Caption Generation Using Long Short Term Memory Networks and Basic Recurrent Neural Network

James Zhao
jjz005@ucsd.edu

Kartik Nighojkar
knighojk@ucsd.edu

Mario Lopez
m9lopez@ucsd.edu

Abstract

In this project, Long Short Term Memory(LSTM) networks and vanilla recurrent neural networks (RNN) were used to generate captions. In order to generate captions, images were first passed into a pre-trained CNN to extract features, and later repeatedly passed into one of the recurrent neural network types. In this project, we tested three primary RNN architecutures: An LSTM, a vanilla RNN, and an LSTM where the encoded image is repeatedly passed along with the embedded words (Architecture 2).

With the default LSTM, we achieved a test loss of 1.4581, Bleu1 score of 65.54, and a Bleu4 score of 7.49. With the vanilla RNN, we achieved a test loss of 1.4928, a Bleu1 score of 65.338, and a Bleu4 score of 7.197. With Architecture 2, we achieved a test loss of 1.4377, a Bleu-1 score of 66.84, and a Bleu-4 score of 8.36.

1 Introduction

In this project, we attempted to generate captions for various images. In order to do this, we begin with a modified version of a pre-trained model, resnet50, as our image encoder. Then we constructed three types of decoders, one using Long Short Term Memory(LSTM) networks, another using recurrent neural network(RNN) and lastly consturcted a second architecture where the embedded image was passed into the LSTM at each time stop. Using these three models, we were able to generate captions.

In order to train our model, we utilized a subset of the COCO dataset. The COCO dataset (Common Objects in Context) consist of images of every day objects usually performing an action, along with several image captions associated with that image.

In addition to evaluating test loss(loss between predicated caption and 'true' caption), we also evaluated Bleu-1 and Bleu-4 scores. Bleu scores generally represent how closely a hypothesis sequence matches a set of reference sequences. Bleu-1 score utilizes 1-grams(single words) to determine the score, while Bleu-4 scores analyze length-4 sub-sequences of words. When performing image generation, we tested two generation techniques: deterministic generation and stochastic generation. We tended to achieve the best results when using stochastic generation. With the default LSTM, we achieved a test loss of 1.4581, Bleu1 score of 65.54, and a Bleu4 score of 7.49. With the vanilla RNN, we achieved a test loss of 1.4928, a Bleu1 score of 65.338, and a Bleu4 score of 7.197. With Architecture 2, we achieved a test loss of 1.4377, a Bleu-1 score of 66.84, and a Bleu-4 score of 8.36.

2 Related Work

Pytorch documentation for torch.nn.LSTM, torch.nn.RNN, torch.nn.Embedding, and torch.multinomial.

Pycocotools for loading images and obtaining reference captions from the COCO dataset.

3 Architecture of the Models

The baseline model consisted of a modified pre-trained CNN(resnet50) as an encoder, and an LSTM with a history of one timestep as a decoder. The pre-trained encoder has its final fully-connected layer replaced with a custom linear layer, so the network outputs a vector with dimensions equal to the embedding size.

The LSTM has a default embedding size of 300, and a hidden size of 512. In order to pass in words from teacher-forcing or generated outputs, the LSTM also contains an embedding layer. This layer embeds a single value (the index of a word in the vocabulary) into its length-300 corresponding embedding, and this embedding layer is learned as the model trains. In order to turn LSTM hidden states into an understandable output, the LSTM also has an additional linear layer from the hidden size to the vocabulary size. The output of this linear layer(a length - vocab size array) contains values, which when passed into a softmax activation function, would result in a probability distribution representing the next character in the sequence (given the hidden states of the previous time step and the embedded input).

On the first pass of the model, the 256x256x3 rgb image is passed into the encoder. After passing through the pre-trained model and custom linear layer, the cnn should output a length-300 embedding. Afterwards, the embedding is passed directly into the lstm, (the previous hidden state is, by default, zero), the previous states are stored, and the output is passed through the (embedding size x vocab size) linear layer to produce an output. Afterwards, the model is teacher-forced, so the expected/label output is passed in to the embedding layer as the input on the next time step. This is

repeated for each other character in the teacher sequence.

When our model is in generation mode(by setting the "self.generation" flag), the encoding proceeds as normal. However, instead of teacher forcing, our model utilizes the output of the LSTM as the input (which is re-embedded). When performing deterministic generation, the word with the highest value(and thus highest probability) is used as the next input. When performing stochastic generation, each output is scaled by a flat amount(temperature), and the softmax of the scaled outputs is sampled using a multinomial distribution.

When performing generation/training with RNN's and Architecture 2, the training/testing procedure is relatively the same. In the case of RNN's, the LSTM is replaced with an RNN. In the case of architecture 2, the embedded input images are also concatenated with the embedding of the teacher forced/generated word.

It may seem like the RNN model would need more epochs in order to fully train due to having less 'memory' from previous timesteps compared to an LSTM (leading to less coherent outputs). However, we generally observed that both the LSTM and RNN were almost fully trained within 10 epochs. This could possibly be due to the relatively straightforward and simple grammar structure of our captions. There isn't neccessarily a need to look very far back in time(which LSTM's excell at), so the weakness of vanilla RNN's isn't as prevalent.

3.1 Tuning Hyperparameters for Baseline

When tuning hyperparameters for the Baseline models, we used the provided hyperparameters as our defaults:

Hidden Size: 512, Embedding Size: 300

Train Loss	Validation Loss
0.9724	1.4400

Tuning Embedding Size

Embedding Size	Train Loss	Validation Loss
450	0.9357	1.4541
150	1.0427	1.4445

When embedding size is increased, the training loss decreases but the validation loss increases. This could indicate the model might be overfitting. When the embedding size decreased, both the training and validation losses increased, suggesting that the embedding might be too small. Thus, we will use the default embedding size as our best hyperparameter.

Tuning Hidden Size

Hidden Size	Train Loss	Validation Loss
784	0.8195	1.4747
256	1.1810	1.4367

When tuning the hidden size, increasing the hidden size once again led to a decreased training loss but increased validation loss, indicating the model may be overfitting. When the hidden size decreased, the training loss increased but the validation loss, interestingly, decreased. This discrepancy could have been entirely due to randomness, but because of the high training loss, we will use the default hidden size as our best hyperparameter.

Generating captions with Different Temperatures with Default Hyperparameters:

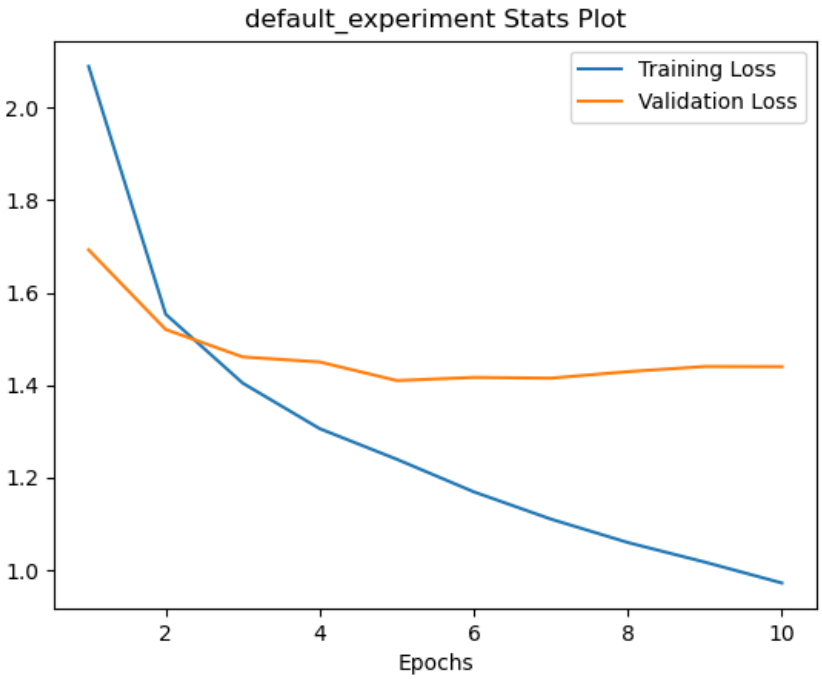


Figure 1: Loss plot of the default hyperparameters/best set of hyperparameters

Temperature	Test Loss	Bleu1	Bleu4
0.01	1.4542	65.63	7.43
0.1	1.4581	65.54	7.49
0.5	1.4481	62.12	5.97
1.0	1.4490	47.94	2.55
2.0	1.4666	12.06	0.79
Deterministic	1.4650	65.65	7.41

All of the test losses are relatively similar, which makes sense because all of these temperatures were tested on the same trained model. Small variations in the order that captions are tested can result it different loss values (from different sequence lengths & different amounts of padding).

In terms of bleu scores, it appears that having a lower temperature generally leads to higher bleu1 and bleu4 scores. However, it's hard to determine what range exactly produces the best bleu1 and bleu4 scores, as these scores are generated as a result of randomness. Nevertheless, based on on our output, having a temperature in the $[0.01, 0.1]$ range generally resulted in good outputs, and deterministic generation (which could be reinterpreted as stochastic generation with a VERY small temperature) led to good results as well. We will maintain 0.1 as our ideal temperature as it led to relatively consistent outputs while still introducing a decent amount of randomness into caption generation.

Results of Best Hyperparameters:

Embedding Size = 300, Hidden Size = 512, Stochastic Generation, Temperature = 0.1

Test Loss	Bleu1	Bleu4
1.4581	65.54	7.49

Our best combination of hyperparmeters ended up being the default hyperparameters we were provided. It seems as though the model's embedding size and hidden size already struck a good balance

between being larger enough to obtain high complexity, without being too large and running the risk of overfitting. In almost every case, either increasing or decreasing one of the hyperparameters resulted in a decreased training or validation loss. Furthermore, the graphs also show signs of potential overfitting, as the validation loss curve tends to slightly slope upwards, suggesting that the model is just about to start overfitting.

3.2 Tuning Hyperparameters for RNN

When tuning for the RNN models, we used the provided hyperparameters as our defaults:

Embedding size = 300, Hidden size = 512, stochastic generation, temperature = 0.1, epochs = 10

Tuning the epochs of Stochastic RNN:

Embedding Size = 300, Hidden Size = 512, Stochastic Generation, Temperature = 0.1

Epochs	Train Loss	Validation Loss
8	1.5031	1.5075
10(base)	1.4928	1.5048
12	1.5232	1.5207
14	1.5503	1.5416

When tuning the epochs for the Stochastic RNN, as the base epochs increased past 10, we saw an increase in both train and validation losses. This result suggests that the base value of 10 epochs was enough epochs for the RNN to train, as adding more epochs causes the model to overfit(early stopping was not implemented).

Tuning the Temperature of Stochastic RNN:

Embedding Size = 300, Hidden Size = 512, Stochastic Generation, epochs 10

Temperature	Test Loss	Bleu1	Bleu4
1	1.5070	45.859	2.416
.1(base)	1.4994	65.0365	7.135
.01	1.5066	65.286	7.620
0.001	1.5225	65.293	7.635

When tuning the temperature, as the base temperature decreased we observed a slight increase in the test losses, but also observed an increase in both BLEU-1 and BLEU-4 scores. When the temperature was too high, however, the test loss increased and both bleu scores significantly decreased. This is likely due to the fact that the a higher temperature leads to a less sharper and more variable distribution.
Based on our results, we observed that a temperature of 0.1 strikes a good balance between having variability while still outputting high bleu-1 and bleu-4 scores.

Tuning the Embedding of Stochastic Rnn:

Embedding Size = 300, Stochastic Generation, epochs 10

Embedding Size	Train Loss	Validation Loss
200	1.5140	1.5053
300(base)	1.4928	1.5048
400	1.5120	1.4998

When we decreased the embedding size, both the train loss and validation losses increased, suggesting the model might not be rigorous enough to produce quality captions. When the base embedding value increased to 400 the Train loss increased, but the Validation loss decreased, suggesting that the model may be over fitting. Thus, we found the default embedding size to strike a good balance between adding complexity to our model and not having too many parameters to the point of overfitting.

Tuning the Hidden of Stochastic RNN:

Hidden Size = 512, Stochastic Generation, epochs 10

Hidden Size	Train Loss	Validation Loss
256	1.5224	1.5224
512(base)	1.4928	1.5048
1024	1.5120	1.5952

When tuning the hidden size, we observed similar results to tuning embedding size. When decreasing the hidden size, both losses increased, suggesting the model may not be complex enough. When increasing the hidden size, the validation loss significantly increased, suggesting the model may be overfitting. Because we achieved the best results with a hidden size of 512, we will utilize that hidden size as our best hyperparameter.

By taking the best RNN generation of Stochastic and the best hyperparmaeters, the following model is generated:

Results of Best Hyperparameters:

Embedding Size = 300, Hidden Size = 512, Stochastic Generation, Temperature = 0.1

Test Loss	Bleu1	Bleu4
1.4928	65.338	7.197

Compared to the LSTM, the RNN is produces slightly worse outputs. The test loss was slightly higher, and both the BLEU-1 score of 65.54 and BLEU-4 score of 7.49 were slightly lower. This could be due to the fact that RNN's are only capable of 'looking back in time' a designated amount of time steps, whereas the LSTM models, through the forget/write/read gate mechanisms, can selectively remember or forget words/outputs from farther time steps. Thus, the LSTM's additional complexity enabled it to outperform the RNN.

In terms of loss curves, both the LSTM and RNN produced relatively similar graphs. Both plots featured validation loss curves that flattened out at around 10 epochs, suggesting that both models were just about to begin overfitting.

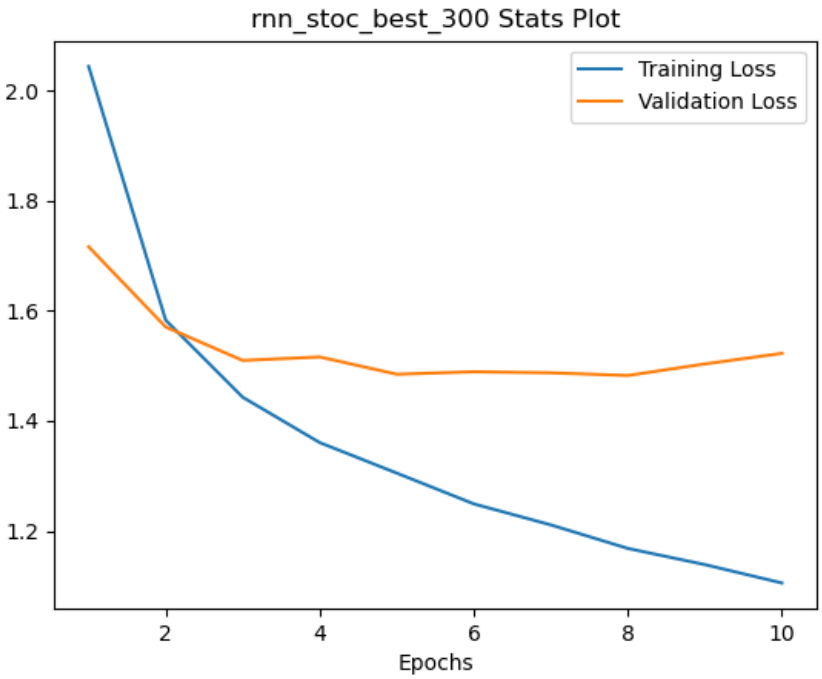


Figure 2: Loss plot of the best hyperparameters for the RNN model.

3.3 Architecture 2

In the second architecture, the input to the LSTM cell at each timestep of the model was not only the word embedding, but the input image concatenated to the word. For this reason, the input size to the LSTM model had to be doubled (as the size of the embedded image feature vector was the same as the size of the embedded word and they were concatenated), and the embedded word <pad >was passed as the input word at timestep $t = 0$.
Using the same hyperparameters as the baseline model, the following loss curves were obtained:

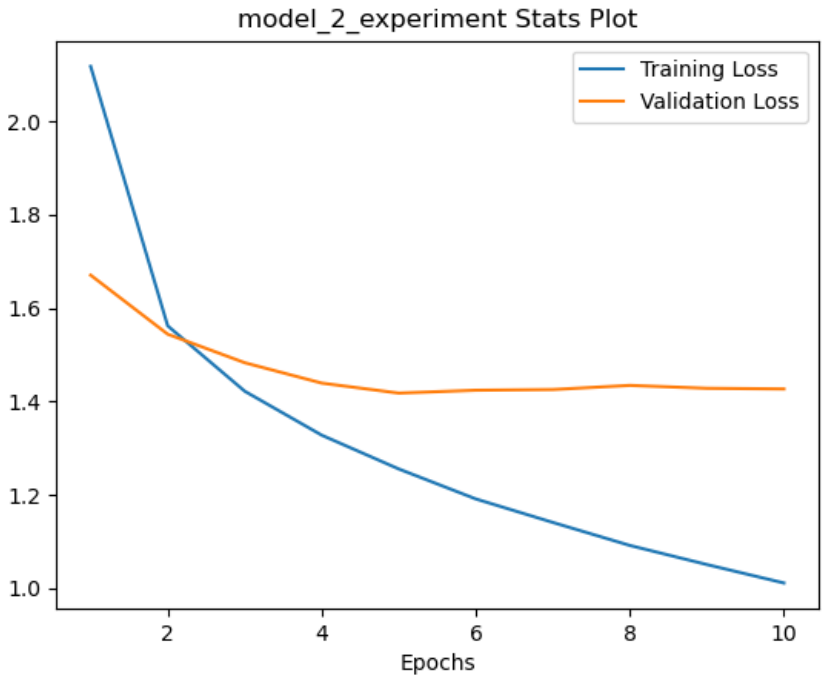


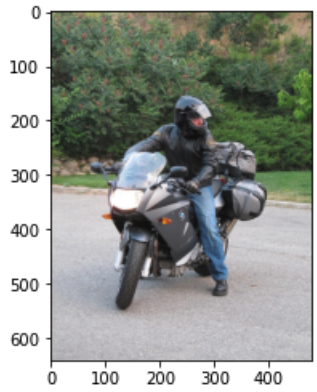
Figure 3: Loss plots when running Architecture 2 using baseline's best hyperparameters. The test loss was 1.4377, the BLEU-1 score was 66.84, and the BLEU-4 score was 8.36

Compared to the baseline model loss curves (Figure 1), the train and validation loss curves for the architecture 2 model are quite similar; the minimum train loss eventually approaches 1.0 and the validation loss seems to level our at around 1.5. However, the test loss for the architecture 2 model was slightly lower than the test loss for the best performing baseline model. Although the improvement could be entirely due to random variance, it also could be because the model 'remembers' the embedding of the original image at every timestep. Thus, the generated captions would be more closely aligned with the image that was passed in. In the case of the default LSTM and RNN models, the original image embedding could be 'lost' after several time steps, and the generated captions could subsequently deviate from the original image's features.

4 Caption Visualization

The model that performed best in terms of generating the minimal test loss was the model using architecture 2 with the best performing hyperparameters from the baseline model. Below are images along with actual and generated captions from the model, split into a group of well predicted captions and badly predicted captions.

4.1 Well Predicted Captions



Actual captions:
a person riding a motorcycle on a road
a person on a motorbike has a helmet on .
a man with a leather jacket sitting on a motorcycle in a street .
a male in a black jacket is on his motorcycle and some bushes and grass behind him
a man making a turn on a motorcycle and looking back .

Generated caption: a man riding a motorcycle down a street .

Actual captions:
old rusted transit train cars sit on the tracks .
an old train makes its way down the track in the country .
a train traveling down the track, with power lines in the back .
a small train is traveling on the railroad .
an orange and white train makes its way down the track .

Generated caption: a train is on the tracks in a field

Actual captions:
a stop sign sits on a street corner
somebody replaced the y with a r in an all way stop sign .
a stop all-way sign which someone has changed to read ”stop all-war”
a red stop sign at a street intersection
the stop sign on the roads is an all-war stop sign .

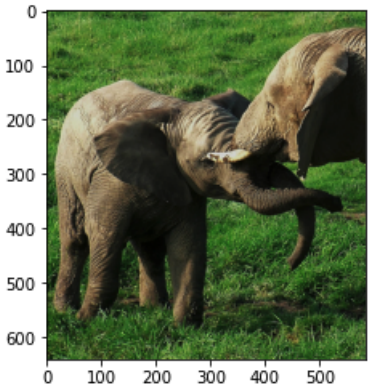
Generated caption: a stop sign with a sticker attached to it .

Actual captions:
a train pulls up to an empty platform .
a yellow train pulling into a train station next to a platform .
a yellow train is currently stopped at the tracks .
a passenger train leaving the train station that is now empty . .
a commuter train approaching an outdoor railway station

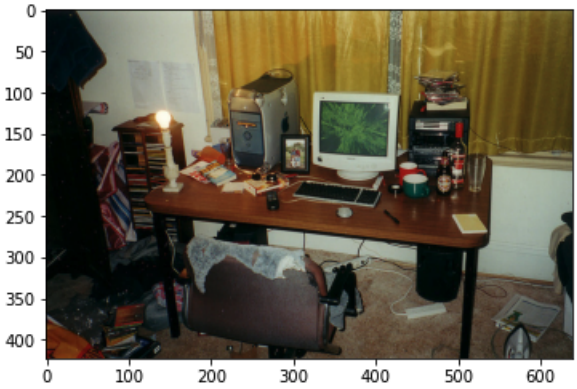
Generated caption: a train is on the tracks at a station .

Actual captions:
two hot dogs on buns sitting on top of a paper plate .
a plate with two hot dogs on buns and a bagel .
two hot dogs have been placed next to two bagels .
a plate full of two hot dogs with different toppings, and a bagel .
a plate of hot dogs and other food .

Generated caption: a hot dog with ketchup on a plate .

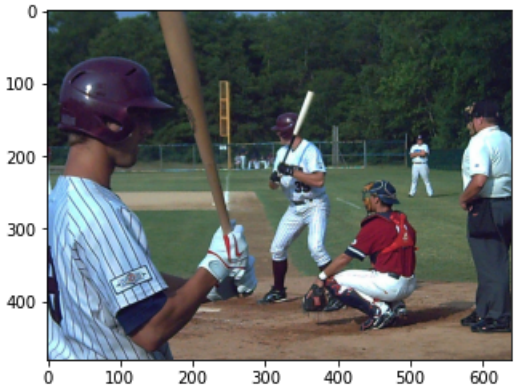


Actual captions:
a mother elephant nuzzling her young, baby elephant .
small elephant leans towards a larger elephant on a field of grass .
two elephants on a field of green of green grass
two elephants standing on a lush green hillside .
mom and baby elephant cuddling in a pasture .
Generated caption: a baby elephant walking in the grass near a forest .



Actual captions:
a cluttered computer desk in a messy room .
a messy desk with a computer, cups, glasses, bottles, books on the desk and the floor .
a desk with a torn chair, keyboard, monitor and mouse .
a table top with a computer on top of it
room with a lamp on a wooden computer desk .

Generated caption: a desk with a computer and a keyboard

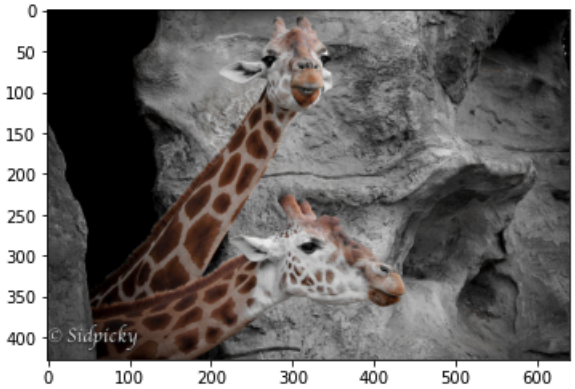


Actual captions:
a baseball player waits for a pitch while an umpire looks on .
a group of baseball players playing a game of baseball .
a batter is practicing while his teammate is at the plate .
a kid playing baseball at the ball park trying to get on base .
one batter is up and another batter is waiting his turn .

Generated caption: a group of men playing baseball on a field .

Actual captions:
a bathroom with a large tub next to a sink .
a bathroom with large faux marble tiles and a built in bathtub .
a black bathroom in someones house with a tiny tub
a bathroom laid in gray marble looks cold and uninviting .
bathtub in a very fancy stone tiled bathroom .

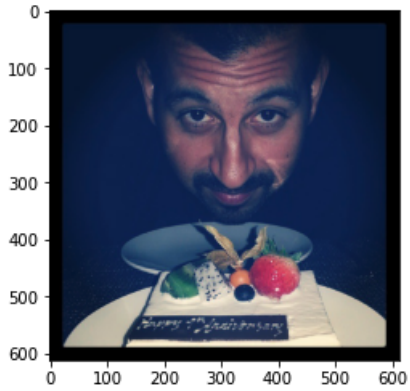
Generated caption: a bathroom with a toilet and a sink .



Actual captions:
a close up of two giraffes with a rock background
two giraffes are peeking out between some rocks .
a couple of giraffe standing in some rocks .
two giraffe standing next to each other near a stone mountain .
two giraffes standing by some rocks with one making a silly face

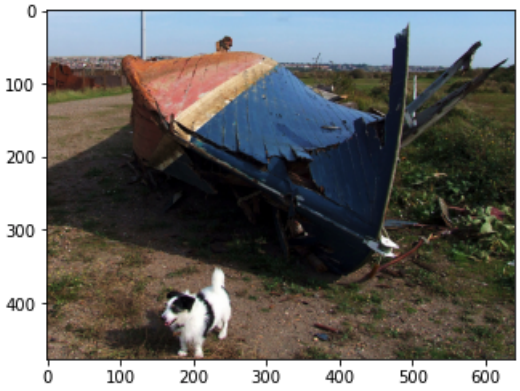
Generated caption: a giraffe standing next to a rock wall .

4.2 Badly Predicted Captions



Actual captions:
a man peers over a small plate behind a napkin lined with pieces of fruit .
a man with beard leaning over a plate and a cake .
a man posing behind a plate of food .
a man poses for a happy anniversary photo .
a man has his chin on a blue plate near some paper and utensils .

Generated caption : a woman is holding a piece of cake with a knife .

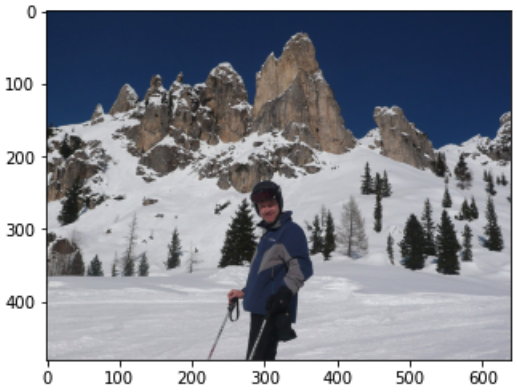


Actual captions:
a small white and black dog standing next to a busted up ship .
a black and white puppy standing in front of a blue and red boat skeleton .
a small black and white dog standing in front of a demolished red and blue boat .
a black and white dog stands in a deserted location next to a dilapidated building
a old wooden boat that is destroyed on a grassy area near a black and white dog .

Generated caption: a dog is laying on the ground with a surfboard .

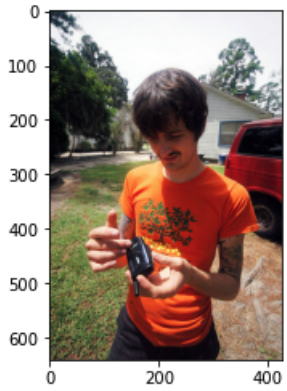
Actual captions:
a white and red traffic sign on top of a metal pole .
a street sign with an 80 in the middle
a sign that reads ”80” is perched on a hilltop .
a road sign is displaying a number in front of trees and fields .
sign with the number ”eighty” set against bright blue sky .

Generated caption: a stop sign with a sign on top of it .



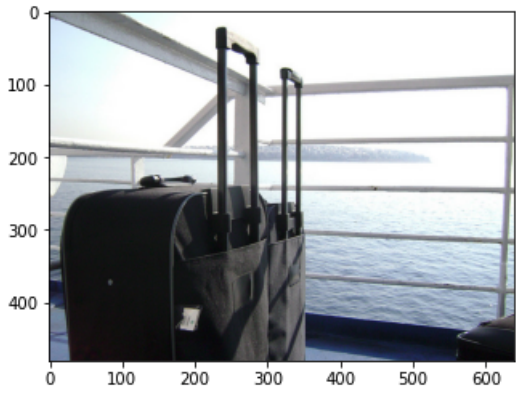
Actual captions:
man in blue and grey jacket on skis in front of a mountain .
a man on skiis standing in front of trees and a mountain .
a person on skis wearing a blue and grey coat
a man walking along in the snow on skis .
a smiling man stands on a snowy hill with some ski poles

Generated caption: a man in a red jacket skiing down a snowy slope .



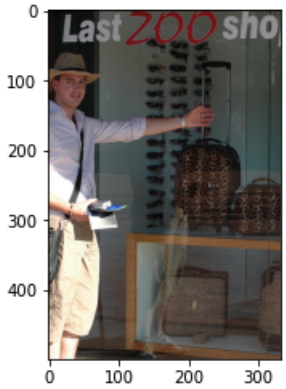
Actual captions:
a man that is standing up holding a cellphone .
a man holding a cell phone in his hands .
a young man in an orange shirt holds a cell phone outdoors .
a young man holding a walkie talkie while wearing an orange shirt .
a man holds onto a black device in the yard .

Generated caption: a man is holding a cell phone to his ear .



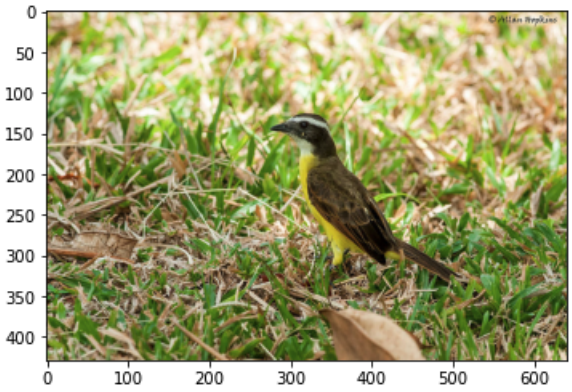
Actual captions:
a couple of bags of luggage sitting on top of a ship .
a couple of black bags are on a boat
a number of suitcases on the boat in the sea
two suitcases sit with the handles up, on a boat .
some black suitcases standing by a white fence

Generated caption: a black bench on a pier next to a fence



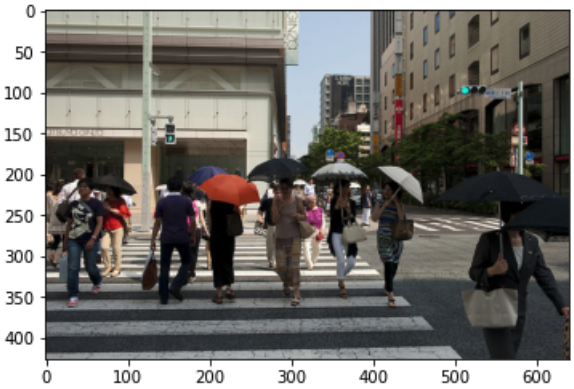
Actual captions:
a man that is standing by some bags
a man is pointing out a brown suitcase .
a man standing in front of a zoo shop window
a man is holding his hand out to point at a window with suitcases on each shelf .
there is a man that is posing in front of a store window
a man standing next to a bat of luggage .

Generated caption: a man standing in front of a large mirror .



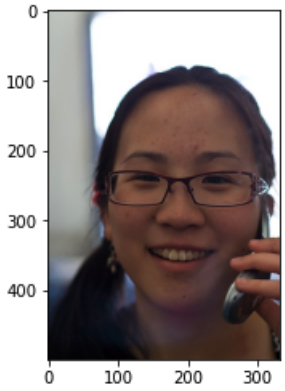
Actual captions:
a yellow and brown bird perched on a leaf on top of a grass covered field .
yellow and green bird sitting in tall grass .
a yellow bellied bird with a white stripe on his head
a black and yellow small bird in the grass
a brown, white and yellow bird standing in the grass

Generated caption: a small bird sitting on a table with a small bird on it .



Actual captions:
many people holding umbrellas walking across a street
a bunch of people walking on the street with umbrellas .
people crossing the street in a cross walk holding umbrellas in the city .
shoppers carrying umbrellas against the sun in an oriental city
the people are walking the streets with their umbrellas up .

Generated caption: a woman walking down a street with a dog .



Actual captions:
smiling girl wearing glasses, with a cellphone up to her ear .
a woman wearing glasses talking on a cell phone .
a girl with glasses smiles while talking on the phone .
a woman standing on a cell phone in a room .
a girl is holding a cell phone to her ear .

Generated caption: a man with a beard and a tie in a suit .