Operating System Implementation of an LSTM-based Tensorflow Mechanism for Generating Image Captions

A PROJECT REPORT

CSA0407-Operating Systems for Unified Endpoint Management -

Submitted to

SAVEETHA INSTITUTE OF MEDICAL AND

TECHNICAL SCIENCES

In partial fulfillment for the award of the degree of

By

1. VAIBHAV (192211869)
2. KESHAV REDDY (192210248)

PS.ARJUN(192111263)

P.SURYA HAMSA VARDHAN(192124008)

Supervisor

Dr.**Terrance Frederick**



SAVEETHA SCHOOL OF ENGINEERING, SIMATS

CHENNAI-602105

MARCH-2024

**ABSTRACT:**

This paper presents the design and implementation of an operating system-based approach for deploying an LSTM-based Tensorflow mechanism aimed at generating captions for images. Leveraging the capabilities of modern operating systems, particularly in handling resource allocation and scheduling, enhances the efficiency and scalability of the caption generation process. The proposed system harnesses the power of LSTM networks within the Tensorflow framework to produce coherent and contextually relevant captions for a wide range of images.

The picture highlights will be separated from Xception which is a CNN model prepared on the imagenet dataset and afterward we feed the highlights into the LSTM model which will be answerable for creating the picture subtitles.Generating descriptive captions for images is a challenging task in computer vision and natural language processing. In recent years, deep learning approaches have shown promising results in this domain. The proposed mechanism consists of two main components: an image encoder and a caption decoder. The image encoder extracts meaningful features from the input image using a pre-trained convolutional neural network (CNN). These features are then fed into the LSTM-based caption decoder, which generates a sequence of words to form a coherent caption. The decoder is trained using a teacher-forcing technique to optimize the generation process.

Introduction:

In recent years, the demand for automated image captioning systems has surged with the proliferation of digital media and the need for efficient content indexing and retrieval. Traditional approaches have relied on static models, often lacking in adaptability and robustness. Leveraging deep learning techniques, such as LSTM networks, has shown promise in generating more accurate and contextually rich captions. However, deploying such models efficiently within an operating system environment remains a challenge. Operating systems provide a fundamental infrastructure for managing hardware resources and facilitating communication between software components. By integrating LSTM-based Tensorflow mechanisms into the operating system environment, we can leverage these capabilities to optimize resource utilization, enhance scalability, and ensure robustness against varying workloads and system conditions. This paper introduces an operating system implementation of an LSTM-based Tensorflow mechanism for generating image captions, aiming to bridge the gap between deep learning research and practical deployment in real-world systems

**OBJECTIVE:**

The primary objective of this research is to develop an operating system-based solution for deploying an LSTM-based Tensorflow mechanism dedicated to generating captions for images. By integrating the model into the operating system's framework, we aim to optimize resource utilization, enhance scalability, and ensure robustness against varying workloads and system conditions. Automated image captioning systems have become increasingly vital in managing the vast amounts of digital media content available today. However, traditional approaches often lack adaptability and robustness. Deep learning techniques, such as LSTM networks, offer a promising solution to this challenge by enabling more accurate and contextually rich caption generation. Despite their potential, deploying such models efficiently within an operating system environment remains a significant hurdle. This research addresses this gap by developing an operating system-based solution tailored specifically for LSTM-based Tensorflow mechanisms dedicated to generating image captions.

Proposed Solution

Integration of LSTM-based Tensorflow mechanism:

We will integrate the LSTM-based caption generation model into the operating system environment, allowing seamless interaction with image data and other system resources.

Optimization for resource utilization:

Leveraging the features provided by the operating system, we will optimize resource allocation and scheduling to ensure efficient utilization of CPU, memory, and storage resources during caption generation.

Adaptability and scalability:

The system will be designed to adapt dynamically to changing system conditions and workload demands, ensuring scalability across different hardware configurations and usage scenarios.

The main aim of this project is to get a little bit of knowledge of deep learning techniques.

We use two techniques mainly CNN and LSTM for image classification. So, 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. The project seeks to construct an image caption generator model by integrating these architectures, commonly referred to as a CNN-RNN model. To harness the strengths of both CNN and LSTM, a pre-trained CNN model, Xception, renowned for its proficiency in extracting intricate features from images, will be employed. The collaborative approach involves using the pre-trained CNN to extract meaningful visual features, which are then fed into the LSTM network. The LSTM, being adept at capturing sequential dependencies, utilizes the information from the CNN to generate descriptive captions for the given images. The synergy between CNN's feature extraction capabilities and LSTM sequential modeling aims to enhance the overall efficiency of the caption generation process. Leveraging a pre-trained model like Xception not only optimizes computational resources but also capitalizes on the knowledge acquired from a vast dataset. This fusion of CNN and LSTM endeavors to create a robust image caption generator capable of comprehensively understanding and describing the visual content within images.

PRODECURE:

Digital image processing **using operating system:-**

The proposed operating system environment will provide a robust platform for digital image processing tasks, including pre-processing and feature extraction. By integrating image processing capabilities with the LSTM-based Tensorflow mechanism, we can enhance the overall accuracy and relevance of the generated captions. Operating systems serve as the backbone of modern computing systems, providing essential services such as process management, memory management, and device control. By leveraging the capabilities of operating systems, we can streamline the deployment and execution of deep learning models for image captioning tasks. This paper presents an innovative approach to integrating LSTM-based Tensorflow mechanisms into the operating system environment, thereby enhancing the efficiency and scalability of automated image captioning systems.The identification of objects in an image would probably start with image processing techniques such as noise removal, followed by (low-level) feature extraction to locate lines, regions and possibly areas with certain textures.

The clever bit is to interpret collections of these shapes as single objects, e.g. cars on a road, boxes on a conveyor belt or cancerous cells on a microscope slide. One reason this is an AI problem is that an object can appear very different when viewed from different angles or under different lighting. Another problem is deciding what features belong to what object and which are background or shadows etc. the human visual system performs these tasks mostly unconsciously but a computer requires skillful programming and lots of processing power to approach human performance. manipulating data in the form of an image through several possible techniques. an image is usually interpreted as a two-dimensional array of brightness values, and is most familiarly represented by such patterns as those of a photographic print, slide, television screen, or movie screen. an image can be processed optically or digitally with a computer.

Recognizing object classes in real-world images is a long standing goal in computer vision. conceptually, this is challenging due to large appearance variations of object instances belonging to the same class. Additionally, distortions from background clutter, scale, and viewpoint variations can render appearances of even the same object instance to be vastly different. further challenges arise from intra class similarity in which instances from different classes can appear very similar. Consequently, models for object classes must be flexible enough to accommodate class variability, yet discriminative enough to sieve out true object instances in cluttered images. These seemingly paradoxical requirements of an object class model make recognition difficult. This paper addresses two goals of recognition are image classification and object detection. The task of image classification is to determine if an object class is present in an image, while object detection localizes all instances of that class from an image. toward these goals, the main contribution in this paper is an approach for object class recognition that employs edge information only. The novelty of our approach is that we represent contours by very simple and generic shape primitives of line segments and ellipses, coupled with a flexible method to learn discriminative primitive combinations. These primitives are complementary in nature, where line segment models straight contour and ellipse models curved contour.

Function:

**Image data management:**

The operating system will provide efficient mechanisms for managing and accessing image data, including storage, retrieval, and manipulation.

**Resource allocation and scheduling:**

Leveraging the operating system's scheduling algorithms, we will ensure fair and efficient allocation of system resources to caption generation tasks.

**Inter-process communication:**

Facilitating communication between the caption generation module and other system components, such as image processing modules, to enable seamless collaboration and data exchange.

Advantages:

Number captions will increase

Exact caption prediction

Large number of dataset of captions

Applications:

* Photography applications
* Film industries
* Medical applications

ALGORITHM:

We cannot pass the string captions directly as input to the neural network because the neural network cannot process string as input so the captions which are in the form of strings to numbers for that process we need to build a vocabulary of numbers. This process is called encoding of captions. Firstly, After preprocessing of the captions given in the training data set we need to create a new space where all words in every caption are taken. Now we have to give numbers to the words sequentially in the dictionary order. Now this space is called a vocabulary library. With the help of this vocab library we'll number each captions by numbering their words accordingly with the vocab library. For a given caption each word is numbered by referring to their values in the already defined vocab library. For example: Let us consider a Vocab Library that we have built by numbering every unique word of the given training captions Now consider the caption={the cat is on the table}.Now this caption can be encoded into numbers using the dictionary and we can encode this caption as caption={5000 450 890 1120 5000 3770}. Now this encoded caption is passed into a neural network(LSTM) for training the model to generate captions. 4.5.DEFINING AND FITTING THE MODEL After collecting the data set and preprocessing the images and captions and building vocabulary. Now we have to define the model for generation of captions. Our proposed model is the ResNet(Residual Neural Network)-LSTM(Long Short Term Memory) model. In this model Resnet is used as an encoder which extracts the image features from the images and converts them into a single layered vector and passes them as input to LSTM . Long Short Term Memory is used as decoder which takes image features as input and also vocabulary dictionary to generate each word of the caption sequentially. 4.5.1.RESNET 50 With the introduction of transfer learning (using knowledge gained in training network on one type of problem and applying the knowledge in another problem of same pattern) using deep neural networks like RESNET(Residual Neural Network) which is a pretrained model for many image recognition and classification became easy. We use this ResNet model in place of Deep Convolutional Neural Network because ResNet is a pretrained model on ImageNet data set to classify the images. So by using the concept of transfer learning we are reducing the computation cost and training time.

If we have used CNN which is not pre trained then the computation cost would have increased and the model takes more time to learn. By using ResNet pretrained model we are also increasing the accuracy of the model. Resnet50 consists of 50 deep convolutional neural network layers. ResNet50 is the architecture of Convolutional Neural Network that we are using in Image Caption Generation Deep Learning Model. The last layer of Resnet50 is removed as it gives classification output and we are accessing the output of the o layer before the last one in order to get the image features as output single layered vector because we don’t need classification output in this paper. The ResNet is preferred compared to traditional deep convolutional neural networks because the ResNet contains residual blocks which have skip connections that ultimately reduce the vanishing gradient problem in CNN and ResNet also decreases the loss of input features compared to CNN. ResNet is having better performance and accuracy in classification of images and extracting image features compared to traditional CNN ,VGG. Below is the figure representing the working of ResNet block and its importance compared to traditional CNN.

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over-fitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid overfitting, both methods use a test set (not seen by the model) to evaluate model performance.

Performance of each classification model is estimated based on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs.

Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

**CODE**

from keras.applications.xception import Xception

def extract\_features(directory):

model = Xception(include\_top=False, pooling='avg')

features = {}

for img in tqdm(os.listdir(directory)):

filename = directory + "/" + img

image = Image.open(filename)

image = image.resize((299,299))

image = np.expand\_dims(image, axis=0)

image = image/127.5

image = image - 1.0

feature = model.predict(image)

features[img] = feature

return features

features = extract\_features(dataset\_images)

from keras.preprocessing.text import Tokenizer

# tokenize the descriptions

tokenizer = Tokenizer()

tokenizer.fit\_on\_texts(descriptions)

# convert the descriptions to sequences

sequences = tokenizer.texts\_to\_sequences(descriptions)

# convert the sequences to padded sequences

data = pad\_sequences(sequences, maxlen=max\_length)

# one-hot encode the descriptions

one\_hot = to\_categorical(data)

# extract features for each image

images = np.array([features[img] for img in train\_imgs])

from keras.layers import Input, LSTM, Dense

# define the LSTM model

inputs1 = Input(shape=(2048,))

fe1 = Dropout(0.5)(inputs1)

fe2 = Dense(256, activation='relu')(fe1)

inputs2 = Input(shape=(max\_length,))

se1 = Embedding(vocab\_size, 256, mask\_zero=True)(inputs2)

se2 = Dropout(0.5)(se1)

se3 = LSTM(256)(se2)

decoder1 = add([fe2, se3])

decoder2 = Dense(256, activation='relu')(decoder1)

outputs = Dense(vocab\_size, activation='softmax')(decoder2)

model = Model(inputs=[inputs1, inputs2], outputs=outputs)

# compile the model

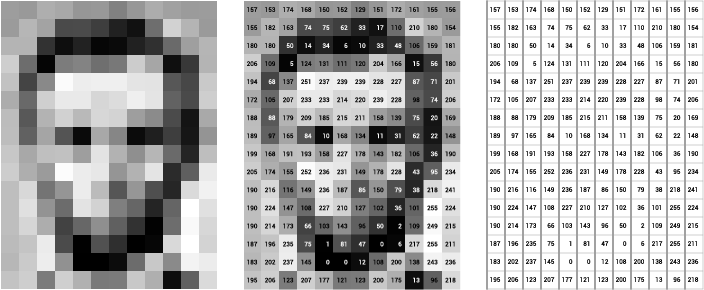
model.compile(loss='categorical\_crossentropy', optimizer='adam')

# train the model

model.fit\_generator(generator, epochs=10, steps\_per\_epoch=len(train\_descriptions))

OUTPUT:

Source: [Openframeworks](http://openframeworks.cc/ofBook/chapters/image_processing_computer_vision.html)



Results and Discussions:

After defining and fitting the model. We trained our model for 50 epochs. It is observed that during the initial epochs of training the accuracy is very low and the captions generated are not much related to given test images. If we train the model for at least 20 epochs then we have observed that the captions generated are somewhat related to the given test images. If the model is trained for 50 epochs we observe that the accuracy of the model increases and the captions generated are much related to the given test images

CONCLUSION:

In conclusion, this paper has proposed an operating system-based approach for deploying an LSTM-based Tensorflow mechanism to generate captions for images. Through the integration of deep learning techniques into the operating system environment, we aim to optimize resource utilization, enhance scalability, and ensure robustness in automated image captioning systems.

By leveraging the capabilities of modern operating systems, we can address the challenges associated with deploying complex deep learning models efficiently. The proposed solution offers several advantages, including improved adaptability to varying workloads and system conditions, streamlined resource allocation, and seamless integration with existing image processing functionalities.Our research highlights the importance of bridging the gap between deep learning research and practical deployment in real-world systems. By developing an operating system-based solution tailored specifically for LSTM-based Tensorflow mechanisms, we aim to provide a robust platform for automated image captioning tasks.captioning deep learning model is proposed in this paper. We have used the RESNET-LSTM model to generate captions for each of the given images. The Flickr 8k data set has been used for the purpose of training the model. RESNET is the architecture of convolution layers. This RESNET architecture is used for extracting the image features and these image features are given as input to Long Short Term Memory units and captions are generated with the help of vocabulary generated during the training process. We can conclude that this ResNet-LSTM model has higher accuracy compared to CNN-RNN and VGG Model. This model works efficiently when we run the model with the help of Graphic Processing Unit. This Image Captioning deep learning model is very much useful for analyzing the large amounts of unstructured and unlabeled data to find the patterns in those images for guiding the Self driving cars, for building the software to guide blind people.

REFERENCES:

[1] Liu, Shuang & Bai, Liang & Hu, Yanli & Wang, Haoran. (2018). Image Captioning Based on Deep Neural Networks. MATEC Web of Conferences. 232. 01052. 10.1051/matecconf/201823201052.

[2] A. Hani, N. Tagougui and M. Kherallah, "Image Caption Generation Using A Deep Architecture," 2019 International Arab Conference on Information Technology (ACIT), 2019, pp. 246-251, doi: 10.1109/ACIT47987.2019.8990998.

[3] S. Das, L. Jain and A. Das, "Deep Learning for Military Image Captioning," 2018 21st International Conference on Information Fusion (FUSION), 2018, pp. 2165-2171, doi: 10.23919/ICIF.2018.8455321

[4] GGeetha,T.Kirthigadevi,G GODWIN Ponsam,T.Karthik,M.Safa,” Image Captioning Using Deep Convolutional Neural Networks(CNNs)” Published under licence by IOP Publishing Ltd in Journal of Physics :Conference Series ,Volume 1712, International Conference On Computational Physics in Emerging Technologies(ICCPET) 2020 August 2020,Manglore India in 2015.

[5] Simonyan, Karen, and Andrew Zisserman. ”Very deep convolutional networks for large-scale image recognition.” arXiv preprint arXiv:1409.1556 (2014). [6] Donahue, Jeffrey, et al. ”Long-term recurrent convolutional networks for visual recogni-tion and description.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[7] Lu, Jiasen, et al. ”Knowing when to look: Adaptive attention via a visual sentinel for im-age captioning.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Vol. 6. 2017.

[8] Ordonez, Vicente, Girish Kulkarni, and Tamara L. Berg. ”Im2text: Describing images us-ing 1 million captioned photographs.” Advances in neural information processing systems. 2011.

[9] Chen, Xinlei, and C. Lawrence Zitnick. ”Mind’s eye: A recurrent visual representation for image caption generation.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[10] Feng, Yansong, and Mirella Lapata. ”How many words is a picture worth? automatic caption generation for news images.” Proceedings of the 48th annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2010.

[11] Rashtchian, Cyrus, et al. ”Collecting image annotations using Amazon’s Mechanical Turk.” Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk. Association for Computational Linguistics, 2010