Automated Image Captioning for Visually Impaired

DR. DAYANAND J[1], SIDDALINGA[2], YUVRAJ[3], SANGAMESH PATIL[4], PANKAJ[5]
[1, 2, 3, 4, 5] *Dept. of Computer Science and Engineering, Guru Nanak Dev Engineering College, Bidar*

Abstract- ***The study of machine learning algorithms and processes that are suited for each image and language process. The use of existing packages in the implementation of machine learning algorithms. Implementation of an Associate in Nursing algorithmic programme that takes a photograph and explains it in full phrases. A list of domain experience and prospective models was developed after the demand analysis. Following that, a study of the available technologies that may aid with the creation of the appliance was conducted, and the look, The solution's implementation and validation began. A deep convolutional neural network is used to extract features and transfer learning is combined with a repeating neural network for description generation in this study. The implementation is done with Keras with a TensorFlow backend. It was possible to obtain Photo exploitation language is described by a trained model.***

Indexed Terms- ***image captioning, machine learning, model.***

I. INTRODUCTION

Over the last few decades, our daily lives have become increasingly reliant on technology. From providing quick answers via search engines to facilitating global, low-cost communication via internet-based messaging programmer to tackling difficult engineering challenges, the internet has made it possible.

Artificial intelligence is a field that has been rapidly increasing as a result of the ever-growing knowledge and data, as well as new technology. More and more businesses are attempting to include Others try to benefit from the knowledge that may be acquired using these approaches by incorporating algorithms for data science and machine learning into their product development. [1].

In artificial intelligence, the algorithm type that is used, sometimes referred to as deep learning or machine learning, The algorithm employed determines this. Applications range from gaining insights into a company's from assisting in the construction of better detecting items from a satellite image using guiding systems (GPS applications and maps), we've got you covered.

The list could go on and on, ranging from translation to text-to-speech, there's something for everyone software, we've got you covered [2].

Computer science, mathematics, and statistics are the three disciplines that make up computer science three disciplines that make up computer science most prominent subjects involved in the development algorithms for machine learning. The method distinguishes Deep learning algorithms and machine learning algorithms are two different types of algorithms. Deep learning employs neural networks, which have enormous computational capacity but also necessitate a large the quantity of data and powerful technology to produce beneficial outcomes. Machine learning is a subclass of it, as well. Its basis is the creation of self-learning neural networks, Machine learning algorithms, on the other hand, are regarded traditional Data-driven algorithms that attempt to learn from it so that they can use it in new ways. In the event where an algorithm for machine learning produces a a poor forecast, the designer must alter retrain the model by changing the parameters if there is a deep learning algorithm that can tell if a forecast is excellent or bad. [3]. Hyperparameter tweaking is the term for this step. It is necessary to feed good data to an algorithm in order for it to be accurate. The datasets clean toys are generally available for toy crafts. and require little or no adjustment, whereas In most cases, real-life data isn't available, dirty and necessitates extensive cleaning and normalization. This stage is known as data preparation It might consume up to 80% of a machine learning engineer's time work to complete. When it comes to unstructured information, the possibilities are endless like photos and text, even publicly available datasets must be changed to meet

the neural network's inputs or standards. [4].

Data can be extracted from nearly anything in their environment. When it hears a noise, for example, It has the ability to express itself in natural language. Machines, on the other hand, are unable to accomplish this. Every sense that humans have can be put to use in this way. Consider what would happen if one of those senses was absent. When it comes to printed information, deaf persons can benefit from machine learning programmer that use to "read" to them, a text-to-speech technology will be used.

Images can also be utilized to extract data. When a person is shown a floral image, she or he will be able to recognize the thing since she or he has seen a flower previously.

However, if that individual is blind, they will be unable to perceive that image at all. What if they could acquire a portion without having to look at the image or have it described to them? of the computer without having to look at the image or be told what it means? In this case, Machine learning has the ability to be useful. [5].

The purpose of this research is to extract useful textual data in the form of short descriptions from a variety of sources, photographs. To ensure complete sustainability, the A text-to-speech engine is a programme that converts text into speech. can be used to read the results. This manner, people with this impairment could have a fully independent experience. These people may feel left out as a result of the development of social media interactions, especially when their relatives and close friends post photos on the internet.

The rest of the paper is formatted as follows: A field is displayed in Section II research as well as various current systems, Section III presents the proposed notion, and Section IV presents the conclusions.

II. WORK ON THE SUBJECT

A top-down and bottom-up attention device that works together is provided in this study, allowing to pay attention to estimate at the level of objects and other types of data prominent visual regions, which serves as a natural base for attracting attention consideration. [6].

Picture The process of creating captions is called process of developing a text that describes an image. Furthermore, defining the material of a picture automatically is a key AI challenge that NLP (natural language processing) with computer vision are linked. We present a solution in this work quick overview of certain technical elements and methodologies for picture description generation. [7]. The study [8] discusses a new difficulty in picture captioning: how to successfully infuse feelings into generated captions while maintaining semantic correspondence between the visual and descriptive materials that are produced [9] presents a comparison study of a model for deep learning picture caption creation that is based on attentiveness. In picture For producing sentence descriptions, the Attention-based encoder-decoder networks are particularly valuable [10]. In addition, the mechanism of attention pays to alterations in the encoder network's output and a more prominent portion of the image. For the decoder network's input, Feature vectors are created by converting feature maps to feature vectors. Important processes for writing image descriptions are also included in the document., as well as datasets that are commonly utilized and assessment measures for calculating performance.

This research focuses on picture to text translation, or, to put it another way, making linguistic make sense of an image's contents Investigating the various algorithms and the optimum combinations of them would be the initial stage in the model architecture process. Because there are so many options to choose from when it comes to most common problems, this section may be extensive.

Following the selection of the suitable collection of methods and models, the following step is to choose a relevant dataset. Many photo datasets are utilised for supervised learning on a wide scale, however specifically for different The majority of them are sorts of applications that have been collected and labelled. An image dataset containing In an ideal world, in the form of written statements, labels that summarise the image's contents would be chosen. To gain knowledge, deep learning algorithms necessitate a large amount of data. produce correct predictions,

hence the dataset must include a large number of entries.

Following the selection of the architecture, models, and dataset, the dataset must be preprocessed to ensure that it has all of the algorithms' required inputs For a neural network that works with images, This normally requires scaling the images, but it also entails textual data tokenization and vectorization in the case of a a language-processing neural network.

The models must be implemented after the data has been preprocessed. Throughout the training part, they must Keep an object-oriented mindset and make yourself visible and easy to utilize.

This section involves running fresh data points through the saved results and analyzing them using certain metrics. If the results are satisfactory, the algorithm might be considered a success. Older machines can be used to train the algorithm, but if the dataset is too huge, it will fail may experience a “Out of Memory” problem. Although it is advised that you practice on a GPU, newer CPUs can also be used. The problem is that if the training isn't done properly, it won't be effective. Technique is sophisticated, convergence will take a long time, ranging from days to weeks.

The findings' various milestones must also be stored during the training process. The learned weights are saved at these checkpoints and can be used to infer the method with new data later. During the training process, certain checkpoints of the findings must also be saved. The learned weights are saved at these checkpoints and can be used to infer the algorithm with new data later. Furthermore, in the event of a difficulty during training, the most recent successful iteration would be kept. The next natural step after training and maintaining the algorithm's output is to put it to the test. This section involves running fresh data points through the saved results and analyzing them using certain metrics. If the results are satisfactory, the algorithm could be regarded as a triumph.

III. PROPOSED IDEA

One of the goals is to improve the quality of life for those who live in rural areas look into the broad subject of machine learning in order to figure out In this instance, the ideal methods to adopt are It's easy to settle for a good but not excellent algorithm or architecture when there are so many to pick from. Considerable study is required when dealing with a topic that requires deep learning and machine learning are two different types of learning technologies to tackle. A multitude of publications and blog postings are available, the bulk of which focus on the offered solutions' theoretical aspects.

Despite the fact that the programmable component of the application is not of the highest caliber for many data scientists. The code's quality becomes a priority when developing an application that will be used in production. Another objective is to develop software that can run in real time; No one wants to be kept waiting for more than a few seconds. This entails investigating and evaluating various platforms and packages for creating deep learning-based applications. As the industry grew, a multitude of tools were available, many of which optimized the compilation and computational components so that engineers could concentrate on enhancing and speeding up the algorithms.

At the same time, integration must be considered in addition to performance. Given the fact that this app will be incorporated, into a number of larger projects, it should be built utilizing an object-oriented approach to make it easy to integrate. As a result, the next subchapters detail the set of data, Data pre-processing and architecture encoder-decoders, training phase, and outcomes The dataset, data pre-processing, network architecture encoder-decoders, training phase, and outcomes are all improved as a consequence provided in the following subchapters.

A. The dataset

MS- COCO is an acronym that stands for "Microsoft-Common Objects in Context," was utilized to train and test the model. It was first released in 2014, and the most recent upgrade was in 2015. [11].

This dataset was generated expressly for this type of situation. It's a "large-scale item identification, segmentation, and captioning dataset," according to their website [12]. It's one of the most popular image-

related datasets applications because it comprises over 80000 tagged photos It also uses a "separate" The labelled dataset is used to train and test the decoder (captions).

Only 40000 captions and their related photos were used in this study, which were divided an 80:20 split ratio was used to divide training and validation (testing) samples. This is in reaction to a performance issue that arose throughout the training procedure.

B. The data pre-processing

In every machine learning method, data preprocessing is a critical step. If you skip this step, the model will raise because it won't get the input it expects, it will generate an error.

Both the deep neural network encoder and the neural network encoder convolutional neural network encoder decoder are both A deep convolutional neural network (DCN) is a type of neural network that encoders and deep recurrent neural network decoders in this application require data preprocessing.

Because the inception-v3 model is used in the deep convolutional neural network encoder is a type of neural network that uses deep convolutional neural networks to encode data photos must be expanded to fit the format that is necessary, i.e. (299, 299), as well as bringing the pixels into the [-1, 1] range. For image processing, TensorFlow provides the “image” module, which allows images read into memory, scaled, and encoded as jpeg Keras' high-level API inception-v3 has a method called "preprocess input." normalizes the pixels in the specified range, as indicated previously.

The language generator decoder, which is a textual data, i.e., the captions, must be preprocessed before using the help of a deep recurrent neural network. In this section, the Keras module "preprocessing" and associated methods are used. The following are the steps taken:

- Tokenization of captions, which involves dividing captions by white spaces and retaining just the words that are unique;
- There is a limit to the number of words that can be used in a vocabulary; The vocabulary in order to preserve memory, is limited to the top 5000 words;
- Transforming text into a numerical sequence;
- Word-to-word translation;
- Increasing the length of all sequences to that of the most extensive.

The end result is a vector containing an integer sequence that has been padded to fit the size of the input dataset's longest caption. The following picture depicts the outcome:

```
In [58]:  1 caption_vector[0]

Out[58]: array([  2, 354, 672,   2, 275,   3,   2,  81, 340,   0,   0,   0,   0,
                  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0,   0,   0,   0,   0,   0,   0])
```

Fig. 1 After pre-processing the caption data, an array was created.

C. The Architecture of Encoders and Decoders

The Encoder-Decoder design, which employs two Recurrent Neural Networks, is commonly utilized Machine translation, for example, is a good example of this. This architecture's main purpose is to reduce the size of the second network's vector of input uses requires.

The first neural network in this situation is a convolutional network, The second neural network is a recurrent neural network. The second neural network's vector output is identical to the first neural network's vector output. a person's input neural network. The following is a schematic of the architecture in general:

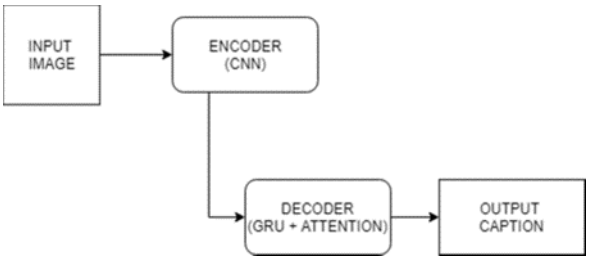

Fig. 2 High-Level Encoder-Decoder Architecture

According to the paper [13], the attention mechanism serves as a link between the encoder and the decoder. This is required since the data provided A single vector representation would otherwise be fed into the decoder. The attention approach allows the decoder to focus on the useful bits by providing Computer from all of the encoder's hidden states [14].

1) The encoder

This step produces a vector with the shape (8, 8, 2048), however the application requires a vector with the shape (64, 2048). A model class comes in handy here called "CNN Encoder" (Convolutional Neural Networks) is a term used to describe a type of neural network. RNNs are a type of neural network that consisting of a single layer that is totally connected and Activation of the ReLU gene (Rectified Linear Units). In the feature extraction stage, the extracted features are read into memory and passes across this entirely interconnected layer.

Putting it another way, the encoder is made up of features extracted Using the Inception-v3 model for transfer learning, which is then transferred over a fully linked layer that additionally handles resizing.

2) The decoder

The encoded features are taken from the encoder by the Decoder for deep recurrent neural networks. The recurrent The Gated Recurrent Unit (GRU) is a form of neural network. In order to guess the following word, the decoder evaluates the image.

The shape of the encoder's vector output, per layer, the number of neurons (units), as well as the size of the vocabulary are all factors to consider, are all inputs to the decoder during the data preparation phase. When calling, the It is necessary to pass the extracted characteristics as well as the reset state.

The context vector and attention weight are received by the attention layer the encoder for deep convolutional neural networks, and it is considered as though it were a different model. The model's embedding is a concatenation of the retrieved context vector and transmitted across the gated recurrent unit network's layers.

Two completely connected layers and a gated recurrent unit layer make up the recurrent neural network architecture. The Keras API's module "layers" can be used to add all of these layers. The GRU layer is a gated recurrent unit layer that takes the units (number of neurons) it should have as an input when it is created. Type "Dense" refers to fully connected layers (also known as "dense" layers). The units are fed into the first completely connected layer, while the vocabulary size is fed into the fully connected second layer.

D. The preparatory stage

The training period is self-explanatory. The algorithms in this technique learn to map the function parameters. This is the most difficult stage, both in terms of computing and programming. This entails running the data through the algorithm numerous times, which necessitates the setting of some parameters and the selection of some functions. The following are the logical stages that occur:

- Using the CNN encoder to extract features;
- Sending the encoder output to the decoder, together with the decoder input is set to 0 (zero) and the hidden state is set to 0 (zero) (the "start" token).
- The decoder's hidden state is fed back into the model in a loop, and the the loss is calculated using the model's predictions.

A total of 40000 entries were used for training, with a split of 80:20: Training accounts for 80% of the budget, 20% for testing, and 20% for assessing.

The entire dataset can be submitted to the algorithm in one batch if the dataset is small enough. The dataset in question, however, is rather huge and batching is required. Batching entails dividing into the data set many equal portions and giving each one to the computer algorithm one at a time. Because the batches must it is necessary to preserve proportionality between the number of batches and the number of occurrences in the dataset. The quantity of the batch employed in this study was 64.

Cross-entropy with logits was chosen as the loss function. Before passing each batch, it is set to zero to determine each batch's specific error values The overall loss is set to zero for each period. We can determine if the model converged appropriately by visualizing these data once the training procedure is completed.

Deep neural networks are a type of artificial intelligence that uses neural "learn" how to from inputs to outputs, map a function (outputs). This is determined by putting the data through its paces

various models multiple times in order to minimize the by error (loss) adjusting the number of epochs is a measure of how long a period of time has passed. The more epochs employed, the longer the algorithm takes to train, but the possibility of a superior the function of mapping grows. The algorithm has been tuned across two and twenty epochs, with the function of loss evolving in both cases, as shown in the diagram below:

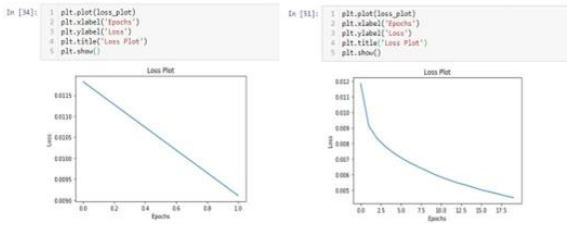

Fig. 3 There have been two epochs of loss evolution and twenty epochs of loss evolution

When looking at the plots, the number of times the algorithm has been applied on the data has increased can be seen as a substantial improvement.

During the training phase, the model's weights must be saved for later usage. The training procedure takes a long time, and all that matters are the weights in the end, or those resulting from the most cost-effective calculation. After every 100 batches, these are kept to maintain consistency a comprehensive backup. The model can be stored in a number of different formats, with the most popular being of which being of which is a "pickle." The “reference point” attribute of type index is used to save the checkpoints in this project.

It's tempting to argue that because the model grew smoothly, it should produce good outcomes and on a logarithmic scale rate after looking at the loss graphs, but this isn't this is constantly the case. The simplest method to thoroughly test the outcome is to feed it more data, previously unknown in this situation, the data points are photographs.

On a Hexa-core CPU, it took about 26 hours to train the models on 32000 instances for 20 epochs.

E. Results

Unfortunately, Due to the restricted processing capability of the device on which the models are trained, it takes a long time to train for a limited number of epochs. To accomplish this, the size of the input dataset must be reduced. In this scenario, the number of photos input is reduced to 80, and the training and testing datasets are split in an 80:20 percent ratio. As a result, 64 photos are used for training. By selecting this proportionality, additional factors such as batch size can be preserved.

In other words, this is the bare minimum of Computer that may be used without reducing other values. It's vital to take it "one step at a time" and rule out each parameter as being incorrect.

There are still 20 captions to print, both real and anticipated, following the reduction of the number of training data. This is merely a quick technique to see if the algorithm is still working without having to wait a long time. Two photos are printed as a sanity check to see how captions grow over time for different types of images. The first image shows a woman is shown tennis court in the first photograph, Boats moored near a river are depicted in the second. The first is straight a forward, the second, on the other hand, is a little more difficult to grasp due to the large number of pieces it comprises. The outcomes are as follows:

Fig. 4 After 20 epochs, the results of images 1 and 2 are shown.

After 20 epochs, the results of images 1 and 2 are shown. The outcomes are abysmal, but the method is effective. Following that, the total number of epochs has been increased to 50. As can be seen in the graph below, the results should improve slightly but not dramatically.

Fig. 5 After 50 epochs, the results of images 1 and 2 are shown.

The first image's prediction improved, correctly recognizing the ball, whereas the the forecast in the second image is erroneous. This indicates that the models have not been adequately trained over a sufficient number of iterations.

The next step is to see if the models can ever converge, even if the quantity of input photos is reduced. The number of epochs is extended to 100 in this case, with the following results:

Fig. 6 The findings of pictures 1 and 2 are exhibited 100 epochs later.

Image 1 has a better outlook than predicted after 100 epochs and can be deemed a good prediction. If you just look at that image, you'll see what I mean, you may believe that the prototypes have the converging that the epochs count required is adequate, However, if you look at image number two, you'll notice that, it's clear that the prediction remains unfavourable, Despite the fact that the white boat was correctly recognized ("a big white" – 6th diagram). The number of epochs can be extended to 200 because this is a dummy dataset with insufficient images to develop a generalized model to examine how the model reacts to complicated images. The expected outcome is that the predicted caption for the first image stays the same, while the second one does improve. The following are the outcomes:

Fig. 7 After 200 epochs, the results of images 1 and 2 are shown.

The first image's projected caption did not alter, as expected, but the second image's focus switched Out of the white yacht to the boats that are teeny-tiny, there's something for everyone, everyone will find something to their liking. Because the paddleboards are used in this area photograph something in the centre that resembles a surfboard, it improved in some ways (people). Even if it has improved, the caption isn't quite right. Given that no algorithm can ever correctly detect everything, one might conclude that this is an outlier. The bad result can be attributed to the input image's poor quality.

After fine-tuning the settings, it's clear that the initial training iteration count the algorithm's convergence was never achieved since the threshold was too low. Increasing the number of epochs is, however, a difficult task computationally costly decision, but it is critical in this circumstance. After a specific number of epochs, the model is finished, the epochs' value can be calculated. In our case, it should be between 50 and 100, or a decent outcome.

CONCLUSION

The goal of the study was to create a deep learning-based application for automated image description while also conducting extensive research in the topic. This prototype will to achieve its purpose of assisting

visually impaired people in making sense of information, it will integrate with a text-to-speech engine is required in the future.

The research component of the study was successful, enabling readers to grasp a variety of machine learning and deep learning techniques. Potential loss function, feed-forward, error propagation through a neural network's backpropagation perceptron, At the same time, activation functions and other artificial neural networks are made up of basic parts investigated, the design of a convolutional neural network, this results in sense of elements of a picture by using matrix computations and dimensionality reduction. A time-series artificial neural network is a type of artificial neural network that analyses data over time is the recurrent neural network, allowing it to make judgments on its own at each stage based on the value of the previous step, allowing it to "memorise" values while computing.

Following extensive investigation into various models and architecture, a convolutional neural network-based composite model was chosen as the encoder. Transfer learning is used to reduce A recurrent neural network, as well as training time and computing complexity is used as a decoder, which is the cutting-edge technology for dealing with textual input, notably forecasting the word that will come next in a sentence (sequential prediction). Because the traditional Because architecture can't remember long words, as a link between the two models, the attention mechanism was introduced, with its output indicating the concatenation of the attention weights and the output of the attention mechanism encoder's features.

Despite the loss's good evolution, the overall model did not behave as expected after 20 epochs of training on 32000 photos, implying that the tuning parameters needed to be adjusted. Because the technology available at the time did not allow for considerably more complicated and time-consuming tasks to be completed processes, to find the problem, the training set of 64 pictures was used to tune the parameters. The issue was that the algorithm had not yet reached a point of convergence and had not been trained for a sufficient number of epochs. Despite the short dataset, the model was able to converge after 100 training epochs. The finding is that it behaves typically, with some photos predicting captions well and others not so well.

General development and training can be done even if the hardware isn't up to scratch, as stated in the "Parameter adjustment" section, at least for prototype. This application, as a prototype, can be enhanced in a number a variety of methods. The first step should be to get a more powerful computer machine for the prototypes to be trained on. This would have an impact how long it takes the model to learn, making it faster and allowing the programmer to do more improve performance of the model (in terms of terminology of forecasting).

To take a step forward farther, rather than captioning photos, the model may be adjusted to caption movies with numerous frames per second. This could be the case used to provide additional unstructured textual data, such as movie descriptions, that can be used to summarise videos.

REFERENCES

[1] J. Anderson, L. Rainie, “Artificial Intelligence and the Future of Humans”. Pew Research Center, 2018. Available online: https://www.pewresearch.org/internet/2018/12/10/artificial- intelligence-and-the-future-of-humans/.

[2] O. Apoorva, Y.M. Sainath, G.M. Rao, “Deep Learning for Intelligent Exploration of Image Details”. International Journal of Computer Applications Technology and Research Volume 6–Issue 7, 333-337, 2017, ISSN: -2319–8656.

[3] F. Chollet, “The limitations of deep learning”, Deep Learning with Python, Section 2, Chapter 9, Essays, 2017.

[4] K. Adnan, R. Akbar, “An analytical study of Computer extraction from unstructured and multidimensional big data”. Journal of Big Data 6, No.91, 2019. Doi:10.1186/s40537-019-0254-8.

[5] O. Vinyals, A. Toshev, S. Bengio et al, “Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge”. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, Issue 4, 2017. DOI:

10.1109/TPAMI.2016.2587640.

[6] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, L. Zhang, "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering". IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6077-6086, 2018.

[7] S. Shabir, S.Y. Arafat, et al, "An image conveys a message: A brief survey on image description generation". 1st International Conference on Power, Energy, and Smart Grid (ICPESG), 2018. DOI: 10.1109/ICPESG.2018.8384519.

[8] Q. You, H. Jin, J. Luo, "Image Captioning at Will: A Versatile Scheme for Effectively Injecting Sentiments into Image Descriptions". Computer Science, 2018. Available online: https://arxiv.org/abs/1801.10121.

[9] P.P. Khaing, M.T. Yu, "Attention-Based Deep Learning Model for Image Captioning: A Comparative Study". International Journal of Image, Graphics and Signal Processing, Vol. 6, pp. 1-8, 2020.