Advances in Intelligent Systems and Computing 1141

Aboul Ella Hassanien
Roheet Bhatnagar
Ashraf Darwish *Editors*

Advanced Machine Learning Technologies and Applications

Proceedings of AMLTA 2020

Springer

Advances in Intelligent Systems and Computing

Volume 1141

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within "Advances in Intelligent Systems and Computing" are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

**** Indexing: The books of this series are submitted to ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink ****

More information about this series at http://www.springer.com/series/11156

Aboul Ella Hassanien · Roheet Bhatnagar ·
Ashraf Darwish
Editors

Advanced Machine Learning Technologies and Applications

Proceedings of AMLTA 2020

Springer

Editors
Aboul Ella Hassanien
Faculty of Computer and Information
Cairo University
Giza, Egypt

Ashraf Darwish
Faculty of Science
Helwan University
Helwan, Egypt

Roheet Bhatnagar
Department of Computer Science
and Engineering
Manipal University Jaipur
Jaipur, Rajasthan, India

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-981-15-3382-2 ISBN 978-981-15-3383-9 (eBook)
https://doi.org/10.1007/978-981-15-3383-9

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Aboul Ella Hassanien dedicated this work to the scientific research group members in Egypt and his wife Nazaha Hassan El Saman.

Professor Roheet dedicated the proceedings to his parents, his wife, and his son Rishaan.

AMLTA 2020—Conference Committees

Chief Patron

K. Ramnarayan, Chairperson, Manipal University Jaipur, India

Patron

Gopalkrishna Prabhu, President, Manipal University Jaipur, India

Co-patrons

Niti Nipun Sharma, Pro-president, Manipal University Jaipur, India
H. Ravishankar Kamath, Registrar, Manipal University Jaipur, India
Jagannath Korody, Dean (FoE), Manipal University Jaipur, India

Honorary Chairs

Ramjee Prasad, Aarhus University, Denmark
Mohamed F. Tolba, Cairo, Egypt

General Chairs

Aboul Ella Hassanien, Cairo University and SRGE, Egypt
Rajveer Singh Shekhawat, Manipal University Jaipur, India

Organizing Chair

Roheet Bhatnagar, Manipal University Jaipur, India

Program Chairs

Sandeep Joshi, India
Vijay Pal Singh, Dhaka, India
Pankaj Vyas, India

Program Co-chairs

Devesh Kumar Srivastava, India
Uma Shankar Rawat, India

Co-chairs

Sandeep Chaurasia, India
Ashish Kumar, India
Vivek Kumar Verma, India

Publication Chairs

Vivek Kumar Verma, India
R. Jaya Krishna, India

Finance Chairs

Sandeep Chaurasia, India
Anita Shrotriya, India

Sponsorship Chair

C. S. Lamba, India

Local Organizing Chair

Venkatesh Gauri Shankar, India

Registration Chair

Akhilesh Kr. Sharma, India

Travel and Transport Chair

Priyank Singh Hada, India

Printing and Media Chair

Yogesh Gupta, India

Special Session and Workshop Chair

Dilbag Singh, India

Ph.D. Forum Chair

Pradeep Kumar Tiwari, India

Publicity Chairs

Hamidreza Boveiri, Iran
Tarun Jain, India

International Advisory Committee

Ajith Abraham, USA
P. K. Mahanti, Canada
Lipo Wang, Singapore
Nitin Kumar Tripathi, Thailand
Suphamit Chittayasothorn, Thailand
Allam Hamdan, Oman
Peter Damoah Afari, Hong Kong
Mridul Malayanil, USA
V. G. Tharinda Nishantha Vidanagama, Sri Lanka
Shalabh Bhatnagar, India
Kumkum Garg, India
Mrinal Kanti Ghose, India
Vandana Bhattacharjee, India
E. Poovammal, India
Yasmeen Pervez, India

Technical Program Committee (TPC)

Edgard Marx, Germany
Islam Amin, Australia
Ivan Ermilov, Germany
Arvind Kilaru, Canada
Ashok Rajan, USA
Moheet Bhatnagar, USA
Shoji Tominaga, Japan
Arvind Kumar Pal, Japan
Takahiko Horiuchi, Japan

Seyedali Mirjalili, Australia
Fatos Xhafa, Spain
Vaclav Snasel, Czech Republic
Janusz Kacprzyk, Poland
Tai-hoon Kim, Korea
Evgenia Theodotou, Greece
Pavel Kromer, Czech Republic
Irma Aslanishvili, Czech Republic
Jan Platos, Czech Republic
Hani M. K. Mahdi, Egypt
Essam Elfakharanyb, Egypt
Ashraf AbdelRaouf, Egypt
Mohamed Abd Elfattah, Egypt
Tahani Alsubait, UK
Evgenia Theodotou, Greece
Natalia Spyropoulou, Greece
Dimitris Sedaris, Greece
Litsa Charitaki, Greece
Elena Amaricai, Greece
Manal Abdullah, KSA
Anandavalli M, KSA
Mohammad Reza Noruzi, Iran
P. K. Upadhyay, UAE
Abdelhameed Ibrahim, Egypt
Ahmad Taher Azar, Egypt
Nitul Dutta, India
Deepak Dembla, India
Sudipta Roy, India
Sonal Jain, India
Neeraj Bhargava, India
Pijush Barthakur, India
Ritu Bhargava, India
Babita Malik, India
Nitu Bhatnagar, India
Ram Naresh Saraswat, India
Pushpendra Kumar, India
Manish Thukral, India
Abhishek Srivastava, India
Swati Srivastava, India
Swagatam Das, India
Gaurav Prasad, India
Jaya Krishna, India
Harish Sharma, India
Gulrej Ahmed, India
Sandeep Joshi, India

Devi Prasad Sharma, India
Bhavna Saini, India
Ashish Jain, India
Nanda Gopal Reddy, India
Jeril Kuriakose, India
Puspendra Singh Sisodia, India
Ajay Prasad, India
Tipendra Pal Singh, India
D. K. Lobiyal, India
Kamakshi Prasad, India
Sunita Singhal, India
Vivekanand Tiwari, India
Tarun Kumar Dubey, India
Amit Rathi, India
Ankit Srivastava, India
Rohit Kumar Gupta, India
Rahul Saxena, India
Punit Sharma, India
Priyank Singh Hada, India
Prakash Ramani, India
Pratistha Mathur, India
Pallavi Bhatt, India
Monika Jain, India
Manoj Bohra, India
Shikha Mundra, India
Anubha Parashar, India
Vivek Singh Sikarwar, India
Rishi Gupta, India
Nitish Pradhan, India
Alka Choudhary, India
Giriraj Methi, India
Rajdeep Chowdhury, India

Preface

This volume constitutes the refereed proceedings of the 5th International Conference on Advanced Machine Learning Technologies and Applications, AMLTA 2020, held in Jaipur, a UNESCO World Heritage Site and capital city of state of Rajasthan, India, during February 13–15, 2020. In response to the call for papers for AMLTA 2020, 148 papers were submitted for presentation and inclusion in the proceedings of the conference. After a careful blind refereeing process, 65 papers were selected for inclusion in the conference proceedings. The papers were evaluated and ranked on the basis of their significance, novelty, and technical quality by at least two reviewers per paper. The papers cover current research in machine learning, data sciences, deep learning, text visualization, modeling optimization complex network, renewable energy, swarm intelligence, biomedical engineering, complex control systems, cyber-security, and data mining.

We express our sincere thanks to the plenary and tutorial speakers, workshop/special session chairs, and international program committee members for helping us to formulate a rich technical program. We would like to extend our sincere appreciation for the outstanding work contributed over many months by the Organizing Committee: local organization chair and publicity chair. We also wish to express our appreciation to the SRGE members for their assistance. We would like to emphasize that the success of AMLTA 2020 would not have been possible without the support of many committed volunteers who generously contributed their time, expertise, and resources toward making the conference an unqualified success. Finally, thanks to Springer team for their support in all stages of the production of the proceedings. We hope that you will enjoy the conference program.

Giza, Egypt	Aboul Ella Hassanien
Jaipur, India	Roheet Bhatnagar
Helwan, Egypt	Ashraf Darwish

Contents

About the Editors

Aboul Ella Hassanien is the Founder and Head of the Scientific Research Group in Egypt (SRGE) and a Professor of Information Technology at the Faculty of Computer and Information, Cairo University. Professor Hassanien has more than 1000 scientific research papers published in prestigious international journals to his credit.

Roheet Bhatnagar works at Manipal University Jaipur (MUJ), where he is currently serving as a Professor at the Department of Computer Science & Engineering. He has been affiliated with the MUJ since 2012, and with the Manipal group since 2008. With vast experience spanning more than 20 years in both industry and academia, he has worked for companies like Xerox Corporation, Samsung SDS and USHA Soft in various capacities. He is a Senior Member of the IEEE and Life Member of the ISTE and ISRS.

Ashraf Darwish is a Professor of Computer Science and former Acting Head of the Mathematics and Computer Science Department at Helwan University, Egypt. Dr Darwish has a wealth of academic experience and has authored numerous publications, including papers in international journals and conference proceedings. Currently, he is the vice chair of the Scientific Research Group in Egypt (SRGE).

Modelling and Optimization

Segregating and Recognizing Human Actions from Video Footages Using LRCN Technique

Meet Pandya, Abhishek Pillai, and Himanshu Rupani

Abstract Computer vision is a vast area of research that includes extracting useful information from images or sequence of images. Human activity recognition is one such field undergoing lots of research. The practical application for this model is vast in various kinds of researches as well as actual practice. This paper proposes a two-model approach using a combination of a convolutional neural network using transfer learning and a long short-term memory model. CNN network is applied to gather the feature vectors for each video, and the LSTM network is used to classify the video activity. Standard activities contain benchpress, horse riding, basketball dunk, etc. A high accuracy level of 94.2% was achieved by the proposed algorithm.

Keywords Human activity recognition · LRCN · UCF-101 · Computer vision · Video analysis · Image processing

1 Introduction

An action is a set of movement performed by a human body in a sequential manner. From a computer vision perspective, action recognition is a classification of a video activity according to the sequence of images which collaboratively form a video. Action recognition was a significant area of research in the last few years due to its enormous applications like human interaction, security, communications and entertainment. Unlike static images, video recognition includes spatial as well as temporal features to identify the task which gives us a more coherent understanding of the actions performed. Researchers tend to incline toward action recognition due

M. Pandya · A. Pillai (✉)
U.V. Patel College of Engineering, Mehsana, India
e-mail: abhishekpillai.ap@gmail.com

M. Pandya
e-mail: meetpandya4715@gmail.com

H. Rupani
L.D. College of Engineering, Ahmedabad, India
e-mail: himanshurupani97@gmail.com

A. E. Hassanien et al. (eds.), *Advanced Machine Learning Technologies and Applications*, Advances in Intelligent Systems and Computing 1141,
https://doi.org/10.1007/978-981-15-3383-9_1

to its vast domain of application in surveillance, real-time gaming, video description, human interaction [1], video captioning [1, 2] and robotics [3, 4]. Sequence learning is a machine learning algorithm in which data is not independent and makes use of contextual information [5, 6]. A convolutional neural network is a conventional model for image classification but for a video each and every frame plays a critical role in analysis.

Long short-term memory networks, commonly termed LSTM, are a certain form of RNN [7], competent of grasping long-term dependency. LSTM also follows a chain-like structure where each cell contains more than one neural network and is interconnected as well as interacting with each other [8]. LSTMs help in rectifying or minimizing the error that could backpropagate through time and layers [9]. By maintaining more constant error, they allow recurrent nets to continue to learn over many time steps [10].

The below paper describes a combined model approach which includes a convolutional neural network that gives the output feature vector for video frames which is then provided to the LSTM cell, and the softmax classifier is used for classification. As a human being recognizes an activity by viewing the series of action and analyzing the attributes for some specific period of time, in this same way the model described below understands and predicts the activity. The rest of the paper is organized as follows: Sect. 2 describes the related work done on the project, Sect. 3 describes the architecture of the model used, experiment result and comparison with other model are provided in Sect. 4, and Sect. 5 concludes the paper with future research directions.

2 Related Work

Activity recognition systems are novel and highly engaging field of research and development having varied implications and applications. Nowadays, action recognition is an exclusive topic in the ongoing research works. You can refer to this [11–13] for related surveys and questions. Basic idea was to classify every frame in video and output the majority vote obtained during this process.

Two-dimensional models are the conventional 2D CNN architecture which only uses spatial features for classification. The basic approach was to take one or more than one frame of video into consideration and apply a pre-trained model to classify the image. Transfer learning algorithm was used to apply a pre-trained model on ImageNet [14] dataset to classify images. The final label could be generated by averaging out all the RGB representation and then applying to three fully connected layers for classification; if not this approach then instead the results are gained by either applying a softmax classifier or max-pooling the spatial features [15, 16].

Another method that should be taken into consideration is using CNN to classify action in-depth data directly by modeling poses in a view-invariant high-dimensional space. A synthetic 3D pose dataset [17] is created, a model is trained on hundreds of poses to create depth maps, and then a dense neural layer can be adopted for final result. Two-dimensional models consider only spatial information, and no temporal information is considered which makes it less efficient.

Three-dimensional CNN model allows capturing both spatial information and temporal information while maintaining the temporal structure of the model. A fully automated model that has no prior knowledge is trained to learn the spatiotemporal features. A recurrent neural network is trained to classify each sequence of features. In 3D convolution, the kernel used to perform convolution has an additional dimension, usually time [18, 19].

Two-stream network uses base two-stream architecture. It is a fusion of both spatial and temporal streams [20], and the spatial net can capture spatial dependency between frames while a temporal net can capture the presence of periodic motion for each spatial location in a video. These networks need to be fused at an early level such that responses for the same pixel position are put in correspondence. Different spatial fusion strategies were used like sum fusion, max fusion, concatenation fusion, conv fusion, etc., to obtain the desired results [21–24]. Two-stream 3D ConvNet architecture is created by inflating the filters and pooling kernels in order to add an extra-temporal dimension. Filters are converted from N × N to N × N × N. Also, the model takes advantage of the pre-trained model weights trained on the ImageNet dataset. The 3D model can then be explicitly trained by ImageNet dataset as pooled activation of video would be same as single static image; hence, the weights are repeated N times along the time dimension, then dividing these weights by N. UCF-101 dataset has the highest accuracy of 98% in the I3D model [25].

3 Model and Architecture

We use LRCN for action recognition challenges, which involves spatial and temporal learning. LRCN is a class of models that combines the spatial and temporal feature learning into a single model. Spatial features are extracted by an extended version of CNN, in which the Inception V3 model is applied using transfer learning, whereas temporal features are extracted by a state-of-the-art recurrent model, LSTM.

3.1 Convolutional Neural Network

Convolutional neural networks (ConvNets) are constructed of stacked feature learning stages. Each stage consists of convolution, pooling and normalization modules [8]. The benefit of CNN is its ability to learn the position and scale-invariant structures in the data, which is important when working with images. ILSVRC 2012 Alexnet introduced a CNN model for image classification which made a major breakthrough in the field of computer vision, making processes like image classification more accurate and efficient [26].

3.2 Transfer Learning

This experiment is done on a presumption that the knowledge gained from one task can be transferred to learn another task. The basic analogy that can be used here is that knowledge gained through learning what is a dot can be used to learn how to draw a line, which in turn is used to draw a rectangle, which is then further stretched to the concept of solid objects. Here, we have used the knowledge gained from learning how to recognize objects in the images in order to learn how to recognize activities in videos. Transfer learning is method in which we leverage the knowledge from a pretrained model and use it to solve a different but related problem [27].

3.3 Inception ConvNet Model

Inception is a range of models designed and trained by Google for classifying objects in the given image, using the ImageNet database. Inception V1 (GoogLeNet) architecture was introduced by Google in the ILSVRC 2014 challenge. The model has an error rate of 6.67% on the ImageNet dataset [28]. Inception model has increased depths and a new level of organization in the form of the inception module. The inception module consists of 3 different kernel sizes rather than 1. It uses 3×3, 5×5 and 1×1 filters, and max pooling function is also performed as shown in Fig. 1. The outputs from each convolution will be concatenated and sent to the next inception module. 5×5 and 3×3 convolutions are computationally very expensive. Thus to make the computational process feasible, dimensionality reduction was applied onto the image before it underwent through convolution. A 1×1 filter-sized CNN was introduced between 3×3 and 5×5 CNNs. GoogLeNet has nine inception modules

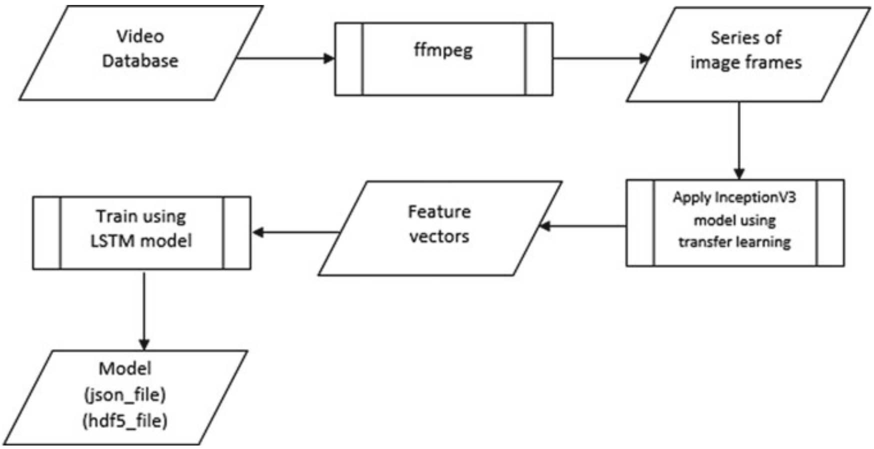


Fig. 1 Data flow diagram

stacked linearly which leads to vanishing gradient problem. To overcome that, two auxiliary classifiers were added which applies softmax on the outputs of inception modules and computes auxiliary loss.

These losses are added to total loss, so total loss is sum of real loss and two auxiliary losses.

$$\text{Loss}_{\text{total}} = \text{Loss}_{\text{real}} + 0.3 * \text{loss1}_{\text{aux}} + 0.3 * \text{loss2}_{\text{aux}} \tag{1}$$

0.3 value was described in the paper [28].

Inception V2 and V3 architectures were introduced back in 2015 with modifications to overcome representational bottleneck. The model has an error rate of 3.5% on the ImageNet dataset. Various principles were considered to improve accuracy and reduce computational burdensomeness more [29]. To eliminate bottleneck representation, an efficient grid size reduction strategy was used which consists of two parallel stride pooling and convolution which were concatenated before going into the next module. A 5 × 5 filter was factorized into two 3 × 3 filters, which proved to be less expensive, described in Fig. 2. More the factorization, cheaper the model would be; hence, asymmetric factorization was introduced. A 3 × 3 filter was factorized into 1 × 3 and 3 × 1 filters which proved to be 33% cheaper than usual as shown in Fig. 3. All the figures describe the three important modules of inception [30]. Inception V3 includes all the upgrades of V2 and in addition an RMS optimizer, batch normalization in the auxiliary classifier and label smoothing [29, 30].

Fig. 2 Inception block A

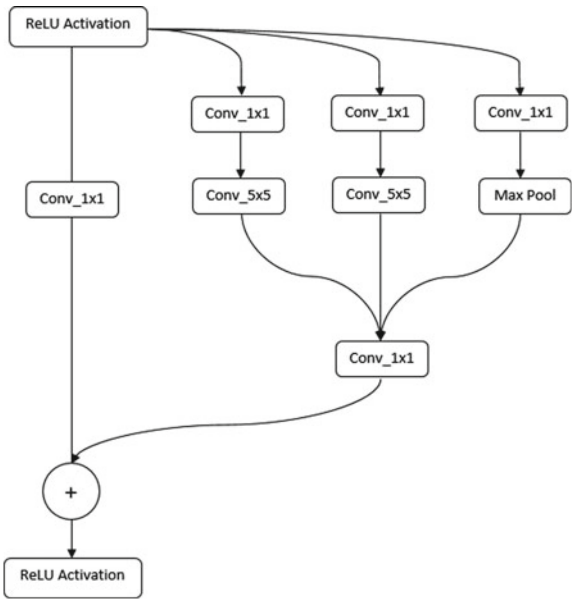
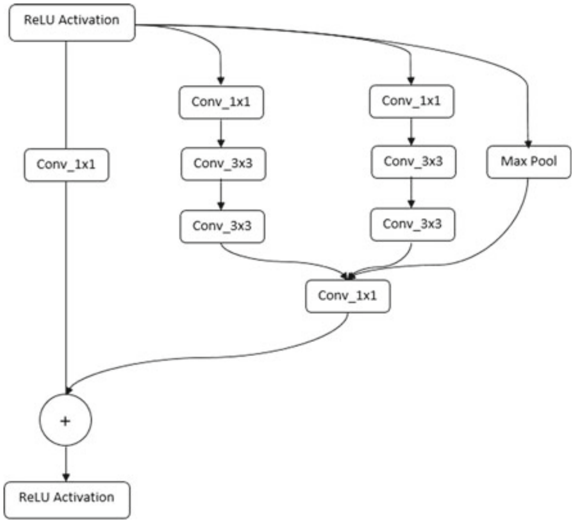


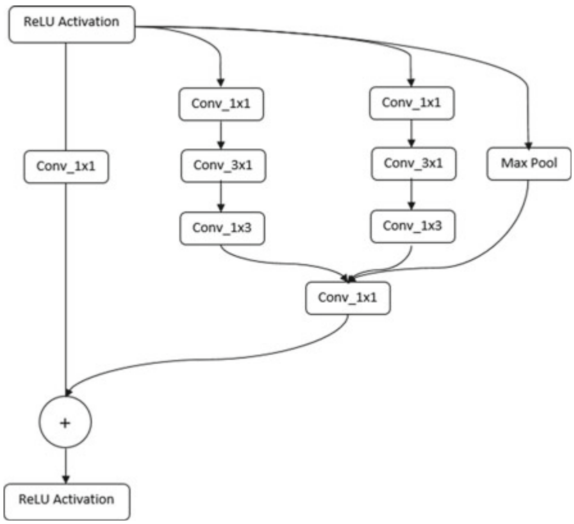
Fig. 3 Inception block B



The flow of images in Inception: Image is passed through 3 simple 3 × 3 CNN filters, and then it is introduced to the Inception module. There are three inception modules as shown in the above figures. Images are introduced to the module alphabetically.

Inception is known to be highly accurate for image recognition. In this model, modified version of the Inception V3 model is used in such a way that the second last layer of the model is taken as an output instead of the last layer. In other words, this model learns only up to high-level features instead of classifiers (Fig. 4).

Fig. 4 Inception block C



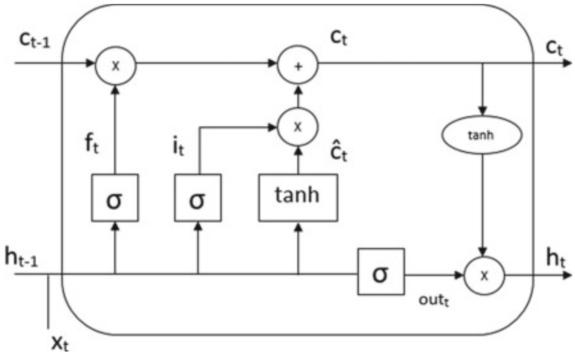


Fig. 5 LSTM neural cell

3.4 *Long Short-Term Memory*

Simply put, where RNN can remember only the recent past, LSTM can find which instances of the past moments are important and which are not, thus capitalizing on only that information which proves to be helpful [7]. LSTM comprises a cell state and its various gates. The cell state is the most important part of an LSTM cell. It transfers information from one time step to another and acts as a conveyor belt for important information learned from the previous inputs. The gates are the components that help the cell state to update in each time step and in the calculation of hidden states as shown in Fig. 5.

The forget gate decides which information is useful and which is not. Input from the previous hidden state and the current input is concatenated and then passed through a sigmoid function where value comes out between 0 and 1, 1 being most useful and 0 not useful or forgettable.

$$f_t = \sigma(W_f \cdot ([h_{t-1}, x_t]) + \text{bias}_f) \tag{2}$$

Input gate is used to update the cell state by using the current input, and tanh function is applied to the concatenated output of hidden state and input, which squishes the value between 1 and −1. A sigmoid operation is also performed on the same input to regulate the network and then performing pointwise multiplication on the output of the tanh and sigmoid functions.

$$i_t = \sigma(W_i \cdot ([h_{t-1}, x_t]) + \text{bias}_i) \tag{3}$$

Cell state updation contains two operations; first, pointwise multiplication of cell state with the forget vector for dropping values near zero, in order to discard the useless values, and second, performing pointwise addition of cell state and input vector that updates the value of cell state which the neural network finds relevant.

$$\hat{C}_t = \tanh(W_c \cdot ([h_{t-1}, x_t]) + \text{bias}_c) \tag{4}$$

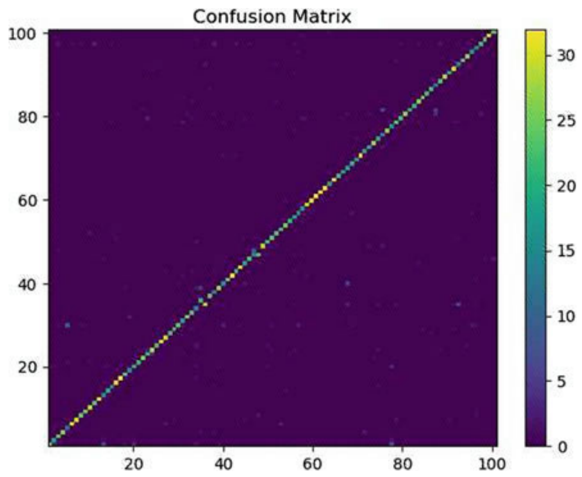


Fig. 6 Confusion matrix

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (5)$$

The output gate is used to compute hidden states for the next input. A pointwise multiplication is performed between the concatenated results of the input vector and hidden state; furthermore, the result is evaluated using a sigmoid function and the output of the tanh function is applied on the newly updated cell state. The hidden state calculated at the end is a product of output gate and current cell state provided that the values of cell state have been rescaled by tanh function.

$$\text{out}_t = \sigma(W_o \cdot ([h_{t-1}, x_t]) + \text{bias}_o) \quad (6)$$

$$h_t = \text{out}_t * \tanh(C_t) \quad (7)$$

To summarize, a forget gate decides which information is useful, an input gate decides which information should be added to the current cell state and the output gate decides what the next hidden state should be (Figs. 6 and 7).

4 Experimentation

This section provides us with the results and benchmarks of the proposed model for the UCF-101 dataset [31]. The dataset is divided into two parts, training set (80%) and testing set (20%). Further, the training dataset is divided into two parts, training set (60%) and validation set (20%).

Before passing each frame of video through a deep learning model, we convert each image into a NumPy array. Here, the image has shape attributes, i.e., width,

Fig. 7 Output example

height, channel, etc. It is henceforth reshaped into an array of shapes, constituting elements like height, width, shape. After that, we expand the dimension of the array by 1°. Then, it is converted from RGB to BGR format and zero-center by mean pixel.

Instead of using the whole video for analysis, we use only 40 equally distributed frames of each video.

This model works by passing each frame of the video, respectively, through a modified Inception V3 model which gives a 2048 length feature vector. This vector contains high-level spatial features of that image. These features are then passed to the LSTM layer which generates a spatiotemporal feature vector. This processed final feature vector is then given to fully connected layers, respectively, starting from a dense layer which constitutes of 512 neurons. The dropout rate is set to 0.2, for preventing overfitting of the data. The last layer is a dense layer of 101 neurons which then results in the final classification required. Before passing the feature vector into LSTM as an input, we reshape it from batch size, seq len to batch size, time steps, seq len. The dataset used for training is UCF-101, and each activity has 100–200 videos for training. It took around 65 min to train the model on the Intel HM370 chipset with 8 GB RAM and NVIDIA Geforce GTX 1050Ti graphics processor (Table 1).

Table 1 Model comparison

Different models	Accuracy (UCF-101) (%)
PoTion: pose motion representation [32]	98.2
Spatiotemporal convolution	97
Temporal 3D ConvNets	92.3
CNN + LSTM model (our model)	94.2