Learning Tools for the Visually Impaired

Radhika Ganapathy[1], Preeti Poddar[2], Varada Harikumar [3], Prof. Ankit Khivasara[4]

[1,2,3]Student, Department of Electronics and Telecommunication Engineering, K.J. Somaiya College of Engineering, Mumbai, Maharashtra, India
[4]Professor, Department of Electronics and Telecommunication Engineering, K.J. Somaiya College of Engineering, Mumbai, Maharashtra, India

Abstract - *Education is a fundamental right that forms the backbone of every child and is, therefore, crucial for it to be available to everyone without any discrimination. Our primary aim with this dissertation is to make general school textbooks used by regular children available for visually impaired students so that they are not deprived of the content and knowledge learned by others. We built an image caption generating model using deep learning which is used to generate text from images of school textbooks, storybooks, and other picture books when passed into the model. We can then use the text generated and convert it into an audio format in different languages which can then help visually impaired students use general textbooks in an audio format. We used the pre-trained image captioning model VGG16 to generate captions. We built our dataset of animated textbook images from scratch and trained our model using the same. We successfully converted the text obtained into an audio format of different languages like English, Hindi, Tamil, Malayalam, etc.*

***Key Words*: visually impaired, deep learning, neural networks, technology, image caption generation, translation, languages, text to speech.**

1.INTRODUCTION

To guarantee equal prospects to all students, the accessibility of educational tools is considered a vital issue. In the field of education, the elementary concept of "non – discrimination" entails the ability of all people to have equal education opportunities, regardless of their social class, ethnicity, background, or physical disabilities. Students with disabilities have the right to receive the same standard of education as their counterparts and also have the right to access and use mainstream educational tools.

Students with disabilities face relevant difficulties both in accessing and in using e-learning tools and, depending on the kind of impairment, the types of obstacles encountered may vary considerably. After some extensive research on the educational products that are already available in the market for visually impaired students, we found a few that are used in schools. Learning is also traditionally dependent on visually-oriented ideas and information. Though this visual information is not widely obtainable in a format to learn the various science-related subjects, there are some assisting technologies for science. These include tactile maps, tactile diagram set for sciences, tactile anatomy atlas, animal models, plants, or 3-dimensional models objects. Students can touch and explore it.

Fig -1: Tactile map

This method will take a lot of time and physical effort to apply to all concepts of school textbooks. Hence, this is not the best solution for our project aim. We found a range of paid applications available in the market that provide features like text-to-speech but are far too expensive for everyone to afford.

There are many students due to their lack of knowledge, who cannot use e-learning systems for educating themselves. Moreover, even if they are aware of such systems, some of the students cannot understand how to use the tool due to their poor command of English. One of the severe problems faced by students is availability. Though a lot of screen reading software such as JAWS, NVDA, Eloquence, etc. are available for English, good software for Hindi and other regional languages are not available.

Our objective is to build a working image-caption generating model using different deep learning and natural language processing practices to recognize the context of an image and label them in English. We are using the TensorFlow and Keras Python Library to implement the pre-trained Oxford Visual Geometry

Group (VGG16). To train the image captioning model, we will be using a unique custom dataset comprised of 14300 images of drawings/illustrations/pictures collected from various school textbooks. The description obtained will then be converted to an audio format using a Text to Speech API.

The figures below illustrate the result of an image caption generating model where an image input gives a textual description of the image as the output.

Fig -2: Predicted Output: A black dog is running through the grass

Fig -3: Predicted Output: man is skateboarding on-ramp

2. LITERATURE SURVEY

Image caption generation is quite a popular research topic in Artificial Intelligence that handles image processing and produces a textual description for any given image. Important tasks in image captioning include locating the image, determining various properties of the image, identifying objects in the image, and finding the interaction between these components.

Deep learning models can learn the features automatically from the training data. Additionally, they can handle a large and varied set of images and videos. Deep learning applications such as Convolutional Neural Networks and Recurrent Neural Networks have been designed to solve problems like image captioning.

2.1 Research Undertaken

A paper on “Unified Vision-Language Pre-Training for Image Captioning and VQA” proposes a unified encoder-decoder model, called the Vision-Language Pre-training (VLP) model, which can be fine-tuned for both vision-language generation and understanding tasks. The VLP is validated in the experiments on image captioning and VQA tasks using three challenging benchmarks: COCO Captions, Flickr30k Captions, and VQA 2.0 dataset. It was observed that compared to the two cases where no pre-trained model was used or only the pre-trained language model was used (i.e., BERT), using VLP significantly speeds up the task-specific fine-tuning and leads to better task-specific models.
Another paper – “A Comprehensive Survey of Deep Learning for the Image Captioning” gives an overview of different datasets available for image captioning.

- MS COCO Dataset: It has many features like object separation, recognition in context, numerous objects per class, over 300,000 images, more than 2 million cases, 80 object categories, and 5 captions per image.

- Flickr30K Dataset: It can be used to automatically represent images and understand the language. It includes 30,000 images from Flickr, as well as 158,000 captions written by human annotators.

- Flickr8K Dataset: It is a dataset of 8000 images culled from Flickr. There are 6000 images in the training data, and 1,000 images in each of the test and production data sets. Every picture in the dataset has five human-annotated reference captions.

- The FlickrStyle10k Dataset: This dataset consists of 10,000 Flickr images with stylized captions. The training data has 7000 images. The validation and test data have 2,000 and 1,000 images.

2.2 Interview

To understand the different educational facilities available for visually impaired children, we decided to interview an individual within this field. We had a telephonic interview with Dr. Salil Jandyal, the Executive Officer at the Victoria Memorial School for the Blind. The following points summarize the questions and answers covered in the interview.

- Educating a visually impaired child is just like educating an average child. Of course, they need some supervision and a teacher who could help them bridge the gap between the course and reality. From his experience, the most crucial part of this role is not to generalize anyone.

- Making sure that every child gets an equal opportunity to access the resources is most important. There is a need to increase the number of blind schools. A considerable number of their students come from rural areas. Such students have to travel far to access our resources. So, one thing that requires a great deal of focus is: increasing the reach of teaching.

- Schools for blind students do include the use of various technologies in our teaching process. Most of the technologies and gadgets are beyond the affordable cost for a middle-class individual. Many kids who live in rural areas do not even know about such technologies. So new upcoming innovations should focus more on making devices accessible , easy to use and affordable.

2.3 Data Augmentation

To build valuable DL models, the validation error must continue to decrease with the training error. Data Augmentation is an extremely powerful technique for realizing this. The augmented data will characterize a complete set of conceivable data points, thus minimalizing the difference between training and validation sets and any future testing sets.

Fig -4: Sample image.

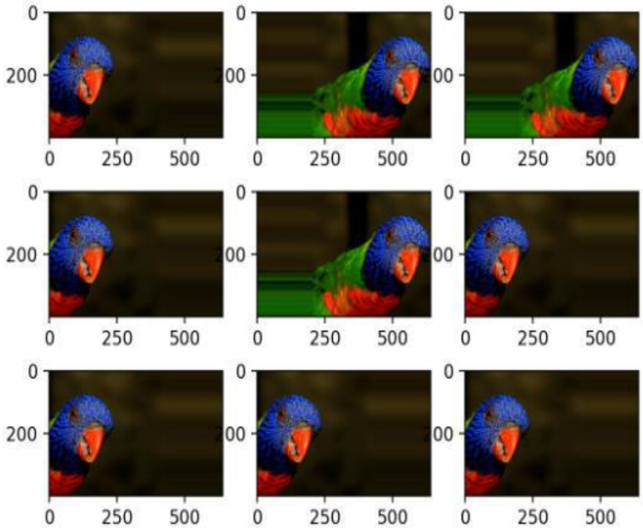


Fig -5: Data Augmentation applied to sample image.

3. PROJECT DESIGN

3.1 Problem Statement

Our project primarily aims to generate textual descriptions of photographs (captions), and at a later stage, convert them into audio format. This technology can then be used to generate audio captions of various images found in educational material (such as textbooks and charts) to provide a seamless learning experience to visually impaired children.

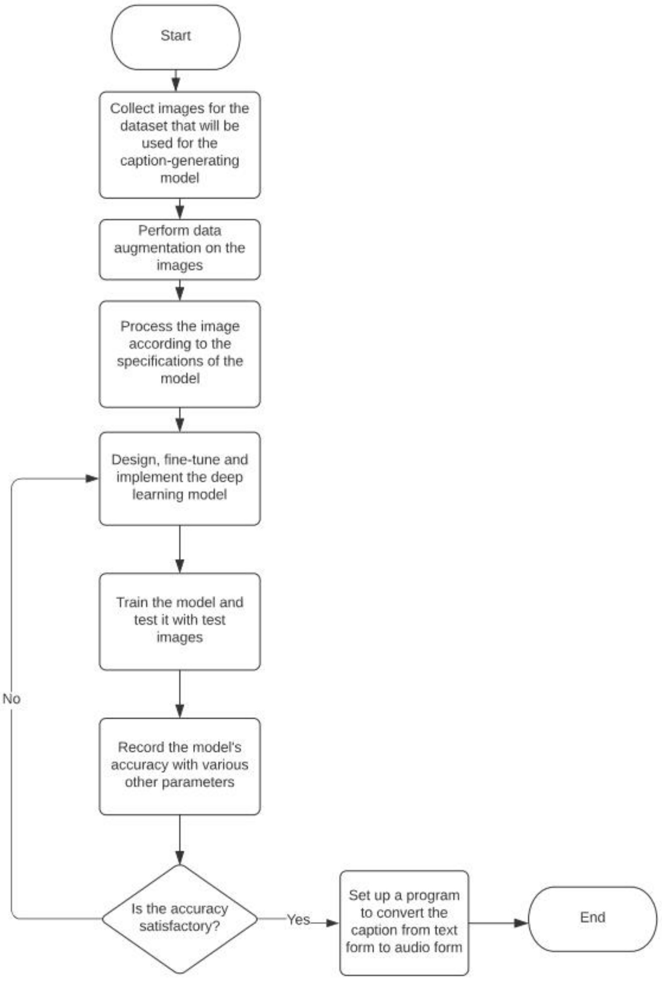

Start
Collect images for the dataset that will be used for the caption-generating model
Perform data augmentation on the images
Process the image according to the specifications of the model
Design, fine-tune and implement the deep learning model
Train the model and test it with test images
Record the model's accuracy with various other parameters
Is the accuracy satisfactory?
No
Yes
Set up a program to convert the caption from text form to audio form
End

Fig -6: Work flow of project

3.2 Custom Dataset

Initially, we trained our image captioning model using the Flickr8k dataset, but we were trying to caption textbook images, and most of them were either handmade or digitally made drawings instead of actual photographs. Since the Flickr8k dataset purely consists of real-life images, the model we trained using this dataset did not give us satisfactory results after testing it with animated images found in textbooks and other children's learning material.

So, we decided to create our own dataset using around 1300 images collected from various CBSE and State Board textbooks of different subjects. We later applied data augmentation to increase the size of this dataset to about 14300 images. As we will understand later in this paper, using such a dataset greatly increased our model's accuracy.

3.3 Data Augmentation

Data Augmentation is a practice that helps in significantly increasing the diversity of data without actually collecting it manually. Geometric transformations like flipping, color modification, cropping, rotation, noise injection, and random erasing are used to augment images in deep learning. Training deep learning models on more data can result in more skillful models. The augmentation methods can create different permutations of the images that can improve the capacity of the fit models to apply a broad view of what they have learned to new images.

3.4 VGG Model

In this paper we use the Oxford Visual Geometry Group or VGG model that won the ImageNet competition in 2014. Keras provides this pre-trained model directly. It is one of the best vision model architectures to date. This model attains 92.7% top-5 test accuracy on the ImageNet dataset, which contains 14 million images belonging to 1000 classes.

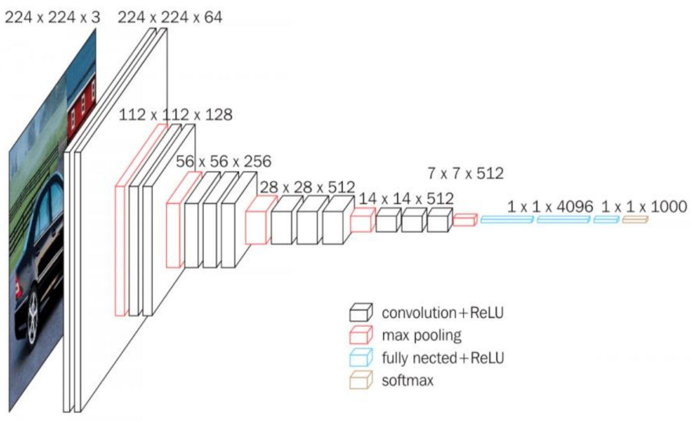


Fig -7: Architecture of VGG Model

The ImageNet Large Scale Visual recognition challenge is a yearly computer vision contest in which, every year, teams contend to complete two tasks. The first task is to detect objects inside an image belonging to 200 classes. The second task is image classification, which means to classify images, each labeled with one thousand categories. Karen Simonyan and Andrew Zisserman developed VGG-16. They were members of the Visual Geometry Group Lab of Oxford University. In 2014. They mentioned their VGG model in the paper "Very Deep Convolutional Networks for Large-scale Image Recognition." This model won prizes in both categories of the ILSVRC challenge.

The unique thing about VGG16 is that instead of having many hyper-parameters, it focuses on having convolution layers of 3x3 filter with a stride one and always using the same padding and max pool layer of 2x2 filter of stride 2.

3.5 Text to Speech

There are many APIs available to translate text to speech in the python language. One of them is the Google Text to Speech API or the *gTTS* API. *gTTS* is a very user-friendly tool that converts the textual input into audio which can be stored as an mp3 file. The *gTTS* API supports several languages like English, Hindi, Tamil, French, German, and many more. The speech can be converted into one of the two available audio speeds, fast or slow. The API is available for python users in the form of the *gTTS* module, which can be installed in the system with a simple "*pip install gTTS*" command. The python *googletrans* module can be used to translate the image captions to various languages.

4. IMPLEMENTATION

4.1 Custom Dataset

For obtaining the best possible results from our image captioning model, we used a custom dataset of textbook images. The custom dataset images were collected from storybooks for children, primary class state books of various boards, primary class CBSE textbooks, and different quizzes based on animated pictures. No specific edits have been made to the images or the captions while collecting it. Image pre-processing is handled by Keras, and captions are cleaned in the program later on. Initially, we collected 700 images, trained the model, and observed the generated captions not to be accurate. Then we slowly went on increasing the custom dataset size to 1100 and then 1300. After training our model on these 1300 images, our captions were still not matching the accuracy we hoped for, and we attributed this to the lack of images in our data set. This is where data augmentation comes into play.

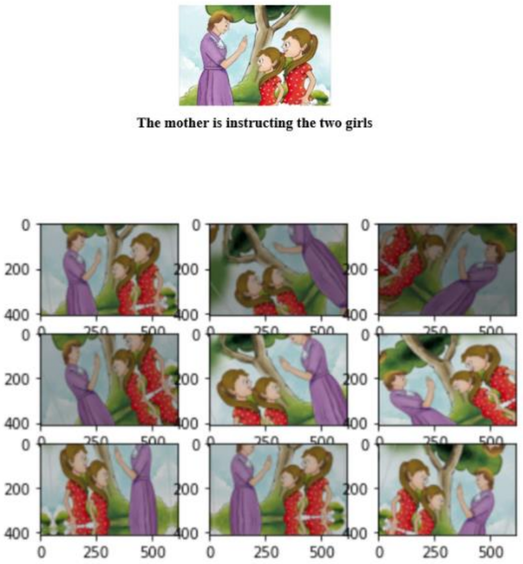


Fig -8: Data Augmentation applied on custom image data

4.2 Generating Features

The VGG 16 model could be used as part of a larger image caption model. The issue is that it is a big model, and it is redundant to run each picture across the network every time we try to test a new language model configuration.

Instead, we can use the pre-trained model to evaluate the photo features in advance and save it to a file. These features can be loaded later and fed into our model as the representation of a particular picture in the dataset. It's the same as running the picture through the entire VGG model; the only difference is that we'll have done it before.

So, we'll load each picture, prepare it for VGG, and gather the predicted features from the VGG16 model. The picture has a 1-dimensional vector of 4,096 elements. This function can be used to prepare photo data for checking our templates, and the generated dictionary can be saved to a file called *features.pkl*.

4.3 Preparing Textual Data

There is one description for each image in the dataset, and the text of the captions needs to be cleaned up a little. Each photograph has its identifier. This identifier appears in the picture filename as well as in the definition text format. After that, we'll go over the list of photo captions one by one. Each picture has a unique identifier that corresponds to a textual caption.

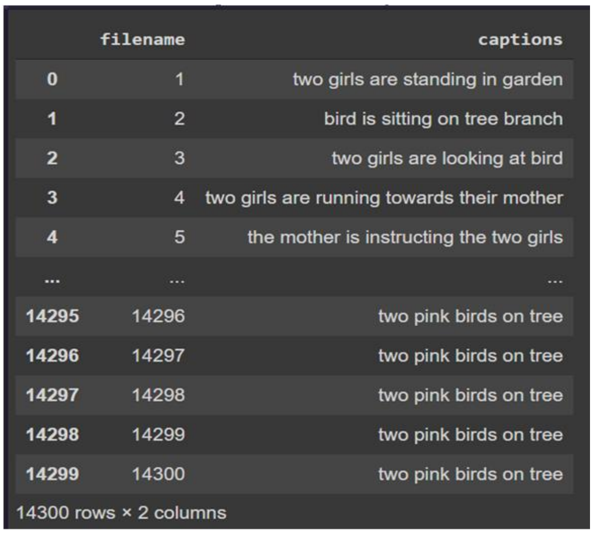
After that, we'll tidy up the caption text. The definitions have also been tokenized and are simple to use.

We'll clean up the text in the following ways to reduce the vocabulary we'll have to deal with:

- Lowercase is applied to all words.
- All punctuation has been eliminated.
- All single-character words (for example, 'a') are omitted.
- Delete all terms that contain numbers.

4.4 Caption Analysis

After preparing descriptions.txt, which has a clear description of every image, we can perform caption analysis. We have stored the filename of the image and its respective caption in the form of a dataframe using the pandas library. The result of this step is as follows:



	filename	captions
0	1	two girls are standing in garden
1	2	bird is sitting on tree branch
2	3	two girls are looking at bird
3	4	two girls are running towards their mother
4	5	the mother is instructing the two girls
...	...	...
14295	14296	two pink birds on tree
14296	14297	two pink birds on tree
14297	14298	two pink birds on tree
14298	14299	two pink birds on tree
14299	14300	two pink birds on tree

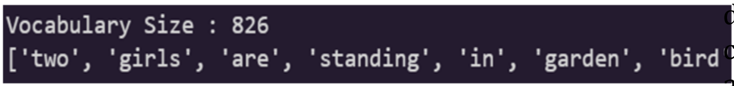
14300 rows × 2 columns

Fig -9: Description if Images

Caption analysis can give us information about:

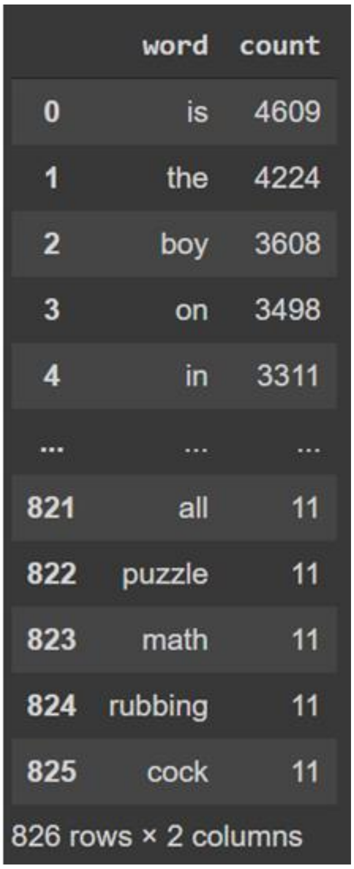
- Vocabulary size: This tells us about how many unique words are present to generate a new caption. It is implicit that the greater the size of this vocabulary, the finer is the caption generated.
- Count of unique words: This helps us to have an idea about the most common and the least common words that are occurring in our captions.

The results of these two things can be seen below:



```
Vocabulary Size : 826
['two', 'girls', 'are', 'standing', 'in', 'garden', 'bird
```

Fig -10: Vocabulary size for captions



	word	count
0	is	4609
1	the	4224
2	boy	3608
3	on	3498
4	in	3311
...	...	...
821	all	11
822	puzzle	11
823	math	11
824	rubbing	11
825	cock	11

826 rows × 2 columns

Fig -11: Frequency of each word

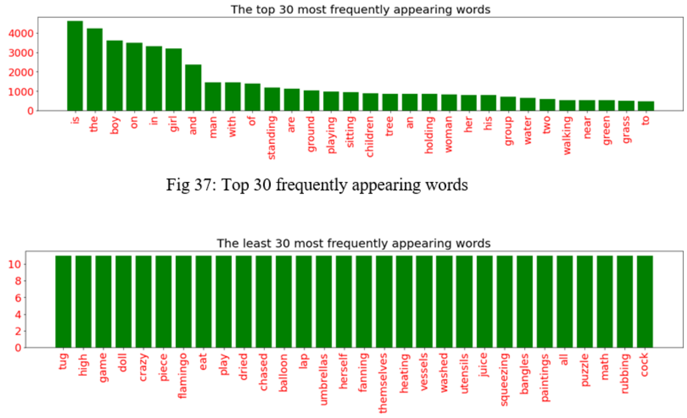


Fig 37: Top 30 frequently appearing words

Fig -12: Frequency of words plotted

4.5 Splitting the Dataset

Using the *train_test_split()* function, the dataset of 14300 images is first divided into training data and testing data. With 0.1 as the test_size ratio, at this stage, the count of testing images is 1430(10% of 14300). Again, using the same function, we now divide the training dataset into training images and development datasets. Therefore, with a test_size ratio of 0.25, the count of development images is 3218 (25% of 12870), and the count of training images is 9652 (75% of 12870). It is to be noted that the *random_state* is selected by the *randint* function, which chooses a random integer between 1 to 50.

4.6 Loading Training Images

We must load the arranged photo and text data that is to be used to fit into our model. We will include around 9652 images and captions in the training dataset to train our model. We'll monitor the model's output on the production dataset during training and use that data to decide when to save our models to disc. The model we'll build will generate a caption based on a snapshot, one word at a time. As input, a sequence of previously generated words will be given. As a result, we'll need a "first word" to start the generation process and a "last word" to finish our caption.

4.7 Encoding Descriptions

The description text will be encoded to numbers before it can be presented to the model as input or likened to our model's predictions. Each description will be split into words. The model will be given one word and the input photo. Using this data, the model will generate the next word. Then the first two words of the description will be given to the model as input with the photo to produce the next word. The model will be trained in this manner. We will encode the input text in the form of integers. These encodings will be given as input to a word embedding layer. We will feed the photo features directly to another part of the model. The output of the model will be a prediction, which will be in the form of a probability distribution that will cover each word belonging to our vocabulary. The output data will therefore be a one-hot encoded version of each word, representing an idealized probability distribution with 0 values at all word positions except the actual word position, which has a value of 1.

4.8 Defining the model:

The Photo Feature Extractor model assumes input photo features to be a vector of 4,096 elements. These are handled by a Dense layer to produce a 256-element representation of the photo. The Sequence Processor model assumes input sequences with a pre-defined length that are fed into an Embedding layer that uses a mask. This mask ignores all the padded values. After that, there is a 256-unit LSTM layer present. The output of the two input models is a 256-element vector. Additionally, both input models use regularization. This is realized by having a 50% dropout. This is to reduce the overfitting of the training dataset. The vectors from the two models are combined in the Decoder model. A Dense 256 neuron layer receives this output. The output of this dense layer is given to a final output layer, which allows a SoftMax prediction for the next word in the series over the entire output vocabulary.

4.9 Fitting the Model

We then fit the model into our custom dataset. After the entire training process ends, we can utilize the saved model with the best stats on our training data set as our final image captioning model. This can be accomplished by allocating a ModelCheckpoint in Keras to oversee the least loss on our validation results.

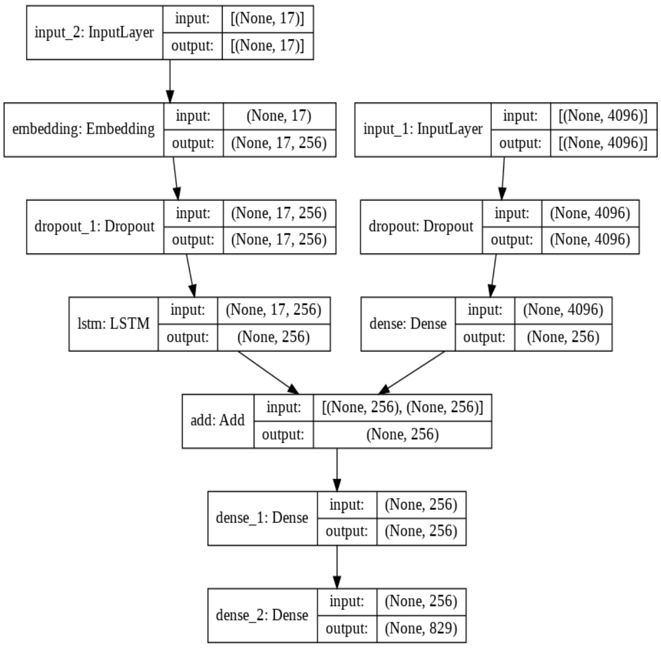
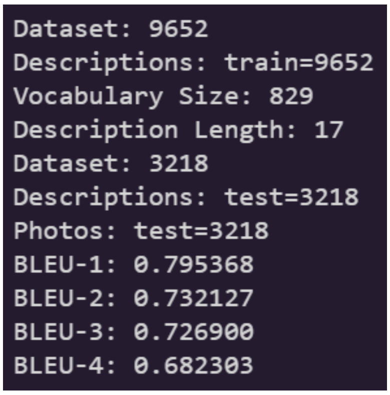


Fig -13: Model summary

4.10 Evaluating the Model

We will evaluate the accuracy and efficacy of the model's predictions on our test data after it has been fitted. We'll test a model by creating captions for all of the images in the test dataset and using a "normal cost feature" to determine the accuracy of those predictions. The following BLEU scores were observed:



```
Dataset: 9652
Descriptions: train=9652
Vocabulary Size: 829
Description Length: 17
Dataset: 3218
Descriptions: test=3218
Photos: test=3218
BLEU-1: 0.795368
BLEU-2: 0.732127
BLEU-3: 0.726900
BLEU-4: 0.682303
```

Fig -14: Work Flow of Project

4.11 Generating the Captions

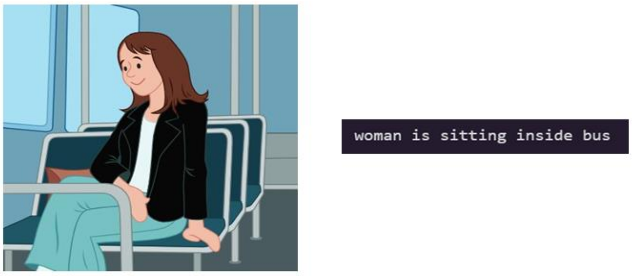
The model file contains almost everything we need to produce captions for new photographs. The features can be fed into our current model as inputs. A picture caption is created with the help of a predefined function. After generating the final caption, we remove the words ‘startseq’ and ‘endseq’ from the sentence as they don’t add any particular meaning to the sentence.

4.12 Generating an Audio file in different languages

Using *gTTs*, we can see the audio file generated in our working directory. The python googletrans module can be used to translate the image captions to various languages. We have converted the caption generated in English to the following local languages: Hindi, Marathi, Malayalam, Tamil, Gujrati, Bengali, Kannada, Telugu.

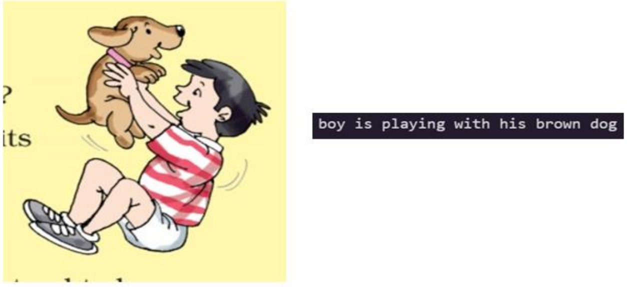
5. RESULTS

The image captioning model gives us a relatively accurate caption. Following are some results for the input images shown below.



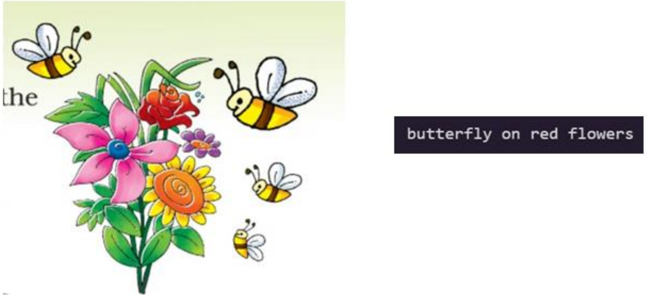
```
woman is sitting inside bus
```

Fig -15: Model result on a test image



```
boy is playing with his brown dog
```

Fig -16: Model result on a test image



```
butterfly on red flowers
```

Fig -17: Model result on a test image

After putting everything together, the project works in the flow mentioned below and gives the following results:

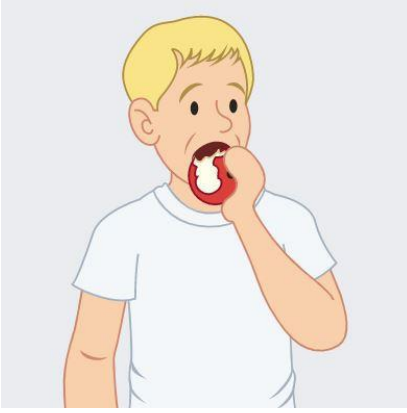
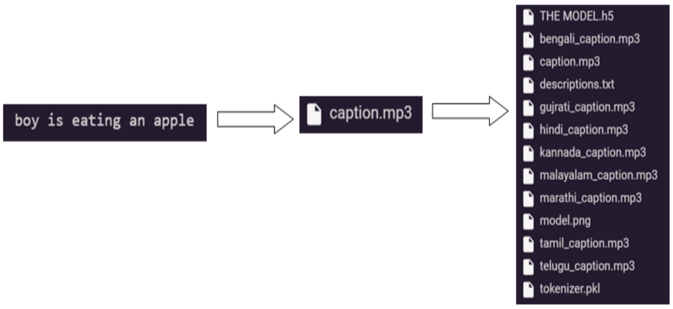


Fig -18: Test image



```
boy is eating an apple
```

Fig -19: Work Flow of Project

On plotting the graphs for model accuracy and model loss, we got the following outputs:

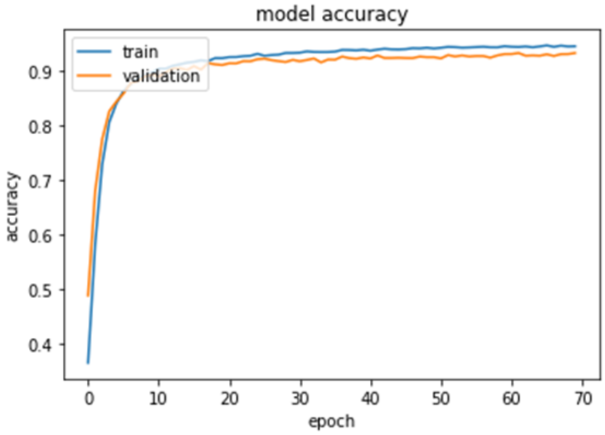


Fig -20: Plot of model accuracy

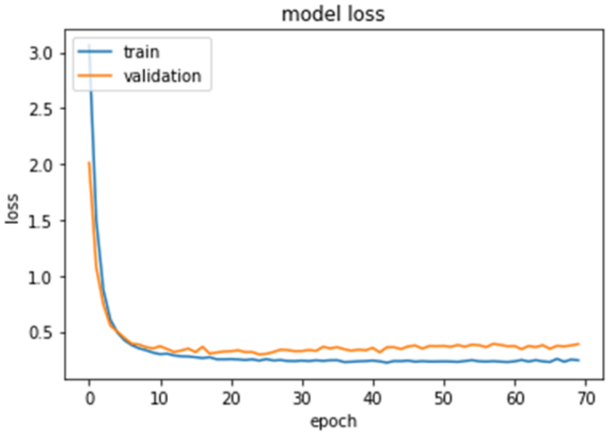


Fig -21: Plot of model loss

6. CONCLUSIONS

Deep Learning has been extensively used to solve real-world problems and has proven to be efficacious most of the time. Thereby putting this technology to use in our project, we used the widely known image caption generating model as the base for our project.

At this junction, we have made considerable progress and the captions are mostly accurate. The captions obtained were then converted into an audio format in different languages like Hindi, Marathi, Tamil, etc., which can serve well to visually impaired students. Generating audio captions from images will prove useful when it comes to converting textbooks and other educational materials into audio format.

This will give visually impaired students access to the general material used, and there will be no compromise in the study materials for them. This will help provide an enhanced experience for these students.

In conclusion, we hope to see this tool to be a useful educational tool and an aid to study for visually impaired children.

7. FUTURE SCOPE

In the coming future we hope to:

- Fine-tuning the model even more by possibly increasing the data set to a greater size.
- Applying some higher-level algorithms and mechanisms to the model for generating better results.
- Running and testing our project on real students and taking into account the reviews.
- Creating a user-friendly implementation of this project, like a website or a mobile application.
- Researching the possibilities of expanding the application of our project in various other similar problems.

REFERENCES

[1] https://neurohive.io/en/popular-networks/vgg16/

[2] https://medium.com/@purnasaigudikandula/recurrent-neural-networks-and-lstm-explained-7f51c7f6bbb9

[3] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[4] https://www.geeksforgeeks.org/convert-text-speech-python/

[5] https://arxiv.org/abs/1906.05963

[6] http://sersc.org/journals/index.php/IJAST/article/view/5927

[7] https://www.researchgate.net/publication/332662087_Review_of_Deep_Learning_Algorithms_and_Architectures

[8] https://www.mdpi.com/2076-3417/9/10/2024

[9] https://arxiv.org/pdf/1810.04020.pdf

[10] https://arxiv.org/abs/1909.11059