Deanship of Graduate Studies

Al-Quds University

Video Categorization Using Data Mining

Noor Al-Sleibi

M. Sc. Thesis

Jerusalem-Palestine

1435/2014

Video Categorization Using Data Mining

Prepared By:

Noor Al-Sleibi

M.Sc.: Computer Science-Al-Quds University-Palestine

Supervisor: Dr. Rashid Jayousi

A thesis Submitted in Palestine Fulfillment of Requirements for the Degree of Master of computer Science / Department of Computer Science / Faculty of Graduate Studies –Al-Quds University.

1435/2014

Al-Quds University

Deanship of Graduate Studies

Computer Science Department

Thesis Approval

Video Categorization Using Data Mining

Prepare`d By: Noor Al-Sleibi

Registration No: 21011572

Supervisor: Dr. Rashid Jayousi

Master thesis submitted and accepted. Date:24/5/2014

The names and signatures of the examining committee members are as follow:

1-Head of Committee: Dr. Rashid Jayousi Signature:………………..

2-Internal Examiner: Dr. Jad Najjar Signature:………………..

3-External Examiner: Dr. Hashem Tamimi Signature:………………..

Jerusalem-Palestine

/2014

Dedication

To my parents, for their love, endless support and encouragement…

To my beloved husband, without his caring support it would not have been possible…

To my brothers, sisters, friends and colleagues…

To all of you I say a big "thanks".

Declaration

I certify that this thesis submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed………………………………………

Noor Al-Sleibi

Date: 24 / 5 /2014

Acknowledgment

First praise is to the Almighty Allah, Lord of all creatures, the Most Gracious, Most Merciful, for his graces and blessings throughout all my life. Without Him, everything is nothing.

My sincere thanks for my supervisor Dr. Rashid Jayousi, for his sincere efforts, interest and time he have kindly spent to guide my research.

I am very grateful to all professors at Al-Quds University-Computer Science department, for the time they have spent to teach me.

Finally, and most important, I would like to thank my husband. His support, encouragement, quiet patience and unwavering love were undeniable.

Abstract

Video categorization using data mining is the area of the research that aims to propose a developed method based on Artificial Neural Network (ANN), which could be used to classify video files into different categories according to the content. In order to test this method, the classifications of video files are discussed. The applied system proposes that the video could be categorized in two classes. The first one is educational while is non-educational. The classification is conducted based on the motion using optical flow. Several experiments were conducted using Artificial Neural Network (ANN) model. The research facilitate access to the required educational video to the learners students, especially novice students. This research objective is to investigate how the effect of motion feature can be useful in such classification. We believe that other effects such audio features, text features, and other factors can enhance accuracy, but this requires wider studies and need more time. The accuracy of results in video classification to educational and non-educational through technique 3 fold cross validation and using (ANN) model is 54%. This result may can be improved by introducing other factors mentioned above.

تصنيف الفيديو باستخدام التنقيب عن البيانات

إعداد: نور حسين محمود الصليبي

إشراف: د. رشيد الجيوسي

ملخص:

نظام تصنيف الفيديو باستخدام التنقيب عن البيانات من خلال تطبيق الشبكة العصبية الاصطناعية(ANN) تم اختباره ضمن هذا البحث، حيث يضمن البحث تصنيف الفيديو بناء على المحتوى إلى مجالات مختلفة. من أجل اختبار هذا الأسلوب، تمت مناقشة تصنيفات ملفات الفيديو في مختلف التخصصات. النظام المقترح تم تطبيقه لتصنيف ملفات الفيديو إلى فئتن، الفئة الأولى تصنيف ملفات الفيديو المختلفة إلى ملفات فيديو تعليمية. الفئة الثانية تصنيف ملفات الفيديو المختلفة إلى ملفات فيديو غير تعليمية، وقد تم تصنيف الفيديو باستخدام نموذج معايير الحركة باستخدام التدفق البصري. وقد أجريت عدة تجارب باستخدام نموذج الشبكات العصبية الاصطناعية (ANN) .بحيث أن هذا البحث يسهل الوصول إلى الفيديو التعليمي المطلوبة للطلاب المتعلمين، وخاصة الطلاب المبتدئين .الهدف من هذا البحث هو التحقيق في كيفية تأثير ميزة الحركة يمكن أن تكون مفيدة في هذا التصنيف . ونحن نعتقد أن تأثيرات أخرى مثل ميزات الصوت، ملامح النص، وعوامل أخرى يمكن أن تعزز الدقة، ولكن هذا يتطلب دراسات أوسع وتحتاج إلى مزيد من الوقت . دقة النتائج في تصنيف الفيديو إلى تعليمية وغير تعليمية من خلال تقنية التحقق من الصحة

(3fold cross Validation) وباستخدام (ANN) كانت 54٪.

Table of Contents

List of Figure

List of Table

List of Appendices

Chapter one

Introduction:

Multimedia resources have become increasingly an important resource of information by different types of users where it included verities of information such as short training videos, interactive learning objects, infotainment resources as well as asynchronous and synchronous communication channels [1]. In the last few years, lecture recordings have become an important educational media since more appropriate technology has become available for recording, processing and distributing of multimodal or multimedia sources [2]. Lecture recording refers to the process of digitally capturing and archiving the content of lectures, conferences, or seminars. It consists of hardware and software components which could work in synergy in order to record audio and visual components of the lecture [3]. These lecture recordings or educational videos show the use of a great variety of media or modalities such as video, audio, lecture media, presentation behavior and formats. Consequently, an increasing number of lecture recordings are available in order to complement face to face and more conventional content-based e-learning approaches but also to provide additional channels for remote students and time-independent access to the lectures [4]. Therefore, many of the lecturers create teaching material for different target groups in joint efforts. This material maintained as a common repository, in order to support the lecturers to create of course material, accordingly, this will propose and lead to support retrieval of reusable components by a powerful retrieval service to video data [5].

The general objective of this work is to segment a given video sequence into its constituent shots, and to identify and classify the different shot transitions in the sequence [6].Video segmentation is based on motion and audio [7]. Motion is a salient feature in video [8]. While audio is used in speech recognition and language understanding technology to produce keywords for each video frame or a group of frames [9]. Video content is approached at different levels: raw data, low-level visual content and semantic content. The raw video data consists of elementary video units together with some general video attributes such as format, frame rate, duration etc. Visual features such as color, shapes, textures etc. characterize low-level visual content. Semantic content contains high-level concepts such as objects and events [10]. In order to extract the semantic content, which is our objective, the motion intensity and the motion of the moving objects, has to be estimated first [8]. Major Issues in multimedia data mining include content-based retrieval and similarity search, generalization and multidimensional analysis, classification and prediction analysis, and mining associations in multimedia data [11].

Neural networks for speech recognition have been used by researcher as an effective recognition tools [12, 13, 14 and 15], where the MATLAB Neural Network toolbox usually use to create, train and tested the networks [16]. Several experiments were conducted in order to build neural network models which include several set of different video types, and different domains. MATLAB tool were chosen, due to the following attractive features: simple programmability, impressive integrated graphical functions, highly optimized for vectorized code, suitable for Digital signal processing (DSP) which deals with the analysis and manipulation of digital signals such as audio and images, availability of MATLAB for a wide variety of platforms. In addition to that, extensibility to a web-based education system such as that sed at UT-Austin [17].

This research aims to develop a specific method for the classification video files to different classes based on analysis content using neural network. The first class is educational. Second class is non-educational. This research is objective to investigate how the effect of motion feature can be useful in such classification. We believe that other effects such audio features, text features, and other factors can enhance accuracy, but this requires wider studies and need more time. The results were effectively, the accuracy of results in video classification to educational and non-educational through technique 3 fold cross validation and using (ANN) model is 54%.

Motivation

Due to the difficulty faced by the students' education, especially novice students to obtain the desired educational video, we search to devise a particular method for the classification and prediction of video files to different fields so that students find the desired video with less time and effort. This were tested or applied several experiments to build Artificial Neural Network model (ANN) therefore this work was present accurate results and satisfactory performance using this model. And to prove that results of this research are accurate than YouTube because this research depends on the analysis content while YouTube depends on the comparison between the keywords.

Objectives

The main objectives of this research are:

1-Classifiing video files to different classes based on content analysis.

2-Facilitate access to the required educational video to the learners students, especially novice students.

3-To obtain the desired educational video with less time and less effort.

This thesis is organized as follows: The literature and related work will be discussed in chapter two. A background of the research including a definition about the problem the technique neural network, classification and classifiers, transfer function, motion estimation, extract video features and the cross validation used in the research was presented in chapter three. The research methodology followed to perform the experiment was presented in chapter four. Experimental results are presented and discussed in chapter five. Finally, the conclusion and future work was given in the last chapter six.

Chapter two

Literature Review

Many studies have been done across countries on data mining applications of data mining were used in a large number of areas, particularly for business and medical purposes.

Data mining is a new technology field and it is important and very helpful in predicting and detecting underlying patterns from large volumes of data. Many researches were published on classification video files depending in content to different areas. The recent study in [18] reviews video of representation and classification using intermediate semantic representations derived from multimodal features of video. The performance was measured through three undirected graphical models for semantic representation and classification of video data by jointly modeling of textual and image features of the data, and perform classification based on such latent representations. The effect of a series of probabilistic models in classification, such as dual-wing harmonium (DWH) model that represents video shots as latent semantic topics derived through jointly modeling the transcript keywords and color-histogram features of the data. The experiments in [18] have demonstrated that these models achieve satisfactory performance on video classification and provide insights to the internal structure and relationships of video categories as well as evaluation to the performance of these models in video classification.

For the classification of video files effectively and in accordance with the requirements of the user, we need to analysis tools for video analysis. In this field the researchers in [19] discussed user requirements for video analysis, from the derivation of a set of functional specifications. These specifications are useful for evaluating existing systems and for guiding the development of new systems, as well as building video analysis tools. So as to provide the reader with an understanding of design issues for video analysis systems, using the VANNA system (Video Annotation and Analysis system), which reflects these guidelines and illustrates a number of unique interface design approaches. The experiments in [19] demonstrate that interface for entering annotations has been designed to reduce both perceptual and cognitive load. The results in [19] showed that the VANNA provides users with several default templates or screen layouts and a dictionary of functions. Users may add, delete, resize or relocate any object on a template, including the video window, by directly manipulating the objects themselves (e.g., cut, copy, paste, drag). Only the functions deemed necessary by the user are presented.

In the sense, a researchers in [20] were described how to use video preferences to extract visual features and closed captions from video to learn a viewer's preferences, the approaches in [20] is to identifying videos of interest to a viewer , the first approach ,is based on the use of specifications of video content including the genre, director, actors, and plot summary. That is relies on viewers profile which match viewer's preferences with video content descriptions. A second approach is collaborative filtering, which users attempt to identify viewers that are considered similar by some measure. In addition to that the researchers in [20] applying hidden markov models (HMM), by combining visual features and closed captions to produce observation symbols for training. The results in [20], was constructed two HMMs: one from the training samples of the movies the viewer rated as liked and others rated as disliked. An analysis of the models shows that, for many users, one of the models would perform well while the other model would perform poorly.

In another study in [21], the researchers presented a set of algorithms for the automatic generation of video abstracts, through decomposition the input video into semantic units, detect and extract semantically rich pieces, in particular text from the title sequence and special events such as dialogs, gun fires, and explosions. Video clips, audio pieces, images and text are extracted and composed to an abstract. Through using MoCA project (Movie Content Analysis) which aim is to automatically determine the contents of digital video, since there is no mathematical measure for the quality of a video abstract .In which the experiments in [21] were presented the abstracts to a set of test persons, when the authors compared the abstracts generated by MoCA project, with abstracts actually shown on television, no obvious difference in quality. The results in [21] showed for last week's episode of a TV series, the scenes generated by project tool were very similar to the ones shown on television.

The researchers in [22] focused on the study of motion activity descriptor for shot boundary detection in video sequences, the motion activity is one of the motion features included in the visual part of the MPEG-7 standard. The motion activity was applied for different video sequences .The idea of video segmentation using motion intensity is to use the extraction of key frames method based on detecting a significant change in the activity of motion, for example extract the motion vectors between images i and image i+2, then calculates the intensity of motion, then repeat this process until reaching the last frame of the video, and comparing the difference between the intensities of successive motion to a specified threshold. The motion activity information is extracted in uncompressed domain based on adaptive rood pattern search (ARPS) algorithm. In addition to that, the experiment in [22] has been for every video sequence determines the number of shots, the number of false detections and the number of non-reported shots. For each sequence, the researchers in [22] draw the curve of the distances between the successive frames. These curves are mainly used to determine the threshold values, but they also give an idea about the capacity of the used technique in detecting transitions. The results demonstrate that the use of motion activity can assure a satisfactory of shot boundary detection rate, and can be useful for real time implementation.

To distinguish and classifying between several things, such as pictures or video clips or television programs or several voices ...etc, need to use certain techniques customized to suit every type, so that each technique differs from the other in the use and in the type. Like accordingly, this researchers in [23] were discussed how to discrimination between five types of TV programs, namely commercials, basketball games, football games, news reports, and weather forecasts. Based on audio features, through applying using Hidden Markov Model, to classifying TV broadcast videos. The experiment data in [23] are divided into two sets: training and testing data sets. The training data set includes 10 minutes audio for each scene class and the remaining 10 minutes audio in each class forms the testing data set. The results in [23] were showed better when attached these visual features with audio features to create new codebook and train the HMM. The experiments proved that the results are better when using the visual features with audio features; the visual information can further improve the classification accuracy.

In addition to that, the researchers in [24] focused on classifying identities of people appearing in broadcast news video into anchor, reporter, or news subject in high-level video analysis, based on multi-modal analysis. This work gives a typical example on how to analyze different video modalities including the speech identity, transcript clues, and temporal video structure, named entities. Through apply a statistical learning approach to combine all the features for person type classification. The experiments conducted in [24] on ABC World News Tonight video, and the different categories of features have been compared. The results were, as follows : There are totally 498 people (or monologue shots) in the test data, among which 247 are news subjects, 186 are anchors, and the rest are reporters, and the classifier was very effective in classifying person types.

Recent studies in [25] reviews, designed several analysis algorithms for video object detection and segmentation in the general framework of multimedia content .This work in [25] is mainly focused on model-based algorithm design, the algorithms are developed in two directions, automatic algorithms and semi-automatic algorithms. The analysis designed video object detection and segmentation algorithms for three applications: (1) Real-time VO segmentation for videophones. (2) Anchorperson detection and segmentation for broadcast news indexing and retrieval. (3) Face detection. The author provided designed several analysis algorithms for video object (VO) detection and segmentation. Which the results in [25] indicate that this system offers a good balance between algorithm complexity and user interaction efficiency.

the researchers in [26] presented how to investigation of the speech recognition classification performance, using two standard neural networks structures as the classifier, feed forward network with back propagation algorithm and the Radial Basis Functions neural network. This work in [26] was showing that neural networks can be very powerful speech signal classifiers. In addition to that the authors in [26] observed that Mel Frequency Ceptstrum Coefficients are a very reliable tool for the pre-processing stage with the good results. Preprocessing part, which is composed of different sub stages, filtering, entropy based endpoint detection and Mel Frequency Cestrum Coefficients. In which the results showed for every word we used 200-recorded samples. From these 200 samples, 100 samples were used for training, while the other 100 were be used to test the network.

In another study the authors in [27] were proposed a simple statistical distance measure between video sequences based on their behavioral content. This measure is non-parametric, and can thus handle a wide range of dynamic events without prior knowledge of the types of events, their models, or their temporal extent. The performance was measure for a variety of video applications, including event-based detection, indexing, temporal segmentation, and clustering of long streams of video sequences .The results in [27] were showed sequence contains four types of frequently occurring activities: walking, jogging, hand-waving, and walking-in-place and the author evaluates the result of clustering is satisfactory.

In another research, the authors in [28] were focused in video clip classification using audio processing to create many different video indices for use video on demand (VOD), educational video indexing, and sport video characterization. In addition to that, examine the audio information for content characterization by demonstrating extraction feature of audio processing from compressed data stream through MPEG encoding of the audio and video stream using in which separate stream using Xing CD MPEG encoder. In which the results showed in [28], 7592 frames with 81 clips demonstrated that average energy is sufficient measure for detecting silent segment, and examine of silent and non-silent characteristic of the segment, and separating silent from voiced or unvoiced data.

In addition to that, the authors in [29] were discussing the use multilayer neural network initialization with decision tree classifier for improving text categorization accuracy, presented a hybrid decision tree and neural network algorithm for building the classifier. The results in [29] were support the conclusion that error based pruning can be used to produce appropriately sized trees, which are directly mapped to optimal neural network architecture with good accuracy.

In this research we a developed method based on Artificial Neural Network (ANN), which could be used to classify videos according to the content into different categories. The video could be categorized in two classes. The first class classification video to educational. Second class classification video to non-educational. The classification is conducted based on the motion through using optical flow. Several experiments are conducted using Artificial Neural Network (ANN) model. The accuracy of results in video classification to educational and non-educational through technique 3 fold cross validation and using (ANN) model is 54%.

Table2.1: Summary of literature review according to area of research

Papers	Problem	Tool	Results
[18]	Discussed video representation and classification using intermediate semantic representations derived from multimodal features of video.	Using dual-wing harmonium (DWH) model that represents video shots as latent semantic topics.	Experiments have demonstrated that these models achieve satisfactory performance on video classification and provide insights to the internal structure and relationships of video categories. In addition, evaluate the performance of these models in video classification.
[19]	Discussed user requirements for video analysis, from to derive a set of functional specifications. These specifications are useful for evaluating existing systems and for guiding the development of new systems, for building video analysis tools	Using the VANNA system (Video Annotation and Analysis system), which reflects these guidelines and illustrates a number of unique interface design approaches.	Experiment on interface for entering annotations has been designed to reduce both perceptual and cognitive load VANNA provides users with several default templates or screen layouts and a dictionary of functions. Users may add, delete, resize or relocate any object on a template, including the video window, by directly manipulating the objects themselves (e.g., cut, copy, paste, drag).
[20]	Determine how to combine the text and visual features and how to capture the temporal relationship of features.	Using hidden Markov models (HMM).	Experiment by combine visual features and closed captions to produce observation symbols for training, constructed two HMMs: one from the training samples of the movies the viewer rated as liked and one from the movies the viewer rated as disliked. The results from combining features were approximately the same as those achieved when generating observation symbols for either type of feature alone. An analysis of the models shows that, for many users, one of the models would perform well while the other model would perform poorly.

[21]	The authors focused on a set of algorithms for the automatic generation of video abstracts, by decompose the input video into semantic units , detect and extract semantically rich pieces, in particular text from the title sequence and special events such as dialogs, gun fires, and explosions. Video clips, audio pieces, images and text are extracted and composed to an abstract.	Using MOCA project(Movie Content Analysis)	Experiments were presented the abstracts to a set of test persons, when the author compared the abstracts generated by MoCA project(Movie Content Analysis) , with abstracts actually shown on television, no obvious difference in quality, the reviews for last week's episode of a TV series the scenes generated by project tool were very similar to the ones shown on television.
[22]	The authors focused on the study of the motion activity descriptor for shot boundary detection in video sequences, the motion activity is one of the motion features included in the visual part of the MPEG-7 standard.	motion activity information is extracted in uncompressed domain based on adaptive rood pattern search (ARPS) algorithm	Experiment For every video sequence were the author determine the number of shots, the number of shots correctly reported, the number of false detections and the number of non reported shots. For each sequence the author, draw the curve of the distances between the successive frames. The results demonstrate that the use of motion activity can assure a satisfactory of shot boundary detection rate, and can be useful for real time implementation.
[23]	Discussed how Discrimination between five types of TV programs, namely commercials, basketball games, football games, news reports, and weather forecasts. Based on audio features	Using Hidden Markov Model to classifying TV broadcast videos	The experiment data are divided into two sets: training and testing data sets. The training data set includes 10 minutes audio for each scene class and the remaining 10 minutes audio in each class forms the testing data set the results were better when attached these visual features with audio features to create new codebook and train the HMM. The experiments proved that the results are better when using the visual features with audio features; the visual information can further improve the classification accuracy.
[24]	The authors focused on Classifying identities of people appearing in broadcast news video into anchor, reporter, or news subject in high-level video analysis,	Using multi-modal analysis	Experiments conducted on ABC World News Tonight video, and the different categories of features have been compared. The results were as follows: There are totally 498 people (or monologue shots) in the

	based on multi-modal analysis.		test data, among which 247 are news subjects, 186 are anchors, and the rest are reporters.
[25]	Authors focused on designed several analysis algorithms for video object detection and segmentation in the general framework of multimedia content. The analysis designed video object detection and segmentation algorithms for three applications: (1) Real-time VO segmentation for videophones. (2) Anchorperson detection and segmentation for broadcast news indexing and retrieval. (3) Face detection.	The algorithms are developed in two directions, automatic algorithms and semi-automatic algorithms.	The Experiments indicate that this system offers a good balance between algorithm complexity and user interaction efficiency.
[26]	Researchers presented how to investigation of the speech recognition classification performance.	feed forward Network with back propagation algorithm and the Radial Basis Functions Neural Network	The results showed for every word we used 200-recorded samples. From these 200 samples, 100 samples were used for training, while the other 100 were be used to test the network.
[27]	The authors proposed a simple statistical distance measure between video sequences based on their behavioral content. This measure is non-parametric, and can thus handle a wide range of dynamic events without prior knowledge of the types of events.	Statistical distance models.	The results were showed sequence contains four types of frequently occurring activities: walking, jogging, hand-waving, and walking-in-place and the author evaluates the result of clustering is satisfactory.
[28]	The authors focused in video clip classification using audio processing to create many different video indices for use video on demand (VOD), educational video indexing, and sport video characterization.	using XingCD MPEG encoder	The results showed, 7592 frames with 81 clips demonstrated that average energy is sufficient measure for detecting silent segment, and examine of silent and non-silent characteristic of the segment, and separating silent from voiced or unvoiced data.
[29]	the authors were discuss the use multilayer neural network initialization with decision tree classifier for improving text categorization accuracy,	Using network And decision tree.	The results were support the conclusion that error based pruning can be used to produce appropriately sized trees, which are directly mapped to optimal neural

	presented a hybrid decision tree and neural network algorithm for building the classifier		network architecture with good accuracy
My research	Classification and prediction videos type in different categories video files in two classes, educational video and, non-educational.	using artificial neural network (ANN)	Conducted several experiments to build a (ANN) model includes several set of different video types. So that the results were satisfactory and accurate, the results of Neural Network through using technique 3fold cross validation are 54%.

. As far as we know and shown in the table above ANN were used to conduct classification for text and voice but not video. In this research we were interested on how to classify and predict the type of videos type in different categories, educational video and non-educational video using data mining Artificial Neural Network (ANN) techniques. Experiments are conducted according to different of video type through using ANN

Chapter three

BACKGROUND

3.1Problem Definition:

Video classification are important topic in the research area, facilitate search and retrieval so that automatic classification based on the contents of the video in different domains, through video segmentation into several frames. Video segmentation and indexing as one mechanism for organizing video information. Video segmentation is problem pertaining to partition the video in meaningful generic modules for efficient storage. Physiological studies have shown that the human brain organizes and recalls information throw its content [28].

The key issue is how to build an effective method to classify videos according to content information. We propose to achieve this approach using ANN, and video classification into different classes based on motion criteria, to educational video and non-educational video using ANN. Despite a number of researches conducted in the classification of video lately, but we found it needed great importance in the classify video based on the information content. The performance results have been quite encouraging in video classification based on content information using ANN. Performance of neural networks learning is known to be sensitive to the initial weights, architecture number of hidden layers, transfer function and neurons in these layers. Traditionally, the initial values of weights are determined randomly in the back propagation neural network. Recently, pattern recognition techniques have been used to initialize weights [30]. As a result, the neural networks constructed from these pattern recognition which implemented using a feed-forward, the network is trained to associate outputs with input patterns.

This research presents our attempt to improve video classification accuracy based on content information using data mining neural network.

3.2Classification and Classifiers

The classification is the most important topics which are used nowadays, classification is an important problem in the field of data mining or machine learning technique used to predict. Classification has been studied extensively in the past [31]. The concept of classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. The model is used to constructing a set of classifiers, which are then applied at test time and then pieced together in training time [32]. The model may be represented in various forms, such as classification, decision trees, or neural networks [33]. The basic classification techniques in data mining method including decision tree induction, neural networks [34].

Classifiers are functions which partition a set into two or more classes, for example classification weather into the set of rainy days and the set of sunny days, the rainy days and sunny days are called classifiers. Classifiers are simple enough to make their analysis easy to understand [35].

3.3Artificial Neural Network (ANN)

Artificial neural networks refer to computing systems whose central theme is borrowed from the analogy of biological neurons. As inspiration of this biological system a mathematical or computational model was designed that simulate the structure and functional aspects of biological neural networks and called Artificial Neural Network

(ANN), but when we use this term in information technology, we refer to it just by neural network.

The neural network simulation idea is that each neuron (node) has a certain number of inputs each holding incoming signal (instance) with a level of importance associated with each input called weight. The input value of a node is the sum of the weighted input values from its incoming inputs, if the sum passes a predefined threshold, and an activation function generates the node output value using the node input value and passes the node output to other nodes in the network. The set of input nodes are called the input layer while the set of output nodes are called the output layer, and in between there are another layer (one or two) called hidden layer. This is called architecture neural network that is divided to feed forward networks and feed backward networks Fig (3.3.1).

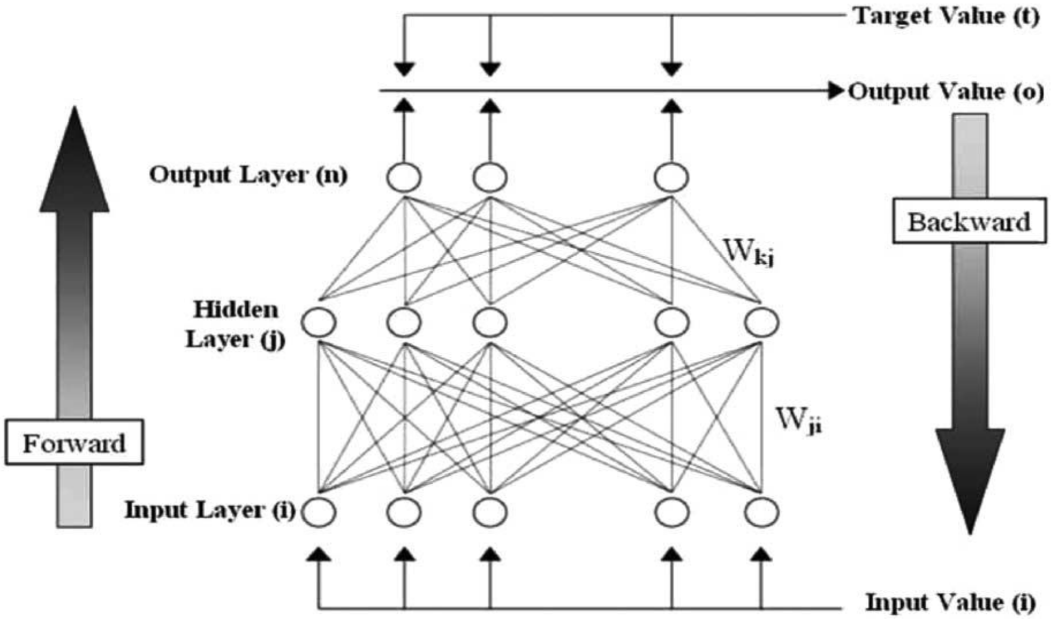


Figure 3.3.1: The structure of a neural network [30].

Each hidden unit is a function, of the weighted sum of the inputs and determines the values of the weights. [37, 38, 39, 40 and 41].

Multilayer neural networks, trained with a back-propagation learning algorithm, are the most commonly used in neural networks [42]. A back propagation network learns by example. You give the algorithm examples of what you want the network to do and it changes the network's weights so that, when training is finished, it will give you the required output (target) for a particular input [43].

The multilayer neural network operates in two modes: training and prediction mode (testing mode). For the training of the multilayer neural networks and for the testing we need two data sets, the training set and testing set. The training mode begins with arbitrary values of the weights, they might be random numbers, and proceeds iteratively. Each iteration of the complete training set is called an epoch.

In each epoch the network adjusts the weights in the direction that reduces the error. The iterative process of incremental adjustment continues the weights gradually converge to the locally optimal set of values. Many epochs are usually required before training is completed.

A multilayer back propagation network can be used in pattern recognition problem, with enough neurons can implement just about any function, back propagation does not always find the correct weights for the optimal solution. We might want to reinitialize the network and retrain several times to confirm that you have the best solution [44].

Feed-forward networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear relationships between input and output vectors.

The popularity of feed-forward networks derives from the fact that they have been applied successfully to a wide range of information processing tasks in such diverse fields as speech recognition, financial prediction, image compression, medical diagnosis and protein structure prediction; new applications are being discovered all the time .

Feed-forward networks are trained, rather than programmed, to carry out the chosen information processing tasks. Training a feed-forward net involves adjusting the network so that it is able to produce a specific output for each of a given set of input patterns. Feed-forward networks have a characteristic layered architecture, with each layer comprising one or more simple processing units called artificial neurons or nodes. Each node is connected to one or more other nodes by real-valued weights (parameters), but not to nodes in the same layer. All feed-forward nets have an input layer and an output layer [45].

Why artificial neural network was selected?

The basic reason lies in the fact that neural networks are able to capture complex relationships and learn from examples and able to adapt when new data become available [46]. Neural networks appear well suited to pattern recognition roles where the matching required is inexact. These flexible matching properties are expected to improve retrieval, particularly for inexperienced end users, neural networks learning allow matching and recognition software to be crafted using the structure of the data itself [47].

Neural network is divided in two phases training and testing .In the training phase after the building of neural network, the input data is fed into the network through the input nodes, along with the desired output data. The neural networks self-adapt to the data and incite appropriate responses. This process of making the network adapt to the data is known as training of a neural network and the algorithms used for this purpose are known as training algorithms. These algorithms can be classified according to their modeling, learning, and validation properties. In the testing phase, the neural network is able to predict the output correctly for new data and hence the network is validated [46].

In this research, as in the methodology in chapter four, we used neural network feed forward to establish a prediction model .The neural network (ANN) is a powerful tool for pattern recognition.

3.4Transfer function

Transfer functions can strongly affect complexity and performance of neural networks. In the process of neural networks training flexible transfer functions are important to well architectures that are well-matched for a given data and learning procedures [48]. Optimal transfer function networks select appropriate functions for a given problem creating architectures (number of neurons, connections and type of neurons) [49].

Many transfer functions are used in the neural network toolbox but there are three of the most commonly used functions in neural network. First transfer is linear transfer function (PURELIN); neurons of this type are used in the final layer of multilayer.

Second transfer is the Log-Sigmoid transfer function (LOGSIG) takes the input, which can have any value between plus and minus infinity, and squashes the output into the range 0 to 1 can be seen in figure 3.4.1. This transfer function is commonly used in the hidden layers of multilayer networks. The third transfer is Tan-Sigmoid transfer function (TANSIG) is related to a bipolar sigmoid which has an output in the range of -1 to +1. Sigmoid output neurons are often used for pattern recognition problems, while linear output neurons are used for function fitting problems [44]. These transfer function can be seen in figure 3.4.1 [44].

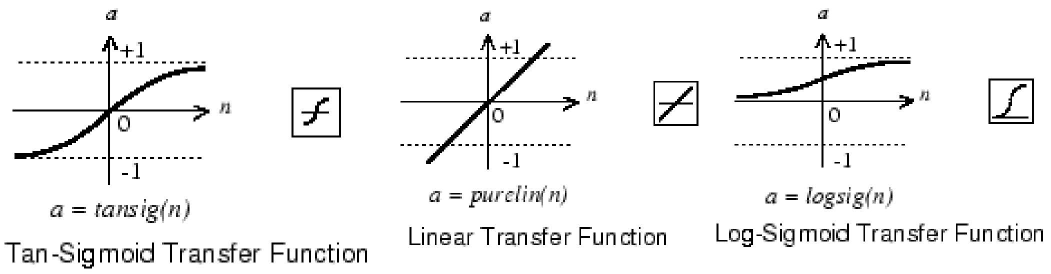


Figure3.4.1: transfer functions TANSIG, PURELIN and LOGSIG [44].

3.5Motion estimation

An image sequence or video is a series of 2-D images that are sequentially ordered in time, video or motion picture cameras can acquire image sequences. Video processing using motion estimation is performed using optical flow. The motion estimation, we mean the estimation of the displacement (or velocity) of image structures from one frame to another in a time sequence of 2-D images [50].

Optical flow estimation is making steady progress as evidenced by the increasing accuracy [51]. Optical flow combines a data term that assumes constancy of some image property with a spatial term that models how the flow is expected to vary across the image [52].

Optical flow approaches calculate vector fields, which determine the apparent velocities of objects in time-varying image sequences. The velocity can be estimation between two images or between current frame and Nth frame back. After we obtain the velocity from the optical flow block, we need to calculate the velocity threshold in order to determine what is the minimum velocity magnitude corresponding to a moving object.

The optical flow (Horn-Schunck) in Mat-lab Simulink deals with the motion as vector, and it concentrates on the accuracy and density motion of objects or events in each frame of video. Where the optical flow 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 [53].

3.6Extract video features

The concept of the "Content-based" is the search will analyze the actual content of the video. The term 'Content' in this context might refer to (1) color is a visual feature such as images and videos. (2) Shapes are a set of measurements that used to define the shapes. (3) Texture as the visual patterns. (4) Audio features used for indexing video and retrieval of Video.

Selected good features that reduced from costs of the time and space in the retrieval process. The interesting features that can be extracted from video data were a basic role in the indexing and retrieval from collection videos. The video-content analysis based on video browsing and retrieval is the partitioning of a video sequence into shots. A shot is defined as an image sequence that presents continuous action which is captured from a single operation. Shots are joined together in the editing stage of video production to form the complete sequence [54], can be seen in fig3.6.1.

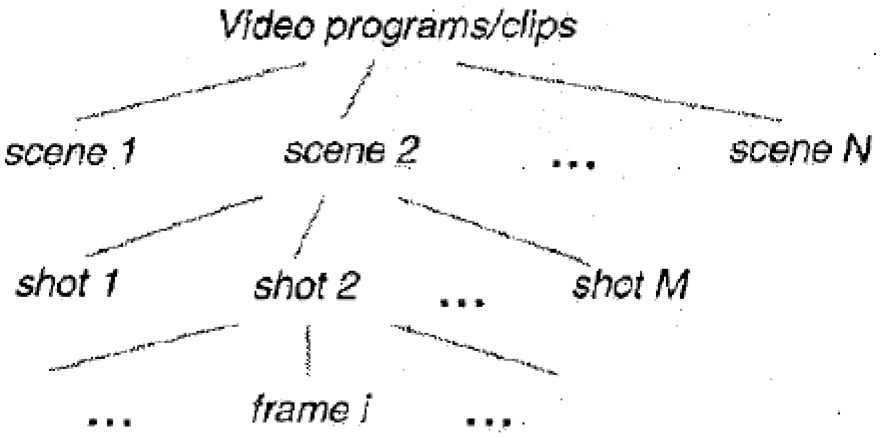


Figure3.6.1: Video segmentation [54].

Content-based search and extracted features of video data has become an important topic in content analysis. Types of video content such as motion, audio, visual information and texture features which are difficult to extract [55]. To extracting motion features calculate motion features at each point with local temporal units divided in order to consider consecutiveness of motions using optical flow [56].

3.7Cross validation

The concept of cross-validation is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one used to train a model and the other used to validate the model. The basic form of cross-validation is k-fold cross-validation. Other forms of cross-validation are special types of k-fold cross-validation or include repeated rounds of k-fold cross-validation such as leave-one-out cross validation.

The cross validation methods are hold-out validation, leave-one-out cross validation, repeated K-fold cross validation and K-fold cross validation.

The mechanism work k-fold cross-validation the data is first partitioned into k equally sized folds**,** then k iterations of training that is dark section and validation that is light section are performed such that within each iteration a different fold of the data is held-out for validation while the remaining K-1 folds are used for learning.In k-fold cross-validation the data is first partitioned into k equally sized segments. then k iterations of training and validation are performed such that within each iteration a different fold of the data is held-out for validation while the remaining K-1 folds are used for learning an example with k=3 in fig (3.7.1).

the objectives of the cross-validation is estimate performance of the learned model from available data using one algorithm and compare the performance of two or more different algorithms and find out the best algorithm for the available data this, or compare the performance of two or more variants of a parameterized model, this objective achieved automatically to estimate accurate performance indicators such as accuracy and precision.

In data mining and machine learning such as the classification 3-fold cross-validation (k = 3) is the most common, and using 3-fold cross-validation one repeatedly uses 90% of the data to build a model and test its accuracy on the remaining 10% [57].

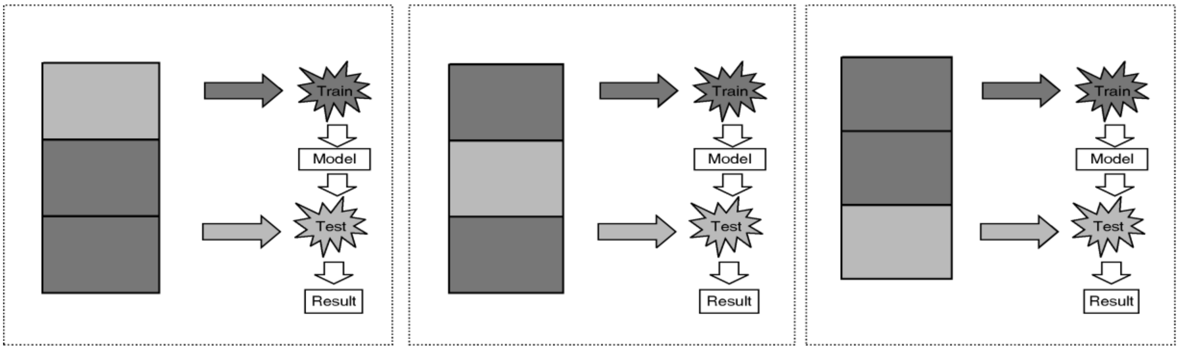


Figure3.7.1 procedure of three fold-cross validation K=3 [57].

Chapter four

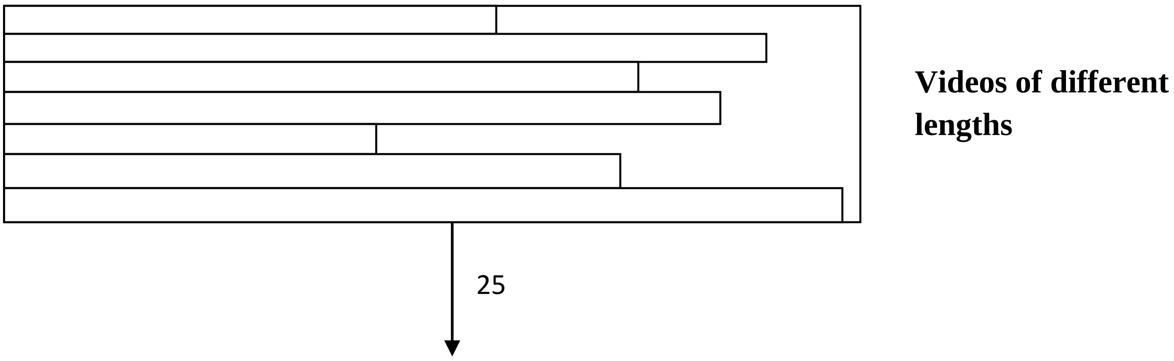
Methodology

For the achievement of the objectives of this research we have started collecting a number of various video files in terms of content and coordination, in order to develop a certain method in the classification of video based on content we used the data mining neural network, can be seen in figure 4.1.1. To test this method, the work has been conducted as follows:

4.1data collection

The data was collected through YouTube, the number of files were downloaded about 38 videos, and type of these videos are a video clip, the size of these videos start from 16.1 MB to the 587MB, length of these video files from 00:01:02 to the 00:01:56, frame width ranges between 280 to 1280, frame height ranges between 268 to 720, frame rate 30 frame/second to all videos.

The various video files have been classified manually into two main classes, educational and non-educational in desired output where the 0 is educational and 1 is non-educational in targets and the input data (videos) are different lengths. We started to read the video files and to get general information about each video file such as duration, number of frames, width and height, etc.



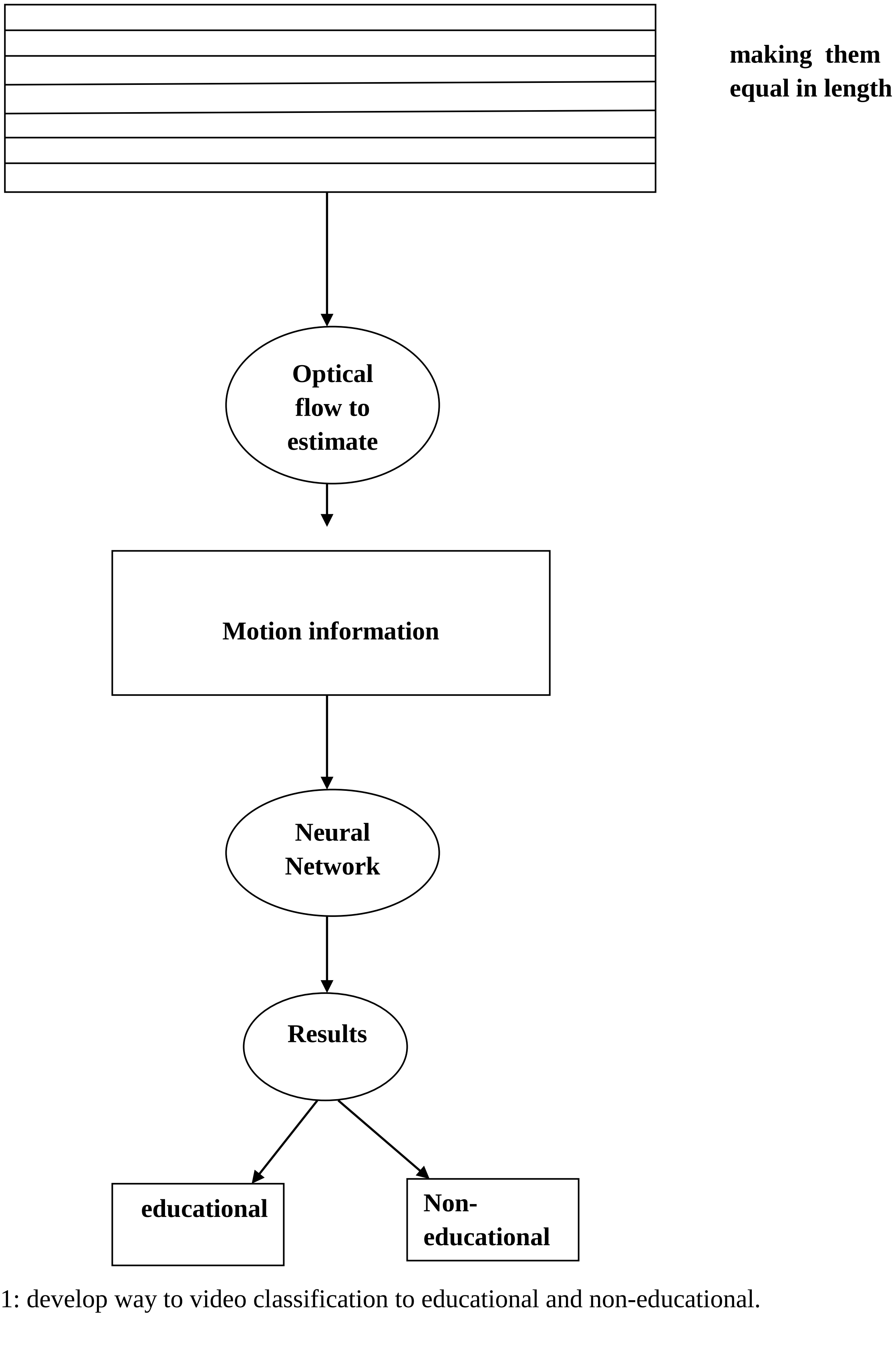


Figure 4.1.1: develop way to video classification to educational and non-educational.

4.2method

We have videos of different lengths and for knowing the appropriate length for all Videos, assuming that human takes half a minute is sufficient to know that each video is a educational or non-educational, the results of estimation motion of optical flow are image optical flow data and real optical flow data in the one minute, the video rate 30 frame/second .In the one minute is 1800 frame in the image optical flow and 1800 frame in the real optical flow. Through using image optical flow data in the ANN model was divided Each video into several samples so that the size of each sample 1800 frame, for example, if we have a video size 2070 frame, this video divided into one sample only 1800 frame, and the remainder negligence. If we have another video in size 5800 frame, and this video are divided into three samples, each sample is 1800 frame, can be seen in figure 4.2.1.

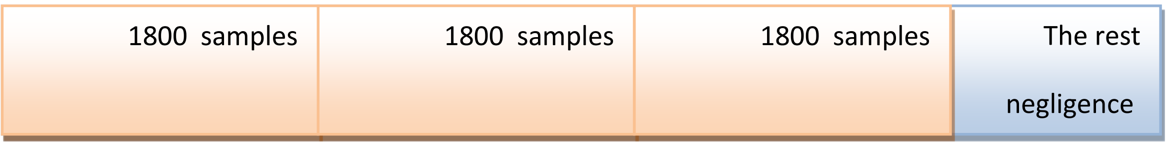


Figure 4.2.1: video divided into samples equal in length and each sample is 1800.

This work was done through calculate the motion of optical flow (see in section 3.5) in each video file from the first frame to the last frame. The number of samples 411 sample of educational and non-educational and the inputs of motion information were inserted into neural network is 450 and the output is 1 .Then the neural network model was built, and through using pattern recognition tool, to classify inputs neurons into a set of target categories, one layer feed -forward network through using back propagation with transfer function TANSIG and PURELIN, the numbers of hidden layer are different hidden layer 10, 20 and 30 is the highest value due to lack of enough space in computer memory and the output neurons is 1 output as shown in figure 4.2.2:

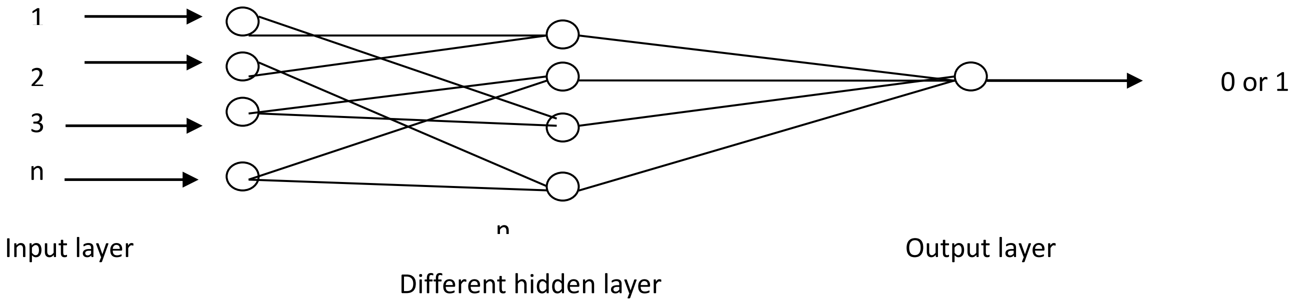


Figure 4.2.2: the neural network

4.3 3fold cross validation

The experiments were conducted using 3fold cross validation (see in section 3.7) in neural network using transfer functions, the number of data are 411 sample of educational and non-educational, these samples are divided into three parts, the first part in the training phase AB and C in the testing, second part AC in the training and B in the testing and third part BC in the training and A in the testing. After each experiment were compared between the results in the training phase with the results of the required output(targets) and the comparison was also in the testing phase, then we take the arithmetic mean of the results of the training phase and the testing phase to each part by the following equation:

$$\sum_{i=1}^{n} (\text{target==output})$$

Where the target is the type (0, 1) and the output is the results of input data in the training phase and testing phase.

Chapter five

Experiments and Results:

Built experiences in this research on the classification of video files based on the content to different domains, the video files were classified in two classes. In the first we assemble samples of the video files of different areas, so that included videos fighting, and videos cartons, videos songs, and video films.etc, and educational videos of different disciplines. The video files were classified into educational, and non educational based on motion estimation optical flow for each video file from first frame to last frame. The Artificial Neural Network model (ANN) was built on several experiments using transfer function TANSIG and PURELIN, each experiment was included 274 sample of educational and non-educational for training phase and 137 samples of new data of educational and non-educational in the testing phase, input neurons 450 frames, output neurons is 1. We use different of hidden layer, start from 10 hidden layer then increasing the number of hidden layer in each experiment to the 20 and 30 , where the results showed accurate and satisfactory. And take the best results in the testing phase based on number of hidden layer 30, and worked another experiment based on PURELIN transfer function to and two layers in neural network the results were poor.

Started to build ANN model based on a set of video files, which consists of the two phases, the training phase and the testing phase, we have many experiences, and these experiences were consisting of 411 sample of educational and non-educational and these samples division into three parts of data in neural network, the three parts are AB in the training and C in testing, AC in the training and B in testing and BC in training and A in testing. One of the results of experiments in neural network using transfer function TANSIG as follow:

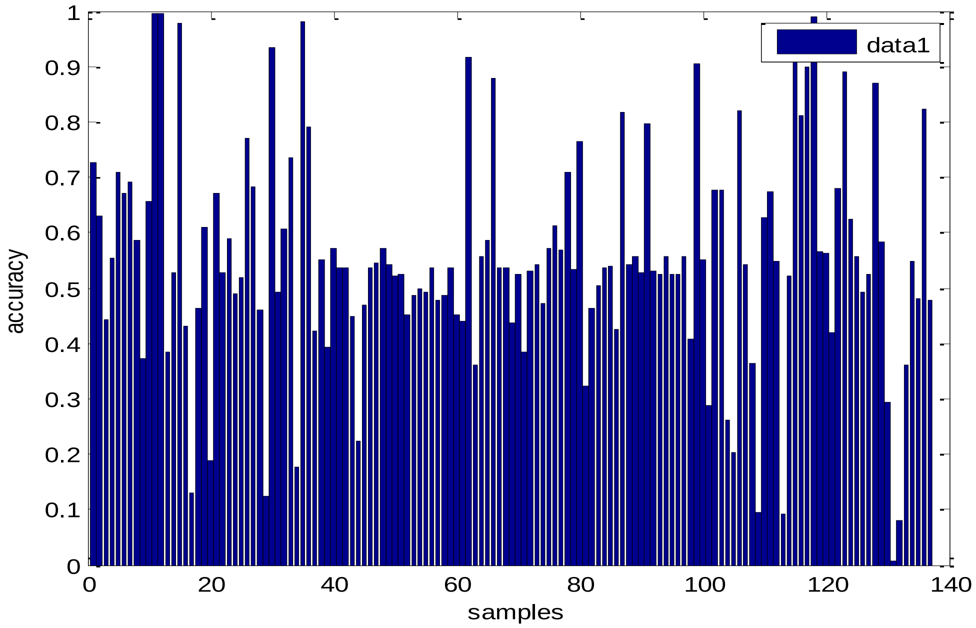


Figure 5.1.1: Overall (bar) AB data of samples of educational and non-educational in the training phase.

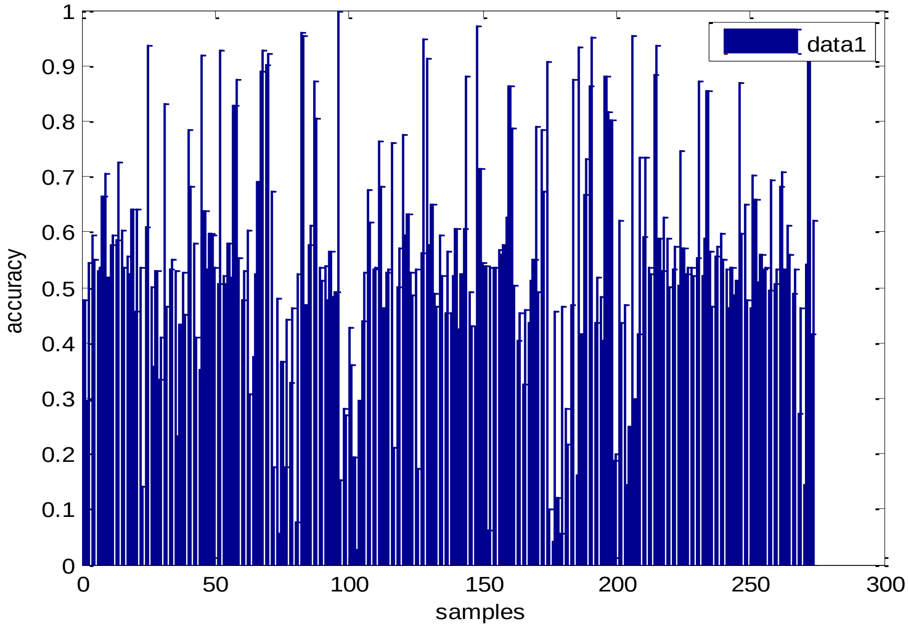


Figure 5.1.2: Overall (bar) C data of samples of educational and non-educational in the testing phase.

As can be shown in fig. (5.1.1) and fig. (5.1.2) the overall bar in Neural network where the number of data AB in the training phase is 274 sample of educational and non-educational, the number of data C in the testing phase is 137 sample of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output and the number of hidden layer is 10.

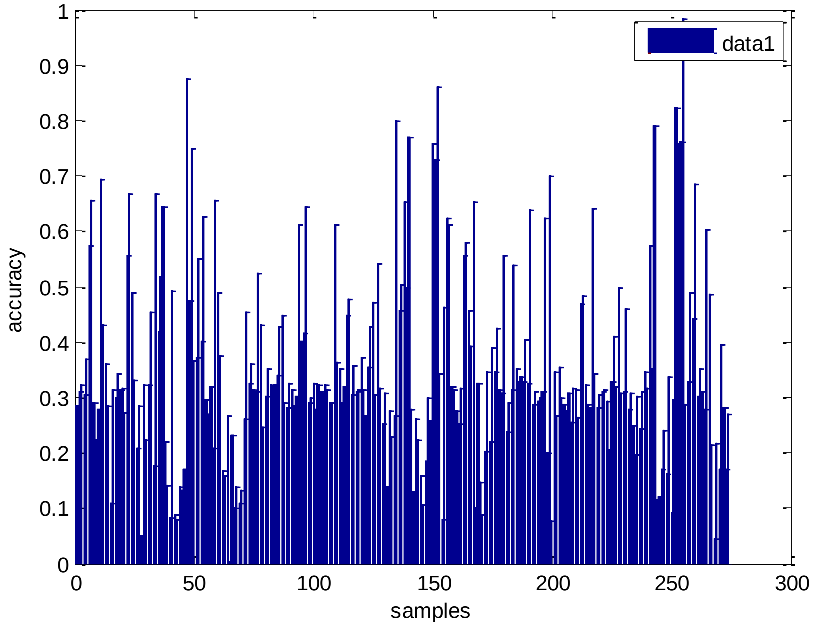


Figure 5.1.3: Overall (bar) BC data of samples of educational and non-educational in the training phase.

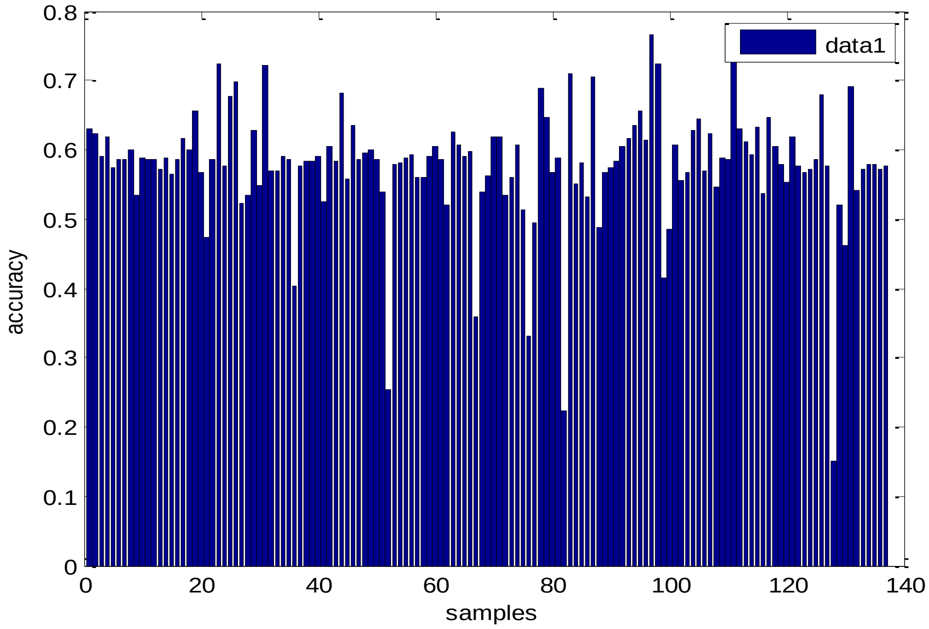


Figure 5.1.4: Overall (bar) a data of samples of educational and non-educational in the testing phase.

As can be shown in fig. (5.1.3) and fig. (5.1.4) the overall bar in Neural network where the number of data BC in the training phase is 274 sample of educational and non-educational, the number of data A in the testing phase is 137 sample of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output and the number of hidden layer is 10.

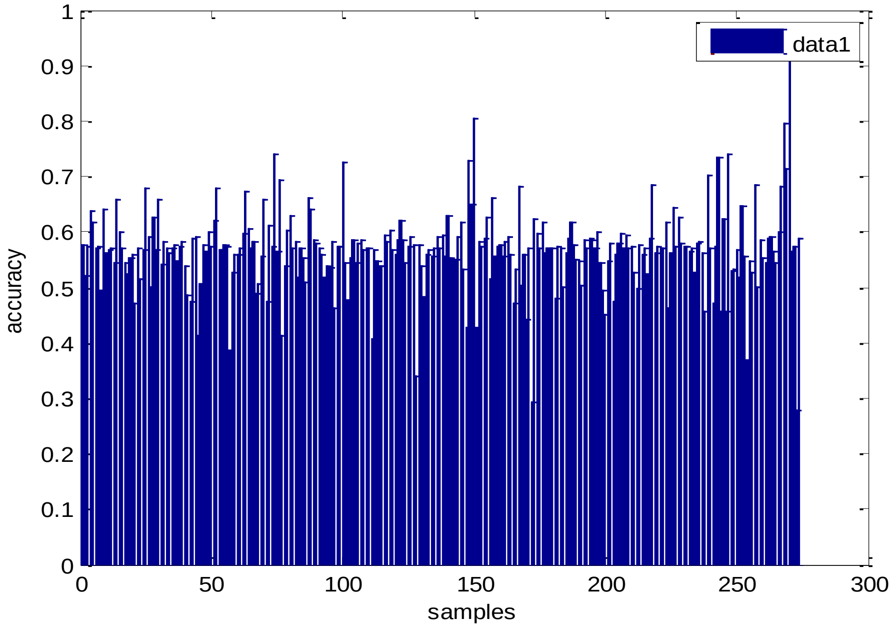


Figure 5.1.5: Overall (bar) AC data of samples of educational and non-educational in the training phase.

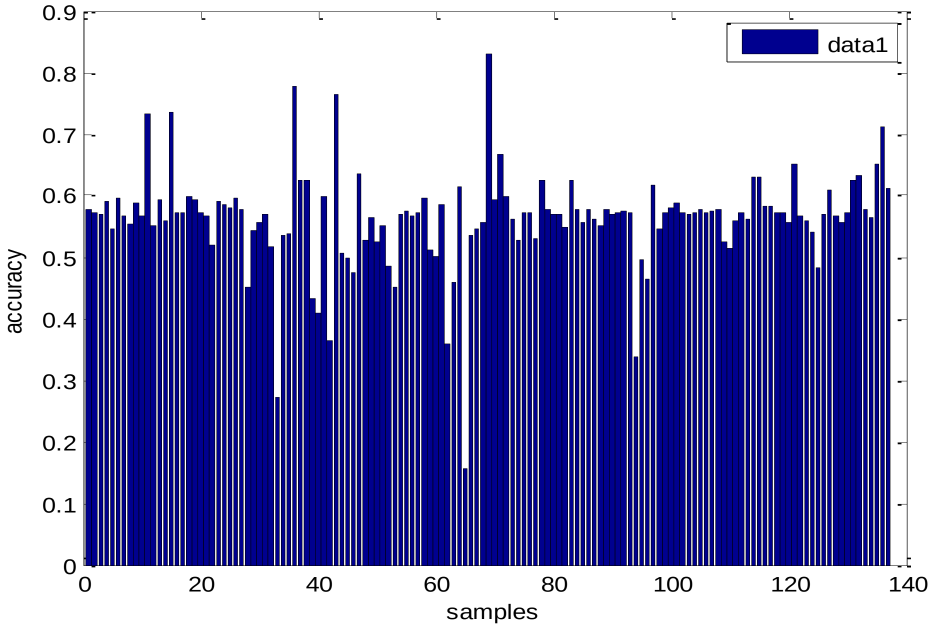


Figure 5.1.6: Overall (bar) B data of samples of educational and non-educational in the testing phase.

As can be shown in fig. (5.1.5) and fig. (5.1.6) the overall bar in Neural network where the number of data AC in the training phase is 274 sample of educational and non-educational, the number of data B in the testing phase is 137 sample of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output and the hidden layer is 10.

The other experiments in Neural network were the data AB in the training is 274 sample and C in testing is 137 sample, AC in the training is 274 sample and B in testing 137 sample and BC in training 274 sample and A in testing 137 sample, and the inputs layer are 450 and the outputs layer is 1 output. We increased the number of hidden layer to 20 and 30 neurons using transfer function TANSIG.

We work another experiment in neural network using transfer function PURELIN, based on the best results of hidden layer is 30. In the training phase the data is AB, the number of data 274 sample of educational and non-educational, in the testing phase the data is C and the number of data 137 sample of educational and non-educational. The results are not satisfactory and poor.

Table 5.1: Summary of experiments according to mean and transfer function TANSIG.

Train/Test	**Hidden 10**	**Hidden 20**	**Hidden 30**
AB train	0.7263	0.5693	0.7555
C test	0.4672	0.5109	0.5328
AC train	0.4489	0.7263	0.6861
B test	0.4964	0.5255	0.5620
BC train	0.6752	0.6460	0.5292
A test	0.4891	0.5547	0.5255

As can be shown in table 5.1: summary of the mean results based on neural network in training phase and testing phase, transfer function TANSIG, the numbers of hidden layer are different hidden layer 10, 20 and 30 is the highest value due to lack of enough space in computer memory.

Table 5.2: Summary of experiments according to mean and transfer function PURLIN.

Train/Test	**Hidden30**
AB train	0.0839
C test	0.1022

As can be shown in table 5.2: summary of the mean results based on neural network in training phase and testing phase, transfer function PURELIN and hidden layer 30.

Table 5.3: Summary of mean and standard deviation

Hidden layer	**Arithmetic mean**	**Standard deviation**
10	0.4842	0.1552
20	0.5303	0.1585
30	0.5401	0.1403

As seen in table 5.3 summary of arithmetic mean and standard deviation for the tesing phase(A, B and C). The values of standard deviation are small and this means it is close to the arithmetic mean.

As seen in the back ground section, Neural Network is a simulation of highly interconnected neurons that provide models of data relationships that accept inputs, apply weighting coefficient and provide their output to be input to other neurons, forward or backward, that continue the process through the network to the final output and these steps are repeated in long and iterative process where the weights applied to each input at each neuron are adjusted to optimize the desired output [58, 59, 60, 61 and 62].

We have made an experiment to a comparison between the Neural Network and YouTube in the Internet, where the results in YouTube are not accurate compared with this search because this research depends on the content analysis but YouTube depends on the comparison between the keyword.

Chapter six

Conclusion

This research aims to develop a specific method for the classification video files to different classes based on analysis content using neural network, facilitate access to the required educational video to the learners students, especially novice students, and to obtain the desired educational video with less time, less effort and direct way. This research is objective to investigate how the effect of motion feature can be useful in such classification. We believe that other effects such audio features, text features, and other factors can enhance accuracy, but this requires wider studies and need more time. The accuracy of results in video classification to educational and non-educational through technique 3 fold cross validation and using (ANN) model were 54%. This results confirm and achieve the objectives of this research.

To test this method in the classification of video files, where was working as follows:

We start to collecting a group of video files that different length and different domains and types, these videos making equal length through assuming that human takes half a minute is sufficient to know that each video is educational or non-educational, through the division of each video to the samples of data of educational and non-educational, so that the size of each sample 1800 frame. If we have a video in size 3600 frame, this video is divided into two samples each sample is 1800 frame. This work was done through calculate the motion of optical flow in each video file from the first frame to the last frame. The number of new data is 411 sample of educational and non-educational and the input to the neural network are 450 .The results in the classification and prediction samples are satisfactory. And we work another experiment to Comparison between YouTube and Neural network where the results of YouTube is not accurate while neural network is accurate and satisfactory to get the required results. The results of neural network are better than YouTube because this research depends on the content analysis while YouTube depends on the comparison between the keywords.

As future work, we intend to expand our work to work on video classification to different classes through using other effects such as audio features,text features and other factors can enhance accuracy.

References

[1]Ramadoss, B.(2008): Discovering Knowledge from Multi-modal Lecture Recordings, proceedings of ICDEM , Tiruchirappalli, India, pp. 1-3, 2008.

[2]Huang, W., Eze, E., Webster, D.(2006): Towards integrating semantics of multi-media resources and processes in e-Learning Multimedia Systems, Springer Berlin / Heidelberg, Vo. 11, No. 3, Mars,2006.

[3]lecture recording , January.(2012): (http://en.wikipedia.org/wiki/Lecture_recording).

[4] Christian, Vanessa, Angelo, Frank.(2009): Indexing and Retrieval of Multimodal Lecture Recordings from Open Repositories for Personalized Access in Modern Learning Settings ,2009.

[5] Andreas and Günter.(2000): Combining Multimedia Retrieval and Text Retrieval to Search Structured Documents in Digital Libraries, Otto-Friedrich University of Bamberg, Faculty of Social and Economic Sciences Feldkirchenstraße 21, D-96045 Bamberg, Germany, 2000.

[6]Don Adjeroh, M. C. Lee, N. Banda, and Kandaswamy.(2009): Adaptive Edge-Oriented Shot Boundary Detection,Hindawi Publishing Corporation Eurasip Journal on Image and Video Processing Volume 2009, Article ID 859371, 13 pages doi:10.1155/2009/859371, 18 May2009.

[7]AmelMalekAbdelati, Jlassibahaeddine, Abdessalem and AbdellatifMtibaa.(2008): Suivid'objet en movement par la method Level Set variationnelle,International workshop on system engineering design & application,Monastir, Tunisia, October 24-26,2008.

[8]AbdelatiMalekAmel, Ben AbdelaliAbdessalem and MtibaaAbdellatif.(2010): Video shot boundary detection using motion activity descriptor,Jouranal of telecommunications, volume 2, issue 1, april 2010.

[9]M-petkovic Centre for Telematics and Information Technology.(2010): Content-based video retrieval, University of TwenteP.O. Box 217, 7500 AE Enschede, The NetherlandsPhone: +31 53 4893725, Fax: +31 53 4894524,2010.

[10] Zhu, Jincheng, Yao Wang.(1999): Audio feature extraction & analysis for scene classification, National Science Foundation under Grant N-IRI-9619114 and by the New York State Center for Advanced Technology in Telecommunicationsat Polytechnic University, Brooklyn, New York,1999.

[11]Jiawei, MichelineKamber.(2001): Data Mining: Concepts and Techniques Published by Morgan Kaufmann,500Sansome Street, Suite 400, San Francisco, CA 94111,2001.

[12] Akram M. Othman, and May H. Riadh.(2008): Speech Recognition Using Scaly Neural Networks, World Academy of Science, Engineering and Technology, vol. 38,2008.

[13] Mohamad Adnan Al-Alaoui, Lina Al-Kanj, Jimmy Azar, and Elias Yaacoub.(2008): Speech Recognition using Artificial NeuralNetworks and Hidden Markov Models, IEEE multidisciplinary engineering education magazine, VOL. 3,2008.

[14] Dou-Suk and Young Lee.(1994): Intelligent judge neural network for speech recognition, Neural Processing Letters , Vol 1,1994.

[15] Peng Lim, Chan Woo, Sim Loh, Rohaizan. (2000):Speech Recognition Using Artificial Neural Networks, wise, vol. 1, pp.0419, first international conference on web information systems engineering (WISE'00)-Volume 1, 2000.

[16] Using MATLAB Version 6 The MathWorks Inc, Natick, MA, 2002.

[17]UmeshRajashekar, Student Member, IEEE and Alan C Bovik, Fellow, IEEE.Interactive dsp education using matlab demos, 2001.

[18] Jun Yang1, Rong Yan2, Yan Liu2 and Eric P. Xing1.(2008): Harmonium Models for Video Classification,Published online 18 January in Wiley InterScience, 2008.

[19]Beverly L. Harrison, Ronald M. Baecker.(2002): Designing Video Annotation and Analysis Systems,Appears in Proceedings Graphics Interface '92, 1992, 157-166,2002.

[20] Darin Brezeale Diane J. Cook,Pullman.(2009):Learning Video Preferences Using Visual Features and Closed Captions,publishedby the IEEE Computer Society,2009.

[21] Rainer Lienhart, Silvia Pfeiffer and Wolfgang Effelsberg.(1997) : "Video Abstracting", University of Mannheim, 68131 Mannheim, Germany, Communications of ACM, pp. xx – yy, December 1997.

[22]AbdelatiMalekAmel, Ben AbdelaliAbdessalem and MtibaaAbdellatif.(2010): Video shot boundary detection using motion activity descriptor, jornal of telecommunications ,volume 2, issue 1, april 2010 .

[23]Zhu Liu, Jincheng Huang and Yao Wang Department of Electrical Engineering Polytechnic University, Brooklyn .(1999): Classification lassification of tv programs based on audio information using hidden markov, supported by the National Science Foundation through its stimulate program under Grant IRI9619114Polytechnic University, Brooklyn, NY 11201,1999.`

[24]Jun Yang, Alexander G. Hauptmann. (2005): Multi-modal Analysis for Person Type Classification in News Video, IS&T/SPIE Symposium on Electronic Imaging, San Jose, CA, January 18-19, pp. 165-172, 2005.

[25]HuitaoLuo .(2000): Algorithms for Video Object Detection and Segmentation with Application to Content-Based Multimedia Systems,Columbia University,2000.

[26] Wouter Gevaert, Georgi Tsenov, Valeri Mladenov. (2010): Neural Networks used for Speech Recognition,journal of automatic control, university of belgrade, vol. 20:1-7,2010.

[27]Lihi Zelnik-Manor Michal. (2001): Event-Based Analysis of Video, Irani,Dept. of Computer Science and Applied Math The Weizmann Institute of Science,76100 Rehovot,2001.

[28]nilesh v.patel and ishoar k.(1996):audio characterization for video indexing,Detroit,MI Proc. SPIE 2670, Storage and Retrieval for Still Image and Video Databases IV, 373 March 13, 1996.

[29]Nerijus remeikis, Ignas skucas, Vida melninkaite. (2004): Text Categorization Using Neural NetworksInitialized with Decision Trees,Faculty of Informatics, Vytautas Magnus UniversityVileikos 8, 3035 Kaunas, Lithuania, 2004.

[30]Raudys, S. (2001): Statistical and Neural Classifiers: an Integrated Approach to Design. Springer-Verlag, NY,2001.

[31] Manish Mehta, Rakesh Agrawal and Jorma Rissanen(2007): SLIQ: A Fast Scalable Classifier for Data Mining, IBM Almaden Research Center 650 Harry Road, Jose, CA 95120.

[32] Tianshi Gao.and Daphne Koller(2011): Active Classification based on Value of Classifier, NIPS-2011.dvi

[33] Han, Jiawei.; Kamber, Micheline(2006): Data Mining : Concepts and Techniques,second edition,modified 20011.

[34] Thair Nu Phyu(2009): Survey of Classification Techniques in Data Mining, Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong.

[35] John Langford. (2005): Tutorial on Practical Prediction Theory for Classification, Journal of Machine Learning Research 6 (2005) 273–306.

[36] Tom M. Mitchell. 1(997), Machine Learning. McGraw Hill,3rd ed.ispn, vol 2, 1997.

[37]Han J . and Kamber M. (2007):Data Mining :Concepts and Techniques- Second Edition,Elsevier Inc.

[38] DTREG (2012):Multilayer Perceptron Neural Networks, A Brief History of Neural Network.(http://www.dtreg.com/mlfn.html).

[39]Smith L.(April 2003):An Introduction to Neural Networks.(http://www.cs.stir.ac.uk/~1ss/NNIntro/InvSlides.html#what).

[40] NeruoSolutions (2012): Introduction to Neural Networks and NeuroSoluations-Video Presentation .

(http://www.nddownloads1.com/videos/nns_and_neurosolutions/nns_and_neurosolutions.html).

[41] SPSS Inc. (2012): PASW Neural Networks 18-statistical package for the social sciences Chicago, USA. (www.spss.com).

[42] Daniel Svozil, Vladimir KvasniEka, JiE Pospichal .(1997): Tutorial Introduction to multi-layer feed-forward neural networks, Chemometrics and Intelligent Laboratory Systems 39 (1997) 43-62, 6 June 1997.

[43]Aberdeens Robert Gordon.(2004):The back propagation algorithm, February (2004).

[44] Mark Hudson Beale, Martin T. Hagan, Howard B. (2013):Demuth Neural Network Toolbox User’s Guide, The MathWorks, Inc. 3 Apple Hill Drive Natick, MA 01760-2098,2013.

[45] neural network , (2013):feedforword neural network (http://www.biochem.ucl.ac.uk/~shepherd/sspred_tutorial/ss-nn.html)

[46] Praveen Kumar Edara .(2003) : Mode Choice Modeling Using Artificial Neural Networks , Falls Church, Virginia, October 2003.

[47] Nitin N. Pise.(2006): Document Classification using Neural Networks , Maharshra Institute of Technology, Pune, India,2006.

[48] Włodzisław Duch , Norbert Jankowski(1991): Survey of Neural Transfer Functions, Department of Computer Methods, Nicholas Copernicus University,ul. Grudzia˛dzka 5, 87-100 Torun´ , Poland.

[49] Norbert Jankowski , Włodzisław Duch(2001): Optimal transfer function neural networks, Department of Computer Methods Nicholas Copernicus University ul.Grudzia˛dzka 5, 87–100 Torun´, Poland.

[50] Dr. Nicholas Beser ,1998:" Video Compression Motion Estimation".

[51]S.Baker,D Scharstine ,J .Lewise ,S.Roth ,M.J.Black .(2007): A database and evaluation methodology for optical flow ",In /CCV,2007.

[52]Deqing Sun, Stefan Roth,Michael J. Black.(2010): Secrets of Optical Flow Estimation and Their Principles,2010.

[53] Bhavana C. Bendale, Prof. Anil R. Karwankar. (2012): Moving Object Tracking in Video Using MATLAB",International Journal of Electronics, Communication & Soft Computing Science and EngineeringISSN: 2277-9477, Volume 2, Issue 1,2012.

[54] Shah, Anchor Kutchhi Polytechnic (2012): CONTENT BASED VIDEO RETRIEVAL SYSTEMS, International Journal of UbiComp (IJU), Vol.3, No.2, and April 2012.

[55] T.N.shanmugam, priya rajendran, (2009): An enhanced content-based video retrieval system based on query, International Journal of Research and Reviews in Applied Sciences ISSN: 2076-734X, EISSN: 2076-7366 Volume 1, Issue 3(December 2009).

[56]Akitsugu Noguchi and Keiji Yanai (2009):Extracting Spatio-temporal Local Features Considering Consecutiveness of Motions, 182-8585, Japan, noguchi-a@mm.cs.uec.ac.jp, yanai@cs.uec.ac.jp, 9th Asian Conference on Computer Vision, Xi'an, September 23-27, 2009.

[57] payamrefaeilzadeh, lei tang, huan liu (2008): Cross-Validation, Comp. by: BvijayalakshmiGalleys0000875816 Date:6/11/08 Time:19:52:53 Stage:First Proof FilePath://ppdys1108/Womat3/Production/prodenv/0000000005/0000008302/0000000016/0000875816.3D Proof by: qc

[58] Jae H. Song, MS, Santosh S. Venkatesh, PhD, Emily A. Conant, MD, Peter H. Arger, MD, Chandra M. Sehgal, PhD (2005) : Comparative Analysis of Logistic Regression and Artificial Neural Network for Computer-Aided Diagnosis of Breast Masses-Academic Radiology,Vol 12 ,No 4, April 2005.

[59]Khemphila A .,Boonjing V .(2010): Comparing performances of logistic regression, decision trees, and neural networks for classifying heart disease patients-Computer Information Systems and Industrial Management Applications (CISIM),2010 International Conference on ,vol.,no., pp.193-198, 8-10 Oct.2010.

[60] Bouzida Y .and Cuppens F. (2006): Neural networks vs. decision trees for intrusion Detection-IEEE / IST Workshop on Monitoring , Attack Detection and Mitigation,2006.

[61] Chandra R., Chaudhary K. and Kummar A.(2007): The Combination and Comparison of Neural Networks with Decision Trees for Wine Classification- School of Science and Technology, The University of Fiji, Lautoka, Fiji 2007.

[62] Kumar K.,(2012):Knowledge Extraction from trained &Network Security (IJINS),Vo1.1, No.4,October 2012,pp.282.

APPENDIX 1: Estimate Motion for Each Video File Using Optical Flow in Matlab tools

```
videoReader=vision.VideoFileReader('funny.avi','ImageColorSpace','Intensity','VideoOutputDataType','uint8');
converter = vision.ImageDataTypeConverter;
opticalFlow = vision.OpticalFlow('ReferenceFrameDelay', 1);
opticalFlow.OutputValue = 'Horizontal and vertical components in complex form';
shapeInserter = vision.ShapeInserter('Shape','Lines','BorderColor','Custom',
'CustomBorderColor', 255);
VideoPlayer = vision.VideoPlayer ('Name','Motion Vector');
Frame = Video Reader ('funny.avi');
 z= frame.NumberOfFrames;
 u=z/1800; %division each video file to 1800 samples%
e=fix (u); % the rest =1%
%filename = 'test1.xlsx';
c=0; q=0;  r=0;
  q=1800*e;
While c<q
    c=c+1;
    Frame = step (video Reader);
    im = step(converter, frame);
    Of = step (optical Flow, im);
    %hold on;
    %plot(c,mean(real(of(:))),'-b',c,mean(imag(of(:))),'-r');
     ofreal(c)=mean(real(of(:)));
    ofimag(c)=mean(imag(of(:)));
 %    lines = videooptflowlines(of, 20);
  %       if ~isempty(lines)
  %     out =  step(shapeInserter, im);
       step(videoPlayer, im);
   %  end
end
%xlswrite(filename,ofreal);
%xlswrite(filename,ofimag);
%hold off
 release(videoPlayer);
release(videoReader);
```

APPENDIX 2: Building Neural Network

```
% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by NPRTOOL
% Created Thu Nov 08 18:54:56 IST 2012
%
% This script assumes these variables are defined:
%
%   data - input data.
%   type - target data.

inputs = data';
targets = type';

% Create a Pattern Recognition Network
hiddenLayerSize = 10;
net = patternnet(hiddenLayerSize);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};


% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% For help on training function 'trainlm' type: help trainlm
% For a list of all training functions type: help nntrain
net.trainFcn = 'trainlm';  % Levenberg-Marquardt

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse';  % Mean squared error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
```

```
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
  'plotregression', 'plotfit'};
% Train the Network
[net,tr] = train(net,inputs,targets);
% Test the Network
outputs = net(inputs);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs);
% Recalculate Training, Validation and Test Performance
trainTargets = targets .* tr.trainMask{1};
valTargets = targets  .* tr.valMask{1};
testTargets = targets  .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,outputs);
valPerformance = perform(net,valTargets,outputs);
testPerformance = perform(net,testTargets,outputs);
% View the Network
view(net)
% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, plotconfusion(targets,outputs)
%figure, ploterrhist(errors)
```

APPENDIX 3 :Cross Validation

```
%   [1] Hastie, T. Tibshirani, R, and Friedman, J. (2001) The Elements of
%       Statistical Learning, Springer, pp. 214-216.

% Copyright 2007-2010 The MathWorks, Inc.
% $Revision: 1.1.8.4 $  $Date: 2010/10/08 17:23:05 $

if nargin < 2
    error(message('stats:crossval:TooFewInputs'));
end

firstInputType = 'func';
if ischar(funorStr)
    funorStr = lower(funorStr);
    if ~ (strcmp(funorStr,'mse') || strcmp(funorStr,'mcr'))
        error(message('stats:crossval:BadFun'));
    else
        firstInputType = 'lossMeasure';
    end
elseif ~isa(funorStr,'function_handle')
    error(message('stats:crossval:BadFun'));
end

n = size(varargin{1},1);
if n <= 1
    error(message('stats:crossval:TooFewDataRows'));
end


nData = length(varargin);
for i = 2:length(varargin);
    if size(varargin{i},1) ~= n
        if ~(ischar(varargin{i}) && size(varargin{i},1) ==1 )
            error('stats:crossval:MismatchedDataRows',...
                'Data arguments X,Y,... must have the same number of rows.');
        else
            nData = i-1;
            break;
        end
    end
end

if strcmp(firstInputType, 'lossMeasure') && nData < 2
    error(message('stats:crossval:NotEnoughVars'));
end

data = varargin(1:nData);
varargin(1:nData)= [];
```

```
pnames = { 'kfold'  'holdout'  'leaveout' 'mcreps' 'stratify' 'partition' 'predfun' ...
    'options' };
dflts =  { []       []         []       1         []        []      []       ...
    statset('crossval') };
[eid,errmsg,nfolds,holdout, leaveout, mcreps, stratify, cvp, predfun, parallelOption] ...
    = internal.stats.getargs(pnames, dflts, varargin{:});
if ~isempty(eid)
    error(sprintf('stats:crossval:%s',eid),errmsg);
end

if strcmp(firstInputType,'lossMeasure')
    if isempty(predfun)
        error(message('stats:crossval:MissingPredfun'));
    end
    if size(data{end},2) ~= 1
        error(message('stats:crossval:YnotColumnVector'));
    end

else %the first input is a function handle
    if ~isempty(predfun)
        warning(message('stats:crossval:UnneededPredfun'));
    end
end

if ~isempty(leaveout) && leaveout ~= 1
    error(message('stats:crossval:UnsupportedLeaveout'));
end

if ~( isnumeric(mcreps) && isscalar(mcreps) && mcreps == round(mcreps)...
        && mcreps >= 1)
    error(message('stats:crossval:BadMcreps'));
end

choices=[];

cvopts = sum( [~isempty(holdout), ~isempty(nfolds) , ~isempty(leaveout)]);
if cvopts > 1
    error(message('stats:crossval:InconsistentOpts'));
elseif ~isempty(cvp)
    if cvopts >0 || ~isempty(stratify)
        error('stats:crossval:InconsistentOpts',...
            ['"Partition" cannot be used with "Kfold","Holdout" ',...
            '"Leaveout" or "Stratify".']);
    elseif ~isa(cvp,'cvpartition')
        error(message('stats:crossval:Badcvp'));
    elseif cvp.N ~= n
        error('stats:crossval:MismatchedDataRows',...
            'The "N" property of "PARTITION" must equal the number of rows in X.');
    end
    choices = 'cvpartition';
    if mcreps > 1 && (strcmp(cvp.Type,'resubstitution') ...
            || strcmp(cvp.Type,'leaveout'))
        mcreps = 1;
        warning('stats:crossval:InvalidMcreps',...
            '"Mcreps" is set to 1 for Leave-one-out cross-validation or Resubstitution.');
    end
else
    if ~isempty(stratify)
        stratify = grp2idx(stratify);
        if size( stratify,1) ~= n
            error('stats:crossval:MismatchedDataRows',...
                '"Stratify" must have the same number of rows as X.');
        end

    end
    if cvopts == 0
        choices = 'kfold';
        cvarg = 10; %default cross-validation option
    elseif ~isempty(nfolds)
        choices = 'kfold';
        cvarg = nfolds;
    elseif ~isempty(holdout)
        choices ='holdout';
        cvarg = holdout;
    elseif ~isempty(leaveout)
        if mcreps > 1
            mcreps = 1;
            warning('stats:crossval:InvalidMcreps',...
                'The value of "Mcreps" is set to 1 for Leave-one-out cross-validation.');
        end
        choices = 'leaveout';
        cvarg = 1;
    end
end

% Process options parallel computation and random number streams.
[useParallel, RNGscheme, poolsz] = ...
    internal.stats.parallel.processParallelAndStreamOptions( ...
        parallelOption,false);
useSubstreams    = RNGscheme.useSubstreams;
streams          = RNGscheme.streams;
useDefaultStream = RNGscheme.useDefaultStream;

if useDefaultStream
    s = RandStream.getDefaultStream;
else
    s = streams{1};
end
```

```
substreamOffset = internal.stats.parallel.freshSubstream(s);
substreamOffset = substreamOffset -1 ;

% Process cvpartition(s)
if ~strcmp(choices,'cvpartition')
    if isempty(stratify)
        cvp = cvpartition(n, choices,cvarg, s);
    else
        cvp = cvpartition(stratify, choices,cvarg, s);
    end
    % We used up a Substream to make the first cvpartition.
    substreamOffset = substreamOffset+1;
end

nData = size(data,2);
numtests = cvp.NumTestSets;
numiter = numtests * mcreps;

if useParallel
    % Get all re-randomized cv partitions up front
    cvparray = cell(mcreps,1);
    cvparray{1} = cvp;
    for j=2:mcreps
        if useSubstreams
            set(s,'Substream',j+substreamOffset);
        end
        cvparray{j} = cvp.repartition(s);
    end
end

% Perform the function evaluations
if strcmp(firstInputType,'func') % the first input is a function handle

    % Initialize before looping to allocate the output array
    funResult = getFuncVal(1, nData, cvp, data, funorStr, []);
    szFunResult = size(funResult);
    loss(numiter,:) = funResult(:)';
    loss(1,:) = funResult(:)';

    if useParallel
        parfor j=2:numiter
            mcrep = 1 + floor((j-1)/numtests);
            i = 1 + mod(j-1,numtests);  % i-th fold within this mc rep
            cvpj = cvparray{mcrep};
            funResult = getFuncVal(i, nData, cvpj, data, funorStr, szFunResult);

            % Accumulate the results
            loss(j,:) = funResult(:)';
        end  % parfor mr=1:mcreps
    else
```

```
        for mr = 1:mcreps
            if (mr==1)
                loopstart=2;
            else
                loopstart=1;
                if useSubstreams
                    set(s,'Substream',mr+substreamOffset);
                end
                cvp = cvp.repartition(s);
            end

            offset = cvp.NumTestSets * (mr-1);

            for i = loopstart:cvp.NumTestSets
                funResult = getFuncVal(i, nData, cvp, data, funorStr, szFunResult);
                loss(offset+i,:) = funResult(:)';
            end
        end  % for mr=1:mcreps
    end

else % the first input is the loss measure

    loss = zeros(numtests,1);
    ismse = strcmp(funorStr,'mse');
    if ~ismse
        data{end}=nominal(data{end}); % mcr => use nominal response
    end

    if useParallel

        parfor j=1:numiter
            mcrep = 1 + floor((j-1)/numtests);
            i = 1 + mod(j-1,numtests);  % i-th fold within this mc rep
            cvpj = cvparray{mcrep};

            [funResult,outarg] = getLossVal(i, nData, cvpj, data, predfun);

            % Accumulate the results
            if ismse
                temploss = sum((outarg-funResult).^2);
            else
                if ~(isnumeric(funResult) || isa(funResult, 'nominal'))
                    funResult = nominal(funResult);
                end
                temploss = sum(outarg ~= funResult);
            end
            loss(j,:) = temploss;
        end

    else
```

```
        for mr = 1:mcreps
            if mr > 1
                if useSubstreams
                    set(s,'Substream',mr+substreamOffset);
                end
                cvp = cvp.repartition(s);
            end
            offset = cvp.NumTestSets * (mr-1);

            for i = 1:cvp.NumTestSets
                [funResult,outarg] = getLossVal(i, nData, cvp, data, predfun);

                % Accumulate the results
                if ismse
                    temploss = sum((outarg-funResult).^2);
                else
                    if ~(isnumeric(funResult) || isa(funResult, 'nominal'))
                        funResult = nominal(funResult);
                    end
                    temploss = sum(outarg ~= funResult);
                end
                loss(offset+i,:) = temploss;
            end
        end

    end

    loss = sum(loss)/ (mcreps * sum(cvp.TestSize));

end  % if/else firstInputType

%
% === Clean-up ====
%

% If the user passed a 'Streams' argument, then we may need to update
% or restore state of the stream(s) that were passed.
% If neither 'UseParallel' nor 'UseSubstreams' was selected, however,
% we do nothing here, because we want the effects we had on the stream state
% to persist outside the function call.

if useSubstreams

    % If we are here, the user passed in a stream, which we used.
    % On entry, that stream had Substream set to substreamOffset.
    % If no cvpartition was passed to us, we constructed one, and
    % incremented substreamOffset to reflect that we had "used up"
    % one Substream.  Afterwards, we used an additional Substream
    % for each of the remaining (mcreps-1) monte carlo repetitions.
    % On exit, we increment the Substream just beyond the last value
    % that we used. This will keep sequences produced by the stream
    % within this function segregated (ie, non-overlapping) from
    % streams produced outside the function, assuming that outside
    % the function the user does not reassign the Substream within
    % the range that we used. This action occurs on the client.
    % There is no need (or advantage) in going to the workers,
    % even if parallel.

    %s = assignStream(streams,useSubstreams);
    set(s,'Substream',substreamOffset+mcreps);

end

end  % of crossval()


function funResult = evalFun(fun,arg)
try
    funResult = fun(arg{:});
catch ME
    if strcmp('MATLAB:UndefinedFunction', ME.identifier) ...
            && ~isempty(strfind(ME.message, func2str(fun)))
        error(message('stats:crossval:FunNotFound', func2str( fun )));
    else
        error(message('stats:crossval:FunError', func2str( fun ), ME.message));
    end
end

end  % of evalFun()

function funResult = getFuncVal(i, nData, cvp, data, funorStr, szFunResult)
arg = cell(2*nData,1);
train = cvp.training(i);
test = cvp.test(i);
% Take subsets of the inputs
for k = 1:nData
    arg{k} = data{k}(train,:);
    arg{nData+k} = data{k}(test,:);
end

% Apply the function to the current subset
funResult = evalFun(funorStr,arg(:));

% Check that size is okay
if ~isempty(szFunResult) && ~isequal(size(funResult),szFunResult)
    error(message('stats:crossval:FunOutSizeMismatched', func2str( funorStr )));
end

end  % of funResult()
```

```
function [funResult,outarg] = getLossVal(i, nData, cvp, data, funorStr)
arg = cell(2*nData,1);
train = cvp.training(i);
test = cvp.test(i);
% Take subsets of the inputs
for k = 1:nData
    arg{k} = data{k}(train,:);
    arg{nData+k} = data{k}(test,:);
end

% Apply the function to the current subset
funResult = evalFun(funorStr,arg(1:end-1));
outarg = arg{end};

% Check that the size is okay
if ~isequal(size(funResult), size(outarg) )
    error('stats:crossval:badFunResultSize',...
        ['The output of ''Predfun'' must a column vector with the ',...
        'number of rows as the number of samples in the',...
        'corresponding test set.']);
end

end  % of getLossVal()

function useParallel = checkOptions(useParallel, useSubstreams, streams)

if length(streams)>1
    MEstream = MException('stats:crossval:BadOptions:Streams', ...
        '''Streams'' parameter must be a scalar.');
    throw(MEstream);
end

if ~isa(streams{1},'RandStream')
    MEstream = MException('stats:crossval:BadOptions:Streams', ...
        '''Streams'' parameter must be a RandStream object.');
    throw(MEstream);
end

% devolve to serial if no parallel environment
if useParallel
    if ~isempty(ver('distcomp'))
        % PCT installed and have license
        poolsz = matlabpool('size');
        if poolsz<1
            % Switching to serial
            warning(message('stats:crossval:NoMatlabpool'));
        end
    end
end
```

```
if useSubstreams
   % Position the starting Substream for bootstrap iterations.
   % We stay where we are if the current substream is at its initial
   % position (has not been used). We increment if the substream has been
   % used. This is to protect against inadvertent overlapping sequences.
   s = streams{1};
   substreamOffset=s.Substream;
   streamStateOnEntry=s.State;
   % first make sure that this RandStream type supports Substreams
   try
      s.Substream = substreamOffset+1;
   catch ME
      % Presumably, if the try failed, the supplied stream is unaltered.
      % But just in case, reset initial state.
      s.Substream = substreamOffset;
      s.State = streamStateOnEntry;
      MEboot = MException('stats:crossval:BadOptions:Streams', ...
         ME.identifier);
      throw(MEboot);
   end
end

end % of checkOptions()

function [useParallel, useSubstreams, streams, substreamOffset] = ...
   processOptions(options)

% First type-check the options
try
   statset(options);
catch ME
   rethrow(ME);
end

useParallel     = strcmpi(statget(options,'UseParallel'), 'always');
useSubstreams   = strcmpi(statget(options,'UseSubstreams'), 'always');
streamArg       = statget(options,'Streams');

% Repackage the Streams argument
if isempty(streamArg)
      streams{1} = RandStream.getDefaultStream;
elseif ~iscell(streamArg)   % we handle stream arguments with a cell array
   streams = {streamArg};
else
   streams = streamArg;
end

% Save some state that Options parameter validity checking will alter
if useSubstreams
```

```
    s = streams{1};
    substreamOnEntry=s.Substream;
    streamStateOnEntry=s.State;
end

% Check for valid Options parameters
useParallel = checkOptions(useParallel, useSubstreams, streams);

% This variable is used only with 'UseSubstreams' selected, but with
% parfor the parser requires definitions even when the value is not used.
substreamOffset = 0;

if useSubstreams
    % Position the starting Substream for monte carlo iterations.
    % The intent is to stay put if the current substream is at its initial
    % position (ie, has not been used). We increment if the substream
    % has been used. This is to protect against inadvertent overlapping
    % sequences.  substreamOffset is set to one less than the first
    % substream we want to be used going forward.

    s = streams{1};
    s.Substream = substreamOnEntry;
    if s.State == streamStateOnEntry
        substreamOffset = substreamOnEntry-1;
    else
        substreamOffset = substreamOnEntry;
    end
end

end   % of processOptions()
```

APPENDIX 4: Overall (bar) AB data of samples of educational and non-educational in the training phase

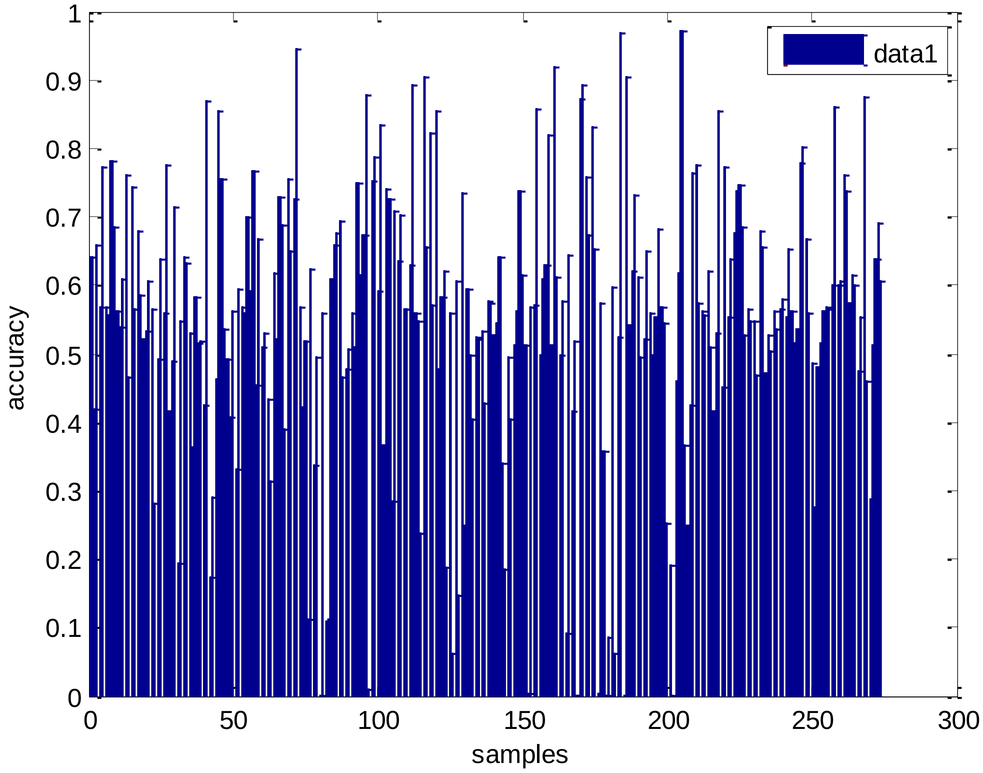


Figure 1: Overall (bar) AB data of samples of educational and non-educational in the training phase.

As can be shown in fig .(1) the overall bar in Neural network where the number of data AB is 274 sample in the training phase of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output and the number of hidden layer is 20.

APPENDIX 5: Overall (bar) C data of samples of educational and non-educational in the testing phase

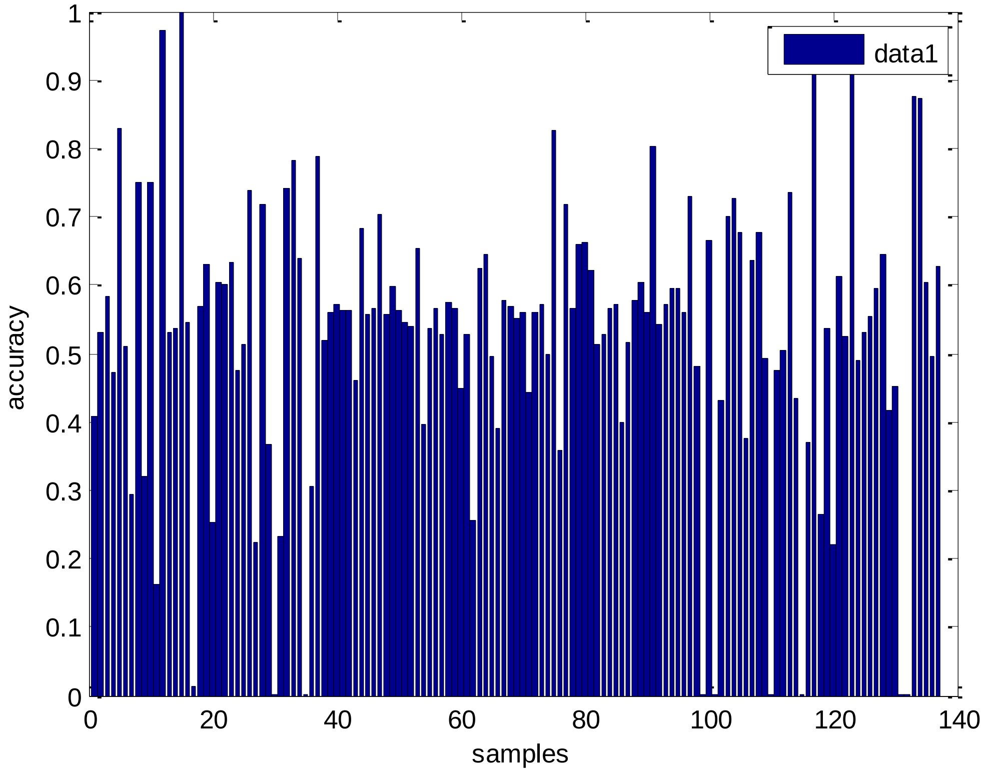


Figure 2: Overall (bar) C data of samples of educational and non-educational in the testing phase.

As can be shown in fig .(2) the overall bar in Neural network where the number of data C 274 sample of data in the testing phase of educational and non-educational, the inputs layer are 450 sample of data and the outputs layer is 1 output and the number of hidden layer is 20.

APPENDIX 6: Overall (bar) AC data of samples of educational and non-educational in the training phase.

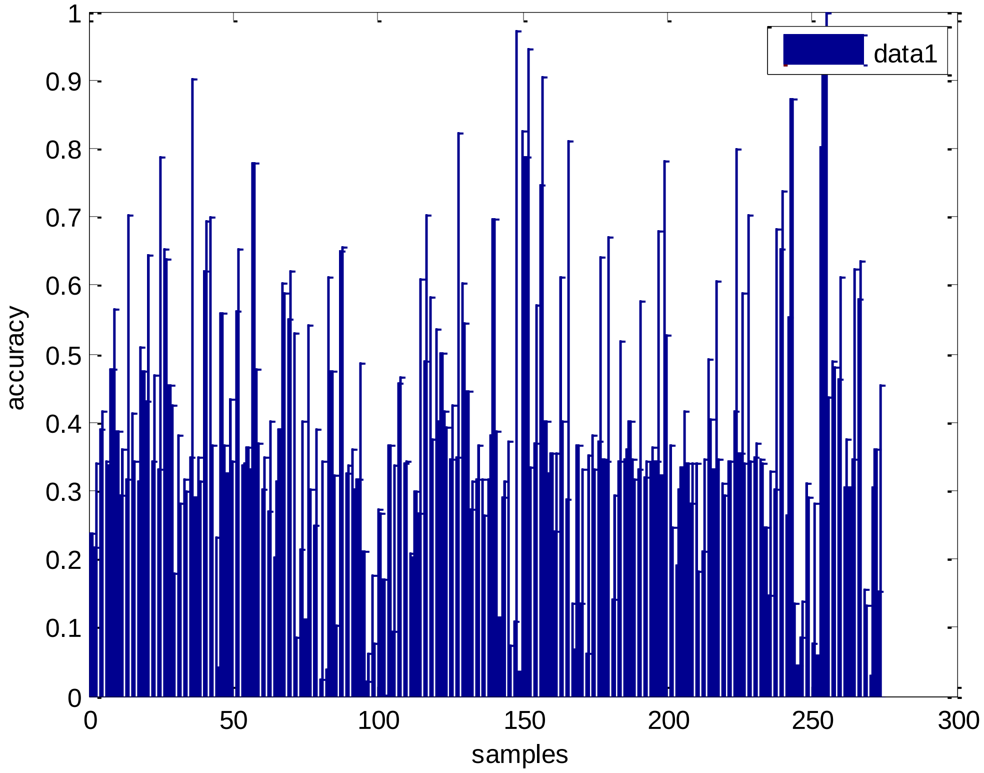


Figure 3: Overall (bar) AC data of samples of educational and non-educational in the training phase.

As can be shown in fig .(3) the overall bar in Neural network where the number of data AC is 274 samples in the training phase of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output and the number of hidden layer is 20.

APPENDIX 7: Overall (bar) B data of samples of educational and non-educational in the testing phase

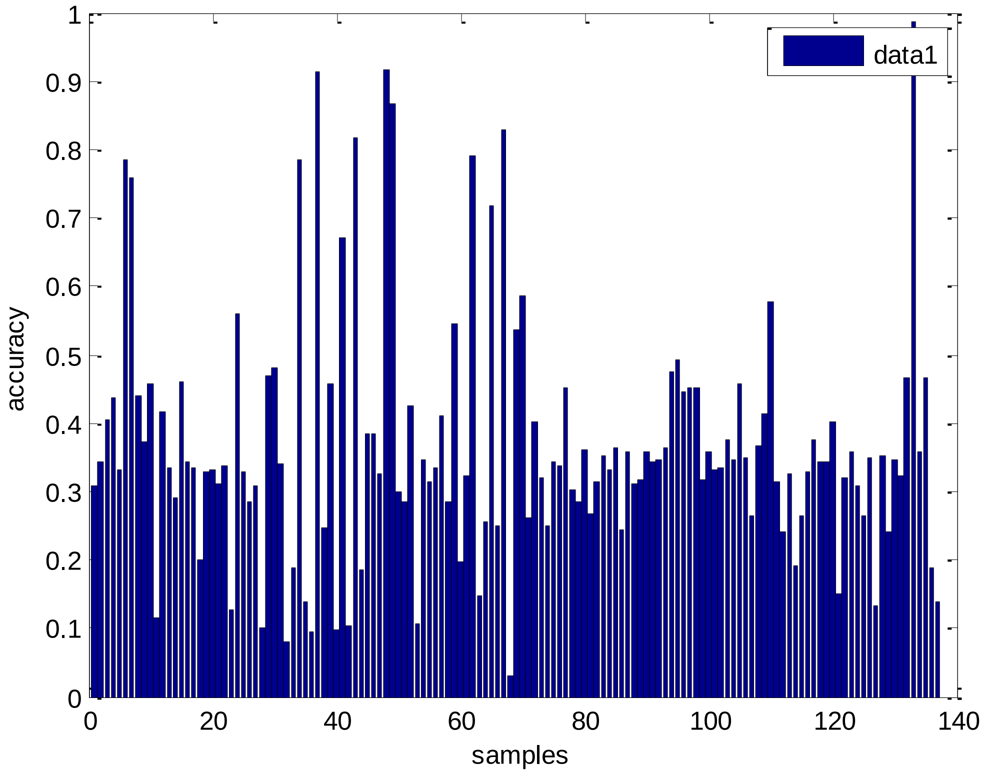


Figure 4: Overall (bar) B data of samples of educational and non-educational in the testing phase.

As can be shown in fig .(4) the overall bar in Neural network where the number of data B is 137 samples in the testing phase of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output and the number of hidden layer is 20.

APPENDIX 8: Overall (bar) BC data of samples of educational and non-educational in the training phase

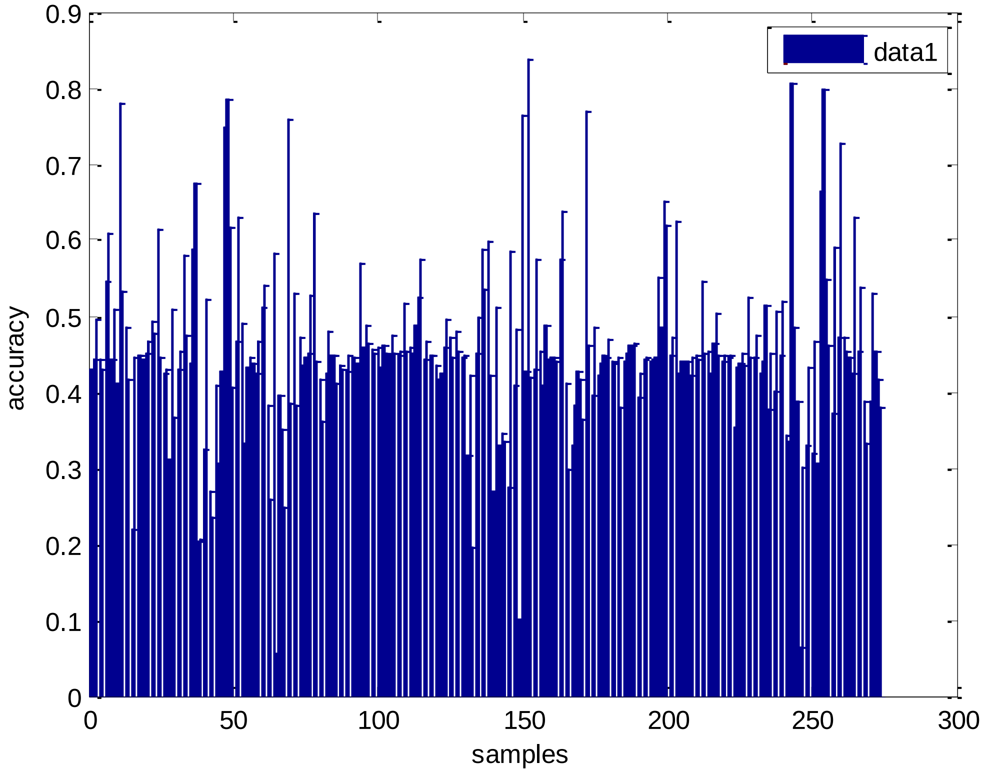


Figure 5: Overall (bar) BC data of samples of educational and non-educational in the training phase.

As can be shown in fig .(5) the overall bar in Neural network were the number of data BC is 274 samples in the training phase of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output and the number of hidden layer is 20.

APPENDIX 9: Overall (bar) A data of samples of educational and non-educational in the testing phase

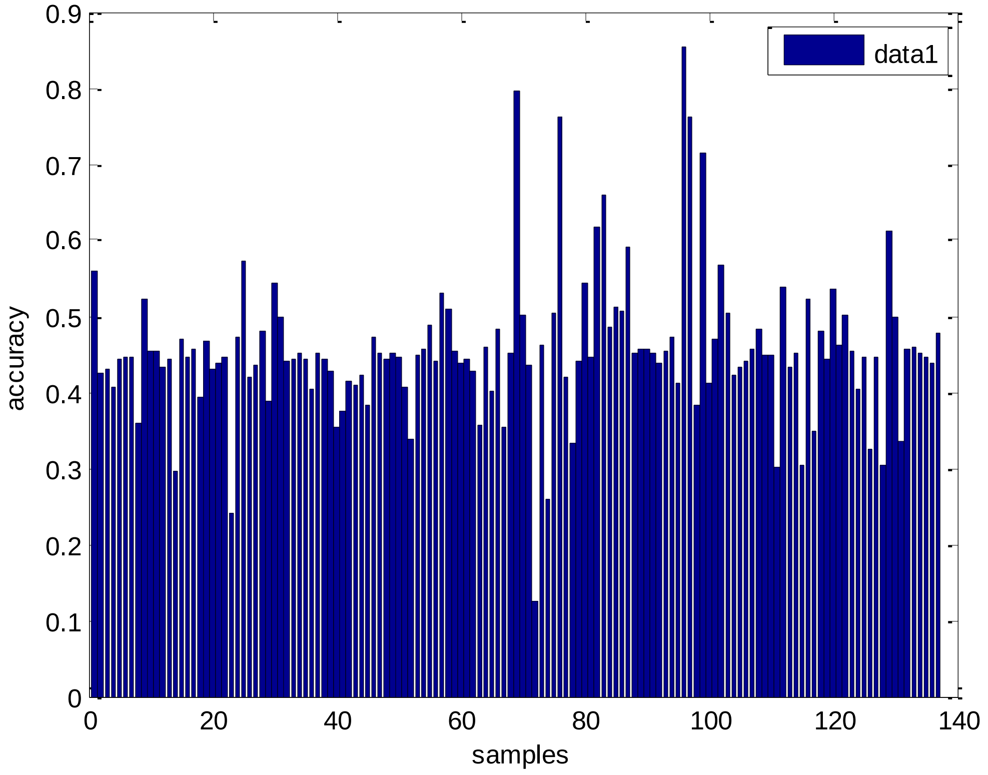


Figure 6: Overall (bar) a data of samples of educational and non-educational in the testing phase.

As can be shown in fig .(6) the overall bar in Neural network where the number of data A is 137 samples in the testing phase of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output and the number of hidden layer is 20.

APPENDIX 10: Overall (bar) AB data of samples of educational and non-educational in the training phase

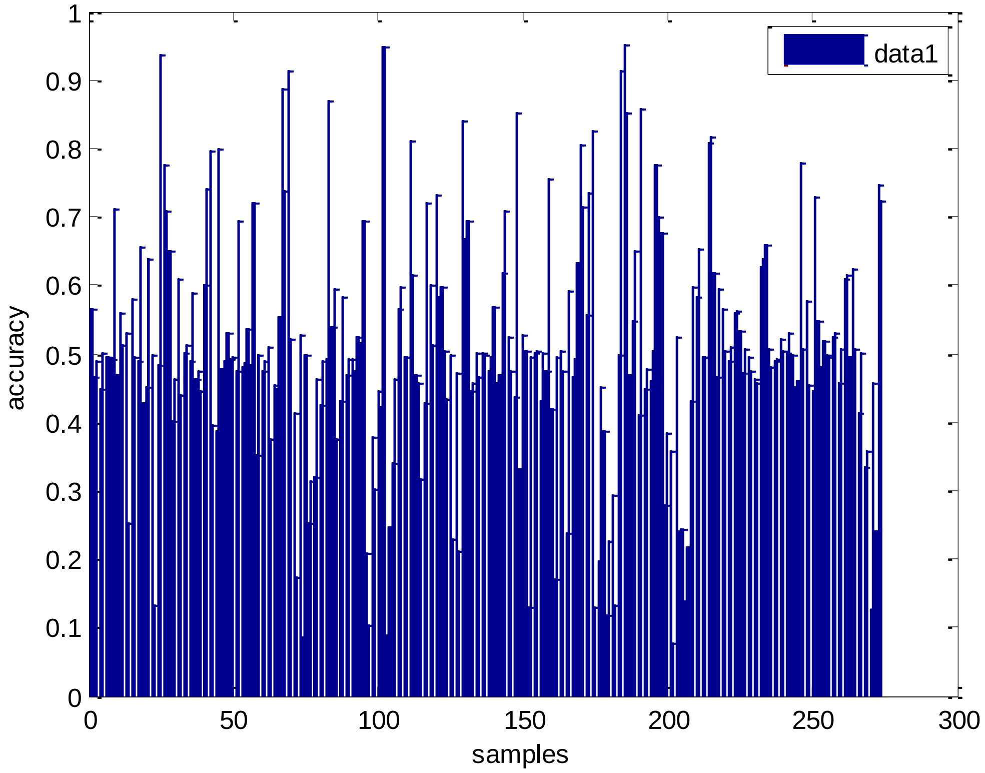


Figure 7: Overall (bar) AB data of samples of educational and non-educational in the training phase.

As can be shown in fig .(7) the overall bar in Neural network where the number of data AB is 274 samples in the training phase of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output and the number of hidden layer is 30.

APPENDIX 11: Overall (plot) C data of samples of educational and non-educational in the testing phase

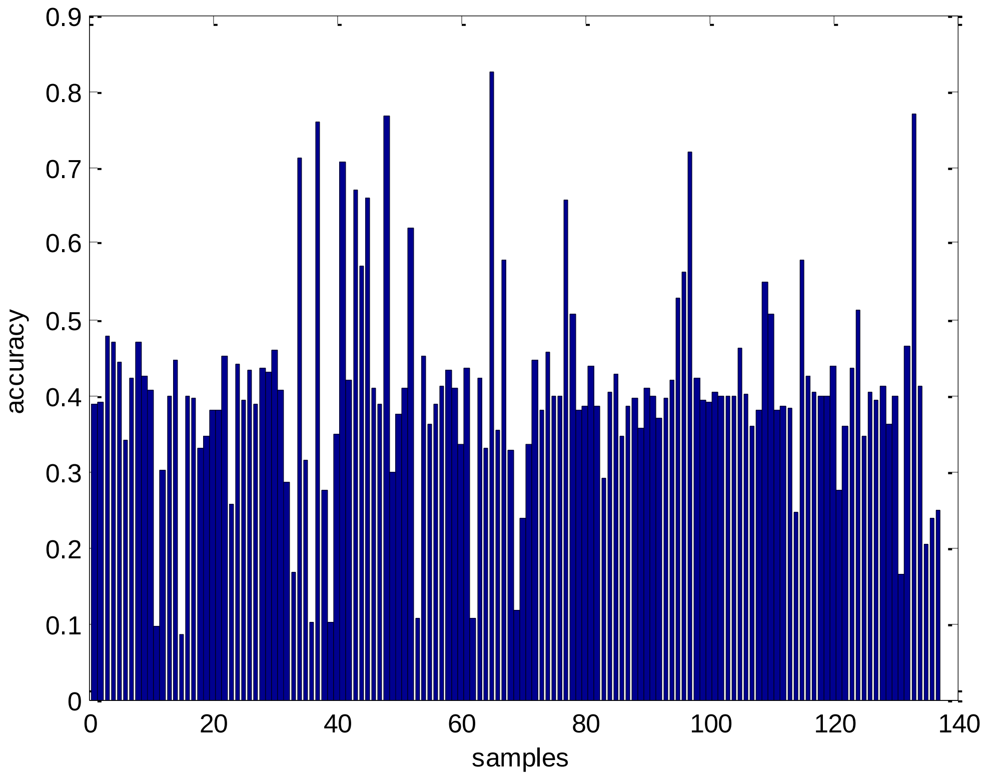


Figure 8: Overall (plot) C data of samples of educational and non-educational in the testing phase.

As can be shown in fig .(8) the overall plot in Neural network where the number of data C is 137 samples in the testing phase of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output and the number of hidden layer is 30.

APPENDIX 12: Overall (bar) AC data of samples of educational and non-educational in the training phase

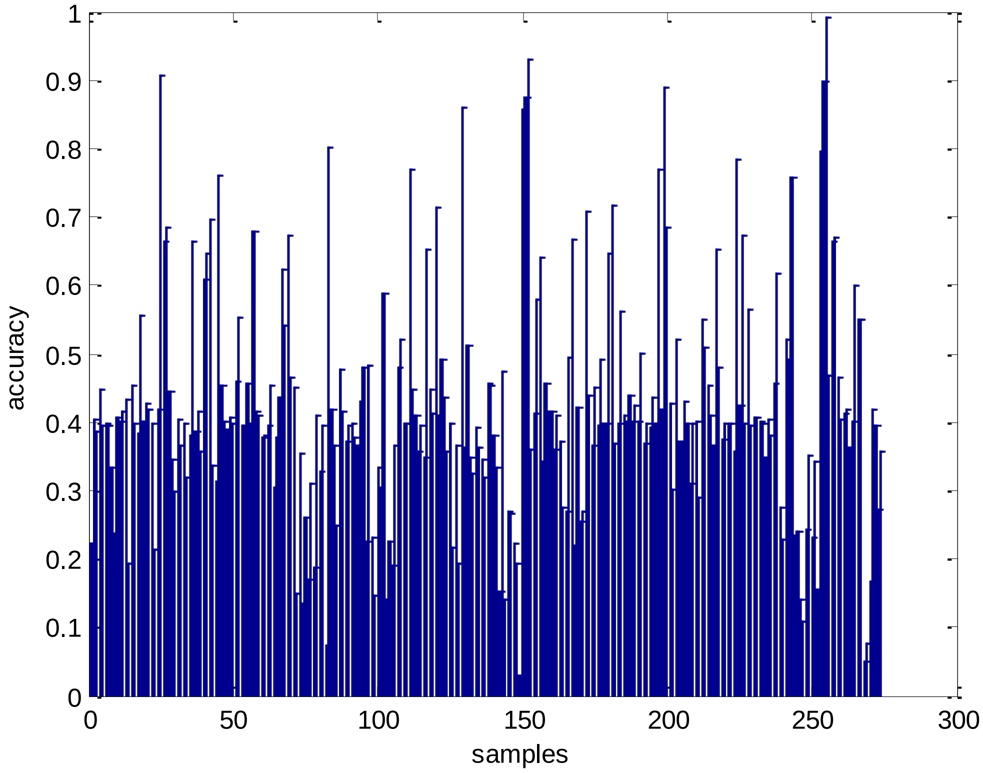


Figure 9: Overall (bar) AC data of samples of educational and non-educational in the training phase.

As can be shown in fig .(9) the overall bar in Neural network where the number of data AC is 274 samples in the training phase of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output and the number of hidden layer is 30.

APPENDIX 13: Overall (plot) B data of samples of educational and non-educational in the testing phase

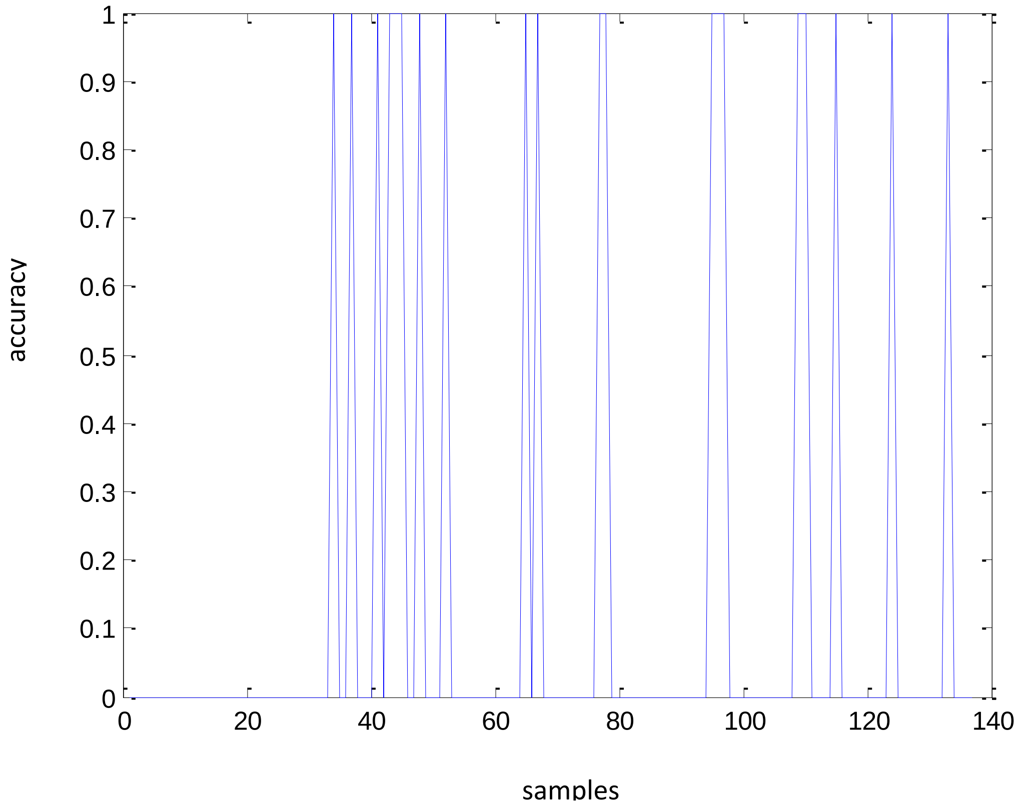


Figure 10: Overall (plot) B data of samples of educational and non-educational in the testing phase.

As can be shown in fig .(10) the overall plot in Neural network where the number of data B is 137 samples in the testing phase of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output and the number of hidden layer is 30.

APPENDIX 14: Overall (plot) BC data of samples of educational and non-educational in the training phase

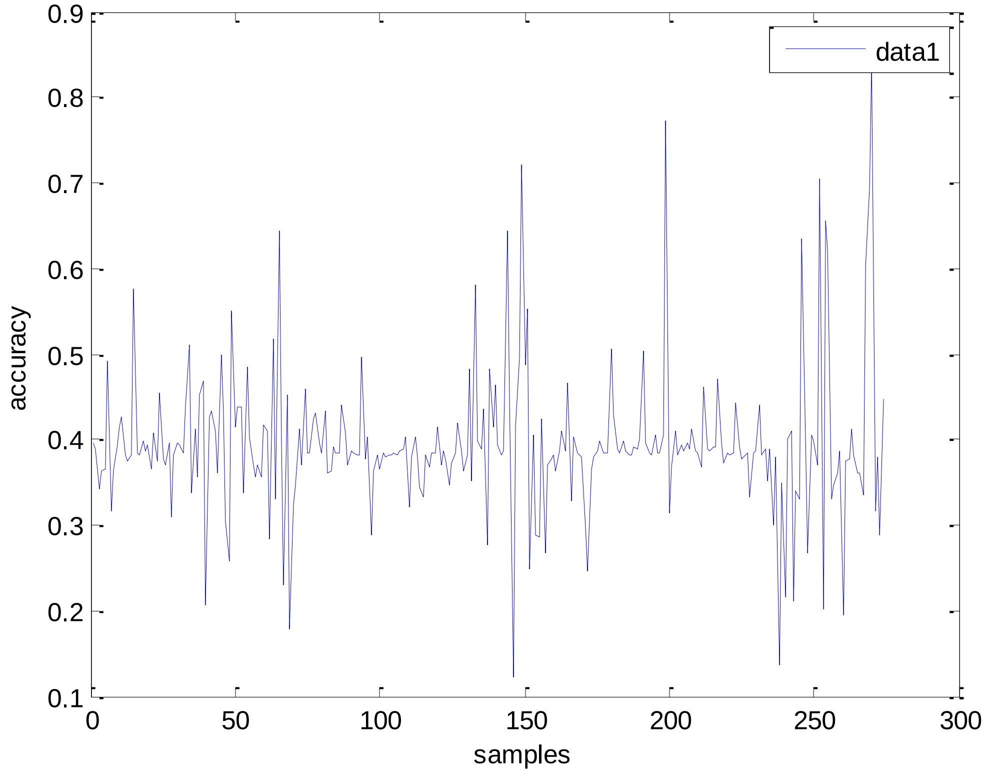


Figure 11: Overall (plot) BC data of samples of educational and non-educational in the training phase.

As can be shown in fig .(11) the overall plot in Neural network where the number of data BC is 274 samples in the training phase of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output and the number of hidden layer is 30.

APPENDIX 15: Overall a data of samples of educational and non-educational in the testing phase

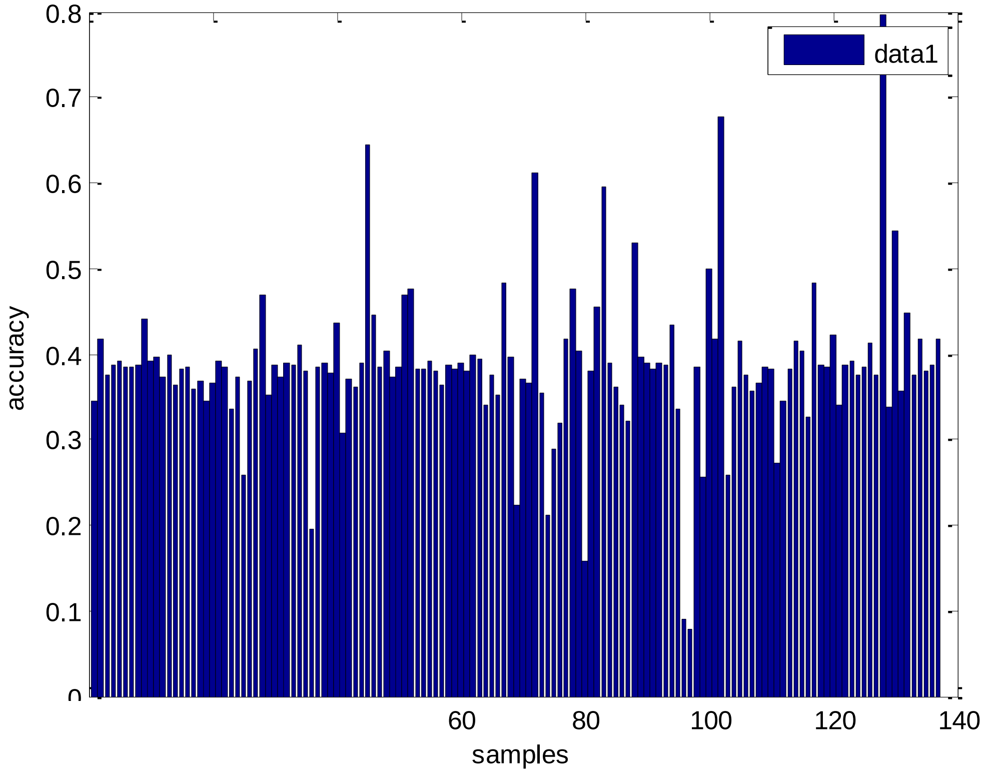


Figure 12: Overall a data of samples of educational and non-educational in the testing phase.

As can be shown in fig .(12) the overall bar in Neural network where the number of data A is 137 samples in the testing phase of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output and the number of hidden layer is 30.

APPENDIX 16: Overall AB data of samples of educational and non-educational in the training phase

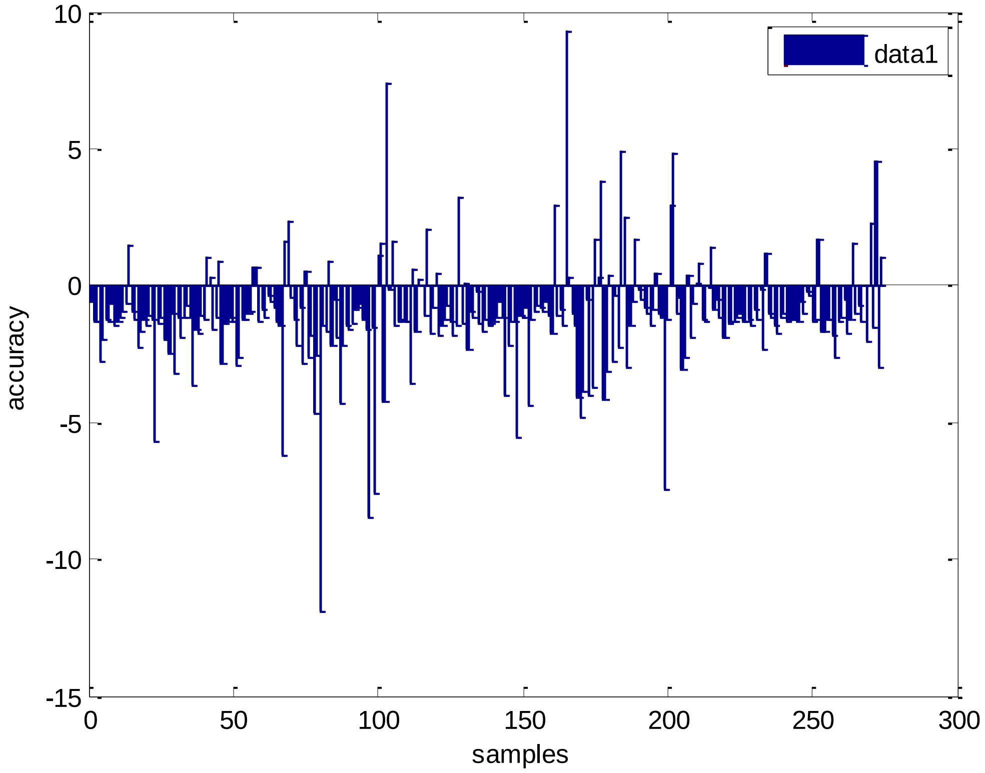


Figure 13: Overall AB data of samples of educational and non-educational in the training phase.

As can be shown in fig .(13) the overall bar in Neural network where the number of data AB is 247 samples in the training phase of educational and non-educational, the inputs layer are 450 and the outputs layer is 1 output, transfer function is PURELIN and the number of hidden layer is 30.

APPENDIX 17: Overall C data of samples of educational and non-educational in the testing phase

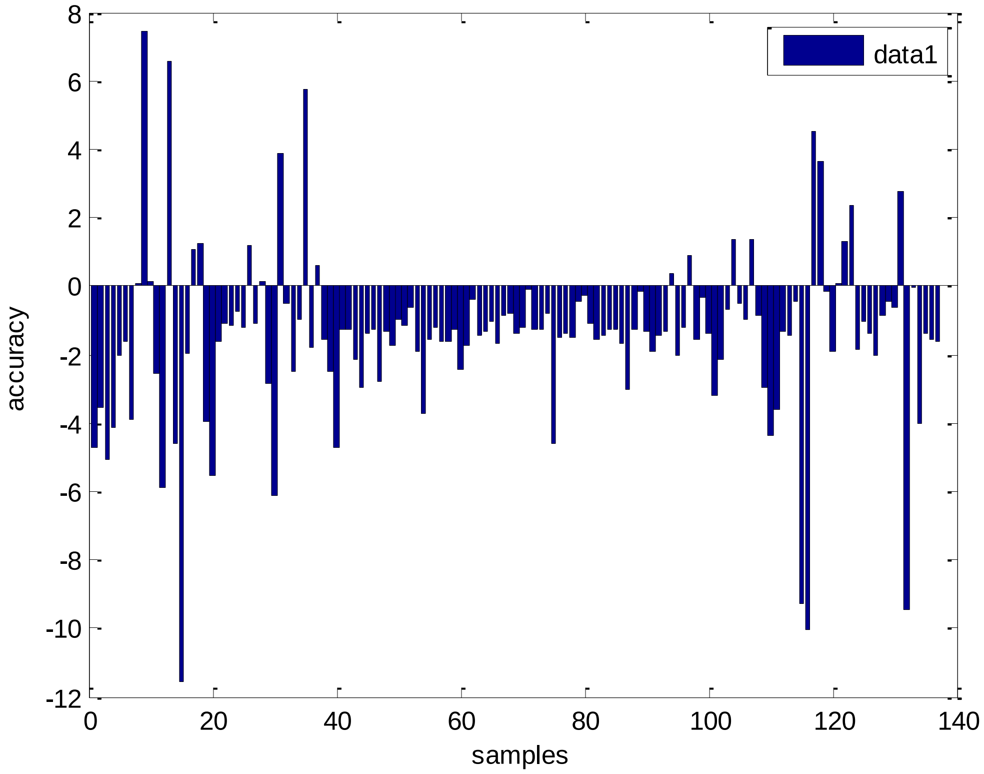


Figure 14: Overall C data of samples of educational and non-educational in the testing phase using transfer function PURLIN and hidden layer 30.

APPENDIX 18: AB train and hidden layer 10,transfer function TANSIG

```
     0.47746     0.2977   0.54454   0.59475   0.55012   0.53022     0.5353   0.66619   0.70644

 0.51998   0.57725   0.59592   0.58686   0.72728   0.60405    0.5369     0.556   0.52376
 0.64239   0.45823   0.64249   0.53621   0.14146   0.60977   0.93748   0.50008   0.35747
 0.52934   0.33591   0.41181   0.83133   0.46676   0.53263   0.55116   0.52966   0.23203
 0.43275   0.52666    0.4526   0.78648   0.68285   0.58123   0.41042   0.35278   0.91817
 0.63798   0.53376   0.59755     0.594   0.53516   0.50818   0.92861   0.52138   0.50806
 0.58083   0.52045   0.82983   0.87674   0.55385    0.4781   0.52966   0.60263    0.3095
 0.37691   0.52593   0.69145   0.88983   0.92842   0.90346   0.92309   0.67335    0.1783
 0.48046   0.05603   0.36644    0.1765   0.44256   0.32951    0.4624  0.077836   0.52478
 0.96058   0.95518   0.47059   0.57743   0.61368     0.872   0.80418   0.53778   0.51389
 0.54087    0.4781   0.56536   0.48296   0.49395   0.99936   0.15391   0.28267   0.27171
 0.42815   0.36193   0.19369  0.027941   0.29685   0.44101    0.5286   0.67796   0.61975
 0.53292   0.53613   0.76575   0.68144   0.46203   0.52895   0.53489   0.76291   0.21141
 0.50192   0.57109    0.7754   0.59571   0.63166   0.52825   0.48578   0.53342   0.17335
 0.56421   0.94811   0.91263   0.57669   0.65161   0.49052   0.46688    0.5951    0.5222
 0.45364   0.56472   0.45463   0.52195   0.60743   0.42554   0.52538   0.60733   0.88291
 0.49201   0.43204   0.42918   0.97152   0.71595   0.54662   0.53897  0.062272   0.53545
 0.53627   0.53786   0.56935   0.55887    0.5766   0.62734   0.86521   0.78863   0.50312
 0.40433   0.45463   0.32694   0.46032   0.43683   0.51396     0.551   0.79036   0.49257
 0.78521   0.67447   0.90756   0.10038   0.04235   0.45887   0.12032  0.055605   0.46669
 0.28245   0.21843   0.46874   0.87599   0.16158   0.93499   0.41766   0.66821   0.73337
 0.86505   0.95183   0.43848   0.51962   0.48345   0.40433   0.88264   0.81753   0.80229
 0.18967   0.20164   0.62188   0.43583   0.46932   0.14402   0.24993   0.95437    0.3003
 0.41734   0.73655   0.59154   0.73404   0.53715   0.52404   0.88495   0.93676   0.59047
 0.53108   0.62722   0.58826    0.5025   0.53306    0.5734   0.50491   0.74737   0.57143
 0.52552   0.53556    0.5363   0.52209   0.55544   0.87373    0.5215   0.59032   0.85437
 0.56558   0.46735   0.55839   0.57475   0.59938   0.55068   0.53432   0.46269   0.53772
 0.48547   0.51211   0.86913   0.59769    0.6511   0.47914   0.46251   0.70456   0.66028
 0.51039    0.5607   0.53352   0.53731   0.49539   0.69378   0.50642   0.53298   0.68294
 0.71031   0.53488    0.6126   0.56067   0.49078   0.53432   0.27291   0.46401   0.14461
 0.54164   0.93643   0.41765  0.62173.
```

APPENDIX 19: C Test and hidden layer 10, transfer function TANSIG

```
0.7258    0.6303    0.4432    0.5547    0.7088    0.6703    0.6920    0.5862    0.3729    0.6559
0.9943    0.9955    0.3852    0.5272    0.9785    0.4305    0.1304    0.4639    0.6109    0.1886
0.6715    0.5276    0.5886    0.4908    0.5205    0.7719    0.6832    0.4614    0.1239    0.9342
0.4932    0.6064    0.7351    0.1764    0.9819    0.7902    0.4231    0.5520    0.3934    0.5718
0.5355    0.5361    0.4500    0.2227    0.4705    0.5365    0.5464    0.5716    0.5417    0.5206
0.5239    0.4520    0.4858    0.4978    0.4929    0.5367    0.4780    0.4876    0.5363    0.4516
0.4412    0.9172    0.3609    0.5566    0.5858    0.8788    0.5377    0.5372    0.4383    0.5248
0.3852    0.5311    0.5437    0.4720    0.5729    0.6140    0.5677    0.7084    0.5346    0.7656
0.3226    0.4621    0.5042    0.5363    0.5386    0.4242    0.8180    0.5437 0.5584    0.5269
0.7959    0.5293    0.5259    0.5561    0.5259    0.5248    0.5563    0.4088    0.9044    0.5521
0.2889    0.6781    0.6783    0.2625    0.2040    0.8215    0.5430    0.3645    0.0952    0.6274
  0.6735    0.5468    0.0926    0.5230    0.9749    0.8123    0.8984    0.9909    0.5648
0.5624    0.4195    0.6794    0.8891    0.6239   0.5572   0.4938  0.5259    0.8697    0.5830
0.2941    0.0062    0.0815     0.3602    0.5485    0.4806    0.8232    0.4786.
```

APPENDIX 20: AC train and hidden layer 20, transfer function TANSIG

```
   0.2378    0.2165    0.3401    0.3916    0.4161    0.3431    0.3365    0.4789    0.5659
0.3861    0.2935    0.3611    0.3183    0.7036    0.4145    0.3436    0.3130    0.5089    0.4738
0.4300    0.6458    0.3444    0.4689    0.3304    0.7881    0.6523    0.6398    0.4539    0.4244
0.1792    0.3810    0.2831    0.3184    0.2983    0.3505    0.9017    0.2917    0.3500    0.3139
0.6219    0.6955    0.6992    0.3668    0.2323    0.0419    0.5609    0.3668    0.3269    0.4342
0.3435    0.5622    0.6548    0.3389    0.3394    0.3655  0.3325    0.7781    0.4791    0.3693
0.3034    0.3505    0.2699    0.4007    0.2043    0.3153    0.3904    0.6029    0.5879    0.5519
0.6213    0.5310    0.0874    0.2148    0.4015    0.1118    0.5420    0.302   0.2502    0.3911
0.0254    0.3426    0.0383    0.6140    0.4765    0.3222    0.1039    0.6512    0.6553    0.3253
0.3385    0.3614    0.3034    0.3158    0.4876    0.2106    0.0226    0.0614    0.1772    0.078
0.2721    0.2683    0.1710    0.0015    0.3657    0.0947    0.3390    0.4589    0.4654    0.3419
0.343    0.2094    0.2045    0.2998    0.2662    0.6108    0.4895    0.7025    0.5835    0.3753
0.5373    0.4027   0.5022    0.4158    0.3942    0.3457    0.4267    0.3501    0.8225    0.6037
0.5457    0.4460    0.2721    0.3147    0.3180    0.3673    0.3159    0.2637    0.3168    0.3816
0.6975    0.3883    0.1154    0.2913    0.3146    0.3725    0.0746    0.1091    0.9708    0.0351
0.8260    0.7893    0.9461    0.3338    0.3700  0.5721    0.7463    0.9051    0.4007    0.3263
0.3540    0.2410    0.3542    0.6135    0.4032    0.2874    0.8120    0.1371    0.0688    0.3679
0.1359    0.3316    0.0636    0.3515    0.3816    0.3316    0.3736  0.6431    0.3461    0.3433
0.6723    0.1419    0.2924    0.3433    0.5202    0.3472    0.3603    0.4031    0.3477    0.3180
0.3312    0.5785    0.3206    0.3439    0.3411    0.3643    0.3438    0.6786    0.3230    0.7810
0.5290    0.3679    0.2470    0.1902    0.3041    0.3334    0.4179    0.3395    0.2816    0.3396
0.3400    0.1830    0.2114    0.3478    0.4936    0.4037    0.3309    0.6071    0.3451    0.3109
0.2938    0.3443    0.3442    0.4164    0.7989    0.3560    0.5905    0.3414    0.7043    0.3447
0.3496    0.3711    0.3466    0.3395    0.2470    0.1489    0.3287    0.3033    0.6835    0.6534
0.7389    0.2635    0.5533   0.8738    0.1364    0.0465    0.0860    0.1397    0.3109    0.2897
0.0785    0.2820    0.0592    0.8026    0.9432    0.9995    0.4383    0.4900    0.4799    0.4640
0.6117    0.3054    0.3768    0.3068    0.3466    0.6256    0.5800    0.6372    0.1551    0.1318
0.0308    0.3054    0.3602    0.1547    0.4549.
```

APPENDIX 21: B test and hidden layer 20, transfer function TANSIG

```
   0.3085    0.3421    0.4054    0.4360    0.3312    0.7860    0.7581    0.4413    0.3735
0.4576    0.1142    0.4154    0.3338    0.2912    0.4614    0.3427    0.3359    0.1994    0.3304
0.3315    0.3128    0.3370    0.1263    0.5597    0.3280    0.2850    0.3085    0.1015    0.4690
0.4800    0.3409    0.0799    0.1888    0.7858    0.1389    0.0951    0.9126    0.2475    0.4573
0.0971    0.6709    0.1049    0.8166    0.1866  0.3838    0.3845    0.3273    0.9153    0.8657
0.2985    0.2849    0.4255    0.1076    0.3459    0.3150  0.3357    0.4111    0.2850    0.5464
0.1972    0.3245    0.7905    0.1490    0.2564    0.7190    0.2490    0.8276    0.0307    0.5365
0.5853    0.2614    0.4018    0.3203    0.2494    0.3435    0.3367    0.4512    0.3014    0.2851
0.3616    0.2688    0.3139    0.3534    0.3331    0.3647    0.2444    0.3595    0.3127    0.3173
0.3582    0.3442    0.3477    0.3631    0.4747    0.4925    0.4454    0.4520    0.4529    0.3162
0.3578    0.3325    0.3340    0.3750    0.3457    0.4575    0.3484    0.2642    0.3664    0.4139
0.577   0.3148    0.2400    0.3263    0.1914    0.2658    0.3301    0.3749    0.3432    0.3437
0.4022    0.1499   0.3204    0.3595    0.3076    0.2632    0.3487    0.1331    0.3510    0.2410
0.3457    0.3232    0.4667 0.9870    0.3585    0.4657    0.1878    0.1389.
```

APPENDIX 22: BC train and hidden layer 30,transfer function TANSIG

```
   0.3966    0.3887    0.3421    0.3638    0.3661    0.4913    0.3164    0.3637    0.3926
0.4157    0.4252   0.3818    0.3741    0.3817    0.5752    0.3842    0.3822    0.3971    0.3857
0.3926    0.3658    0.4083   0.3751    0.4553    0.3768    0.3712    0.3966    0.3104    0.3808
0.3963    0.3934    0.3846    0.4378    0.5114    0.3370    0.4127    0.3569    0.4520    0.4678
0.2074    0.4254    0.4335    0.4091    0.3612    0.4985    0.4327    0.3044    0.2583    0.5492
0.4153    0.4382    0.4376    0.3366    0.4854    0.4024   0.3694    0.3571    0.3712    0.3560
0.4164    0.4094    0.2835    0.5187    0.3309    0.6441    0.4000    0.2307    0.4520    0.1779
0.3240    0.3459    0.4131    0.3711    0.4585    0.3848    0.3837    0.4229   0.4303    0.3960
0.3835    0.4324    0.3610    0.3643    0.3913    0.3839    0.3842    0.4397    0.4069   0.3691
0.3867    0.3852    0.3821    0.3814    0.4954    0.3783    0.4035    0.2883    0.3623    0.3823
0.3653    0.3850    0.3788    0.3808    0.3829    0.3835    0.3821    0.3869    0.3887    0.4023
0.3209   0.3803    0.4037    0.3791    0.3442    0.3321    0.3818    0.3673    0.3850    0.3840
0.4156    0.3695    0.3856    0.3771    0.3464    0.3733    0.3834    0.4187    0.3877    0.3631
0.3829    0.4819    0.3523   0.5799    0.3972    0.3898    0.4367    0.2770    0.4832    0.4157
0.4631    0.3943    0.3810    0.3870   0.6431    0.4442    0.1226    0.4139    0.4954    0.7219
0.4867    0.5528    0.2489    0.4061    0.2883    0.2854    0.4238    0.2676    0.3713    0.3775
0.3814    0.3631    0.3874    0.4107    0.3863    0.4663    0.3286    0.4035    0.3834    0.3810
0.3805    0.2977    0.2456    0.3665    0.3797    0.3870    0.3986   0.3831    0.3839    0.3831
0.5066    0.4276    0.3883    0.3837    0.3993    0.3876    0.3822    0.3815   0.3914    0.3880
0.3998    0.5026    0.3953    0.3839    0.3830    0.4047    0.3839    0.3847    0.4049    0.7720
0.3146    0.3687    0.4093    0.3808    0.3946    0.3877    0.3960    0.3878    0.4112    0.3862
0.3838    0.3681    0.4624    0.3887    0.3858    0.3918    0.3907    0.4699    0.3969    0.3715
0.3837    0.3823    0.3845    0.4417    0.3918    0.3776    0.3789    0.3841    0.3330    0.3831
0.3863    0.4393    0.3812    0.3878    0.3507    0.3894    0.2996    0.3802    0.1352    0.3500
0.2164    0.4012    0.4104    0.2113    0.3405    0.3315    0.6352    0.3978    0.2665    0.4059
0.3976    0.3707    0.7050    0.2020    0.6561    0.6212    0.3308    0.3465    0.3616    0.3869
0.1953    0.3753    0.3782    0.4118    0.3812   0.3597    0.3619    0.3347    0.6043    0.6948
0.8394    0.3155    0.3794    0.2881    0.4464.
```

APPENDIX 23: A test and hidden layer 30,transfer function TANSIG

```
0.3454   0.4184   0.3764   0.3861   0.3926   0.3845   0.3847   0.3868   0.4411   0.3910
0.3957  0.3741   0.3987   0.3648   0.3813   0.3841   0.3590   0.3688   0.3444   0.3657
0.3913   0.3839  0.3349   0.3723   0.2595   0.3693   0.4050   0.4683   0.3529   0.3881
0.3721   0.3889   0.3862   0.4117   0.3802   0.1949   0.3851   0.3892   0.3775   0.4370
0.3074   0.3711   0.3613   0.3906  0.6452   0.4462   0.3853   0.4027   0.3735   0.3841
0.4680   0.4757   0.3828   0.3834   0.3912   0.3792   0.3627   0.3882   0.3835   0.3902
0.3802   0.3979   0.3953   0.3403   0.3766   0.3515   0.4838   0.3973   0.2223   0.3699
0.3659   0.6125   0.3555   0.2117   0.2888   0.3201   0.4173  0.4771   0.4032   0.1570
0.3805   0.4554   0.5952   0.3887   0.3615   0.3393   0.3205   0.5291   0.3958   0.3891
0.3815   0.3902   0.3874   0.4340   0.3362   0.0894   0.0783   0.3843   0.2562  0.4995
0.4172   0.6776   0.2588   0.3605   0.4143   0.3765   0.3565   0.3664   0.3845   0.3832
0.2722   0.3462   0.3821   0.4162   0.4045   0.3254   0.4824   0.3865   0.3842   0.4231
0.3413  0.3865   0.3923   0.3747   0.3836   0.4139   0.3746   0.7968   0.3375   0.5436
0.3565   0.4473  0.3760   0.4179   0.3805   0.3865   0.4166.
```

APPENDIX 24: Vedio Files

materials	videos name	e/n
exam	cpa exam review	educational
action	omg The Worst Fight Scene EVER	non-educational
economic	REG Installment Sales for Taxes	educational
cartoon	cartoon	non-educational
cartoon	cartoon jebs jobs	non-educational
mathematical	functions	educational
biological	Stanford's Sapolsky On Depression in U.S	educational
cartoon	Tom and Jerry Cartoon - Baby Butch	non-educational
cartoon	Mickey & Minnie = Hansel & Gretel	non-educational
computer science	Gregory Chaitin Lecture Carnegie-Mellon University 2000 Pt 2	educational
dance	RIO_ BRAZIL. 30 DANCERS	non-educational
computer networks	Emergence of Networks & Reference Models	educational
film	film new moon	non-educational
medicine	the human gname and individualized medicine	educational
song	song(Elissa Hussein Al Jasmi Fadel)	non-educational
art	Appreciating art what is contemporary art?	educational

Chemistry	Chemistry 1A - Lecture 1	educational
economics	introduction to economics	educational
geography	Free Video Lectures on Geography for CBSE Class X	educational
English	English Professor James Warren	educational
cartoon	cartoon fall in-hare	non-educational
action	action movie	non-educational
cartoon	Pink Panther Cartoon	non-educational
cartoon	cartoon dementia	non-educational
		educational
English	Study English IELTS Preparation	educational
programming	programming abstractions	educational
physical	water and wastewater engineering	educational
history	Early Women's Movement	educational
dance	ZUMBA DANCE CRAZE IN CEBU PHILIPPINES	non-educational
Mechanical Engineering	Mini-lecture_ The science of micro bubbles (UCL)	educational
religion	Sam Harris lecture on religious faith Part 01	educational
magic	The Magic of Consciousness	educational
mathematical	contemporary mathematical	educational
mathematical	Differential Equations1	educational
mathematical	Approximating functions and derivatives	educational