AI 825 VISUAL RECOGNITION(PART 2)

MINI PROJECT REPORT

Team ID- 33

Aditya Kaka(IMT2019002)

Amitha Reddy (IMT2019023)

Samhitha Perala(IMT2019521)

May 08, 2022

Contents

1 Question 1: Image Captioning

1.1 Introduction

This is called the CNN LSTM model, specifically designed for sequence prediction problems with spatial inputs, like images or videos. This architecture involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to perform sequence prediction on the feature vectors. In short, CNN LSTMs are a class of models that are both spatially and temporally deep and sit at the boundary of Computer Vision and Natural Language Processing. The dataset will be in the form (image, captions). The model to generate these captions follow an encoder-decoder architecture which uses an abstract image feature vector as an input to encoders. Object detection models, Convolutional Neural Network (CNN) and attention-based Recurrent Neural Network have helped a lot to improve the image caption results.

1.2 Flickr8K data

The Flickr8K dataset consists of almost 8000 32x32 colour images. The data-set is divided into a training batch of 6000 images, test batch with 1000 images and a validation set of 1000 images . The test batch contains exactly 1000 randomly-selected images. The training batch contains the remaining images in random order.

1.3 CNN-LSTM system Architecture

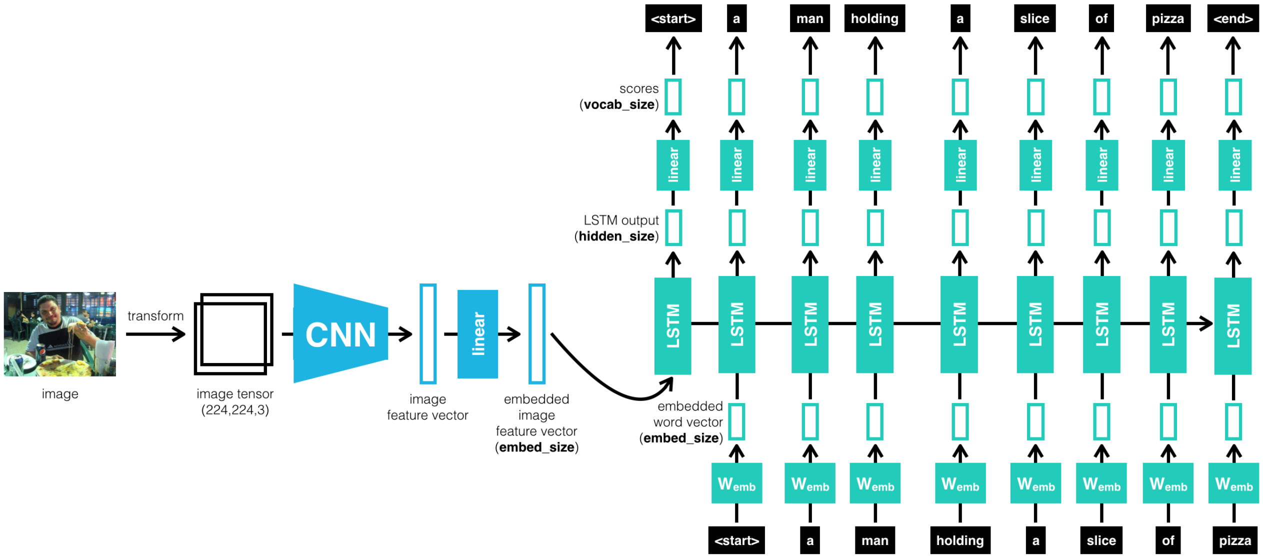


Figure 1: Image Caption Generator with CNN Architecture

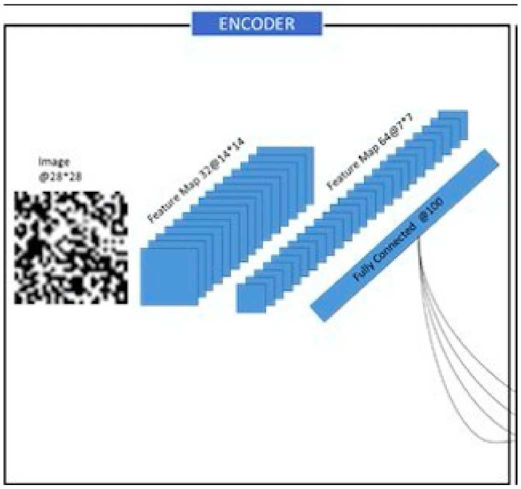


Figure 2: Encoder

Architecture: To extract and encode picture data into a higher dimensional feature space, we employed the ResNet pre-trained Convolutional Neural Network model as an encoder.

Convolutional Layer: This layer requires input data, a filter or feature detector, and a feature map, among other things. A 3x3 matrix is the most common filter size. The filter is then applied to a portion of the picture, and the dot product between the input pixels and the filter is determined. After that, the dot product is loaded into an output array. The filter then moves and continues the operation until the kernel has covered the entire picture. A feature map, activation map, or convolved feature is the ultimate result of a sequence of dot products from the input and the filter.

Pooling Layer: Dimension reduction is carried out via the pooling layer. The pooling process sweeps a filter across the whole input, similar to the convolutional layer, however this filter does not contain any weights. Max pooling picks the pixel with the highest value to send to the output array as the filter passes over the input. Average Pooling determines the average value inside the receptive field to send to the output array as the filter passes over the input.

Fully Connected Layer: This layer performs classification tasks based on the characteristics retrieved by the preceding layers and their various filters. While convolutional and pooling layers often utilise ReLu functions to categorise inputs, FC layers typically use a softmax activation function to provide a probability from 0 to 1. As the visual data passes through the CNN's layers, it begins to detect bigger components or forms of the object, eventually identifying the desired item.

For this task, the encoder needs to extract image features of various sizes and encode them into vector space which will be sent to a RNN in later stages. Since our CNN need not classify images but encode features, we removed the fully connected layers and the max pool layers at the end of the network. As we can see from the above picture, the circled layers are removed from our architecture.

1.4 LSTM Decoder

Long short-term memory networks are a type of recurrent neural network that expands the memory capacity. RNNs can recall inputs for a long time because to LSTMs. A gated cell can be compared to this memory. There are three gates in an LSTM: input, forget, and output. These gates control whether fresh input should be allowed (input gate), whether it should be deleted (forget gate), or if it should have an influence on the output at the current timestep (impact gate) (output gate).

The decoder must create picture captions word by word using LSTMs for our objective. The encoded image feature vectors from CNN and the encoded image captions produced in the data preprocessing step are sent into the decoder. We concatenate the embedded captions of all previous words and the encoded pictures and feed them to the LSTM to get the next state of the LSTM in each iteration of the LSTM network. Then, using the softmax activation function, fully connected layers may forecast the probability of current word embedding based on the current state and add them to the word embedding prediction matrix.

Loss function: The nature of our RNN output is a succession of word occurrences, and we apply Cross Entropy Loss to improve the quality of the RNN output. This is the most accurate way to assess the effectiveness of a classification model that produces a probability value between 0 and 1. We employed Adam optimizer, which is an adaptive learning rate optimization technique particularly intended for deep neural network training.

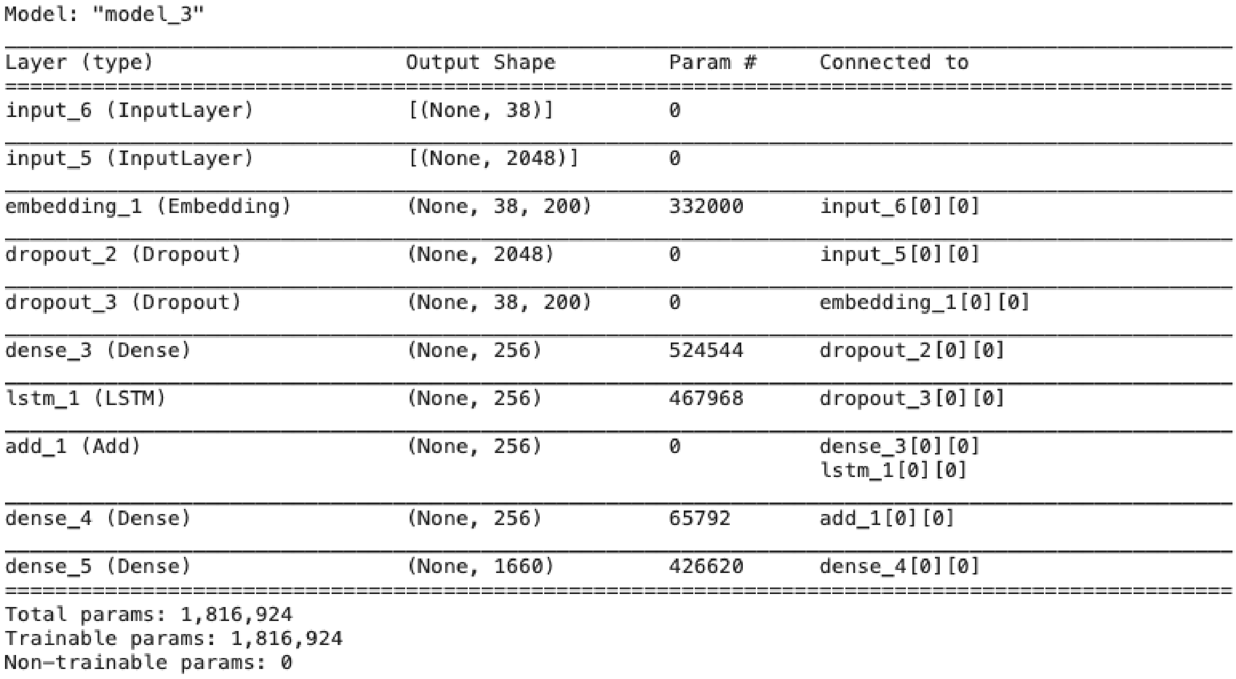
Here we need to convert the captions for training images into embedded captions, for that we need to create word embeddings. we can create either by using the corpus of data we have as captions or either we can use pretrained word embeddings like glove. Here we tried our image captioning using glove embeddings.

1.5 Beam Search

Decoding the most probable output sequence entails going through all of the potential output sequences and ranking them according to their likelihood. Beam Search is a popular strategy in this situation. The procedure is a best-first search strategy that iteratively evaluates the set of the k best sentences up to time t as candidates for generating sentences of size t + 1, keeping just the best k of them since this better approximates the likelihood of obtaining the global maximum.

1.6 Model Size Constraint

The image was first given to the Inception v3 model, and then the word embeddings were extracted from the pretrained glove. To avoid overfitting, we added a dropout layer before moving on to the dense layer. The dense layer's output is coupled to the LSTM. At the end, I created two linear dense layers that provide subtitles for the input image.



```
Model: "model_3"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_6 (InputLayer)            [(None, 38)]         0
__________________________________________________________________________________________________
input_5 (InputLayer)            [(None, 2048)]       0
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, 38, 200)      332000      input_6[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 2048)         0           input_5[0][0]
__________________________________________________________________________________________________
dropout_3 (Dropout)             (None, 38, 200)      0           embedding_1[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                 (None, 256)          524544      dropout_2[0][0]
__________________________________________________________________________________________________
lstm_1 (LSTM)                   (None, 256)          467968      dropout_3[0][0]
__________________________________________________________________________________________________
add_1 (Add)                     (None, 256)          0           dense_3[0][0]
                                                                 lstm_1[0][0]
__________________________________________________________________________________________________
dense_4 (Dense)                 (None, 256)          65792       add_1[0][0]
__________________________________________________________________________________________________
dense_5 (Dense)                 (None, 1660)         426620      dense_4[0][0]
==================================================================================================
Total params: 1,816,924
Trainable params: 1,816,924
Non-trainable params: 0
```

1.7 Training the model and Experiments

The train and test images have been given names in txt files. Those photographs were grabbed from the images folder and encoded, with the encoded images put into train and test image encodings, respectively, according to the names in the files. We stored these encodings into .pkl files to make loading them easier in the future. We utilised glove word embeddings to train our described model using these training picture encodings.

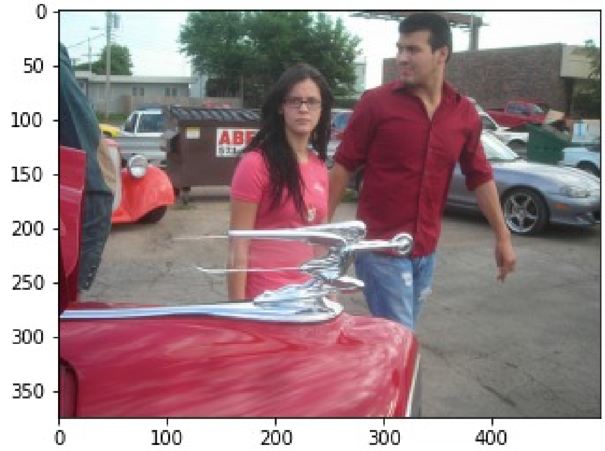
With category loss entropy as the loss function and an Adam optimizer with a batch size of 3, we trained our model for 25 epochs. We experimented with these values and saved the model, which will be utilised to perform the captioning next time without having to start from scratch.

1.8 Evaluation of test data using BLEU

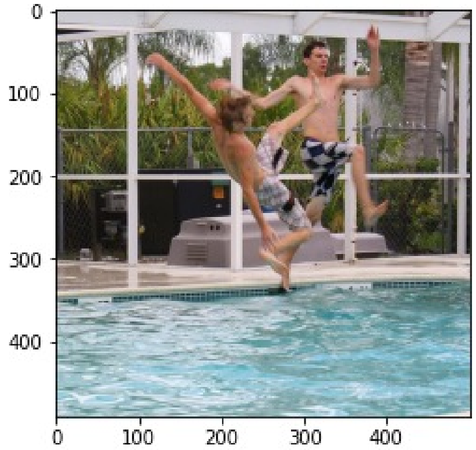
BLEU(bilingual evaluation understudy) will generate a number between 0 and 1. The score represents how similar the provided text is to the reference text, with values closer to 1 indicating more comparable texts. In actuality, a perfect score is impossible to get because a translation must precisely match the reference. Human translators are incapable of accomplishing this. The sentence-bleu() function in NLTK is used to compare a candidate sentence to one or more reference sentences.

We have 5 different captions for each test image provided. so these 5 captions are given as a reference array and the caption given by the system has been passed into this function to evaluate bleu score.

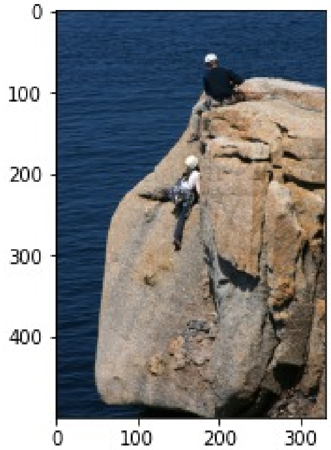
1.9 Observation and Results



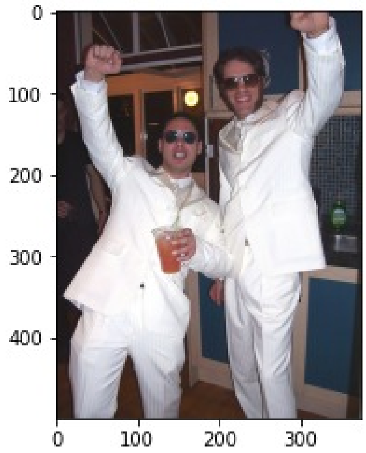
```
Greedy Search: a man in a red shirt is walking past a city street
Beam Search, K = 3: a group of people sit on a bench in front of a building
Beam Search, K = 5: a group of people sit on a bench in front of a bus
Beam Search, K = 7: a group of people sit on a bench in front of a bus
Beam Search, K = 10: a group of people sit on a bench in front of a bus
```



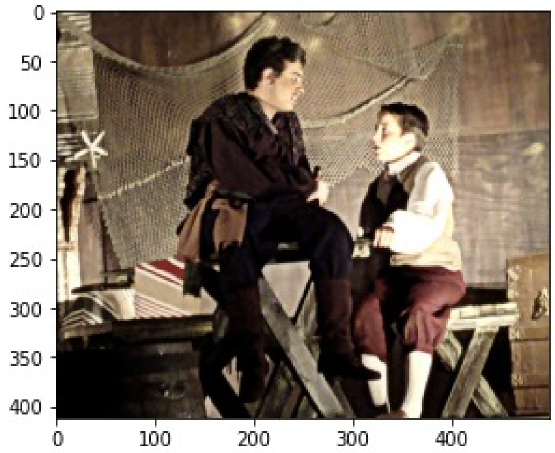
```
Greedy Search: a boy in a pink swimsuit is jumping into a pool
Beam Search, K = 3: a young boy in a pink bathing suit jumps into a swimming pool
Beam Search, K = 5: a young boy jumps into a swimming pool
Beam Search, K = 7: a young boy jumps into a swimming pool
Beam Search, K = 10: a young boy in a bathing suit jumps into a pool
```



```
Greedy Search: a man is climbing a rock face
Beam Search, K = 3: a man sits on the edge of a rock formation
Beam Search, K = 5: a man climbing a rock face
Beam Search, K = 7: a man climbing a rock face
Beam Search, K = 10: a man climbing a rock face
```



```
Greedy: a group of people are sitting on a bus
Beam Search, K = 3: a group of people are sitting on a bus
Beam Search, K = 5: a group of people are sitting on a bus
Beam Search, K = 7: a group of people are sitting on a bench in front of a crowd
Beam Search, K = 10: a group of people are sitting on a bench in front of a bus
```



```
Greedy Search: a man in a red shirt is standing by a store
Beam Search, K = 3: a man wearing a black hat and hat is standing next to a store
Beam Search, K = 5: a group of people are sitting on a bench in front of a crowd
Beam Search, K = 7: a group of people are sitting on a bench in front of a bus
Beam Search, K = 10: a group of people are sitting on a bench in front of a bus
```

BLEU score

- for k = 3: 0.4126796654279265
- for k = 5: 0.3932466292843753
- for k = 7: 0.3291055377149112

2 Question 2: Modified Image Captioning

2.1 Introduction

Modified system 1 will perform analysis on the trained model we have obtained after solving question 1. It's about how we designed efficient experiments to bring out the 'weakness' in the existing system.

2.2 Language Bias

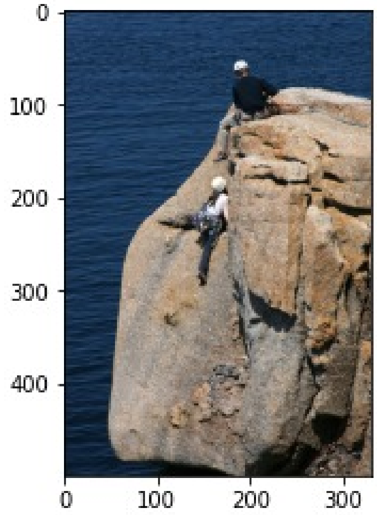
Most machine learning algorithms are known to capture and exploit training data biases. Some biases are useful to learning, while others are detrimental. Image captioning algorithms, in particular, have a tendency to exaggerate biases in training data (for example, if a word appears in 60% of training sentences, it may be predicted in 70% of test sentences). Due to an over-reliance on the learnt prior and image context, this can result in inaccurate captions in domains where unbiased captions are sought or required. As a consequence, when machines analyse language to learn word embeddings, women appear in near proximity to terms like family and arts compared to men, whereas men appear in close proximity to words like profession, science, and technology.The stereotypical associations exist for gender, race, age, and intersections among these characteristics. The production of gender-specific caption words (e.g. man, woman) based on a person's appearance or image context have to be studied deeply .

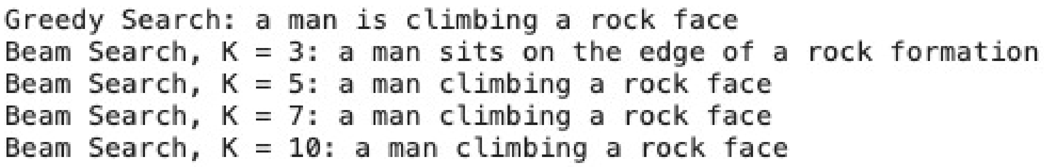
When gender evidence is blocked in a scenario, we provide a new Equalizer model that favours equal gender probability and confidence predictions. Instead of using contextual clues to produce a gender-specific prediction, the resulting model is compelled to gaze at a person. The Appearance Confusion Loss and the Confident Loss, which make up our approach, are general losses that can be introduced to any description model to reduce the effects of undesirable bias in a description data set. When describing photos with individuals and specifying their gender, the suggested model has to give lower inaccuracy than previous model.

Equalizer is based on the following hypotheses: if there is no evidence to support a gender decision in an image, the model should be confused about which gender to predict (enforced by an Appearance Confusion Loss term), and if there is evidence to support a gender decision in an image, the model should be confident in its prediction (enforced by a Confident Loss term). In addition to the traditional cross entropy loss, we train our model to maximise these additional losses.

2.3 Observations and Results

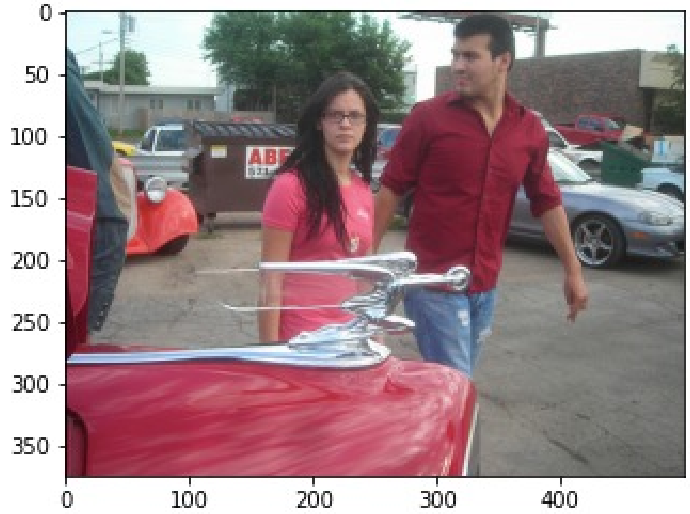
- The person climbing the rock is a woman but the model detects as a man.

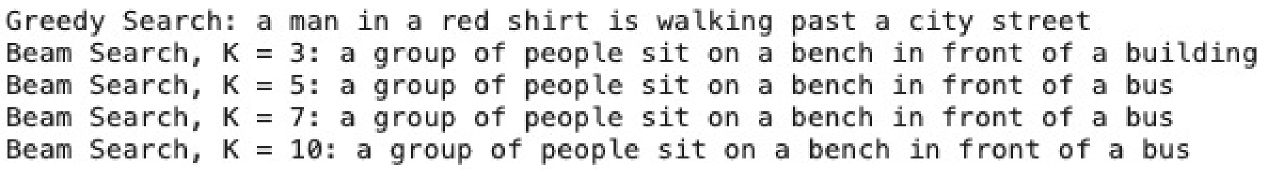




```
Greedy Search: a man is climbing a rock face
Beam Search, K = 3: a man sits on the edge of a rock formation
Beam Search, K = 5: a man climbing a rock face
Beam Search, K = 7: a man climbing a rock face
Beam Search, K = 10: a man climbing a rock face
```

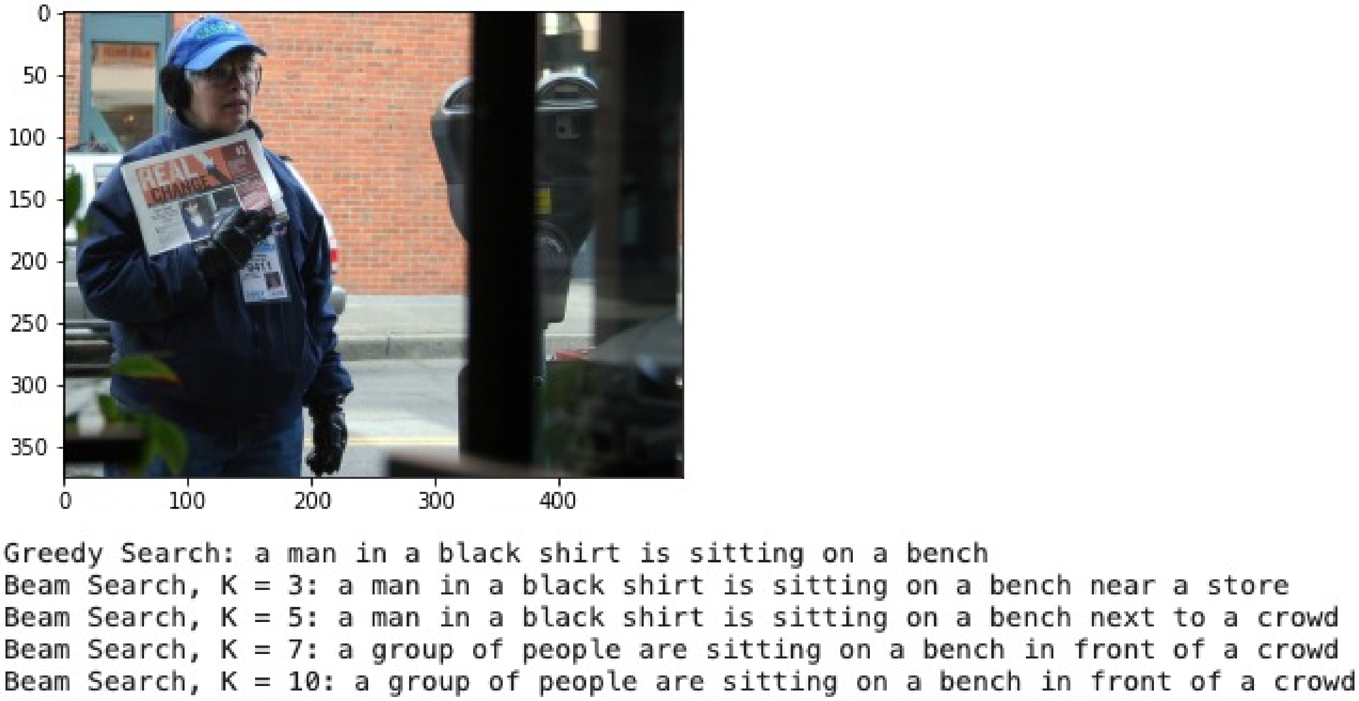
- There are both man and woman but the model detects only man.





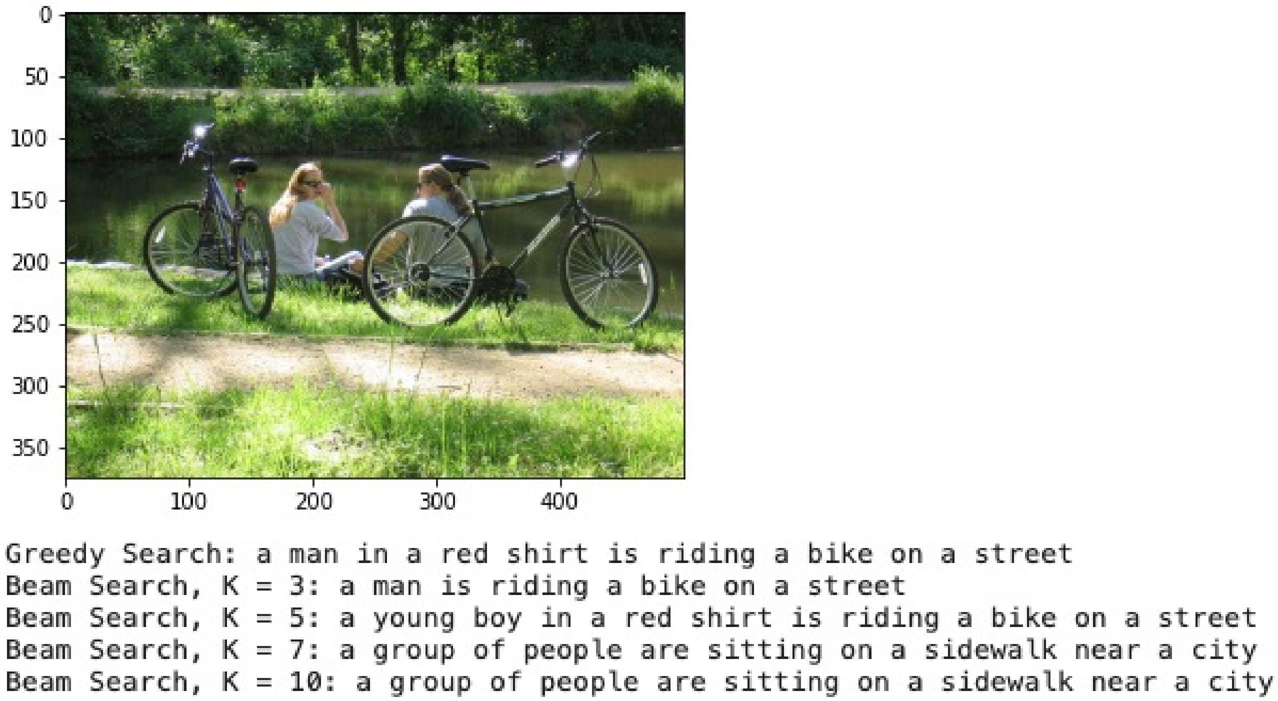
```
Greedy Search: a man in a red shirt is walking past a city street
Beam Search, K = 3: a group of people sit on a bench in front of a building
Beam Search, K = 5: a group of people sit on a bench in front of a bus
Beam Search, K = 7: a group of people sit on a bench in front of a bus
Beam Search, K = 10: a group of people sit on a bench in front of a bus
```

- This is a woman holding a newspaper but the model detects it as a man.



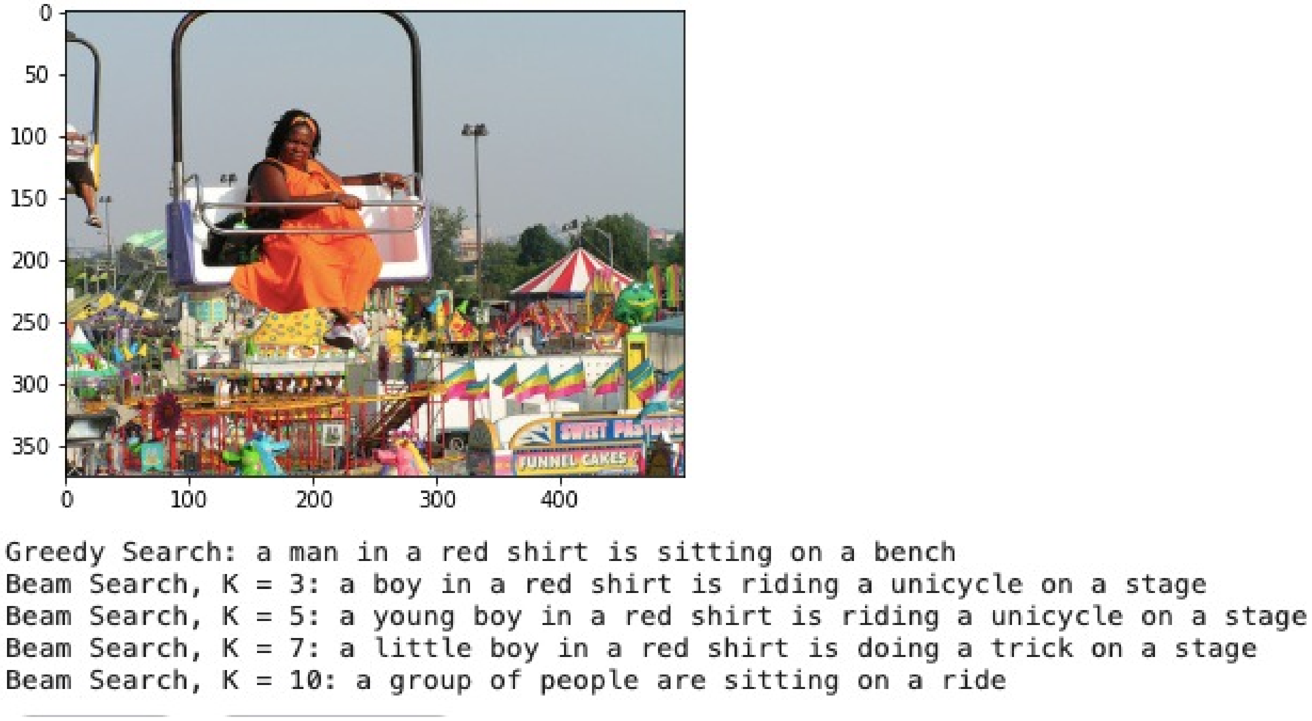
```
Greedy Search: a man in a black shirt is sitting on a bench
Beam Search, K = 3: a man in a black shirt is sitting on a bench near a store
Beam Search, K = 5: a man in a black shirt is sitting on a bench next to a crowd
Beam Search, K = 7: a group of people are sitting on a bench in front of a crowd
Beam Search, K = 10: a group of people are sitting on a bench in front of a crowd
```

- This image contains two women sitting along with their cycles but model detected them as men



```
Greedy Search: a man in a red shirt is riding a bike on a street
Beam Search, K = 3: a man is riding a bike on a street
Beam Search, K = 5: a young boy in a red shirt is riding a bike on a street
Beam Search, K = 7: a group of people are sitting on a sidewalk near a city
Beam Search, K = 10: a group of people are sitting on a sidewalk near a city
```

- The image has a woman sitting but model detects as a man



```
Greedy Search: a man in a red shirt is sitting on a bench
Beam Search, K = 3: a boy in a red shirt is riding a unicycle on a stage
Beam Search, K = 5: a young boy in a red shirt is riding a unicycle on a stage
Beam Search, K = 7: a little boy in a red shirt is doing a trick on a stage
Beam Search, K = 10: a group of people are sitting on a ride
```

3 Citations

- https://www.brookings.edu/research/detecting-and-mitigating-bias-in-natural-language-proc
- https://medium.com/@stepanulyanin/captioning-images-with-pytorch-bc592e5fd1a3