**Implementation**

**Dataset**

The dataset used is Fliker8k dataset. It consists of 8091 images of various sizes. These all images in .jpg format. There are around 6,000 images for training, 1,000 images for developing and another 1,000 images for testing. In the dataset there are two folders in the first folder there are images and in another folder, there are text files.Contains text files describing train\_set ,test\_set. Flickr8k.token.txt contains 5 captions for each image i.e. total 40460 captions[2].

**Implementation flow**

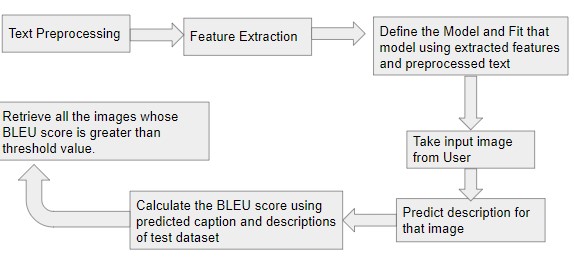


Fig 4. Implementation flow

As shown in Fig. 4 first we will pre-process the text. In the original dataset, we have 5 captions for each image. So we will clean those captions and make a text file to store those captions. Now we have an image id and its caption. After that, we have used ResNet50, which is provided by Keras API. As we are using this for feature extraction, so we will remove the last layer i.e softmax layer. Here in input will pass image and model will return a vector of (1,1000) shape[1].

For every training image, we are resizing it to (224,224) and then passing it to ResNet50 for feature extraction[1].

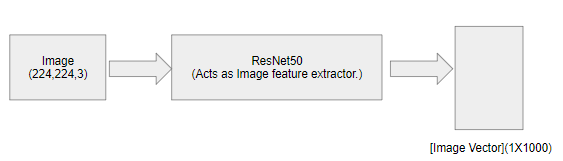


Fig 5. Feature extractor

After extracting features of all images we will store those features in pickle file for future use. Then we will create the model using the Model API. And train that model using the preprocessed text and extracted features. Here we will break the sequence into two parts for different lengths and those two parts are input sequence and output sequence. So our model is trained to predict the next word of sequence for the given input image and input sequence. In this way, the model will generate the whole caption for the given image.

To train the model in the above-described way we have to encode all the captions in the following format:

* “startseq “ + caption + “ endseq”

The reason behind startseq and endseq is,

* Startseq - This will act as our first word when the feature extracted image vector is fed to decoder. It will kick-start the caption generation process[2].
* Endseq - This will tell the decoder when to stop. We will stop predicting words as soon as endseq appears or we have predicted all words from the trained dictionary whichever comes first[2].

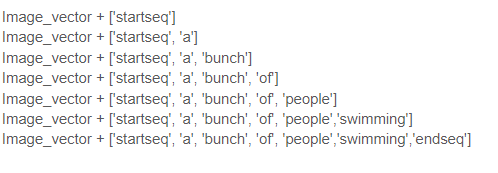


Fig 6. Example of captioning

After completing the training of the model, we will give an image as input and generate the caption for that image by iteratively predicting the next word for the given input sequence. Now we will compare that predicted caption with the captions of the test dataset and keep those images, whose BLEU score of the caption is greater than the threshold. After that, we will sort matched images in decreasing order of BLEU score. And output those images to the user.

**Model proposed**

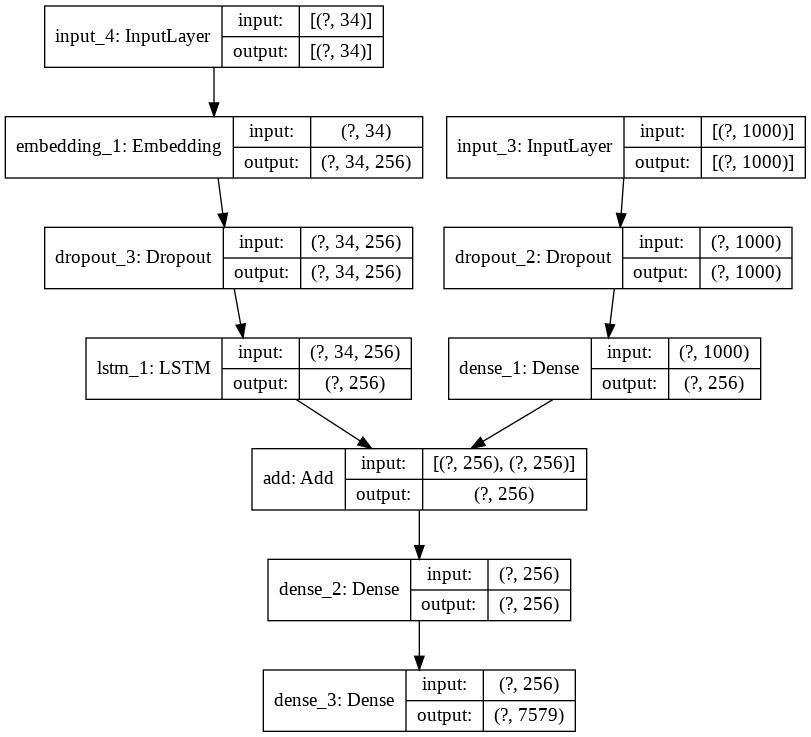


Fig 7. Proposed Model

**Result**

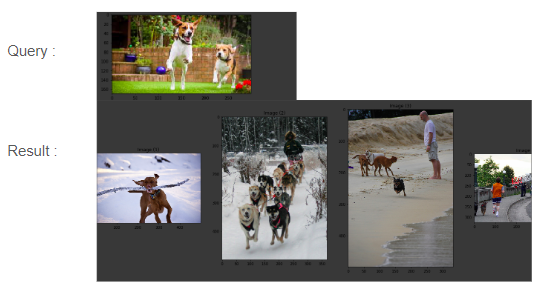


Fig 8. Result

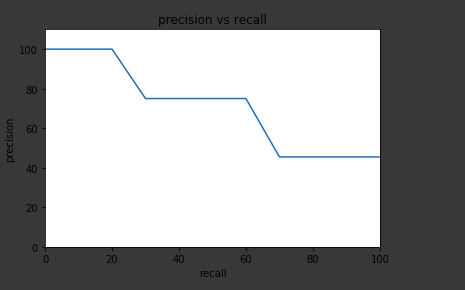
****

Fig9.System precision versus recall curve

The Fig9. shows the system precision versus recall curve for the dog image query. The curve is called based on the 11 precision method[12].

**References**