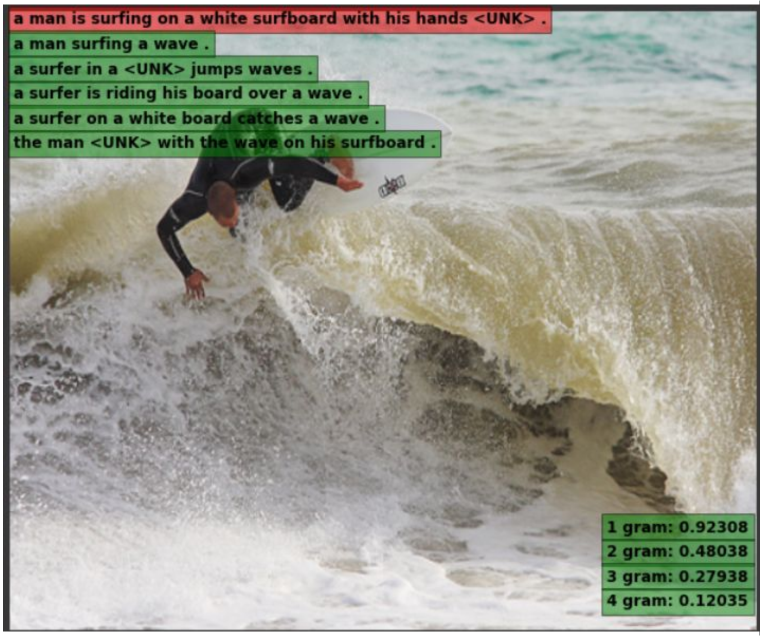
Yoav raytsfeld
Almog amsalem

Before we start

Before we start, we want to show you the performance of the model we will build. it will give you an idea of what you would expect by the end of the report. As you can see, our model can generate a caption without errors for some images below:



our model BLEU score on test dataset:

	1-gram	2-gram	3-gram	4-gram
count	1000.000000	1000.000000	1000.000000	1000.000000
mean	0.488320	0.239379	0.129634	0.078967
std	0.150254	0.166527	0.130623	0.094952
min	0.125000	0.029525	0.017825	0.012918
25%	0.383352	0.067420	0.040332	0.029847
50%	0.477688	0.217930	0.079548	0.046118
75%	0.571429	0.333372	0.169062	0.082487
max	1.000000	1.000000	1.000000	1.000000

Flickr8k Dataset

We will use Flickr8k dataset to train our model. The dataset contains 8000 of images each has 5 different captions. Having more than one caption for each image is necessary because an image can be described in many ways.

For example as we can see from the image, and respectively the 5 different captions:

1. A child in a pink dress is climbing up a set of stairs in an entryway .
2. A girl going into a wooden building .
3. A little girl climbing into a wooden playhouse .
4. A little girl climbing the stairs to her playhouse .
5. A little girl in a pink dress going into a wooden cabin .

As we can see, there are some different interpretation among them:

- caption 1 is more descriptive than the other examples
- How the toddler is labeled as a “child ”, “little girl” or just a “girl”.
- “playhouse “ vs “wooden cabin ”

Having different captions helps a model catch these subtleties and be able to generalize better.

Those 8000 images are divided into 3 sets:

1. Training set (6000 images): We use it for training our model.
2. Validation set (1000 images): We use it for assessing our model’s performance while training.
3. Test set (1000 images): We use it for assessing our model’s performance after training.

Data Preprocessing

Image Preprocessing

The images we received were in a variety of shapes, however (as an introduction to the training part) we are using a pre-trained model and by that have to resize those images to size 224X224. in addition we need ImageNet mean and std values.

transforms.Normalize((0.485, 0.456, 0.406), (0.229, 0.224, 0.225))

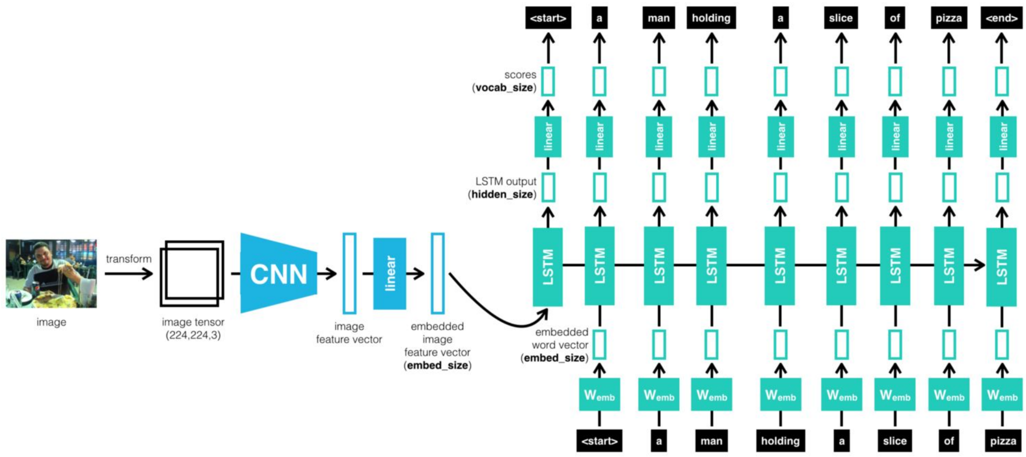
Caption Preprocessing

Since we are working on a text dataset and the neural network needs a numbers sequence as input we need to create a vocabulary of all the unique words present across all the 8000*5 (40,000) image captions in the data set. This means we have 8763 unique words. But since we are creating a predictive model, we would not like to have all the words present in our vocabulary but the words which are more likely to occur or which are common. This helps the model become more robust to outliers and make less mistakes. We set the threshold to minimum 4 to be adding to the vocabulary. This vocabulary will give us an access to the words bank and the ability to encoding/decoding them to indexes. as written below, we add another 4 words to our vocabulary for flagging to the model:

- “<SOS>” - start of sentence
- “<EOS>” - end of sentence
- “<PAD>” - To create an equal series length
- “<UKN>” - For all words that have not passed a certain threshold value

Training Phase

Model architecture



To do the image captions task we need to ensemble two types of neural networks. The first is CNN. We used a ResNet 152 and it doen the features extraction actions. but we had to do minor adjusting to it by replacing his classification head with an embedding layer and freezing the other trainable parts of it. this will generate a sequence input to the next net, the LSTM net which tries to predict the next word from a given sequence

Loss Function and Optimization

Lstm represents a probability distribution over all words, so we can use loss function for multiclass classification problems: cross-entropy loss. To minimize the loss, the optimizer needs the gradient of the loss function which tells the optimizer how much and in which direction it needs to adjust each model's parameter. We used an ADAM optimizer.

Batch Training

Due to batches, we need to append some captions with "padding words" in order that captions within a batch have the same length. These padding words have to be encoded in such a way they won't increase the loss. So we encode them as <PAD> words and then ignore them.

Model parameters

Learning rate -------- 1e-3

Batch size -------------- 32

Epochs ----------------- 150

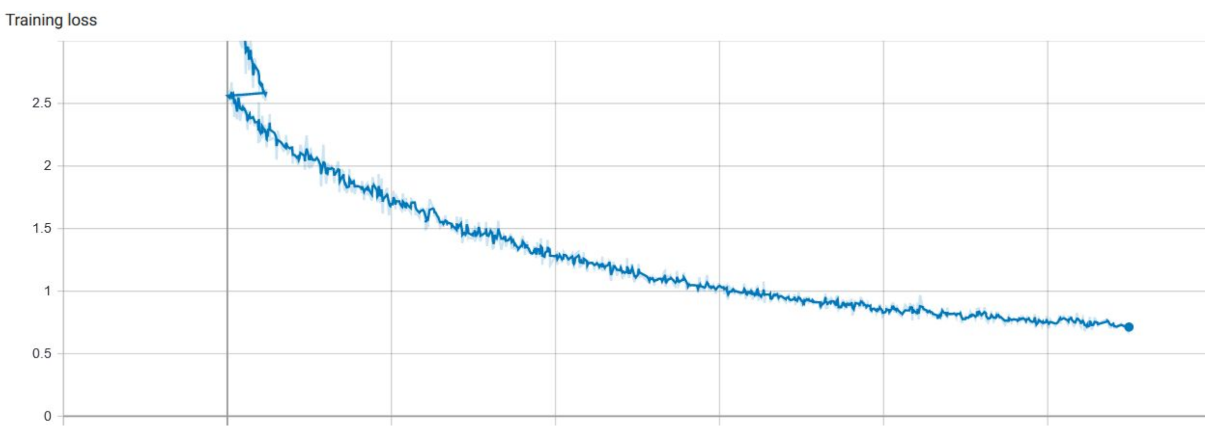
Dropout rate ----------- 50%

Embedding size ------ 512

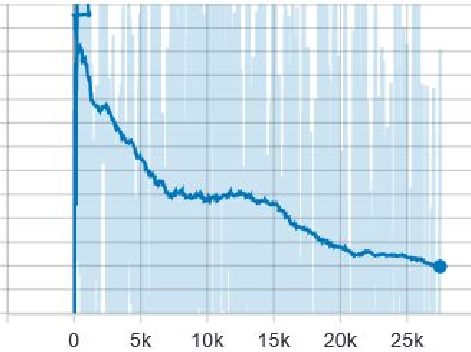
LSTM hidden size --- 512

LSTM num of layers - 2

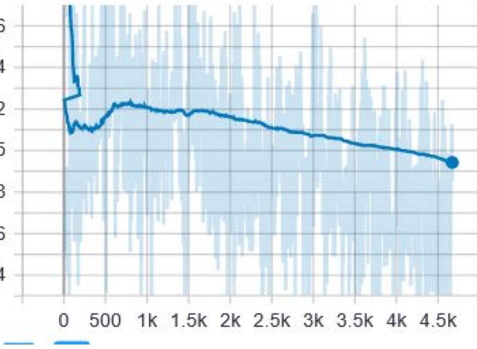
Graphs and visialition



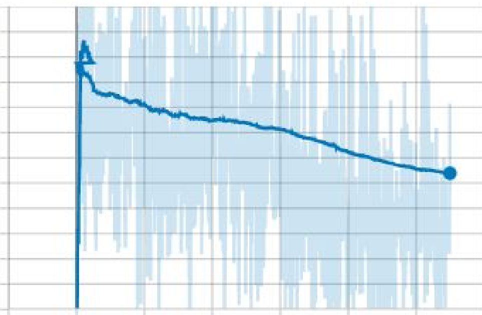
1-gram bleu_train



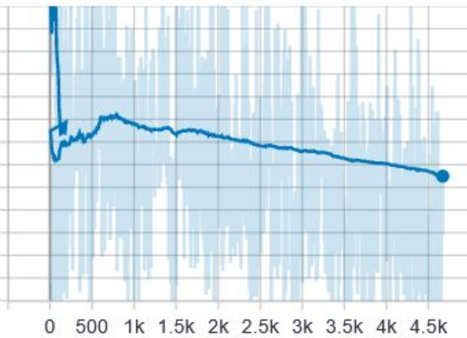
1-gram bleu_val



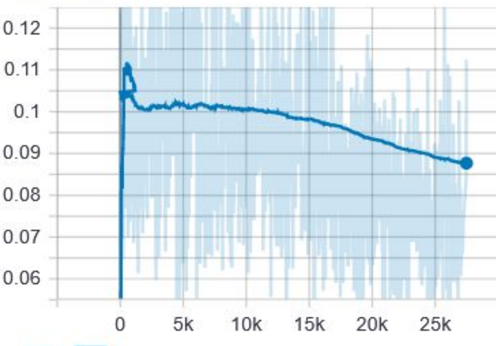
2-gram bleu_train



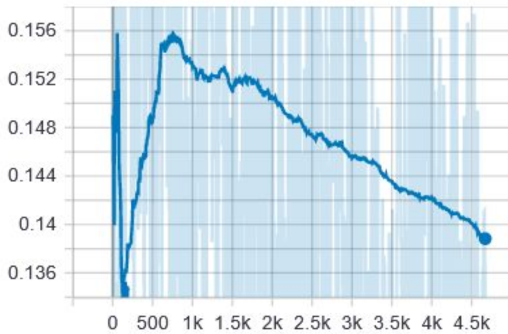
2-gram bleu_val



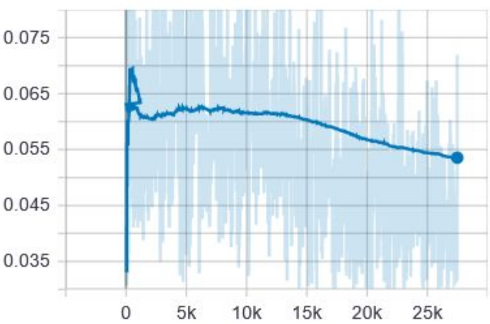
3-gram bleu_train



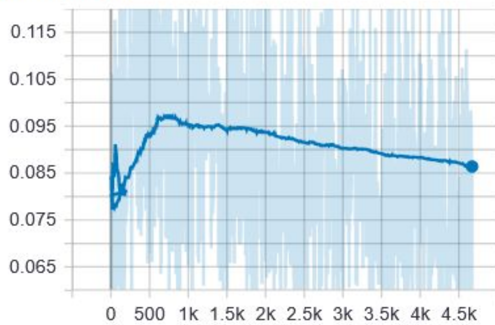
3-gram bleu_val



4-gram bleu_train



4-gram bleu_val



Inference Phase

At the inference phase we expect the model to generate the most probable caption given an image. For this task we had to make some changes to the logic of our training:

1. Feed the image into the Image Embedding Model (CNN) which will produce an image embedding of the image.
2. The image embedding will be the input for the Sequence Model (LSTM). It will yield the probability distribution of the first word.
3. Choose the first word by selecting the word with the highest probability in that distribution.
4. The word embedding will be the input for the LSTM at the next iteration . It will yield the probability distribution of the second word.
5. Repeat a similar process (3 - 5) until the end-of-sentence word (EOS) is generated or the maximum of length is reached.

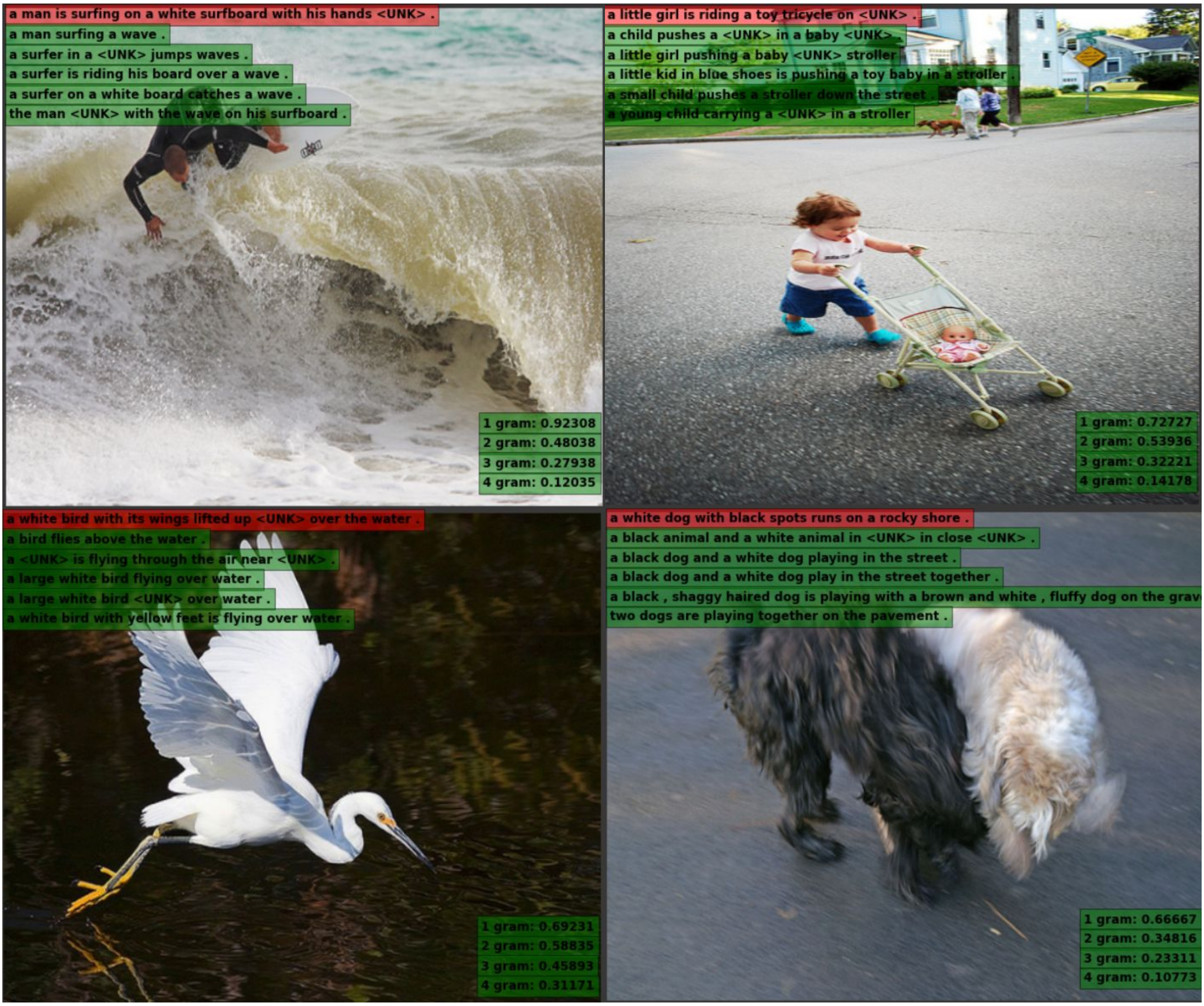
Quantitative Assessment

BLEU metric commonly used in sentences translation problems but it can be used in image captions problems as qualitative assessment of the ground truth caption and the model output. Basically, they assess a generated caption by comparing it to the reference captions.
We can classify the captions generated by our model in the three main categories.
By comparing the 1 gram score of the output to the sorting range.
The low threshold set to be mean - std
The upper threshold set to be mean + std
And the classes which we define are:

- "**Unsuccessful prediction**" - the prediction was not related to actual captions. Score less than low threshold.
- "**Partly success**" - as the name suggests, the model was able to caption something in the image. Score between low and upper thresholds.
- "**Accurate description of the image**". Score above the upper threshold.

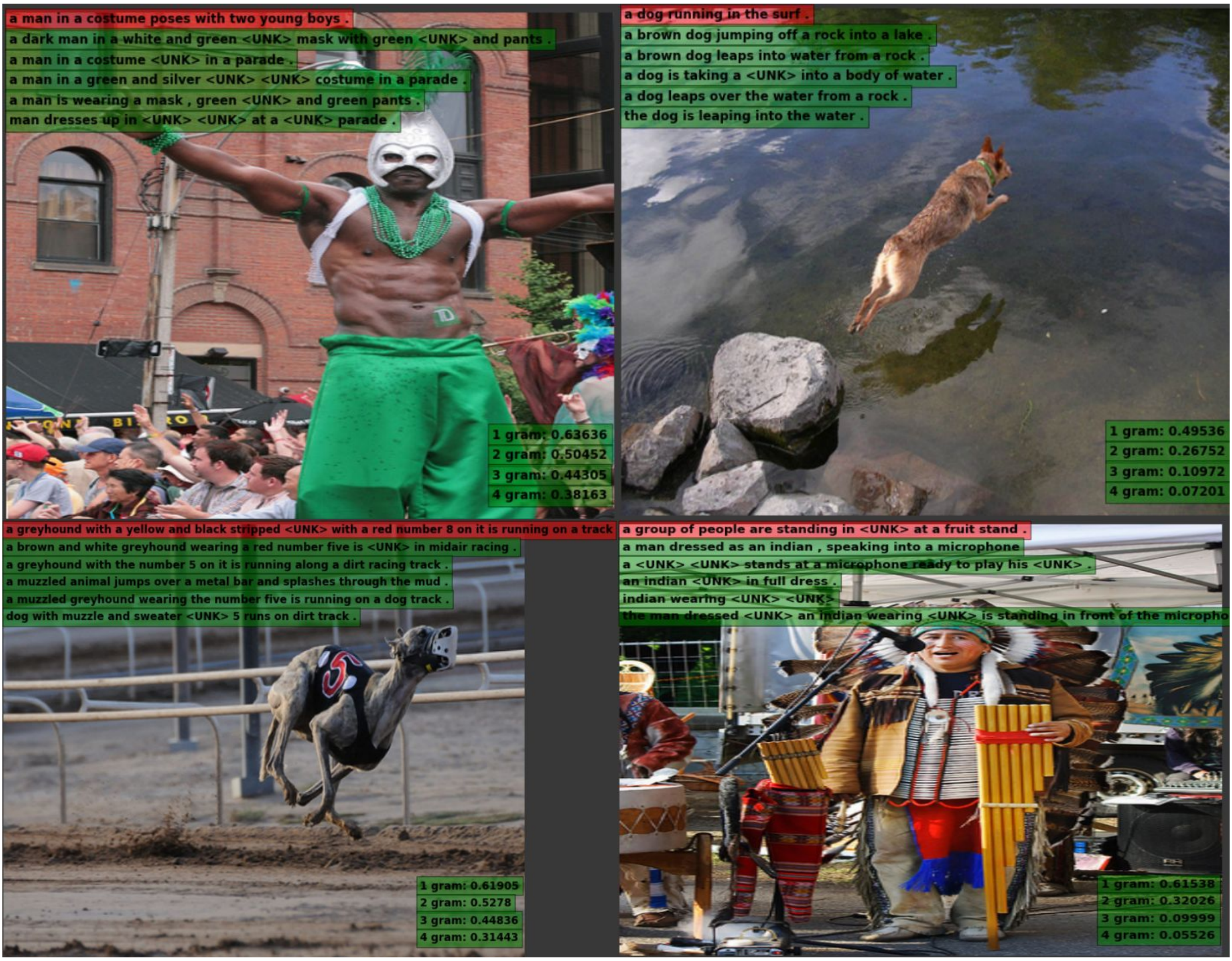
Now let’s see some examples

<u>Accurate description of the image</u>




a man in a red football uniform and helmet looks to the left .
a man is wearing a sooners red football shirt and helmet .
a <UNK> sooners football player wearing his jersey number <UNK> .
a sooners football player <UNK> the number <UNK> and black <UNK> .
guy in red and white football uniform
the <UNK> <UNK> is wearing a red and white <UNK> .
28
SOONERS
28
1 gram: 0.78571
2 gram: 0.54973
3 gram: 0.13879
4 gram: 0.06917
a brown dog running in a yard .
a brown dog running
a brown dog running over grass .
a brown dog with its front paws off the ground on a grassy surface near red and purpl
a dog runs across a grassy lawn near some flowers .
a yellow dog is playing in a grassy area near flowers .
1 gram: 0.875
2 gram: 0.70711
3 gram: 0.55362
4 gram: 0.42729
a group of men play basketball .
a player from the white and green <UNK> team dribbles down court <UNK> by a playe
four basketball players in action .
four men playing basketball , two from each team .
two boys in green and white uniforms play basketball with two boys in blue and white
young men playing basketball in a competition .
1 gram: 0.71429
2 gram: 0.34503
3 gram: 0.13625
4 gram: 0.08784
solvi.org
a dog is jumping over a gate .
a dog lays on a <UNK> on the porch .
a dog rolls on a <UNK> <UNK> on a porch and <UNK> his back .
a <UNK> dog rolling over on a <UNK> <UNK> on a porch .
a white dog rolling on its back on a porch .
the <UNK> dog rolls on its back .
1 gram: 0.625
2 gram: 0.29881
3 gram: 0.11667
4 gram: 0.07386

Partly success



Unsuccessful prediction