SMART INTERNZ

AI EXTERNSHIP

TEAM - 522

PROJECT TITLE- IMAGE CAPTION GENERATOR

SANTHOSH NARAYANAN

20BCE1309

VIT CHENNAI

TATHAGATA BISWAS

20BCE1844

VIT CHENNAI

NARASAMBATTU SRI SISHVIK

20BCE1735

VIT CHENNAI

KARTHEEK ANUSURI

20BAI10344

VIT BHOPAL

1)Introduction

Extraction of relevant information from photos is becoming more and more crucial in the age of voluminous visual content. This problem is solved by picture captioning, which enables computers to produce textual descriptions of an image's content and context. The goal of the project is to create an image caption generator that can analyze visual data, comprehend its nuances, and provide captions that offer detailed and extensive explanations. The project takes advantage of CNNs' capacity to extract sophisticated visual information from pictures by integrating CNNs, which are excellent at processing visual data. These properties are used as input by LSTM networks, which can identify sequential relationships and produce textual descriptions that are logical and contextually appropriate. The model can successfully transform visual input into insightful and evocative subtitles because of this combination.

The goal of this project has three different objectives. By utilizing deep learning techniques, it tries to improve the comprehension and interpretation of visual input. The initiative enables machines to bridge the gap between the visual and language domains, leading to a more thorough comprehension of pictures by extracting visual elements and fusing them with written information. The second goal of the project is to automate caption generation, doing away with the necessity for manual annotation or description. With the help of automation, a lot of visual material may be efficiently tagged and explained while also saving time and effort. The final goal, the picture caption generator, also has useful applications in a variety of fields. The automatic creation of captions may improve user experience, increase content discoverability, and enable accessibility for those who are blind on everything from social networking platforms to e-commerce websites. The idea can also be applied in areas where descriptive captions might improve comprehension and communication, such as content development, digital media archives, autonomous cars, and educational settings.

The difficulty of comprehending visual material and producing insightful captions is generally addressed by the effort on developing an image caption generator with CNN and LSTM. The project seeks to contribute to the field of multimedia comprehension and ease the interpretation and exchange of visual information with its practical applications and potential to improve numerous sectors.

2)Literature Survey

Existing Problem:

Failure to Accurately Capture Image-Text Relationships: Image captioning algorithms may encounter difficulties in correctly capturing the connections between the content of the images and the accompanying textual descriptions. As a result, the captions may be grammatically inaccurate or have nothing to do with the picture.

Restricted Vocabulary and Word Ambiguity: Captioning models may have a restricted vocabulary, which makes it challenging to produce a variety of context-relevant captions. Word ambiguity, in which a single word can have several meanings depending on the context, can also be a problem.

Criteria for Evaluation: Choosing the right criteria for evaluation while captioning images might be difficult. The quality and relevance of the generated captions might not be fully captured by conventional measures like BLEU or METEOR.

Insufficient Training Data: The model's ability to generalize well and provide correct captions can be hampered by incomplete or unbalanced training data.

Proposed Solution:

LSTM (Long Short Term Memory) and CNN (Convolutional Neural Networks) will be used to create the caption generator. The picture features will be taken from the CNN model Xception, which was trained on the imagenet dataset, and fed into the LSTM model, which will provide the image captions.

3)Theoretical Analysis

3.1. Block Diagrams:

Working of Deep CNN:

Convolutional Neural networks are specialized deep neural networks which can process the data that has input shape like a 2D matrix. Images are easily represented as a 2D matrix and CNN is very useful in working with images. CNN is basically used for image classifications and identifying if an image is a bird, a plane or Superman, etc.

It scans images from left to right and top to bottom to pull out important features from the image and combines the features to classify images. \ The network employs a special mathematical operation called a "convolution" instead of matrix multiplication. The architecture of a convolutional network typically consists of four types of layers: convolution, pooling, activation, and fully connected.

Convolutional Layer:

Applies a convolution filter to the image to detect features of the image. Here is how this process works:

A convolution—takes a set of weights and multiplies them with inputs from the neural network.

Kernels or filters—during the multiplication process, a kernel (applied for 2D arrays of weights) or a filter (applied for 3D structures) passes over an image multiple times.

ReLU Activation Layer:

The convolution maps are passed through a nonlinear activation layer, such as Rectified Linear Unit (ReLu), which replaces negative numbers of the filtered images with zeros.

Pooling Layer:

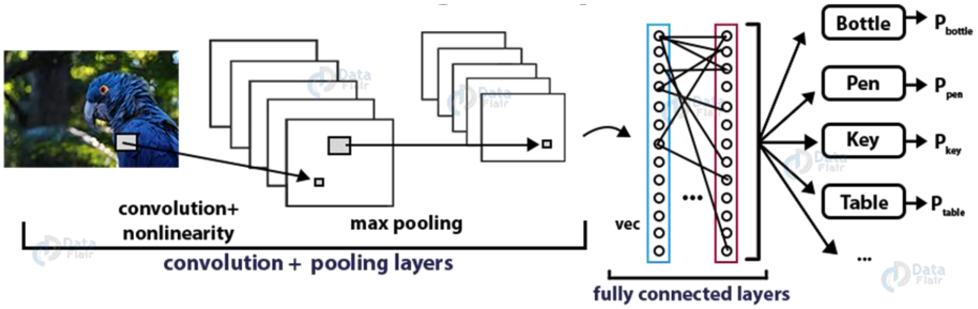
The pooling layers gradually reduce the size of the image, keeping only the most important information. For example, for each group of 4 pixels, the pixel having the maximum value is retained (this is called max pooling), or only the average is retained (average pooling).

Pooling layers help control overfitting by reducing the number of calculations and parameters in the network.

After several iterations of convolution and pooling layers (in some deep convolutional neural network architectures this may happen thousands of times), at the end of the network there is a traditional multi layer perceptron or "fully connected" neural network.

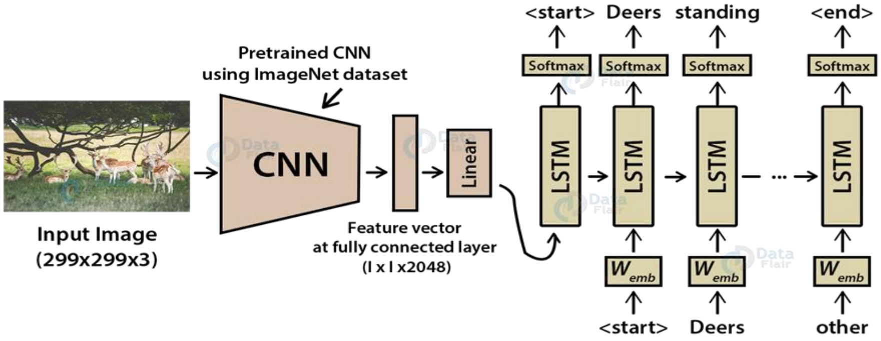
Fully Connected Layer:

In many CNN architectures, there are multiple fully connected layers, with activation and pooling layers in between them. Fully connected layers receive an input vector containing the flattened pixels of the image, which have been filtered, corrected and reduced by convolution and pooling layers. The softmax function is applied at the end to the outputs of the fully connected layers, giving the probability of a class the image belongs to – for example, is it a car, a boat or an airplane.



LSTM Cell Structure: LSTM stands for Long short term memory, they are a type of RNN (recurrent neural network) which is well suited for sequence prediction problems. Based on the previous text, we can predict what the next word will be. It has proven itself effective from the traditional RNN by overcoming the limitations of RNN which had short term memory. LSTM can carry out relevant information throughout the processing of inputs and with a forget gate, it discards non-relevant information They are composed out of a sigmoid neural net layer and a pointwise multiplication operation. The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means "let nothing through," while a value of one means "let everything through!". An LSTM has three of these gates, to protect and control the cell state.

3.2. Hardware / Software designing (Model):



To make our image caption generator model, we will be merging these architectures. It is also called a CNN-RNN model.

- CNN is used for extracting features from the image. We will use the pre-trained model Xception.
- LSTM will use the information from CNN to help generate a description of the image.

We will define 5 functions:

- load_doc(filename) – For loading the document file and reading the contents inside the file into a string.
- all_img_captions(filename) – This function will create a description dictionary that maps images with a list of 5 captions.
- cleaning_text(descriptions) – This function takes all descriptions and performs data cleaning. This is an important step when we work with textual data, according to our goal, we decide what type of cleaning we want to perform on the text. In our case, we will be removing punctuations, converting all text to lowercase and removing words that contain numbers.
  So, a caption like "A man riding on a three-wheeled wheelchair" will be transformed into "man riding on three wheeled wheelchair"
- text_vocabulary(descriptions) – This is a simple function that will separate all the unique words and create the vocabulary from all the descriptions.
- save_descriptions(descriptions, filename) – This function will create a list of all the descriptions that have been preprocessed and store them into a file. We will create a descriptions.txt file to store all the captions.

model = Xception(include_top=False, pooling='avg')

The function extract_features() will extract features for all images and we will map image names with their respective feature array.

For loading the training dataset, we need more functions:

- load_photos(filename) – This will load the text file in a string and will return the list of image names.
- load_clean_descriptions(filename, photos) – This function will create a dictionary that contains captions for each photo from the list of photos. We also append the <start> and <end> identifier for each caption. We need this so that our LSTM model can identify the starting and ending of the caption.
- load_features(photos) – This function will give us the dictionary for image names and their feature vector which we have previously extracted from the Xception model.

Computers don't understand English words, for computers, we will have to represent them with numbers. We will map each word of the vocabulary with a unique index value. Keras library provides us with the tokenizer function that we will use to create tokens from our vocabulary and save them to a "tokenizer.p" pickle file.

We calculate the maximum length of the descriptions. This is important for deciding the model structure parameters.

To define the structure of the model, we will be using the Keras Model from Functional API. It will consist of three major parts:

- Feature Extractor – The feature extracted from the image has a size of 2048, with a dense layer, we will reduce the dimensions to 256 nodes.
- Sequence Processor – An embedding layer will handle the textual input, followed by the LSTM layer.
- Decoder – By merging the output from the above two layers, we will process the dense layer to make the final prediction. The final layer will contain the number of nodes equal to our vocabulary size

4)Experimental Investigations:

Analysis or the investigation made while working on the solution

The technique is also called transfer learning, we are using the Xception model which has been trained on imagenet dataset that has 1000 different classes to classify. We can directly import this model from the keras.applications . One thing to notice is that the Xception model takes 299*299*3 image size as input. We will remove the last classification layer and get the 2048 feature vector.

model = Xception(include_top=False, pooling='avg')

The function extract_features() will extract features for all images and we will map image names with their respective feature array.

For loading the training dataset, we need more functions:

- load_photos(filename) – This will load the text file in a string and will return the list of image names.
- load_clean_descriptions(filename, photos) – This function will create a dictionary that contains captions for each photo from the list of photos. We also append the <start> and <end> identifier for each caption. We need this so that our LSTM model can identify the starting and ending of the caption.
- load_features(photos) – This function will give us the dictionary for image names and their feature vector which we have previously extracted from the Xception model.

Computers don't understand English words, for computers, we will have to represent them with numbers. So, we will map each word of the vocabulary with a unique index value. Keras library provides us with the tokenizer function that we will use to create tokens from our vocabulary and save them to a "tokenizer.p" pickle file.

Our vocabulary contains 7577 words.

We calculate the maximum length of the descriptions. This is important for deciding the model structure parameters. Max_length of description is 32.

To make this task into a supervised learning task, we have to provide input and output to the model for training. We have to train our model on 6000 images and each image will contain a 2048 length feature vector and the caption is also represented as numbers. This amount of data for 6000 images is not possible to hold into memory so we will be using a generator method that will yield batches.

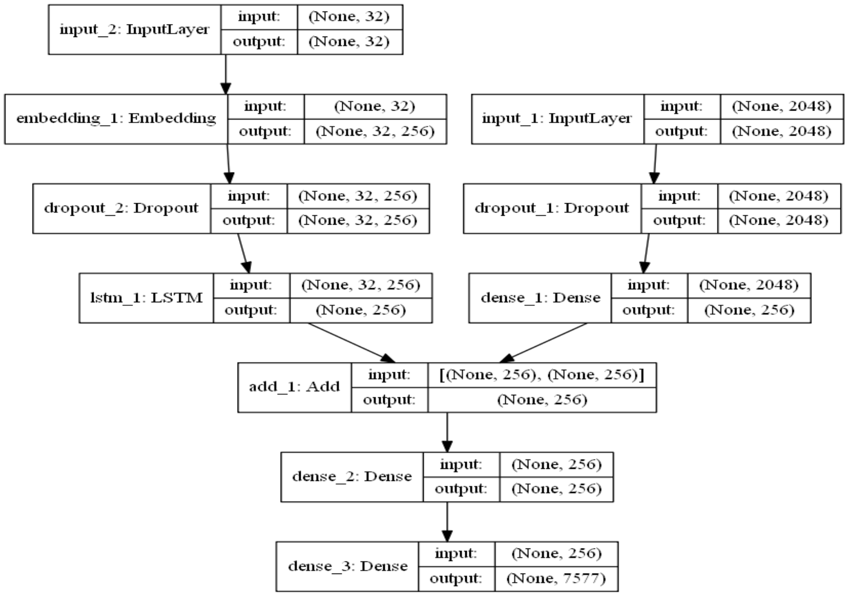
The generator will yield the input and output sequence.

For example:

The input to our model is [x1, x2] and the output will be y, where x1 is the 2048 feature vector of that image, x2 is the input text sequence and y is the output text sequence that the model has to predict.

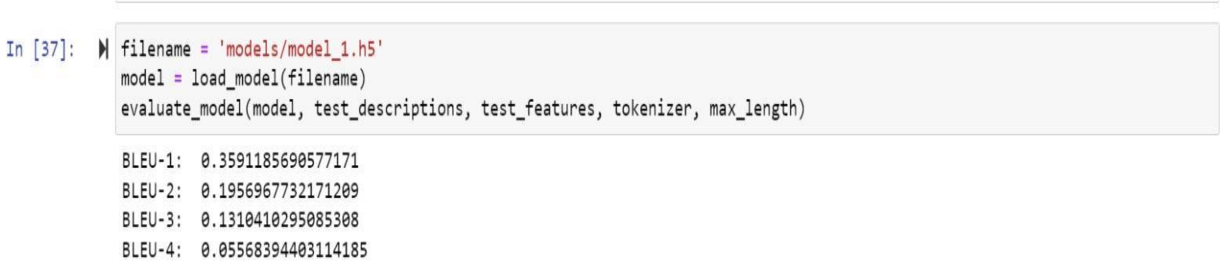
x1(feature vector)	x2(Text sequence)	y(word to predict)
feature	start,	two
feature	start, two	dogs
feature	start, two, dogs	drink
feature	start, two, dogs, drink	water
feature	start, two, dogs, drink, water	end

5) Flowchart:



6)Result:

The BLEU score of the model after the iterations.

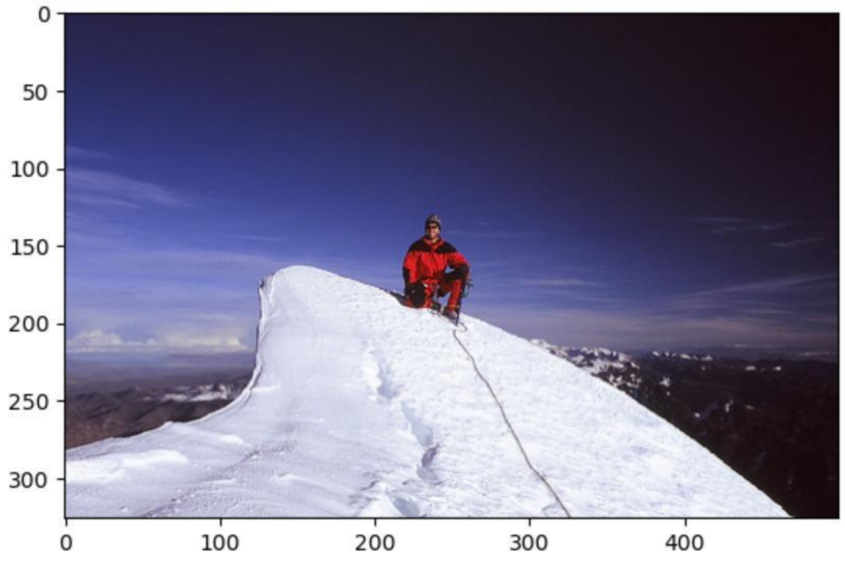


```
In [37]:  filename = 'models/model_1.h5'
          model = load_model(filename)
          evaluate_model(model, test_descriptions, test_features, tokenizer, max_length)

          BLEU-1:  0.3591185690577171
          BLEU-2:  0.1956967732171209
          BLEU-3:  0.1310410295085308
          BLEU-4:  0.05568394403114185
```

The Image and Caption

```
        start man in red shirt is walking through the snow end
Out[18]: <matplotlib.image.AxesImage at 0x253a64096f0>
```



Website

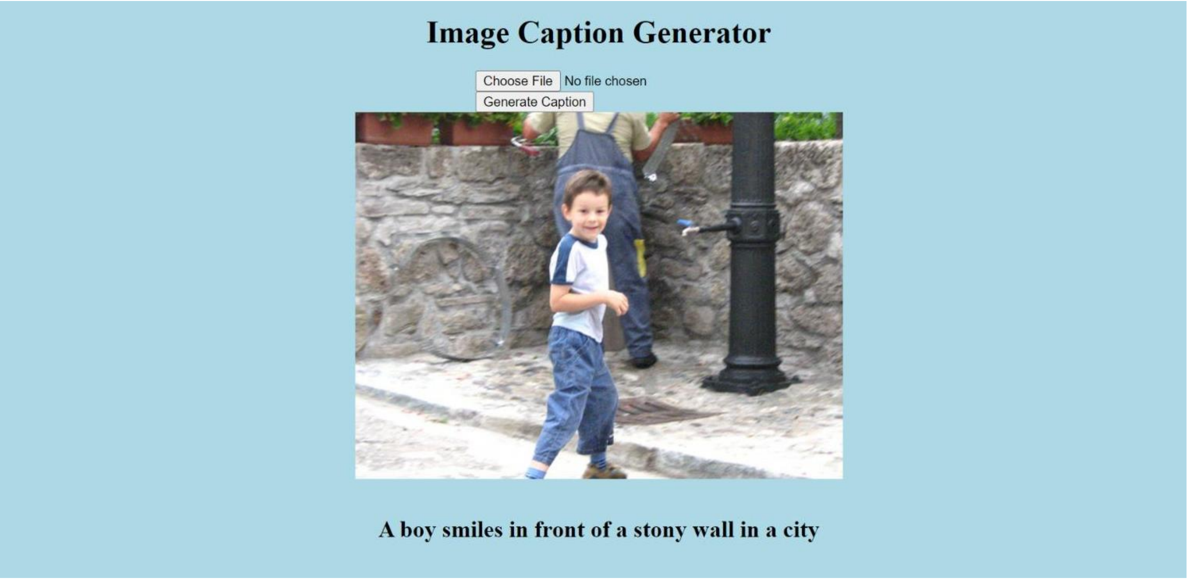


Image Caption Generator

A boy smiles in front of a stony wall in a city

7)Advantages and Disadvantages:

The advantages and disadvantages of the proposed solution of building an Image Caption Generating model with the CNN and LSTM process are furnished below:

Advantages:

- Improved image understanding: The model can efficiently extract visual information from pictures using the CNN component and provide contextually appropriate captions using the LSTM component. As a result, it is possible to create captions that are correct and a better comprehension of the image's content.
- Contextual and coherent captions: LSTM networks are excellent at identifying sequential relationships and producing content that is both coherent and contextually appropriate. In comparison to more straightforward methods, this enables the picture caption generator to generate captions that are more detailed and relevant.
- Ability to handle complex images: The capacity of CNNs to handle complicated and multidimensional visual input is well established. The image caption generator can successfully extract complex visual elements from photos by utilizing CNNs, which enables it to produce captions that accurately represent the specifics of the visual material.
- Transfer learning from pre-trained CNNs: Starting points include pre-trained CNN models like VGG, ResNet, or Inception. Even when training on a different problem, transfer learning enables the model to profit from the understanding and acquired representations of the pre-trained CNNs. Significant training time and computing resources may be saved in this way.

Disadvantages:

- Complex model architecture: Implementing and training CNN and LSTM architectures can be challenging. A solid understanding of deep learning ideas is necessary to comprehend the complexities of both systems and integrate them successfully. For newcomers like us or those not familiar with these architectures, it may provide difficulties.
- Data requirements and annotations: An enormous quantity of annotated data, including pictures and the words that go with them, is often needed to train an

image caption generator. Such datasets can be time- and resource-intensive to create or acquire. Additionally, human bias and inaccuracy may be introduced by manually annotating the data.

- Difficulty in evaluating caption quality: It might be difficult to judge how well produced captions are done. Although there are well-established assessment criteria like BLEU, CIDEr, and ROUGE, they might not be able to capture all aspects of caption quality, such inventiveness or the ability to catch finer details in the picture content. Accurately determining the best assessment criteria and rating the model's performance might be difficult.
- Limited control over caption generation: Although CNN and LSTM models are capable of producing meaningful captions, they might not have precise control over the final product. It may be difficult to direct the model to produce captions that comply with particular specifications or limitations, such as producing captions in a particular style or concentrating on a specific set of picture properties.
- Handling of ambiguous or abstract images: It might be difficult for the model to caption visuals that are unclear or abstract. Comparatively to photographs with obvious and recognisable items, it could be harder to come up with correct and insightful descriptions for such images. Such situations call for extra approaches or methods, such as the use of more sophisticated model structures or attention processes.

When selecting whether to use CNN and LSTM to create an image caption generator, it's crucial to weigh the pros and downsides. The method has a lot to offer in terms of comprehending images and producing captions that make sense, but it also has limitations in terms of model complexity, data needs, computer resources, and assessment.

8)Applications:

The method of using CNN and LSTM to create an image caption generator has many uses in a variety of fields. Here are several situations in which this remedy may be used:

Social media: On social media sites, image captioning may increase accessibility and user engagement. Social media platforms may enhance content discoverability, offer accessibility for users who are blind, and enhance user experience by automatically creating captions for user-uploaded photographs.

E-commerce: Product descriptions may be generated automatically on online shopping platforms using image captioning. As a result, product descriptions are made more thorough and comprehensive, searchability is improved, and users' overall buying experiences are enhanced.

Content creation: Bloggers, journalists, and photographers that create content can use picture caption generating to make their process more efficient. By automatically producing captions for their photographs, they may focus on other parts of content development rather than manually describing and labeling each image.

Digital media archives: Large digital media collections may be searched and organized more easily with the help of image captioning. By enabling effective keyword-based searches and classification, automatic caption generation for photographs might make it simpler to access certain images from the archive.

Autonomous vehicles: Autonomous cars can employ image caption generation to produce textual descriptions of their surroundings. The utilization of this information can enhance situational awareness, increase safety, and help passengers—especially those who are blind or visually impaired—understand their environment.

Visual aids for the visually impaired: The creation of visual aids for the blind and visually handicapped can benefit from image captioning. Visually challenged people can access visual material and better comprehend the visual world through written explanations by creating captions for photos.

Content recommendation: In order to increase the precision and relevance of suggestions, image captioning can be employed in content recommendation systems. Recommendation systems can provide consumers more individualized and contextually relevant suggestions by comprehending the content of photos.

News and media organizations: Automatically creating captions for photographs used in articles, reports, or news stories may be made easier by image captioning for news and media organizations. This can increase the content's readability and accessibility and provide readers more context.

Education and training: To give interactive and descriptive comments for instructional pictures, diagrams, or visual aids, image captioning can be used in educational contexts. This improves students' accessibility and learning experience.

Medical imaging: The automatic production of captions or explanations for medical pictures like X-rays, MRI scans, or histopathological images can be used in the area of medicine. This can make it easier for medical practitioners to comprehend and interpret medical pictures.

These are only a few examples of the various uses for CNN and LSTM-generated picture captions. The approach may be modified and used in several different fields where visual material analysis, comprehension, and communication are important.

9)Conclusion:

This project is an intriguing and useful project that combines the power of computer vision with natural language processing by building an image caption generator utilizing CNN and LSTM. This attempts to close the gap between visual comprehension and language creation by using Convolutional Neural Networks (CNNs) to extract visual information from images and Long Short-Term Memory (LSTM) networks to generate informative captions.

The benefits of this project include enhanced visual comprehension, the capacity to handle complicated images, and the ability to produce contextually appropriate and logical captions. Transfer learning is made possible by using pre-trained CNN models, which saves time and computing resources. The picture caption generator may be developed and implemented in an optimal setting using Python because of its strong libraries and frameworks.

However, there are also challenges and limitations to consider. The complex model architecture and computational requirements may pose obstacles, and the availability of annotated datasets could be a bottleneck. Evaluating the quality of generated captions and achieving fine-grained control over the caption generation process can be challenging. Additionally, the handling of ambiguous or abstract images may require further strategies.

Despite these challenges, the project holds immense potential and can be applied across various domains. It finds applications in social media, e-commerce, content creation, autonomous vehicles, digital media archives, and many more areas. It offers opportunities for enhancing user experience, improving accessibility, streamlining workflows, and advancing the fields of computer vision and natural language processing.

Undertaking this project provides valuable knowledge and practical experience in deep learning, transfer learning, data preprocessing, model evaluation, and

interdisciplinary collaboration. It contributes to the development of multimedia understanding and showcases the capabilities of deep learning in extracting meaningful information from visual data.
In summary, the project on building an image caption generator with CNN and LSTM offers a promising avenue for exploring the fusion of computer vision and natural language processing, with real-world applications and the potential to make significant contributions to various domains.

10)Future Scope:

The development of an image caption generator using CNN and LSTM offers an interesting future scope with opportunities for more research and development. Here are some potential developments and future prospects for the project:

Advanced model architectures: Researchers can investigate more complex models that include multimodal architectures, transformer-based models, or attention processes. The quality and accuracy of the produced captions can be improved by using attention techniques to guide the model's attention to pertinent picture areas. Transformer-based models can better the fluency and coherence of the output captions by capturing long-range relationships. Better alignment between visual characteristics and produced captions may be achieved by fusing picture and text modalities using multimodal architectures.

Enhanced evaluation metrics: Future study may find it useful to focus on creating enhanced assessment measures that are targeted particularly for picture captioning. It may be possible to measure caption quality more accurately and compare models with more accuracy by developing new metrics that are more in line with human judgment and take semantic comprehension into account.

Incorporating commonsense reasoning: The quality and comprehension of the generated captions may be enhanced by adding common sense reasoning skills to the picture caption generator. This entails teaching the model to draw conclusions from common knowledge, extrapolate from the available data, and provide captions that are consistent with the context of the picture shown in the image.

Domain-specific image captioning: The picture caption generator may be tailored to certain areas or specialized applications to create new opportunities. For instance, creating picture captioning models for specific areas like medical imaging, fashion, or architecture might result in descriptions that are more precise and specifically suited to those fields.

User feedback and iterative improvement: The picture caption generator may be improved over time by putting in place methods for user input and iterative development. To iteratively enhance the model and boost performance based on actual user experiences, it may be utilized to let users score and give feedback on the generated captions.

These directions for the future show the possibility for more development and creativity in the area of creating picture captions using CNN and LSTM. Pushing the frontiers of computer vision, natural language processing, and multimodal comprehension, exploring these areas can result in caption production that is more precise, inventive, and contextually relevant.

11)Bibliography :

References of previous works or websites visited/books referred for analysis about the project, solution previous findings etc:

1.https://data-flair.training/blogs/python-based-project-image-caption-generator-cnn/
2.https://github.com/MiteshPuthran/Image-Caption-Generator
3.https://github.com/b01902041/Deep-Virtual-Try-on-with-Clothes-Transform#results
4.https://github.com/akshatchaturvedi28/Image-Caption-Generator-with-GUI/blob/main/HOW%20TO%20RUN.txt
5.https://blog.paperspace.com/deploying-deep-learning-models-flask-web-python/

Appendix:

GitHubLink- https://github.com/Santhosh-1917/Image_Caption_Generator_AI