

A Novel GA-SVM Model For Vehicles And Pedestrial Classification In Videos

AKINTOLA K.G.

Department of Computer Science,
Federal University of Technology
Akure, Ondo State, Nigeria

Abstract

The paper presents a novel algorithm for object classification in videos based on improved support vector machine (SVM) and genetic algorithm. One of the problems of support vector machine is selection of the appropriate parameters for the kernel. This has affected the accuracy of the SVM over the years. This research aims at optimizing the SVM Radial Basis kernel parameters using the genetic algorithm. 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.

1 Introduction

Object classification in videos is the process of recognizing the classes of objects detected in videos. It 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 [2].

Recently, many research works have been carried out in literature on object classification in videos. Heikkila and Silven, [7], presents a real-time system for monitoring of cyclists and pedestrians. The classification algorithm adopted is learning vector quantization however, the classification accuracy obtained is low. The classification rate is low. The authors in [5] classified moving object blobs into general classes such as 'humans' and 'vehicles' using neural networks. 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 dispersednes (perimeter²/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 dispersednes. The authors in [9] used Artificial Neural Network approach for the classification 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 classification 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. In [2], a neuro-genetic model for moving object classification in videos is presented. A genetic model is used to obtain optimum weights of a neural network. The optimum weights are used later by the multilayer feed-forward neural network model to classify objects as human or vehicle. The model is compared with a neural network trained using back-propagation algorithm on a set of objects detected from real life videos. The neuro-genetic model outperforms with classification rate of 99.09% while the back-propagation neural network achieves the classification rate of 98.5% .

In [11] SVM is used to recognize facial expressions in videos. Automatic facial feature trackers are used to locate faces in videos. Features are then extracted

which are then supplied to SVM to recognize facial expression. The classification accuracy shows that SVM is capable of recognizing facial expressions. In [12], SVM is used to recognize detected objects in video for tracking purposes. A simple background subtraction technique is used to extract the object. Moment features of the detected objects are calculated and fed into SVM for classification. In [11] and [12] more classification accuracy could have been achieved by optimizing the parameters of the SVM. The hybrid of GA and SVM have been used in [13] for classification of satellite images. There is the need to improve the object classification in video surveillance applications.

Recently SVM have been reported as an efficient classifier. SVM is based on the statistical learning method based on structural risk minimization instead of the empirical risk minimization to improve the generalization ability of a model. It is however realized that the selection of appropriate kernel and kernel parameters can have great influence on the performance of the model [13]. The approach that have been used for optimizing the hyper-parameters of the SVM is the grid search which is always time consuming and does not perform well [13]. In this paper, a genetically optimized support vector machine is presented for human object classification in videos.

2 Support Vector Machines (SVM)

Support Vector Machines are a set of supervised learning methods used in classification and regression. This machine learning technique was proposed by Vapnik in 1995. The classification problem can be restricted to consideration of the two-class problem without loss of generality. In this problem, the goal is to separate the two classes by a function which is induced from available examples [16].

The motivation for SVM is to create a classifier that will work well on unseen examples, that is, generalizes well. The objective is to create a classifier that uses the structural minimization principle as against those that use empirical error minimization principle. Consider the example in Figure 1. There are many possible linear classifiers that can separate the data, but there is only one that maximizes the

margin (maximizes the distance between it and the nearest data point of each class). This linear classifier is termed the optimal separating hyper-plane. Intuitively, this boundary is expected to generalize well as opposed to the other possible boundaries.

Let m -dimensional training input vectors, x_i ($i=1, \dots, m$) belong to class 1 or 2 and the associated labels be $y_i = 1$ for class 1 and -1 for class 2. If these data are linearly separable, a decision function satisfying Equation 1 can be constructed.

$$D(x) = w^T x_i + b \quad (1)$$

where w is an m -dimensional vector and b is a bias term.

If the training data is linearly separable, then no training data satisfy:

$$w^T x_i + b = 0 \quad (2)$$

This hyper-plane should have the best generalization capability. As shown in Figure 1, the $+1$ and the -1 are the training dataset which belong to two classes. The plane H series are the hyperplanes that separate the two classes. The optimal plane H is found by maximizing the margin value $2/\|w\|$. Hyperplanes H_1 and H_2 are the planes on the border of each class and also parallel to the optimal hyper-plane H . The data located on H_1 and H_2 are called support vectors [8].

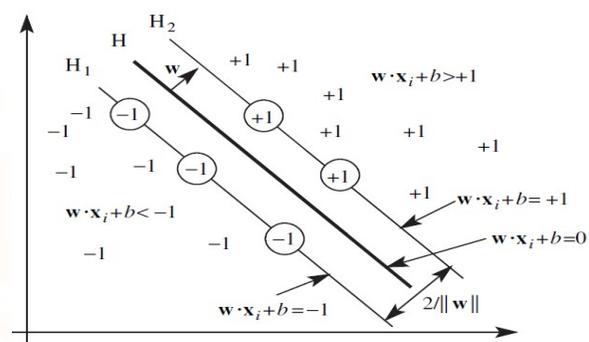


Figure 1 The SVM Binary Classification

Given a set of training samples x_1, x_2, \dots, x_m with labels $y_i \in \{-1, +1\}$, a hyper-plane $w \cdot x + b$ separates the data into classes such that:

$$w \cdot x_i + b \geq 1 \quad \text{if } y_i = 1 \quad (3)$$

$$w \cdot x_i + b \leq -1 \quad \text{if } y_i = -1 \quad (4)$$

$$\forall i.$$

These constraints can be expressed in compact form as:

$$y_i(w \cdot x_i + b) \geq 1 \quad (5)$$

which can be written as:

$$y_i(w \cdot x_i + b) - 1 \geq 0 \quad (6)$$

It has been shown that if no hyper-plane exists (because the data is not linearly separable), a penalty terms ξ_i is added to account for misclassifications. This can be translated to the following minimization problem:

$$\text{minimize: } |w|^2 / 2 + C \sum_i \xi_i \quad (7)$$

where C , the *capacity*, is a parameter which allows us to specify how strictly the classifier can fit the training data. This can be translated into the following dual problem:

$$\text{maximize: } \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (8)$$

subject to: $0 \leq \alpha_i \leq C$

$$\sum_i \alpha_i y_i = 0$$

which can be solved by standard techniques of quadratic programming. The vector w that defines the optimum separating hyperplane is obtained by:

$$w = \sum_{i=1}^m \alpha_i y_i \cdot x_i \quad (9)$$

The threshold b of the optimal separating hyper-plane is obtained by:

$$\sum_{j=1}^m \alpha_j y_j x_j \cdot x_i + b = y_i \quad (10)$$

$$b = y_i - \sum_{j=1}^m \alpha_j y_j x_j \cdot x_i \quad (11)$$

The prediction of new patterns is given by

$$\text{Class}(x_i) = \text{sign} \left(\sum_{i=1}^m \alpha_i y_i x_i \cdot x_k + b \right) \quad (12)$$

The training samples for which the Lagrange multipliers are non-zero are called support vectors. Samples for which the corresponding Lagrange multiplier is zero can be removed from the training set without affecting the position of the final hyperplane. The classification framework outlined above is limited to linear separating hyperplanes. It is possible however to use a non-linear hyper-plane by first mapping the sample points into a higher dimensional space using a non-linear mapping. That is, by choosing a map $\phi: \mathcal{R}^n \rightarrow \psi$ where the dimension of ψ is greater than n . A separating hyper-plane is then found in the higher dimensional space. This is equivalent to a non-linear separating surface in \mathcal{R}^n . The data only ever appears in our training problem in the form of dot products, so in the higher dimensional space the data appears in the form $\phi(x_i) \cdot \phi(x_j)$. If the dimensionality of ψ is very large, this product could be difficult or expensive to compute. However, by introducing a *kernel function* such that:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \quad (13)$$

Equation 13 can be used in place of $x_i \cdot x_j$ everywhere in the optimization problem and never need to know explicitly what ϕ is. The development of a SVM classification model depends on the selection of kernel function K . There are several kernels that can be used in SVM models. These include linear, polynomial, Radial Basis Function (RBF) and sigmoid function. The Radial Basis Kernel is given by:

$$K(x_i, x_j) = \exp \left(-\frac{1}{2\sigma^2} \|x_i - x_j\|^2 \right) \quad (14)$$

The RBF is by far the most popular choice of kernel types used in SVM. This is mainly because of their localized and finite responses across the entire range of the real x -axis. After solving for w and b , the class a test vector x_i belongs to is determined by evaluating $w \cdot x_i + b$ or $w \cdot \phi(x_i) + b$ if a transform to a

higher dimensional space has been used. It can be shown that the solution for w is given by:

$$w = \sum_i \alpha_i y_i x_i \quad (15)$$

Therefore, $w \cdot \phi(x_t) + b$ can be rewritten thus:

$$\sum_i \alpha_i y_i \phi(x_i) \cdot \phi(x_t) + b = \sum_i \alpha_i y_i K(x_i, x_t) + b \quad (16)$$

Thus, the Kernel function can be used rather than actually making the transformation to higher dimensional space since the data appears only in dot product form. The σ and C are the free hyper-parameters the users can supply. These two parameters has a great role to play in the performance of the SVM. Several choices have used to select optimal values of these parameters. These include cross-validation, Particle swarm optimization and genetic optimization. In this paper, however, the genetic program approach is employed.

3 Genetic Algorithm

3.1 Genetic Algorithms

Genetic Algorithm (GA) was developed by John Holland in 1975. The algorithm is based on the mechanics of the natural selection process (biological evolution). The concept of the GA is that the strong tends to adapt and survive while the weak tends to die out [17]. The technique is about generating a random initial population of individuals, each of which represents a potential solution to a problem [14]. The process begins by coding the genes in a form that represents the solution to the particular problem. In Holland's Genetic Algorithm, each feasible solution is encoded as a chromosome (string of zeros and ones) also called a genotype. Then initial population size is specified and a number of chromosomes of the population size are then created. Optimization is based on evolution, and the "Survival of the fittest" concept. Each chromosome is given a measure of fitness via a fitness (evaluation or objective) function. The fitness of a chromosome determines its ability to survive and produce offspring [17]. As reproduction

takes place, the crossover operator exchanges parts of two single chromosomes to produce children and the mutation operator changes the gene value in some randomly chosen location of the chromosome [10]. The process of evaluation, selection, and recombination is iterated until the population converges to an acceptable solution. Genetic Algorithms (GA) has been applied in various areas of computer vision such as weights optimization of artificial neural networks [2;10;14;15], video segmentation [4].

3 SVM parameter selection based on Genetic Algorithm

Using the Genetic algorithm, the two parameters C and σ of SVM model can be optimized. The process of optimizing these parameters is shown in Figure 2.

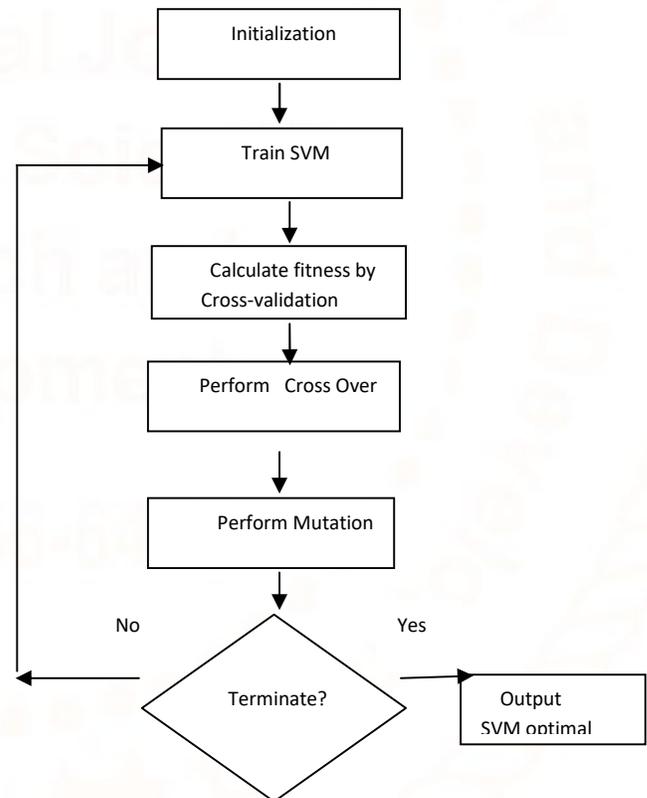


Figure 2 Flowchart of GA for SVM parameters optimization.

- a. **Initialization.** To start with, the initial population is made up of chromosomes chosen at random or based on heuristically selected strings. 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. The genes represent the value of σ and C and 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. Figure 3 shows a typical chromosome.

01010	10011
-------	-------

Figure 3: Sample Chromosome Encoding of C and σ of the RBF using strings.

In this research work, C and σ values of the RBF are simultaneously coded as one gene. Each of the gene is coded using randomized binary numbers of length 5. The minimum value is 0.1 while the maximum value is 100. The number of bits to represent the value of a gene must satisfy:

$$v \geq \log_2 \left(\frac{\max of x_i - \min of x_i}{\Delta x} + 1 \right); i=1,2,3,, \quad (17)$$

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 RBF. The length of chromosome is obtained by concatenating the bits representing each gene as shown in Figure 3.

b. Fitness Function by Cross-Validation.

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 SVM works by classification, the classification rate is used as the fitness function. The Accuracy is computed as:

$$\text{Accuracy} = \frac{\text{correctly}}{\text{total}} * 100 \quad (18)$$

c. **Perform 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:

$$p_i = \frac{1}{\sum_{i=0}^N f_i} = \frac{f_i}{f_{sum}} \quad (19)$$

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.

d. **Perform Crossover.** 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 this gave birth to two new offspring

- e. **Mutation.** As for two-value code strings, mutation operation is to reverse the gene values within a random number generated between zero and one.
- f. **Termination Criterion:** The termination condition is the maximum number of generations.

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 0.5, and network's terminative condition is MAXGEN of 100.

4 Object Classification using the proposed GA-SVM model

4.1 Data Acquisition

The data used for this research work are obtained from videos of moving objects taken in Nigeria roads. The moving objects are detected using background subtraction algorithm and the features are extracted from the silhouettes of the detected objects.

4.1.1 Object Segmentation

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 [6] 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:

$$\hat{f}(x, h) = \frac{1}{nh} \sum_{i=1}^n k\left(\frac{x-x_i}{h}\right) \quad (20)$$

$k(\cdot)$ is the function satisfying :

$$\int k(x)dx = 1 \quad (21)$$

$k(\cdot)$ is referred to as the Kernel, h is a positive number, usually called the bandwidth or window width. The Gaussian Kernel is given by:

$$K_N = (2\pi)^{-\frac{d}{2}} \exp\left(-\frac{1}{2}r\right) \quad (22)$$

where $r = \|x\|^2$

The Epanechnikov kernel is given by:

$$K_E = \begin{cases} -\frac{1}{2}c_d^{-1}(d+2)(1-r) & \text{if } r < 1 \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

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 the pixel belongs to the background or the foreground (as shown in Figure 4). 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.

4.1.2 Feature Extraction

The blobs extracted from videos used to extract the feature vectors which were are then normalized by dividing the vector by the sum of the lengths. These vectors are then used to train the support vector machine.

Instead of segmenting the silhouettes into 8x8 non-overlapping blocks as shown in [9], radial features are directly calculated from the detected objects as shown in Figure 4. This captures the shape and a series of the features encodes the motion information.

The centroid of the contour (c_x, c_y) is calculated using Equation 24. 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 4.

$$(c_x, c_y) = \frac{1}{n} (\sum_{t=1}^n x_t, \sum_{t=1}^n y_t) \quad (24)$$

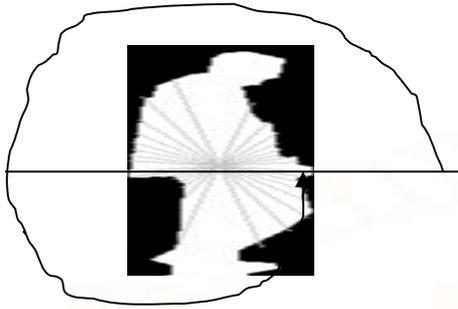


Figure 4: The radial Distance Shape Features Extraction

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 largest value in the vector will at this point be 1.0, which will be the longest of the axes projected. Figures 5a and 5b show the extracted features from human and vehicles respectively.

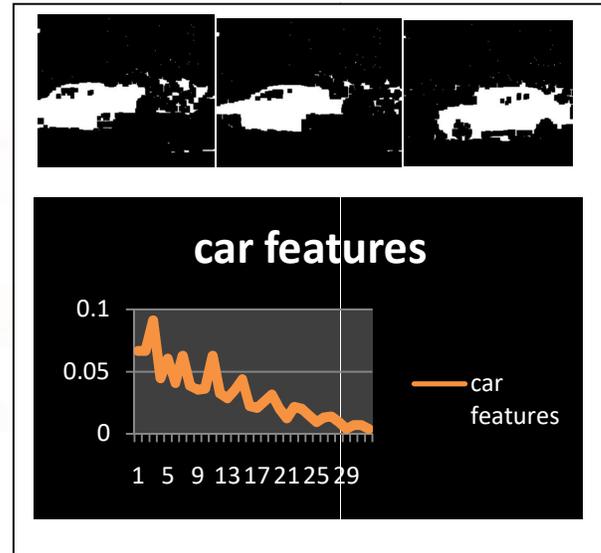


Figure 5b: Typical Radial 1- D distance feature vector from Humans (left) and vehicles (right)

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 centroid of the object, then the length of each line to the contour boundary of the object is given by:

$$l_i = \sum_{k=c}^w \delta(p(k, l)) \quad (25)$$

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 $k \tan(\theta)$, $\delta(\cdot)$ is a binary function that returns 0 or 1.

$$\delta(p(k, l)) = \begin{cases} 1 & \text{if } p(k, l) \in S \\ 0 & \text{otherwise} \end{cases} \quad (26)$$

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 $j = 1, 2, 3, \dots, n$ and n is given by $n = 360/\theta_{min}$ where θ_{min} 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. See [1] for more details of the segmentation algorithm.

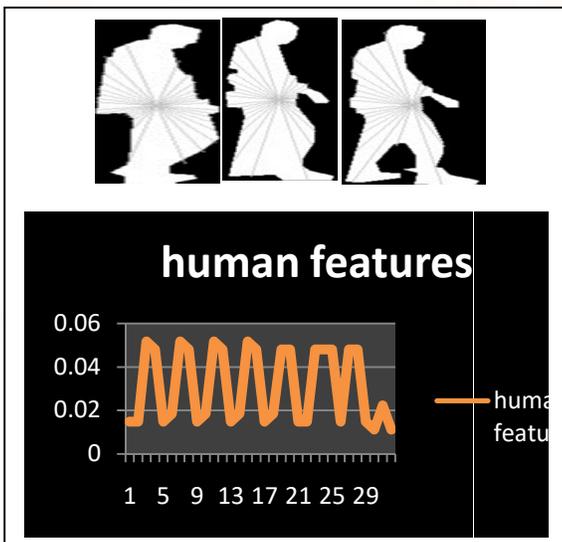


Figure 5a: Typical Radial 1- D distance feature vector from Humans (left) and vehicles (right)

4.2 GA-SVM Object Classification

In this work, the training of the SVM consists of providing the SVM with data for vehicles and pedestrians. Data for each class consist of a set of n dimensional vectors. A Radial Basis Function (RBF) kernel is applied to the SVM which then attempts to construct a hyper-plane in the n -dimensional space, attempting to maximize the margin between the two input classes. The SVM type used in this work is C-SVM using a non-linear classifier where C is the cost hyper-parameter [2]. The SVM is trained using 1D radial signals extracted from the silhouettes of the labeled images of humans and vehicles. Given a training set of human and vehicle images consisting of multiple labeled images of each human and vehicle classes an SVM classifier is trained as follow: 1D features data are extracted from the training set images to create the vector $\mathbf{X} = (x_1, x_2, \dots, x_L)$ for human and $\mathbf{Y} = (y_1, y_2, \dots, y_K)$ for vehicle images. where L and K are the total number of training images for class human and vehicle respectively. To train the SVM, $\mathbf{X} = (x_1, x_2, \dots, x_L)$ are used as the positive labeled training data and $\mathbf{Y} = (y_1, y_2, \dots, y_L)$ are used as the negative labeled training data. The SVM is then trained to maximize the hyper-plane margin between their respective classes (Φ_1, Φ_2). To classify an object F , it must be assigned to one of the p possible classes (Φ_1, Φ_2), in this case p takes two values. For each object class, Φ_p , an SVM is used to calculate this class that the object F belongs to given measurement vector D where in this case D is the set of 1D radial features.

5 Implementation of the Proposed Model

In order to evaluate the classification performance of the proposed SVM model, some video data were gotten from major roads in Akure, Nigeria. The background subtraction algorithm was applied to detect objects from the videos. Features of these objects are extracted and stored in a database. The data are divided randomly into the 80 training dataset 333 test dataset. The 80 training dataset consist of 40 vehicles and 40 pedestrians. The test dataset consists of 100 vehicles and 223 pedestrians. The class vehicle is assigned 1 while the class human is assigned 0. The GA-SVM model is trained using [3]. In order to

compare the classification performance of the proposed model with other classifiers such as the normal SVM, K-Means, K-NN, ANN, GA-ANN, the training is done with the same set of data and the same testing dataset is used to evaluate each classifier. In the proposed GA-SVM model, the SVM adopts Radial Basis Function as the kernel. The parameters C and γ of the SVM are optimized using Genetic Algorithm. Then these optimal parameters are used to train the SVM model. In the normal SVM model, the SVM use RBF as the kernel function. The parameters C and γ are randomly selected. In the K-Nearest Neighbour training, data for vehicles and pedestrians are provided for the K-NN. As many instances of vehicles and pedestrians are provided with their corresponding class labels. After the training is done, K-Nearest can then be used to classify test instances. To classify a new instance, the instance is supplied and the number of neighbors k . This k defines the neighborhood in which training data is consulted. The test data is compared with the training data. The nearest K -instances are checked and the majority nearest to k dictates the class that the instance belongs. In this research, k is set to 5. For the Classification Using K-Means Algorithm, Given a training set of human and vehicle classes, a K-Means clustering algorithm is trained using the vector $\mathbf{X} = (x_1, x_2, \dots, x_L)$ for human and $\mathbf{Y} = (y_1, y_2, \dots, y_L)$ for vehicle data. where L is the total number of training images. To train the K-means, the number of classes and the mean of each class is provided. K-Means now clusters each data item to any of the classes based on the Euclidean distance of the data to the mean of each cluster. The minimum distance from each class determines which cluster the data falls. Now after the entire instance are clustered, the means are re-evaluated again and the process continues until a given terminating condition is fulfilled. To cluster a new data, the distances of the new data to each of the evaluated cluster means are calculated and the one with minimum is the class of the new data. Object classification using ANN uses input nodes, which are 32 in number, the hidden layer which has 32 neurons and the output layer which has one neuron. The following steps are adopted in the Neural Network modeling for object classification. In this approach, shape features extracted from the detected moving objects are used to train a neural network classifier in order to recognize human and vehicle classes. The Learning Algorithm used is the back propagation algorithm. The dimension of input data is thirty two.

The network has one hidden layer having thirty two neurons and one output neuron. The network is trained using the back-propagation. This is a supervised learning algorithm in which the input vectors are supplied together with the desired output. The back propagation algorithm (BPN) learns during the training epoch. Several epochs are needed before the network can sufficiently learn and provide a satisfactory result. The number of epochs used is 500, with the momentum of 0.5 and learning rate of 0.3. The initial weights are randomly initialized to small random numbers less than 1 using random number generators. The object classification using using neuro-genetic model (GA-ANN) applies genetic algorithm to optimize the weights before the ANN is trained with the optimized weight. In this case, there are (32*32+32) number of weights which signifies a chromosome. The 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. This combination gives an excellent empirical performance.

6 Results and discussion

Table 1 shows the analysis result for GA-SVM, SVM, GA-ANN, K-MEANS, K-NN. GA-SVM. For the GA-SVM, 80 data are used to train the model.. The support vectors' values from the training are then saved. The gamma parameter obtained is 19.28 and the C value is 6.49. For testing, 333 data items consisting of 110 vehicle data and 223 pedestrian data are used. Out of the 110 vehicles, all are classified correctly while for 223 human class data items, only 2 are misclassified. Error in classification is 0.61% and the accuracy is 99.39%. For the normal SVM, error in classification is 0.9% and the accuracy is 99.1%. For GA-ANN, error in classification is 0.9% and the accuracy is 99.1%. For K-MEANS, error in classification is 1.2% and the accuracy is 98.8% and for ANN, error in classification is 1.5% and accuracy is 98.5% and for K-NN, error in classification is 4.8% and accuracy is 95.2%. Figure 6 shows the classification accuracy. From the figure, it is showed that the proposed model has the highest classification accuracy.

Table 1 Comparison of Classification Algorithms

S	PARAMETER	GA	S	GA	K-MEANS	ANN	K-NN
1	Number of observed features	413	41 3	413	413	41 3	41 3
2	Number of training set	80	80	80	80	80	80
3	Number of classes	2	2	2	2	2	2
4	Dimension of input features	32	32	32	32	32	32
5	Number of test set	333	33 3	333	333	33 3	33 3
6	Total Recognized	331	33 0	330	329	32 8	31 7
7	Classification accuracy	99.39%	99.1%	99.1%	98.8%	98.5%	95.2%

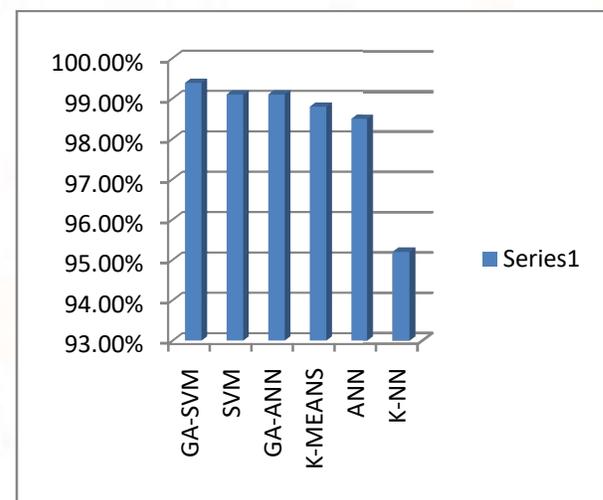


Figure 6: Classification Accuracy of the classifiers

7 Conclusion

In this paper, SVM with GA is applied to object classification in videos. In the proposed model, an optimized RBF kernel function and parameter C of SVM is obtained. Video data of moving objects have been used to evaluate the model. The experimental results show that the proposed classifier performs better than the SVM in which parameters are chosen randomly. The comparative analysis of the model with that of SVM, GA-ANN, K-MEANS, K-NN, ANN shows that the model shows more excellent performance in terms of classification accuracy. In future other optimization techniques will be looked into in order to investigate the performance of the SVM model on other optimization techniques.

References

- [1] Akintola K.G, Tavakkolie A. (2011). Robust Foreground Detection in Videos using Adaptive Colour Histogram Thresholding and Shadow Removal. *Proceedings of the 7th International Symposium on Visual Computing*, Vol. part ii JSVC'11, pp. 496-505, Berlin, Heideiberg, Germany, September, 2011.
- [2] Akintola K.G., Olabode O. (2014) A Neuro-Genetic Model For Moving Object Classification In Video Surveillance Systems. *International Journal of research in Computer Applications and Robotics* Vol. 2, No8 pp. 143-151, July, 2014.
- [3] 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>.
- [4] Chiu P., Girgensohn A., Polak W., Rieffel, E, Wilcox L. (2000). A Genetic Algorithm for video Segmentation and Summarization. *Proceedings of IEE International Conference on Multimedia and Expo*, Vol. 3, pp. 1329-1332, New York, 30th July-2nd August, 2000.
- [5] Collins, R. T., Lipton, A. J., Fujiyoshi, H., and Kanade, T. (2000). *A System for Video Surveillance and Monitoring: VSAM Final Technical Report, CMU-RI-TR-00-12*, Robotics Institute, Carnegie Mellon University, May 2000.
- [6] Elgammal A., Duraiswami R, Harwood D., and Davis L.S. (2002). Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance. *Proceedings of the IEEE* Vol.90, No.7, pp. 1151-1163, JULY 2002.
- [7] Heikkila J. and Silven O. (1999) A real-time system for monitoring of cyclists and pedestrians. *Prococeedings of Second IEEE Workshop on Visual Surveillance*, pp. 74–81, Fort Collins, Colorado, June 1999.
- [8] Hochreiter S., (2009). Theoretical Bioinformatics and Machine Learning Lecture Notes. *Institute of Bioinformatics*, Johannes Kepler University A-4040 Linz, Austria.
- [9] 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.
- [10] Montana D.J., and Davis L.D. (1989). Training Feed-Forward Networks using Genetic Algorithms. *Proceedings of the International Joint Conference on Artificial Intelligence*, IJCAI-89. Morgan Kaufman Publishers, Deutrot, MI, pp.762-767.
- [11] Michael P. and El-Kaliouby R. (2003) Real time facial expression recognition in video using Support Vector Machine in proceedings of *ICM Nov 5-7, 2003 Vancour, British Columbia, Canada*.
- [12] Ramya G and Srclatha M. (2014) Multiple Object Tracking using Support Vector Machine. *Global Journal of Researches in Engineering Subtraction* Vol. 14 Issue 6 Version 1.

- [13] Ravindra M Chandraslekhar G. (2014) Classification of Satellite Images based on SVM Classifier using Genetic Algorithm. International Journal of Innovative Research in Electrical Electronics Instrumentation and Control Engineering Vol. 2 Issue 5, May 2014
- [14] Rilley J., Ciesielski V.B. (1998). An Approach to Training Feed-Forward and Recurrent Neural Networks. *Proceedings of the Second International Conference on Knowledge-Based Intelligent Electronic Systems (KES'98)*, pp. 596-602, Adelaide, April 1998, USA.
- [15] VishwaKarma D.D. (2012). Genetic Algorithm Based Weights Optimization of Artificial Neural Network, *International Journal of Advanced Research in Electrical, Electronics and Instrumentation* Vol. 1. No 3, August 2012.
- [16] Vapnik V,N.(1995). The Nature of Statistical Learning Theory. *Springer Verlag*, New York, 1995. ISBN, 0-387-94559-8.
- [17] Wainwright R.L. (1993). Introduction to Genetic Algorithms Theory and Applications. The seventh Oklahoma Symposium on Artificial Intelligence. November, 19,1993. The University of Tilsa Cao Smith College Avenue, Tulsa Oklaoma.