International Journal of Research and Reviews in Applied Sciences And Engineering (IJRRASE) Vol 8. No.1 – 2016 Pp-45-50

available at : www.ijcns.com
ISSN: 2231-0061

OBJECT TRACKING AND ACTION RECOGNITION VIA NEURAL NETWORKS WITH SVM CLASSIFIER

K. ABIRAMI, M.E Applied Electronics,
K.R.VINOTHINI, Assistant Professor of ECE Dept,
A.V.C College Of Engineering

ABSTRACT --In this paper a fully automated deep model, which learns to classify human actions without using any prior knowledge, is proposed. Video tracking and human action recognition is gaining interest from many computer vision researchers because of its wide variety of potential applications. Human actions recognition a challenging task in computer vision is addressed with neural networking scheme. For instance: surveillance, advanced human computer interaction, content-based video retrieval, or athletic performance analysis. In this research, we focus to recognize some human actions such as waving, running, etc. Kalman filtering process is applied after background registration. Morphological processing is applied to the resulting filtering process. Video tracking is done through back propagation neural networking. For classifying different actions of a person SVM classifier is used. Tracking and the subsequent classification process are discussed separately to focus on the novelties of recent research.

KEYWORDS---Video tracking and human action recognition, Neural Networking, SVM classifier.

I.INTRODUCTION

Video tracking is the process of locating a moving object (or multiple objects) over time using a camera. It has a variety of uses, some of which are: human-computer interaction, security and surveillance, video communication and compression, augmented reality, traffic control, medical imaging and video editing. Video tracking can be a time consuming process due to the amount of data that is contained in video.

The objective of video tracking is to associate target objects in consecutive video frames. The Association can be especially difficult when the objects are moving fast relative to the frame rate. Another situation that increases the complexity of the problem is when the tracked object changes orientation over time. For these situations video tracking systems usually employ a motion model which describes how the image of the target might change for different possible motions of the objects.

Human activity recognition can serve many application areas, ranging from visual surveillance to Human Computer Interaction (HCI) systems. Visual Surveillance uses it as the video technology is becoming progressive, the visual surveillance systems have undertook a rapid development process, and have more or less become a part of our daily routine [1]. Human activity understanding can help to find

fraudulent events such as burglaries, snatching, thefts , violent actions, etc. and can serve to track patients who need special attention (like identifying the well-being of a lonely person, detecting a falling person). Since, ubiquitous computing has increased the presence of HCI systems almost everywhere. A recently evolving thread is in the area of electronic games where we imitate the actions of a real world human being to create his avator on system. Due to substantial decrease in the cost of video capturing devices, videos have become a considerable part of the today's personal visual data. Automatic recognition of those data files, together with movies and other video clips helps information retrieval.

II. DESIGN OF THE NEURAL NETWORK FOR OT

In existing system, video tracking is based on two layer convolution networks. The two layer convolution algorithm does not predict the error completely. In proposed system, the video tracking is done by neural networks back propagation algorithm. Back propagation is a three layer algorithm.

Design of a neural network involves the selection of its model, architecture, learning algorithm, and activation functions for its neurons according to the need of the application. The objective of our application is to locate a specific airplane in the frames grabbed from a movie clip playing at the speed of 25frames/second.

A. Selection of the ANN Model

The application, at hand, for which a neural network is to be designed, is a kind of function approximation problem. It may be noted that a back-propagation neural network (BPNN) with one (or more) sigmoid-type hidden layer(s) and a linear output layer can approximate any arbitrary (linear or nonlinear) function [2]. The number of hidden layers is normally chosen to be only one to reduce the network complexity, and increase the computational efficiency [3]. Thus, a BPNN is selected for the application at hand, and it consists of three layers: one input layer (of source nodes), one hidden layer (with tangent hyperbolic sigmoid activation function), and one output layer (with pure linear activation function), as shown in FIG 1.

B. Input Layer

The input layer of a neural network is determined from the characteristics of the application inputs. There are 320x240 (i.e. 76800) pixels in each frame coming from a movie (or camera). Each pixel contains three elements (red, green, and blue components). Thus, the total number of elements in a frame is 3x76800 (i.e. 230400). If all these elements are directly put into the neural network, it will be almost impossible to process the image in real-time with a standard PC. Therefore, a preprocessing stage must be incorporated to reduce the size and dimensionality of the input pattern. Firstly, the color frame is converted into a gray level image, using the following expression for every pixel [4]:

$$y = (0.212671)r + (0.71516)g + (0.072169)b \quad (1)$$

where y is the gray level value of the pixel in the output image, and r, g, and b are the red, green, and blue components of the pixel in the input color image, respectively. The values of y, r, g, and b are in the range [0, 255]. Secondly, the gray level image is down-sampled simply by extracting 1st, 5th, 9th, etc. rows and columns, while skipping all other rows and columns in the image. The size of the image is now reduced to 80x60 (reduction factor of 4 with respect to both the number of rows and the number of columns). Thus, the total number of elements is reduced from 230400 to only 4800 with a total reduction factor of 48 (i.e. 3x4x4). Thirdly, the data of the down-sampled image is normalized, so that the value of each element can be in the range

[0.0, 1.0], instead of [0, 255] for fast convergence during the training phase of the ANN. The normalization is done using (2):

$yn = y / 255$ (2)

Where *y*n is the normalized value.

Finally, the resulting image matrix is reshaped to form a standard pattern (column-vector), by concatenating the rows of the image matrix, and then transposing the large row-vector to make it a 4800-element column-vector. Therefore, the number of input nodes in the proposed BPNN becomes 4800.

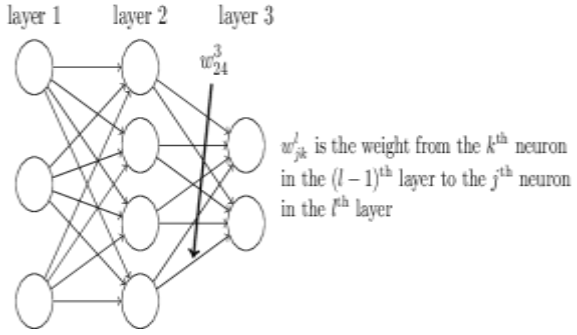


FIG 1. The 3 Layers of Back Propagation Algorithm

C. Hidden Layer

Hidden layer automatically extracts the features of the input pattern [3], and reduces its dimensionality further. There is no definite formula to determine the number of hidden neurons. In this research, a hit-and-trial method was used to identify the number of neurons in the single hidden layer. It was found that only 50 hidden neurons could accomplish the task at hand quite reasonably. The tangent hyperbolic activation function was chosen for the hidden layer after comparing its converging results with those of the logistic sigmoid function. The tangent hyperbolic function and its fast approximation are given in (3):

$$a_{i1} = \tanh(n_{i1}) = \frac{e^{n_{i1}} - e^{-n_{i1}}}{e^{n_{i1}} + e^{-n_{i1}}} \cong \frac{2}{1+e^{-2n_{i1}}} - 1 \qquad (3)$$

where $ai1$ is ith element of **a**1 vector containing the outputs from the hidden neurons, and $ni1$ is ith element of **n**1 vector containing net-inputs going into the hidden neurons. **n**1 vector is calculated as:

$$\mathbf{n}1 = \mathbf{W}10\mathbf{p} + \mathbf{b}1 \quad (4)$$

where **p** is the input pattern, **b**1 is the vector of bias weights on the hidden neurons, and **W**10 is the weight matrix between 0^{th} (i.e. input) layer and 1st (i.e. hidden) layer. Each row of **W**10 contains the synaptic weights of the corresponding hidden neuron.

D. Output Layer

The output layer of the network is designed according to the need of the application output. Since the output of the neural network is expected to produce the row and column coordinates of the target (with respect to the top-left pixel position), the number of output neurons will be two. Since the frame size is 320x240, the values of the row and column coordinates of the target will be in the ranges [0, 240] and [0, 320], respectively. Thus, the pure linear activation function is selected for the output neurons, and expressed as:

$$\mathbf{a}2 = \mathbf{n}2 \quad (5)$$

where **a**2 is the column-vector coming from the second output layer, and **n**2 is the column-vector containing the net inputs going into the output layer. **n**2 is calculated as:

$$\mathbf{n}2 = \mathbf{W}21\mathbf{a}1 + \mathbf{b}2 \quad (6)$$

where **W**21 is the synaptic weight matrix between the first (i.e.hidden) layer and the second (i.e. output) layer, and **b**2 is the column-vector containing the bias inputs of the output neurons. Each row of **W**21 matrix contains the synaptic weights for the corresponding output neuron.

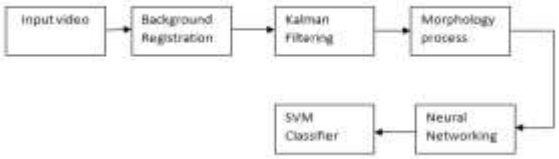
The designed architecture of the proposed BPNN is shown in Fig. 1.

The dimensions of the vectors and matrices are shown under their names, where *m*0 (= 4800) is

the number of input nodes, $m1$ (= 50) is the number of hidden neurons, and $m2$ (= 2) is the number of output neurons.

III. ACTION RECOGNITION

The following FIG.2 shows the operation of video tracking and action recognition



Action recognition is performed by using SVM classifiers. SVM means support vector machine classifiers. SVM optimization produces maximum predictive accuracy while automatically avoid over fitting of trained data. SVM projects the data into kernel space. Then its built a linear model in this kernel space. SVM standard tools for data mining and machine learning.

SVM has a higher generalization capability and provides high accuracy. SVM creates a hyperplane for classifying the data into a high dimensional space for separating the data with different labels. On each side of the hyperplane created initially, two separate hyperplanes are created. SVM tries to find that hyperplane which maximizes the distance between the two parallel hyperplanes. A wisely done separation means largest distance between the hyperplane and the nearest training data point of any class.

The query video sequence is given as input video. For the given input sequence background registration process is implemented to collect the information of various background features in each frame. After background registration process kalman filtering is done to avoid the unessential information in the video sequence. Morphological processing is applied after filtering. Applied Morphological techniques probe each video frame with a small shape or template called a structuring element. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbourhood of pixels. Neural Networking is used for tracking purpose. Knowing grey level difference between target and estimated region containing the tracked object, we employ a Neural Network to evaluate the corrective vector which is used to find the actual position of the target. Finally SVM classifier is used for action recognition. SVM classifier classifies a spatio-temporal feature descriptor of a human figure in a video, based on training examples.

A. Training phase

In the training phase all the features of the taken dataset is collected. The step by step process is explained as follows.

B. Training Video collection

The actions to be recognized must be trained as features in prior to the testing video. Videos containing such data must be collected. These dataset maybe either previously collected dataset like Weizmann or can be collected instantly.

C. Background registration process

The goal of background registration step is to construct reliable background information from the video sequence. According to FDM, pixels not moving for a long time are considered as reliable background pixels. The procedure of Background Registration can be shown as where SI is Stationary Index, BI is Background Indicator, and BG is the background information. The initial values of BI, BG, and BI are all set to "0." Stationary Index records the possibility if a pixel is in background region. If is high, the possibility is high; otherwise, it is low. If a

pixel is "not moving" for many consecutive frames, the possibility should be high. When the possibility is high enough, the current pixel information of the position is registered into the background buffer. In addition, Background indicator is used to indicate whether the background information of current position exists or not.

D. Kalman filter

Kalman filtering eliminates all the unessential information of a video frame. There are two types of equations for the Kalman filter. The first are the prediction equations. These equations predict what the current state is based on the previous state and the commanded action. The second set of equations known as the update equations look at your input sensors, how much you trust each sensor, and how much you trust your overall state estimate. This filter works by predicting the current state using the prediction equations followed by checking how good of a job it did predicting by using the update equations. This process is repeated continuously to update the current state.

IV. SIMULATION OUTPUT AND RESULTS

The following FIG.3 shows the input video sequences

FIG.3 Input video sequence

The object tracking and action recognition output is shown in the following FIG.4

FIG.4 Object tracking and action recognition output

V.CONCLUSION

In this paper, the performance of human action recognition and video tracking based on depth and RGB video sequence is analyzed. In Depth based HAR, we took the advantage of depth information in feature description because of its insensitiveness toward illumination changes, method based on global features perform better than local features based. Our two-steps scheme automatically learns spatiotemporal features and uses them to classify the entire sequences one direction of our future work is to use the learned action bases for image tagging, so that we can explore more detailed semantic understanding of human actions in images.

REFERENCES

[1] Paul M., Haque S. M. E., Chakraborty S., "*Human detection in surveillance videos and its applications - a review*", Springer EURASIP Journal on Advances in Signal Processing, 2013, pp.1-16.

[2] Howard Demuth, Mark Beale, "*Neural Network Toolbox for Use with MATLAB": User's Guide (v. 4)*, The Mathworks, Inc., 2001.

[3] Simon Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Ed., Pearson Education, Delhi, 1999.

[4] A.Krizhesvsky, I. Sutskever, and G. E. Hinton, *"Imagenet classification with*

deep convolutional neural networks," in NIPS, 2012, pp. 1106-1241.

[5] Gers, F.A., Schraudolph, N.N., Schmidhuber, J.: "*Learning precise timing with LSTM recurrent networks."* Journal of Machine Learning Research 3, 115–143 (2003)

[6] Ikizler, N., Cinbis, R., Duygulu, P.: "*Human action recognition with line and flow histograms*." In: International Conference on Pattern Recognition, pp. 1–4 (2008)

[7] Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: "*What is the best multistage architecture for object recognition?*" In: International Conference on Computer Vision, pp. 2146–2153 (2009)

[8] Jhuang, H., Serre, T., Wolf, L., Poggio, T.: "*A biologically inspired system for action recognition."* In: International Conference on Computer Vision, pp. 1–8 (2007)