International Journal For Technological Research In Engineering
Volume 10, Issue 10, June-2023 *ISSN (Online): 2347 - 4718*

A.I. IMAGE CAPTIONING BOT

[1]ANMOL DEMBLA, [2]DHRUV MALHOTRA, [3]ANSH GOYAL, [4]VAIBHAV JAMDAGNI, [5]MR. PRADEEP KUMAR
[1,2,3,4]STUDENTS, [5]GUIDE
COMPUTER SCIENCE & ENGINEERING
HMR INSTITUTE OF ENGINEERING TECHNOLOGY & MANAGEMENT

Abstract— Using Machine Learning to generate captions for images is a really useful tool in recent years. They can be used in a variety of fields from aiding blind people to identifying criminals for police.

In this paper, we will be studying the RNN Model to generate captions for images with high accuracy.

Keywords- Python, Machine Learning, Neural Network, RNN Model, pyttsx3

I. INTRODUCTION

Oftentimes, we often stumble upon things or places that we wish to know more about but can't find desired information even from the internet because we don't have relevant captions to search.

The A.I. Image Captioning Bot is a system based on Machine Learning which helps to identify images as input and generate captions using its dataset.

The proposed system can be helpful in various fields like giving useful information about objects we don't know about or helping the blind people by reading aloud captions of said objects or even in development of self-driving cars.

II. PROPOSED SYSTEM

For this system, we have used the RNN Model to identify images and generate captions for it.

A recurrent neural network (RNN) is a type of artificial neural network which uses sequential data or time series data. These deep learning algorithms are commonly used for ordinal or temporal problems, such as language translation, natural language processing (nlp), speech recognition, and image captioning; they are incorporated into popular applications such as Siri, voice search, and Google Translate.

The system will first ask the user to enter the image the user wants to generate captions for. The system will then use the RNN Model and its training dataset to generate captions for the input image. The pyttsx3 library of Python will be used to read aloud the generated captions.

III. NEED

i. We can create a product for the blind which will guide them traveling on the roads without the support of anyone else. We can do this by first converting the scene into text and then the text to voice

ii. Automatic driving is one of the biggest challenges and if we can properly caption the scene around the car, it can give a boost to the self-driving system.

IV. OBJECTIVES

i. Train the model to recognise images and generate captions.

ii. Improve the model to increase the accuracy of the model.

iii. Use text-to-speech library for model to speak the generated captions.

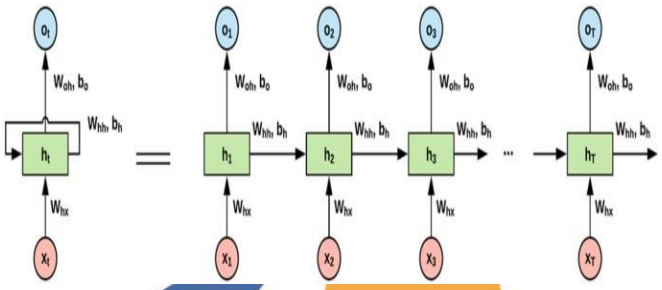
V. TECHNOLOGIES USED Python:

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation via the off-side rule.
Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.
Python consistently ranks as one of the most popular programming languages.

Jupyter Notebook:
Project Jupyter is a project to develop open-source software, open standards, and services for interactive computing across multiple programming languages. It was spun off from IPython in 2014 by Fernando Pérez and Brian Granger. Project Jupyter's name is a reference to the three core programming languages supported by Jupyter, which are Julia, Python and R. Its name and logo are an homage to Galileo's discovery of the moons of Jupiter, as documented in notebooks attributed to Galileo. Project Jupyter has developed and supported the interactive computing products Jupyter Notebook, JupyterHub, and JupyterLab.

Flask:
Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

RNN Model:
RNN or recurrent neural network is a class of artificial neural networks that processes information sequences like temperatures, daily stock prices, and sentences. These algorithms are designed to take a series of inputs without any predetermined size limit. More importantly, what makes RNN unique is that these algorithms process sequences by retaining the memory of the previous value or state in the sequence. So, in RNNs the output of the current step becomes the input of the next step and so on. This means, at every stage, the model considers both the current input and all of the previous outputs.



VI. IDENTIFY, RESEARCH AND COLLECT IDEA

There are several steps we can take to identify, research, and collect ideas for an image captioning project using Machine Learning:

Identify the problem or challenge we are trying to solve: The first step is to clearly define the problem or challenge that we are trying to address with our image captioning project. This will help us to focus our research and ensure that we are collecting ideas that are relevant and useful for our specific goals.

Research existing approaches and techniques: Once we have defined our problem or challenge, we can begin researching existing approaches and techniques for solving it. This can include reading academic papers and articles, reviewing existing software libraries and frameworks, and searching online for relevant resources and examples.

Collect ideas and potential solutions: As we research existing approaches and techniques, make a list of potential ideas and solutions that we think might be relevant and useful for our project. This can include specific algorithms or techniques, as well as potential tools and frameworks that we might use to implement our solution.

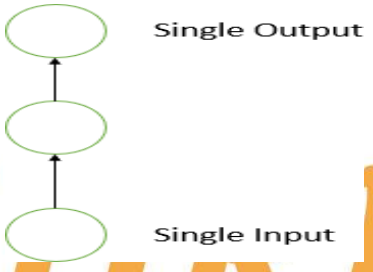
Evaluate and prioritize your ideas: Once we have a list of potential ideas and solutions, it is important to evaluate and prioritize them based on their feasibility, potential impact, and other relevant criteria. This will help us to identify the most promising and relevant ideas to pursue further in our project.

Overall, identifying, researching, and collecting ideas for an image captioning project using Machine Learning is a process of continuous learning and exploration, and may involve reviewing a wide range of resources and approaches to find the best solution for our specific needs and goals.

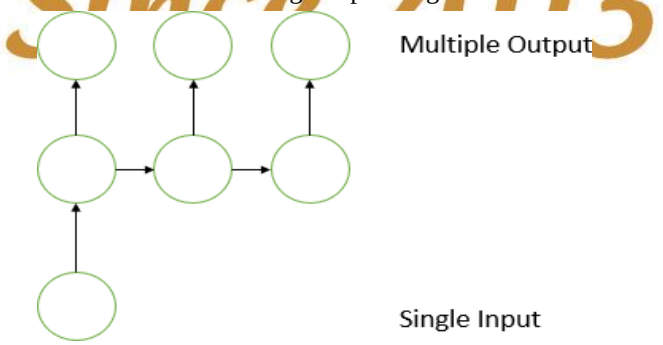
VII. MODELS

A recurrent neural network (RNN) is a type of artificial neural network which uses sequential data or time series data. These deep learning algorithms are commonly used for ordinal or temporal problems, such as language translation, natural language processing (nlp), speech recognition, and image captioning; they are incorporated into popular applications such as Siri, voice search, and Google Translate.

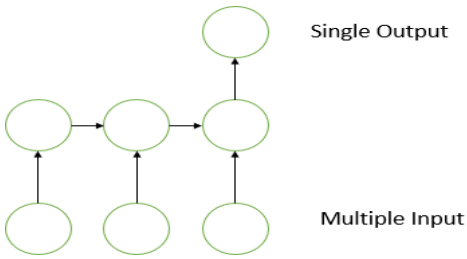
There are 4 common types of RNN Model: One-to-One :
The simplest type of RNN is One-to-One, which allows a single input and a single output. It has fixed input and output sizes and acts as a traditional neural network. The One-to-One application can be found in Image Classification.



One-to-Many :
One-to-Many is a type of RNN that gives multiple outputs when given a single input. It takes a fixed input size and gives a sequence of data outputs. Its applications can be found in Music Generation and Image Captioning.

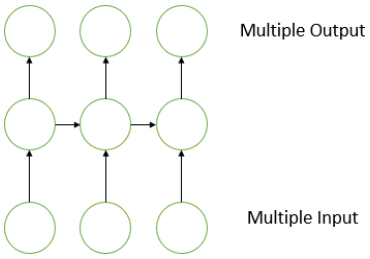


Many-to-One :
Many-to-One is used when a single output is required from multiple input units or a sequence of them. It takes a sequence of inputs to display a fixed output. Sentiment Analysis is a



common example of this type of Recurrent Neural Network.

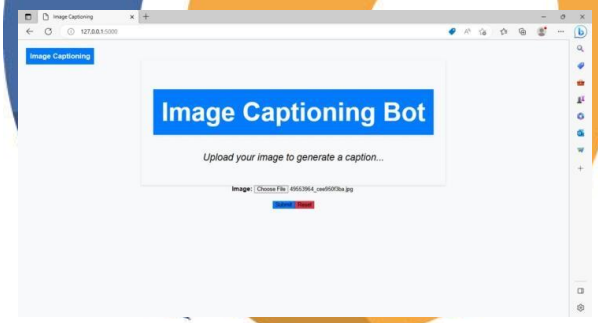
Many-to-Many :
Many-to-Many is used to generate a sequence of output data from a sequence of input units.



VIII. METHODOLOGY

I. Read the research paper and create the database.
II. Make UI frontend using HTML and CSS.
III. Data collection and Data cleaning for Machine Learning models
IV. Loading the training set, data pre-processing of images and data preparation using generator function.
V. Implement word embedding and RNN Model Architecture.
VI. Use pyttsx3 library for text-to-speech conversion of generated captions.
VII. Test the model on various levels

IX. SCREENSHOTS



X. CONCLUSION

Overall, the RNN Model proved to be an effective tool to identify and generate captions for images. We have successfully tested the system on various images and have gotten satisfied results.

In future work, we could improve the performance of the system by exploring other deep learning algorithms or neural network concepts such as convolutional neural networks (CNNs) or Long Short-Term Memory (LSTM).

REFERENCES

i https://www.ijert.org/image-caption-generator
ii https://doi.org/10.1051/matecconf/201823201052
iii https://www.researchgate.net/publication/365636832 I mage_Caption_Bot
iv https://www.researchgate.net/publication/3712 22337_Automatic_image_caption_generation_using_ de ep_learning
v https://www.ijert.org/image-captioning-system- using- recurrent-neural-network-lstm
vi https://www.ijert.org/image-caption-generator

