International Journal of Current Advanced Research
ISSN: O: 2319-6475, ISSN: P: 2319-6505, Impact Factor: SJIF: 5.995
Available Online at www.journalijcar.org
Volume 7; Issue 3(B); March 2018; Page No. 10559-10566
DOI: http://dx.doi.org/10.24327/ijcar.2018.10566.1794

Research Article

VEHICLES AND PEDESTRIANS CLASSIFICATION USING RADIAL SHAPE DESCRIPTORS FOR VIDEO SURVEILLANCE

Akinyokun O. C[1]., Akintola K. G[2]., Angaye C. O[2] and Arekete S. A[3]

[1]Department of Computer Science, Federal University of Technology, Akure, Ondo State, Nigeria
[2]Department of Computer Science, Niger Delta University, Wilberforce Island, Nigeria
[3]Department of Computer Science, Redeemer's University, Ede, Nigeria

A R T I C L E I N F O

Article History:
Received 4th December, 2017
Received in revised form 16th January, 2018 Accepted 05th February, 2018
Published online 28th March, 2018

Key words:
Object Classification, Kernel Density, Neural Network, Segmentation, Normalized,

A B S T R A C T

Moving object classification is a requirement in smart visual surveillance systems as it allows the system to know the kind of object in the scene and be able to recognize the actions the object can perform. This paper presents a neural-network machine learning approach for real time object classification in videos. This is necessary for the higher layer surveillance system to detect actions being performed by the moving objects. Fast kernel density estimation background subtraction algorithm is used for object segmentation. Radial distance signal features are then extracted from the silhouettes of the detected objects. The radial distance signals features are then normalized and fed into a multilayer feed-forward neural network to classify the object as human or vehicle. The neural network is trained using hybrid of genetic and back-propagation algorithm using features extracted from objects detected from real life video surveillance of human and vehicles and recognition was performed on some selected test data from the video objects. The recognition rate of 98.52% is achieved over back propagation algorithm and 99.13% with genetically trained neural network. is recorded. A comparison of this classifier with some other classifiers in terms of recognition accuracy shows a better performance than K-NN and K-Means classifiers while having the same recognition accuracy with support vector machines.

INTRODUCTION

Object classification in videos is an important requirement in surveillance systems as it aids understanding of the intentions or actions that the object can perform. For instance human beings can sit, walk, run or fall while vehicles can move, run, over-speed or crash. Object classification is a challenging task because of various object poses, illumination and occlusion. This research was motivated by the need to use the result of the object classification in higher layer of surveillance system for action recognition. In such systems, a higher degree of recognition accuracy and higher response time is highly required. A parallel- based kernel density estimation algorithm was used for an adaptive background subtraction [1, 2]. The algorithm was applied to both outdoor and indoor environments and was found very fast and robust. Unlike the previous systems in the literature that uses motion information, our system is based on shape information. It has been recognized in [3] that the shape information from silhouettes extracted from the segmented region is invariant to color and texture changes. Moreover, computing the distance signals is very fast and demands small storage space which is good qualities for a surveillance system which requires fast operation and storage minimization. Connected component analysis is carried out using morphological opening and closing operations. The resulting blobs extracted are then used to extract the feature vectors which were gotten by calculating the distance from the centroid of the object to the contour outline starting at right hand side and moving in anticlockwise direction. The feature vector is then normalized by dividing the vector by the sum of the lengths. These vectors are then used to train a neural network using supervised learning approach.

*Corresponding author: **Akinyokun O. C**
Department of Computer Science, Federal University of Technology, Akure, Ondo State, Nigeria

Related works

Many research works have been carried out in literature on object classification using neural networks [3-8]. In many of the researches, the background subtraction adopted is not robust to quazzi-stationary backgrounds, sudden illumination changes and efficient coding. Shape-based and motion-based approach are used for object classification. Shape-based classification is the use of an object shape properties such as the bounding rectangle, area and silhouette of detected object regions. The authors in [9] classified moving object blobs into general classes such as 'humans' and 'vehicles' using viewpoint-specific neural networks and trained for each camera. Each neural network is a standard three-layer network. Learning in the network is accomplished using the back propagation algorithm. Input features to the network are a mixture of image-based and scene based object parameters namely image blob dispersedness (perimeter^2/area (pixels)); image blob area (pixels); apparent aspect ratio of the blob bounding box; and camera zoom. There are three output classes, namely human, vehicle and human group. This approach fails to discriminate object with similar dispersedness. In [3], a stereo-and neural network approach for pedestrian detection in videos is presented. The motivation for the project is to develop a surveillance system that can avoid dangerous situations. This is achieved using a stereo-based segmentation and neural network based recognition. This system performs well in pedestrian detection especially stationary pedestrians but fails to incorporate motion cues into the system which should have enhanced the performance. The authors in [5] used Neural Network approach for the recognition of human motion on a still camera. It is noted that task to classify and identify objects in the video is difficult for human operator. Object is detected using background subtraction technique. The detected moving object is divided into 8x8 non-overlapping blocks. The mean of each of the blocks is calculated. All mean value is then accumulated to form a feature vector. A neural network is trained using the generated feature vectors. Experiment performed shows a good recognition rate but the object detection algorithm used cannot work under a quasi-stationary background. Not only this, the computational time of the features is time consuming which is a problem to surveillance systems.

Motion method uses the object's motion characteristics to distinguish the object. The authors in [4] used the variance of compactness of the object to classify target object as single person or group of persons or vehicle. It is noted that vehicles are more consistent in their motion because they are rigid objects whereas humans shift some parts forward to maintain balance. Base on this, the variance of motion direction is employed to measure motion consistency. These features are obtained from the optical flow of the motion which can be computationally expensive. The problems addressed in this project are on coding of the silhouettes of the actors that can be processed in real time and that will require moderate storage space. Furthermore, silhouette is used to shape features that are robust to sudden illumination changes. The issue of better algorithm for background subtraction that can work in a quazi-stationay background is also addressed. A robust and fast object tracking algorithm is also used. Thus, this research focuses on fast background subtraction using parallel adaptive kernel density technique.

System Description

The operational procedure of the system is presented in Figure 1.

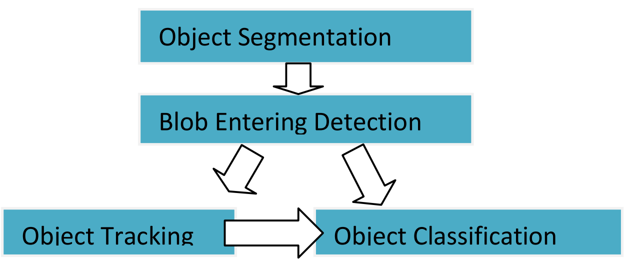


Figure 1: System description

Kernel Density Estimation (KDE) is the mostly used and studied nonparametric density estimation algorithm. The model is the reference dataset, containing the reference points indexed natural numbered and has been used in [10] for foreground detection. The algorithm assumed that a local kernel function is centered upon each reference point and its scale parameter (the bandwidth). The common choices for kernels include the Gaussian and the Epanechnikov kernel. The algorithm is presented as follows. Let $x_1, x_2, \dots, x_n \in R^d$ be a random sample taken from a continuous, univariate density *f*, KDE is given by:

$$\hat{f}(x,h) = \frac{1}{nh}\sum_{i=1}^{n} k\left(\frac{x-x_i}{h}\right) \qquad (1)$$

K (.) is the function satis fying : $\int k(x)dx = 1$ (2)

K (.) is refered to as the Kernel, *h* is a positive number, usually called the bandwidth or window width. The Gaussian Kernel is given by: $K_N = (2\pi)^{-\frac{d}{2}} \exp\left(-\frac{1}{2}r\right)$ (3)

where r is $\|x\|^2$ is a non-negative number called the norm of x which can be given as $x^T x$.

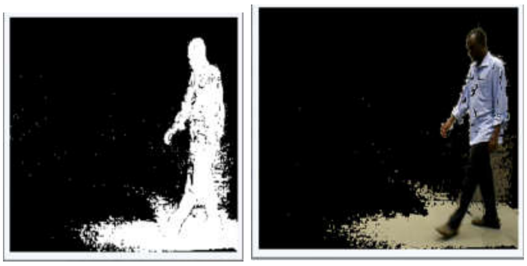
The Epanechnikov kernel is given by:

$$K_E = \begin{cases} -\frac{1}{2}c_d^{-1}(d+2)(1-r) & if\ r < 1 \\ 0 & otherwise \end{cases} \qquad (4)$$

where *d* is the dimension of feature space, c_d is the volume of the *d*-dimensional sphere. KDE for background modeling involves using a number of frames (training frames) to build the probability density of each pixel location. The adaptive threshold of each pixel is found after the construction of the histogram.

For every pixel observation, classification involves determining if it belongs to the background or the foreground (as shown in Figure 2). The first few initial frames in the video sequence (called learning frames) are used to build histogram of distributions of the pixel color. No classification is done for these learning frames, but for the subsequent frames depending on whether the obtained value exceeds the threshold or not. If the threshold is exceeded, background classification is done, otherwise foreground classification. Typically, in a video sequence involving moving objects, at a particular spatial pixel position a majority of the pixel observations would correspond to the background. Therefore, background clusters would typically account for much more observations than the foreground clusters. This means that the probability of any background pixel would be higher than that of a foreground pixel. The pixels are ordered based on their corresponding value of the histogram bin which relies on the adaptive threshold in the previous stage. The pixel intensity values for the subsequent frames are estimated and the corresponding histogram bin is evaluated with the bin value corresponding to the intensity determined.

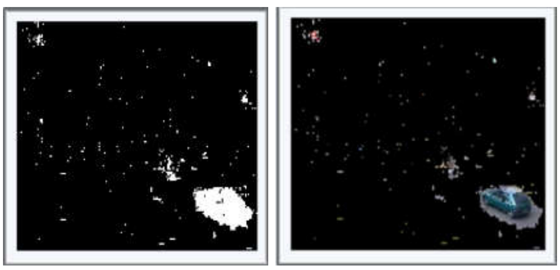
(a) background **(b)**Moving objects moved



(c) Noisy Mask with shadow **(d)** Colored foreground with shadow

(e) Foreground Mask with shadow removed and morphological operation performedHuman Segmentation process

(a) background **(b)**Moving objects moved



(c) Noisy Mask with shadow **(d)** Colored foreground with shadow

(e) Foreground Mask with shadow removed and morphological operation performed Vehicle Segmentation Process

Figure 2 Human and Vehicle Objects Detection Results

Spatio-Color Histogram Algorithm for Scalable Object Tracking

The proposed algorithm is composed of two stages. First, is the appearance correspondence mechanism. Once detected, appearance models are generated for objects appearing in the scene. The model is the estimate of probability distribution of colour of pixel. Multiple models are developed for a single object and used in subsequent frames to match the set of detected and target models. In the second phase, occlusion and object merge and separation are handled. The foreground object detected in previous stage is passed to the object tracker. This information is the appearance model of the object. A multi-part tracking algorithm is adopted and each silhouette was segmented into upper-body area and lower-body area to give a histogram of colures in Heu Saturated and Value (HSV) color space. This approach is good enough at discriminating individuals because of varying intensity in identical objects with similar color and occlusion. The approach makes use of the object color histograms of previous frame to establish a matching between objects in consecutive frame. The method is also able to detect object occlusion, separation and gives it appropriate label during and after occlusion.

Feature Extraction

Instead of segmenting the silhouettes into 8x8 non-overlapping blocks as shown in [5], radial features are directly calculated from the detected objects as shown in Figure 3. This captures the shape and a series of the features encodes the motion information.

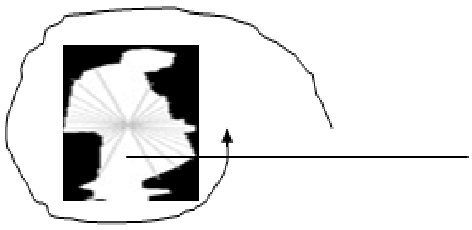


Figure 3 The radial Distance Shape Features Extraction

The centroid of the contour (c_x, c_y) is calculated using Equation (5). From the centroid, a pre-defined number of axes are projected outwards at specified regular angles to the nearest edges of the contour in an anti- clockwise direction as shown in Figures 3 and 4.

$$(c_x, c_y) = \frac{1}{n}\left(\sum_{t=1}^{n} x_t , \sum_{t=1}^{n} y_t\right) \quad (5)$$

where *x,y* denote the location of each pixel and *n* is the total number of pixels in the shape region.

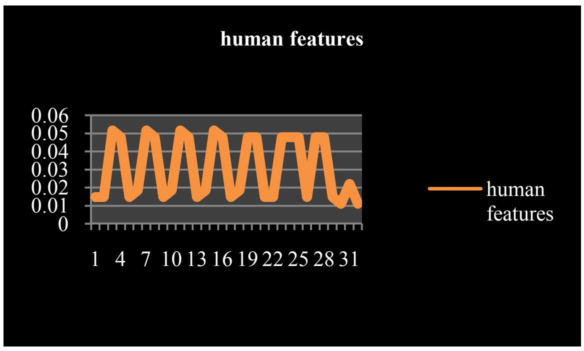


Figure 4a A typiocal 1D distance feature vector generated from pedestrian silhouete

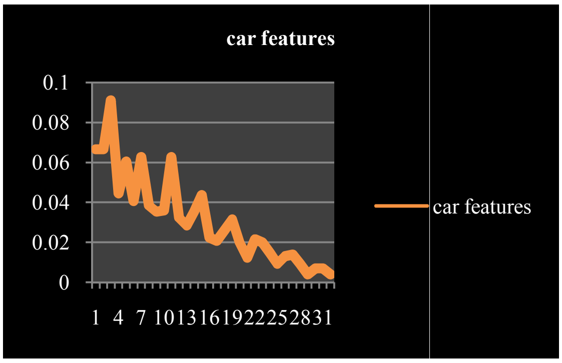


Figure 4b A typical 1D distance feature vector generated from vehicle silhouete

The distance from the centroid to its nearest edge along a predefined angle is then stored. This is done for a set of angular vectors. The dimension of each vector equals the number of axes being projected from the centroid. The vector is then normalized to ensure that the vector is scale invariant and the values range between 0 and 1. Figure 4a and 4b show the execrated features from human and vehicles respectively. The top panel of Figure 4a shows the extracted human silhouette and the lines projected from the centre of the object to its contour boundary. The collection of the length of these lines forms the feature vector. The line lengths are plotted in the lower plane. After normalization the lengths ranges between 0-1. This process is repeated for the vehicle silhouette as shown in Figure 4b. It can be seen that the features discriminate the objects. Appendix A shows sample data from pedestrians and vehicles.

Let S be the segmented object region within the frame, l_i be a line projected from the centroid to the object boundary at angle i to the horizontal line passing through the centoroid of the object, then the length of each line to the contour boundary of the object is given by:

$$l_i = \sum_{k=c}^{w} \delta(p(k,l)) \tag{6}$$

k and l are the co-ordinates in the x and y directions respectively c is the centre point and w is the contour boundary. l is given by ktan(θ), $\delta(.)$ is a binary function that returns 0 or 1.

$$\delta(p(k,l)) = \begin{cases} 1 & if\ p(k,l) \epsilon\ S \\ 0 & otherwise \end{cases} \tag{7}$$

where p(k, l) is the pixel value of the object.

The number of lines in each image containing an object as well as the number of neurons in the input layer is j where j = 1, 2, 3, …., n and n is given by n = 360/θmin where θmin is the smallest of the angles. Angular size of 10 degrees interval is used to obtain 36 regions beginning with 10^0. Thirty two (32) of these regions were selected as feature vectors.

Design of Neural Network for Object Classification

Artificial Neural Network (ANN) is a widely used methodology for machine learning. It emulates biological neural network with several interconnected neurons. However, ANN only utilizes a very limited set of concepts from its biological counterpart with one or more layer of neurons, which are fully or partially connected. Each connection between two nodes has a weight, which encapsulate the "knowledge" of the system. By processing existing cases with inputs and expected outputs, existing weights would be adjusted based on differences between actual and expected outputs.

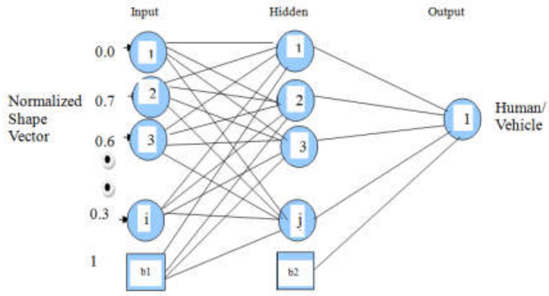


Figure 5 Neural Network Model for the Pedestrian/Vehicle Detection.

The general ANN learning process involves the computation of temporary outputs, comparison of outputs in the desired targets, adjustment of weights and process repetition (if necessary).The research being reported developed a model for training the neural network whose architecture is presented in Figure 5. Modeling was based on the data collected from the contour of the moving objects.

The back propagation neural network learns through a training epoch, which is described *for each input entry in the training data set by* feeding input entry data into the network (feed forward), initializing weights and checking output against desired value and feedback error (back-propagate) by calculating error gradients and updating the weights between both input and output hidden layers. The network parameters for the proposed model consist of an input layer with 32 neurons, one hidden layer with 4 neurons and an output layer with 1 neuron (MLP 32 : 32: 1) as shown in Figure 5. 1000 epochs were used with momentum and learning rate of 0.3 respectively. Based on random generators, the initial weights were initialized to small numbers less than 1. The architecture is gotten from experimental trials.

Back-propagation Learning Algorithm

Back-propagation algorithm is a supervised learning algorithm where an error function is defined (based on the training set) and minimized by adjusting the weights using hill climbing algorithm while the Mean Square Error (MSE) serves as the performance index. The error is calculated as the difference between the target (t) and the actual value (a) of the network output as follows:

$$MSE = \frac{1}{n}\sum_{k=1}^{n}(t_k - a_k)^2 \tag{8}$$

n is the number of training set based on multilayer perceptrons, back-propagation algorithm was used to adjust the weights and biases of the network in order to minimize the mean square error overall output and examples. The adjustment gives a generalization of the least mean square algorithm whereby an error function is defined to be minimized by using the gradient descent algorithm. The generalized delta rule calculates the error for the current input example and back-propagates it from layer to layer. The training algorithm is presented in [11].

Consider a neural network with one hidden layer (Figure 5) indexes i over output neurons, j, over hidden neurons and q

over input patterns. The MSE overall neurons and input patterns are given by:

$$MSE = \frac{1}{2}\sum_q \sum_i (d_i^q - o_i^q)^2 \quad q = 1, ,,, , N, i = 1 ,..., 32 \tag{9}$$

where d_i^q is the target output of neuron i on input pattern q, o_i^q is the actual output of neuron i for input patter q N is the total number of training patterns

The network is trained using the back propagation rule with sigmoid transfer function [11].

Genetic Trained Neural –Network

A Neuro-genetic training algorithm is adopted. The schematic diagram of the model is shown in Figure 6

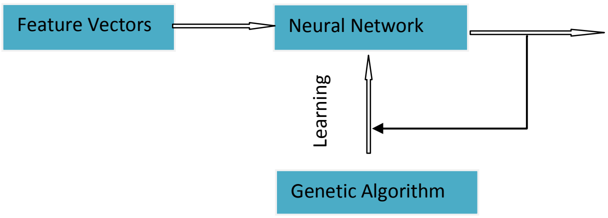


Figure 6 Neuro-Gentic Training model

In using Genetic Algorithm to solve this problem, the following three phases are used:

Initialization: Genetic algorithms operate on a set of strings. This set of strings is known as a population and is through the process of evolution to produce new set of strings. To start with, the initial population could be made up of chromosomes chosen at random or based on heuristically selected strings. The initial population is a wide varieties of structures. Population size affects the efficiency of the performance of a GA.

The first step in GA implementation is the determination of a genetic encoding scheme, that is, to denote each possible point in the problem's search space as a characteristic string of defined length. This is in order to be sure that GA will not only optimize network configuration but, in the meantime, genetic training will proceed on weight values. Figure 7 shows a typical chromosome generated from the network above. The genes are gotten from the weights at each of the nodes in the network. These genes are concatenated to form a chromosome. Several of these chromosomes are randomly generated from the network architecture to form the initial population of the solution space.

01011100000	1001010101	1111000010	100011110	10101010101	1111110000	....	10101010101

Figure 7 Sample Chromosome Encoding of ANN using strings.

In this research work, weight values between each layer of the multi-layer feed-forward neural network are simultaneously coded as one gene. Each of the gene which represents the weights of the neural network is coded using randomized binary numbers of length 10. The minimum value is zero while the maximum value is one. The number of bits to represent the value of a gene (weight) must satisfy:

$$v \geq log_2(\frac{\max of\ x_i - \min of\ x_i}{\Delta x} + 1);\ \mathrm{i} = 1,2.3........$$

where v is the string length, min $of\ \ x_i$ is the minimum value of the gene, max $of\ \ x_i$ is the maximum value of the gene and Δx is the error tolerance.

A chromosome represents one neural network structure and weight value of the neuron network is mapped to this very chromosome. The length of chromosome is obtained by concatenating the bits representing each gene.

Definition of Evaluation Function: The suitability of the solutions to the problem is determined. When GA is applied to solve a problem, the definition of the evaluation function to evaluate the problem-solving ability of a chromosome is important. Since Neural Network works by error minimization, the objective function adopted in this work is the *Mean-Square Error (MSE)* between the Neural Network output and the desired output. The reciprocal of the MSE is then used as the evaluation function. The initial strings are randomly generated and converted into real number by Equations 10 and 11.

$$w_n = min + \frac{max - min}{2^{v-1}} y_n \tag{10}$$

where $y_n = \sum_{j=0}^{v-1} 2^j b_j$ (11)

and w_n is the real weight, min is the minimum value of the gene, in this case 0 is used, max is the maximum value of the gene, in this case 1 is used and y_n is the equivalent decimal value of the gene given and b_j is the bit value.

These decoded values are then used to train the neural network using back-propagation. The error of desired computed output is calculated using Mean Squared Error as given by Equation (9). The fitness function used is the reciprocal of the Mean Squared Error as given by:

$$f(MSE) = \frac{1}{MSE} \tag{12}$$

Application of Genetic operators: Three GA operators adopted in this work are:

Selection: The fitness of the new offspring is calculated and sorted in the descending order. So chromosomes of highest fitness values are selected for the next generation. In this research work, the roulette wheel method is adopted for selection. The probability of selection is given by:

$$p_i = \frac{1}{\sum_{i=0}^{N} f_i} = \frac{f_i}{f_{sum}} \tag{13}$$

in which; f_i is the fitness value of individual i, f_{sum} is the total fitness value of population; P_i is the selective probability of individual. It is obvious that individuals with high flexibility values are more likely to be reproduced during the next generation.

Crossing: In this research work, one-point crossing is adopted. The specific operation is to randomly set one crossing point among individual strings. When crossing is executed, partial configuration of the anterior point and posterior point are exchanged, and gave birth to two new offspring

Mutation: As for two-value code strings, mutation operation is to reverse the gene values within a random number generated between zero and one.

Genetic control parameters dictate how the algorithm will behave. Changing these parameters can change the computational result. These parameters are population size, crossing probability, mutation probability and network termination condition. In this work population size N is 50, crossing probability P_c is 0.8, mutation probability P_m is 0.015, and network's terminative condition is MAXGEN of 100.

Experiment and Result

ANN models performances were measured by the mean relative percentage error, which represents the accuracy of prediction of the degree of scatter. The calculation of the relative error for each case in the testing set is given by:

$$A = \frac{(a-u)}{a} \times 100\% \tag{14}$$

a and *u* are the actual value and predicted value respectively.

The network was trained using 80 data items consisting of 40 vehicles and 40 pedestrians. The class vehicle was assigned the value 1 while the class human was assigned the value 0. The network was then trained using back-propagation algorithm and the weights from the training were recorded. 110 vehicle data and 223 human data were used to test the system. The thresholds for vehicle and human were set to 0.9 and 0.1 respectively. The value 1 and value 4 misclassifications were recorded for vehicles and human data respectively. Error in forecast was 1.46%, which implied accuracy of 98.54%. For genetically trained Neural Network, an error rate of 0.875 and accuracy of 99.125 is achieved.

The model has been applied in real life scenario to recognize vehicles and human. Figure 8 shows the performance of the algorithm on PETS2001 dataset. The PETS2001 dataset presents a challenging tracking problem in the field of automated visual surveillance which is a benchmarked surveillance data. Figure 9 shows the performance of the algorithm in an indoor environment. Figure 10 shows the performance of the algorithm in an outdoor environment at FUTA community. It is observed that the classifier successfully classified the objects irrespective of sizes and environmental conditions. This establishes its adaptability to various environmental changes.

Figure 8 Classifications of Vehicles in an Outdoor Environment

Figure 9 Classification of Humans in an Indoor Environment

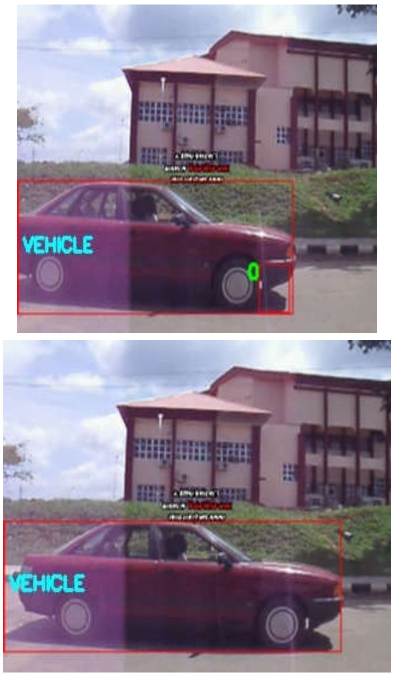


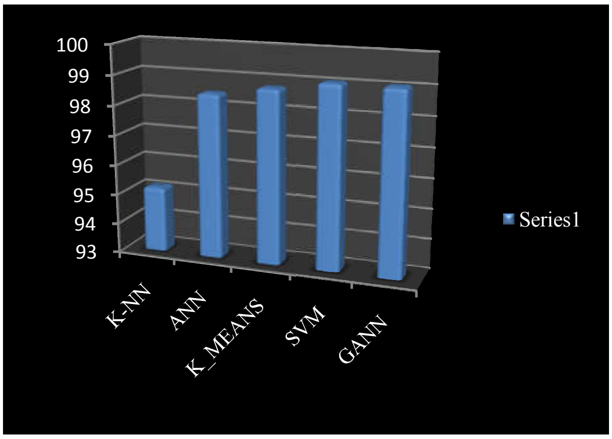
Figure 10 Classification of Vehicles in Outdoor Environment

Other classifiers have been tested to compare the performance of this research work. A support vector machine is trained using 80 data items consisting of 40 vehicles and 40 pedestrians. The class vehicle is assigned 1 while the class human is assigned 0. The model is then trained using Libsvm package [12] adopting Radial Basis Kernel (RBF) function. The support vectors' values from the training phase are then saved. For testing, 333 data items are used to test the system. Out of this, 110 are vehicle data while 223 are pedestrian data. Out of the 110 vehicles, all are classified correctly while for 223 human class data items, only three are misclassified. Error in forecast is 0.874% and the accuracy is 99.125%.

A K-MEANS clustering algorithm using 80 data items consisting of forty vehicles and 40 pedestrians is adopted. The class vehicle is assigned 1 while the class human is assigned 0. For testing, 333 data items were used to test the system. Out of this, 110 are vehicle data while 223 were pedestrian data. Out of the 110 vehicles, 1 was misclassified while for 223 human class data items, only three are misclassified. Error in forecast is 1.201% and the accuracy is 98.79%. .

A K-NN model with k=5 is adopted. The class vehicle is assigned 1 while the class human is assigned 0. For testing, 333 data items were used to test the system. Out of this 110 were vehicle data while 223 were pedestrian data. Out of the 110 vehicles, six were misclassified while for 223 human class data items, eight are misclassified. Error in forecast is 4.804% and the accuracy is 95.195% .

Figure 11 shows the bar chart of the classification accuracies of the classifiers. It is known from the figure that the Support Vector Machine (SVM) classifier has the same performance as Genetically Trained Neural Network) with the highest classification accuracy of 99.15% while K-Nearest Neighbor has the lowest performance of (95.20%).



Figre 11 Classifiers Recognition Accuracy

The performance of the committee of the classifiers is analyzed. Figure12 shows the result of using one classifier alone, two classifiers, up to five classifiers to detect vehicle or pedestrian in videos. Vehicle data, human data and combined human and vehicle data were analyzed using the decision level fusion of the classifiers. Each line shows the recognition rate of using at least n-classifiers (where n ranges from 1 to 5). When n is 1, that is when any one of the individual classifier recognizes the object, the highest recognition rate of 99.70% is observed and when n is 5, that is when all the 5 classifiers are required to recognize the object, the lowest recognition rate of 94.89 % is achieved. Thus the higher the number of classifiers needed to recognize the objects in the decision level fusion, the lower the recognition performance because we are required all the classifiers to recognize the object. If any of them fails to recognize the object, then the object cannot be recognised.

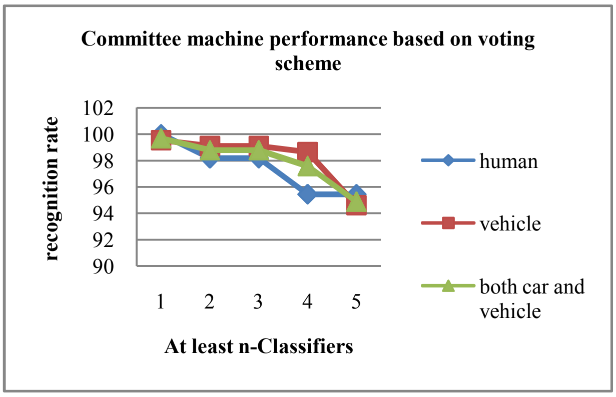


Figure 12 Committee Machine Performance Based on Voting

CONCLUSIONS

This paper has reported on the design and implementation of a neural network model and lightweight object shape descriptors for fast object classification in videos. Radial distance signal features were extracted from the silhouettes of the detected objects while the distance signals features were normalized and fed into a multilayer feed-forward neural network for the classification of the object as human or vehicle. The classification algorithm was implemented on a set of objects detected from real life video surveillance of human while network training and recognition were performed with selected objects. The algorithm performed well at discriminating between human and vehicular classes of objects with the recognition rate of 99.13%. The neural network on the shape descriptor also performed well at object classification. The comparison of this model with other models show a better performance in terms of recognition accuracy. This is because the neural network initial weights were selected using genetic algorithm. Future work will focus on the use of other classifiers such as radial basis network, recurrent network for better classification accuracy.

References

1. Akintola K. G, (2015). 'Development of Adaptive Video Analytic Framework for Real-Time Human Identity and Activity Recognition'. *PhD Thesis, Federal University of Technology, Akure.*
2. Akintola K. G. and Tavakollie A, (2011). 'Robust Foreground Detection in Videos Using Adaptive Colour Histogram Thresholding and Shadow Removal'. *Proceedings of the 7th International Symposium on Visual Computing, Las-Vegas, NV, September, 2011.*
3. Zhao L. and Thorpe C. E, (2000). 'Stereo and Neural Network-Based Pedestrian Detection'. *IEEE Transactions on Intelligent Transportation Systems, Vol. 1, No. 3, pages 148-154.*
4. Zhuo L. and Aggrawal J. K, (2006). 'Object Tracking in an Outdoor Environment using Fusion of Features and Cameras'. *Journal of Images and Vision Computing, Vol. 24, pages 1244-1255.*
5. Modi R. V. and Mehta T. B. (2011). 'Neural Network-Based Approach for Recognition of Human Motion Using Stationary Camera'. *International Journal of Computer Applications, Vol. 25, No. 6, pages 975-887.*
6. Teschioni A. Oberti F. and Regazzoni C, (1999). 'A Neural Network Approach for Moving Objects Recognition in Color Image Sequences for Surveillance Applications. *Proceedings of the Conference on Non-linear Signal and Images Processing (NSIP '99), Antalya, Turkey, pages 28-32.*
7. Khashman A, (2008). 'Automatic Detection, Extraction and Recognition of Moving Objects'. *International Journal of Systems Applications, Engineering and Development, Vol., 2, Issue 1, pages 43-51.*
8. Kavita P. and Patil S. V, (2012). 'Tracking and Counting Human in Visual Surveillance Systems. *International Journal of Electronics and Communication Engineering and Technology, Vol. 3, Issues 3, pages 139-146, www.iame.com/ijecet.asp*
9. Collins R. T, Lipton A. J, Fujiyoshi H and Kanade T, (2000). 'A System for Video Surveillance and Monitoring: VSAM Final Report'. *Technical Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, May 2000.* http://www.researchgate.net/publication/2464973_A_System_for_video_Surveillance_and_Monitoring:
10. Elgammal A. Duraiswami R, Harwood D. and Davis L. S, (2000). 'Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance'. *Proceedings of the IEEE, Vol. 90, No. 7, pages 1151-1163..*
11. Koprinska I, (2013). 'Lecture Notes on Artificial Neural Networks, *School of Information, University of Sydney. Accessed July 2013, http://sydney.edu.au/engineering/it/~comp3308/*
12. Chang C.C and Lin, C. J. (2001) LIBSVM: A Library for Support Vector Machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

How to cite this article:

Akinyokun O. C *et al* (2018) 'Vehicles And Pedestrians Classification Using Radial Shape Descriptors for Video Surveillance', *International Journal of Current Advanced Research,* 07(3), pp. 10559-10566.
DOI: http://dx.doi.org/10.24327/ijcar.2018.10566.1794

****