International Journal of Computer Applications Technology and Research
Volume 6–Issue 7, 333-337, 2017, ISSN:-2319–8656

Deep Learning for Intelligent Exploration of Image Details

Okanti Apoorva
Department of CSE
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad,India

Y.Mohan Sainath
Department of CSE
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India

G.Mallikarjuna Rao
Department of CSE
Professor
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India

Abstract: Automatic image captioning is the task where given an image the system must generate a caption that describes the contents of the image. Once you can detect objects in photographs and generate labels for those objects, you can see that the next step is to turn those labels into a coherent sentence description. Most of the approaches involve the use of very large convolution neural networks (CNN) for the object detection in the photographs and then a recurrent neural network (RNN) like an LSTM (Long short-term memory) to turn the labels into a coherent sentence. In our proposed approach we have tailored the CNN and LSTM and has been tested with CIFAR 10 and MNIST datasets. The experimentation resulted 94.67% accuracy with 25 random iterations.

Keywords Deep Learning, Convolutional Neural Network, Long Short Term Memory, Recurrent Neural Networks, CIFAR10, MNIST.

1. INTRODUCTION

Automatic Image Caption Generation:

Automatically generating captions of an image is a task very close to the heart of scene understanding, is one of the primary goals of computer vision. Caption generation models must be powerful enough to solve the computer vision challenges of determining which objects are in an image, and also be capable enough to capture and express their relationships in a natural language. Due to this caption generation has long been viewed as a difficult problem. It poses considerable challenge for machine learning algorithms, as it amounts to mimicking the remarkable human ability to compress huge amounts of salient visual information into descriptive language.

Background:

Recently, several methods have been proposed for generating image descriptions. Many of these methods are based on recurrent neural networks and inspired by the successful use of sequence to sequence training with neural networks for machine translation. One major reason image caption generation is well suited to the encoder-decoder framework of machine translation is because it is analogous to “translating” an image to a sentence. Generating automatic descriptions from images requires an understanding of how humans describe images. An image description can be analyzed in several different dimensions. We assume that the descriptions that are of interest for this survey article are the ones that verbalize visual and conceptual information depicted in the image, i.e., descriptions that refer to the depicted entities, their attributes and relations, and the actions they are involved in. Outside the scope of automatic image description are non-visual descriptions, which give background information or refer to objects not depicted in the image (e.g., the location at which the image was taken or who took the picture). Also, not relevant for standard approaches to image description are perceptual descriptions, which capture the global low-level visual characteristics of images (e.g., the dominant color in the image or the type of the media such as photograph, drawing, animation, etc.).[1][6]

The general approach of the studies in this group is to first predict the most likely meaning of a given image by analyzing its visual content, and then generate a sentence reflecting this meaning. All models in this category achieve this using the following general pipeline architecture: 1. Computer vision techniques are applied to classify the scene type, to detect the objects present in the image, to predict their attributes and the relationships that hold between them, and to recognize the actions taking place. 2. This is followed by a generation phase that turns the detector outputs into words or phrases. These are then combined to produce a natural language description of the image, using techniques from natural language generation (e.g., templates, n-grams, grammar rules) [5].

The approaches reviewed in this section perform an explicit mapping from images to descriptions. Explicit pipeline architecture, while tailored to the problem at hand, constrains the generated descriptions, as it relies on a predefined set of semantic classes of scenes, objects, attributes, and actions. Moreover, such architecture crucially assumes the accuracy of the detectors for each semantic class, an assumption that is not always met in practice.

Big Data is essentially a special application of data science, in which the data sets are enormous and require overcoming logistical challenges to deal with them. The primary concern is efficiently capturing, storing, extracting, processing, and analyzing information from these enormous data sets.

Processing and analysis of these huge data sets is often not feasible or achievable due to physical and/or computational constraints. Special techniques and tools (e.g., software, algorithms, parallel programming, etc.) are therefore required. Big Data is the term that is used to encompass these large data sets, specialized techniques, and customized tools. It is often applied to large data sets in order to perform general data analysis and find trends, or to create predictive models. A primary component of big data is the so-called three Vs (3Vs) model. This model represents the characteristics and challenges of big data as dealing with volume, variety, and velocity. Companies such as IBM include a fourth "V", veracity.

1.1 Deep Learning

Deep learning is a type of machine learning that trains a computer to perform human-like tasks, such as recognizing speech, identifying images or making predictions. Instead of organizing data to run through predefined equations, deep learning sets up basic parameters about the data and trains the computer to learn on its own by recognizing patterns using many layers of processing. Deep learning methods have ability to continuously improve and adapt to changes in the underlying information pattern, presents a great opportunity to introduce more dynamic behavior into analytics. When put in other terms the deep learning can also be defined as the study of artificial neural networks and related machine learning algorithms which contain more than one hidden layer.[7][8][17]

1.2 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is comprised of one or more convolutional layers (often with a subsampling step) and then followed by one or more fully connected layers as in a standard multi-layer neural network. The architecture of a CNN is designed to take advantage of the 2D structure of an input image (or other 2D input such as a speech signal). This is achieved with local connections and weights' followed by some form of pooling which results in translation invariant features. Another benefit of CNNs is that they are easier to train and have many fewer parameters than fully connected networks with the same number of hidden units. The convolutional neural network is also known as shift invariant or space invariant artificial neural network (SIANN), which is named based on its shared weights architecture and translation invariance characteristics. Convolutional networks may include local or global pooling layers, which combine the outputs of neuron clusters. The CNN also has various combinations of convolutional layers and fully connected layers. A point wise non linearity is applied after every layer or at the end of each layer. To improve generalization and to reduce the number of free parameters, a convolution operation on small regions of input is introduced. [9][10][11]

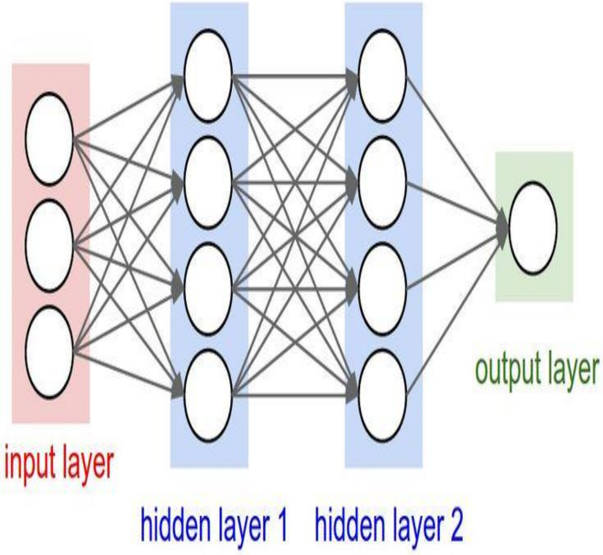


Figure 1. A Simple Convolutional Neural Network.

1.3 Long Short Term Memory

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They work tremendously well on a large variety of problems, and are now widely used. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behaviour, not something they struggle to learn. All recurrent neural networks have the form of a chain of repeating modules of neural network. This design is typical with "deep" multi-layered neural networks, and facilitates implementations with parallel hardware. [12][18]

LSTM blocks contain three or four "gates" that they use to control the flow of information into or out of their memory. These gates are implemented using the logistic function to compute a value between 0 and 1.

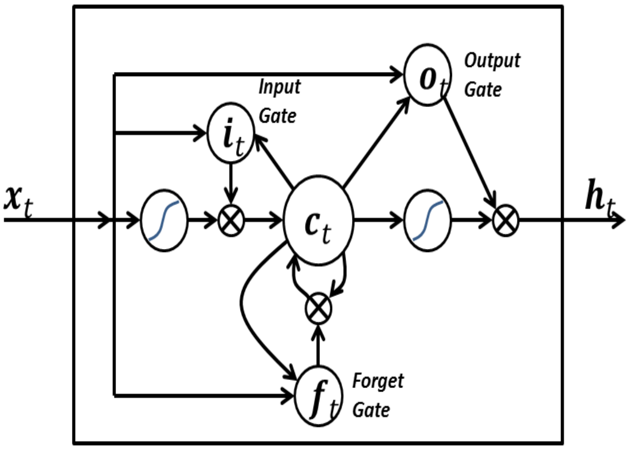


Figure 2: LSTM Gates

2. PROPOSED SYSTEM

In this image analysis we use convolutional neural networks and long short term memory for character recognition and image caption generation. Generating automatic descriptions from images requires an understanding of how humans describe images. An image description can be analyzed in several different dimensions. We assume that the descriptions that are of interest for this survey article are the ones that verbalize visual and conceptual information depicted in the image, i.e., descriptions that refer to the depicted entities, their attributes and relations, and the actions they are involved in.

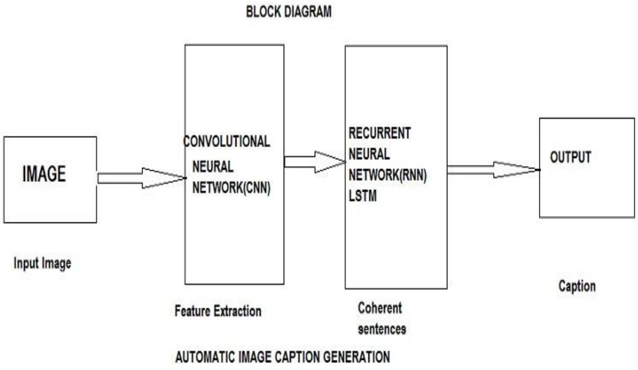

BLOCK DIAGRAM
IMAGE
Input Image
CONVOLUTIONAL NEURAL NETWORK(CNN)
Feature Extraction
RECURRENT NEURAL NETWORK(RNN) LSTM
Coherent sentences
OUTPUT
Caption
AUTOMATIC IMAGE CAPTION GENERATION

Figure 3: Block Diagram of Image Caption Generation

Outside the scope of automatic image description are non-visual descriptions, which give background information or refer to objects not depicted in the image (e.g., the location at which the image was taken or who took the picture). Also, not relevant for standard approaches to image description are perceptual descriptions, which capture the global low-level visual characteristics of images (e.g., the dominant color in the image or the type of the media such as photograph, drawing, animation, etc.).

CNN Architecture (ConvNet architectures)

Convolutional Layer, Pooling Layer and Fully-Connected Layer (exactly as seen in regular Neural Networks). We will stack these layers to form a full ConvNet architecture.

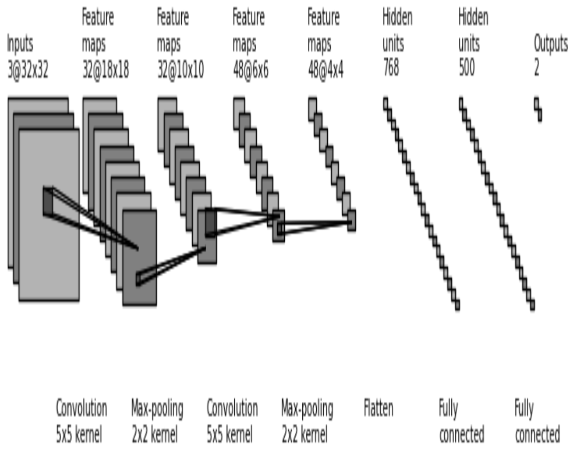


Figure 4: CNN for Image Caption Generation

Input [32x32x3] will hold the raw pixel values of the image, in this case an image of width 32, height 32, and with three color channels R, G, B. Convolutional layer will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume. This may result in volume such as [32x32x12] if we decided to use 12 filters. Relu layer will apply an element wise activation function, such as the max (0, x) thresholding at zero. This leaves the size of the volume unchanged ([32x32x12]). Pool layer will perform a down sampling operation along the spatial dimensions (width, height), resulting in volume such as [16x16x12]. FC (i.e. fully-connected) layer will compute the class scores, resulting in volume of size [1x1x10], where each of the 10 numbers correspond to a class score, such as among the 10 categories of cifar-10. As with ordinary neural networks and as the name implies, each neuron in this layer will be connected to all the numbers in the previous volume, [2][3].

LSTM Architecture:

The core of the LSTM model is a memory cell c encoding knowledge at every time step of what inputs have been observed up to this step. The behavior of the cell is controlled by “gates” – layers which are applied multiplicatively and thus can either keep a value from the gated layer if the gate is 1 or zero this value if the gate is 0. In particular, three gates are being used which control whether to forget the current cell value (forget gate f), if it should read its input (input gate i) and whether to output the new cell value (output gate o). The definition of the gates and cell update and output are as follows

$$
\begin{aligned}
i_t &= \sigma(W_{ix}x_t + W_{im}m_{t-1})\\
f_t &= \sigma(W_{fx}x_t + W_{fm}m_{t-1})\\
o_t &= \sigma(W_{ox}x_t + W_{om}m_{t-1})\\
c_t &= f_t * c_{t-1} + i_t * h(W_{cx}x_t + W_{cm}m_{t-1})\\
m_t &= o_t * c_t\\
p_{t+1} &= \mathrm{Softmax}(m_t)
\end{aligned}
$$

Where * represents the product with a gate value, and the various W matrices are trained parameters. Such multiplicative gates make it possible to train the LSTM robustly as these gates deal well with exploding and vanishing gradients. The nonlinearities are sigmoid σ (·) and hyperbolic tangent h (·). The last equation m_t is what is used to feed to a Softmax, which will produce a probability distribution p_t over all words [2][4].

3. EXPERIMENTATION

Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy which emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly braces or keywords), and a syntax which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale. To install any package in python, we use a pip command [13][14].

Syntax: Pip install <package- name>

To implement our project, we have used some libraries as Keras and Tensor flow

3.1 Keras Library

Keras is a high-level neural networks API, written in Python and capable of running on top of either TensorFlow or Theano and Runs seamlessly on CPU and GPU. It was developed with a focus on enabling fast experimentation [15].

3.2 Tensor flow

TensorFlow is Google Brain's second generation machine learning system, released as open source software on November 9, 2015. Among the applications for which TensorFlow is the foundation, are automated image captioning software, such as Deep Dream [16].

3.3 Dataset

The databases used in this image analysis are MNIST database and CIFAR10 database. The MNIST database is used in hand written character recognition and CIFAR 10 is used in image caption generation.

MNIST:

MNIST dataset (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems. The MNIST database contains 60,000 training images and 10,000 testing images. Half of the training set and half of the test set were taken from NIST's training dataset, while the other half of the training set and the other half of the test set were taken from NIST's testing dataset. The different machine learning methods are used on the dataset and different error rates are found. We have implemented the above LSTM with MNIST dataset. To have reasonable convergence we limited the epochs to less than 10.

CIFAR 10:

The CIFAR-10 dataset consists of 60000, 32x32, colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class. The classes are completely mutually exclusive. There is no overlap between various classes.

4. EXPERIMENTAL RESULTS:

4.1 Automatic Caption Generation

Automatically generating captions of an image is a task very close to the heart of scene understanding — one of the primary goals of computer vision. Not only must caption generation models be powerful enough to solve the computer vision challenges of determining which objects are in an image, but they must also be capable of capturing and expressing their relationships in a natural language. For this reason, caption generation has long been viewed as a difficult problem. It is a very important challenge for machine learning algorithms, as it amounts to mimicking the remarkable human ability to compress huge amounts of salient visual information into descriptive language [5].

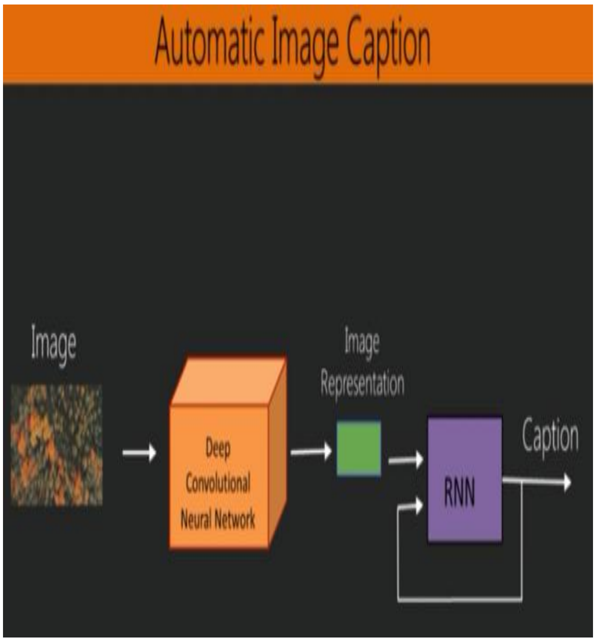


Figure 5: Architecture of Image Caption Generation

4.2 Character Recognition Experiment

In character recognition we use MNISTdataset .Here the data is processed by the convolutional neural networks. The neural network takes the input from the MNIST dataset trains and tests them and gives their accuracy values after being processed through the multiple number of hidden layers.

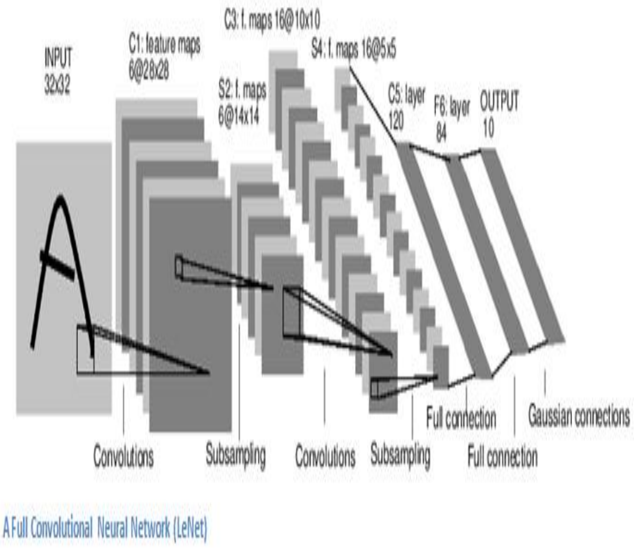


Figure 6: CNN for Character Recognition

5. CONCLUSION:

From the experimentation it can be seen how the learned attention can be exploited to give more interpretability into the models generation process, and demonstrate that the learned alignments correspond very well to human intuition.

The following gives sample snapshots of our experimentation. From the table we can conclude that as the epochs are increased recognition accuracy is increased (for 10 epoch's accuracy of recognition increased to 100%)

Table 1: Results obtained using the proposed model for MNIST & CIFAR10 dataset

Research highlights:

Number of Epochs	Recognition Accuracy (%)	
	MNIST Dataset	CIFAR10 Dataset
2	60	65
4	72	77
6	75	79.5
8	80	82
9	97	100
10	100	100
20	100	100

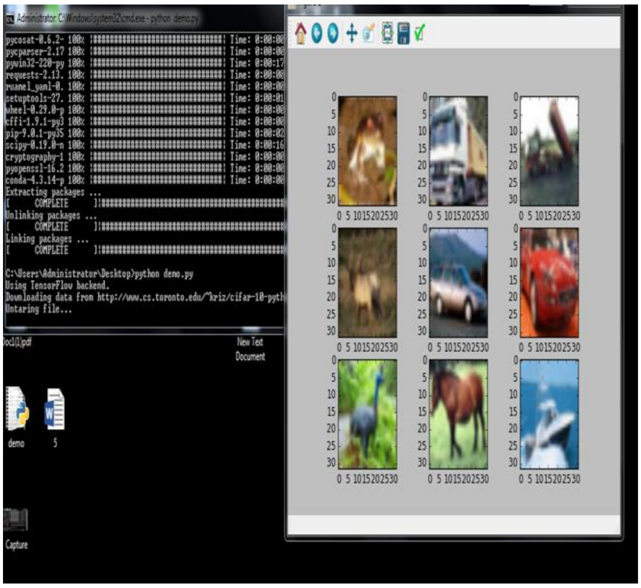
(i) The paper proposes Deep learning approach for the exploration of Image Details.

(ii) Convolutional Neural Networks and LSTM are used to evaluate performance.

(iii) Different Data sets MNIST and CIFAR 10 are used.

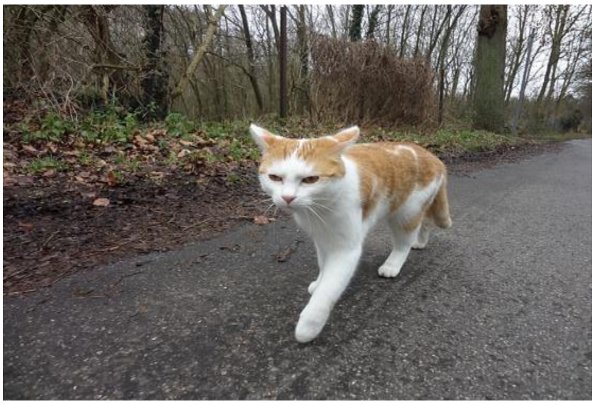
(iv) Multilayer approach has given true recognition rate above >90%.

TRAINING CIFAR 10 DATASET



```
Administrator: C:\Windows\system32\cmd.exe - python demo.py
pycosat-0.6.2- 100% |################################| Time: 0:00:00
pycparser-2.17 100% |################################| Time: 0:00:00
pywin32-220-py 100% |################################| Time: 0:00:17
requests-2.13. 100% |################################| Time: 0:00:00
ruamel_yaml-0. 100% |################################| Time: 0:00:00
setuptools-27. 100% |################################| Time: 0:00:01
wheel-0.29.0-p 100% |################################| Time: 0:00:00
cffi-1.9.1-py3 100% |################################| Time: 0:00:00
pip-9.0.1-py35 100% |################################| Time: 0:00:02
scipy-0.19.0-n 100% |################################| Time: 0:00:16
cryptography-1 100% |################################| Time: 0:00:00
pyopenssl-16.2 100% |################################| Time: 0:00:00
conda-4.3.14-p 100% |################################| Time: 0:00:00
Extracting packages ...
[      COMPLETE      ]|########################################
Unlinking packages ...
[      COMPLETE      ]|########################################
Linking packages ...
[      COMPLETE      ]|########################################

C:\Users\Administrator\Desktop>python demo.py
Using TensorFlow backend.
Downloading data from http://www.cs.toronto.edu/~kriz/cifar-10-pyth
Untaring file...
```



OUTPUT

AUTOMATIC IMAGE CAPTION GENERATION

A Cat is walking on road

Figure 7: Experimentation of the image caption generation

Acknowledgement: We are thankful to our beloved director P.S Raju, Principal J.N Murthy for their kind support. We would like to thank coordinator of Parallel programming and operating systems lab (MODROB) for allowing the resources to carry our experimentation in the lab.

6. REFERENCES:

[1]. Automatic Description Generation from Images: A Survey of Models, Datasets, and Evaluation Measures by Raffaella Bernardi , Ruket Cakici , Desmond Elliott.

[2]. Learning CNN-LSTM Architectures for Image Caption Generation by Moses Soh.

[3]. Andrej Karpathy, Fei-Fei Li: Automated Image Captioning with ConvNets and Recurrent Nets.

[4]. Sepp Hochreiter and Jurgen Schmidhuber: Long Short Term Memory.

[5]. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.

[6]. Karol Gregor, Ivo Danihelka, Alex Graves, and Daan Wierstra.DRAW: A recurrent neural network for image generation.

[7]. Deep Learning: by Ian Goodfellow and Yoshua Bengio and Aaron Courville.

[8]. Deep Learning: Methods and Applications by Li Deng, Dong Yu

[9]. Neural Networks and Deep Learning by Michael Nielsen.

[10]. Introduction to Machine Learning: Smola and Vishwanathan.

Websites:

[11].https://en.wikipedia.org/wiki/Convolutional_neural_network

[12].https://en.wikipedia.org/wiki/Long_short-term_memory

[13].https://www.tutorialspoint.com/python/

[14].https://en.wikipedia.org/wiki/Python_(programming_language)

[15]. https://en.wikipedia.org/wiki/Keras

[16]. https://en.wikipedia.org/wiki/TensorFlow

[17]. https://en.wikipedia.org/wiki/deeplearning

[18]. https://en.wikipedia.org/wiki/recurrent neural network

Authors:

Okanti Apoorva is pursuing master degree in computer science in Gokaraju Rangaraju Institute of Engineering and Technology. Her research interests are Neural Networks and Image Processing.

G.Mallikarjuna Rao is working as professor in CSE department of Gokaraju Rangaraju Institute of Engineering and Technology. His research areas are Machine Vision, Parallel Computing and Neural networks.