PROFESSIONAL TRAINING REPORT

entitled

IMAGE CAPTION GENERATOR USING DEEP LEARNING

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Artificial Intelligence and Machine Learning

by

R.SRICHARAN

[41611193]

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A++" by NAAC JEPPIAAR NAGAR, RAJIV GANDHISALAI,CHENNAI – 600119

OCTOBER 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with A++ Grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai – 600 119
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training is the bonafide work of **Mr. R.Sricharan(41611193)** who carried out the project entitled **"Image caption generator using deep learning"** under my supervision from June 2023 to October 2023.

Internal Guide
Mrs.Parveen.A,Assistant Professor,CSE

Head of the Department
Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner **External Examiner**

DECLARATION

I, **R.SRICHARAN(41611193),** hereby declare that the Professional Training Report-I entitled "**Image Caption Generator using Deep learning"** done by me under the guidance of **Mrs.Parveen.A,Assistant Professor,CSE** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Artificial Intelligence.

DATE: 4th October 2023

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D.**, **Dean**, School of Computing, **Dr. S.Vigneshwari M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Internal Guide **Mrs.Parveen.A,Assistant Professor,CSE** for his/her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my phase-1 professional Training.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Automatically creating the description or caption of an image using any natural language sentences is a very challenging task. It requires both methods from computer vision to understand the content of the image and a language model from the field of natural language processing to turn the understanding of the image into words in the right order. In addition to that we have discussed how this model can be implemented on web and will be accessible for end user as well. Our project aims to implement an Image caption generator that responds to the user to get the captions for a provided image. The ultimate purpose of Image caption generator is to make users experience better by generating automated captions. We can use this in image indexing, for visually impaired persons, for social media, and several other natural language processing applications. Deep learning methods have demonstrated state- of- the-art results on caption generation problems. What is most impressive about these methods is a single end-to- end model can be defined to predict a caption, given a photo, instead of requiring sophisticated data preparation or a pipeline of specifically designed models. In this an **Image caption generator**, Basis on our provided image ,It will generate the caption from our trained model. The basic idea behind this is that users will get automated captions when we use or implement it on social media or on any applications.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Project Description: The Image Caption Generator project is an exciting application of computer vision and natural language processing that aims to automatically generate descriptive captions for images. This project combines the power of Convolutional Neural Networks (CNNs) for image analysis and Recurrent Neural Networks (RNNs) for natural language generation to create a system capable of understanding and describing the content of images.

Project Goals:

Automatic Image Description: The primary goal of this project is to develop a system that can take an image as input and generate human-readable captions that describe the content ofthe image accurately.

Cross-Modal Understanding: The project aims to bridge the gap between visual and textual data by teaching the model to understand the relationship between what it sees in an image and how it can be described in words.

User-Friendly Interface: Create a user-friendly interface (web or mobile application) for users to upload images and receive automatic image captions.

Scalability: Ensure that the system can handle a variety of image types and is scalable for processing a large number of images efficiently.

Customization: Allow users to fine-tune the model for specific domains or adjust the level of detail in generated captions.

Key Components:

Image Preprocessing: Input images need to be preprocessed to be compatible with the model. This may involve resizing, normalization, and data augmentation.

Convolutional Neural Network (CNN): A pre-trained CNN, such as VGG16 or Inception, isused to extract features from the images. These features are crucial for understanding the content of the image.

Recurrent Neural Network (RNN): A recurrent neural network, typically an LSTM or GRU,is used for generating captions. It takes both the image features and previously generated words into account to produce a coherent and descriptive caption.

Natural Language Processing: The NLP component is responsible for tokenizing words,building a vocabulary, and generating captions using the RNN.

Model Training and Fine-Tuning: Training the model involves using a large dataset of images and their corresponding captions. Fine-tuning may be necessary to adapt the model to specific domains or to improve caption quality.

Challenges:

Data Collection: Gathering a diverse and extensive dataset of images with accurate captions can be challenging.

Model Complexity: Developing and training a deep learning model that integrates both CNN and RNN components can be computationally expensive.

Overfitting: Ensuring that the model generalizes well to unseen data and doesn't overfit the training set.

Evaluation Metrics: Defining metrics to evaluate the quality of generated captionsobjectively.

Applications:

Accessibility: The system can be used to assist individuals with visual impairments byproviding textual descriptions of images.

Content Tagging: E-commerce websites and social media platforms can automatically tagimages with relevant descriptions.

Education: Enhancing educational materials by providing descriptive captions for images and diagrams.

Content Management: Automatically generating metadata for large image databases.

Future Improvements:

Multimodal Models: Consider using more advanced models that can handle both images and text simultaneously, such as the Vision-Language Pre-training (VLP) models

Better Fine-Tuning: Improve the fine-tuning process to make it easier for users to customize the model for specific applications.

Real-Time Processing: Optimize the system for real-time image captioning to support applications like live video captioning.

Multilingual Support: Extend the system to generate captions in multiple languages.

CHAPTER 2

LITERATURE REVIEW

2.1 SURVEY

Caption generation is a challenging artificial intelligence problem where a textual description must be generated for a given photograph. It requires both methods from computer vision to understand the content of the image and a language model from the field of natural language processing to turn the understanding of the image into words in the right order. Recently, deeplearning methods have achieved state-of-theart results on examples of this problem. Deep learning methods have demonstrated state-of-the-art results on caption generation problems. What is most impressive about these methods is a single end-to-end model can be defined to predict a caption, given a photo, instead of requiring sophisticated data preparation or a pipeline of specifically designed models. RNN's have become very powerful. Especially for sequential data modelling. Andrej Karapathy has very nicely explained the use of RNN's in his blog The Unreasonable Effectiveness of Recurrent Neural Networks. There are basically4 types of RNN.

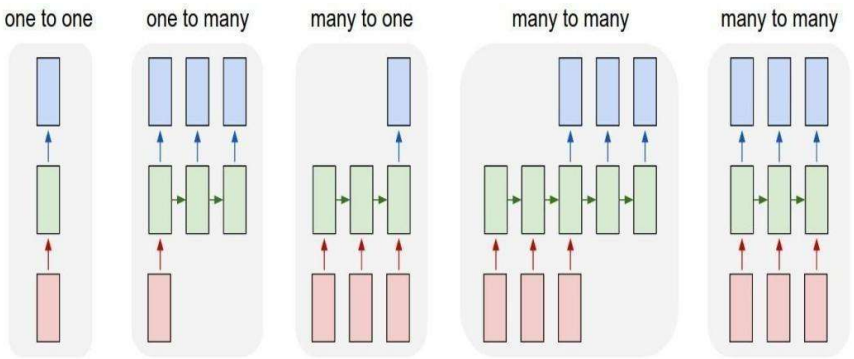


Figure 2.1.1-Four types of RNN

A. Convolutional Neural Network(CNN) Convolutional Neural Network (CNN) is a Deep Learning algorithm which takes in an input image and assigns importance (learnable weights and biases) to various aspects/objects in the image, which helps it differentiate pooling layer .when the image

is passed through one convolution layer, the output of the first layer becomes the input for the second layer. This process continues for all subsequent layers. After a series of convolutional, nonlinear and classification. The neural network consists of several convolutional layers mixed with nonlinear pooling layers, it is necessary to attach a fully connected layer. This layer takes the output information from convolutional networks. Attaching a fully connected layer to the end of the network results in an N dimensional vector, where N is the number of classes from which the model selects the desired class.

B. Long short-term Memory: LSTM networks are a specialized type of recurrent neural network (RNN) designed to capture long-range dependencies in sequential data, making them useful for tasks like machine translation and speech recognition. They were developed to address the issue of vanishing gradients that hindered traditional RNNs when dealing with deep sequences. LSTMs excel at processing time series data by maintaining relevant information and discarding irrelevant information using a "forget gate." This makes them more effective in overcoming the short-term memory limitations of traditional RNNs.

C. CNN LSTM: Architecture The CNN-LSTM architecture involves using CNN layers for feature extraction on input data combined with LSTMs to support sequence prediction. This model is specifically designed for sequence prediction problems with spatial inputs, like images or videos. They are widely used in Activity Recognition, Image description,video description.

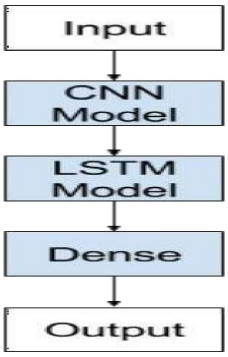


Figure 2.1.2 -Architecture of CNN-LSTM

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 OBJECTIVE OF THE PROJECT

Objective for Image Caption Generator:

The primary objective of developing an Image Caption Generator is to create a robust and versatile AI system capable of generating descriptive and contextually relevant captions for images. This project serves various purposes and can be tailored to specific applications, but the core objective remains constant:

Automatic Image Understanding and Description: Develop a model that can analyze the visual content of images and generate relevant, informative, and human-like captions that accurately describe the visual elements within the images.

User-Friendly Interface: Create an accessible and user interface (web, mobile app, or API)to allow users to easily upload images and receive automatic image captions, making the technology accessible to a broader audience.

Scalability: Ensure the system is designed to handle a wide range of image types, sizes, and complexities efficiently, making it suitable for both small-scale and large-scale image captioning tasks.

Customization: Allow users to fine-tune the model for specific domains, applications, or preferences, enabling them to adapt the system to their unique requirements.

Potential Outcomes:

Improved Accessibility: Enable individuals with visual impairments to access and understand visual content on the internet and in their daily lives.

Enhanced Content Management: Streamline content organization and search by providing automated textual descriptions for images, improving content discoverability and SEO.

Educational Enhancement: Improve educational materials, textbooks, and online courses by automatically generating informative image captions for diagrams and illustrations.

Content Tagging and Social Media: Automatically tag and caption images on e-commerce websites and social media platforms, enhancing user experiences and accessibility.

Multilingual Support: Extend the system to generate captions in multiple languages to serve a global audience.

3.2 REQUIREMENTS

3.2.1 HARDWARE REQUIREMENTS

1. Requires a 64-bit processor and operating system

2.Processor: Intel Core i5-10600 / AMD Ryzen 5 3600

3.Memory: 8 GB RAM

4.Graphics: NVIDIA RTX 1080 or above/ AMD Radeon RX 57005.Network:

Broadband Internet connection

6.Storage: must have 5GB available space

3.2.2 SOFTWARE REQUIREMENTS

1. Python as the primary programming language.

2. Deep learning framework (e.g., TensorFlow, PyTorch, Keras).

3. GPU drivers (if using GPU) and CUDA Toolkit (for NVIDIA GPUs).

4. Image processing libraries (e.g., OpenCV, PIL).

5.NLP libraries (e.g., NLTK, spaCy).

5. Text editor:

vscode,jupyter notebook.

Data:

Access to a dataset of images with corresponding captions.

Common datasets include MS COCO, Flickr8K, and Flickr30K.

Model and Architecture:

Choose a suitable deep learning model and architecture (e.g., CNN-RNN, Transformer).

Evaluation Metrics: Understand evaluation metrics like BLEU, METEOR, and CIDEr forcaption quality assessment.

Knowledge:

Deep learning expertise in computer vision and NLP.Proficiency

in Python programming.

Ethical Considerations:

Awareness of ethical concerns, such as bias in caption generation.

Testing and Debugging:

Rigorous testing and debugging of the model for improved performance.

Documentation and Best Practices:

Stay updated with research papers and best practices in image captioning.

User Interface :

Consider UX/UI design if building a user-facing application.

CHAPTER 4

DESIGN DESCRIPTION OF PROPOSED PROJECT

4.1 PROPOSED METHODOLOGY

Creating an image caption generator involves several key steps, including data collection, model architecture, training, and deployment. Here's a detailed methodology for the project:

1. Data Collection:

- Gather a diverse dataset of images with corresponding descriptive captions. Popular MS COCO dataset, Flickr30k, or custom datasets.
- Ensure captions are accurate, relevant, and ideally written by humans to ensure high-qualitytraining data.

2. Data Preprocessing:

- Resize and normalize images to a consistent format.
- Tokenize captions into words or subwords, creating a vocabulary.
- Pad or truncate captions to a fixed length for uniform input to the model.

3. Model Architecture:

- Utilize a pre-trained Convolutional Neural Network (CNN) like VGG16, ResNet, or Inception to extract image features. This CNN acts as the encoder.
- Employ a Recurrent Neural Network (RNN) with LSTM or GRU cells as the decode, the decoder generates captions word by word.

4. Data Split:

- Divide the dataset into training, validation, and testing sets. A common split might be 80%training, 10% validation, and 10% testing.

5. Model Training:

- Train the image encoder (CNN) and caption decoder (RNN) separately, using the training set.
- Use teacher forcing during training, where the model is fed the ground truth words from the training captions to predict the next word.
- Optimize using gradient descent-based algorithms like Adam or RMSprop.

6. Hyperparameter Tuning:

- Experiment with hyper parameters such as learning rate, batch size, and Model architecture to optimize model performance.

7. Evaluation:

- Evaluate the trained model on the validation and testing sets using caption quality metrics(e.g., BLEU, METEOR, CIDE.

8. Fine-Tuning :

- Fine-tune the model for specific domains or applications, adjusting the vocabulary or training data as needed.

9. User Interface Development:

- Create a user-friendly interface (web or mobile app) for users to upload images and receive automatic captions.

10. Model Deployment:

- Deploy the trained model along with the interface, ensuring it's accessible to users.

11. User Feedback Loop:

- Collect user feedback to improve the model and interface iteratively.

12. Scalability and Performance Optimization:

- Ensure the system can handle real-time image captioning for large volumes of images efficiently.

13. Multilingual Support:

- Extend the system to generate captions in multiple languages.

14. Documentation and Maintenance:

- Document the project, including model architecture, data sources, and usage instructions.Regularly update and maintain the system as needed.

4.1.1 Ideation Map/System Architecture

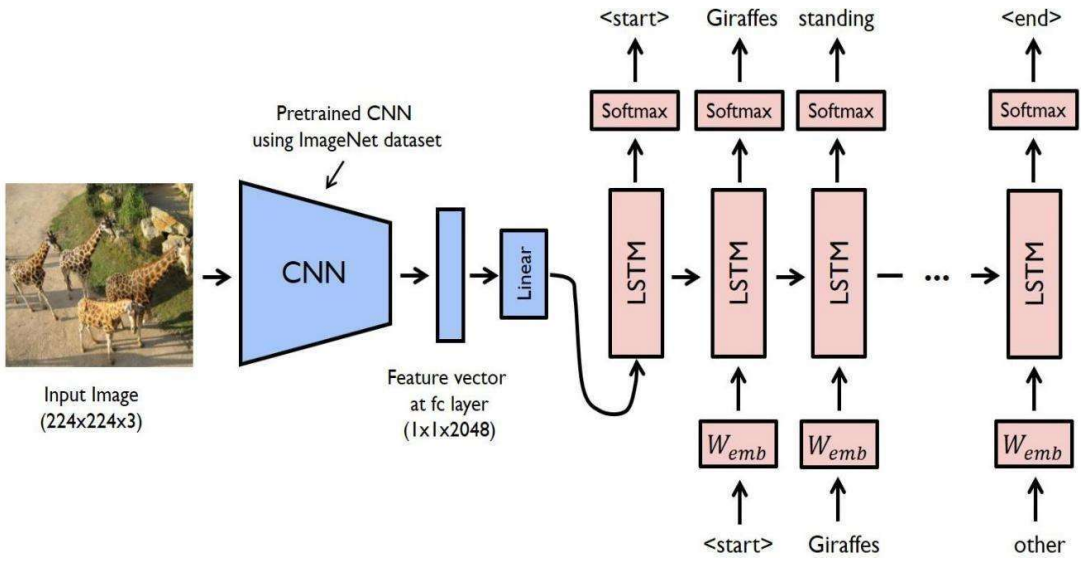


Figure 4.1.1.1 Architecture of image caption generator

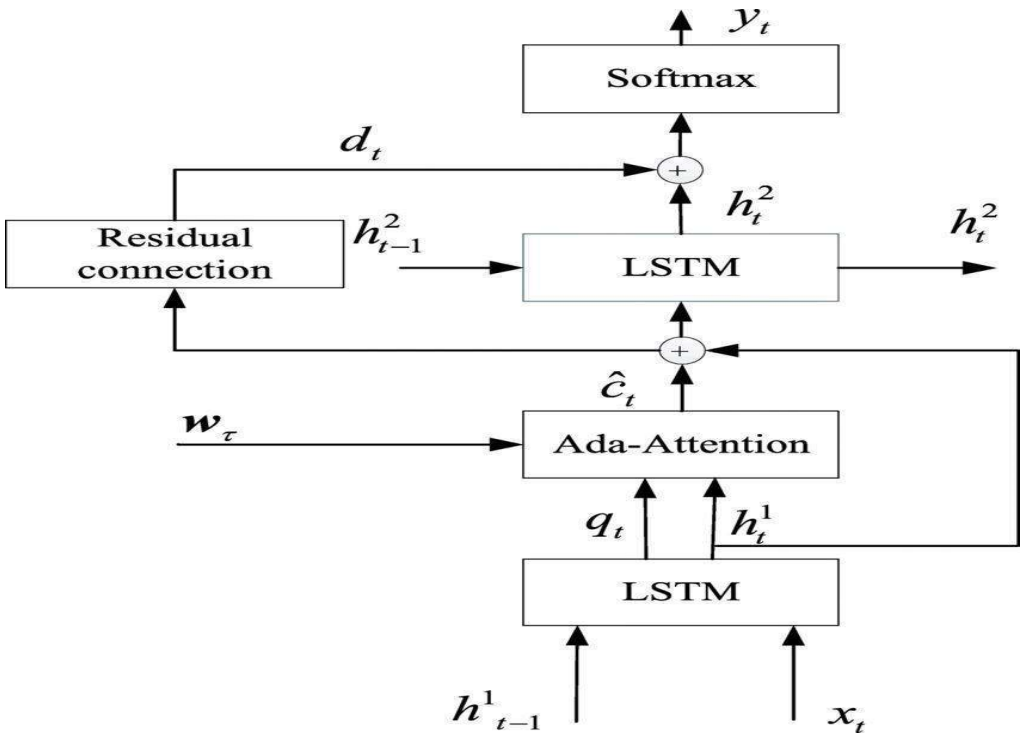


Figure 4.1.1.2- Flow chart of image caption generator

4.1.2 Various Stages

The model is trained to maximise the probability of p(S|I), where S is the sequence of words produced by the model and each word St. is produced using a lexicon that was created using the training dataset. A deep vision convolutional neural network (CNN) is fed the input image I, which aids in object detection in the image. Recurrent Neural Network (RNN) receives the picture encodings and uses them to create a relevant phrase for the image. The model can be compared to a language translation RNN model, where the goal is to maximise the p(T|S), where T is the translation to the sentence S. However, in our model the encoder RNN which helps in transforming an input sentence to a fixed length vector is replaced by a CNN encoder. Recent research has shown that the CNN can easily transform an input image to a vector.

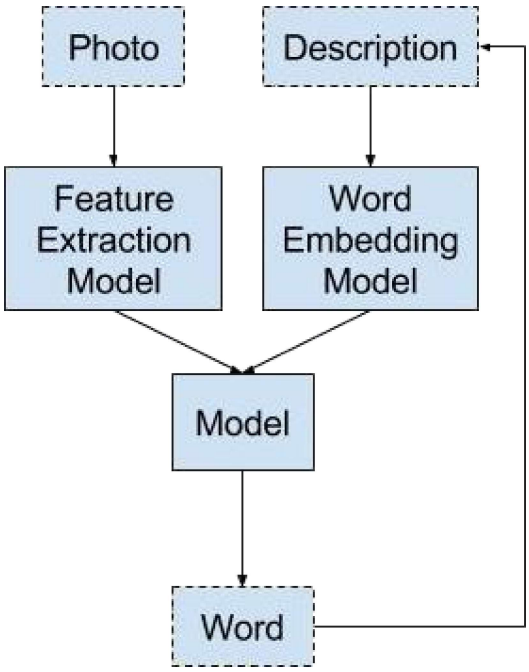


Figure 4.1.2.1- Stages of image caption generator

Recurrent Neural Network:

A feed forward neural network with an internal memory is known as a recurrent neural network. The result of the current input depends on the previous computation, making RNNs recurrent in nature because they carry out the same function for every data input. The output is created, copied, and then delivered back into the recurrent network. It takes into account both the current input and the output that it has learned from the prior input when making a decision. RNN One type of artificial neural network called a recurrent neural network (RNN) has connections between nodes that create a directed graph along a temporal sequence. It can display temporal dynamic behaviour as a result of this. RNNs, which are derived from feed-forward neural networks, can process input sequences of varying length by using their internal memory state.

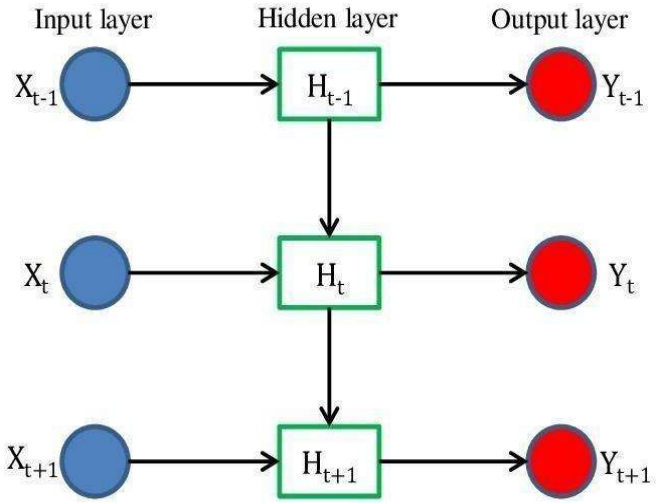


Figure 4.1.2.2 - Architecture of RNN

Convolutional Neural Network:

Convolutional neural networks (CNN, or ConvNet) are a class of deep neural networks used most frequently to analyse visual vision in deep learning. Based on the shared-weight convolution kernels' translation invariance properties and shift invariance architecture, they are often referred to as shift invariant or space invariant artificial neural networks (SIANN). They can be used in a variety of fields, including image and video recognition, recommender systems, image classification, image segmentation, and medical image analysis. They can also be used in brain-computer interfaces, natural language processing, and financial time series.

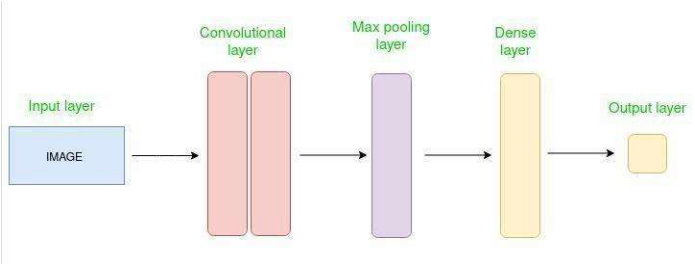


Figure 4.1.2.3- CNN architecture

Multilayer perceptrons are modified into CNNs. Fully linked networks, or multilayer perceptrons, are those in which all of the neurons in one layer are connected to all of the neurons in the following layer. These networks are vulnerable to overfitting data because of their "fully connectedness." Changing the weights when the loss function is decreased and randomly trimming connections are common methods of regularisation.

Long Short Term Memory:

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition. LSTM Inner ModelA challenging area of deep learning is LSTMs. Understanding LSTMs and how concepts like bidirectional and sequence-to-sequence relate to the field can be challenging. LSTM has feedback connections in contrast to traditional feed forward neural networks. It can analyse whole data sequences in addition to single data points (like photos) (such as speech or video).

For instance, LSTM can be used for tasks like linked, unsegmented handwriting identification, speech recognition, and network traffic anomaly detection, or IDSs (intrusion detection systems). A cell, an input gate, an output gate, and a forget gate make up a typical LSTM unit. The three gates control the information flow, and the cell retains values across arbitrary time intervals.

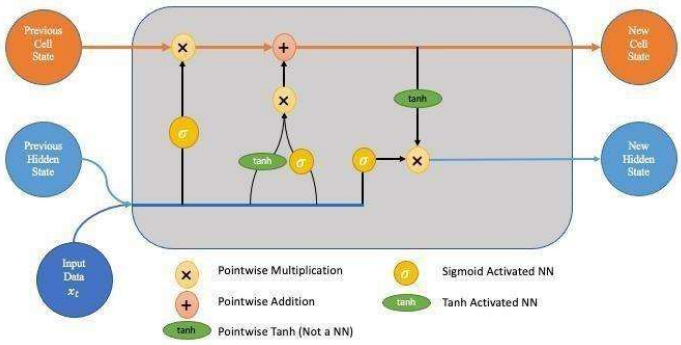


Figure 4.1.2.4 LSTM structure

4.1.3 Internal or Component design structure

Data Collection and Priorities:

- **Image Dataset**: Collect images with matching titles. Common choices include MS COCO,Flickr8K, and Flickr30K.
- **Text preprocessing**: Clean and preprocess header text including tokenization, shorthand, and symbol and special character removal.

Feature excluded:

Image Feature Extraction: Use a pre-trained Convolutional Neural Network (CNN) such as InceptionV3 or ResNet to extract high-quality image features. These features effectively represent visual information.

- **Text embeddings**: Convert pre-processed captions to word embeddings using methods like Word2Vec or use pre-trained embeddings like GloVe to represent textual data.

- **Encoder-decoder architecture**: Use an encoder-decoder architecture, where the encoder processes the image content and the decoder generates text.

Training:

- **Loss function**: Use an appropriate loss function, typically cross-entropy loss, to measure the difference between predicted and actual subjects.
- **Optimization**: Use optimization algorithms such as ADAM or RMS prop to update modelweights during training.
- **Validation and evaluation**: Check model performance in the validation set and use evaluation metrics such as BLEU, METEOR, and CIDEr to assess caption quality.

Decoding the presentation:

- **Greedy decoding**: Implement a greedy decoding technique for captions by selecting the most likely word at each time step.

4.1.4 Working principles

Image caption generation is a fascinating field in artificial intelligence that combines computer vision and natural language processing (NLP) to generate descriptive textual captions for images. The working principles for image caption generation can be broken down into several key steps:

1. Image Input:

- The process begins with an image as the input. The image can be of various types, such as photographs, illustrations, or screenshots.

2. Feature Extraction:

- The image is passed through a pre-trained Convolutional Neural Network (CNN), such as VGG16, Inception, or ResNet. The CNN extracts high-level features from the image. These features are often flattened and become a fixed-length vector.

3. Contextual Information:

- The extracted image features provide a condensed representation of the visualinformation. This representation is passed to the caption generator to provide contextual information about the image.

4. Sequence Generation:

- The caption generation process involves Recurrent Neural Networks (RNNs) or their variants, such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU). The RNN is initialized with a special "start" token, indicating the beginning of the caption.

5. Word Prediction Loop:

- The RNN processes the initial image context and generates one word at a time. It uses the previously generated words as context for predicting the next word in the sequence.
- The predicted word is selected based on the RNN's internal state and the currentimage context.
- The process continues iteratively until a "stop" token or a maximum caption length is reached.

6. Vocabulary and Language Model:

- The system relies on a predefined vocabulary that contains a list of possible words or subwords (n-grams). This vocabulary helps constrain the model's output to a set of valid words.

7. Attention Mechanism :

- Some advanced caption generation models incorporate an attention mechanism. This mechanism allows the model to dynamically focus on different parts of the image when generating each word in the caption, leading to more accurate and contextually relevant descriptions.

8. Caption Evaluation:

- The generated caption is evaluated using various metrics such as BLEU (Bilingual Evaluation Understudy), METEOR (Metric for Evaluation of Translation with Explicit ORdering), CIDEr (Consensus-based Image Description Evaluation), or ROUGE (Recall-Oriented Understudy for Gisting Evaluation).
- The evaluation metrics assess the quality and relevance of the generated caption compared to human-authored references.

9. Fine-Tuning :

- The system may be fine-tuned to adapt to specific domains or improve caption quality. Fine-tuning can involve training the model with additional data or custom datasets.

10. User Interface :

- image caption generator is designed for user friendly interaction.

Multilingual Support :

- To support multiple languages, the system can be extended to generate captions invarious languages.

4.2 Features

The features of an image caption generation system are the key components,capabilities,and characteristics that enable it to analyze images and produce descriptive textual captions. Here are the essential features of an image caption generation system.

Image Analysis:

Feature Extraction:

- The system employs a pre-trained Convolutional Neural Network (CNN) to extract meaningful and high-level features from input images. This process involves identifying objects, shapes, and patterns within the images.

Natural Language Processing (NLP):

Caption Generation:

- Utilizing Recurrent Neural Networks (RNNs), typically LSTM or GRU cells, the system generates captions word by word. It maintains the context of previously generated wordsto produce coherent sentences.

Vocabulary and Language Model:

Vocabulary:

- The system uses a predefined vocabulary that contains a list of valid words or subwords, ensuring that the generated captions are composed of linguistically correct terms.

Attention Mechanism

Attention:

- Some advanced models incorporate an attention mechanism that allows the system to focus on different regions of the image when generating each word, leading to more contextually relevant descriptions.

Fine-Tuning

Customization:

- The system can be fine-tuned to adapt to specific domains or improve captions.

Evaluation Metrics

Quality Assessment:

- The system is equipped with evaluation metrics like BLEU, METEOR, CIDEr, or ROUGE to objectively assess the quality and relevance of generated captions comparedto human-authored references.

Multilingual Support

Language Flexibility:

- The system can be extended to generate captions in multiple languages to cater to a global audience.

User Interface

Accessibility:

- To enhance user interaction, the system may include a user-friendly web or mobile interface for users to upload images and receive automatic captions.

Real-Time Processing

Live Captioning:

- The system may be optimized for real-time image captioning, which is useful for applications such as live video captioning.

Scalability

Efficiency:

- The system is designed to handle a large number of images efficiently, making it suitablefor both small-scale and large-scale applications.

Accessibility and Inclusion

Assistive Technology:

- The system can be used to provide textual descriptions of images for individuals withvisual impairments, enhancing their accessibility to visual content.

Content Management

Tagging and Organization:

- Automated image captioning helps with content organization by providing metadata for images, making content more discoverable and searchable.

Educational Enhancement

Enhanced Learning Materials:

- The system can be used to improve educational materials, textbooks, and online coursesby generating informative image captions for diagrams and illustrations.

Content Tagging and Social Media:Enhanced

User Experience:

- E-commerce websites and social media platforms can automatically tag and caption images to enhance user experiences and accessibility.

Content Moderation

Automated Moderation:

- The system can assist in content moderation by flagging or filtering inappropriate images based on their generated captions.

Future Expansion

Advanced Models:

- The system can be expanded with more advanced models, such as Vision-Language Pre-training (VLP) models, to further improve performance and contextual understanding.

4.2.1 Novelty of the proposal

The novelty of a proposal for an image caption generation project lies in the unique and innovative aspects that set it apart from existing systems and approaches. While image caption generation itself is not a new concept, a novel proposal could introduce distinct features, methodologies, or applications that contribute to its originality. Here are somepotential areas of novelty:

1. Advanced Model Architectures:

- Introducing a novel deep learning architecture or incorporating the latest advancements in computer vision and NLP, such as Vision-Language Pre-training (VLP)enhance the quality and contextuality of image captions.

2. Multimodal Capabilities:

- Combining multiple modalities like text, images, and audio to generate captions, making the system more versatile and adaptable to diverse types of data.

3. Domain-Specific Fine-Tuning:

- Offering a unique approach to fine-tuning the model for specific industries or use cases such as medical imaging, fashion, or art, to tailor the system for specialized applications.

4. Real-Time Image Captioning:

- Developing a system that can generate captions for live video streams in real-time enabling applications like live event captioning or live news reporting.

5. Multilingual Support and Translation:

- Extending the system to not only generate captions in multiple languages but also offer automatic translation of captions for global accessibility.

6. Accessibility:

- Innovating in the area of accessibility by incorporating features that assist individuals with disabilities beyond visual impairments, such as cognitive or hearing impairments.

7. Interactive User Interface:

- Creating an intuitive, interactive user interface that not only generates captions but also allows users to engage with images and captions in unique ways, such as interactive storytelling.

8. Content Moderation and Ethical AI:

- Incorporating advanced content moderation capabilities to ensure ethical and Responsible use of image captions, addressing issues like bias and misinformation.

9. Privacy and Security:

- Implementing novel privacy and security measures to protect sensitive image data, especially in applications where privacy is a significant concern.

10. AI-Generated Art and Creativity:

- Exploring artistic and creative applications of image captioning, such as generating poetic or abstract descriptions for images, or using image captions to inspire new formsof visual art.

11. Augmented Reality Integration:

- Integrating image captioning with augmented reality (AR) applications, where users can view images through AR devices and receive real-time captions.

12. Emotion and Sentiment Analysis:

- Incorporating sentiment analysis into the image captioning process to generate captions that reflect the emotions and sentiments conveyed by the images.

13. Collaborative Image Captioning:

- Creating a system that allows multiple users to collaboratively contribute to generating captions for the same image, enabling a crowdsourced approach to image description.

CHAPTER 5

CONCLUSION

We reviewed deep learning-based picture captioning techniques in this study. We've provided a taxonomy of picture captioning methods, illustrated general block diagrams of the main groups, and highlighted the advantages and disadvantages of each. We talked about several evaluation criteria and datasets, as well as their advantages and disadvantages. The results of the experiment are also briefly summarised. We briefly discussed possible possibilities for future research in this field. Although deep learningbased image captioning techniques have made significant strides in recent years, a reliable technique that can provide captions of a high calibre for almost all photos has not yet been developed. Automatic picture captioning will continue to be a hot research topic for some time to come with the introduction of novel deep learning network architectures utilized here is Flickr 8k which includes nearly 8000 images, and the corresponding captions are also stored in the text file. Although deep learning-based image captioning techniques have made significant strides in recent years, a reliable technique that can provide captions of a high calibre for almost all photos has not yet been developed. Automatic picture captioning will continue to be a popular study topic for some time to come with the introduction of novel deep learning network architectures. With more people using social media every day and the majority of them posting images, the potential for image captioning is very broad in the future. Therefore, they will benefit more from this project.

Future scope:

Due to the internet's and social media's exponential rise in image content, image captioning hasrecently become a significant issue. This article reviews the many image retrieval studies conducted in the past while highlighting the various methods and strategies employed. There isa huge potential for future research in this area because feature extraction and similarity computation in photos are difficult tasks. Picture RETRIEVAL USING IMAGE CAPTIONING Using features like colour, tags, the histogram, and other features, current image retrieval algorithms calculate similarity. Since these approaches are independent of the image's context, findings cannot be entirely correct. Therefore, a thorough study of picture retrieval using the image context, such as image captioning, will help to resolve this issue in the future.

REFERENCES

1. https://imagecaptioninge9fd5517f350https://ieeexplore.ieee.org/document/827124
2. https://machinelearningmastery.com/develop-a-deep-learningcaption-generation-model-in-python/
3. https://medium.com/@raman.shinde15/image-captioning-withflickr8k-dataset bleu- 4bcba0b52926
4. https://blog.clairvoyantsoft.com/image-caption-generator-535b8e9a66ac
5. https://www.matecconferences.org/articles/matecconf/abs/2018/91/matecconf_eitce2018_01052/matecconf_eitce2018_01052.html
6. https://www.analyticsvidhya.com/blog/2018/04/solving-an-imagecaptioning-task-using- deep-learning/
7. Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. J. Artif. Int. Res., 47(1):853–899, May 2013.
8. Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. IEEE Trans. Pattern Anal. Mach. Intell., 39(4):664–676, Apr. 2017.
9. Polina Kuznetsova, Vicente Ordonez, Alexander C. Berg, Tamara L. Berg, and Yejin Choi. Collective generation of natural image descriptions. pages 359–368, 2012.
10. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C. Lawrence ´ Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, Computer Vision – ECCV 2014, pages 740– 755, Cham, 2014`
11. Springer International Publishing. [7] S. Liu and W. Deng. Very deep convolutional neural network based image classification using small training sample size. pages 730–734, Nov 2015.
12. Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences.

13. Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. CoRR, abs/1411.4555, 2014.

14. Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. CoRR, abs/1502.03044, 2015.

15. Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. 2011. Learning photographic global tonal adjustment with a database of input/output image pairs. InComputer Vision and Pattern Recognition (CVPR), 2011

APPENDIX

CODE:

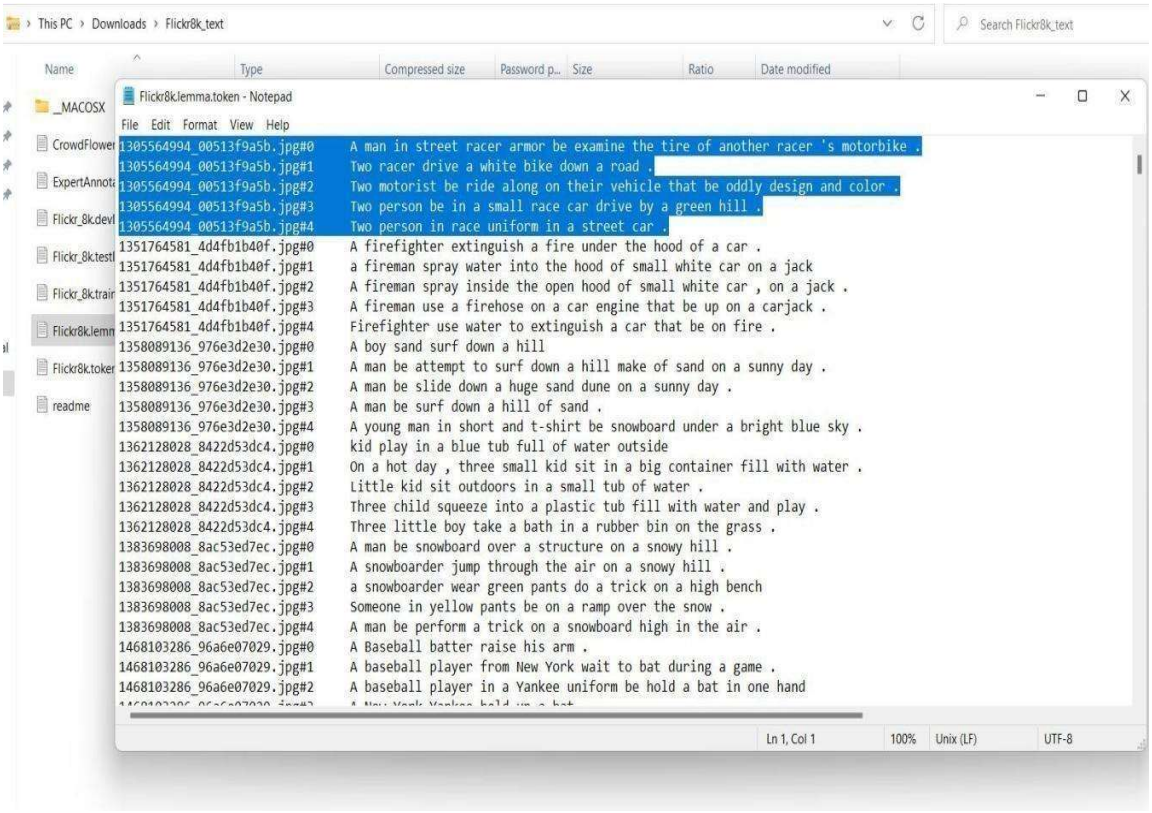
We have imported all the necessary libraries.And we have given the input images.

1.Image preprocessing:

```
import numpy as np
import pandas as pd
import cv2
import os
from glob import glob

images_path = "C:/Users/SRICHARAN/OneDrive/Documents/image caption generator/Images/"
images = glob(images_path+'*.jpg')
len(images)

images[:5]

import matplotlib.pyplot as plt

for i in range(5):
    plt.figure()
    img = cv2.imread(images[i])
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    plt.imshow(img)

from keras.applications import ResNet50

incept_model = ResNet50(include_top=True)

from keras.models import Model
last = incept_model.layers[-2].output
modele = Model(inputs = incept_model.input,outputs = last)
modele.summary()
```

2. Perform Data Cleaning

As we see all image captions are available in the Flickr 8k.token file of the Flickr_8k_text folder. If you analyze this file carefully, you can drive the format of image storing, each image and caption separated by a new line and carry 5 captions numbered from 0 to 4 along with.



3. Text preprocessing:

The format of our file is image and caption separated by a newline (“\n”) i.e, it consists of the name of the image followed by a space and the description of the image in CSV format. Here we need to map the image to its descriptions by storing them in a dictionary.we intially given 1000 images from our flickr dataset from kaggle it consist of 8000 images if u have better gpu u can load more images.

```
images_features = {}
count = 0
for i in images:
    img = cv2.imread(i)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img = cv2.resize(img, (224,224))

    img = img.reshape(1,224,224,3)
    pred = modele.predict(img).reshape(2048,)

    img_name = i.split('\\')[-1]

    images_features[img_name] = pred

    count += 1

    if count > 1000:
        break

    elif count % 100 == 0:
        print(count)
len(images_features)
print(images_features)

caption_path = r"C:/Users/SRICHARAN/OneDrive/Documents/image caption
generator/Flickr8k.token.txt"
captions = open(caption_path, 'rb').read().decode('utf-8').split('\n')
len(captions)

captions_dict = {}
for i in captions:
    try:
        img_name = i.split('\t')[0][:-2]
        caption = i.split('\t')[1]
        if img_name in images_features:
            if img_name not in captions_dict:
                captions_dict[img_name] = [caption]

            else:
                captions_dict[img_name].append(caption)
        else:
            print(f"Image {img_name} not found in images_features.")
```

4. Image visualization with captions:

Visualize the images using matplotlib with captions now it will try to visualize the images first and display the images and compare it with the text data given we have given 40000 sentences from this it will gather words.

```
import matplotlib.pyplot as plt

for i in range(5):
    plt.figure()
    img_name = images[i]


    img = cv2.imread(img_name)

    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    caption = captions_dict.get(img_name.split('\\')[-1])

    if caption:
        plt.xlabel(caption[0])  # Display the first caption
    else:
        plt.xlabel("Caption not found")
    plt.imshow(img)
    plt.show()

def preprocessed(txt):
    modified = txt.lower()
    modified = 'startofseq ' + modified + ' endofseq'
    return modified

for k,v in captions_dict.items():
    for vv in v:
        captions_dict[k][v.index(vv)] = preprocessed(vv)

count_words = {}
count = 1
for k,vv in captions_dict.items():
    for v in vv:
        for word in v.split():
            if word not in count_words:
                count_words[word] = count
                count += 1
len(count_words)
```

5. Create vocabulary:

Machines are not familiar with complex English words so, to process model'sdata they need a simple numerical representation. That's why we map every word of the vocabulary with a separate unique index value. An in-built tokenizer function is present in the Keras library to create tokens from our vocabulary.

```
THRESH = -1
count = 1
new_dict = {}
for k,v in count_words.items():
    if count_words[k] > THRESH:
        new_dict[k] = count
        count += 1

new_dict['<OUT>'] = len(new_dict)
for k, vv in captions_dict.items():
    for v in vv:
        encoded = []
        for word in v.split():
            if word not in new_dict:
                encoded.append(new_dict['<OUT>'])
            else:
                encoded.append(new_dict[word])


        captions_dict[k][vv.index(v)] = encoded
```

6. Create a data Generator:

For training the model as a supervised learning task we need to feed it with input and output sequences. Total 8000 images with 2048 length feature vector and the caption represented as numbers are present in our training sets. It's not possible to hold such a large amount of data into memory so we are going to use a generator method that will yield batches

```
Batch_size = 5000
VOCAB_SIZE = len(new_dict)

def generator(photo, caption):
    n_samples = 0

    X = []
    y_in = []
    y_out = []

    for k, vv in caption.items():
        for v in vv:
            for i in range(1, len(v)):
                X.append(photo[k])

                in_seq= [v[:i]]
                out_seq = v[i]

                in_seq = pad_sequences(in_seq, maxlen=MAX_LEN, padding='post',
truncating='post')[0]
                out_seq = to_categorical([out_seq], num_classes=VOCAB_SIZE)[0]

                y_in.append(in_seq)
                y_out.append(out_seq)

    return X, y_in, y_out
```

7.Model prediction:

Import all the required module and train the model.And model fit .

```
X = np.array(X)
y_in = np.array(y_in, dtype='float64')
y_out = np.array(y_out, dtype='float64')

from tensorflow.keras.preprocessing.sequence import pad_sequences
from keras.utils import to_categorical
from keras.utils import plot_model
from keras.models import Model, Sequential
from keras.layers import Input
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Embedding
from keras.layers import Dropout
from keras.layers import add
from keras.callbacks import ModelCheckpoint
from keras.layers import Dense, Flatten,Input, Convolution2D, Dropout, LSTM, TimeDistributed,
Embedding, Bidirectional, Activation, RepeatVector,Concatenate
from keras.models import Sequential, Model
```

8. Create CNN-RNN-LSTM representation

The RNN processes the initial image context and generates one word at a time.
It uses the previously generated words as context for predicting the next word in the sequence.
The predicted word is selected based on the RNN's internal state and the currentimage context.The process continues iteratively until a "stop" token or a maximum caption length isreached.

```
embedding_size = 128
max_len = MAX_LEN
vocab_size = len(new_dict)

image_model = Sequential()

image_model.add(Dense(embedding_size, input_shape=(2048,), activation='relu'))
image_model.add(RepeatVector(max_len))

image_model.summary()

language_model = Sequential()

language_model.add(Embedding(input_dim=vocab_size, output_dim=embedding_size,
input_length=max_len))
language_model.add(LSTM(256, return_sequences=True))
language_model.add(TimeDistributed(Dense(embedding_size)))

language_model.summary()

conca = Concatenate()([image_model.output, language_model.output])
x = LSTM(128, return_sequences=True)(conca)
x = LSTM(512, return_sequences=False)(x)
x = Dense(vocab_size)(x)
out = Activation('softmax')(x)
model = Model(inputs=[image_model.input, language_model.input], outputs = out)

# model.load_weights("../input/model_weights.h5")
model.compile(loss='categorical_crossentropy', optimizer='RMSprop', metrics=['accuracy'])
model.summary()

model.fit([X, y_in], y_out, batch_size=512, epochs=50)


inv_dict = {v:k for k, v in new_dict.items()}

model.save('model.h5')

model.save_weights('mine_model_weights.h5')

def getImage(x):

    test_img_path = images[x]

    test_img = cv2.imread(test_img_path)
    test_img = cv2.cvtColor(test_img, cv2.COLOR_BGR2RGB)
```

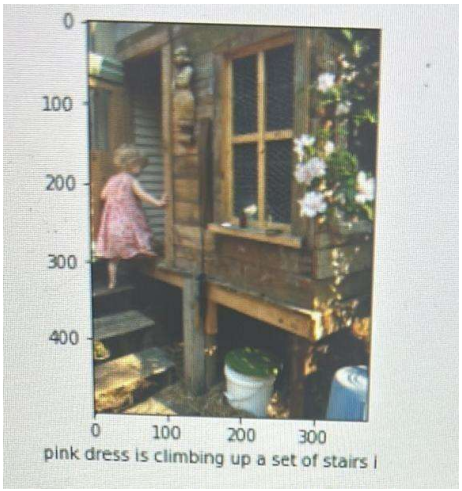
We train with 150 epoch if your gpu is better u can try with more epoch if your gpu is less it might take hours to load

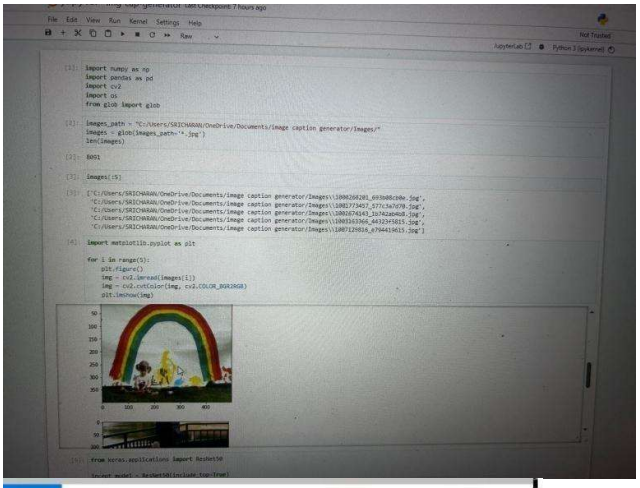
9.Model prediction:

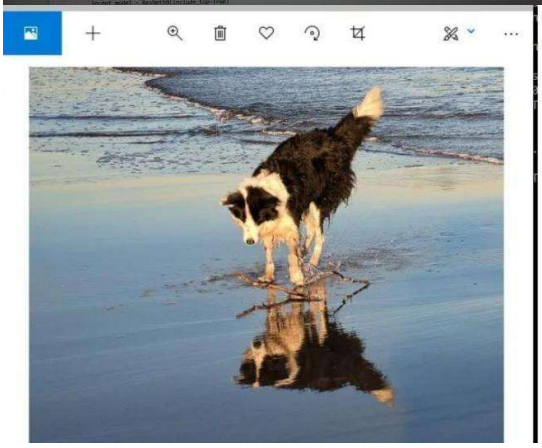
After successful model training, our task is to test the model accuracy by inputting test image data.we predict the model with images to get the output we want.

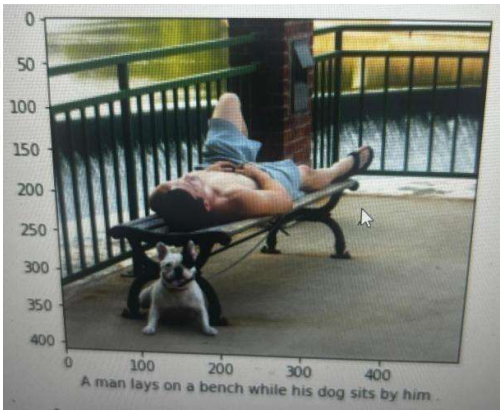
```
# PREDICTIONS
for i in range(5):

    no = np.random.randint(1500,7000,(1,1))[0,0]
    test_feature = modele.predict(getImage(no)).reshape(1,2048)

    test_img_path = images[no]
    test_img = cv2.imread(test_img_path)
    test_img = cv2.cvtColor(test_img, cv2.COLOR_BGR2RGB)


    text_inp = ['startofseq']

    count = 0
    caption = ''
    while count < 25:
        count += 1

        encoded = []
        for i in text_inp:
            encoded.append(new_dict[i])

        encoded = [encoded]

        encoded = pad_sequences(encoded, padding='post', truncating='post', maxlen=MAX_LEN)


        prediction = np.argmax(model.predict([test_feature, encoded]))

        sampled_word = inv_dict[prediction]

        caption = caption + ' ' + sampled_word

        if sampled_word == 'endofseq':
            break

        text_inp.append(sampled_word)

    plt.figure()
```

SCREENSHOTS

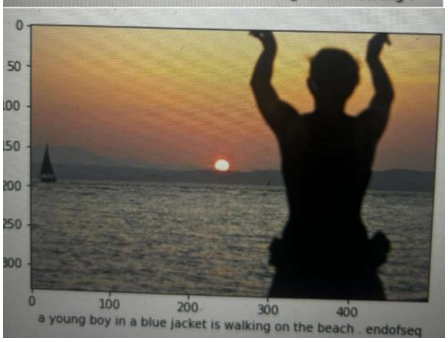


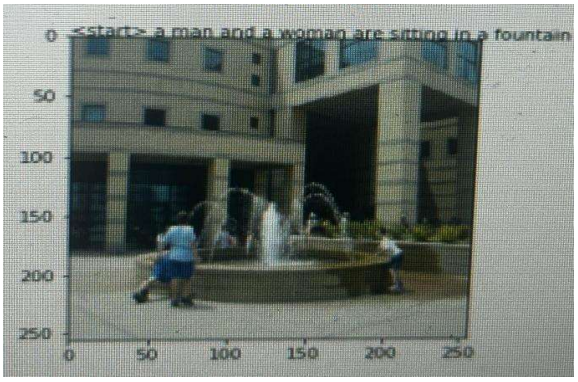





A man lays on a bench while his dog sits by him .


A man in an orange hat starring at something .


a young boy in a blue jacket is walking on the beach . endofseq


<start> a man and a woman are sitting in a fountain