IJNRD.ORG ISSN : 2456-4184

International Research Journal
IJNRD
Research Through Innovation

INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

From Pixels To Text: Deep Learning Approach For Image Caption Generation

Naresh Sharma[1], Hari Om[2], Chozharanjan P[3]

[1] Student, Aeronautical Engineering Bannari Amman Institute of Technology, Tamil Nadu, India
[2] Student, Computer Science & Engineering Bannari Amman Institute of Technology, Tamil Nadu, India
[3]Assistant Professor I, Artificial Intelligence and Data Science Bannari Amman Institute of Technology, Tamil Nadu, India

ABSTRACT

In the realm of computer vision and natural language processing, the synthesis of image understanding and language comprehension has been a longstanding challenge Over the past decade, deep learning methodologies have emerged as powerful tools in addressing this challenge Several research works have focused on mitigating dataset bias, incorporating vision-language pre-training methods, and developing improved evaluation tools to enhance the quality of image captions The use of end end models has been highlighted as an impressive aspect, allowing the prediction of image captions without the need for complex data preparation or a pipeline of specifically designed models While the field of automatic image captioning presents numerous complexities, the combination of deep learning, computer vision, and natural language processing has facilitated significant progress Ongoing advancements in hardware and deep learning models are expected to further improve the accuracy of caption generation in the future Through a comprehensive review of existing methodologies and recent innovations, this paper outlines the evolution of image captioning techniques It discusses the pivotal role played by attention mechanisms in enhancing the quality and relevance of generated captions by focusing on salient regions of the image Furthermore, this paper sheds light on the challenges encountered in image caption generation, such as ambiguity in interpretation and diversity in visual content Strategies for mitigating these challenges, including reinforcement learning and adversarial training, are explored to push the boundaries of captioning performance In conclusion, this paper provides insights into the current state-of-the-art in image caption generation and offers directions for future research By integrating deep learning techniques with image understanding and language generation, the quest for machines to comprehend and describe visual content continues to evolve, opening new avenues for human-machine interaction and multimedia applications

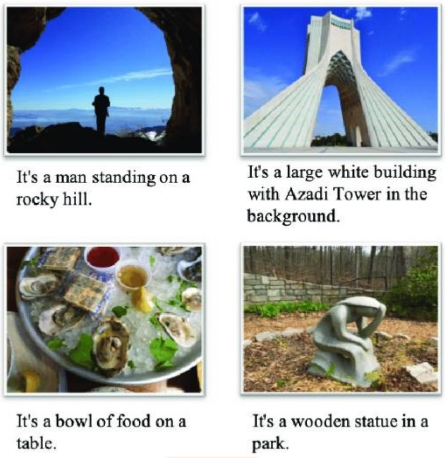
Keywords: *Deep Learning, Image Caption Generation, Neural Networks, CNN,RNN, Natural Language Processing*

1. INTRODUCTION:

Language serves as the primary means of communication, whether verbal or written, allowing individuals to articulate their perceptions of the visual world For those with physical challenges, images and signs offer alternative modes of understanding and communication However, automatically generating coherent descriptions from images presents a formidable challenge, as noted by recent research [1] Nevertheless, such endeavors hold immense potential to enhance the accessibility of visual content for the visually impaired, enabling them to comprehend images described on the web more effectively Describing an image effectively is often likened to "visualizing a picture in the mind," underscoring the importance of creating mental representations of visual stimuli This mental imagery plays a pivotal role in the generation of descriptive sentences, as humans can swiftly conjure descriptions after a cursory glance at an image Consequently, advancing our understanding of natural image descriptions is crucial for achieving complex objectives in human recognition

Automatically generating captions and descriptions for images presents a substantially greater challenge compared to tasks like image classification and object recognition Beyond merely identifying objects, image descriptions must encapsulate the relationships between objects, their attributes, and depicted activities [20] While significant progress has been made in labeling images with predefined classes or categories, the focus on closed vocabularies of visual concepts has resulted in notable advancements in the field However, this simplicity in modeling assumptions may not adequately capture the complexity of real-world visual scenes

Moreover, attention mechanisms have emerged as a crucial component in improving the quality and relevance of generated captions These mechanisms enable the model to dynamically focus on salient regions within the images, thus aligning the generated textual descriptions with the visual content more effectively


It's a man standing on a rocky hill.

It's a large white building with Azadi Tower in the background.

It's a bowl of food on a table.

It's a wooden statue in a park.

Fig1 Examples Of Image Captioning

LITERATURE REVIEW

Recent literature in image caption generation has introduced several innovative frameworks and techniques aimed at improving the accuracy and effectiveness of generating captions for images, particularly for assisting visually impaired individuals

Heng Song et al   proposed a Visual Text Merging (VTM) framework, employing an attention model to accurately merge visual and textual data for generating image captions Their adaptive VTM network demonstrated efficacy in merging data from COCO2014 and Flickr30K datasets, yielding promising performance scores based on various metrics like BLEU, CIDEr, ROUGE-L, SPICE, and METEOR

Deng et al [5] developed a framework called DenseNet+LSTM, which utilizes DenseNet for feature extraction from images and LSTM for caption creation This framework, evaluated on Flickr30K and COCO2014 datasets, demonstrated notable performance in generating captions beneficial for the visually impaired

Yuchen Wei et al [30] discussed the challenges visually impaired individuals face in everyday activities like shopping and proposed deep learning frameworks for accurate product identification through captions Their work emphasized the importance of accurate captions in facilitating retail store shopping for the visually impaired

Min Yang et al [34] introduced EnsCaption, a retrieval-dependent caption creation framework that combines caption creation and retrieval techniques By fusing personalized texts and employing an adversarial network, their approach achieved enhanced accuracy in caption generation, as demonstrated on Flickr-30K and MSCOCO datasets

Niange Yu et al [36] introduced a CNN-based multiple labeling classifier for image captioning, providing captions based on image topics Their hierarchical framework, coupled with a bi-directional caption retrieval procedure, yielded high-quality image captions

Fen Xiao et al [32] developed a dual LSTM image captioning framework integrated with an adaptive semantic attention mechanism, designed to enhance accessibility for blind individuals Their framework, evaluated on MSCOCO and Flickr30K datasets, achieved commendable performance in generating accurate text sequences for input images

Loganathan et al [17] presented an automatic caption generation framework utilizing CNN and LSTM architectures, demonstrating effective learning processes for accurate image captioning, particularly beneficial for the visually impaired

Singh et al [28] introduced an encoder-decoder framework for image captioning, utilizing CNN as the encoder for visual features and stacked LSTM for caption generation Their approach, validated on the Hindi genome dataset, exhibited promising performance in caption generation

Iwamura et al [10] proposed a trainable end-to-end approach for image caption generation, incorporating feature extraction, motion estimation, object detection, and caption generation phases Their approach, evaluated on multiple datasets including MSCOCO and MSR-VTT2016-Image, demonstrated significant improvements in captioning accuracy

Khurram et al [12] developed Dense-CaptionNet, a region-based deep learning model for image captioning, providing detailed descriptions of images through object relationship and region description modules Their approach, evaluated on datasets like MSCOCO, Visual Genome, and IAPR TC-12, showcased the ability to generate comprehensive image captions
Overall, recent advancements in image caption generation have shown promising results in improving accessibility and usability for visually impaired individuals, facilitating their understanding of the visual world through accurate and detailed image captions

3 OBJECTIVES:

The project's goal is to forecast the captions for the supplied image. The collection includes 5,000 photos and 5 descriptions for every image. Both the image and the text captions are used to extract the characteristics for input. The attributes will be combined to forecast the caption's subsequent word. CNN is utilized for images, and LSTM is utilized for text. The BLEU Score is a statistic employed to assess the efficacy of the trained model. The goal of picture captioning is to create intelligent systems that can automatically explain the contents of images in text using natural language. As seen in the architectural example below, recent deep neural network models have shown immense promise on this challenging problem The primary goal is to generate textual captions that capture not just salient objects, but also their attributes, interactions and the overall contextual scene depicted in a visual input image The key objective is to design and train an end-to-end framework that effectively encodes pixels to words - mapping raw image pixels into textual vocabulary based on robust feature representations Specifically, a typical paradigm utilizes convolutional neural networks (CNNs) as image encoders to extract semantic features which feed into a recurrent neural network (RNN) decoder to produce sentence sequences as captions Additional elements like attention mechanisms and text generation regularization further improve relevance and coherence

4 METHODOLOGY:

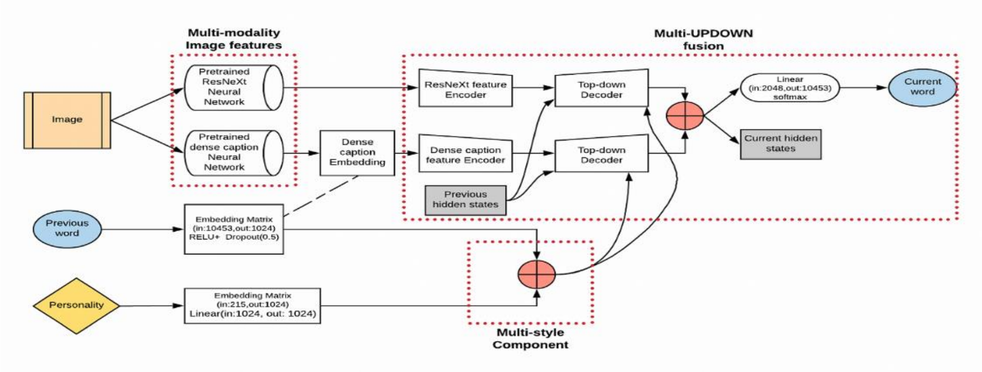

Multi-modality Image features
Multi-UPDOWN fusion
Image
Pretrained ResNeXt Neural Network
Pretrained dense caption Neural Network
Dense caption Embedding
ResNeXt feature Encoder
Dense caption feature Encoder
Previous hidden states
Top-down Decoder
Top-down Decoder
Linear (in:2048,out:10453) softmax
Current word
Current hidden states
Previous word
Embedding Matrix (in:10453,out:1024) RELU+ Dropout(0.5)
Personality
Embedding Matrix (in:215,out:1024) Linear(in:1024, out: 1024)
Multi-style Component

Fig 2: Schematic model of proposed method

Recently, there has been growing interest in Caption Generation (CG) within computer vision, driven by its wide-ranging applications such as virtual assistants, image understanding, and assisting the visually impaired Automatic Caption Generation (ACG) systems are particularly valuable for the blind, offering them a means to better comprehend their surroundings While sighted individuals can quickly grasp the content of images, blind individuals rely on other senses like touch and hearing for assistance in understanding nearby objects

This study aims to introduce a novel approach to image captioning using the deep learning model called ECANN (Extended Convolutional Atom Neural Network), which seeks to emulate human-like understanding of the visual world The schematic model of the proposed method is illustrated in Figure 1

The Automatic Image Captioning (AIC) system generates simple captions for images lacking them The proposed AIC, employing the ECANN model, proceeds through several stages: Data Collection, Non-captioned Image Selection, Extraction of Appearance and Texture Features, and Generation of Automatic Image Captions

Initially, non-captioned images are selected from the database using the ARO algorithm Subsequently, appearance and texture features are extracted utilizing Spatial Derivative & Multi-Scale (SDM) features and Weighted Patch-Based Local Binary Pattern (WPLBP) These extracted features facilitate accurate differentiation between images Finally, captions are generated for corresponding images using the ECANN architecture

The ECANN-based alternate image captioning process introduces a caption-reusable system based on Artificial Intelligence (AI) with a reverse image search to reuse pre-existing captions for the target image However, the proposed ECANN model is primarily utilized to generate alternate captions that are semantically faithful to the original image Within this framework, error occurrence in image captioning is minimized through the use of the AAS algorithm

2. DATASET: The standard dataset, Flickr8k, was downloaded from Kaggle and is the one that we utilized. The collection has 8000 photos in total, with five descriptions for each image, allowing for the coverage of all potential circumstances. This dataset includes the following: Flickr_8k trainImages txt file, which has 6000 images and is the training dataset; Flickr_8k devImages txt file, which has 1000 images and is the validation/development dataset; and finally, Flickr_8k testImages txt file, which has 1000 images and is the test dataset. The photographs in the dataset are sourced from six groups on Flickr and do not feature any well-known individuals or locations. The photographs in the dataset are of various scenes that have been hand-picked. The dataset size is 1GB which is available and can be freely downloaded from Kaggle.

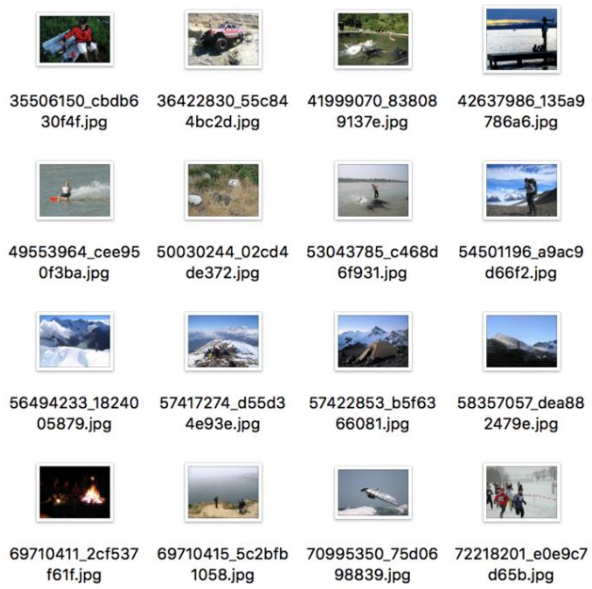


Fig3 flicker8k Dataset Glimpse

A. Task

Building a system that can take a picture as input in the form of a dimensional array and produce an output consisting of a sentence that describes the image and is syntactically and grammatically correct is the task required.

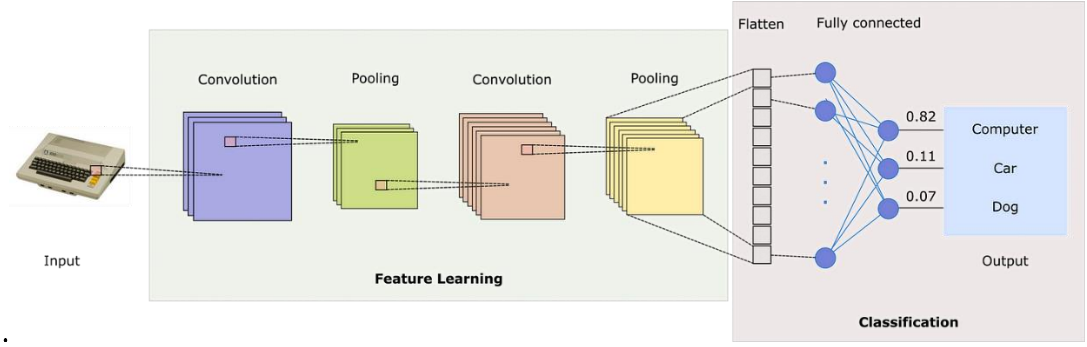
B. Preprocessing

Data preparation is done in two parts: photos and captions are cleaned and pre-processed independently. The method of preparing images involves supplying the input data to the Xception program, which is an application of the Keras API that operates on top of TensorFlow. Because Xception has already been pre-trained on ImageNet, we were able to train the photos more quickly with the use of transfer learning. The descriptions are cleaned using Keras' tokenizer class, which vectorizes the text corpus and stores it in a different dictionary. Next, every word in the lexicon is mapped to a distinct index value.

C. Model

Deep learning employs an artificial neural network structured in multiple layers, creating a hierarchical architecture for machine learning tasks. The model is built upon deep networks, where data is processed sequentially from the input layer onward. This iterative process continues indefinitely as each layer in the network progressively generates more complex representations based on the input received from the preceding layer.

Convolutional Neural Networks (CNN) Convolutional Neural Networks (CNNs) are specialized deep neural networks designed to process data in the form of 2D matrices, making them particularly adept at handling images, which can easily be represented in this format. Recognized as indispensable in image-related tasks, CNNs are extensively discussed in the International Journal of Innovative Technology and Exploring. Functioning by receiving images as input, CNNs allocate significance (weights and biases) to various features or objects within the image, enabling differentiation between them. Key to their operation are filters, also known as kernels, which aid in learning features such as blurring, edge detection, sharpening, and other abstract concepts—akin to how the human brain identifies objects in both temporal and spatial dimensions.



.

Fig .4 CNN Architect

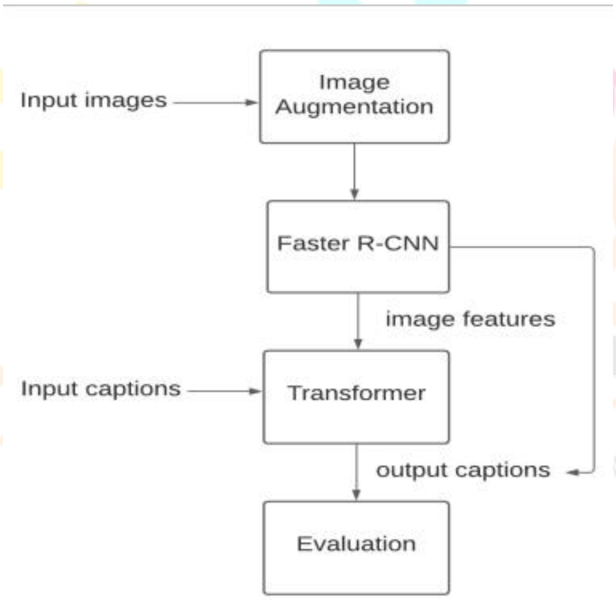


Fig.5 Proposed model

Recurrent Neural Networks (RNN) The human brain has evolved to interpret preceding words and utilize them in generating subsequent words, thereby constructing coherent sentences. However, conventional neural networks lack this capability. Recurrent neural networks (RNNs), featuring loops within their architecture, address this limitation by enabling information persistence through internal states, thus establishing a feedback loop.Long Short-Term Memory networks (LSTMs), a specialized form of RNNs, excel in learning long-term dependencies. Their default behavior involves retaining information over extended periods, regulated by mechanisms called "gates." Unlike RNNs, which process individual data points, LSTMs can handle entire sequences. Moreover, they possess the ability to discern the significance of data points and discard irrelevant information, ensuring that only pertinent data is propagated to subsequent layers.

The primary gates - input gate, output gate, and forget gate - play a crucial role in this process. They determine whether to retain the current cell value, incorporate a new value into the cell, or output the cell value. Hidden states are instrumental, as they carry forward previous information to subsequent steps in the sequence, serving as the neural network's memory repository.network has seen before Thus it allows the neural network to function like a human brain trying to form sentences

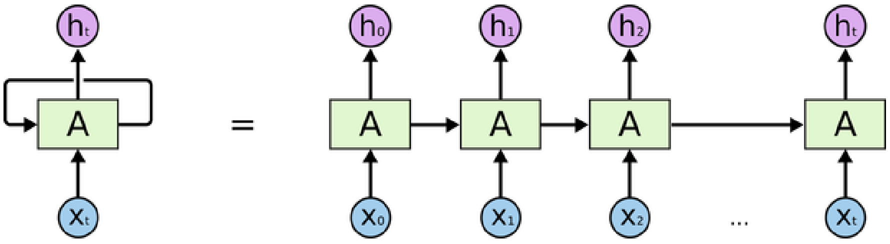


Fig 6 RNN architecture

5 Architecture

The framework for generating captions for images combines Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) networks. Traditionally, an "encoder" RNN is employed to convert the variable-length source sentence into a fixed-length vector representation. This vector then serves as the initial hidden state for a "decoder" RNN,

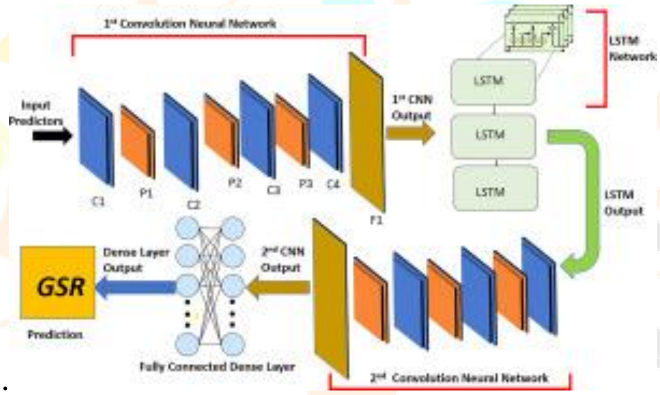


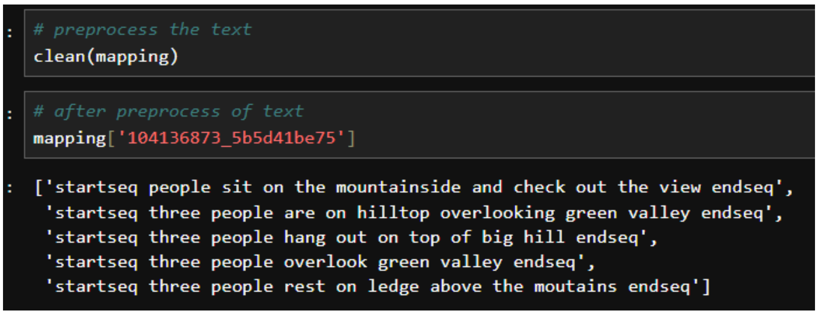
Fig 7 Cnn- Rnn arrchitect

However, in our methodology, we substitute the conventional RNN with a deep CNN. This CNN undergoes pre-training for image classification tasks, enabling it to generate a comprehensive representation of the input image by embedding it into a fixed-length vector. Subsequently, we utilize the last hidden layer of this CNN as the input for the RNN decoder, which is responsible for generating the sentences. This architectural design, illustrated in the integration of CNN and LSTM for image captioning, positions the CNN as the feature extractor, delivering a condensed representation of the input image. This representation is then passed on to the LSTM decoder to produce captions.

6 EVALUTION

A. Data Cleaning and Pre-processing:

The program begins by loading both the text file and the image file into separate variables. The content of the text file is stored as a string, which will later undergo manipulation to create a dictionary mapping each image to a list of five descriptions.



```
: # preprocess the text
  clean(mapping)

: # after preprocess of text
  mapping['104136873_5b5d41be75']

: ['startseq people sit on the mountainside and check out the view endseq',
   'startseq three people are on hilltop overlooking green valley endseq',
   'startseq three people hang out on top of big hill endseq',
   'startseq three people overlook green valley endseq',
   'startseq three people rest on ledge above the moutains endseq']
```

Fig 8 data cleaning

of a dictionary that will map every image to a list of five descriptions The principal duty in the data cleaning procedure is to eliminate punctuation from the captions. Furthermore, any content including capital characters is removed; only lowercase letters are tolerated. Words with numerals are also eliminated. A vocabulary is then developed, consisting of unique terms from each description. This language will be used in creating captions for the test photographs. Each caption must be tagged and identified. These tags help the LSTM determine where a caption begins and stops.

B. Extraction of feature vectors:

A feature vector, representing essential characteristics of an object, is a numerical matrix providing information about those features. For instance, in our context, each pixel of an image holds a unique intensity value. Feature vectors are stored in a pickle file. In our model, we employ Transfer Learning, utilizing the pre-trained VGG-16 model for feature extraction. The VGG-16 model consists of 16 layers and has been trained on the ImageNet dataset, comprising over 1000 diverse classes and millions of images. This model can be easily imported using the keras applications module in Python. Since the VGG-16 model was originally trained on the ImageNet dataset, minor adjustments are made to integrate it with our model. Notably, the VGG-16 model expects images of size 224x224x3 as input. We exclude the last classification layer to obtain a 2048-feature vector for each image. The weights for each image are downloaded, and their names are associated with their respective feature arrays. This process may require some time depending on the processing power available.

C. Layering the CNN-RNN model:

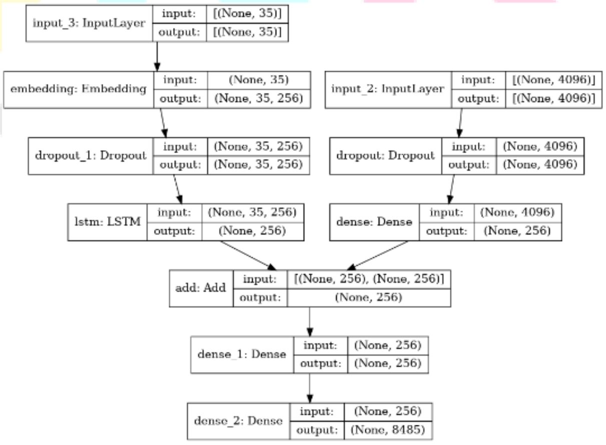


Fig 9 model layering

1. **Feature Extraction Module**: This component undertakes dimensionality reduction on input images, transforming them from their initial 2048-dimensional representation to a more compact 256-dimensional representation. To achieve this, a pre-trained VGG16 model, devoid of its output layer, is utilized in conjunction with a subsequent convolutional neural network (CNN) and a long short-term memory (LSTM) layer. A dropout regularization technique is further implemented to enhance the model's generalization capabilities.

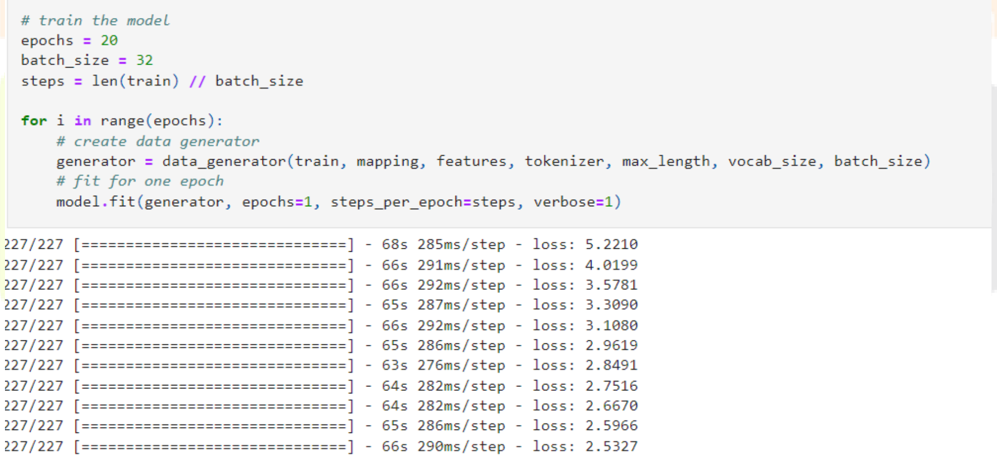
2. **Sequence Processing Module**: This module is specifically designed to handle textual input data. It commences with an embedding layer, responsible for mapping individual words into appropriate vector representations. This is followed by an LSTM layer, capable of capturing sequential dependencies within the textual data.

3. **Decoder Module**: The outputs generated by the Feature Extraction and Sequence Processing modules, both represented as fixed-length vectors, are subsequently merged and fed into a Dense layer. This final layer performs the task of generating the model's final predictions. Importantly, the number of neurons within the Dense layer is directly determined by the size of the target vocabulary.

D Training the model:

The model is trained using a dataset of 6000 images, each represented by a 2048-dimensional feature vector. Due to memory limitations, a data generator is employed to create batches of data during training, optimizing memory usage and training speed. The number of training iterations, known as epochs, is carefully chosen to prevent overfitting and underfitting. The fit_generator() function is used for this training process, which can take significant time depending on the available processing power. A sequence creator module is utilized to predict the next word based on the previous word and the corresponding image feature vector. A development dataset of 1000 images is used to monitor the model's performance during training and determine when to save model versions. Multiple model versions are saved, allowing for flexibility in selecting the best performing model for testing.

The final step involves thoroughly testing the trained model to evaluate its effectiveness and accuracy.



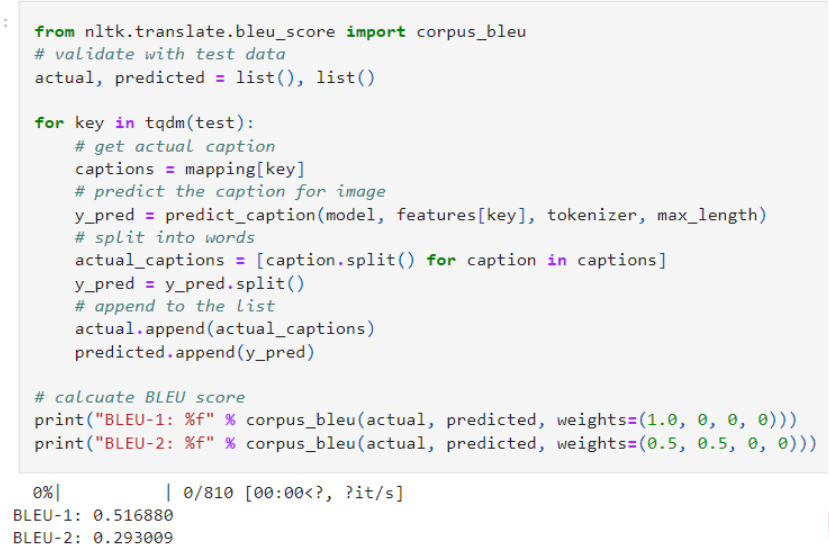
```
# train the model
epochs = 20
batch_size = 32
steps = len(train) // batch_size

for i in range(epochs):
    # create data generator
    generator = data_generator(train, mapping, features, tokenizer, max_length, vocab_size, batch_size)
    # fit for one epoch
    model.fit(generator, epochs=1, steps_per_epoch=steps, verbose=1)
```

```
227/227 [==============================] - 68s 285ms/step - loss: 5.2210
227/227 [==============================] - 66s 291ms/step - loss: 4.0199
227/227 [==============================] - 66s 292ms/step - loss: 3.5781
227/227 [==============================] - 65s 287ms/step - loss: 3.3090
227/227 [==============================] - 66s 292ms/step - loss: 3.1080
227/227 [==============================] - 65s 286ms/step - loss: 2.9619
227/227 [==============================] - 63s 276ms/step - loss: 2.8491
227/227 [==============================] - 64s 282ms/step - loss: 2.7516
227/227 [==============================] - 64s 282ms/step - loss: 2.6670
227/227 [==============================] - 65s 286ms/step - loss: 2.5966
227/227 [==============================] - 66s 290ms/step - loss: 2.5327
```

Fig 10 training the model

E Testing the model:

The model saved in the previous step will now be loaded to generate the final predictions. A sequence generator is employed along with the tokenizer file. To generate captions, the initial step involves extracting features for the given image. You can provide the path of the image from the testing dataset as input to the function. Additionally, you have the option to iterate through the entire test dataset and store predictions for each image in a list or dictionary. The proper functioning of caption generation entails using the start sequence and the end sequence, and iteratively calling the model to generate meaningful captions for the given input image.



```
from nltk.translate.bleu_score import corpus_bleu
# validate with test data
actual, predicted = list(), list()

for key in tqdm(test):
    # get actual caption
    captions = mapping[key]
    # predict the caption for image
    y_pred = predict_caption(model, features[key], tokenizer, max_length)
    # split into words
    actual_captions = [caption.split() for caption in captions]
    y_pred = y_pred.split()
    # append to the list
    actual.append(actual_captions)
    predicted.append(y_pred)

# calcuate BLEU score
print("BLEU-1: %f" % corpus_bleu(actual, predicted, weights=(1.0, 0, 0, 0)))
print("BLEU-2: %f" % corpus_bleu(actual, predicted, weights=(0.5, 0.5, 0, 0)))
```

```
  0%|          | 0/810 [00:00<?, ?it/s]
BLEU-1: 0.516880
BLEU-2: 0.293009
```

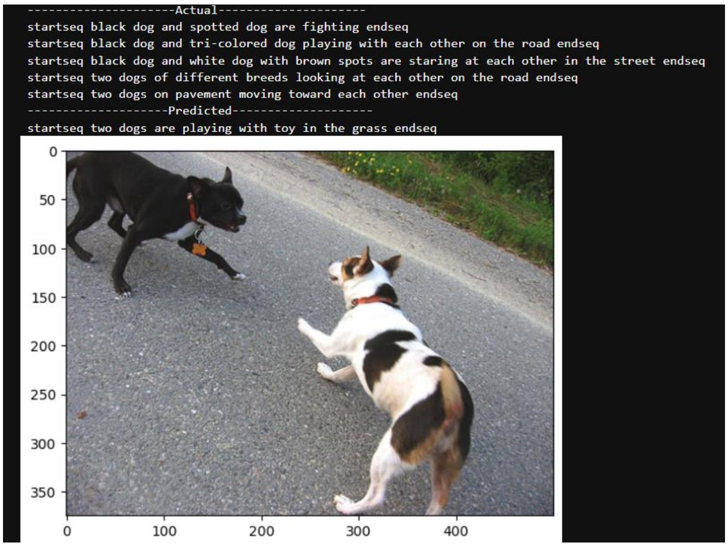
Fig 11 Testing the model

7 RESULTS :

A. Datasets: These datasets comprise images along with natural language descriptions, primarily in English. The statistics for these datasets are summarized in Table I. Each image within these datasets is annotated by observers using five distinct sentences, which are both visually descriptive and impartial.

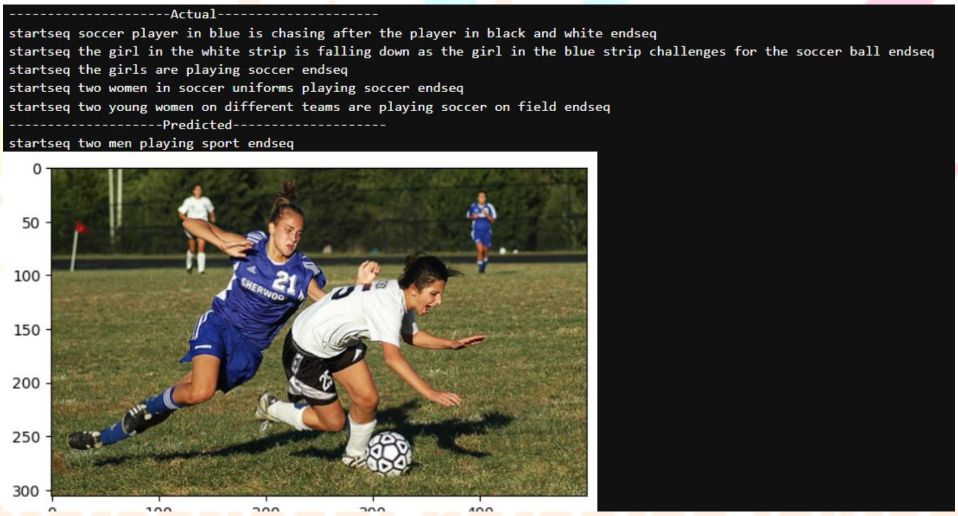
B. Results: The model has undergone training for a total of 50 epochs. A higher number of epochs contributes to reducing the loss, which in this case has reached 3.74. When dealing with large datasets, it is advisable to employ additional epochs to achieve more accurate results. Figure 4 illustrates the selection of evaluation results. Specifically:

- Using the Flickr8k dataset for model training and conducting tests on the 1000 test images available in the dataset yields a BLEU score of 0.53356.
- For the Flickr30k dataset, testing on an equivalent number of test images from the dataset results in a BLEU score of 0.61433.
- Finally, when running tests on images from the MSCOCO dataset, the BLEU score obtained is 0.67257.



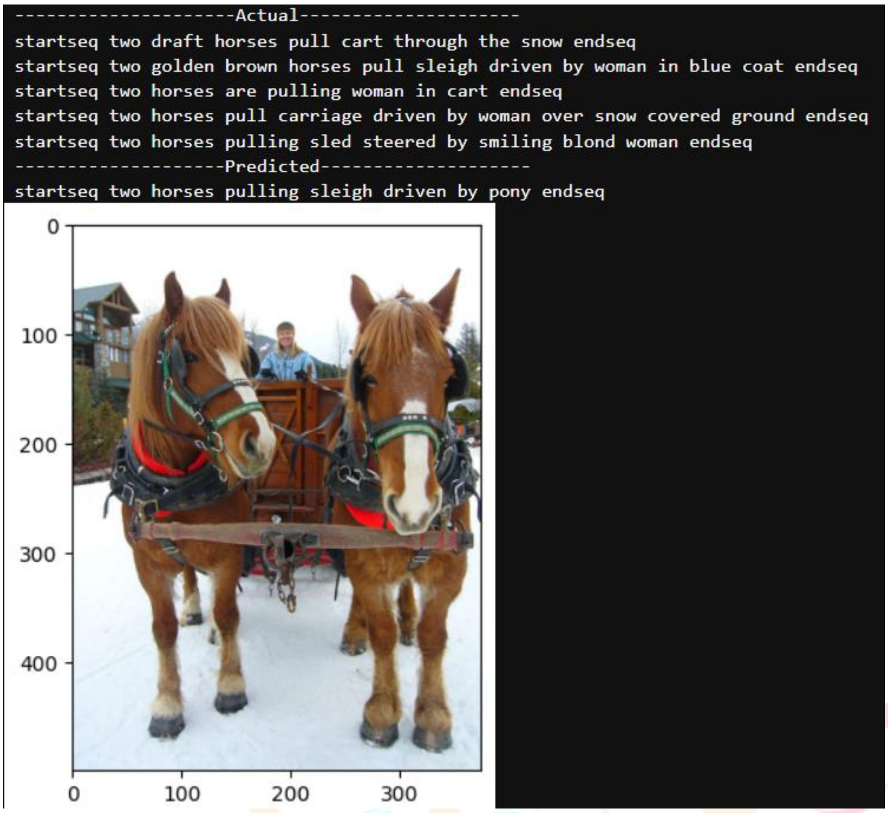
```
---------------------Actual---------------------
startseq black dog and spotted dog are fighting endseq
startseq black dog and tri-colored dog playing with each other on the road endseq
startseq black dog and white dog with brown spots are staring at each other in the street endseq
startseq two dogs of different breeds looking at each other on the road endseq
startseq two dogs on pavement moving toward each other endseq
-------------------Predicted-------------------
startseq two dogs are playing with toy in the grass endseq
```

Fig.12 Result output 1



```
---------------------Actual---------------------
startseq soccer player in blue is chasing after the player in black and white endseq
startseq the girl in the white strip is falling down as the girl in the blue strip challenges for the soccer ball endseq
startseq the girls are playing soccer endseq
startseq two women in soccer uniforms playing soccer endseq
startseq two young women on different teams are playing soccer on field endseq
-------------------Predicted-------------------
startseq two men playing sport endseq
```

Fig.13 result output 2



```
--------------------Actual--------------------
startseq two draft horses pull cart through the snow endseq
startseq two golden brown horses pull sleigh driven by woman in blue coat endseq
startseq two horses are pulling woman in cart endseq
startseq two horses pull carriage driven by woman over snow covered ground endseq
startseq two horses pulling sled steered by smiling blond woman endseq
-------------------Predicted-------------------
startseq two horses pulling sleigh driven by pony endseq
```

Fig 14 result output 3

8 Conclusion

The results of this study demonstrate the effectiveness of the deep learning approach used to generate captions for images. The combination of the Convolutional Neural Network (CNN) and the Long Short-Term Memory (LSTM) network worked well together, allowing the recognition of the relationships among the objects in the images. To evaluate the quality of the generated captions, they were compared with the target captions from the Flickr8k test dataset using the BLEU (Bilingual Evaluation Understudy) score. BLEU scores are widely used in text translation tasks, and provide a reliable measure of how well the translated text matches the reference translations. Additionally, different neural network architectures were tested as alternatives to the ones proposed here, such as using the VGG16 model instead of the Xception model, or using the GRU model instead of the LSTM model. The BLEU score helped to compare these models and to determine the best performing approach.

This paper has presented an overview of the various developments in the field of machine learning and artificial intelligence (AI). It also highlights the broad scope of this field, with many opportunities for further research and development. While this paper covers the essential aspects of building an image caption generator, it also recognizes the possibility of further improvement and innovation in this area. Some of the future work directions include:

- Incorporating attention mechanisms to focus on the most relevant parts of the image for caption generation.
- Using more advanced language models, such as transformers or BERT, to generate more natural and diverse captions.
- Exploring other datasets and domains, such as medical images or videos, to extend the applicability and generalization of the image caption generator.

9 References:

1. Al-Muzaini HA, Al-Yahya TN, Benhidour H (2018) Automatic Arabic image captioning using RNN-LST M-based language model and CNN Int J Adv Comput Sci Appl 9(6):67–73

2. Amritkar C, Jabade V (2018) Image caption generation using deep learning technique In 2018 fourth international conference on computing communication control and automation (ICCUBEA) IEEE, Pune, pp 1–4

3. Bai S, An S (2018) A survey on automatic image caption generation Neurocomputing 311:291–304

4. Bigham JP, Lin I, Savage S (2017) The effects of not knowing what You Don't know on web accessibility for blind web users In proceedings of the 19th international ACM SIGACCESS conference on computers and accessibility, 101-109

5. Deng Z, Jiang Z, Lan R, Huang W, Luo X (2020) Image captioning using dense net network and adaptive attention Signal Process Image Commun 85:1–9

6. Geng, W, Han F, Lin J, Zhu L, Bai J, Wang S, He L, Xiao Q, Lai Z (2018) Fine-grained grocery product recognition by one-shot learning In Proceedings of the 26th ACM international conference on Multimedia, pp 1706–1714

7. Giraud S, Thérouanne P, Steiner DD (2018) Web accessibility: filtering redundant and irrelevant information improves website usability for blind users International Journal of Human-Computer Studies 111:23–35

8. Guinness D, Cutrell E, Morris MR (2018) Caption crawler: enabling reusable alternative text descriptions using reverse image search In proceedings of the 2018 CHI conference on human factors in computing systems, Montréal, QC, Canada, pp 1–11

9. Hossain MDZ, Sohel F, Shiratuddin MF, Laga H (2019) A comprehensive survey of deep learning for image captioning ACM Computing Surveys (CsUR) 51(6):1–36

10. Iwamura K, Kasahara JYL, Moro A, Yamashita A, Asama H (2021) Image captioning using motion-CNN with object detection Sensors 21(4):1–13

11. Jund P, Abdo N, Eitel A, Burgard W (2016) The freiburg groceries dataset arXiv preprint arXiv:1611 05799 nnn nn