



Ulm University | 89069 Ulm | Germany

**Faculty of
Engineering, Computer Science
and Psychology**
Neural Information Processing

Neural Image Caption

Project DLA at Ulm University

Submitted by:

Khrystyna Semkiv

Priya Shukla

Raghu Shantharam


Reviewer:

Prof. Dr. Friedhelm

Schwenker

2023

Neural Image Caption Report-part1

Khrystyna Semkiv, Priya Shukla, Raghu Shantharam ✉ 
Ulm University, Germany

1 Introduction

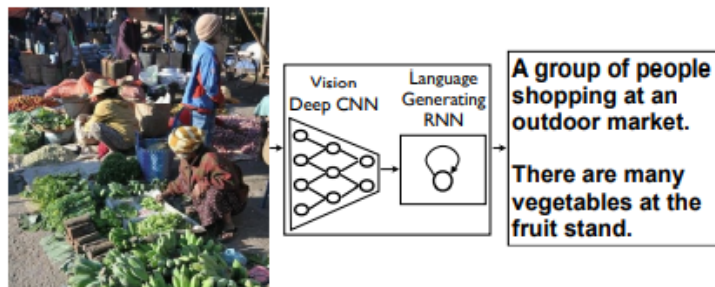
Describing the contents of an image automatically is a challenging problem for which there are not many solutions. However, in artificial intelligence, this problem of image description connects computer vision and natural language processing. This paper introduces a generative model based on a deep learning architecture that is used to generate natural sentences describing an image. This model is trained to maximize the likelihood of the target description sentence given the training image.

Several experiments have been done both qualitatively and quantitatively and it is seen that this model is quite accurate than any of the previous techniques that will be described in the next section. This model is also fluent in learning the language solely from the image description.

The main motivation of this idea is that describing the content of an image automatically using properly formed English sentences is challenging, however, if this is achieved it could have major benefits such as helping visually impaired people better understand the contents of the images. This activity is significantly harder than the image classification or object recognition tasks.

In this work a single joint model is proposed, this model takes an image I as the input and is trained to maximize the likelihood $p(S|I)$ of producing a target sequence of words $S = \{S_1, S_2, \dots\}$ where each word S_t comes from a given dictionary, that describes the image adequately.

The other motivation for this work comes from recent advances in machine translation, where the task is to transform a sentence S written in a source language, into translation T in the target language maximizing $p(T|S)$. The translation can be done in a simple way using RNNs. An "encoder" RNN reads the source sentence and transforms it into a rich fixed-length vector representation, which in turn is used as the initial hidden state of a "decoder" RNN that generates the target sentence.



■ **Figure 1** NIC, our model, is based end-to-end on a neural network consisting of a vision CNN followed by a language-generating RNN. It generates complete sentences in natural language from an input image, as shown in the above example.

In this paper the encoder RNN is replaced by a CNN, over the last few years it is shown that CNNs can produce a rich representation of the input image by embedding it to a fixed-length vector, such that this representation can be used for a variety of vision tasks. This forms the basis to use a CNN as an image "encoder", by the first pre-training it for

image classification and the last hidden layer as an input to the RNN decoder that generates the sentences. This model is called Neural Image Caption (NIC) and is shown in the above figure.

2 Related works

As the task of machine translation exist for many years there were many approaches to implement it in our everyday life. The idea of combining natural language processing and computer vision led us to the possibility of generating novel descriptions for the images.

The first ideas of combining these two areas of research appeared in the last century [5]. The challenging task was to provide not only the list of objects on the video and their tasks but also an ensemble behavior of all of them. Thus, one of the works was aimed at the possibility of introducing a description of the ensemble behavior of objects in real-time. Traffic jam videos were chosen as the research object. The algorithm clearly separated stationary and dynamic motion, and also provided a behavior characteristic of a certain group of objects from the video.

Another important part of the model is the structure of the generated sentence. Here the challenging task is to create grammatically correct, intelligible descriptions. In [10] researchers provide us with a well-described sentence formation that underlines many models today. The final description includes information about objects (or/and stuff), their description (attributes), and the spatial relationship (preposition) of objects to each other and has the structure: object/stuff description + spatial representation (object – preposition – object). Additionally, template linguistic constraints were applied to address correct grammar challenges.

With the development of neural networks and their success in object detection and speech recognition as well as in NLP, the idea of using this approach to solve the task of description generation of the image became quite popular. Along with this, a lot of works appeared, that combined pictures and their corresponding descriptive sentences in one space for training the algorithm.

Firstly, the images and sentences are processed separately. The next step is to combine the feature vector that represents an image and corresponding reference sentences together in one multimodal embedded space. In the paper, [13] the dataset that contains 5 independently created descriptions of each image was used. The idea is to train the model, to have a high inner product with related sentence vectors to the image and to have a low inner product with incorrect pairs. To optimize algorithms the ranking cost function can be used [13] [9]. The alternative cost function can be a squared loss that calculates the distances, as well as maximizes the likelihood probability of the correct description [14].

Previous works have shown good results in this area in relation to the relevant capabilities of their times. Moreover, many ideas are borrowed from these works, which are used in modern models. However, all previously listed methods are limited by the fact that they are not able to properly process the pictures that have not been seen before. This can lead either to misclassification or to an error in the algorithm. Nowadays methods are aimed at solving this question.

One of the common model structures that became quite popular in neural machine translation is based on the encoder-decoder. Here, neural networks are used as a complete end-to-end translation system. In addition, certain works have shown that the process of sentence formation in machine translation can be simplified many times by using recurrent neural networks (RNNs) and still show successful results. The current work was inspired by

the success of these achievements [14].

In the paper [2] researchers were using RNNs autoencoder to translate phrases from English into French. The model is consist of two recurrent neural networks (RNNs):

1. Encoder: RNN encodes a sequence of symbols in a vector of fixed length;
2. Decoder: RNN decodes fixed vector in another sequence of symbols;
3. Hidden unit: LSTM, which decides which information to store and which can be ignored.

It also can control the amount of information that is carried from the previous hidden state to the current one. As a result, we have a more compact representation of information; After the training, the model can be used in both directions. On the one hand, it can generate the target sequence from the input. On another hand, it can estimate the ‘score’ of the input-output, which is simply a probability. Moreover, the algorithms were able to generate phrases that do not overlap completely with the target sentences [2].

Since we use the image as an input, the encoder part was replaced with a convolution neural network and uses a similar approach that is described in [13] to train the model. But still, RNN is preferable for sentence generation.

However, the deeper the neural network is constructed, the more the problem of vanishing and exploding gradients arises. To deal with it, the LSTM, which is another form of RNN, can be used [9]. The idea is similar to that used in [2] as a hidden unit but to the whole decoder part. To predict the next word, the combination of the image and the context of previously generated words are used [14] [9].

Lastly, [11] uses an approach that is very similar to the current proposal that will be described in Section 3. The difference is that NIC uses a more powerful RNN, that has direct access to the image since it was provided to its input. That significantly improved the final result of the model [4].

3 Architecture

Oriol et. al [14], suggests a neural and probabilistic framework to generate descriptions for images. They have achieved state-of-the-art results for a powerful sequence model by maximizing the probability of the correct translation given an end-to-end input sequence. Applicable for both training and inference.

A recurrent neural network encodes the variable length input into a fixed dimensional vector and uses this representation to “decode” it to the desired output sentence. Hence, the reference paper [14] uses the same principle of translating the image(instead of an input sentence in the source language) to its description.

For an image the probability of the correct description was maximized directly with the formula:

$$\theta^* = \arg \max_{\theta} \sum_{(I,S)} \log p(S|I; \theta) \quad (1)$$

θ = parameters of our model, I is an image, S = correct transcription of the image. The length of S is unbounded, as it could be any sentence. Therefore the chain rule is applicable to model the joint probability over S_0, \dots, S_N where N is the length of this particular example as

$$\log p(S|I) = \sum_{t=0}^N \log p(S_t|I, S_0, \dots, S_{t-1}) \quad (2)$$

While designing and training RNNs, the most common challenge is vanishing and exploding gradients [6] and the choice of $f(3)$ is based on the ability to deal with these challenges.

LSTM has worked successfully for translation [2] and sequence generation [1]. Additionally, it also addresses the challenges of RNNs mentioned above. The model of LSTM includes a memory cell c that encodes knowledge at every time step of what inputs have been observed up to this step as shown in the figure 2. The cell's behaviour is controlled by the gates. The gates(layers) are applied multiplicatively. So it keeps the value from the gated layer if the gate is 1 or forgets if the gate is 0. The model uses three gates based on the choice to control if the current cell value has to be forgotten (forget gate f) if it should read its input(input gate i), and whether to output the new cell value (output gate o). The below equations describe the gates, cell update and output:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1}) \quad (4)$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1}) \quad (5)$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1}) \quad (6)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1}) \quad (7)$$

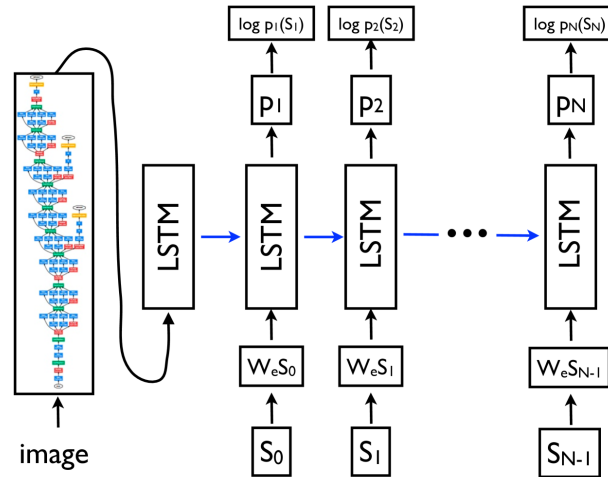
$$m_t = o_t \odot c_t \quad (8)$$

$$p_{t+1} = \text{Softmax}(m_t) \quad (9)$$

where \odot represents the product with a gate value, and the various W matrices are trained parameters. As these multiplicative gates work well against exploding and vanishing gradients [6], it is possible to train the LSTM. There are also non-linearities like a sigmoid and hyperbolic tangent. To produce a probability distribution p_t over all of the words, the last equation m_t is fed to the Softmax.

Training

As defined by $p(S_t|I, S_0, \dots, S_{t-1})$, after seeing the images and all preceding words, LSTM predicts each word of the sentence.



■ **Figure 3** LSTM model combined with a CNN image embedder (as defined in [7]) and word embeddings. The unrolled connections between the LSTM memories are in blue and they correspond to the recurrent connections in Figure 2. All LSTMs share the same parameters. [14].

As shown in figure 3 all LSTMs share the same parameters and the output m_{t-1} of the LSTM at time $t-1$ is fed to the LSTM at time t , therefore a copy of the LSTM memory is created for the images and each sentence word [14]. Then all recurrent connections are transformed into feed-forward connections in the unrolled version. If I denotes the input image and $S = (S_0, \dots, S_N)$ a true sentence describing this image, Then

$$x_{-1} = CNN(I) \quad (10)$$

$$x_t = W_e S_t, \quad t \in 0 \dots N - 1 \quad (11)$$

$$p_{t+1} = LSTM(x_t), \quad t \in 0 \dots N - 1 \quad (12)$$

Oriol et. al [14] has used S_t of dimension equal to the size of the dictionary to represent each word as a one-hot vector. To signal the start and end of the sentence, a special start word S_0 and a special stop word S_N are used. The special stop word S_N emits a signal to the LSTM that a completed sentence has been generated. Lastly, the same space is used to map the image by using a vision CNN and the words by using a word embedding W_e . At $t=-1$, only once input is given as Image I to inform the LSTM about the image contents.

The reference paper [14] mentions that feeding the image at each step as an extra input has hampered the result badly which is empirically verified, as the network exploits noise in the image and overfits more easily. Therefore, they have not fed the image at each step.

The below equation describes loss as the sum of the negative log-likelihood of the correct word at each step:

$$L(I, S) = - \sum_{t=1}^N \log p_t(S_t) \quad (13)$$

With respect to all the parameters of the LSTM, the top layer of the word embeddings W_e and image embedder CNN, the loss is minimized.

Inference - There are multiple approaches to generating sentences for an image, with NIC. The first method is Sampling, where we sample the first word according to p_1 , then provide the corresponding embedding as input and sample p_2 , we can continue like this until we sample the special end-of-sentence token or some maximum length. Another method is Beamsearch, which generates sentences of size $t + 1$, we iteratively consider the set of the k best sentences up to time t and keep only the resulting best k of them. It can be formulated as below equation

$$S = \arg \max_{S'} p(S'|I)$$

The reference paper [14] uses the BeamSearch approach in the mentioned experiments, with a beam of size 20. Using a beam size of 1 (i.e., greedy search) degraded their results by 2 BLEU points on average.

4 Datasets

5 different datasets that consist of images and corresponding description sentences to each of them in English were used. [14] There are two datasets that were created by the same group of people: Flickr8k and Flickr30k. Flickr30k is about 4 times bigger than Flickr8k. The description of the images is based on the same vocabulary, so there are fewer discrepancies between the reference sentences. Unlike the MSCOCO dataset which is much bigger than Flickr40k, since the process of the data collection was done differently it has a larger mismatch.

Dataset name	size		
	Train	Valid	Test
PASKAL	-	-	1000
Flickr8k	6000	1000	1000
Flickr30k	28000	1000	1000
MSCOCO	82783	40504	40775
SBU	1M	-	-

■ **Table 1** Statistics of the datasets [14]

The PASCAL dataset is the one that consists of 1000 samples in the test set. It was collected independently from Flickr and MSCOCO. In this case, transfer learning was applied by using other datasets to train the model. The largest dataset that was used is SBU. It contains 1 million samples. In comparison to all previous datasets instead of human generation description, it has labels (also can be counted as weak labeling), that initially are captioned. The advantage of it is the large number of data that is available for training. However, since the vocabulary is larger and noisier, it is more difficult to process it.

The statistics of the datasets are summarized in Table 1.

5 Evaluation Metrics

In order to check the effectiveness of NIC several experiments were conducted using different data sets, some of the experiments used are described below.

5.1 Human Evaluation

This is the most reliable but at the same time the most time-consuming experiment. As part of this, people were asked to give a subjective score on the usefulness of each description that was generated using the image.

The graders were asked to evaluate each generated sentence with a scale from 1 to 4.

To conduct this experiment, an Amazon Mechanical Turk experiment was set up. Here each image was rated by 2 people. The percentage of agreement between the workers was approximately 65 percent. In case of disagreement, the scores were averaged and the average was recorded as the score.

5.2 BLEU

BLEU¹ is a standard estimation approach that is used in machine translation. The idea of it is to compare generated sentences with the references. The range of it is set to be from 0 to 1, where 0 is a perfect mismatch and 1 is a perfect match. In the paper [14], the result of BLEU is represented in percentages, therefore, the score of this evaluation range from 0 to 100.

The approach compares n-grams in the generated sentence and reference sentences and tries to match them. For example, a 1-gram or unigram processes each word separately, and 2-gram or a bigram processes each word pair.

¹ <https://machinelearningmastery.com/calculate-bleu-score-for-text-python/>

In the paper, the BLEU-1 score is used to estimate the final performance of the algorithms. Since the dictionary contains 5 reference sentences to every image, the generated sentence was compared to each of them and the average was taken as a final score. The model was also estimated by the BLEU-4 using the same approach of taking the average as a final score. Lastly, Ranking uses the proxy task of ranking a set of available descriptions of given images. In this, an evaluation is performed using Recall@K, the higher the better, which means several images for which the correct caption is ranked within the top-K retrieved results (and vice-versa for sentences) [9]. Median rank is also calculated of the closest ground truth result from the ranked list, the lower the rank better the ground truth. However, it can be a complicated task as the complexity to describe images grows as the dictionary grows to make the job of transforming the description generation task into a ranking task unsatisfactory. As the number of combinations of possible sentences grows exponentially with the size of the dictionary [14]. The computational complexity also increases to evaluate such a large amount of data efficiently stored for each image.

It's quite unrealistic that the number of predefined sentences will grow exponentially hence the likelihood that such sentences will fit a new image also drops. Speech recognition also uses the same to produce the sentence corresponding to a given acoustic sequence. Sentences can be generated using a state-of-the-art approach from a large dictionary. hence, it is beneficial to focus on evaluation metrics than ranking [14].

5.3 Ranking

Lastly, Ranking uses the proxy task of ranking a set of available descriptions of given images. In this, an evaluation is performed using Recall@K, the higher the better, which means several images for which the correct caption is ranked within the top-K retrieved results (and vice-versa for sentences) [9]. Median rank is also calculated of the closest ground truth result from the ranked list, the lower the rank better the ground truth. However, it can be a complicated task as the complexity to describe images grows as the dictionary grows to make the job of transforming the description generation task into a ranking task unsatisfactory. As the number of combinations of possible sentences grows exponentially with the size of the dictionary [14]. The computational complexity also increases to evaluate such a large amount of data efficiently stored for each image.

It's quite unrealistic that the number of predefined sentences will grow exponentially hence the likelihood that such sentences will fit a new image also drops. Speech recognition also uses the same to produce the sentence corresponding to a given acoustic sequence. Sentences can be generated using a state-of-the-art approach from a large dictionary. hence, it is beneficial to focus on evaluation metrics than ranking [14].

6 Results

NIC is data-driven and trained end-to-end, experiments were conducted using different data sets to see how these data sets along with parameters like the size of the data sets, how NIC deals with weak labels, and so on might impact the overall performance of the NIC. For this experiments on five different data sets were conducted which helped in understanding the model in depth.

6.1 Generation Diversity Discussion

Having trained the model to be able to give $p(S|I)$ as the output, the immediate question is whether the model generates novel captions and whether they are diverse and precise.

A man throwing a frisbee in a park.
A man holding a frisbee in his hand.
A man standing in the grass with a frisbee.
A close up of a sandwich on a plate.
A close up of a plate of food with french fries.
A white plate topped with a cut in half sandwich.
A display case filled with lots of donuts.
A display case filled with lots of cakes.
A bakery display case filled with lots of donuts.

■ **Figure 4** N-best examples from the MSCOCO test set. Bold lines indicate a novel sentence not present in the training set.

The table in Figure 4 shows the N-best list extracted using the NIC model. We can notice that the samples are diverse and may also show different aspects of the same image.

In bold are the sentences that are not present in the training set. If we take the best candidate, the sentence is present in the training set 80 percent of the time. This is expected since the amount of training data is quite small, so it is easy for the model to pick "exemplar" sentences and use them to generate descriptions.

The top 15 generated sentences are analyzed and it is noticed that about half of the time there is a complete novel description. Also, the BLEU score is similar indicating that they are of enough quality and still they are quite diverse and precise.

6.2 Human Evaluation

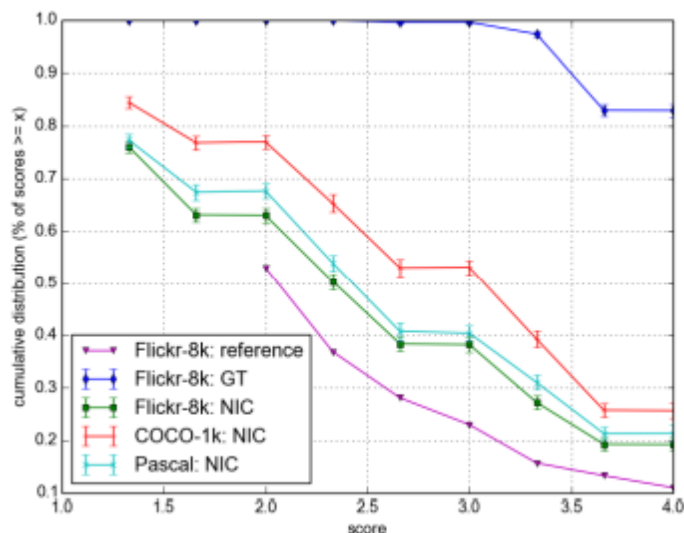
Figure 5 shows the result of human evaluations of the text descriptions provided by NIC from the given image. It is compared to a reference system along with ground truth on various data sets.

NIC is better than the reference system but not as good as the ground truth. BLEU is not a perfect metric as it fails to capture all the differences between NIC and human descriptions assessed by raters.

In Figure 6, images are rated on a scale of 1 to 4. 1 indicates that the image is well described and 4 is the worst case where the description of the image is different from what appears in the image.

The 1st column is where we have the best description of the image and in the last column, we have the case where the images described by the NIC are incorrect. It is also interesting to see, for instance in the second image of the 1st column the NIC model is able to notice the Frisbee given its size.

As observed from the above figure, this technique is quite accurate but it would be even more precise if the number of raters could be increased from 2 to say 7 or 10, which means that we have more people evaluating the description generated from the images so we would be sure that the prediction is accurate.



■ **Figure 5** Flickr-8k: NIC: predictions produced by NIC on the Flickr8k test set(average score: 2.37); Pascal: NIC: (average score: 2.45); COCO-1K: NIC: A subset of 1000 images from the MSCOCO test set with descriptions produced by NIC (average score: 2.72); Flickr-8k: ref: these are results from [11] on Flickr8k rated using the same protocol, as a baseline (average score: 2.08); Flickr-8k: GT: the groundtruth labels were rated from Flickr8k using the same protocol. This provides us with a "calibration" of the scores(average score: 3.89)

Approach	PASCAL	Flickr8k	Flickr30k	SBU
BabyTalk [10]	25			
m-RNN [11]		55	58	
MNLM [9]		56	51	
Im2Text [12]				11
NIC [14]	59	66	63	28
Human [14]	69	68	70	

■ **Table 2** BLEU-1 scores [14]

6.3 BLEU

The results of BLEU were compared with other models that also used this metric to evaluate the accuracy of the generated description, as well as with the human BLEU score [14]. Both BLEU-1 and BLEU-4 have higher values than those for the other models. It is a bit smaller than the human score, but still the closest one to it. The results of BLEU-1 are presented in Table 2.

The BLEU estimation technique is inexpensive and easy to understand, which is why it is widely used in machine translation. Moreover, it doesn't depend on the language and can be used when there is more than one ground truth. However, it does not take into account intelligibility or grammatical correctness. However, there is still debate about whether such evaluation methods are reliable and if they match human judgements [14] [4].

Nonetheless, it is a subjective view of how well this method corresponds to human judgments. With all the advantages of BLEU, it can be still a preferable approach to estimate the final performance. It is obvious that the description of the picture is a subjective



■ **Figure 6** A selection of evaluation results, grouped by human rating.

assessment of the seen, therefore, one picture can have many possible descriptions and all of them will be correct. It should be taken into account that the number and quality of reference sentences can affect the final evaluation.

6.4 Analysis of Embeddings

In order to describe the image better word embedding vectors are used. The advantage is that they are independent of the size of the dictionary. These word embeddings can be jointly trained with the rest of the model.

Word	Neighbors
car	van, cab, suv, vehicule, jeep
boy	toddler, gentleman, daughter, son
street	road, streets, highway, freeway
horse	pony, donkey, pig, goat, mule
computer	computers, pc, crt, chip, compute

■ **Figure 7** Nearest neighbors of a few example words.

The table in Figure 7 shows a few example words, the nearest other words found in the learned embedding space.

Some of the relationships learned by the model will help the vision component. For example, having "horse", "pony" and "donkey" close to each other will encourage CNN to extract features that are relevant to those animals that look like a horse.

It is hypothesized that in extreme cases such as "unicorn" and "horse" there is much more information provided that would otherwise be completely lost with more traditional bag-of-words-based approaches.

This is a nice feature that captures the minute details in the image and is crucial in deciding what exactly is visible in the image. For example, if we have an image where a man is driving a car, the description of the image by the NIC would be that a man is driving a car and if the man is driving a Bus, the description would be the man is driving a Bus and not

Approach	Image annotation			Image Search		
	R@1	R@10	Med r	R@1	R@10	Med r
DeFrag [8]	13	44	14	10	43	15
m-RNN [11]	15	49	11	12	42	15
MNLM [9]	18	55	8	13	2	10
NIC [14]	20	61	6	19	64	5

■ **Table 3** Recall@k and median rank on Flickr8k [14]

Approach	Image annotation			Image Search		
	R@1	R@10	Med r	R@1	R@10	Med r
DeFrag [8]	16	55	8	10	45	13
m-RNN [11]	18	51	10	13	42	16
MNLM [9]	23	63	5	17	57	8
NIC [14]	20	61	6	17	57	8

■ **Table 4** Recall@k and median rank on Flickr30k [14]

the man is driving a car. We could also add another subset that could capture more details, say for example also the brand of the automobile in the 1st row of Figure 7. The description would now be such that a man is driving a Mercedes Benz car. That is the NIC would be able to learn and predict even more detailed features with this inclusion of the new subset.

6.5 Ranking

Orion et al mentioned ranking is not effective to evaluate the description generation from images though many papers report ranking scores, using the set of testing captions as candidates to rank test images. Also, some of them use ranking as an optimization technique. MNLM, an approach that works best on these metrics has also implemented a ranking aware loss. Surprisingly, NIC has reported good results on both ranking tasks; ranking descriptions given images, and ranking images are given descriptions [14] see Tables 3 and 4. They have borrowed the idea to normalise the Image annotation task from what [11] used.

7 Further Improvement

Our main work will be directed at checking the efficiency of this algorithm with a much smaller dataset since our capabilities are limited by the power and memory of our computers. It also includes changes in hyperparameters and adding different regularization techniques such as augmentation.

Since one of the motivations of this idea is to help visually impaired people better understand the contents of the image. Our proposal is to convert the text description into speech so that people will also be able to hear the description of the image.

More people can be involved in the human evaluation, at least 7 to 10 which might provide us with a more precise estimation.

Another idea is to add novel-generated descriptions to the vocabulary. However, the complexity of the algorithms will grow too.

We can also borrow an idea for the decoder part from [11]. Instead of using an image only once at the decoder’s input, it can be used at each hidden state. That can increase the

accuracy of the final performance. However, in this case, the complexity of the model grows too.

Another idea was proposed by researchers in the paper that tells to use the same model, but with unsupervised learning.

8 Conclusion

In this paper NIC based approach is used to generate a reasonable description in English using images as the input to the neural network.

NIC is based on a convolution neural network (CNN) that encodes an image into a compact representation, followed by a recurrent neural network (RNN) that generates the corresponding sentence.

The model is trained to maximize the likelihood of the sentence given the image.

Several experiments have been conducted using different data sets, where we have observed that NIC performs way better than the other approaches though not as perfect as the ground truth which is expected behavior.

The BLEU score of NIC is the closest to BLEU human score.

It is also observed that as the size of the available data sets for image description increases, so will the performance of NIC. For validation and test sets, we need to have good labels.

Neural Image Caption Report-part2

K Semkiv, P Shukla, R Shantharam

9 Introduction

As we have discussed in section 7 we have planned to implement models with some improvements. However, out of all the points we could implement some of them as listed below.

- We have used pictures from the gallery with different situations.
- We train our model on a small dataset as mentioned in section 10, hence we could introduce some regularization techniques like augmentation and change hyperparameters.
- From the 1 we mentioned the motivation. Hence, we extended it to implement the text-to-speech part explained in section 14.
- We also did human Evaluation with a group of 8 people as explained in section 6.2

10 Dataset

In our experiment, we used the Flickr8k dataset. It is a publicly available dataset, with a collection of sentence-based image descriptions. It contains 8,000 images and each of them is captioned in five different ways with a clear description of the events and entities. The pictures are combined from 5 different flicker groups but are not distinguishable based on any well-known people or locations. They are manually picked to include different scenes and situations.

We couldn't work with other datasets because of the limited computational power of our laptops and we have limited datasets that could be accessible publicly and have well-defined captions. But with flickr8k:

1. It is small in size (the model can be trained easily on low-end laptops/desktops).
2. Data is properly labelled.
3. The dataset is available for free.

Hence, Flickr8k is the best option among the available datasets.

11 Algorithm

11.1 Library

We have used NumPy and TensorFlow libraries. TensorFlow and NumPy are two popular Python libraries used for scientific computing and data analysis.

TensorFlow is an open-source library developed by Google that is mainly used for building and training deep learning models. It provides a framework for creating computational graphs, which are composed of a series of mathematical operations. TensorFlow allows users to easily define, optimize, and execute these graphs on a variety of devices, including CPUs, GPUs, and TPUs (Tensor Processing Units). TensorFlow is widely used for a range of applications, including image and speech recognition, natural language processing, and robotics.

NumPy, on the other hand, is a fundamental package for scientific computing in Python. It provides a powerful N-dimensional array object, along with a collection of functions for performing operations on these arrays. NumPy arrays are similar to lists in Python, but they are more efficient for numerical operations and can be used to represent vectors, matrices,

and higher-dimensional tensors. NumPy also provides a variety of mathematical functions for performing operations such as matrix multiplication, dot products, and Fourier transforms.

Both TensorFlow and NumPy are widely used in data science and machine learning applications and are essential tools for anyone working in these fields

11.2 Pre-processing

We have performed pre-processing on both the images and text that resulted in the reduced dataset of size 7643 images out of which we have 6114 trained images and the rest 1529 are test images.

11.2.1 Image

- We have fixed the dimensions of the images to 299x299.
- Choosing images with a length of captions. The word length is set to be between 5 to 25.
- Some random images augmentations like random flip, random contrast, or random rotation are performed for example- RandomFlip: horizontal and vertical, RandomRotation: 0.2, RandomContrast: 0.3.

11.2.2 Text

- Adding <start> and <end> tokens at the beginning and end of the statements respectively.
- We have also replaced the uppercase string with lowercase strings.
- Also removed the string chars for example "<", ">".

11.3 CNN

Convolutional Neural Network (CNN) model using the EfficientNetB0 architecture from the efficient net module in Keras. It returns a Keras Model object. First, we instantiate the EfficientNetB0 model with input shape (*IMAGE SIZE, 3), where IMAGE SIZE is a tuple specifying the dimensions of the input images, and 3 is the number of colour channels (RGB). The top (classification) layers of the model are not included, so the output of the model will be the feature maps.

The output of the feature extractor is reshaped using a Reshape layer. Finally, the construction of a new Keras Model object that takes the input of the EfficientNetB0 model and outputs the reshaped feature maps. This new model is returned as the output of the Encoder.

11.4 Encoder

The encoder block consists of a Transformer model for natural language processing tasks. The TransformerEncoderBlock class represents a single encoder block in the Transformer model, which consists of a multi-head self-attention layer followed by a position-wise feedforward layer. The constructor of the TransformerEncoderBlock needs input dimensions of the input embedding, the hidden layer, and the number of attention heads, respectively. It initializes the first attention layer as a MultiHeadAttention layer with the given number of heads and key dimensions and initializes two LayerNormalization layers and a Dense layer for the feedforward layer. the encoder block first applies layer normalization to the inputs and then applies the dense layer to it. and further applies layer normalization to the sum of the inputs and the attention output. Finally, it returns the output of the layer normalization.

Then we have PositionalEmbedding with a positional embedding layer in the Transformer model, which adds position information to the input embeddings. It takes a fixed Vocabulary size of 1000, a sequence length of 25, and embedding dimensions of 512. It initializes two Embedding layers - one for the tokens and one for the positions - with the given vocabulary size and sequence length, and embedding dimension. It first computes the length of the input sequence and then generates a tensor of position indices. It then passes the inputs to the token embeddings layer and the position indices to the position embeddings layer and adds the resulting embeddings to obtain the final embedded output with positional information.

11.5 Decoder

This block uses a TensorFlow/Keras layer that implements a transformer decoder block. Transformer decoder blocks are used in transformer models, which are a popular architecture for natural language processing tasks such as machine translation and language modelling. The block needs dimensions of the embedding space, the feedforward layer which is 512 units, and the number of heads to use in the multi-head attention layers. layers of MultiHeadAttention layers, each of which consists of a scaled dot-product attention mechanism followed by a feedforward neural network with a residual connection and layer normalization. We have chosen dimensions for the image embeddings and token embeddings to be 512, and units in the per-layer feed-forward network to be 512 as well. Further, normalizing the activations of the previous layer for each given example in a batch independently, rather than across a batch like Batch Normalization. i.e. applying a transformation that maintains the mean activation within each example close to 0 and the activation standard deviation close to 1. It has three layers for normalization and the layer with an activation function ReLu(Rectified Linear Unit,) the most commonly used activation function in deep learning because of its simplicity and effectiveness in reducing the vanishing gradient problem. It is defined mathematically as:

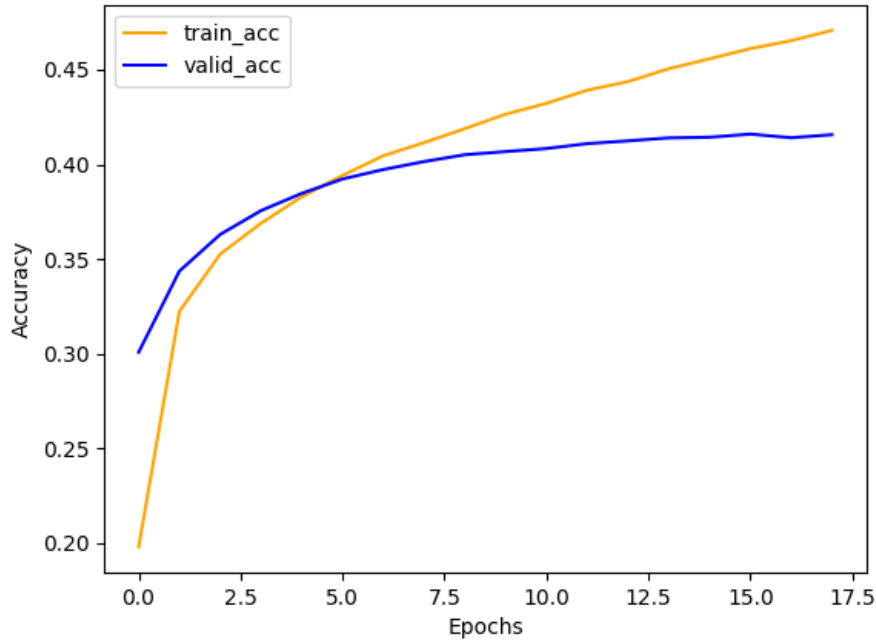
$$f(x) = \max(0, x) \quad (13)$$

It introduces non-linearity into the network and is computationally efficient to compute. After the MultiHeadAttention layers, the output is fed into a base model, which can be a simple pooling layer followed by a dense layer and an output layer. Dropout is also applied to prevent overfitting. Finally, the predicted class is outputted as the output of the model. We have fixed the vocabulary size to 1000, and the length of the sequence to 25 words.

There are two layers of dropout with rates of 0.3 and 0.5 respectively. For the dataset part, we make batches of 256 images and keep shuffling them.

12 Training

To train the model, the categorical crossentropy function is used. As an optimization technique, Adam is used. It has a lot of hyperparameters that may slow down the training process. Number of epochs initially was specified as 30. However, the algorithm starts to overfit on the halfway. That is why it is necessary to use an early stopping. Additionally, the adaptive learning rate depends on the number of epochs. It affect the number of iterations after which the learning rate will be recalculated.



■ **Figure 8** The accuracy of the initial NIC algorithm

13 Experiment

First of all, we run the initial code to see its capabilities. The validation accuracy saturates at 0.4. Losses for both: training and validation set are high. On the other hand, there is no big gap between training and validation sets.

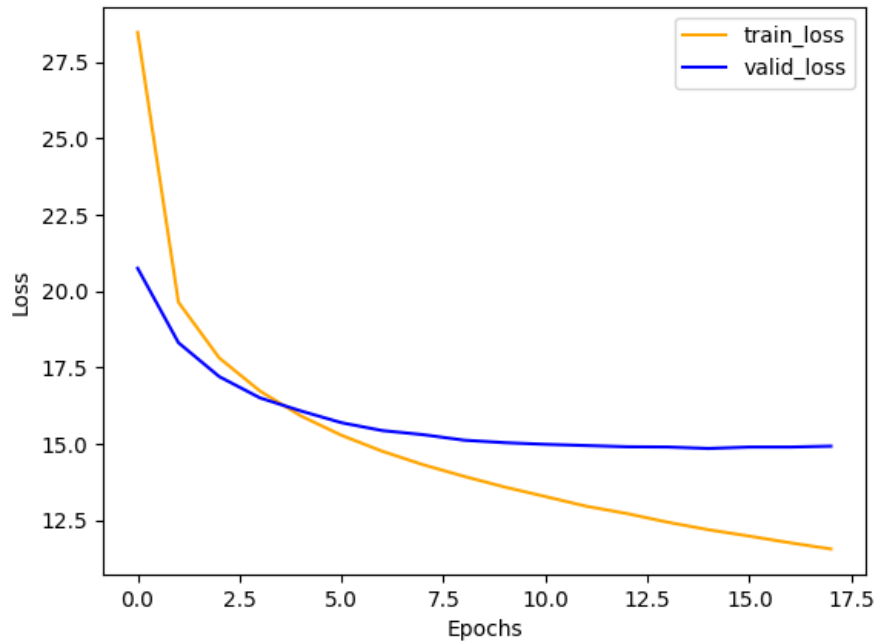
Since we could reach only around 50 percent of accuracy by using the initial code, we have to think how to improve the algorithms, so we can increase the quality of overall performance. First of all, we have applied little changes to see how the algorithm would behave. On this step we have discovered our main limitation - the possibilities of our devices. Every change has affected the computational complexity that resulted in the increasing of the operation time. For more details, see some of the examples in table 5.

However, in order to improve NIC algorithms it is necessary to apply more significant changes than the usual adjustment of hyperparameters. In this case, our laptops couldn't handle such a task. That is why, we had to change the task in the process. We conducted the sensitivity test by evaluating the impact of changes in each part on the final output. As the assessment technology we have used the human evaluation technique 16. The sensitivity test provides an information of crucial parts of the algorithms that should be taken into account while designing better version of current NIC algorithm.

We have highlighted several crucial points:

1. The size of the dataset

Decreasing/Increasing the size of the dataset even on 10 percent affect the quality of generated sentences. Adding more pictures provides new combination of words and vocabulary, due to which the generated sentences take on a more varied form. While removing pictures dramatically degrades sentences by forming them with multiple uses of



■ **Figure 9** The loss of the initial NIC algorithm

Model	Time	Number of epochs
Original	35 min	18
Reduced data-set (4000 in total)	14 min	15
Without dropout (decoder)	more than 2 hours	18
Increasing dropout (decoder)	45min/epoch (crashed after 2 hours)	-
Adding image pre-processing block	more than 5 hours	18
Changing optimizer	45min/epoch (crashed after 2 hours)	-

■ **Table 5** Operating time for different models

the same words.

2. Reference sentences

It was said before[14], that the quality of the labeling is very important. This is an advantage of Flickr8k/40k, because the vocabulary for the reference sentences is the same. To understand how sensitive is that point, we have adjusted just 10 random sentences of different pictures. The impact was more than expected because the generated sentences lost their meaning and looked more like a set of words.

3. Dropout in the decoder

This aspect is very important for the structure of the generated sentences. The idea of changing it appeared to reduce the number of connection, that also reduces the operation time and number of weights. However, it loses also a lot of important information, because the unique structures of neuron connections disappear.

4. Base model for the classification

In the encoder part, to provide classification task, the base model is chosen. Mostly, this is the model that is pre-trained on the ImageNet. Nonetheless, there are different base models: EfficientNetB0 (the one used in our code), inceptionv3, CoCA, etc. Pre-trained models are trained for different purposes, so may not be suitable in some of the cases. Choosing the right model is a critical task.

5. Dataset

During the researching we have noticed that there are some pictures that are predicted always the best (it could always recognise the dog properly, but never could identify a cat). Additionally, in some cases, generated sentences are very similar. This led us to an assumption that the classes are very unbalanced. Flickr8k might not be the choice for such a difficult task.

All the other changes, such as: changing parameters of pre-processing images, deeper the neural network, changing the size of epochs, the batchsize, adjusting learning rate, as well as using optimizer with less hyperparameters didn't result in any changes.

14 Text to Speech

One of the main motivations of the idea is to help visually impaired people better understand the content of the images. In order to achieve this objective it is not just sufficient to convert the images into text, rather it would be even more beneficial if we convert the text also into an audio format so that the visually challenged people can hear and get to know about the image.

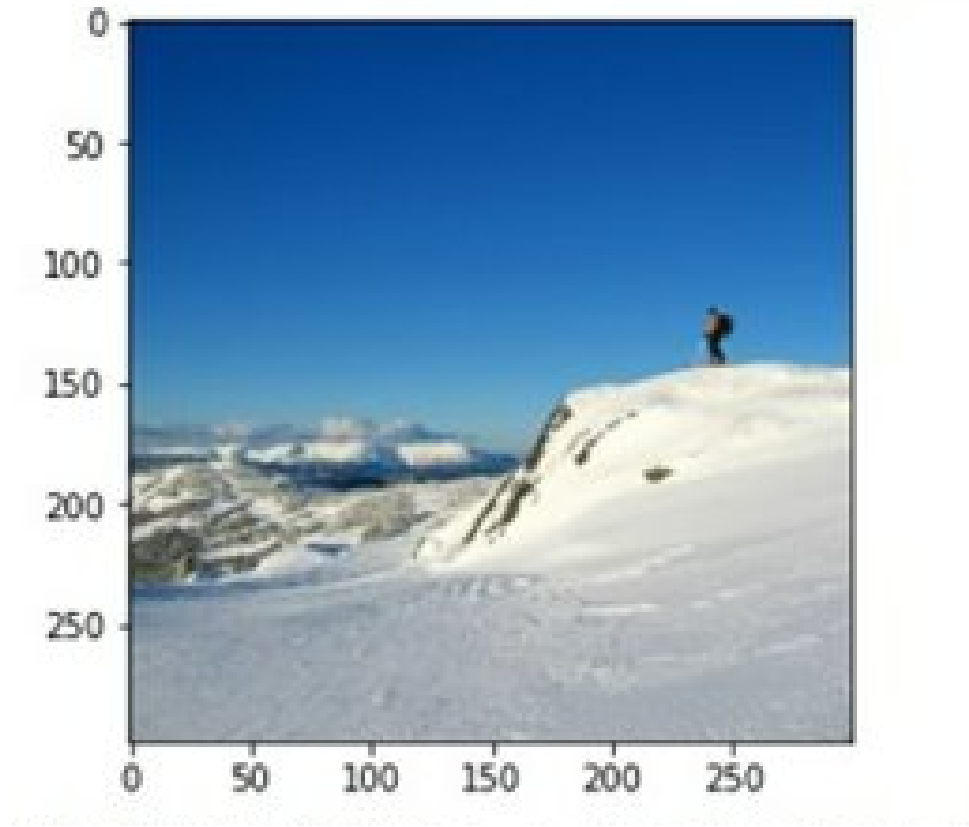
This has been achieved using the "gTTS" python library that converts text into speech, we then save it in MP3 format and then use the Audio function to play the audio file.

15 BLEU

The BLEU score is a metric that enables us to estimate the efficiency of the NIC model. BLEU score on terms compares the candidate sentence against the reference sentences and then estimates how well the candidate sentence is blended in accordance with the reference sentences. In this way, it rates the score between the range of 0-1, respectively.

To implement the BLEU score, we used the NLTK module which consists of the `sentence_blue()` function. It enables us to pass the reference sentences and a candidate sentence. Then, it checks the candidate sentence against the reference sentences.

If a perfect match is found, it returns 1 as the BLEU score. If no match at all, it returns 0. For a partial match, the BLEU score is between 0 and 1.



■ **Figure 10** Predicted Caption: A man stands on a snowy mountain, BLEU score -> 1.0

The above figure shows an example of the BLEU score prediction that was predicted using the `sentence_blue()` library in our code.

By default, the `sentence_bleu()` function searches for 1 word in the reference sentences for a match. We can have multiple words in the queue to be searched against the reference statements. This is known as N-gram.

1-gram: 1 word, 2-gram: pairs of words, 3-gram: triplets and so on.

16 Human Evaluation

As already explained in the section 6.2, we have also done human evaluation for our set of experiments where we have 5 different algorithms and captions generated by each algorithm.

However, the ground truth has also gone through human evaluation as we can see from all the graphs that it is not absolute 4 that is the high score it has also been taken as a mean from all the scores by different person's ratings. One person has chosen the ground truth for the images with the same set vocabulary as used in the Flickr8k dataset. We have 8 people who have evaluated the 10 pictures combined with the flickr8k dataset and pictures from our phone galleries. With the group selected 8 people with their individual scores from 1 to 4, where 1 is the worst and 4 is the best.

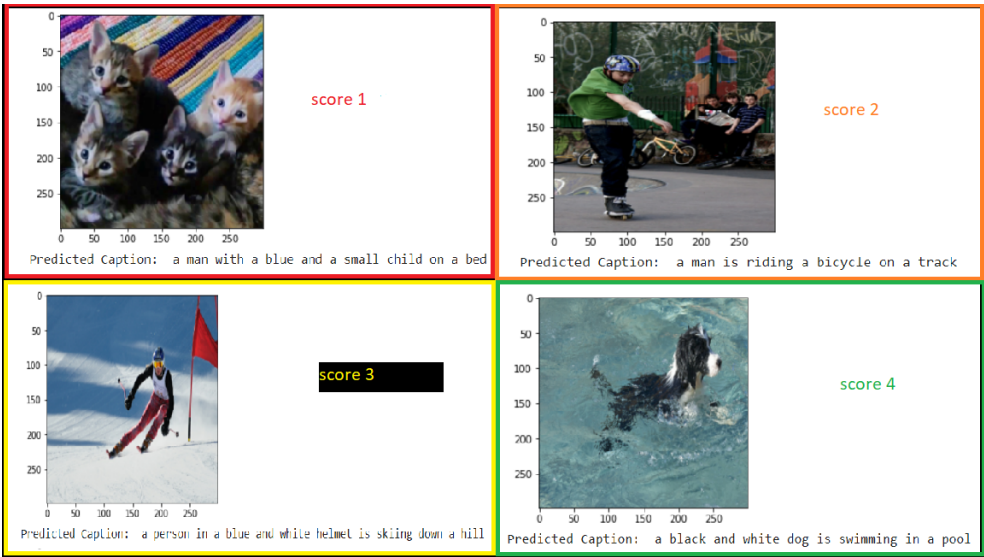


Figure 11 Caption generated for different images with score from 1 to 4

To remove the bias every person who did the evaluation wasn't informed about the algorithms changes. We have our

- Original algorithm, the result can be seen in Figure 12.

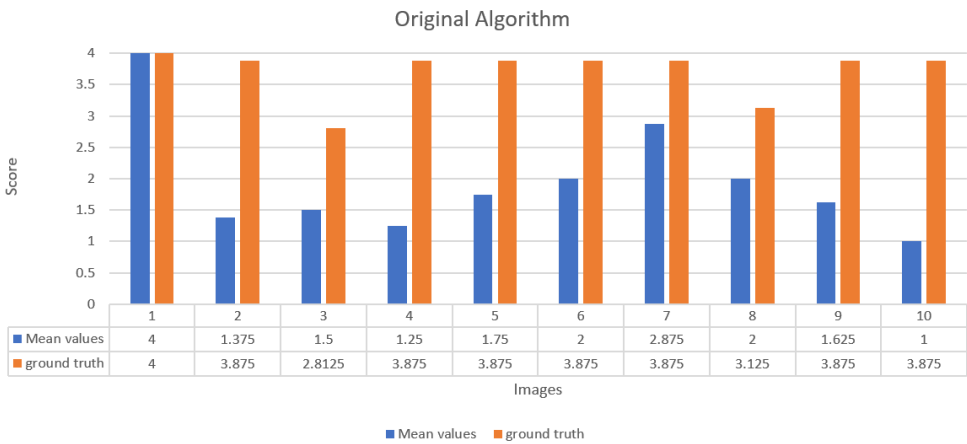


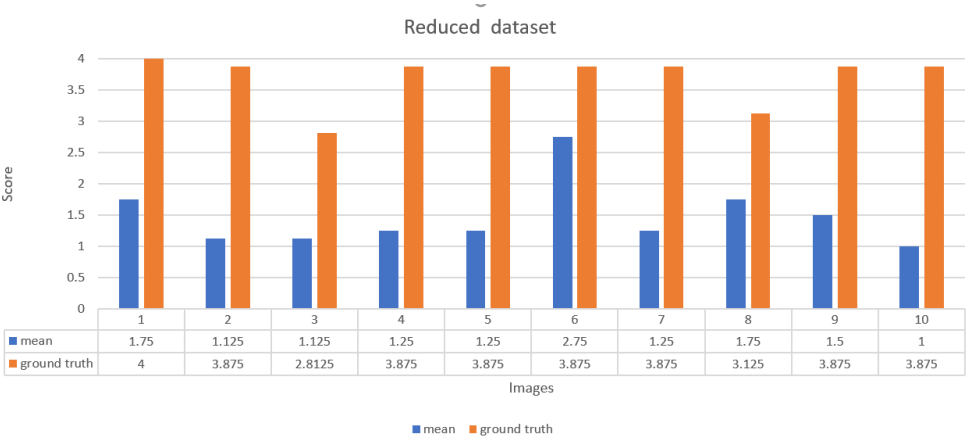
Figure 12 Human Evaluation for Original algorithm.

■ Original algorithm without early stopping, the result can be seen in Figure 13.



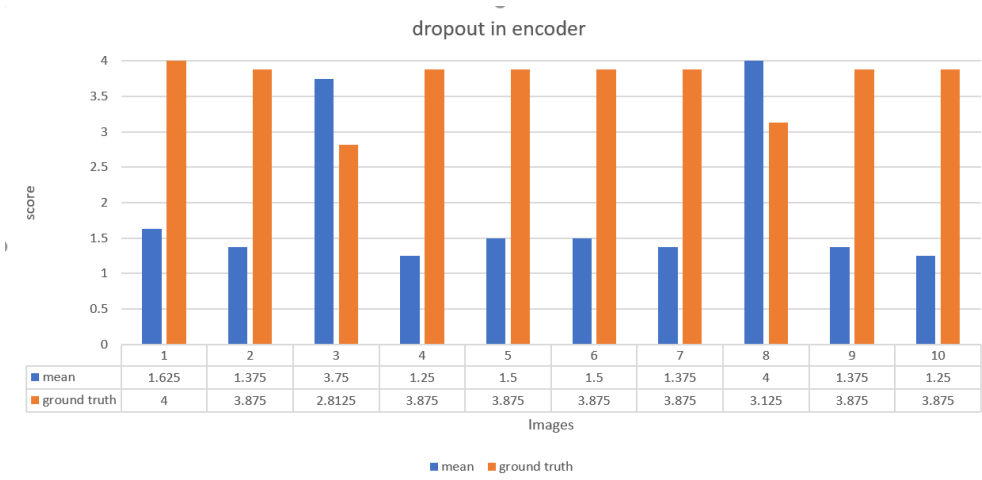
■ **Figure 13** Human Evaluation for algorithm without early stopping

■ Algorithm with a reduced dataset, the result can be seen in Figure 14.



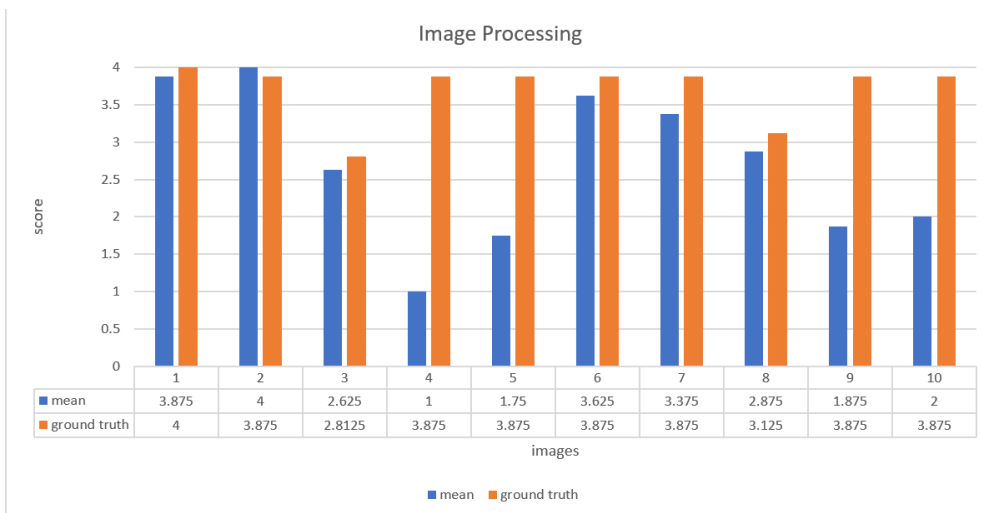
■ **Figure 14** Human Evaluation for the algorithm with a reduced dataset

■ Introducing dropout in the encoder, the result can be seen in Figure 15.



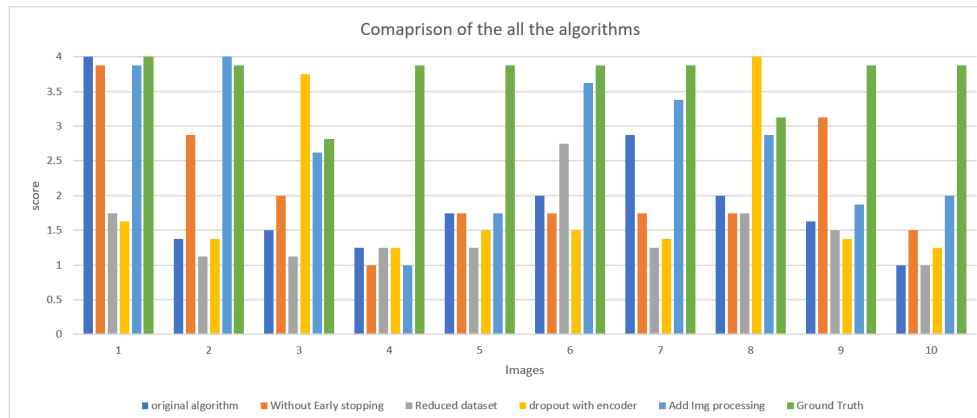
■ **Figure 15** Human Evaluation for the algorithm with dropout in the encode

■ Adding additional image processing block, the result can be seen in Figure 16.



■ **Figure 16** Human Evaluation for the algorithm with additional image processing block

Finally the comparison of all the algorithms with ground truth.



■ **Figure 17** Comparison of all the algorithms.

17 Conclusion

We have implemented the BLEU score discussed in section ?? to provide an additional objective assessment of the results. We have also added Text-to-speech to complete the motivation to present its scope to visually impaired people. For the Human Evaluation 16 we have results as shown in Figure17. However, we have some critical points about the performance of the algorithm as it has performed poorly. The point of improvement stresses using big data which is updated with new updates, hence the algorithm from everywhere and has different kinds of images. As we have seen in the images it can't identify images of cats, it seems the dataset has really good images of dogs, and babies and it keeps using the word blue more often. In our experiment part we have highlighted the sensitive parts of the algorithm we tried to improve. One of the major challenges is also to train the algorithm on a huge dataset which is not feasible unless we have access to a powerful device so complicated machine learning methods can be performed. The reference paper does not allow access to their codebase hence a fair comparison is not possible. Overall, it could be fun and interesting to process our pictures, but this should not be used as a preparation tool.

Apart from the above points, it's not clear why the original author has chosen the limit to words in a sentence between 5 to 25. We have tried to change the length but it only degraded the performance has generated worse captions for images.

18 Opinion on Reference Paper

The reference is well structured and has explained all the concepts in a simplified manner. It includes mathematical equations to approach the problem. The pictures, flow charts, and table are easily comprehensible and for complex well, the references are provided and can be understood with detailed research and study. The reference paper has trained the model on different datasets however we could only do it for Flickr8k. They have introduced different evaluation methods like ranking, BLEU, and human evaluation. From our side the idea is innovative and it has multiple applications including a good social cause with helping visually impaired people only if the descriptions were more accurate. However, the paper lacked motivation. It also doesn't suggest any improvements or possibilities other than unsupervised learning. The main problem was also no access to the technical code.

References

- 1 Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013. URL: <http://arxiv.org/abs/1308.0850>, arXiv:1308.0850.
- 2 Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. URL: <http://arxiv.org/abs/1406.1078>, arXiv:1406.1078.
- 3 Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013. URL: <http://arxiv.org/abs/1310.1531>, arXiv:1310.1531.
- 4 Desmond Elliott and Frank Keller. Image description using visual dependency representations. In *Conference on Empirical Methods in Natural Language Processing*, 2013.
- 5 R. Gerber and N.-H. Nagel. Knowledge representation for the generation of quantified natural language descriptions of vehicle traffic in image sequences. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 2, pages 805–808 vol.2, 1996. doi:10.1109/ICIP.1996.561027.
- 6 Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997. arXiv:<https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf>, doi:10.1162/neco.1997.9.8.1735.
- 7 Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL: <http://arxiv.org/abs/1502.03167>, arXiv:1502.03167.
- 8 Andrej Karpathy, Armand Joulin, and Li Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS’14, page 1889–1897, Cambridge, MA, USA, 2014. MIT Press.
- 9 Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539, 2014. URL: <http://arxiv.org/abs/1411.2539>, arXiv:1411.2539.
- 10 Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Baby talk: Understanding and generating simple image descriptions. In *CVPR 2011*, pages 1601–1608, 2011. doi:10.1109/CVPR.2011.5995466.
- 11 Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. Explain images with multimodal recurrent neural networks. *CoRR*, abs/1410.1090, 2014. URL: <http://arxiv.org/abs/1410.1090>, arXiv:1410.1090.
- 12 Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL: <https://proceedings.neurips.cc/paper/2011/file/5dd9db5e033da9c6fb5ba83c7a7e9bea9-Paper.pdf>.
- 13 Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014. URL: <https://aclanthology.org/Q14-1017>, doi:10.1162/tacl_a_00177.
- 14 Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555, 2014. URL: <http://arxiv.org/abs/1411.4555>, arXiv:1411.4555.