DEEP LEARNING BASED IMAGE CAPTION GENERATION BASED ON CONTEXT

Ashna C P
Roll No: 15
Reg.No: KTE22MCA-2015

Guided By
Prof. Shilpa M Thomas
Department of Computer Applications
Rajiv Gandhi Institute of Technology, Kottayam

March 17, 2024

Contents

1. Introduction
2. Current state of Art
3. Motivation
4. Objectives
5. Literature Survey
6. Proposed Methodology
7. Materials and Methods
8. Results
9. Conclusion and Future Scope
10. Implementation Status and Plan
11. Reference
12. Git History

Introduction

- Image caption generation integrates visual perception and linguistic expression.
- Advances visual comprehension by enabling computers to describe images, enhancing human-computer interaction.
- It addresses the need for intelligent systems to understand visual content.
- Utilizing CNNs and RNNs, it aims for accurate image captioning.

Current state of Art

- Early image captioning methods lacked contextual understanding, resulting in generic or inaccurate descriptions, and thereby limiting their practical relevance.
- Deep learning enhances context-based image captioning, reduces manual inspection and integrating visual and textual inputs for more meaningful captions.

Motivation

- Deep learning-based context-based image caption generators enhance user experience, accessibility, and searchability with accurate descriptions.
- This technology allows for the creation of personalized and contextual image captions, leading to increased engagement and user satisfaction.
- With the aid of deep learning, context-based image caption generators streamline caption creation for large image sets, ensuring consistency and quality while saving time and resources.

Objectives

- The objective of this project is to build a working model of Image caption generator by implementing CNN with LSTM.
  - Learn CNN and LSTM concepts to develop an Image Caption Generator.
  - Implement a CNN-LSTM model where image features are extracted from Xception.
  - Utilize LSTM to generate descriptive captions for images.
  - Collect and preprocess image datasets from repositories like Kaggle.
  - Train the CNN-LSTM model with extracted image features.
  - Evaluate model performance using metrics like BLEU for caption quality.

Literature Survey

Table 1: Literature Survey

Sl No.	Title	Author	Model	Accuracy	No. of images in dataset
1	Context-based Image Caption using Deep Learning	M. Israk Ahmed et al.	Resnet101	78.3%	113,287
2	Show, Attend and Tell: Neural Image Caption Generation with Visual Attention	Fu yuesheng et al	GoogLeNet	96.88%	600
3	Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering	Anderson et al	CNN	70.3%	5100
4	Image Captioning with Semantic Attention	You et al	GoogleNet	75.06%	30000

Proposed Methodology

The proposed method includes several key stages.

- Data Collection: Gather image datasets from diverse sources, including repositories like Kaggle.
- Feature Extraction: Extract image features using the Xception pre-trained CNN model.
- Training: Utilize the extracted features to train the LSTM model for generating image captions.
- Evaluation: Assess the model's performance using evaluation metrics like BLEU.
- Fine-Tuning: Refine the model parameters and architecture to enhance caption generation accuracy.
- Validation: Validate the model's effectiveness through rigorous testing on unseen data.

Proposed System Architecture

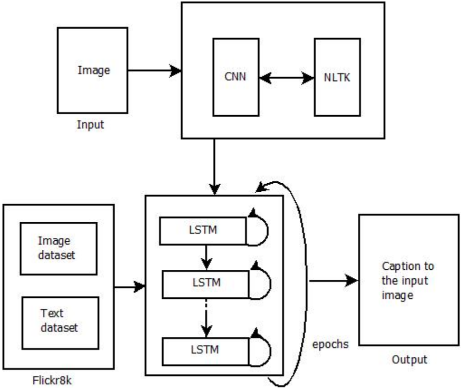
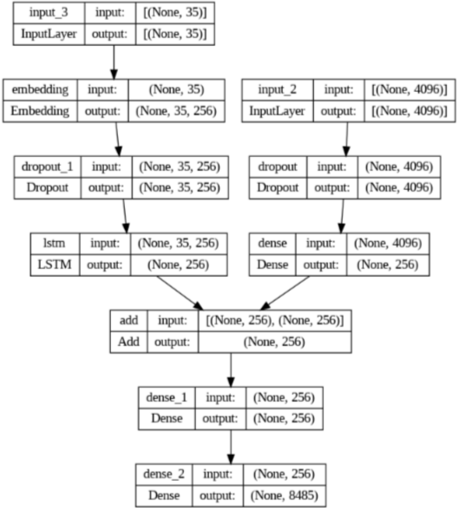


Figure 1: System Architecture

Proposed System - Model Architecture



input_3 | input: | [(None, 35)]
InputLayer | output: | [(None, 35)]

embedding | input: | (None, 35)
Embedding | output: | (None, 35, 256)

input_2 | input: | [(None, 4096)]
InputLayer | output: | [(None, 4096)]

dropout_1 | input: | (None, 35, 256)
Dropout | output: | (None, 35, 256)

dropout | input: | (None, 4096)
Dropout | output: | (None, 4096)

lstm | input: | (None, 35, 256)
LSTM | output: | (None, 256)

dense | input: | (None, 4096)
Dense | output: | (None, 256)

add | input: | [(None, 256), (None, 256)]
Add | output: | (None, 256)

dense_1 | input: | (None, 256)
Dense | output: | (None, 256)

dense_2 | input: | (None, 256)
Dense | output: | (None, 8485)

Figure 2: Model Architecture

Data Flow Diagram

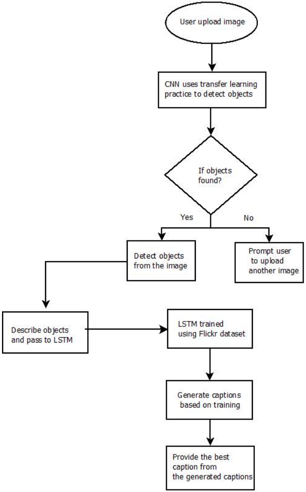

User upload image
CNN uses transfer learning practice to detect objects
If objects found?
Yes
No
Detect objects from the image
Prompt user to upload another image
Describe objects and pass to LSTM
LSTM trained using Flickr dataset
Generate captions based on training
Provide the best caption from the generated captions

Figure 3: DFD

Materials and Methods - Dataset Details

- The Flickr8k dataset contains 8091 images with 5 english captions per image.
- This dataset is available on kaggle and has a size of 1GB.
- Divided the dataset into 6000 images for training, 1000 for validation and 1000 for testing.
- Cleaned the description by removing punctuations and converted all words to lowercase and removed numbers.

Software Tools

- **Language :** Python
- **Dataset :** Flickr8k
- **Operating System :** OS Independent
- **Platform:** Google Colab

Hardware Tools

- **Processor :** Intel I3
- **Speed :** 1.6 Ghz
- **RAM :** 4GB (min)
- **Hard Disk :** 500 GB

Results

- Actual captions
  - startseq child playing on rope net endseq
  - startseq little girl climbing on red roping endseq
  - startseq little girl in pink climbs rope bridge at the park endseq
  - startseq small child grips onto the red ropes at the playground endseq
  - startseq the small child climbs on red ropes on playground endseq
- Predicted caption
  - startseq little girl grips the red ropes endseq

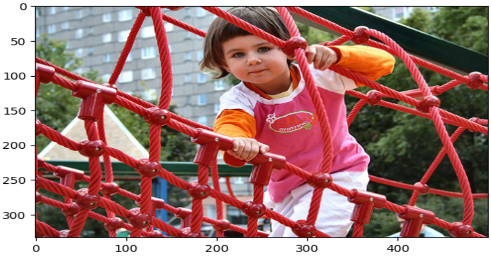


Figure 4: An image used for generating caption.

Conclusion and Future Scope

Conclusion

- Integration of CNN and LSTM networks enables object detection and image captioning.
- Through efficient feature extraction, image caption generator demonstrates a significant advancement in AI-driven image understanding, paving the way for enhanced accessibility, content retrieval etc.

Future Scope

- Enhance the predictions by using more training example. For example by using Flickr32k dataset which has upto 32000 images.
- Extending the model to caption live video frames promises advancements in accessibility tools and security systems.

Implementation Status and Plan

Table 2: Implementation Status and Plan

Task	Status	Remarks
Dataset Collection	Completed	
Dataset Preprocessing	Completed	
Feature extraction using VGG16	Completed	
Preprocessing of text data	Completed	
Implementation using CNN and LSTM model	Completed	
Training	Completed	
Model Evaluation	Completed	
Fine-Tuning the Best-Performing Model	Yet to Start	Planning to Complete by April 15th 2024

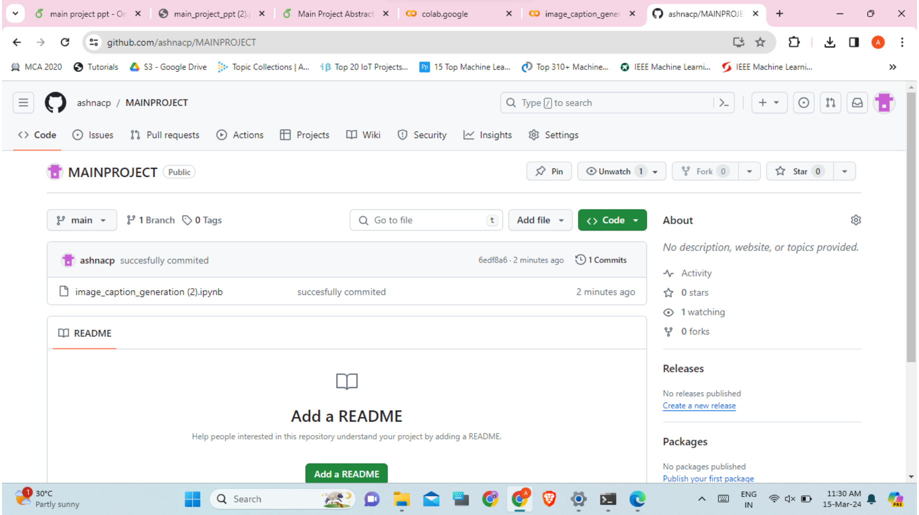
Reference

[1] Abhaya Agarwal and Alon Lavie. “Meteor, m-bleu and m-ter: Evaluation metrics for high-correlation with human rankings of machine translation output”. In: *In Proceedings of the Third Workshop on Statistical Machine Translation. Association for Computational Linguistics, 115–11*. 2008, pp. 235 243. DOI: 10.1109/ICCCSP52374.2021.9465499.

[2] Ahmet Aker and Robert Gaizauskas. “Generating image descriptions using dependency relational patterns.”. In: *Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 1250–1258*. 9 (2010), pp. 113599–113611. DOI: 10.1109/ACCESS.2021.3105112.

[3] P. Anderson. “Spice: Semantic propositional image caption evaluation”. In: *In European Conference on Computer Vision. Springer, 382–398*. 11 (Jan. 2018), pp. 217–223. DOI: 10.25165/ijabe.v11i4.2690.

Git History



github.com/ashnacp/MAINPROJECT

ashnacp / MAINPROJECT

MAINPROJECT Public

1 Branch 0 Tags

ashnacp succesfully commited 6edf8a6 · 2 minutes ago 1 Commits

image_caption_generation (2).ipynb succesfully commited 2 minutes ago

About

No description, website, or topics provided.

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Add a README

Help people interested in this repository understand your project by adding a README.

Figure 5: Git history

Thank you!