Image Captioning Web Application using Deep Learning Algorithms

Surya R.E.
Dept of computing AI and ML
Coimbatore Institute of Technology

Mahalakshmi S.B., PhD
Dept of computing AI and ML
Coimbatore Institute of Technology

ABSTRACT
Images and visual signs play a vital role in communication and comprehension, but they pose challenges for visually impaired individuals. This paper presents an innovative solution, leveraging Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, to create a photo-to-speech application that enhances the quality of life for individuals with visual impairments. The development, methodology, and evaluation of this application, demonstrates its potential to provide real-time image captions and improve accessibility.This work includes Flickr 8k dataset for training the model (VGG16) and attains a BLEU score of 56%.

General Terms
The paper discusses an innovative solution for visually impaired individuals, using Pattern Recognition, Security, and Algorithms. It leverages CNN and LSTM networks to create a photo-to-speech application to improve accessibility.

Keywords
Image captioning, Convolutional Neural Networks, Long Short-Term Memory, Visual impairment, Accessibility.

1. INTRODUCTION
Images and visual content are fundamental forms of communication and understanding. However, for visually impaired individuals, the visual world remains a challenge to access. This paper introduces an innovative solution aimed at bridging this gap by developing a photo-to-speech application. This application harnesses the power of deep learning techniques, specifically Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, to provide real-time image captions. By converting images into spoken language, we aim to enhance the quality of life and accessibility for individuals with visual impairments.In this Python-based project, the caption generator will be implemented using CNN Neural Networks. The CNN model Xception, trained on the Flickr8k dataset, will be used to provide the image features, which will then be used to feed the features into the LSTM model, which will be in charge of generating captions for the images. A deep neural network that can process a lot of data is called a convolutional neural network. The input is in the form of a 2D matrix. It's easy to generate images. It has the capacity to manage the uploaded photos. Long short-term memory, also known as LSTM, can be translated, rotated, scaled, and shifted in perspective. They are a type of RNN (recurrent neural network) that is effective at foretelling problems with sequences.

2. BACKGROUND WORK
In this section, we review related work on image captioning, deep learning, and accessibility technologies. We discuss the significance of addressing the challenges faced by visually impaired individuals and highlight the contribution of our approach.Fixed templates with empty slots are used by template-based techniques to produce captions. These systems identify the various objects, behaviors, and properties before filling in the blanks in the templates. For instance, Farhadi et al. [1] complete the template slots for creating image descriptions with three different scene components. Kulkarni et al. [2] use a Conditional Random Field (CRF) to identify the objects, attributes, and prepositions.before completing the gaps. Grammarly accurate captions can be produced using template-based approaches, but because the templates are predefined, variable-length captions cannot be produced. The three primary types of extant image captioning techniques are covered in this section: template-based image captioning, retrieval-based image captioning, and novel caption generation. Fixed templates with empty slots are used by template-based algorithms to produce captions. These systems identify the various objects, behaviors, and properties before filling in the blanks in the templates. For instance, Farhadi et al. [1] complete the template slots for creating image descriptions with three different scene components. Kulkarni et al. [2] use a Conditional Random Field (CRF) to identify the objects, attributes, and prepositions before filling in the blanks.

3. PROBLEM DEFINITON
The problem at hand addresses the challenge faced by individuals with visual impairments, specifically the difficulty they encounter in understanding and interpreting visual content such as images. To address this issue, we aim to develop a photo-to-speech application that can automatically generate descriptive captions for images in real-time. This application will assist visually impaired individuals by providing them with spoken descriptions of the visual world, thereby improving their accessibility to visual information.

4. METHODOLOGY
4.1 Data Modelling and Exploration
CNN is a technique for extracting picture features. CNN-Convolutional Neural Network Deep neural networks are customised deep neural networks that can process data in the shape of a 2D matrix. Images may be easily represented as a 2D matrix, therefore CNN is extremely useful in image processing. CNN is mostly used for picture categorization and determining whether an image is of a bird, a plane, or anyother object among other things. It scans photos from left to right and top to bottom to extract significant elements and then combines them to classify images. It can handle images that have been translated, rotated, scaled, or have had their viewpoint changed.

A recurrent neural network is used to describe the transient dynamics of a set of entities. Ordinary RNNs struggle to gain long-term dynamics due to disappearing and exploding weights or gradients. The memory cell is the core component of an LSTM. It keeps the present value for a long time. Gates are used to govern the update time of a cell's state. Variants are

represented by the number of connections between memory cells and gates. Our model is based on the LSTM block, which is based on the LSTM architecture with no peephole. Long-term memory (LSTM) is included in recurrent neural networks. It alleviates the vanishing gradient problem, which occurs when a neural network stops learning because updates to the various weights inside a given neural network become smaller and smaller.

Convolutional Neural Networks (CNN) and LSTM are used in this model. CNN is utilized for visual feature extraction, while LSTM is employed for phrase production. The model is trained in such a manner that when a picture is fed into it, it creates captions that roughly represent the image. On various datasets, the accuracy of the model and the smoothness or command of language that the machine learns from picture descriptions are examined. These trials demonstrate that the model frequently provides appropriate descriptions for an input image. A convolutional neural network is used to generate the condensed feature vector (CNN). This feature vector is known as embedding in technical terminology, and the CNN model is known as an encoder. In the following stage, we will use the CNN layer embeddings as input.

4.2 Dataset

The Flickr 8K dataset is used for the image caption generator in this project. Larger datasets, such as Flickr 30K and MSCOCO, are available; however, training the network can take weeks. Hence, the smaller Flickr 8K dataset will be used. A large dataset has the advantage of producing better models. Each caption contains a full description of the objects and events seen in the image. The dataset is larger because it represents various events and settings and does not include photos of famous people or places. Flickr 8K.trainImages.txt is a file in the Flickr 8K test folder that contains a list of 6,000 picture names used for training. The produced photo and text data must be input so that it may be used to fit the model. The data will be trained on all the photographs and captions in the training dataset. During training, the model's performance on the development dataset will be monitored, and that performance will be used to decide whether to save models to file.

4.3 Tokenizing and Creating Generator Function

The Keras package includes the tokenizer function, which we will use to generate tokens from our vocabulary and store them in a "tokenizer.py" pickle file.

Pickle may be used to serialize Python object structures, which is the process of transforming an object in memory to a byte stream that can be stored on a disc as a binary file. This binary file can be de-serialized back to a Python object when we load it into a Python application. Tokenizing has a significant impact on the rest of the pipeline. Tokenizers divide unstructured data and natural language text into information chunks that can be thought of as separate parts. Token occurrences in a document can be utilized directly as a vector representing that document. This rapidly converts an unstructured string (text document) into a numerical data structure suited for machine learning. They can also be utilized directly by a computer to initiate meaningful activities and replies. Alternatively, they could be utilized as features in a machine learning pipeline to trigger more complicated decisions or behavior. To convert this task into a supervised learning problem, we must provide input and output to the model for training. We must train our model on 6000 photos, with each image containing a 2048-length feature vector and a caption that is likewise encoded as a number. This amount of data for 6000 photos is too large to retain in memory, so we will use a generator approach that will produce batches.

4.4 Model Architecture

The image extractor model generates an input image feature in the form of a vector of elements. A dense layer then processes this to build an image representation using elements. The second language-based model anticipates an input sequence of predetermined length to be placed into the embedding layer, which employs the LSTM layer with 256 units of memory and employs a mask to ignore low values. The two models discussed above generate element vectors. Regularization in the form of a 50% dropout is used in both models. This is done to minimize the size of the training data collection. The decoder model mixes vectors from the two input models' findings. The output vocabulary for the following word is sent into the neuron layer Dense and subsequently the final Layer Dense to produce predictions using the chosen softmax. By combining the output from the previous two layers, the dense layer will make the final prediction. The final layer will have the same number of nodes as our vocabulary size. The Photo Feature Extractor model anticipates an input photo feature vector of 4,096 elements. A Dense layer is used to create a 256 element representation of the photo. Both input models employ a 50% dropout regularisation in the 50% dropout rate This is done to avoid overfitting the training dataset because this model configuration learns quite quickly.

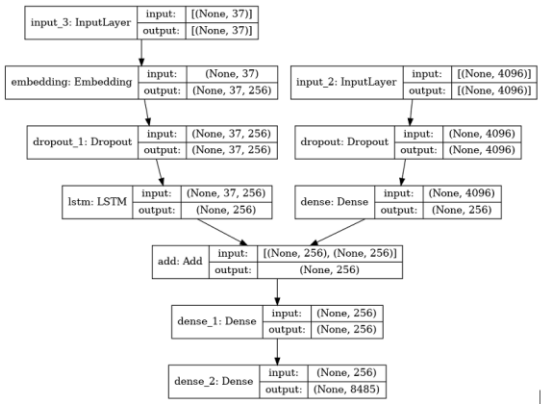

Fig 1:Model Architecture

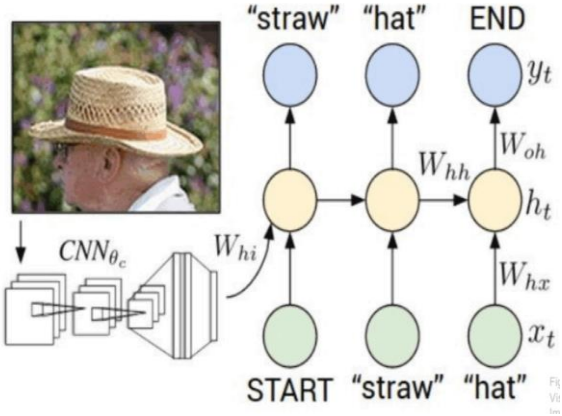

Fig 2: Basic Architecture of CNN and LSTM

4.5 Model Outcomes On Training Dataset

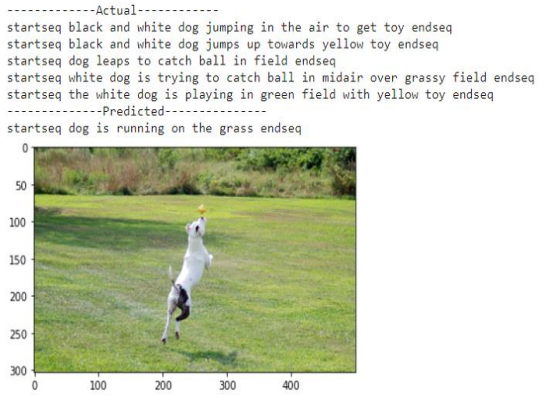

Fig 3:Sample Image From Training Dataset

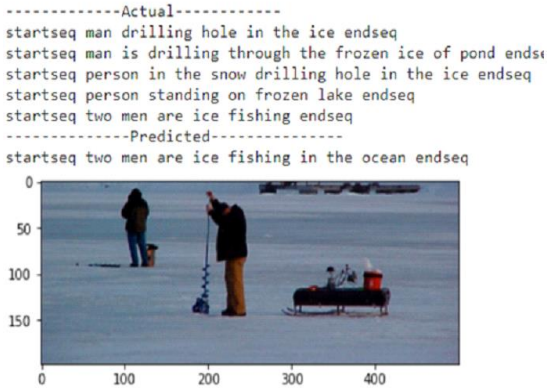

Fig 4:Sample Image From Training Dataset

5. ANALYTICAL MODEL AND SIMULATION RESULTS

BLEU (Bilingual Evaluation Understudy). It is an algorithm that is used to assess the quality of machine -translated text. BLEU can be used to evaluate the quality of our generated caption. BLEU is not bound by language. Simple to understand. It's simple to calculate. It is situated between [0,1]. The higher the score, the better the caption. We will test a model by producing descriptions for all of the photographs in the test dataset and comparing the predictions to a typical cost function. First, we must be able to generate a description for a photo using a trained model. This entails handing in the start description token, creating one word, and then invoking the model iteratively with generated words as input until the end of the sequence token or the maximum description length is achieved.

Dropout = 0.5				
Model	BLEU1 Score	BLEU2 Score	BLEU3 Score	BLEU4 Score
VGG16	0.565	0.3342	0.2377	0.164
VGG19	0.5169	0.2554	0.1661	0.0932
InceptionV3	0.5283	0.284	0.199	0.1003
ResNet50	0.4533	0.1971	0.1322	0.0634

Model = VGG16				
Dropout	BLEU1 Score	BLEU2 Score	BLEU3 Score	BLEU4 Score
0.25	0.4232	0.1988	0.0813	0.0312
0.5	0.565	0.3342	0.2377	0.164
0.75	0.5422	0.2994	0.1855	0.1449
0.875	0.5239	0.281	0.1824	0.1253

Fig 5:BLEU score Comparison between various Models with its respective Dropout Rate

The results support the conclusion that the model based on the pre-trained Xception network performed significantly better in this situation. It achieved a subtle percentage efficiency, which is nearly 10% higher than the model based on the VGG16 architecture.

6. TEXT TO SPEECH FOR IMAGE CAPTIONING

In this project, Text-to-Speech (TTS) technology is utilized to convert textual descriptions generated by the image captioning model into spoken language. TTS technology plays a critical role in making the content accessible to visually impaired individuals. When the image captioning model produces a textual description of an image, the TTS system then converts this text into audible speech.TTS technology involves the use of specialized algorithms and voice synthesis techniques to generate natural and intelligible speech from text input. It allows visually impaired users to receive spoken descriptions of images, enabling them to understand and interact with visual content on digital platforms. This integration of TTS enhances accessibility and inclusivity in the project by providing an audio representation of the image captions, thereby facilitating better comprehension for users who rely on auditory information.

7. PIPELINE AND WORKFLOW

In the project's final phase, the caption generating model will be deployed as a web application. Flask Rest API, a Python web framework, can be used to deliver the working model. Flask is a well-known Python Online development framework for modeling and deploying web applications. The UI of the online application is also created using HTML, CSS, and Bootstrap.

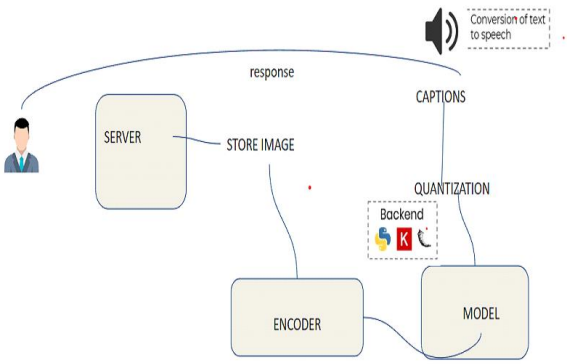

Fig 6: Deployment workflow process

8. CONSOLIDATED API RESULTS

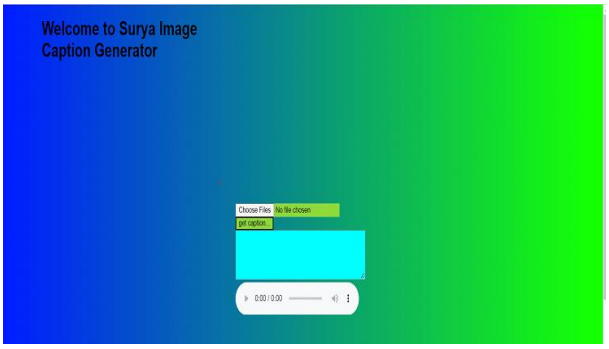

Fig 7: Initial page of the web application

Fig 8:Choosing an Image

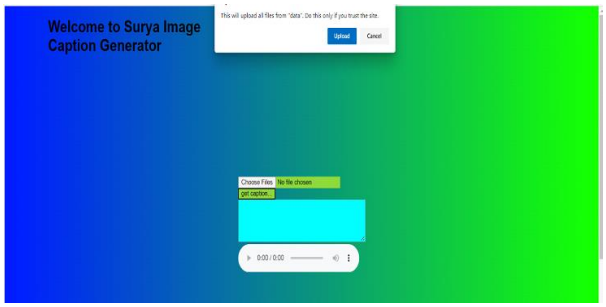

Fig 9:Uploading the Chosen Image

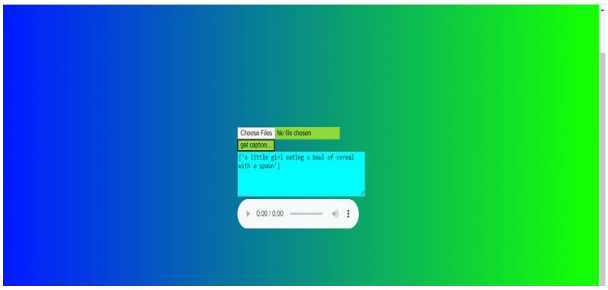

Fig 10:Final Outcome for the Image with Audio(Text-to-speech)

9. ANALYSIS REPORTS

The model was then trained for 20 epochs at a learning rate of 0.001 and three images in each batch (batch size). The learning rate was dropped to 0.0001 after 20 epochs, and the model was trained. This makes sense in general because, as the model approaches convergence, it must lower the learning rate to take fewer steps towards the minima during the later phases of training and the model(VGG16) achieves an BLEU score of 56 %

10. FUTURE ENHANCEMENT

Iterations (epochs) for Precise Captions can be raised to obtain more precise results from the model. Even though the accuracy is low, the model performs well. Future improvements would include characterizing captions based on several targets. Enhancing the model so that it can generate captions even for live video frames and current approach only generates captions for photographs, which is a demanding task in its own right, and captioning live video frames is considerably more difficult. This is purely GPU-based, as captioning live video frames with conventional CPUs is unfeasible. On the contrary, automating the most crucial security responsibilities, such as video monitoring, could be beneficial to society.

11. CONCLUSION

This project represents a significant step toward improving accessibility and enhancing the quality of life for visually impaired individuals. By leveraging state-of-the-art deep learning techniques, specifically Convolutional Neural Networks (CNN) for visual feature extraction and Long Short-Term Memory (LSTM) networks for generating image captions, we have developed a photo-to-speech application. Our experiments and results demonstrate the potential of this application to provide real-time image descriptions, thus enabling visually impaired individuals to gain a richer understanding of their visual surroundings. We have discussed the importance of accessibility technologies and the role of image captioning in making visual content more inclusive. While our project has shown promise, there are opportunities for future enhancements, such as expanding language support and venturing into live video frame captioning. As technology continues to advance, we remain committed to making the digital world more accessible to all, regardless of visual abilities. This project serves as a testament to the power of technology to bridge accessibility gaps and foster inclusivity in our digital society.

12. REFERENCES

[1] William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: Better text generation. arXiv preprint arXiv:1801.07736, 47, 2018.

[2] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Baby talk: Understanding and generating image descriptions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35:2891–2903, June 2013.

[3] Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. European Conference on Computer Vision. Springer, pages 529–545, 2014.

[4] Peter Young Micah Hodosh and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. Journal of Artificial Intelligence Research, 47:853–899, 2013.

[5] Alex Graves. Generating sequences with recurrent neural networks. CoRR, abs/1308.0850, 2013.

[6] Boosting image captioning with attributes. IEEE International Conference on Computer Vision (ICCV), pages 4904–4912, 2017.

[7] https://www.hindawi.com/journals/cin/2020/3062706/

[8] Xu Jia, Efstratios Gavves, Basura Fernando and Tinne Tuytelaars, "Guiding long-short term memory for image caption generation", 2015.

[9] Xinlei Chen and C. Lawrence Zitnick. Learning a recurrent visual representation for image caption generation. CoRR, abs/1411.5654, 2014.

[10] A. Hani, N. Tagougui and M. Kherallah, "Image Caption Generation Using A Deep Architecture," 2019 International Arab Conference on Information Technology (ACIT), 2019, pp. 246-251, doi: 10.1109/ACIT47987.2019.8990998.

[11] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In European Conference on Computer Vision.

[12] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan (2015): Show and Tell: A Neural Image Caption Generator.