

One other sense for Visually Special people using Image captioning

DISSERTATION

Submitted in partial fulfillment of the requirements of the
M.Tech Data Science and Engineering Degree program

By

Sachin Laxkar
2020SC04810

Under the supervision of
Amit Kale
(Lead)

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
Pilani (Rajasthan) INDIA
(March, 2023)

A REPORT

ON

One other sense for Visually Special people using Image captioning

BY

Name of the
Student

ID.No.

Discipline

Sachin Laxkar

2020SC04810

Data Science

Submitted in partial fulfillment of the requirements of the M.Tech Data Science and
Engineering Degree program

AT

Morgan Stanley

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

Pilani (Rajasthan)

INDIA

(March ,2023)

Acknowledgement

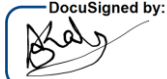
I want to start by expressing my gratitude to Mr. Amit Kale, who is my project's supervisor, for his insightful counsel. I was tremendously motivated to work on this project by him. My idea benefited much from his openness to inspire me. He provided me with other examples that were relevant to the subject of my project, for which I also like to thank him.

Additionally, I cannot thank enough to Mr. N. Srinivasan, Professor at BITS Pilani off campus programme for his continuing advice and useful input all during the dissertation process.

I wish to thank my company, Morgan Stanley, for their assistance and approval of my enrollment in this course. I wish to express my gratitude to my study group for their moral support and backing. Finally, I want to give special thanks to my family and friends for their understanding and support as I worked to finish this project.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
CERTIFICATE

This is to certify that the Dissertation entitled **One other sense for Visually Special people using Image captioning** and submitted by Mr. Sachin Laxkar, IdNo. **2020SC04810** in partial fulfillment of the requirements of DSECLZG628T Dissertation, embodies the work done by him/her under my supervision.

DocuSigned by:


BC2C28FA32DB4AE...

Signature of the Supervisor Name:

Amit Kale (Lead)

Date: 3/13/2023

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

FIRST SEMESTER 2022-23

DSECLZG628T DISSERTATION

Dissertation Title : One other sense for Visually Special people using Image captioning

Name of Supervisor: Amit Kale

Name of Student : Sachin Laxkar

ID No. of Student: 2020sc04810

ABSTRACT:

Although it is a difficult effort, being capable of automatically describe an image's content using properly constructed English phrases could have a significant impact by improving the understanding of visually impaired persons. The captions created from these photographs can then be read aloud to the blind in order to help them understand what is going on around them.

The ability of computer vision and deep learning to generate captions from a collection of realistic images is quite astonishing. Some academics and other colleagues are attempting to generate captions according to the image fed as input using deep recurrent architecture. They are still making an attempt to produce captions that make sense.

Deep learning's field of image-to-text synthesis is expanding, but there are still many issues to be addressed before the technology can be sufficiently matured. In this work, we would use a dataset of flicker to generate captions both naturally and based on images or phrases

Keywords: Natural language Processing, NLP, CNN, RNN, Attention, Captions, Captioning, Image Features,

List of Symbols and Abbreviations Used

1. LSTM - Long short-term memory
2. CNN - Convolutional Neural Network
3. ReLU - Rectified Linear Activation Unit
4. RNN – Recurrent Neural Network
5. NLP – Natural Language Processing

List of Figures

Figure 1- Image Caption Fragmentation 10

Figure 2: Applications of Image Generation.....12-13

Figure 3: Configuration..... 14

Figure 4: Workflow Diagram 15

Figure 5: LSTM Architecture 18

Figure 6: DensNet Architecture 19

Figure 7: Caption Model Architecture 19

Figure 8: Trained and Generated Images 20

Figure 9: Image Caption Generated Model 21

Figure 10: Attention Model Architecture 22

Figure 11: Attention Model Working..... 23

Figure 12: Larger Picture of Attention Model..... 23

Figure 13: An Image in Detailed Form in Attention Mechanism 24

Figure 14: Before Applying Attention Model..... 25

Figure 15: After Applying Attention Model 26

Figure 16: Attention Model Training 27

Figure 17: Attention Model Training Cont..... 28

List of Tables

Table 1- Comparison of Caption generation Approaches 17

Table of Contents

Acknowledgement	3
List of Figures.....	7
List of Tables.....	8
Chapter 1 - Introduction.....	10
Chapter 2 – Literature Review.....	11
Chapter 3 – Applications	12
Chapter 4 - Proposed Methodology	14
Chapter 5 – Comparison of Caption generation Approaches.....	17
Chapter 6 – LSTM-CNN Models.....	18
Chapter 7 – Pattern Generation Progress (LSTM).....	20
Chapter 8 – Attention Model	22
Chapter 9 – Pattern Generation Progress(Diffusion Model)	24
Chapter 10 – Attention Model Training.....	27
Chapter 11 - Conclusion	29
Chapter 12 – Future Works	30
Chapter 13 - References.....	31
Check list of items for the Final report.....	33

Chapter 1 - Introduction

Although it is a difficult effort, being capable of automatically describe an image's content using properly constructed English phrases could have a significant impact by improving the understanding of visually impaired persons. The captions created from these photographs can then be read aloud to the blind in order to help them understand what is going on around them.

The ability of computer vision and deep learning to generate captions from a collection of realistic images is quite astonishing. Some academics and other colleagues are attempting to generate captions according to the image fed as input using deep recurrent architecture. They are still making an attempt to produce captions that make sense.

Deep learning's field of image-to-text synthesis is expanding, but there are still many issues to be addressed before the technology can be sufficiently matured. In this work, we would use a dataset of flicker to generate captions both naturally and based on images or phrases.

The paper puts out a suggestion for improving the technology's use for the benefit of those who are blind. The project comprises a series of Images that sends it to an algorithm for image description. The image is processed and converted to text by the image captioning system. The captions are produced using both computer vision and natural language processing. The user is then supplied a speech file that is a transcription of the written description of the image

In order to do this, we used simple pre-processing methods such image scaling and attempted CNN, LSTM, and ATTENTION models to create image descriptions.

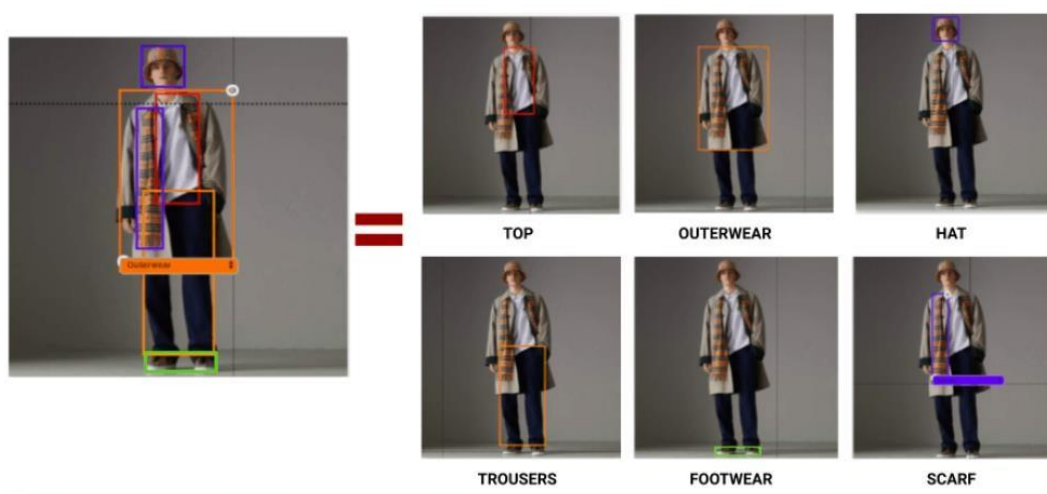


Figure 1: Image Caption Fragmentation

Chapter 2 – Literature Review

Recent study has heavily emphasized the creation or construction of captions, with varying degrees of success. The methods range from simple machine learning to sophisticated deep learning methods. The CNN-LSTM and CNN-ATTENTION models are being used to generate captions, and this work is ongoing. Many research papers by different scholars that were presented at conferences and in journals around the world have been listed. The underlying theories and concepts we use are explained in all of these articles.

[1] Generation of Captions Using CNN:

Recent study has heavily emphasized the creation or construction of captions, with varying degrees of success. The methods range from simple machine learning to sophisticated deep learning methods. The CNN-LSTM models are being used to generate captions, and this work is ongoing. Many research papers by different scholars that were presented at conferences and in journals around the world have been listed.

It creates a fresh caption that mimics the event happening in image. The system modifies activation functions while switching to convolutional layers from layer pooling. The outcomes demonstrate that the proposed model is capable of producing simple captions from the seed photos.

[2] Caption Generation : A Shared Perspective:

Artificial intelligence (AI) technologies used to translate the image's sequence of pixels into words are not as new as they were five or more years ago. Smooth and effective image captioning is now achievable in a variety of contexts, from social media to e-commerce, thanks to improved performance, accuracy, and dependability. With a downloaded photo, tags are automatically created. This technology might make the world more accessible to the blind.

This report discusses picture captioning technology use cases, its fundamental design, benefits, and drawbacks. Also, we use a model that can accurately describe the contents of the input image.

Image captioning could be resolved using computer vision and NLP as a vision-language aim. Convolutional neural networks, recurrent neural networks, and other appropriate models are incorporated by the AI component to achieve the goal.

[3] Captioning using Attention Mechanism :

A weighted average on the encoded vectors is generated at each time step to direct the caption decoding process in the present encoder/decoder frameworks for picture captioning. Nevertheless, the decoder has minimal knowledge of the relationship between the attended vector and the provided attention query, which could lead to inaccurate answers.

This research propose an Attention on Attention module, which extends the conventional attention mechanisms to determine the relevance between attention results and queries. Attention on Attention first generates an information vector and an attention gate using the attention result and the current context, then adds another attention by applying element-wise multiplication to them and finally obtains the attended information, the expected useful knowledge.

[4] RNN Perspective :

In order to use RNNs for picture captioning, one must approach the task as a problem of sequence generation, with the objective of coming up with a string of words that would adequately describe the image. The RNN uses the image as input to produce a series of words, each of which is conditioned on the words that came before it.

A convolutional neural network (CNN) and a recurrent neural network are the two primary parts of a standard architecture for image captioning employing RNNs to accomplish this (RNN).

High-level characteristics are extracted from the image using the CNN. A sequence of convolutional and pooling layers are applied to the input image to provide a fixed-length vector representation of the image. The items in the image, their spatial connections, and other significant aspects of the image are all described in this vector.

[5] Use of Densest :

A DenseNet produces densely connected blocks because each layer is feedforward coupled to every other layer. Each layer creates a set of new feature maps that are sent to all succeeding layers in the block after taking the feature maps of all layers in the block as input. This lowers the number of parameters and increases the network's efficiency by enabling the network to reuse characteristics computed in earlier layers.

Many dense blocks that are separated by transition layers make up a dense network. While the transition layers lower the amount of feature maps and spatial dimensions of the input before forwarding it to the next dense block, the dense blocks contain multiple layers of convolutional and pooling processes.

The main benefit of DenseNets is that they stimulate gradient flow and feature reuse, which help to solve the vanishing gradient issue in very deep networks. Due to employing fewer parameters than conventional CNNs, DenseNets can achieve state-of-the-art performance on a variety of image classification applications.

[6] NLP View Point :

Natural language processing (NLP) methods are employed in image captioning to produce textual descriptions of the images. Recurrent neural networks (RNNs), which take an image as input and produce a string of words that represent the contents of the image, are commonly used to generate the captions. The calibre of the features that are taken from the image, however, has a significant impact on the quality of the generated captions.

A good product always combines the various essential and accessible tech stacks. The stacks listed above are either extremely important or are only available in image captions I believe.

Chapter 3 – Applications

The proposed solution can be utilized best in Fashion and Home Décor industry.

- Automatic Image Annotations for Blind People
- Clothing Patterns
- Home Décor Patterns
- Footwear Patterns
- Bags/Purse Patterns
- AI Image Captioning for social media



Clothes Patterns

1. Match the vocabulary with the pictures:



1. Write the missing words:

_____ boots	
_____ shirt	
_____ trousers	
_____ hat	
_____ socks	
_____ skirt	



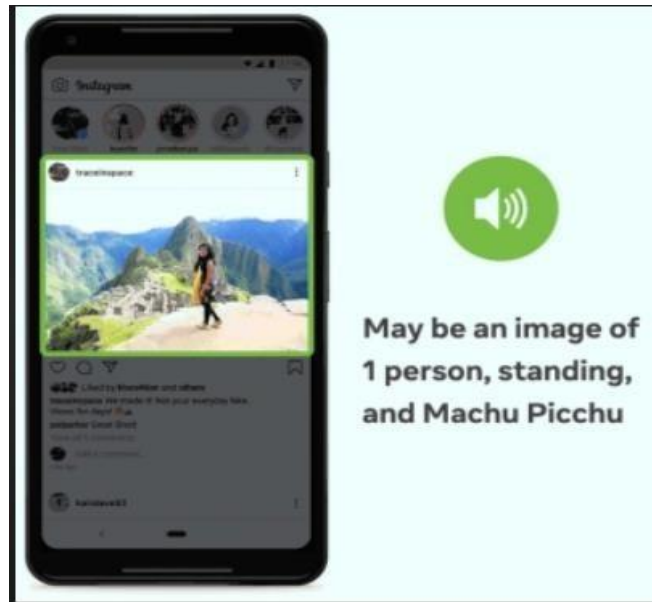


Figure 2: Applications of Caption Generation

Chapter 4 - Proposed Methodology

The following section will cover the scraped dataset details, followed workflow, metric functions to check the performance of the models.

1. Materials Used

Below listed are the materials and tools used:

- Google Colab – For executing Python code
- Keras, Tensorflow - A deep learning open source library along with required packages to train deep learning model
- Matplotlib – To visualize the results
- OpenCV

2. Configuration Used for Training

- RAM: 83.5 GB
- GPU: 40 GB
- DISK SPACE: 166.8 GB

Python 3 Google Compute Engine backend (GPU)
Showing resources since 10:32 PM

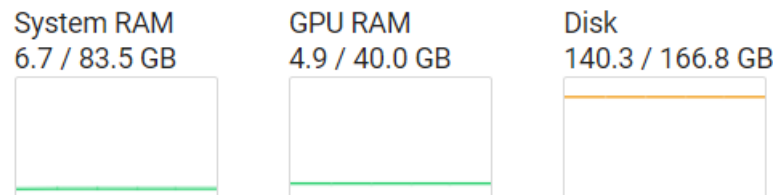


Figure 3: Configuration

3. Dataset details

Images were imported from internet using Kaggle API extensions and python program to simplify the extraction process.

4. Workflow diagram:

The workflow diagram demonstrates the different detection steps and procedures used to pinpoint issues in retinal or fundus images that are supplied with the set of modules. The following modules are used to implement the current system:

- Load Images
- Caption Text Pre-processing Steps
- Tokenization and Encoded Representation
- Image Feature Extraction
- Custom Data Generation
- Modelling
- Train Model LSTM / Attention

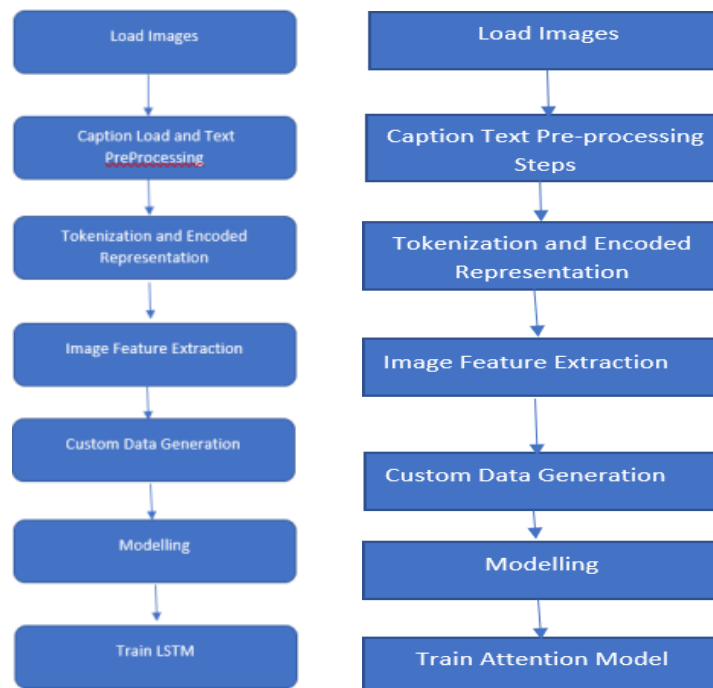


Figure 4: Workflow Diagram

5. Preprocessing Steps:

5.1 Resize Image:

The first step is to resize all the images to the same shape. All the images in our dataset are of varying sizes. Thus, converting all of them to $224 * 224 * 3$.

5.2 Tokenization:

- The words in a sentence are separated/tokenized and encoded in a one hot representation
- These encodings are then passed to the embeddings layer to generate word embeddings

Chapter 5 – Comparison of Caption generation Approaches

	CNN-LSTM	ATTENTION
PROS	It can depict how the words in the caption are related in time. This enables the model to produce relevant captions that accurately reflect the content of the image.	It enables the model to generate the caption while paying attention to particular areas of the image.
CONS	The application and training of LSTMs can be computationally expensive, especially when dealing with huge datasets or complicated models.	To train attention mechanisms to effectively focus on particular portions of the input data, huge volumes of training data are needed.

Table 1: Comparison of Caption generation Approaches

Chapter 6 – LSTM-CNN Models

In this project CNN-LSTM models are used. This section will describe the architectures of each model that are used.

The image embedding representations are concatenated with the first word of sentence ie. starseq and passed to the LSTM network. The LSTM network starts generating words after each input thus forming a sentence at the end.

LSTM Model:

```

▶ input1 = Input(shape=(1920,))
  input2 = Input(shape=(maximum_len,))

  img_features = Dense(256, activation='relu')(input1)
  img_features_reshap = Reshape((1, 256), input_shape=(256,))(img_features)

  sentenceFeatures = Embedding(vocabulary_length, 256, mask_zero=False)(input2)
  merged = concatenate([img_features_reshap, sentenceFeatures], axis=1)
  sentenceFeatures = LSTM(256)(merged)
  x = Dropout(0.5)(sentenceFeatures)
  x = add([x, img_features])
  x = Dense(128, activation='relu')(x)
  x = Dropout(0.5)(x)
  output = Dense(vocabulary_length, activation='softmax')(x)

  caption_model = Model(inputs=[input1, input2], outputs=output)
  caption_model.compile(loss='categorical_crossentropy', optimizer='adam')

```

Figure 5: LSTM Architecture

The model's architecture demonstrates how an input words are processed before going through a few dense layers and LSTM layer with the ReLU activation function.

The model's architecture demonstrates how an input noise is processed before going through a few convolutional layers with the ReLU activation function and Batch Normalization.

CNN Model : To be used for Feature extraction

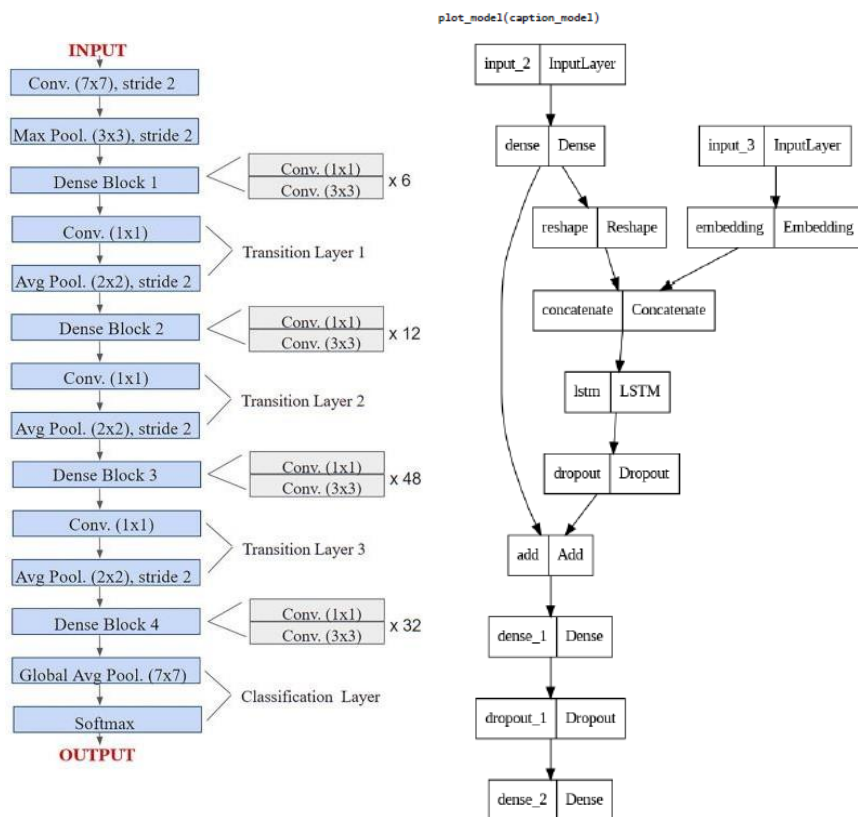


Figure 3 : DenseNet201 Architecture

Fig : DenseNet Architecture
 Fig : Caption Model Architecture

Chapter 7 – Pattern Generation Progress

LSTM

In order to generate the captions, I have feeded few random images there and checked the caption generation

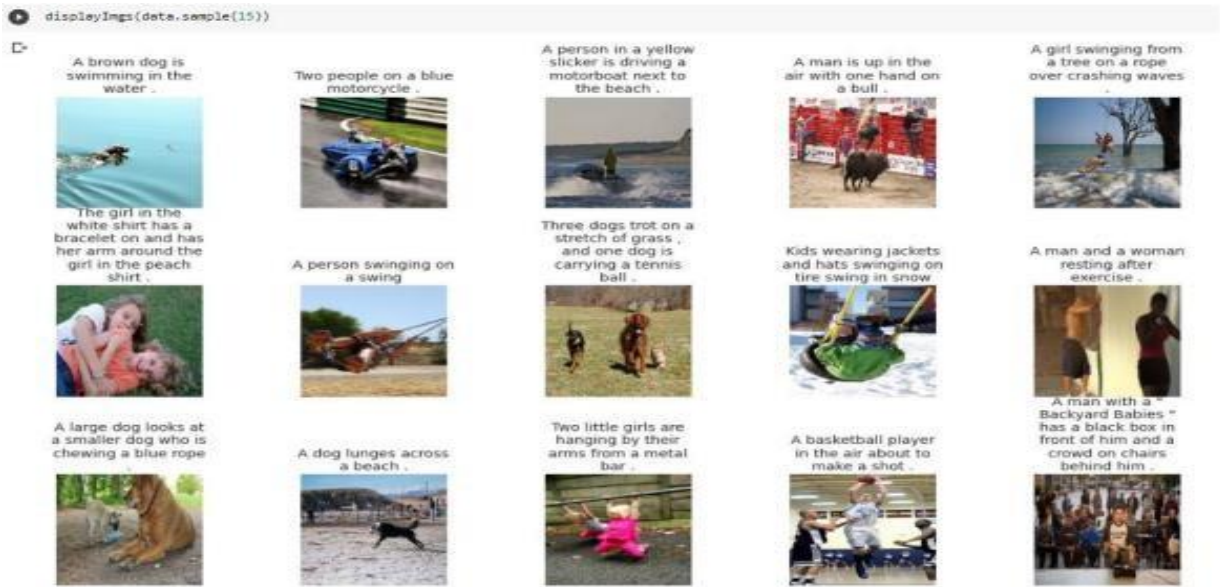


Figure 4 – Before Modeling

Figure 5 – After Modeling

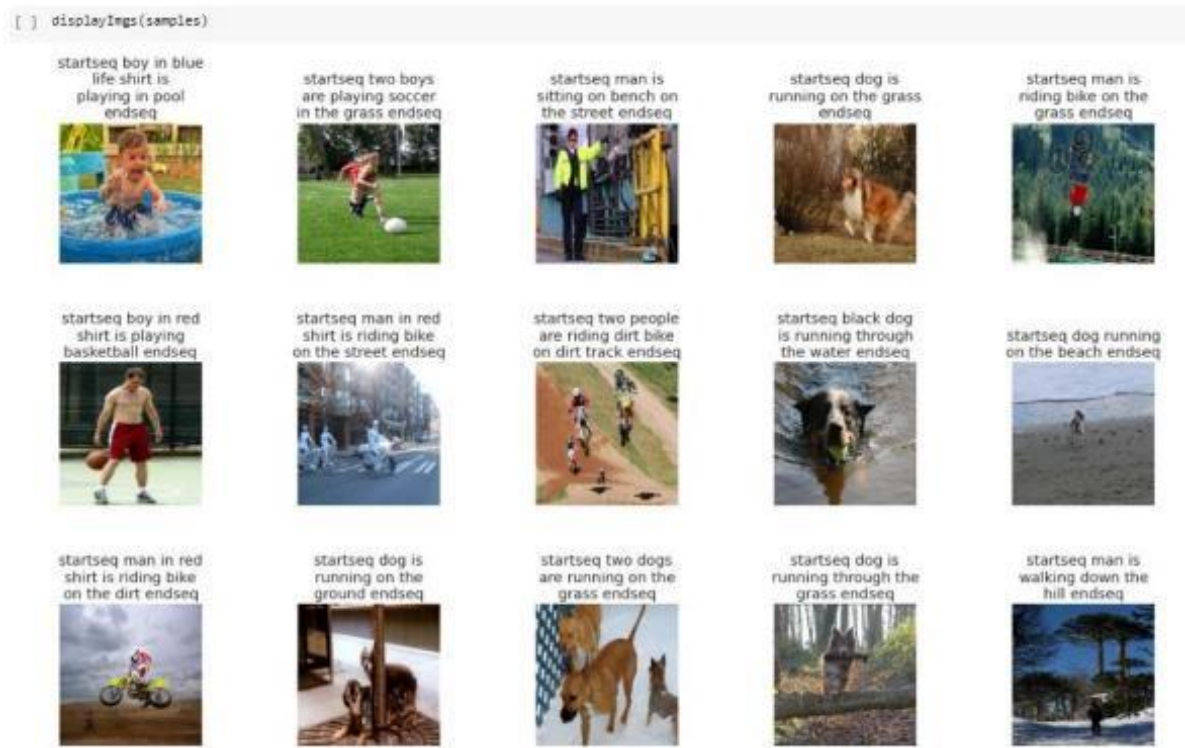


Figure 7: Training and Generated Images

LSTM generate the caption with accuracy by feeding the image in different-different level of architecture.

NOTE : Image has been taken for reference.

e.g. :

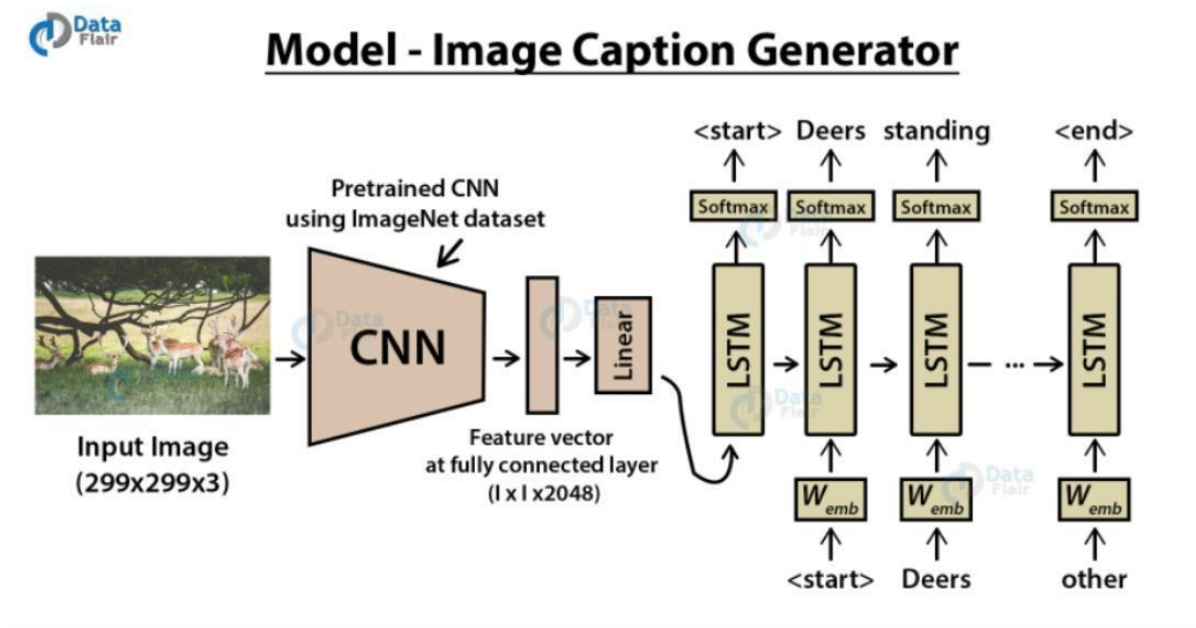


Figure 8: Image Caption Generator LSTM

Image caption generator is somehow a way to identify / generate the caption as per the feed image. Once input image gets detected to system, it pretrained into CNN using flickr8k data where it passes through different-different layers to the system.

Layers of CNN-LSTM Model

Chapter 8: Attention Model

The three processes that commonly make up the attention mechanism are computing the relevance of each input component to the present output, computing a weighting for each component depending on its relevance, and merging the weighted inputs to produce the output.

The ability to selectively focus on various elements of the input sequence or image is one of the key benefits of attention models. This feature can be especially helpful for jobs like image captioning, where some elements of the image may be more crucial than others. Due to their ability to filter out unimportant portions of the input, attention models have the potential to be more effective than conventional sequence-to-sequence models.

```

▶ input1 = Input(shape=(1920,))
  input2 = Input(shape=(maximum_len,))

  img_features = Dense(256, activation='relu')(input1)
  img_features_reshap = Reshape((1, 256), input_shape=(256,))(img_features)

  sentenceFeatures = Embedding(vocabulary_length, 256, mask_zero=False)(input2)
  merged = concatenate([img_features_reshap,sentenceFeatures],axis=1)
  sentenceFeatures = LSTM(256)(merged)
  att = Attention()([merged,sentenceFeatures])
  x = Dropout(0.5)(att)
  x = add([x, img_features])
  x = Dense(128, activation='relu')(x)
  x = Dropout(0.5)(x)
  output = Dense(vocabulary_length, activation='softmax')(x)

  Attmodel = Model(inputs=[input1,input2], outputs=output)
  Attmodel.compile(loss='categorical_crossentropy',optimizer='adam')

```

Figure 10: Attention Model Architecture

The relevance of each component to the current output determines which elements of the input sequence or image the attention model will pay attention to at each time step. As a result, the model's focus can be adjusted dynamically, producing output sequences that are more precise.

The weighted Attention Score is computed by the Attention Module using the encoded image from the Encoder and the concealed state from the Sequence Decoder.

The input sequence is merged with the Attention Score after being sent through the Embedding layer.

The Sequence Decoder receives the combined input sequence and generates an output sequence along with a new hidden state. After analyzing the output sequence, the sentence generator produces its anticipated word probabilities.

This cycle is now repeated for the following timestep. The following timestep uses the updated hidden state of the Decoder from this timestep. Unless a "End" token is predicted or the sequence reaches its maximum length, we keep doing this.

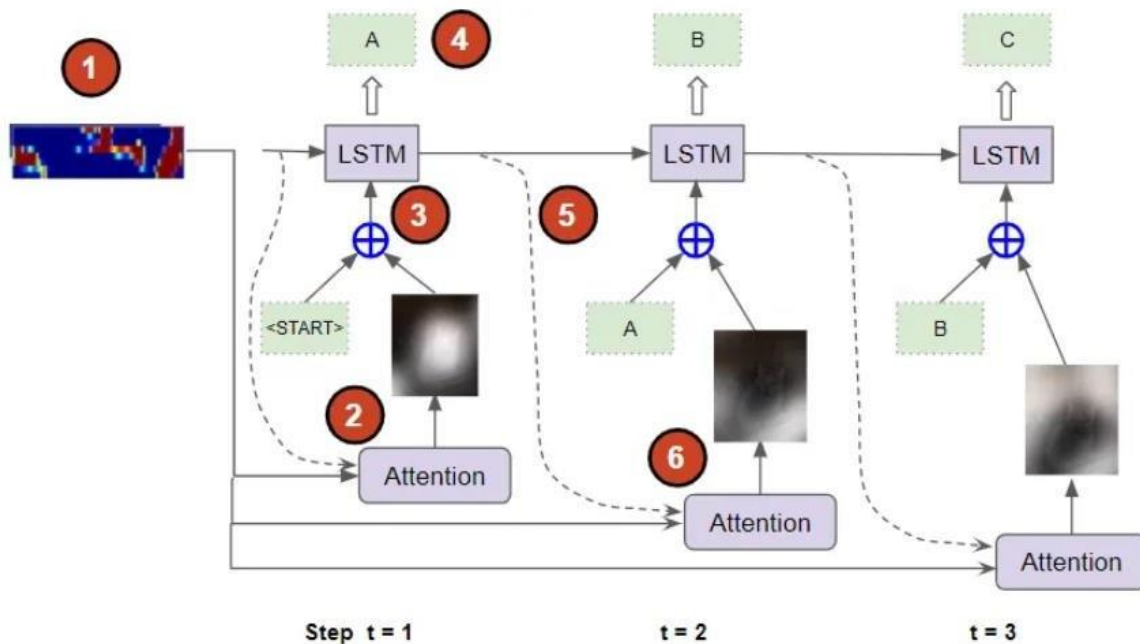


Figure 11: Attention Model Working in Modeling

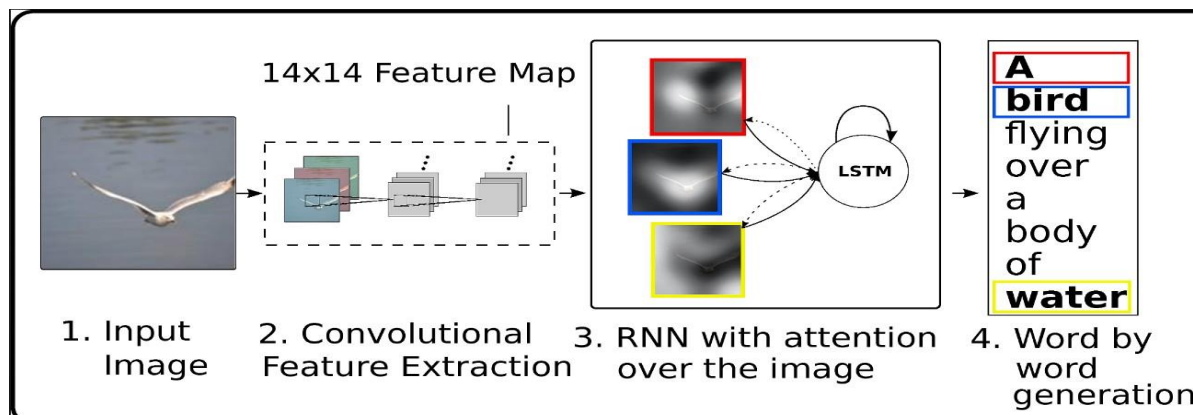


Figure 12 : Larger Picture for Attention Mechanism

Chapter 9 – Pattern Generation Progress ATTENTION :

Caption generation in Attention is specifically emphasize the detail of an Image

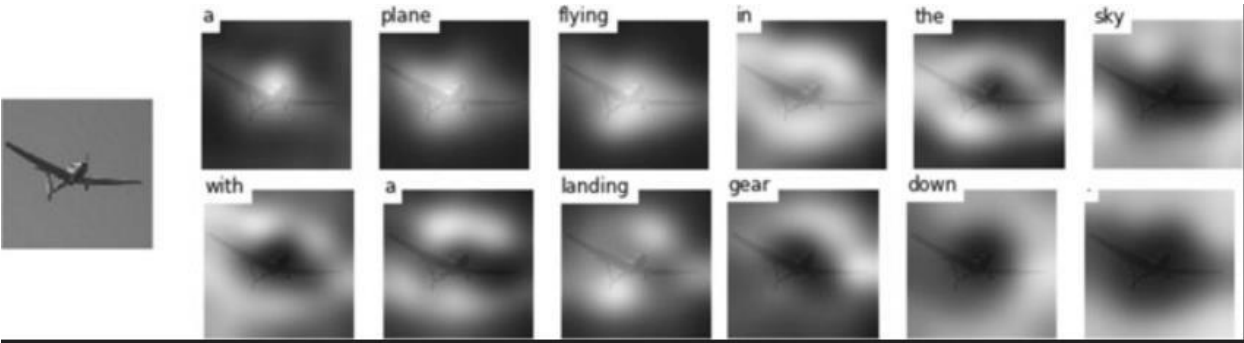


Figure 13: An Image in Detailed Form in Attention Mechanism



Before Modeling
Figure 14: Before Applying Attention Model

After Modeling

```
display_images(samples)
```

startseq startseq
dog is running
through the water
endseq



startseq startseq
boy in red shirt is
climbing rock wall
endseq



startseq startseq
two dogs are playing
in the air endseq



startseq startseq
two children are
playing in the grass
endseq



startseq startseq
man in blue shirt is
playing with ball
endseq



startseq startseq
man is climbing rock
endseq



startseq startseq
man in blue shirt is
sitting on the top
of rock endseq



startseq startseq
two children are
playing in the water
endseq



startseq startseq
the man is holding
his head endseq



startseq startseq
young boy is sitting
in the floor endseq



startseq startseq
little boy is
sitting on the floor
endseq



startseq startseq
woman in blue shirt
is sitting on the
floor endseq



startseq startseq
two dogs are running
in the grass endseq



startseq startseq
dog is running
through the snow
endseq



startseq startseq
man in red jacket is
riding bike endseq



Figure 15: After Applying Attention Model

Chapter 10 – Attention Model Training

Model Parameters:

- Optimizer: Adam
- Learning Rate: 2e-4
- Epochs: 50 (But trained till 10)
- Early Stopping : At 15
- Image Size: 224

```
Epoch 1/50
537/537 [=====] - ETA: 0s - loss: 5.8958
Epoch 1: val_loss improved from inf to 5.66698, saving model to model.h5
537/537 [=====] - 283s 515ms/step - loss: 5.8958 - val_loss: 5.6670 - lr: 0.0010
Epoch 2/50
537/537 [=====] - ETA: 0s - loss: 5.6548
Epoch 2: val_loss improved from 5.66698 to 5.61720, saving model to model.h5
537/537 [=====] - 263s 490ms/step - loss: 5.6548 - val_loss: 5.6172 - lr: 0.0010
Epoch 3/50
537/537 [=====] - ETA: 0s - loss: 5.5813
Epoch 3: val_loss improved from 5.61720 to 5.57686, saving model to model.h5
537/537 [=====] - 262s 487ms/step - loss: 5.5813 - val_loss: 5.5769 - lr: 0.0010
Epoch 4/50
537/537 [=====] - ETA: 0s - loss: 5.5255
Epoch 4: val_loss improved from 5.57686 to 5.55377, saving model to model.h5
537/537 [=====] - 262s 487ms/step - loss: 5.5255 - val_loss: 5.5538 - lr: 0.0010
Epoch 5/50
537/537 [=====] - ETA: 0s - loss: 5.4646
Epoch 5: val_loss improved from 5.55377 to 5.51946, saving model to model.h5
537/537 [=====] - 260s 484ms/step - loss: 5.4646 - val_loss: 5.5195 - lr: 0.0010
Epoch 6/50
537/537 [=====] - ETA: 0s - loss: 5.4115
Epoch 6: val_loss improved from 5.51946 to 5.49026, saving model to model.h5
537/537 [=====] - 257s 479ms/step - loss: 5.4115 - val_loss: 5.4903 - lr: 0.0010
Epoch 7/50
537/537 [=====] - ETA: 0s - loss: 5.3644
Epoch 7: val_loss improved from 5.49026 to 5.47294, saving model to model.h5
537/537 [=====] - 254s 473ms/step - loss: 5.3644 - val_loss: 5.4729 - lr: 0.0010
Epoch 8/50
537/537 [=====] - ETA: 0s - loss: 5.3172
Epoch 8: val_loss improved from 5.47294 to 5.45745, saving model to model.h5
537/537 [=====] - 255s 475ms/step - loss: 5.3172 - val_loss: 5.4575 - lr: 0.0010
Epoch 9/50
537/537 [=====] - ETA: 0s - loss: 5.2755
Epoch 9: val_loss improved from 5.45745 to 5.44195, saving model to model.h5
537/537 [=====] - 256s 477ms/step - loss: 5.2755 - val_loss: 5.4419 - lr: 0.0010
Epoch 10/50
537/537 [=====] - ETA: 0s - loss: 5.2333
Epoch 10: val_loss improved from 5.44195 to 5.42662, saving model to model.h5
```

Figure 16 : Attention Model Training

```

Epoch 10/50
537/537 [=====] - ETA: 0s - loss: 5.2333
Epoch 10: val_loss improved from 5.44195 to 5.42662, saving model to model.h5
537/537 [=====] - 254s 473ms/step - loss: 5.2333 - val_loss: 5.4266 - lr: 0.0010
Epoch 11/50
537/537 [=====] - ETA: 0s - loss: 5.1919
Epoch 11: val_loss did not improve from 5.42662
537/537 [=====] - 254s 473ms/step - loss: 5.1919 - val_loss: 5.4291 - lr: 0.0010
Epoch 12/50
537/537 [=====] - ETA: 0s - loss: 5.1520
Epoch 12: val_loss did not improve from 5.42662
537/537 [=====] - 256s 476ms/step - loss: 5.1520 - val_loss: 5.4268 - lr: 0.0010
Epoch 13/50
537/537 [=====] - ETA: 0s - loss: 5.1143
Epoch 13: val_loss did not improve from 5.42662

Epoch 13: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
537/537 [=====] - 255s 475ms/step - loss: 5.1143 - val_loss: 5.4335 - lr: 0.0010
Epoch 14/50
537/537 [=====] - ETA: 0s - loss: 5.0338
Epoch 14: val_loss did not improve from 5.42662
537/537 [=====] - 254s 473ms/step - loss: 5.0338 - val_loss: 5.4285 - lr: 2.0000e-04
Epoch 15/50
537/537 [=====] - ETA: 0s - loss: 5.0136
Epoch 15: val_loss did not improve from 5.42662
Restoring model weights from the end of the best epoch: 10.
537/537 [=====] - 254s 472ms/step - loss: 5.0136 - val_loss: 5.4297 - lr: 2.0000e-04
Epoch 15: early stopping

```

Figure 17: Attention Model Training Cont.

Chapter 11 - Conclusion

For sequence modelling and natural language processing, two common deep learning techniques are LSTM (Long Short-Term Memory) and attention models.

Recurrent neural networks (RNNs) of the LSTM variety are made to manage long-term dependencies in sequence data. It is commonly utilized in applications including speech recognition, language modelling, and machine translation.

Contrarily, attention is a mechanism that allows the model to concentrate on those portions of the input sequence that are crucial to the task at hand. It is now a crucial part of many cutting-edge neural architectures for tasks involving natural language processing, such as question answering, machine translation, and summarization.

In this paper, there is a working and comparison of LSTM and ATTENTION has been explained. Both the models have pros and cons. LSTM were able to show results at 12 epochs while Attention models did the same at 15 epochs. The model training computation was higher in case of Attention models and took more time than LSTM, but Attention model generated a bit accurate image. Here in case of Attention Model we pass the Image from Encode and Decoder which adds the additional layer of sharpening; thus model is gets trained with more sophistication but in contrast LSTM is not having such mechanism. Attention models are computationally heavy and expensive thus we should choose this the model as per the need and size of data.

Chapter 12 – Future Works

Future research will concentrate on improving the efficiency, speed, and reliability of caption creation algorithms or processes. In addition, more complex models will be used in this project, which can capture more characteristics and yield more accurate findings.

Chapter 13 - References


1. "Show and Tell: A Neural Image Caption Generator" by Vinyals et al. (2015) - This paper introduced the first end-to-end neural network model for image captioning, which uses a convolutional neural network (CNN) to extract image features and a long short-term memory (LSTM) network to generate captions.
2. "Deep Visual-Semantic Alignments for Generating Image Descriptions" by Karpathy and Fei-Fei (2015) - This paper introduced a model that learns a joint embedding of images and their corresponding captions, allowing for the generation of captions for new images.
3. "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering" by Anderson et al. (2018) - This paper introduced a model that uses a bottom-up attention mechanism to extract salient image features and a top-down attention mechanism to generate captions.
4. "Neural Baby Talk" by Lu et al. (2018) - This paper introduced a model that generates captions in a "baby-talk" style, which has been shown to improve the interpretability and accuracy of image captioning models.
5. "Unified Visual-Semantic Embeddings: Bridging Vision and Language with Structured Meaning Representations" by Zhu et al. (2018) - This paper introduced a model that learns a unified embedding space for images and their corresponding captions, enabling the generation of captions for new images and the retrieval of images based on textual queries.
6. "Aligning Cross-Domain Images and Texts: A Domain Adaptation Approach to Image Captioning" by Chen et al. (2020) - This paper introduced a domain adaptation approach to image captioning, which enables the model to generalize to new domains with limited labeled data.
7. "Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning" by Lu et al. (2017) - This paper introduced a model that uses an adaptive attention mechanism with a visual sentinel to determine which parts of the image to focus on at each time step of the caption generation process..
8. "Image Captioning with Semantic Attention" by You et al. (2016) - This paper introduced a model that uses semantic attention to attend to different regions of the image based on the semantics of the words being generated in the caption.
9. "Look, Imagine and Match: Improving Textual-Visual Cross-Modal Retrieval with Generative Models" by Zhang et al. (2018) - This paper introduced a model that uses a generative adversarial network (GAN) to improve the alignment between textual and visual features, improving both image captioning and visual-textual retrieval tasks.
10. "Self-critical Sequence Training for Image Captioning" by Rennie et al. (2017) - This paper introduced a training strategy that uses reinforcement learning to optimize caption generation, resulting in improved caption quality compared to traditional maximum likelihood training..
11. "Oriented Scene Text Detection with Sequentially Optimized Detection Networks" by He et al. (2018) - This paper introduced a model for detecting and recognizing text in natural scenes, which can be useful for generating captions for images with text.
12. "Multimodal Residual Learning for Visual QA" by Fukui et al. (2016) - This

paper introduced a model that combines image and text modalities for visual question answering, which involves generating a textual answer to a question about an image.

13. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer" by Raffel et al. (2019) - This paper introduced a transformer-based language model that can be fine-tuned for a wide range of natural language processing tasks, including image captioning.
14. "Multimodal Transformer for Unaligned Multimodal Language Sequences" by Chen et al. (2020) - This paper introduced a model that uses a transformer-based architecture to generate captions for multimodal inputs, such as images with associated audio or text.
15. "X-Linear Attention Networks for Image Captioning" by Yang et al. (2021) - This paper introduced a model that uses a novel X-linear attention mechanism to attend to both spatial and channel dimensions of image features, resulting in improved captioning performance..
16. Smriti P. Manay, Smruti A. Yaligar, Y. Thathva Sri Sai Reddy & Nirmala J. Saunshimath,
17. Ansar Hani; Najiba Tagougui; Monji Kherallah' Image Caption Generation Using A Deep Architecture. Feb,2020.
18. "Adaptive Attention Span in Transformers for Image Captioning" by Chen et al. (2021) - This paper introduced a model that uses an adaptive attention span mechanism in transformer-based architectures to dynamically adjust the receptive field of the model, resulting in improved captioning performance.
19. "Up-Down Captioner: A Multi-Scale Perceptual Up-Down Model for Generating Image Descriptions" by Anderson et al. (2018) - This paper introduced a model that uses a multi-scale approach to generate image captions, leveraging both global and local features of the image.
20. "Transformer-based Joint Training for Multimodal Translation and Image Captioning" by Wang et al. (2021) - This paper introduced a model that uses a transformer-based architecture to jointly train image captioning and multimodal translation tasks, improving the alignment between different modalities.
21. "Dense Captioning with Joint Inference and Visual Context" by Johnson et al. (2016) - This paper introduced a model that generates multiple captions for an image, each focusing on different regions of the image, providing a more detailed description of the scene

Check list of items for the Final report

- | | |
|---|------------------|
| a) Is the Cover page in proper format? | Y / N |
| b) Is the Title page in proper format? | Y / N |
| c) Is the Certificate from the Supervisor in proper format? Has it been signed? | Y / N |
| d) Is Abstract included in the Report? Is it properly written? | Y / N |
| e) Does the Table of Contents page include chapter page numbers? | Y / N |
| f) Does the Report contain a summary of the literature survey? | Y / N |
| i. Are the Pages numbered properly? | Y / N |
| ii. Are the Figures numbered properly? | Y / N |
| iii. Are the Tables numbered properly? | Y / N |
| iv. Are the Captions for the Figures and Tables proper? | Y / N |
| v. Are the Appendices numbered? | Y / N |
| g) Does the Report have Conclusion / Recommendations of the work? | Y / N |
| h) Are References/Bibliography given in the Report? | Y / N |
| i) Have the References been cited in the Report? | Y / N |
| j) Is the citation of References / Bibliography in proper format? | Y / N |

DocuSigned by:

 BC2C28FA32DB4AE...

Signature of the Supervisor

Name: Amit Kale

Date: 3/13/2023

DocuSigned by:

 7F221BBE9F13442...

Signature of Student

Name: Sachin Laxkar

Date: 3/13/2023