**APPROACH:**

The image-text similarity is achieved in two steps. During the first step a caption is generated for the given input image. In the next step, a similarity score is computed and displayed by comparing the generated caption with the human description (input sentence) using cosine similarity.

Caption generation is a challenging artificial intelligence problem where a textual description is generated for a given image. Generating a caption needs an image and caption dataset, preparing photo and text data, develop a deep learning model and training it on the dataset. Finally, the model is used to predict captions for the new images.

This work uses Flickr8K dataset (Contains 8092 photographs in JPEG format) and Flickr8k\_text (Contains a number of files containing different sources of descriptions for the photographs). The dataset has a pre-defined training dataset (6,000 images), development dataset (1,000 images), and test dataset (1,000 images). During the preparation of the photo data the conent of the photos is interpreted using a pre-trained model. In this work, InceptionV3 is used from Keras library. The “photo features” using the pre-trained model are pre-computed which can be saved/loaded to/from a file. The final layers are removed from the loaded model, as this is the model used to predict a classification for a photo. We are not interested in classifying images, but we are interested in the internal representation of the photo right before a classification is made. These are the “features” that the model has extracted from the photo. The dataset contains multiple descriptions for each photograph and the text of the descriptions requires some minimal cleaning. We will Convert all words to lowercase, remove all punctuation., remove all words that are one character or less in length and remove all words with numbers in them. Finally, the dictionary of image identifiers and descriptions are saved to a new file named descriptions.txt. The prepared photo and text data is loaded so that it can be used to fit the model.

The deep learning model has three main parts. A photo feature extractor which is a InceptionV3 pretrained on Imagenet dataset. The features extracted on all of the photos is used as an input. Further, the model has a sequence processor and decoder. The sequence processor is a word embedding layer for handling the text input, followed by a Long Short-Term Memory (LSTM) recurrent neural network layer. The feature extractor and sequence processor output a fixed-length vector. These are merged together and processed by a Dense layer to make a final prediction.

The Photo Feature Extractor model expects input photo features to be a vector of 2048 elements. These are processed by a Dense layer to produce a 256 element representation of the photo.

The Sequence Processor model expects input sequences with a pre-defined length (34 words) which are fed into an Embedding layer that uses a mask to ignore padded values. This is followed by an LSTM layer with 256 memory units.

Both the input models produce a 256 element vector. Further, both input models use regularization in the form of 50% dropout. This is to reduce overfitting the training dataset, as this model configuration learns very fast.

The Decoder model merges the vectors from both input models using an addition operation. This is then fed to a Dense 256 neuron layer and then to a final output Dense layer that makes a softmax prediction over the entire output vocabulary for the next word in the sequence.

The model is trained for 30 epochs using cross entropy loss with adam optimizer. As this is the first cut solution, a model is generated and tested on sample images qualitatively. The generated output is shown in the notebook. The model performance can be further tuned and tested on validation and testing dataset respectively. The corpus BLEU score can be used to find out how close the actual and predicted descriptions are close to each other.

The model is used to generate the captions for a new image and the generated caption is compared with the input sentence using cosine similarity.

**FUTURE WORK:**

* Using a bigger dataset to train the model.
* Using a better pre-processing.
* Using a different pre-trained model for extracting features from images or training a model from scratch.
* Using attention models.
* Using BLEU Score to evaluate and measure the performance of the model on validation and test set.
* Using Beam search instead of greedy search while making predictions.
* Extensive hyper-parameter tuning.
* Use better metrics to obtain the similarity score.