compatibility we chose this method of scaling where the pixels are adjusted. As our data is unstructured we didn't go through the standard data quality assessments where atomicity, uniqueness and completeness etc.. comes into consideration. As our model primary goal is to predict the caption of the image given the image to the model and a text file which has the manual description.

Model: To make everything ready to run the model we first gave manual description for each of the image (see Appendix A.) then we re-sized the images for compatibility issues and up-sampling. Coming to the model we are running we are figuring it out on the ways we can blend both object detection and image captioning methods because in the market there are several stunning algorithms which are good at object detection or image captioning but we don't have any algorithm where it works best for both. We are figuring it out the ways we can implement the better way. As of now we first planned to classify the images and then add description to the images to manually feed the model with the corrections and then re-run the model for this entire process to make it happen. We are trying using the neural talk 2 and with only CNN algorithm which works the best for it with tuning and tweaking several parameters for it.

	A	B	C	D	E	F	G	H
1	image_name| comment_number| comment							
2	1000092795.jpg| 0| Two young guys with shaggy hair look at their hands while hanging out in the yard .							
3	1000092795.jpg| 1| White males are outside near many bushes .							
4	1000092795.jpg| 2| Two men in green shirts are standing in a yard .							
5	1000092795.jpg| 3| A man in a blue shirt standing in a garden .							
6	1000092795.jpg| 4| Two friends enjoy time spent together .							
7	10002456.jpg| 0| Several men in hard hats are operating a giant pulley system .							
8	10002456.jpg| 1| Workers look down from up above on a piece of equipment .							
9	10002456.jpg| 2| Two men working on a machine wearing hard hats .							
10	10002456.jpg| 3| Four men on top of a tall structure .							
11	10002456.jpg| 4| Three men on a large rig .							
12	1000268201.jpg| 0| A child in a pink dress is climbing up a set of stairs in an entry way .							
13	1000268201.jpg| 1| A little girl in a pink dress going into a wooden cabin .							
14	1000268201.jpg| 2| A little girl climbing the stairs to her playhouse .							
15	1000268201.jpg| 3| A little girl climbing into a wooden playhouse							
16	1000268201.jpg| 4| A girl going into a wooden building .							

Figure 4: Showing the description given to the segments of the image manually – for train images.

```
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 .
```

Figure 5: Showing the description given to the segments of the image manually for the test images.

Code Python 3

Reading the dataset

```
[22]: import os
      train_images_list = os.listdir('Image_dataset/Image_dataset/images/')
```

```
[13]: sample_size = 40
      train_images_list = train_images_list[:sample_size]
```

```
[14]: import os
      import cv2
      import numpy as np
      import tensorflow as tf
      from matplotlib import pyplot as plt
      import random
```

```
[15]: size = (256, 256)
      num_channels = 3
```

```
[16]: train = np.array([None] * sample_size)
      real_images = np.array([None] * sample_size)
```

```
[18]: j = 0
      for i in train_images_list:
          real_images[j] = np.array(plt.imread('Image_dataset/Image_dataset/images/'+ i))
          train[j] = np.array(plt.imread('Image_dataset/Image_dataset/images/' + i))
          j += 1
```

Figure 6: Reading the data, re-sizing the image and using different packages to read the image.

In the above figure 6, it depicts the steps we went through as an approach for this research where we first read the dataset and then we re-sized the images in the dataset. We used OpenCV to read and classify the images and then used the method “imread” to load the image from our files and as of now we are using the imread function which can display the image with the colour unchanged rather changing it to grayscale or colour where by opting the unchanged method it specifies to load the image as such including alpha channel.

The next steps include training the model by feeding both the raw datasets and also the text file which has the description for the segmented images and then give the vocabulary adjusting functions which can help the prediction to happen with the same vocabulary used for describing the images. The visualizations include the image, the originally given description for the image and the predicted caption by the model for the same image.

```
[19]: j  = 0
      for i in train:
          train[j] = cv2.resize(i, size)
          train[j] = train[j].reshape(1, size[0], size[1], num_channels)
          j += 1
```

```
[20]: train = np.vstack(train[:])
```

```
[21]: plt.imshow(np.squeeze(train[0]))
      plt.show()
```

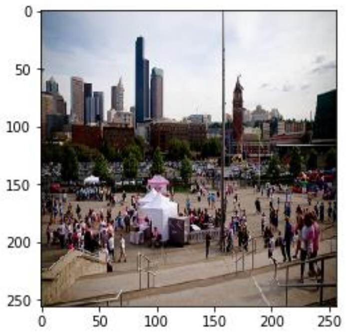


Figure 7: Displaying the image using imread method in Open CV and also after re-sizing the image.

Apart from these steps we are using pandas method to read the csv file where we gave a manual description for each segment of the image with different comment number and comments. Then we used few functions to define the vocabulary and for the model to identify and understand the description of the image. We are going through several trials and errors methods to figure out the best way to retrieve the results with easy computation power.

Image Captioning – Analysis and Results

```
In [17]: import pandas as pd

In [19]: train_captions = pd.read_csv('C:/Users/jayma/Downloads/flickr30k_images/flickr30k_images/results.csv', delimiter='|')

In [20]: def get_images_id(names):
             names = [int(x.split('_')[-1].split('.')[0]) for x in names]
             return names

In [21]: train_captions.columns = ['image_name', 'comment_number', 'comment']

In [22]: def images_map_caption(train_images_list, train_captions):
             caption = []
             for i in train_images_list:
                 caption.append(train_captions[train_captions['image_name'] == i]['comment'].iat[0])
             return caption

In [23]: captions = np.array(images_map_caption(train_images_list, train_captions))
         print(captions.shape)

         (40,)
```

Figure 8: Reading the data, importing the captions which will be used for comparing the original caption with predicted caption.

In the above figure 8, it depicts the steps we went through as an approach for this research. Here, we are importing the captions for all 30k images. The file consists of three parts as image number, comment number and the comment. These comments will be used for checking the predicted comment by comparing with original comment.

Convolution Layer

Training Model ¶

```
In [31]: def create_weights(shape, suffix):
             return tf.Variable(tf.truncated_normal(shape, stddev=0.7), name='W_' + suffix)

         def create_biases(size, suffix):
             return tf.Variable(tf.zeros([size]), name='b_' + suffix)
```

```
In [34]:  def conv_layer(inp, kernel_shape, num_channels, num_kernels, suffix):
             filter_shape = [kernel_shape[0], kernel_shape[1], num_channels, num_kernels]
             weights = create_weights(shape=filter_shape, suffix=suffix)
             biases = create_biases(num_kernels, suffix=suffix)
             layer = tf.nn.conv2d(input=inp, filter=weights, padding='SAME', strides=[1, 1, 1, 1], name='conv_' + suffix)
             layer += biases
             layer = tf.nn.relu6(layer, name='relu_' + suffix)
             #layer = tf.nn.max_pool(layer, ksize=[1, 2, 2, 1], strides=[1, 2, 2,1], padding= 'SAME')
             return layer
```

Figure 9: Defining the convolution layer after which ReLU layer is used.

In the above figure 9, it depicts the steps we went through as an approach for this research. Here, we are defining the convolutional layer. The primary purpose of convolution is to find features in your image using the feature detector put them into a feature map and by having them in a future map, it still preserves the spatial relationships between pixels which is very important for us to know because if they are completely jumbled up. And Rectified Linear Unit is used in order to increase the non-linearity in the image.

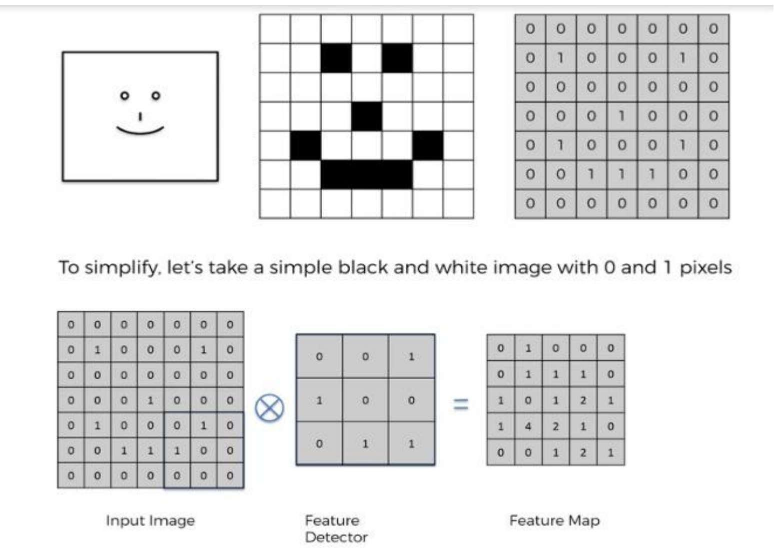


Figure 10: Example of how the convolutional layer works.

In the above figure 10, it depicts an example of how a convolutional layer works. In order to provide a proper explanation, we have chosen a simple image, the image will be converted into 0 and 1 pixels. And feature detector will be used to get the accurate/compressed pixel of image which will be stored as in the form of feature map. And in the next phase of convolution, we are using rectified linear unit in order to increase the non-linearity in the image.

Pooling

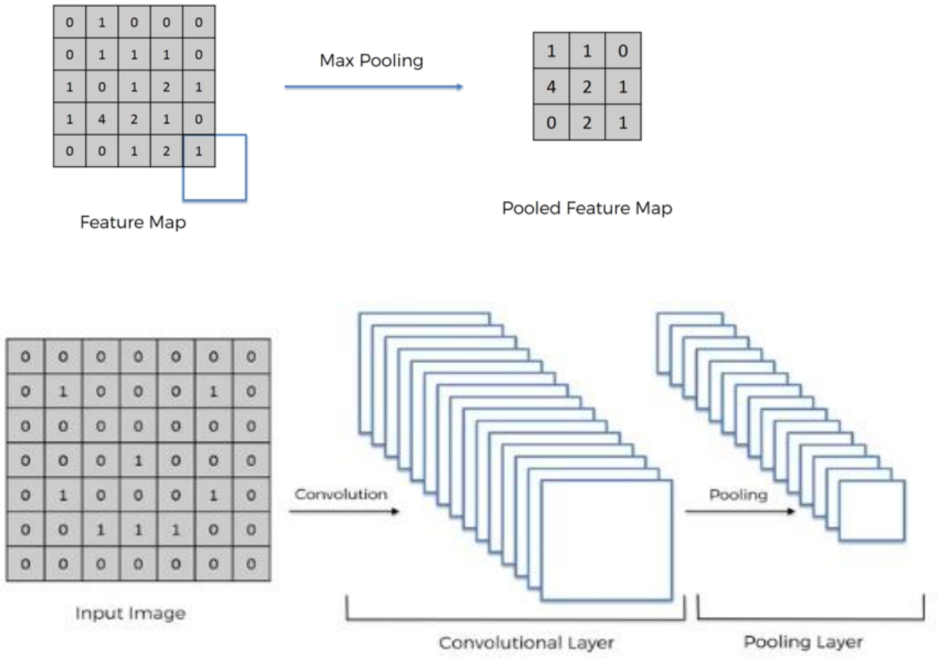


Figure 11: Example of how the pooling works.

In the above figure 11, it depicts an example of how pooling works. In order to reduce the distortion, we are using pooling. Pooling is nothing but taking out the integers from the feature map. Pooling is also referred as down sampling which indicates the function it does in image processing.

Flattening

```
In [35]: def flatten_layer(layer, suffix):
             layer_shape = layer.get_shape()
             num_features = layer_shape[1:4].num_elements()
             layer = tf.reshape(layer, [-1, num_features], name='flat_' + suffix )
             return layer
```

Figure 12: Defining the flattening function.

In the above figure 12, it depicts the steps we went through as an approach for this research. Here, we are defining the flattening function. In this process the pooled feature map will be flattened in order to go through next artificial neural network phase.

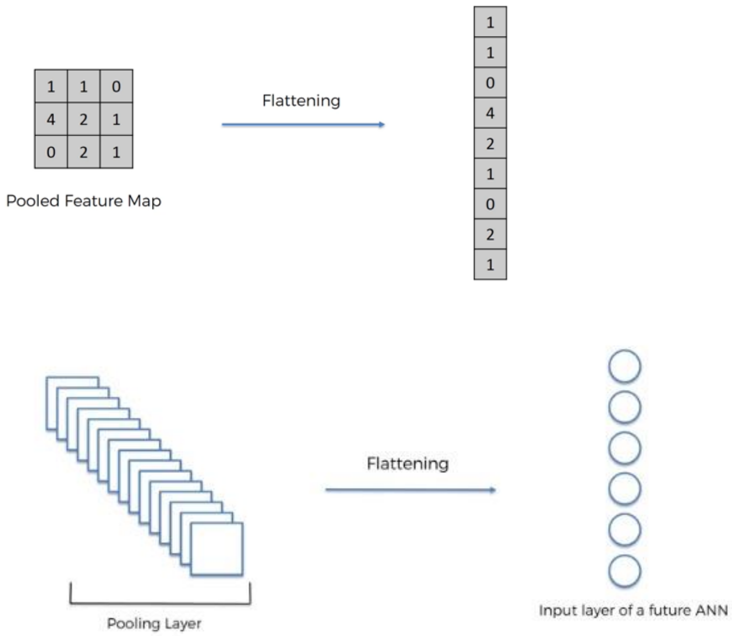


Figure 13: Example of how the flattening works.

In the above figure 13, it depicts an example of how a flattening works. In order to use artificial neural network in the next phase of our project, in flattening phase, the pooled feature map will be converted in a single column which will act as an input variable in next phase.

Dense Layer

```
In [36]: def dense_layer(inp, num_inputs, num_outputs, suffix, use_relu=True):
             weights = create_weights([num_inputs, num_outputs], suffix)
             biases = create_biases(num_outputs, suffix)
             layer = tf.matmul(inp, weights) + biases
             layer = tf.nn.relu(layer)
             return layer

In [37]: def rnn_cell(Win ,Wout, Wfwd, b, hprev, inp):
             h = tf.tanh(tf.add(tf.add(tf.matmul(inp, Win), tf.matmul(hprev, Wfwd)), b))
             out = tf.matmul(h, Wo)
             return h, out
```

Figure 14: Defining the dense layer for image processing.

In the above figure 14, it depicts the steps we went through as an approach for this research.

Here, we are defining the dense layer. The primary purpose of purpose of using dense layer in our project is to get high accuracy in predicting and captioning the image.

Figure 15: Example of how neural network works.

In the above figure 15, it depicts an example of how neural network works. In this phase each image will go through all the mentioned steps. The steps will be in the order of convolution, rectified linear units, pooling, flattening and neural network phase after which image will be recognised and captioning will be done.

Proposed development platforms

Our proposed development platform is python and we will begin by performing the data techniques like data discovery, exploratory analysis and data cleaning which will help us in getting a consistent data that can be used for analysing and classifying the images we have in our dataset.

As there is a computational challenge to handle large amounts of images and process them TPU's are used and coding is done in the Google Colab for easy sharing among the team members which has the Jupyter notebook framework representation. The programming languages like Python, Linux are to be used as of now. Regarding hardware to answer the complexities and to address the challenges from the processing the TPU's of google cloud are to be used. We have to research and explore more on the packages to be used for this project.

Software: Tensor flow, Google Colab, Python 3.7, Jupyter notebook.

Hardware: TPU, Google Cloud, 512 SSD, 16 GB RAM, Intel Core i7.

Data Analysis and Visualizations

In the dataset section the dataset images and the csv files with which we are training the model is well demonstrated and the below picture depicts the information where the image was given a description which recognises per picture wise. Were as the Figure 5 which is the structured dataset has the segments of the image which were given the description per segment where each image was divided into 4 segments and given the description accordingly and also the entire image was given a generic description which depicts the action or a play in the picture as a whole.

Figure 16: Showing the image and the description provided to the each image.

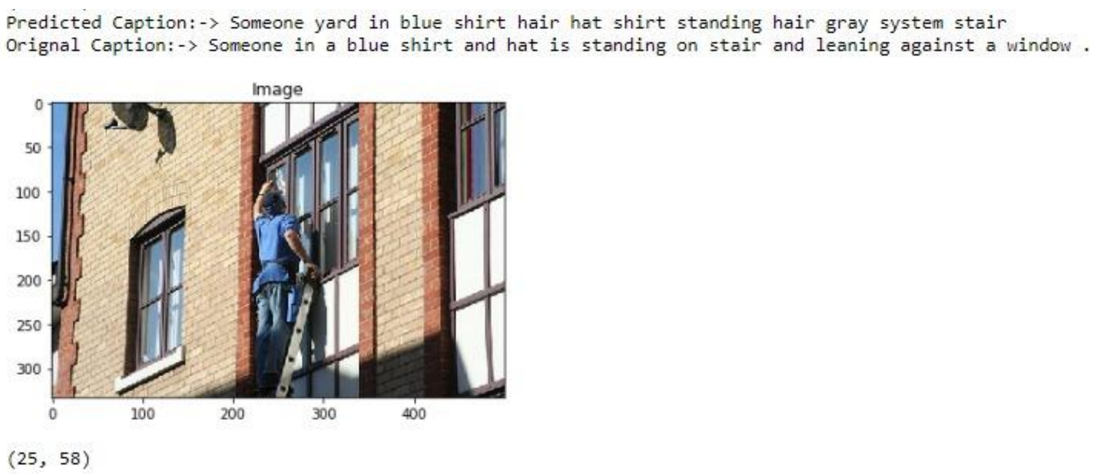
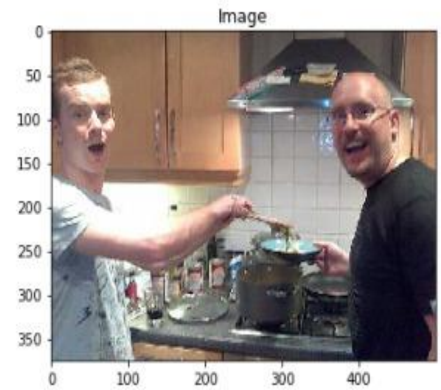


Figure 17: Showing the image with the actual and the predicted captions.

In Fig 17 it shows the original caption which was given to the image and the predicted caption which is predicted by our algorithm. So far for this particular picture the prediction was somewhat related whereas not into bits and pieces.

```
Predicted Caption:-> Two men stove window shirt gray Two climbing hands leaning one on operating giant pulley look hands the
are . black shirt A <s>
Orignal Caption:-> Two men one in a gray shirt one in a black shirt standing near a stove .
```




```
(29, 58)
```

Figure 18: Showing the misclassification of the image in the predicted caption.

In the Fig 18 the original picture was wrongly classified and as we can see in the image the background of the kitchen the tiles has the grill structure which is common for the window's and the machine is trained as so through most of the images in the dataset and thus classified but here the context and the picture is different.

```
After  100 iterations: Cost =  6.011802816390992 and Accuracy:  14.300495013594627 %
After  200 iterations: Cost =  3.54848427772522 and Accuracy:  34.00809735059738 %
Optimization finished!
And it's time to check
(29, 58)
Predicted Caption:-> Two young guys with shaggy are window the is out climbing entry hair black shirt are black <s> out syst
em hard of Several leaning a pulley an .
Orignal Caption:-> Two young guys with shaggy hair look at their hands while hanging out in the yard .
```


```
(25, 58)
```

Figure 19: Showing the accuracy of the model when run in the local systems without using any TPU's or GPU's etc.

In the Fig 19 there is some relation we can see to the picture and the predicted caption by the machine and to the original picture too. But the accuracy with the model was very less which is 34% for this entire process and training. But it took long hours and days to run these 200 iterations on the local machine and used Google Colab to further furnish the results with the support of TPU's, GPU's provided and to resolve the issues like time and computing complexity.

Figure 20: Showing the results of Google Colab where predictions are more accurate compared to before ones.

In the above Fig 20 it depicts the results of the model predictions and when compared with the original ones they had high correlation and relevance is also high in regards to the image and the description provided.

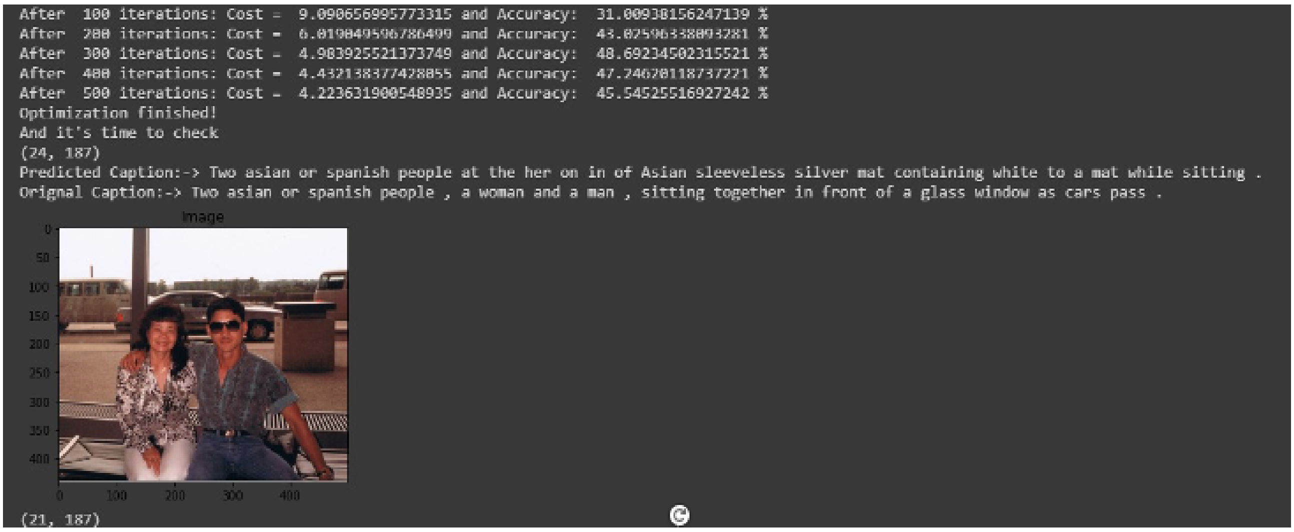


Figure 21: Showing the results and the accuracy of the model.

In the above figure showing the accuracy of the model which is close to 50% which is high and improved accuracy compared to the ones which are retrieved when used on local machines. The predictions are also matched and showing the high relevance in terms of describing the image.

Lessons Learnt

This project is a coagulation of both computer vision and the NLP and together used under the roof of Image Captioning using the R-CNN model where there is a very high spectrum to learn new things and in the boarder way. Initially, it was presumed to be very hard as computer vision itself involves many components and when NLP is an added addition to it making up such flavour is tough and definitely not as easy as said.

Lot of research is done and crawled on different research papers and web resources to understand the way to implement it and then again working on how well this can be implemented without any flaws and with limited free and available resources.

In class presentation topics were also helpful to decide on the methods and the approach to implement it and to understand the extended version of the concept very clearly and well. Research is the best way to deal and sustain in the world of data science and working with Professor like Dr. Liao is always a pleasure and always very informative and have many solutions and marvellous ideas with her which can definitely change the ones perception to the problem.

Future Work

Future work is a lot to be done as an addition enhancements to this project where as we have implemented this using the tensor flow and the experiments are to be conducted in the pyTorch and then work on it to increase the accuracy attained in this project. Due to the time limitation we couldn’t stretch this project more than what we imagined but the research and proceedings, experiments are yet to be done with lot more changes in the parameters of the model and the models suggested in the research paper considering the IOU, learning rate and many more metrics for improving the accuracy of the model attained in the project.

Conclusions

Encapsulating the entire project in few sentences, when started with the main objective which is implementing an image captioning model where the model is trained with the dataset which is the image dataset and then structured csv file which has the segmented data. Several thoughts which popped out of the concept of image captioning, were answered and bridged with the concepts of computer vision, NLP where the data was segmented into 4 parts and then given description to the 4 segments separately and given an index and reference number in the csv file and a separate description was given to an entire image and thus the caption is predicted for each image based on the R-CNN model which involves different layers of computations within it like pooling, flattening, dense layer and convolution layer to caption the image with appropriate description. The concepts like computer vision and image captioning are amalgamated to build a model where the caption of the image can be predicted by the model given the image from the dataset. Displaying both the manual feed to the model and the predicted ones.

Appendix

```
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 .
```

	A	B	C	D	E	F	G	H
1	image_name| comment_number| comment							
2	1000092795.jpg| 0| Two young guys with shaggy hair look at their hands while hanging out in the yard .							
3	1000092795.jpg| 1| White males are outside near many bushes .							
4	1000092795.jpg| 2| Two men in green shirts are standing in a yard .							
5	1000092795.jpg| 3| A man in a blue shirt standing in a garden .							
6	1000092795.jpg| 4| Two friends enjoy time spent together .							
7	10002456.jpg| 0| Several men in hard hats are operating a giant pulley system .							
8	10002456.jpg| 1| Workers look down from up above on a piece of equipment .							
9	10002456.jpg| 2| Two men working on a machine wearing hard hats .							
10	10002456.jpg| 3| Four men on top of a tall structure .							
11	10002456.jpg| 4| Three men on a large rig .							
12	1000268201.jpg| 0| A child in a pink dress is climbing up a set of stairs in an entry way .							
13	1000268201.jpg| 1| A little girl in a pink dress going into a wooden cabin .							
14	1000268201.jpg| 2| A little girl climbing the stairs to her playhouse .							
15	1000268201.jpg| 3| A little girl climbing into a wooden playhouse							
16	1000268201.jpg| 4| A girl going into a wooden building .							

Code   Python 3

Reading the dataset

[22]:
```
import os
train_images_list = os.listdir('Image_dataset/Image_dataset/images/')
```

[13]:
```
sample_size = 40
train_images_list = train_images_list[:sample_size]
```

[14]:
```
import os
import cv2
import numpy as np
import tensorflow as tf
from matplotlib import pyplot as plt
import random
```

[15]:
```
size = (256, 256)
num_channels = 3
```

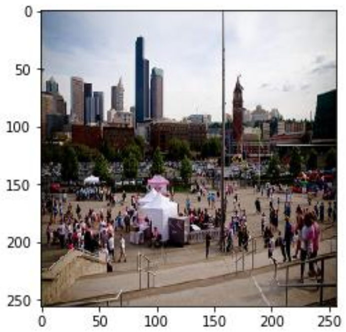
[16]:
```
train = np.array([None] * sample_size)
real_images = np.array([None] * sample_size)
```

[18]:
```
j = 0
for i in train_images_list:
    real_images[j] = np.array(plt.imread('Image_dataset/Image_dataset/images/'+ i))
    train[j] = np.array(plt.imread('Image_dataset/Image_dataset/images/' + i))
    j += 1
```

[19]:
```
j  = 0
for i in train:
    train[j] = cv2.resize(i, size)
    train[j] = train[j].reshape(1, size[0], size[1], num_channels)
    j += 1
```

[20]:
```
train = np.vstack(train[:])
```

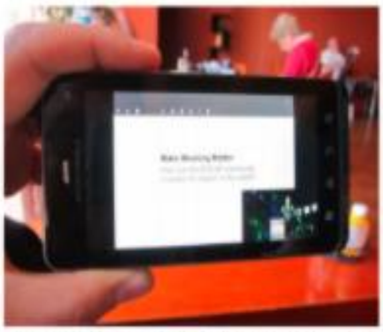
[21]:
```
plt.imshow(np.squeeze(train[0]))
plt.show()
```



Below is the image showing one of the images from the raw dataset.

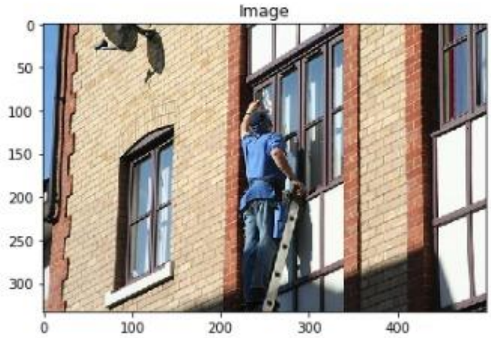
A baseball game in progress with the batter up to plate.

A brown bear standing on top of a lush green field.



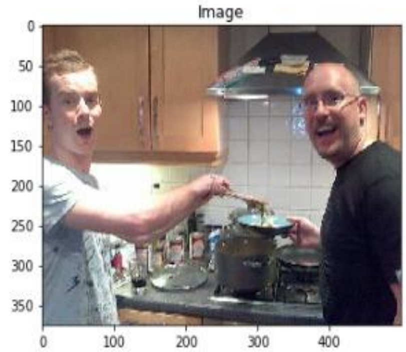
A person holding a cell phone in their hand.

```
Predicted Caption:-> Someone yard in blue shirt hair hat shirt standing hair gray system stair
Orignal Caption:-> Someone in a blue shirt and hat is standing on stair and leaning against a window .
```




```
(25, 58)
```

```
Predicted Caption:-> Two men stove window shirt gray Two climbing hands leaning one on operating giant pulley look hands the
are . black shirt A <s>
Orignal Caption:-> Two men one in a gray shirt one in a black shirt standing near a stove .
```




```
(29, 58)
```

```
After  100 iterations: Cost =  6.011802816390992 and Accuracy:  14.300495013594627 %
After  200 iterations: Cost =  3.54848427772522 and Accuracy:  34.00809735059738 %
Optimization finished!
And it's time to check
(29, 58)
Predicted Caption:-> Two young guys with shaggy are window the is out climbing entry hair black shirt are black <s> out syst
em hard of Several leaning a pulley an .
Orignal Caption:-> Two young guys with shaggy hair look at their hands while hanging out in the yard .
```


```
(25, 58)
```

```
Predicted Caption:-> Young blond man in a blue and yellow jacket smiling man stands person .
Orignal Caption:-> Young blond man in a blue and yellow jacket smiling while standing in front of a net .
```


```
(21, 187)
```

```
Predicted Caption:-> bride and groom walking side by side out to focus .
Orignal Caption:-> bride and groom walking side by side out of focus on pathway next to brick building .
```


```
(18, 187)
```

```
After  100 iterations: Cost =  9.090656995773315 and Accuracy:  31.00938156247139 %
After  200 iterations: Cost =  6.019049596786499 and Accuracy:  43.02596338093281 %
After  300 iterations: Cost =  4.983925521373749 and Accuracy:  48.69234502315521 %
After  400 iterations: Cost =  4.432138377428055 and Accuracy:  47.24620118737221 %
After  500 iterations: Cost =  4.223631900548935 and Accuracy:  45.54525516927242 %
Optimization finished!
And it's time to check
(24, 187)
Predicted Caption:-> Two asian or spanish people at the her on in of Asian sleeveless silver mat containing white to a mat while sitting .
Orignal Caption:-> Two asian or spanish people , a woman and a man , sitting together in front of a glass window as cars pass .
```


```
(21, 187)
```

References

[1] Argall, B., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and Autonomous Systems 57(5), 469–483 (2009) CrossRef Google Scholar.

[2] Lebret et al. (2015). *Phrase-based Image Captioning.* Retrieved from https://arxiv.org/abs/1502.03671.

[3] Karpathy, A., & Li, F. (2013). *Automated Image Captioning with ConvNets and Recurrent Nets.* Retrieved from https://cs.stanford.edu/people/karpathy/sfmltalk.pdf.

[4] Kulkarni et al. (2017). Understanding and Generating Simple Image Descriptions. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, No. 12, December. Retrieved from: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6522402

[5] Gmauf, T. (March20, 2018). Tensor flow records? What are they and how to use them? Retrieved from https://medium.com/mlreview/multi-modal-methods-image-captioning-from-translation-to-attention-895b6444256e.

[7] University of Illinois at Urbana Champaign. (n.d.). Retrieved from https://forms.illinois.edu/sec/1713398.

[6] Britz, D. (2016). *Attention and Memory in Deep Learning and NLP*. Retrieved from: http://www.wildml.com/2016/01/attention-and-memory-in-deep-learning-and-nlp/.