ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

ADDIS ABABA INSTITUTE OF TECHNOLOGY (AAiT)

ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT

Intelligent System Based Automatic Prediction of Drought Using Satellite Images

By

Taye Tolu Mekonnen

A thesis submitted to the School of Graduate Studies of Addis Ababa Institute of Technology in partial fulfillment of the requirements for the degree of

Master of Science in Computer Engineering

Addis Ababa University

August 2011

ABSTRACT

Early days of technology employed traditional methods like rainfall distribution, duration of sunshine, and the related meteorological data to forecast a forthcoming event regarding drought. Recently, satellite remote sensing has been considered as an appropriate tool for deriving information in spatial and temporal domains by providing multi-spectral reflectance data at regular intervals. Satellites from centers like National Oceanic and Atmospheric Administration (NOAA) and Meteosat capture the spectral reflectance from green plant and provide this frequently to monitor green vegetation conditions on the ground. Using the red and infrared band reflectances a vegetation index called Normalized Difference Vegetation Index (NDVI) was derived; which is vital to access the evolution of drought as well as predict crop yield.

The aims of this study are to analyze series of deviation of NDVI images, extract virtual drought objects from the series, and investigate for drought patterns from historical image for growing season. Subsequent to this, appropriate prediction model of these patterns was developed for early measures while within the same season. And it was applied on the subset of image data with reported drought occurrences in Ethiopia.

In this study, the virtual drought objects extracted from images over the growing season (June - September) were found to exhibit a given pattern for the historical drought years. After producing the descriptors of drought objects in the series using principal component analysis, combined and separate artificial neural network (ANN) models were used to predict these patterns. In the combined approach all the descriptors of the object in the next time step were predicted all at the same time while in the separate approach the prediction was made one by one. Accordingly, the models designed to forecast the future state of drought object using these two approaches yielded promising results. Especially, the three and four time lag combined ANN prediction model produced an overall RMSE of 20.80 and 16.43, respectively, which was a better result compared to a 36.92 RMSE of separate ANN approach. It is understood that this work will give new views for ways in drought prediction for early warning and crop condition monitoring at near real-time.

Keywords: Drought prediction, NDVI images, Virtual drought object, Intelligent system

ACKNOWLEDGMENT

In the first place, I would like to record my gratitude to my advisor Dr. Kumudha Raimond for her supervision, advice, and guidance from the very early stage of this research as well as giving me extraordinary experiences throughout the work. Her involvement with her originality has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

I gratefully acknowledge Ato Getachew Berhan for his ideas, assistance and crucial contribution, which made him an important factor for the accomplishment of this thesis. Above all and the most needed, he provided me unwavering encouragement and support in various ways.

Words fail me to express my appreciation to W/t Betelhem Abi whose dedication, love and persistent confidence in me, has taken the load off my shoulder. I owe her for being unselfishly let her intelligence, passions, and ambitions collide with mine.

Finally, I would like to thank everybody who was important to the successful realization of the thesis, as well as expressing my apology that I could not mention personally one by one. Especially, the cooperation of staff members of National Meteorological Agency, and Central Statistical Agency of Ethiopia cannot be left unmentioned.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Network
AOI	Area of Interest
ARIMA	Autoregressive Integrated Moving Average
AVHRR	Advanced Very High Resolution Radiometer
DPPC	Disaster Prevention and Preparedness Commission
FEWSNET	Famine Early Warning Systems Network
GIS	Geographic Information System
ILWIS	Integrated Land and Water Information System
LDA	Linear Discriminant Analysis
LLE	Locally Linear Embedding
MLP	Multilayer Perceptron
MSG	Meteosat Second Generation
NDMC	National Drought Mitigation Center
NDVI	Normalized Difference Vegetation Index
NIR	Near Infra Red
NOAA	National Oceanic and Atmospheric Administration
PC	Principal Component
PCA	Principal Component Analysis
RMSE	Root Mean Square Error
SARIMA	Seasonal Autoregressive Integrated Moving Average

CHAPTER 1

INTRODUCTION

1.1 Background

Attributed to the unusual variation in climate, drought has far been identified as a normal, recurrent feature of climate [1]. It is an ecological incident which is manifested in the reduction of farming production, causing social crisis and transforming into political disputes. This protracted period of deficient precipitation causes extensive damage to crops, and results in loss of yield. As responsive measure, drought assessment and monitoring missions has been conducted.

However, most of the efforts so far have been based on conventional methods that rely on the availability of meteorological point data like amount of rainfall and the related indices. These methods require a tiresome data collection, and provide a scarce information as a given data represents a wide geographical region. This makes the approach unreliable to take risk measures as precaution. Therefore, producing a more reliable and an all round information for concerned bodies is very important.

Recently, data availability driven by technological advancement from remote sensing is engrossing to overcome the limitations of the traditional drought indices. Remote sensing data or data from satellite sensors can provide continuous datasets that can be used to detect the onset of a drought, its duration and magnitude [2]. These sensors capture spectral bands reflected from the surface of the earth and provide information for monitoring green vegetation conditions on the ground: dense, scarce, or depleted.

Through thorough scientific investigations, a number of researchers have tried to derive information from satellite images to accompany with other ecosystem's data for better and more accurate prediction of drought [3]. The image-based analysis provides a more profound information, especially, for areas where ground monitoring using meteorological data is scarce. This approach requires a good knowledge of image processing for feature extraction.

The remote sensing satellites provide data at specific intervals (e.g., Meteosat Second Generation (MSG) data for every 15 minutes). However, little effort had been exerted to the proper drought feature extraction and interpretation from remote sensing data [55]. Thus, the

available information is bulky, and difficult for decision makers to analyze [4]. In addition, even though drought has its own state and behavior, there had been little past efforts done to detect drought by its own properties as spatial object [1].

In remote sensing images, as long as pixel sizes remained typically coarser than, or at the best, similar in size to the objects of interest, emphasis was placed on per-pixel analysis, or even sub-pixel analysis for this conversion, but with increasing spatial resolutions alternative paths have been followed, aimed at deriving objects that are made up of several pixels [6]. Pixels of similar spectral reflectance and closer spatial locations are grouped to define object of interest. These objects can represent either physical features on earth, such as roads, parcels, water bodies or virtual geographical objects like drought. The virtual objects are non-physical features on the ground but they are measurements having geographic information [7]. The objects in this case are defined based on some attribute of physical features.

Temporal and spatial patterns of drought vary from one time to another and an area to another [8, 9]. For a specific area, if relationships between drought object characteristics are identified with a certain time lag, taking appropriate action based on the associations may reduce the impacts of future droughts. To monitor drought and help in drought decision making, this study has attempted to identify relationships between consecutive drought objects for a given spatial location using time series data-mining and prediction algorithms. This has the potential to assist in proactive drought decision making including mitigation and drought risk management. For this purpose, a time series data analysis has been used to predict the parameters of the drought for a given spatial location to help in drought monitoring.

1.2 Statement of the Problem

Despite the growing number of freely existing satellite images for environmental monitoring and early precaution of hazards like drought, most countries have been constrained to using the meteorological point data, like amount rainfall. This point data is insufficient, cumbersome to collect, and requires large number of human intervention. In addition, the collection of data is performed at wide geographical coverage (E.g. Woreda level in Ethiopia), which makes it much generalized irrespective of the likely variation in the vegetation condition with in the selected region itself.

In addition, the image mining algorithms and approaches to be used for extraction of knowledge from satellite images need to incorporate the likely temporal-variation of the characteristics of vegetation indices that leads to drought, and hence time series analysis of the images. Therefore, it needs a good image processing technique which is designed in such a way that the model will make predictions for the onset of drought from a series of image data. That is, given the feature of drought object in an image for successive dekadals(a ten day intervals in this work) of three or four, its feature for the next dekadal will be predicted so that early measures could be taken even while within the growing season itself.

In general, the algorithms developed thus far have done a little to integrate with machine intelligence so that once the data is available for a given part of the growing season of a year; it can automatically predict the nature of the upcoming event [1, 3, 10, 11]. Even those which tried to use the fuzzy system to amalgamate intelligence to the mining has selected membership functions just on the basis of personal judgment than scientific knowledge, thus needs further study [11].

1.3 Objectives

The main objective of this thesis work was to develop an intelligent system based image mining technique for predicting drought from satellite images. Thus, the specific objectives intended to be addressed are:

i. To analyze satellite images to determine how drought can be defined
ii. To select appropriate image processing technique to process and extract features of drought objects
iii. To develop an intelligent system for identifying, and predicting drought in advance for decision makers.
iv. Simulation and performance evaluation of the developed system.

1.4 Methodology

The research plan was performed in four major phases: literature review, image data collection, image analysis and feature extraction, and drought modeling and prediction.

i. Literature Review**:** A literature survey was conducted on the area of image processing for every stage of the system. Available books, journals, case studies, previous research

works & guidelines were surveyed in order to have a clear understanding of the subject matter.

ii. Data Collection**:** The necessary image repository and historical records of drought occurrences were collected from National Meteorological Agency, Ethiopia Disaster Prevention and Preparedness Commission (DPPC) and the website of NOAA satellite database (since 1981) & Meteosat (since 2005)

iii. Image analysis and feature extraction**:** This phase encompasses three typical steps. First, the images were analyzed to produce deviation maps of the grasped images. Then, the selection and implementation of suitable image denoising and object localization was performed. Finally, the features of consecutive drought objects for a given spatial location were extracted.

iv. Prediction and characterization of drought: This final phase of the research work was targeted at characterizing of drought objects on the spatio-temporal distribution of the geographical map.

The features collected in step three were given to the prediction model to predict the status of the specified spatial location with regard to drought. In addition, the classification of the predicted features was performed to characterize it depending on its severity.

Generally, the core thesis work is made up of data acquisition, noise removal, localization of drought objects, reducing the dimension of each and every object in the series, and finally, the prediction of the next drought object feature in the series.

1.5 Scope of the Thesis Work

From its superficial view, this work is bounded at prediction of drought depending on the state of the plants within the cultivation season so that early measures will be taken within the same season. It assumes that the growth of crops is already in progress when their health begins to be affected in such a way that they cannot harvest the sunlight for photosynthesis.

On top of that, this work is delimited at predicting the future state of a region whose states in terms of intensity of group of pixels is observed to have a difference compared to the regions around. Therefore, it is not valid if all the regions under investigation exhibit the same intensity value throughout.

Furthermore, for the prediction model to make sense, it is required that the evolution of drought over the growing season be of a particular pattern. With regard to drought in this thesis, the patterns used are from the reportedly drought occurrences in the past. Therefore, drought prediction method, under this thesis, can only be applied to series of the particular study and the related with little deviation tolerance.

1.6 Contribution

So far, a number of researches have been conducted to extract knowledge from satellite images for early warning on the onset of drought. The focus, though, are the classification of drought from individual images and the prediction on the likely occurrence of famine during the production season, given the characteristics of intensity of image of a given region during the growing (cultivation) season. In addition, the classification itself is performed at pixel level, considering each pixel as an object. But it is likely that neighboring ground areas could as well be suffering from the same problem. To cover these areas and treat like one spatial virtual object a number of neighboring pixels with more or less similar features have to be grouped. And prediction will be performed at object level. Thus, after a thorough literature survey, this work is the first to consider a series of images for prediction, to use group of pixels for defining drought; and above all, to try to predict the state of drought within the growing season itself so that a very early precaution and, if possible, probable amendments will be performed sooner than later. Lastly, it incorporates the nonlinearity in nature using the ANN employed here to make use of the inherent natural behavior of fluctuation of object characteristics in the image over time.

1.7 Organization of the Thesis

The thesis report is organized in the following manner: In chapter two a review of literature for each and every step of the thesis work is presented. In the subsequent chapter, the discussion on the preparation of an image for prediction from preprocessing to feature selection is presented. The introduction to neural networks is provided in chapter four. An in depth discussion of the design and the logical flow of the methods implemented in this work is described in chapter five. Chapter six analyzes the results and discusses the performance of the overall system, while chapter seven concludes and gives recommendations for future work.

CHAPTER 2

LITERATURE REVIEW

In this chapter literature survey of the detailed processes employed in the thesis work is presented. It includes the drought basics and its definitions in terms of different parameters, and the image processing technique involved in time series prediction of drought from the vegetation index-based satellite images.

2.1 Drought Understanding

Even though drought can be classified as one of the natural hazards, there is no universal definition for it. Its definitions are region specific, reflecting differences in climatic characteristics as well as incorporating different physical, biological and socioeconomic variables, and it is usually difficult to transfer definitions derived for one region to another [12]. The impact of drought is less obvious and spreads over large geographical area than those of other natural hazards like earthquake, volcanic eruption, etc [13]. For instance, an agricultural drought's impact can accurately be assessed when crop are harvested, which is few months after symptoms began to appear [14]. These characteristics of drought have increased difficulties of accurate, reliable and timely estimates of severity and impacts of drought for early preparedness plans [15].

Drought is generally classified into four different classes: meteorological, agricultural, hydrological and socioeconomic droughts [15]. Meteorological drought is defined usually on the basis of the degree of dryness (in comparison to some "normal" or average amount) and the duration of the dry period. Definitions of meteorological drought must be considered as region specific since the atmospheric conditions that result in deficiencies of precipitation are highly variable from region to region. Agricultural drought links various characteristics of meteorological (or hydrological) drought to agricultural impacts, focusing on precipitation shortages, differences between actual and potential evapo-transpiration, soil water deficits, reduced ground water or reservoir levels, and so forth. Plant water demand depends on prevailing weather conditions, biological characteristics of the specific plant, its stage of growth, and the physical and biological properties of the soil. Hydrological drought is associated with the effects of periods of precipitation (including snowfall) shortfalls on surface or subsurface water supply (i.e., stream flow, reservoir and lake levels, ground water).

Socioeconomic definitions of drought, on the other hand, associate the supply and demand of some economic good with elements of meteorological, hydrological, and agricultural drought. It differs from the aforementioned types of drought because its occurrence depends on the time and space processes of supply and demand to identify or classify droughts.

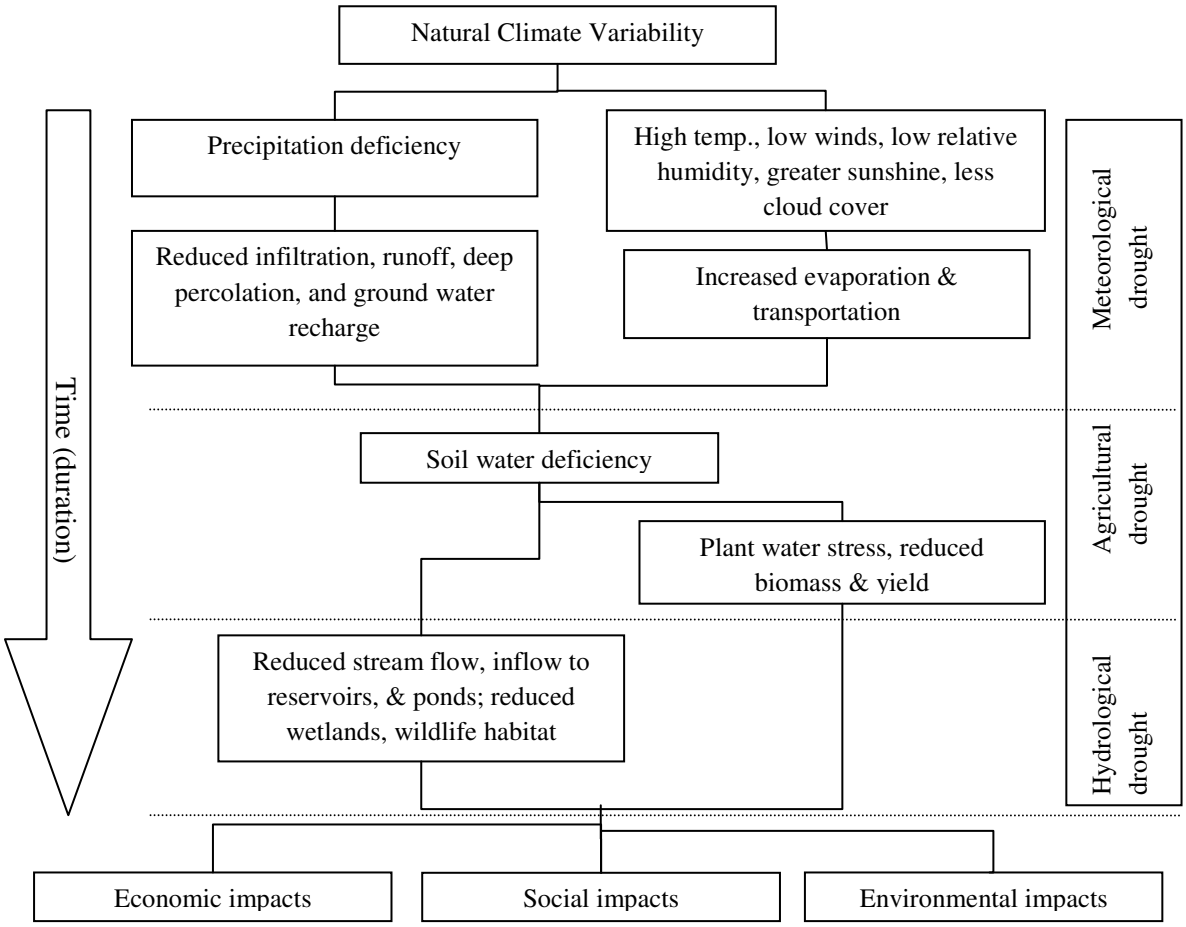


Fig 2.1 Sequence of Impacts of Different Classes of Drought [53]

2.1.1 Drought Indices

Drought indices are continuous functions of rainfall /or temperature, river discharge or other measurable variables. Based on the type of drought, several indices have been developed for the quantification of drought. For example, meteorological data based indices include Palmer Drought Severity Index (PDSI), Crop Moisture Index (CMI) and Standardized Precipitation Index (SPI). Rainfalls, temperature, evaporation, vegetation health, soil moisture, stream flow, and so on are the parameters that are used in drought analysis for scientific evaluation of drought situations [16].

Conventional methods of drought monitoring suffer from limitations with regard to timeliness, objectivity, unreliability, and inadequacy, but satellite sensors provide spatial information on vegetation stress caused by drought conditions [12]. Hence, a comprehensive study on drought can be conducted using satellite sensors.

2.1.2 Remote Sensing-based Drought Indices

The development of earth observation satellites from 1980 onwards equipped with sensors mainly in the optical domain opened a new road for drought monitoring and detection. This technology allowed for the derivation of truly spatial information at global or regional coverage at a consistent method and a high repetition rate. Numerous indices were developed to describe the state of the land surface, mainly of vegetation, with the potential to detect and monitor anomalies such as drought [17]. A recent review of remote sensing based drought monitoring was given by Bayarjargal et al. [18].

The commonly used remote sensing-based drought indices are the Normalized Difference Vegetation Index (NDVI), first applied for drought monitoring by Tucker and Choudhary [19] and the Standard Vegetation Index (SVI) [20]. As satellites revolve the earth continuously, they collect ground information and provide this information in regular intervals covering a wide area of the ground.

When drought exists, due to reduction of precipitation, the capacity to carry out photosyntetic process on the part of vegetation is notably reduced [3]. A healthy green vegetation condition is characterized by a maximum radiation absorption in the red and large reflection in the neighboring near-infrared region. In addition, for unhealthy vegetation condition, the reflectance in red region increases while in near infrared region decreases. Combining these properties, Tucker's vegetation index (NDVI) is produced as follows [11]:

$$NDVI = (\lambda_{NIR} - \lambda_{RED})/(\lambda_{NIR} + \lambda_{RED}) \quad (2.1)$$

where λ_{NIR} and λ_{RED} are the reflectance in the near infra red (NIR) and RED band, the values vary in [-1, 1].

As NDVI is not sensitive to influences of soil background reflectance at low vegetation cover and lag vegetation response to precipitation deficiency, NDVI itself does not reflect drought or non drought condition [15]. But the anomaly of NDVI from its long term mean can define

drought. The deviation of NDVI is calculated as the difference between the NDVI at current time step, such as ten day period (dekadal), and a long-term mean of the same time step for each pixel. When $NDVI_{deviation}$ is negative, it indicates the below normal vegetation condition and, probably, drought. Furthermore, the severity of drought can be characterized in the amount of negative deviation. The formula is given as [11]:

$$NDVI_{dev} = NDVI_{i,dekad} - NDVI_{i,mean} \quad (2.2)$$

where $NDVI_{i,dekad}$ is the value for the i^{th} dekad, and $NDVI_{i,mean}$ is the long term mean value of the historical $NDVI_i$ for same time dekad i and it is given as

$$NDVI_{i,mean} = \frac{1}{N} \times \sum_{i=1}^{N} NDVI_i \quad (2.3)$$

where N is the number of historical *NDVI* images considered for the mean computation and $NDVI_{dev}$ value range is between -1 and 1.

The severity level of crop unhealthy condition is measured in terms of $NDVI_{dev}$ value. The more the deviation is closer to -1, the worse the vegetation health. Regarding the pixel level deviation of NDVI values for classification of the extent of severity, a recent work in Ethiopia was done by Getachew et al. [55], which is summarized in table 2.1.

Table 2.1 NDVI Deviation Classes and Range of Values [55].

No	Drought classes	Range of $NDVI_{dev}$ values
1	Severe drought (extremely dry)	<= -0.2
2	Drought (moderately dry)	>-0.2 and <= -0.05
3	Near normal	> -0.05 and <=0.1
4	Above optimum (extremely wet)	> 0.1

This research focuses on the agricultural drought. Satellite image-based NDVI is the most commonly used index for crop health evaluation. The test performed so far to check the validity of remote sensing-based drought indices had shown that NDVI based images produce better results for vegetation changes and drought monitoring because of its high correlation with precipitation [21].

2.2 Image Preprocessing and Segmentation

Because of the susceptibility of the reflectance from the surface to different interferences like cloud, satellite images suffer from noises. This necessitates preprocessing to reduce or remove those noises before segmentation is performed on the image.

2.2.1 Preprocessing

Different algorithms have been developed to increase both the accuracy and the interpretability of the digital data during the image processing phase. Preprocessing of satellite images commonly comprises a series of sequential operations, including atmospheric correction or normalization, image registration, geometric correction, and masking (e.g., for clouds, water, irrelevant features) [22, 23].

Traditional filters like mean and Gaussian filters smooth noise in the image, they also smooth the edges, blurring them and changing their location. S. K. Weeratunga et al., [24] applied partial differential equation (PDE)-based non-linear diffusion techniques to overcome these problems and had shown producing good result. In the same article, the comparison among different non-linear approaches showed that non-linear isotropic diffusion-based denoising produced lower mean square error (MSE) compared to the best methods based on wavelet thresholding and spatial filters.

In [25], the analysis for performance evaluation of non-linear filtering algorithms for under water images showed that spatial median filter produces lower MSE and higher Peak Signal to Noise Ratio (PSNR) compared to the mean and other types of median filters. But it also demonstrated that Gaussian noises are removed poorly than salt and pepper type noise by this filter.

Dr. G. Padmavathi et al., [26] performed a comparison among homomorphic, anisotropic diffusion and wavelet denoising by average filters for under water image pre-processing. The research outcome demonstrated that anisotropic diffusion reduced speckles better than wavelet and homomorphic filters had least computation time, better than spatial filtering for most types of noises, preserved edges and resulted in fewer segmented regions for the succeeding segmentation.

Jiying Yuan et al., [27] applied the anisotropic diffusion algorithm produced by Perona P. and Malik J. [28] to high resolution satellite images. According to this research outcome, compared

to any linear low-pass filter such as Gaussian filter, the P&M filter leads to a better performance in image segmentation at a coarse scale due to its multi-scale property.

2.2.2 Segmentation

In the paper of Vincenzo BARRILE et al., [30] the object-oriented analysis of satellite images used a segmentation to fuse different pixels to define an object depending on given threshold level. The algorithm proceeds fusing adjacent polygons beginning from every pixel of the image until the change of observable heterogeneity between the two original polygons and the new generated polygon does not exceed the threshold defined from the customer.

Leen-Kiat Soh et al., [29] described a segmentation technique that integrates traditional image processing algorithms with techniques adapted from knowledge discovery in databases (KDD) and data mining to analyze and segment unstructured satellite images of natural scenes. Here, the segmentation procedure employed include dynamic thresholding, generation of spectral, spatial, and textural features, and, finally, conceptual clustering to cluster the regions found in the image in to N classes-thus determining the number of classes in the image automatically.

In [31], the segmentation was performed to extract fire object from satellite images. Here, the center of the object was defined based on the maximum intensity pixel and the boundary delineation was determined on the intensity deviation from the center as per the threshold defined.

In the work by Ujjwal Maulik et al., [32] the importance of unsupervised pixel classification in remote sensing satellite imagery using clustering in the spectral domain was demonstrated. The problem of classifying an image into different homogeneous regions is viewed as the task of clustering the pixels in the intensity space. In this work, the challenge of detecting regions or clusters varying sizes was addressed using differential evolution based fuzzy clustering technique.

In [33] and [48], unsupervised clustering methods were used for data clustering and image segmentation was proposed. To evaluate the nonlinear image region segmentation, quantitative statistical measures have been used, such as the gray level energy, discrete entropy, relative entropy, mutual information and information redundancy.

2.3 Dimensionality Reduction and Feature Extraction

The number of pixels grouped together to define a given object (eg: drought) in an image is large. Hence, the dimension of an object defined in terms of number of pixels would be very huge to use all these pixels as an input to subsequent processing unit like prediction or classification. To obtain a more discriminant representation of data, and to reduce the high dimensional image data for optimal number of features, researchers have produced a bunch of algorithms.

The work by Charles Bouveyron et al., [34] used different dimensionality reduction approaches for the classification of objects in natural images. This work demonstrated that the representation of data in lower dimension using Principal Component Analysis (PCA) and Linear Discriminant Analysis(LDA), produced good recognition result while that of Locally Linear Embedding (LLE) was found to be worse than either of the two.

Further investigations by Aleix M. Martinez et al. [35] in due of identifying the best approach among the linear dimensionality reduction algorithms exposed that PCA outperforms LDA when size of learning database is small. In many practical domains one never knows in advance the underlying distributions for the different classes. Thus, in practice it would be difficult to ascertain whether or not the available training data is adequate for the job.

In the paper of P. E. Robinson et al.,[36] PCA algorithm performed better at face detection tasks than the LDA and PCA algorithms combined in that order. Though the PCA and then LDA algorithm being applied together found to produce better recognition result than PCA alone, the computation time was far worse than PCA.

2.4 Time Series Prediction

Granger and Newbold [39] describe a time series as ‘‘. . . a sequence of observations ordered by a time parameter.’’ Time series may be measured continuously or discretely. Accumulations of remote sensing images measured discretely at regular intervals would be an example of the later. Time series forecasting is the use of a model to forecast future events based on known past events to predict data points before they are measured. This early knowledge of the future states could be helpful for various applications like drought monitoring.

In the study by Tae-Woong Kim et al., [38] a conjunction model based on dyadic wavelet transforms and neural networks is presented to forecast droughts. Neural networks have shown

great ability in modeling and forecasting nonlinear and non-stationary time series in a water resources engineering, and wavelet transforms provide useful decompositions of an original time series. Neural networks were used to forecast decomposed sub-signals in various resolution levels and reconstruct forecasted sub-signals.

The paper of Mishra A.K. et al., [37] compares linear stochastic models (Autoregressive Integrated Moving Average (ARIMA)/ Seasonal (SARIMA)), recursive multistep neural network (RMSNN) and direct multi-step neural network (DMSNN) for drought forecasting. The models were applied to forecast droughts using standardized precipitation index (SPI) series as drought index. The results obtained from three models and their potential to forecast drought over different lead times are presented in this paper. Out of the many observations, the results obtained from the models show that recursive multi-step approach is best suited for one month ahead prediction. In addition, when longer lead time of four months is considered direct multi-step approach outperforms recursive multi-step and ARIMA models.

Increasingly, the capability of neural networks for prediction of time series variables has been drawing the attention of the research community. In [41] it was shown that the dynamics of nonlinear systems that produce complex time series can be captured in an artificial neural network model, trained with back-propagation algorithm, in a multi-step prediction framework. Holger R. Maier et al., [42] made a review of modeling issues and applications in the use of neural networks for the prediction and forecasting of water resources variables. The review performed on 43 papers dealing with the use of neural network models for the prediction and forecasting of water resources variables yielded that all but two of the papers used feed forward networks. Furthermore, the application of both long term and short term load forecasting using neural networks produced promising results [40, 43].

But the most recent work related to drought prediction done by Siqi Ding et al., [15] evidenced that Markov chain with fuzzy membership function has the potential to be applied in vegetative drought prediction and provide benefit for early warning system. Here, the dynamics of vegetative drought is modeled using Markov chains applied to fuzzy states or classes.

CHAPTER 3

SATELLITE DATA PROCESSING FOR DROUGHT CHARACTERIZATION

3.1 Introduction

This research depends on data provided by satellite for drought characterization and its prediction. Hence, this chapter of the report deals with the nature and derivation of the remote sensing data used for the work, and its processing to extract the features of drought objects in spatial distribution for prediction. The necessary image processing techniques required before prediction are all addressed.

3.2 Source and Nature of Remote Sensing Image Data for Drought Monitoring

This section provides the details of sources of the basic data used in this work and the derivation of new set of image data for drought description.

3.2.1 Source of Data

Remote sensing data are available from satellites of MSG and NOAA center for environmental monitoring [4, 21].

MSG is the new European system of geostationary meteorological satellites together with the associated infrastructure. It was developed to succeed the highly successful series of original Meteosat satellites that has served the meteorological community for over two decades since it was first launched in 1977 [44]. The advanced Spinning Enhanced Visible and Infrared Imager (SEVIRI) radiometer onboard MSG series of geostationary satellites enable the Earth to be scanned in 12 spectral channels, from visible to thermal infrared, at 15-minute intervals. Each of the 12 channels has one or more specific applications, either when used alone or in conjunction with data from other channels. Out of these 12 channels, channels 1 and 2 were used in this thesis for detecting vegetation condition. These two visible channels are well-known from similar channels of the Advanced Very High Resolution Radiometer (AVHRR) instruments that are flown on NOAA satellites; and can be used in combination to generate vegetation indices (such as NDVI) [44].

NOAA is owned by the US government. The sensor on board NOAA missions that is relevant for earth observation is the AVHRR. NOAA and National Aeronautics and Space Administration (NASA) have jointly produced long-term AVHRR datasets that have been

processed in a consistent manner for global change research. These datasets cover the period from July 1981 to present. The datasets are 10-day composites of daily data (red, NIR, and thermal wavelengths), mapped to a global equal area projection at 0.10 resolution [44]. There are three 10-day composites per month; the first is for days 1 through 10, the second is for days 11 through 20, and the third is for the remaining days. The data contain NDVI, a highly correlated parameter to surface vegetation, derived from the visible and near IR channel reflectance [44, 45]. This pathfinder dataset has gone through many stages of calibration and correction [46].

3.2.2 Derivation of the NDVI Deviation Image

Historical NOAA AVHRR data was obtained from Famine Early Warning Systems Network (FEWSNET) website. The metadata for this image was well documented and accordingly some pre-processing was done. The data was encoded using 8 bits and according to the metadata – water and cloud pixels are represented with value 255 and NDVI with unknown equivalent ground feature is represented with value 253. Taking these into consideration, the raw data was converted to its NDVI value based on the recommendation of the metadata using Remote Sensing and Geographic Information System (GIS) software called Integrated Land and Water Information System (ILWIS).

NOAA AVHRR image was found for the whole of Africa and sub-map of Ethiopia was extracted from this image. For this purpose a new geo-reference with GeoRef corners in ILWIS 3.7 software was produced. This new geo-reference was applied to all the NOAA AVHRR historical image data. Thus, the extracted image of Ethiopia was used as input to $NDVI_{dev}$ calculation.

After processing the dataset, the deviation of normalized difference vegetation index ($NDVI_{dev}$) was calculated using equation (2.2).

For example, as the image data available for 1981 is not complete, the deviation of NDVI image for the reportedly historical drought occurrences in 1984 was produced using the mean of the images of the years 1982 and 1983 (Fig 3.1). This mean and corresponding $NDVI_{i,dekadal}$ computation was performed for every time step (dekadals). Fig 3.2 shows the deviation of NDVI for the first dekadal of growing season of the year 1984 (June 1-10, 1984).

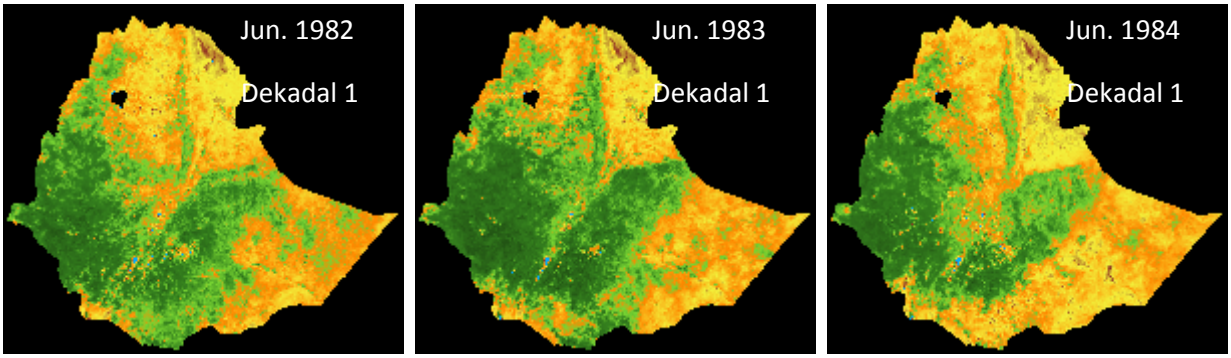


Fig 3.1 Sample NDVI Images of the Years 1982, 1983 & 1984 for the 1st Dekadal of June

Using the mean of the first two images the deviation of the third image from the mean was produced and shown in Fig 3.2.

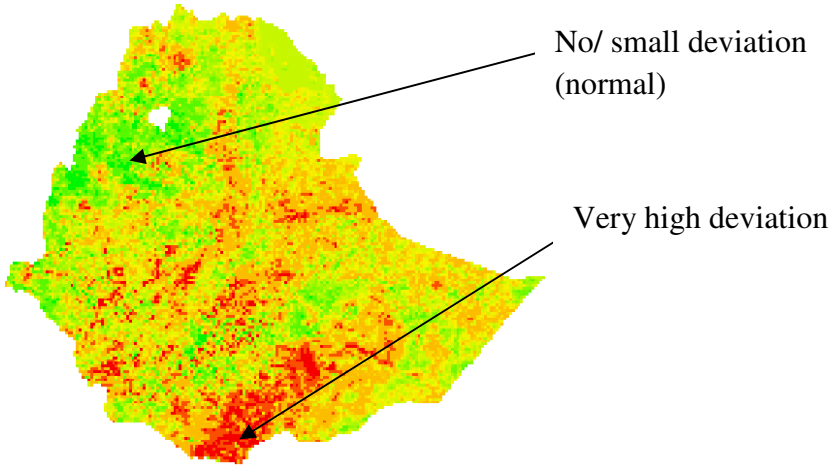


Fig 3.2 Sample Deviation of NDVI

The distribution of color in Fig 3.2 is according to the deviation of NDVI values, with two extreme values indicated in the figure. The same procedure was repeated for all the dekadals from June to September. The NDVI images and the derived deviation of NDVI images for 1984 are shown in Appendix B.

3.3 Noise Removal and Smoothing

Though some preprocessing has been performed on the data of the satellite images by the providers (FEWSNET), some smoothing and noise removal techniques has to be applied. This is because the data could still suffer from sharp changes due to occurrence of missing values, and persistence of clouds and atmospheric conditions for more than a dekadal [4]. Furthermore,

satellite images are subjected to noises due to data dropout, transmission problem, etc [47]. For example, speckle noise is a ubiquitous artifact that limits the interpretation of optical coherence of remote sensing image.

Therefore, this section provides relevant image smoothing and noise removal as part of a pre-processing techniques. As a responsive measure to the occurrence of noises in image different types of filtering algorithms have been developed. The most popular types of filters are divided into two main categories: linear and non-linear filters.

Linear filters: These are filters that accept input signals to produce output signals, subject to the constraint of linearity which is given as [22]:

$$y = L(x) \tag{3.1}$$

where *L (*)* is a linear operator, i.e., it satisfies both the superposition and identity principles.

Mean or averaging and Gaussian filters are among the most frequently used filters of this type. These filters are good at smoothing noise but they equally smooth the edges, blurring them and changing their location [24, 27].

Non linear filters: A nonlinear filter is a signal-processing device whose output is not a linear function of its input. An important class of nonlinear filters is the statistical filters, e.g. the different types of median filter, which require sorting of its input. The reportedly best filter, spatial median filter, performs better at removing different types of noises (e.g: salt and pepper, and speckle type noises) than any of the linear types [25].

In addition, there are some typical filters in image processing for noise removal and smoothing like anisotropic diffusion and wavelet denoising. Anisotropic diffusion filtering is better than wavelet and spatial median filter at speckle reduction and other most types of noises [25]. It is also good for the subsequent segmentation process to result in optimum number of segments [27].

Attributed to the characteristics of the image and aim of smoothing under this study anisotropic diffusion was used for satellite image denoising. This filter is good at removing speckles or sharp noises that are likely to occur in satellite data due to insufficient registration, and it also doesn't need one to know the noise pattern or power spectrum, if any.

Anisotropic diffusion filter

In image processing and computer vision, anisotropic diffusion, also called Perona–Malik diffusion, is a technique aiming at reducing image noise without removing significant parts of the image, typically edges, lines or other details that are important for the interpretation of the image [28]. It is an efficient nonlinear technique for simultaneously performing contrast enhancement and noise reduction.

The main concept of anisotropic diffusion is diffusion, a physical process for balancing concentration changes. In the filtering context, the image intensity can be seen as a "concentration" and the noise can be modeled as little concentration inhomogeneities. These irregularities could be smoothed by diffusion.

The equation for anisotropic diffusion is [24]:

$$\begin{cases} \dfrac{\partial I}{\partial t} = div(c(|\nabla I|).\nabla I) \\ I(t=0) = I_o \end{cases} \tag{3.2}$$

Here, I_o is the original image, I is the evolution result of the original image at different time t, *div* is the divergence operator, ∇ is a gradient operator and c is the diffusion coefficient.

Perona and Malik [28] proposed two options for the diffusion coefficient:

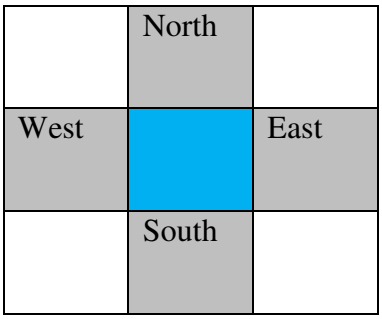
$$c(x) = \frac{1}{1+(\nabla I / k)^2} \qquad OR \qquad c(x) = \exp(-(\nabla I / k)^2) \tag{3.3}$$

where, k is the diffusion parameter.

The anisotropic diffusion method can be iteratively applied to the output image considering 2-dimensional network structure of 4- neighboring nodes for diffusion conduction as [28]:

$$I^{(n+1)} = I^{(n)} + \lambda \times \begin{bmatrix} c\left(\left|\nabla_{North} I^{(n)}\right|\right).\nabla_{North} I^{(n)} + c\left(\left|\nabla_{East} I^{(n)}\right|\right).\nabla_{East} I^{(n)} \\ + c\left(\left|\nabla_{West} I^{(n)}\right|\right).\nabla_{West} I^{(n)} + c\left(\left|\nabla_{South} I^{(n)}\right|\right).\nabla_{South} I^{(n)} \end{bmatrix} \tag{3.4}$$

where parameter k~[20,100] is gradient modulus threshold that controls the conduction, step size $\lambda <= 0.25$, n is the iteration number, and the subscripts North, East, West and South indicate the location of the neighbor pixel as shown in Fig 3.3.



.

Fig 3.3 Implementation of Anisotropic Diffusion

An eight neighborhood network is implemented in this work to obtain better results.

3.4 Segmentation to Extract Virtual Drought Object

In drought object characterization, data processing and information dissemination, the concept of virtual reality is very important. Virtual reality is a computer graphic technology that can be used to emulate the real world in three dimensions, with which users can participate in the virtual environment by walking or flying. The virtual geographic objects can be defined as measurements having geographic information but do not represent physical features on earth [7]. The drought objects in this case are defined based on some attributes of physical features, such as NDVI from satellite images. Thus, to clearly identify drought objects and use it for further processing, segmentation of the image is necessary.

The purpose of image segmentation is to partition an image into meaningful regions with respect to a particular application, e.g., drought object extraction. The segmentation is based on measurements taken from the image and might be grey level, colour, texture, depth or motion.

From the high level view, there are two main classes of image segmentation approaches. These are:

i) **Hard-coded (crisp) segmentation**: This is a traditional segmentation approach which is based on accept or reject principle. A pixel can either belong to an object or not; no other alternative. It can be represented mathematically as [29, 30]:

$$A(x) = \begin{cases} 1, & if \quad x \in A \\ 0, & if \quad x \notin A \end{cases} \tag{3.5}$$

Some of the widely used techniques of this type are static or global threshold, adaptive threshold, region growing using thresholding, k-means clustering and hysteresis threshold. But the crisp method falls short in image segmentation because it doesn't incorporate the intensity heterogeneity characteristics of object in image which is caused by heterogeneity of object material and blurring of the imaging device.

ii) Fuzzy segmentation: this is the segmentation approach in which belongingness to a class is defined by a membership function the value of which is in the interval [0, 1]. Despite the accept or reject principle of crisp approach, fuzzy segmentation gives smooth range of class membership between 0 and 1, including the boundaries. Some of the approaches here include fuzzy thresholding, segmentation based on clustering, region growing using membership function and fuzzy connectedness.

Since the intention of the segmentation step in this work is to extract the drought objects of different classes (e.g. low, medium, severe, etc) which appear on the NDVI deviation image, and taking into consideration the computational complexity and other limitations, k-mean clustering technique was selected for this work. The details of this algorithm are provided below.

K-Means clustering: K-Means algorithm is an unsupervised clustering algorithm that classifies the input data points into multiple classes based on their inherent distance from each other [48]. The algorithm assumes that the data features form a vector space and tries to find natural clustering in them. The points are clustered around centroids $u_i, \forall i = 1....K$ which are obtained by minimizing the objective [48]

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - u_i)^2 \tag{3.6}$$

where there are K clusters S_i, $i = 1, 2,\ldots, K$ and u_i is the centroid or mean point of all the points $x_j \in S_i$.

In this work the iterative version of the algorithm which includes the following steps was implemented. Inputs to the algorithm is a two dimensional image. The steps are [48]:

a) Compute the distribution of the intensities.
b) Initialize the centroids with K random intensity values.

c) Repeat the following steps until the cluster labels of the pixel do not change anymore.

i) Cluster the points based on distance of their intensities from the centroid intensities.

$$c^{(i)} := \arg \min_j \| x_i - u_j \|^2 \tag{3.7}$$

ii) Compute the new centroid for each of the clusters.

$$u_k := \frac{\sum_{i:C(i)=k} x_i}{N_k} \ , \ k = 1, \ldots, K \tag{3.8}$$

where K is a parameter of the algorithm (the number of clusters to be found), *i* iterates over all the intensities within cluster, N_k is the number of pixels in k^{th} cluster, j iterates over all the centroids and u_k are the centroid intensities. This segmentation results in different drought objects in the image.

3.5 Feature Extraction or Dimensionality Reduction of Drought Objects

The segmentation process in the previous section results in a number of virtual drought objects in the image. Each object for a given geographical location can be localized and sliced for the succeeding step. But the problem is that it would be difficult to consider an object as a vector of pixels and use it for the prediction model because of the curse of dimensionality. Thus, reducing the dimension of drought objects and selecting the most principal representation is an important issue. Therefore, this section is important for two main reasons:

1) To obtain a more discriminant representation of data
2) To reduce the high dimensional image for optimal number of features to be used as an input to the prediction model.

There are two categories of techniques usually employed for this purpose. These are:

i. **Linear methods:** These are methods that incorporate the linearity rule between input and output data [49]. The most common methods are Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Canonical Correlation Analysis (CCA).

ii. **Non-linear methods:** According to a negative definition, non linear methods are those which do not satisfy the linearity law. Locally-Linear Embedding (LLE) is one such example.

From the observed features and performances of the methods LLE produced the worst performance of all the methods in terms of classification results and computation time [34].

Furthermore, as the aim of dimensionality reduction here is accurate signal representation in lower dimensions than class discriminatory information in lower dimensional space, PCA is the preferred algorithm [35, 49]. The details of the algorithm are presented as follows:

Principal Component Analysis (PCA): The central idea of PCA is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible the variation present in the data set. This is achieved by transforming to a new set of variables, the principal components (PCs), which are uncorrelated, and which are ordered so that the first few retain most of the variation present in all of the original variables.

Given a N^2-dimensional vector representation of each drought object, the PCA [35] can be used to find a subspace whose basis vectors correspond to the maximum-variance directions in the original space. Let W represent the linear transformation that maps the original N^2-dimensional space onto a K-dimensional feature subspace where normally $K << N^2$. The new feature vectors y_i are defined by $y_i = W^T x_i$; i = 1,..., N. The columns of W are the eigenvectors e_i obtained by solving the eigenstructure decomposition $\lambda_i e_i = Qe_i$, where $Q = XX^T$ is the covariance matrix and λ_i the eigenvalue associated with the eigenvector e_i .

The first stage of the PCA system is the training stage. A set of drought images which is made up of classes of images of objects that should be recognized by the system is used as a training set. The training set is used to create a covariance matrix of the training data whose strongest eigenvalues will form the basis of the vector space spanned by all the training objects.

Suppose that Γ is an N^2 x1 vector, corresponding to an NxN drought image I,

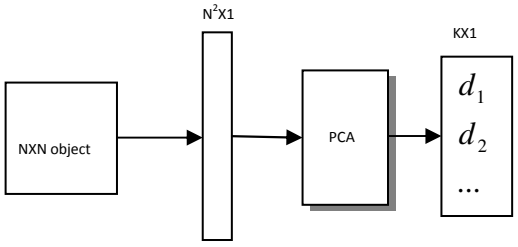


Fig 3.4 Dimensionality Reduction Using PCA, where $K<<N^2$

The idea is to represent Γ (*Φ=Γ – mean object*) in a low-dimensional space [34, 35]:

$$\Phi - mean = d_1\mu_1 + d_2\mu_2 + ... + d_K\mu_K \, (K << N^2) \tag{3.9}$$

where μ_i's are eigenvectors, d_i's are descriptors, and Φ is the difference between the vector Γ and mean of objects in the training set of PCA.

Steps for the computation of eigenvectors:

Step 1: obtain object images $I_1, I_2, ..., I_M$ (drought objects)

Here, M is the total number of sample drought objects in training of PCA.

Step 2: represent every image I_i as a vector Γ_i

Step 3: compute the average object vector Ψ:

$$\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i \tag{3.10}$$

Step 4: subtract the mean drought object:

$$\Phi_i = \Gamma_i - \Psi \tag{3.11}$$

Step 5: compute the covariance matrix C:

$$C = \frac{1}{M}\sum \Phi_n \Phi_n^T = AA^T \qquad (N^2 \times N^2 \quad matrix)$$

$$where \quad A = [\Phi_1 \quad \Phi_2 \quad ... \quad \Phi_M] \qquad (N^2 \times M \quad matrix) \tag{3.12}$$

Step 6: compute the eigenvectors μ_i of AA^T

As the matrix AA^T is very large, the computation of the eigenvectors will be very difficult. Thus, the alternative solution is to use the following procedure [35]:

Step 6.1: consider the matrix A^TA (*MxM* matrix)

Step 6.2: compute the eigenvectors v_i of A^TA*:*

$$A^T A v_i = \mu_i v_i \tag{3.13}$$

The relationship between v_i *and* μ_i is given as [34]:

$$A^T A v_i = \mu_i v_i \Rightarrow AA^T A v_i = \mu_i A v_i \Rightarrow$$

$$CAv_i = \mu_i Av_i \quad or \quad C\mu_i = \mu_i\mu_i \quad where \quad \mu_i = Av_i \tag{3.14}$$

Thus, AA^T and A^TA have the same eigenvalues and their eigenvectors are related as $\mu_i = Av_i$. In addition, the AA^T can have up to N^2 eigenvalues and eigenvectors, while A^TA can have up to M

eigenvalues and eigenvectors. The M eigenvalues of A^TA (along with their corresponding eigenvectors) correspond to the M largest eigenvalues of AA^T (along with their corresponding eigenvectors).

Step 6.3: compute the M best eigenvectors of AA^T: $\mu_i = Av_i$. And it is important to normalize μ_i such that $\| \mu_i \| = 1$.

Step 7: keep only K eigenvectors (corresponding to the K largest eigenvalues).

Representing drought objects in this eigenvector space requires the following two steps:

- project a new drought object on to the best K eigenvectors:

$$\Phi_i - mean = \sum_{j=1}^{K} d_j \mu_j, \quad \Rightarrow d_j = \mu_j^T \Phi_i \tag{3.15}$$

  where μ_i 's are called eigendroughts

- Compute the descriptor vector for new object

$$\Omega_i = \begin{bmatrix} d_1^i \\ d_2^i \\ \dots \\ d_K^i \end{bmatrix} \tag{3.16}$$

  where Ω_i is called *descriptor/feature* vector of each object.

The descriptor vectors represent drought object with few principal components. Thus, these vectors are easier to handle and use for further processing like as an input for prediction network than the whole pixel making up the object. Using the above PCA analysis the descriptors for each drought object at a given location were extracted.

CHAPTER 4

ARTIFICIAL INTELLIGENCE: NEURAL NETWORKS

4.1 Introduction

Artificial Intelligence (AI) is the science and engineering of making intelligent machines, especially intelligent computer programs [54]. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.

All the developments that took place in the field of AI and related topics can be classified into eight specialized branches [54]:

1. *Problem Solving and Planning*: This deals with systematic refinement of goal hierarchy, plan revision mechanisms and a focused search of important goals.
2. *Expert Systems:* This deals with knowledge processing and complex decision-making problems.
3. *Natural Language Processing*: Areas such as automatic text generation, text processing, machine translation, speech synthesis and analysis, grammar and style analysis of text etc. come under this category.
4. *Robotics*: This deals with the controlling of robots to manipulate or grasp objects and using information from sensors to guide actions etc.
5. *Computer Vision*: This topic deals with intelligent visualization, scene analysis, image understanding and processing and motion derivation.
6. *Learning*: This deals with research and development in different forms of machine learning
7. *Genetic Algorithms*: These are adaptive algorithms which have inherent learning capability. They are used in search, machine learning and optimization.
8. *Neural Networks*: This topic deals with simulation of learning in the human brain by combining pattern recognition tasks, deductive reasoning and numerical computations.

In this work the focus is on ANN because of its reportedly good result in the prediction of time series patterns [37-43]. Thus, the overview of the overall neural network working principle and the details of the algorithm used for this research is provided in this chapter.

4.2 Artificial Neural Network

ANN is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information [52]. The human brain provides proof of the existence of massive neural networks that can succeed at those cognitive, perceptual, and control tasks in which humans are successful [51]. This natural behavior of biological neurons led to the derivation of a novel structure of the information processing system in computing environment. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example.

4.2.1 Biological Neuron

In the human brain, a typical neuron collects signals from others through a host of fine structures called *dendrites* (Fig 4.1) [52, 53]. The neuron sends out spikes of electrical activity through a long, thin stand known as an *axon*, which splits into thousands of branches. At the end of each branch, a structure called a *synapse* converts the activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes.

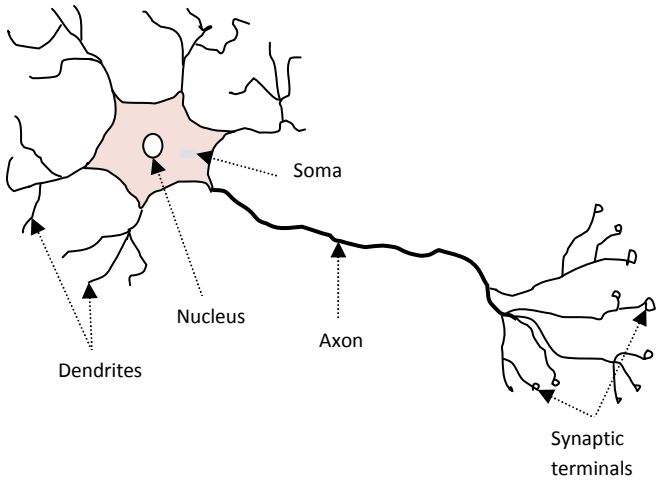


Fig 4.1 Components of a Biological Neuron [53]

4.2.2 Artificial Neuron

An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the testing mode. In the training mode, the neuron can be

trained to fire (or not), for particular input patterns. In the testing mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.

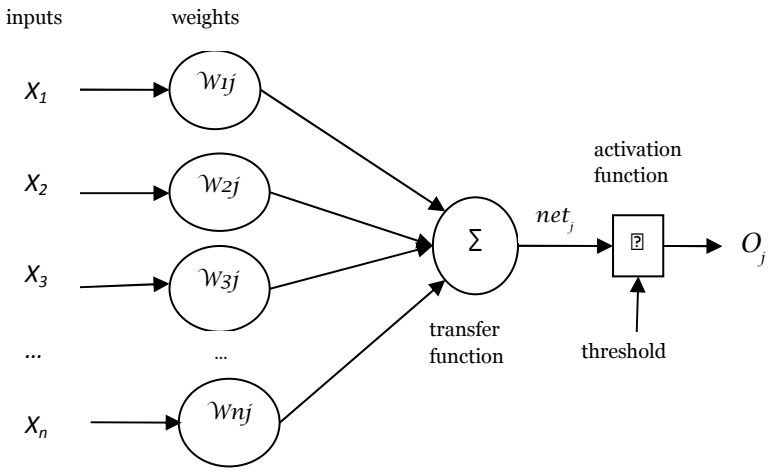


Fig 4.2 The Mccullogh-Pitts Model Artificial Neuron [52]

Neurons work by processing information. They receive and provide information in the form of spikes. In the McCullogh-Pitts' model (Fig 4.2) of artificial neurons, spikes are interpreted as spike rates, and synaptic strength are translated as synaptic weight [53]. The excitation means positive product between the incoming spike rate and the corresponding synaptic weight and inhibition means negative product between the incoming spike rate and the corresponding synaptic weight.

Nonlinear generalization of the McCullogh-Pitts neuron [51, 53]:

$$O_j = \varphi(net_j) \quad \& \quad net_j = \sum_{i=1}^{n} w_{ij} x_i \tag{4.1}$$

where O_j is the neuron's output, n is the vector of inputs, w is the vector of synaptic weights, net_j is the weighted sum of inputs to neuron j, and φ is activation function.

4.2.3 Multilayered Neural Network Architecture

An artificial neural network is composed of many artificial neurons that are linked together according to specific network architecture (Fig 4.3) [53]. A first wave of interest in neural networks (also known as connectionist models or parallel distributed processing) emerged after

the introduction of simplified neurons by McCulloch and Pitts [52]. The basic processing elements of neural networks are called artificial neurons, or simply neurons or nodes.

The basic architecture consists of three types of neuron layers: input, hidden, and output layers as shown in Fig 4.3. In feed-forward networks, the signal flow is from input to output units, strictly in a feed-forward direction. The data processing can extend over multiple (layers of) units, but no feedback connections are present.

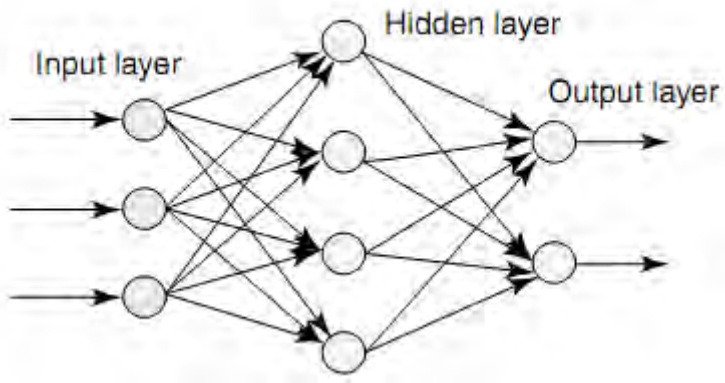


Fig 4.3 Multilayered Artificial Neural Network [56]

There are two approaches that can be used to configure the neural networks so that the application of set of inputs produces desired set of outputs. These are either setting the weights manually or teaching the neural network to learn certain patterns. The later approach is the most commonly used as we want the network learn the pattern by itself and accordingly update its weights.

At a high level, the tasks performed using neural networks can be classified as those requiring supervised or unsupervised learning. In supervised learning, a teacher is available to indicate whether a system is performing correctly, or to indicate a desired response, or to validate the acceptability of a system's responses, or to indicate the amount of error in system performance. This is in contrast with unsupervised learning, where no teacher is available and learning must rely on guidance obtained heuristically by the system examining different sample data or the environment. A concrete example of supervised learning is provided by "classification" problems, whereas "clustering" provides an example of unsupervised learning.

Practically, neural networks are used to simulate the actual situations in the sense of computational task. Some of the application areas include classification, clustering, vector

quantization, pattern association, function approximation, forecasting, control applications, optimization and search problems.

Because of the nature of the problem to be solved by this research, which is prediction of drought, supervised learning method is preferred. In addition, as back propagation algorithm for learning the appropriate weights in supervised approach is one of the most common models used in ANNs, the details of the algorithm is presented in Appendix A.

CHAPTER 5

DESIGN AND IMPLEMENTATION

The overall system model in this thesis has incorporated the following four major steps: preprocessing, segmentation, dimensionality reduction and prediction. The high level view of the overall system is shown in Fig 5.1.

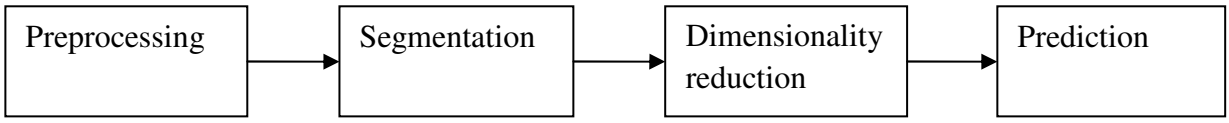


Fig 5.1 General Scheme of the Design

The details of the design and implementation of each major step is provided in detail in this chapter.

5.1 Preprocessing

As mentioned in sections 3.1 and 3.2, the pre-processing techniques include NDVI deviation image preparation and noise removal and/or smoothing. Thus, the high level view of this step is shown in the block diagram of Fig 5.2.

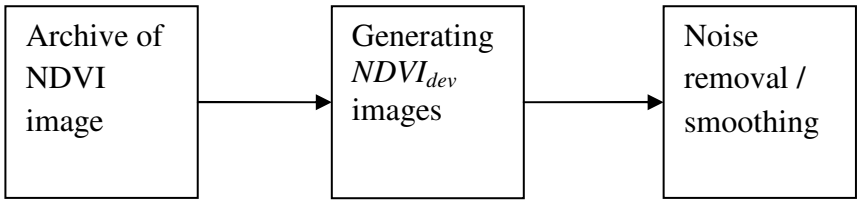


Fig 5.2 Main Steps in Pre-processing

The image data available for this work is the archive of NDVI images of NOAA-AVHRR satellite from the FEWSNET website (1981-2010) and the satellite data from Meteosat (since 2005). The later requires the processing of the data to produce the NDVI image using equation (2.1) and at the same time the historical data available is limited. But, for the training and testing purposes of the intelligent system applied in this work, we need to have some historical occurrences of drought in the country (Ethiopia). This necessitated the use of long term available data from FEWSNET website. Using this available database, the image for the

historical occurrence of drought during 1984 in Ethiopia was used for the implementation of this work.

5.1.1 Generation of Deviation of NDVI-images

This sub-step includes the investigation of historical drought occurrences in the country, the selection of the corresponding NDVI images, and generation of the deviation of the NDVI maps using the mean of previous images to this historical drought time.

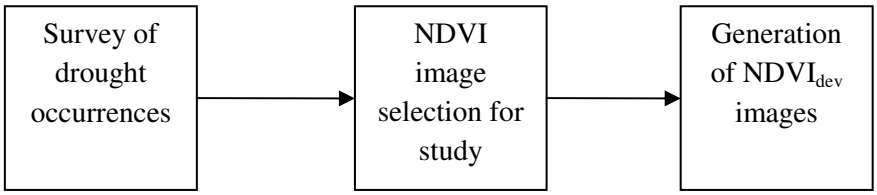


Fig 5.3 Steps in Producing $NDVI_{dev}$ Images

i) Survey of Historical Drought Occurrences: After thorough investigation, well-recognized drought occurrences were found to be in the years 1984, 2002, and 2009. During those times, different regions were suffering from shortage of crop yield, and hence famine. For example, in 1984 drought happened in North Eastern part of the country.

Therefore, for this thesis work, NDVI images of those years mentioned above along with the knowledge of regions which were under threat were considered**.**

ii) NDVI Image Selection for Study: After investigation of historical records of drought in the country in step (i), the NDVI images of the country during those years mentioned under the survey were selected to produce the deviation of the NDVI.

iii) Generation of Deviation of NDVI ($NDVI_{dev}$): Here the deviation of the NDVI images from their corresponding historical means for every selected growing season under survey was produced using equation (2.2). Since the growing season includes the months June to September, the NDVI deviation produced is for each dekadal within this interval, i.e., three dekadals per month and four months per growing season yields twelve $NDVI_{dev}$ images for one series. Sample series of $NDVI_{dev}$ images for the North Eastern part of Ethiopia in the year 1984 from the area of interest (AOI) indicated in Fig 5.4 are shown in Fig 5.5. Appendix B gives all the series of deviation of NDVI images for the growing season of 1984.

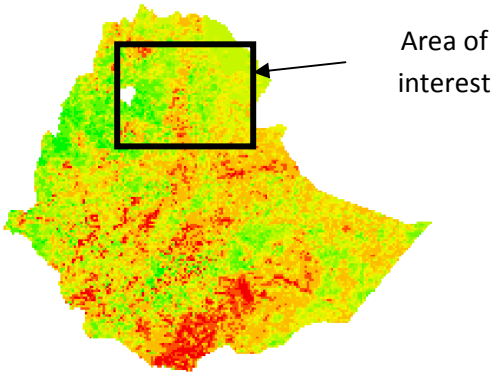
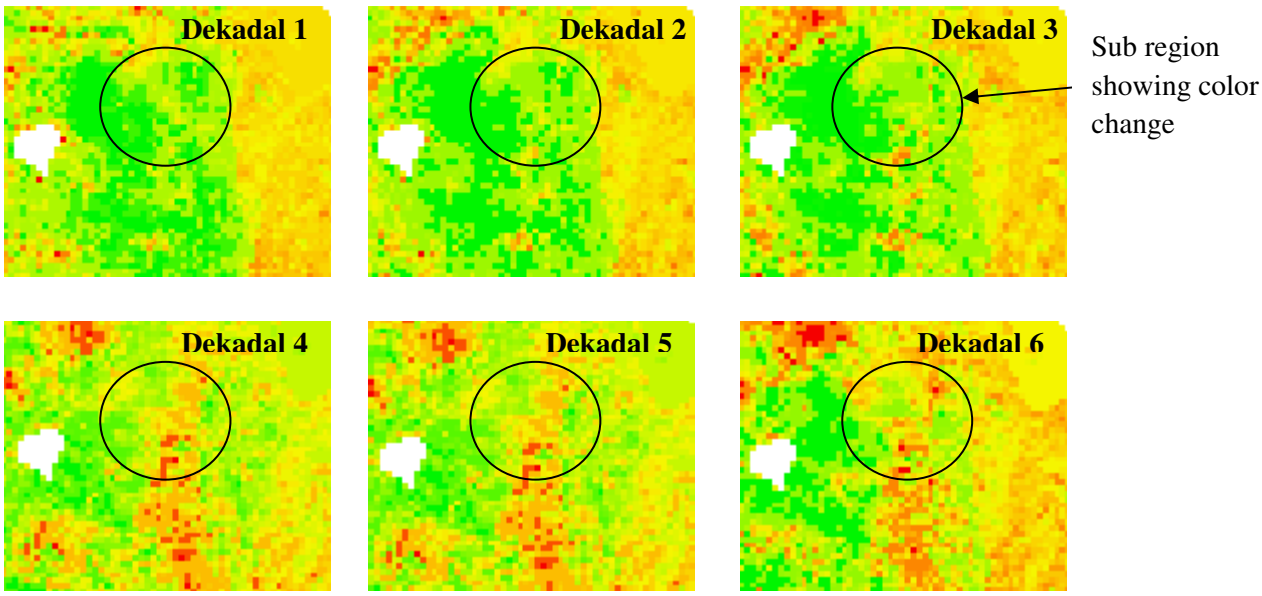


Fig 5.4 AOI to Show the Series of Dekadal

From the AOI shown in Fig 5.4 the following regions are extracted for all the 12 dekadals in the growing season of the year 1984.



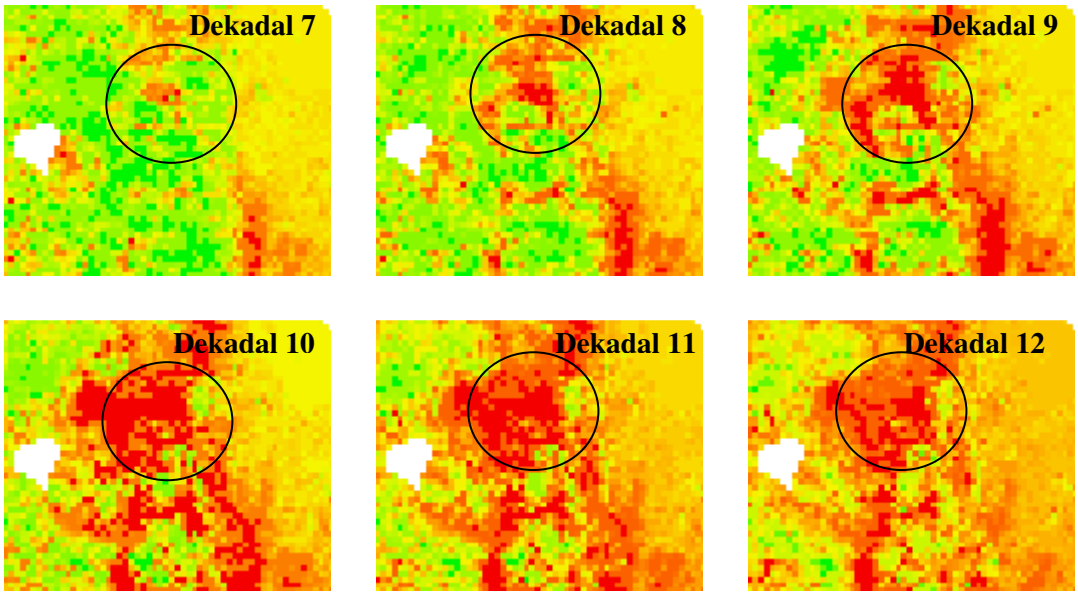


Fig 5.5 Sample Series of Dekadal Images for the AOI of the Growing Season from Jun. to Sep., 1984

The red regions indicate severe unhealthy vegetation condition, the gold ones correspond to moderate droughts, and the green regions indicate normal conditions. As can be observed from circled sub-region in the series of images, there is a change in the color of regions from green to very red that leads to drought.

Once the regions of study are selected, they have to be denoised using the technique mentioned in section 3.3.

5.1.2 Noise Removal

The presence of some insufficient registrations and noisy regions due to missing values like the white region (Lake Tana) in the above samples necessitates the application of image noise minimization step. Therefore, the image smoothing approach described in section 3.3 using anisotropic diffusion was applied to each and every image in the series. Sample image after denoising using this technique is shown in Fig 5.6.

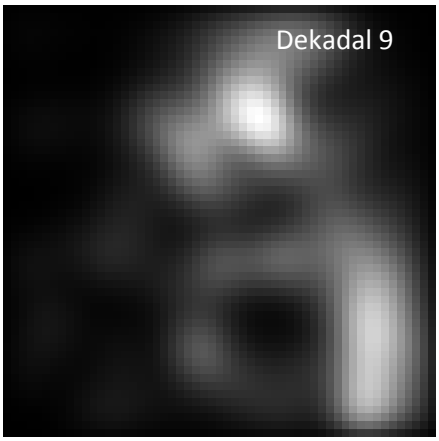


Fig 5.6 Result of Anisotropic Diffusion Applied to Dekadal 9 in the Series of Fig 5.5

In Fig 5.6 the more white the region is the more affected the vegetation health is, and hence, the more apparent drought will be. The result is after the iterative diffusion process for 25 iterations for denoising using equation (3.4) and the step size of $\lambda = 0.25$. It is observable that speckles are removed and the image is smoothed showing inclusions clearly. Moreover, the advantage of this filtering algorithm is that it doesn't require the knowledge of the noise pattern or power spectrum beforehand. But parameter selection and iteration loop selection all affect the final results.

5.2 Image Segmentation

As mentioned in section 3.4 the purpose of the segmentation step is to partition an image into meaningful virtual drought objects of different classes (e.g. low, medium, severe drought objects, etc) and background region with normal vegetation condition. In this work the intention was to mark the boundaries of group of pixels that defines a given drought object. As the intensity values of pixels are of great importance, modification of these values affects the final prediction. Hence, the values were retained through using the segmentation approach only for masking. For this purpose, the k-means clustering algorithm presented in chapter three was used to produce the different clusters of an image region. This segmented image gives different regions with uniform intensity throughout a cluster (Fig 5.7). Hence, it affected the intensity values of some pixels in order to group them to the nearest cluster mean. Thus, to retain the pixel values unchanged, the segmented image was used just to overlay on the original image for locating the groups of pixels defining a drought object. Accordingly, the objects having similar intensities in the segmented (masking) image were extracted from the original (non-segmented) image. The effect of this segmentation to a denoised image can be observed in Fig 5.7.

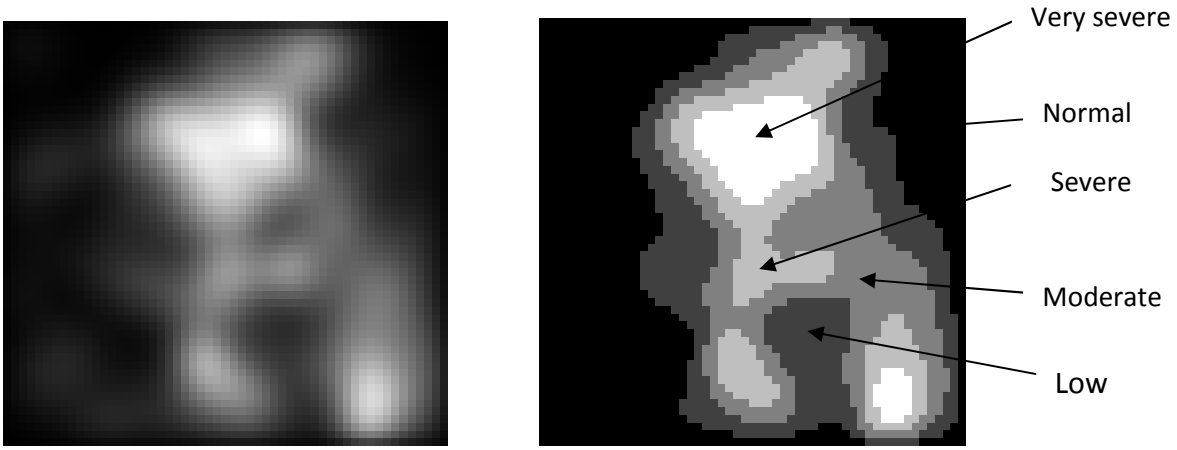


Fig 5.7 Segmentation Using K-Means Clustering: to the Left is the Image before Segmentation and to the Right is Segmented (Masking) Image.

The segmentation parameter k was considered to be five (k=5) depending on the validity measure provided in equation 5.3 and accordingly, table 5.1 was devised. The different classes are pointed at in Fig 5.7.

As shown in the figure, the segmentation results clearly indicate the extent of crop unhealthy condition as manifested using the gray scale image intensity classes. The arrows point to some of those classes, like very severe drought with white region, severe drought with less white, etc until all the abstract drought objects are located and finally, a normal background region with black color. This segmented image was overlaid on the original image to extract drought objects. This segmentation was applied to all the images in a given series. Hence, virtual drought objects can clearly be identified.

Table 5.1 Different Classes of Drought Considered for Segmentation

Class of Segmentation	Name of Drought Class	Description
1st class	Very severe	High deviation of vegetation condition from normal
2nd class	severe	Crops are unhealthy but not as threatening as that of the first class
3rd class	Moderate	Still there is abnormality but less than what is in the first two classes
4th class	Low	There is an anomaly in crop health and attention is required
5th class	Normal	The vegetations are normal

Cluster validity measure: Many criteria have been developed for determining cluster validity [57], all of which have a common goal to find the clustering which results in compact clusters which are well separated. In this work the ratio of the distance within cluster (intra- class) to between cluster centers (inter-class) is used to select a good number of clusters. The intra and inter class measures are given as [57]:

$$intra\_class = \frac{1}{N}\sum_{i=1}^{K}\sum_{x\varepsilon C_i}||x - z_i||^2 \quad (5.1)$$

and

$$inter\_class = \min\left(\left|\left|z_i - z_j\right|\right|^2\right),\ i = 1,2,\dots,K-1; j = i+1,\dots,K \quad (5.2)$$

where N is the number of pixels in the image, K is the number of clusters, x is each pixel intensity value in the given cluster and z_i is the cluster centre of cluster C_i.

Since the within class similarity and across class dissimilarity is a requirement for clustering, we want to minimize the ratio of intra-to-inter class measures. And this is given as [57]:

$$validity = \frac{intra\_class}{inter\_class} \quad (5.3)$$

Hence, using this validity measure the number of classes was devised as given in table 5.1.

The clustering approach was applied to every image in a given series. After the clustering approach the extraction of drought objects from the non-segmented (original) images was performed using the corresponding segmented images. For instance, a rectangular region in Fig 5.8 shows a very severe drought. Thus, this rectangular region was overlaid on the original image to extract the drought object belonging to very severe drought class as shown in Fig 5.9. This process was performed to each and every image in the series. Each specific region is extracted from each and every image in the series. For example, the region indicated by very severe drought in Fig 5.7 was indicated by the rectangular mask in Fig 5.8 in the segmented image. The extraction for all the 12 images in the series of Fig 5.5 was performed and sample is shown in Fig 5.9.

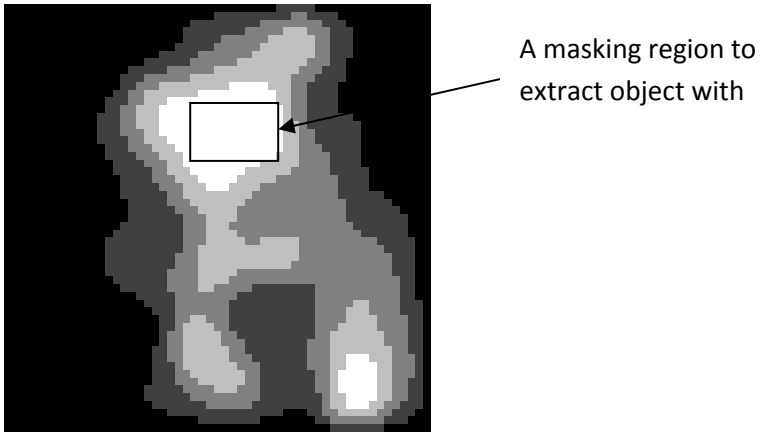


Fig 5.8 Sample Area Used as Mask

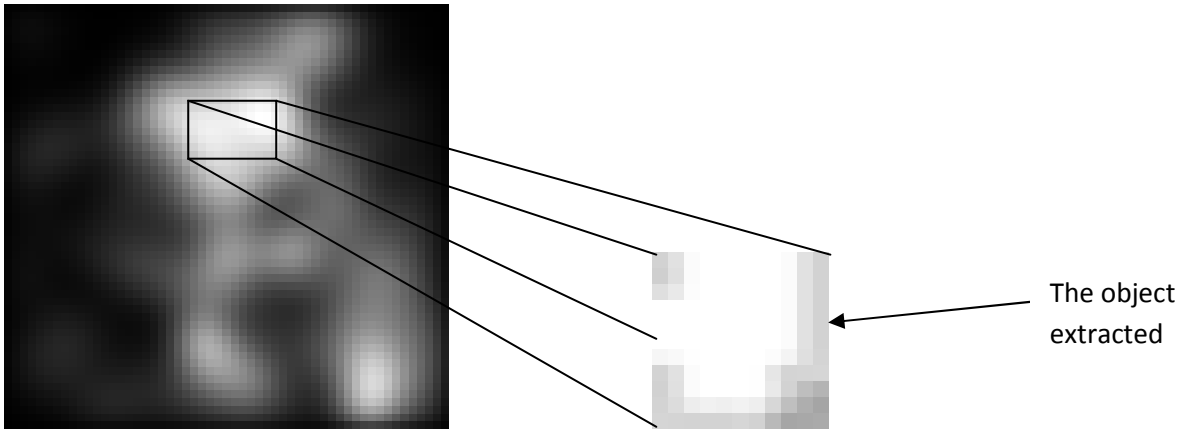


Fig 5.9 Extracting Objects Using the Masks from the Segmented Image

This object extraction procedure was applied to all the images in the series for a specific location. For clarity purpose the mean of intensity value of the objects over the twelve dekadals for a specific region is shown in Fig 5.10.

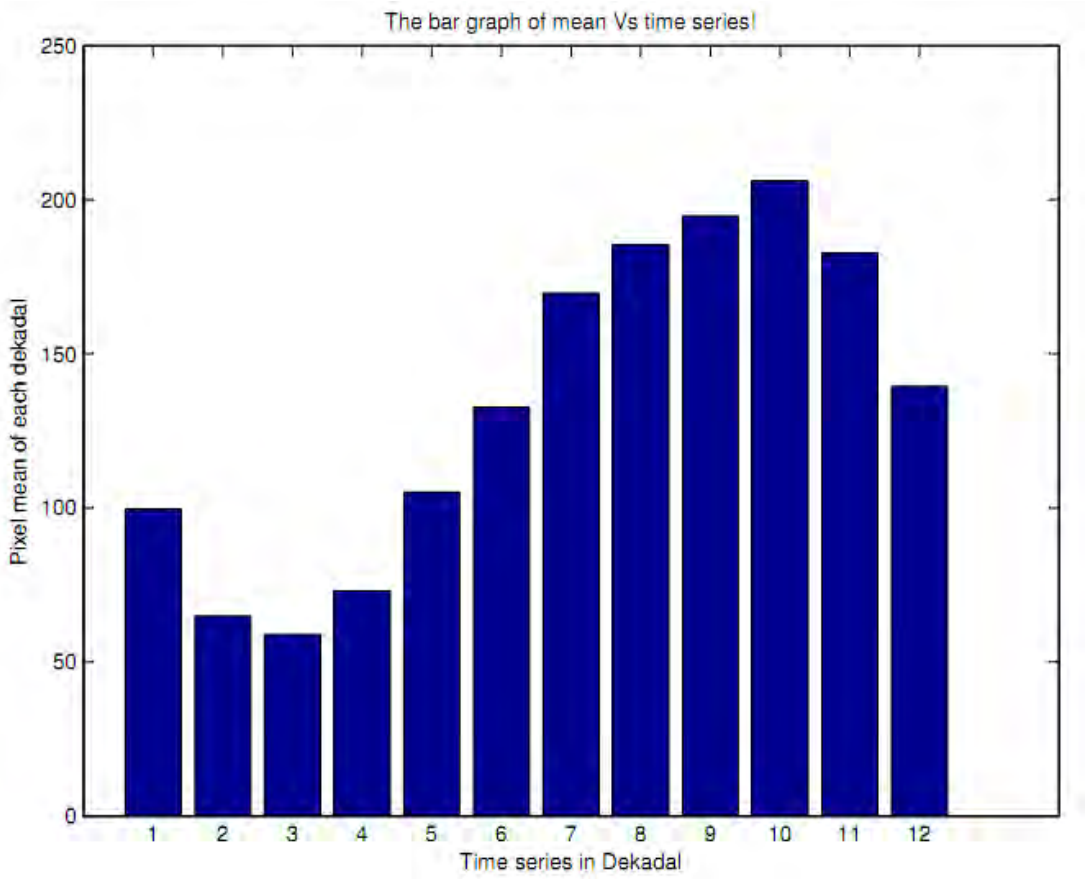


Fig 5.10 The Bar Graph of Intensity Means of Drought Objects Over Twelve Dekadals

The distribution of the mean intensity values in Fig 5.10 shows that the mean intensity value increases over time and reaches the maximum around the first ten days of September. For further illustration, the pattern of drought for two different locations which experienced moderate and severe drought is shown in Fig 5.11. The horizontal axis indicates the time in dekadals with in the growing season which includes the months June, July, August and September. For this number of months we have 12 ten day (dekadal) points. In the figure below, 1 indicates the first ten days of the June month, 2 indicates the next ten days(from June 11 to 20), and so on.

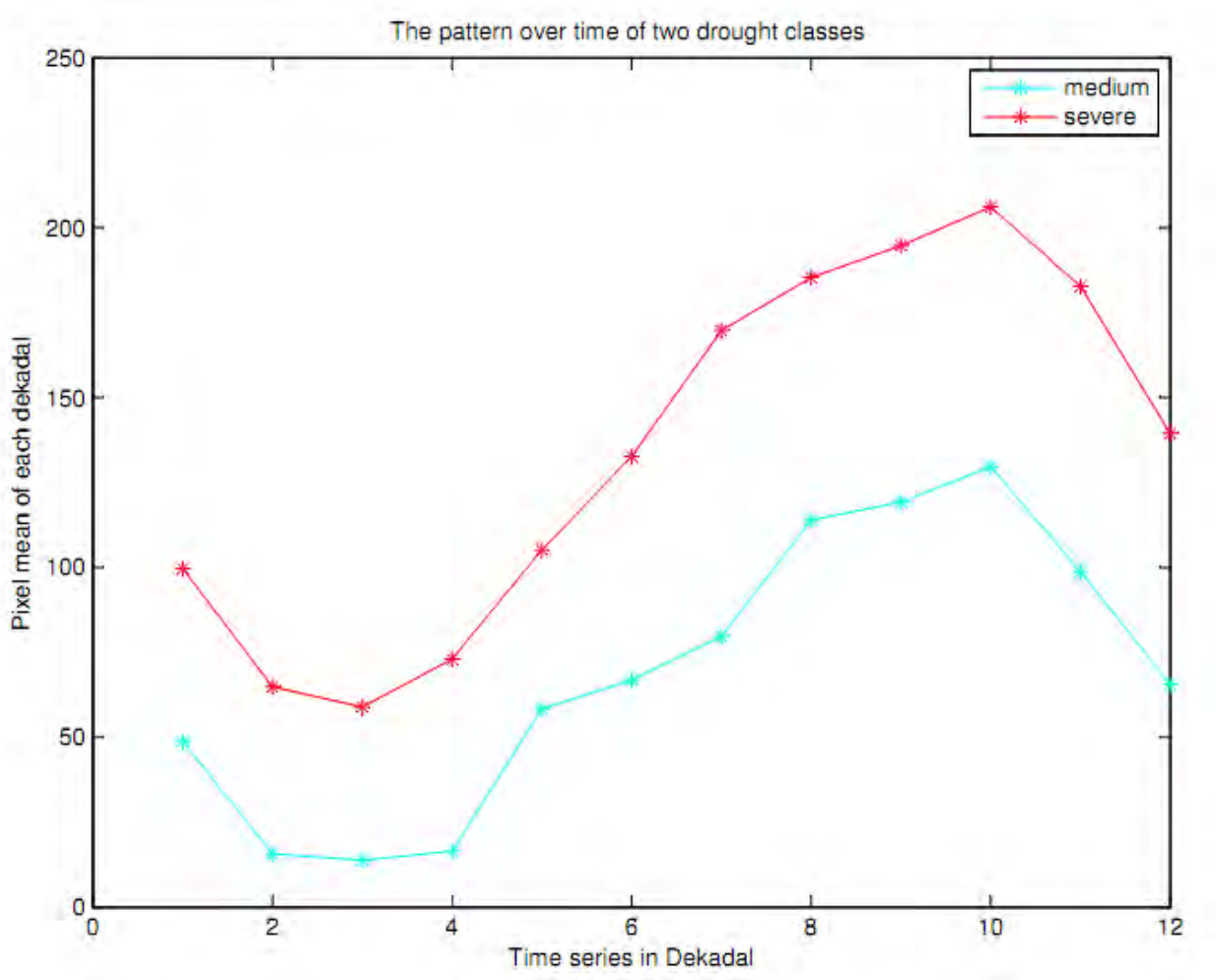


Fig 5.11 Sample Means of the Intensities of Drought Objects over Time During a Growing Season(1984).

It is observable from the graph in Fig 5.11 that the pattern followed by different classes of drought is similar though the values at each time step in the series differ. This difference in the intensity values helps to note that the extent of severity depend on the pattern followed in the graph. For example, the pattern identified as medium in Fig 5.11 has lower mean intensity value than the severe one at every time step in the series.

5.3 Dimensionality Reduction

After the division of an image into spatially continuous and disjoint regions, i.e. the objects and background, the next step is to take those regions and use as an object of interest for dimensionality reduction

The dimensionality reduction step is required because there is a virtual drought object with number of pixels. As these pixels are large in number, it will be difficult to use all of them as features for characterization of objects. Therefore, keeping the information embedded in the pixels, the reduction of this feature space is important. This low dimensional feature space makes the subsequent prediction model easy to realize. Therefore, each object at specific time in the series was given to PCA to extract the most representative feature vector of that object in lower dimension.

There are three main steps undertaken in this reduction technique:

i) Creating representative database for drought objects from different classes

ii) Training the PCA using the representative drought objects and creating a lower dimensional feature space called eigenvector (eigendroughts).

iii) Projections of each drought object in to this feature space to produce its descriptor vector in the lower dimension using equation (3.15).

The overall work performed in PCA analysis here is shown in diagram of Fig 5.12.

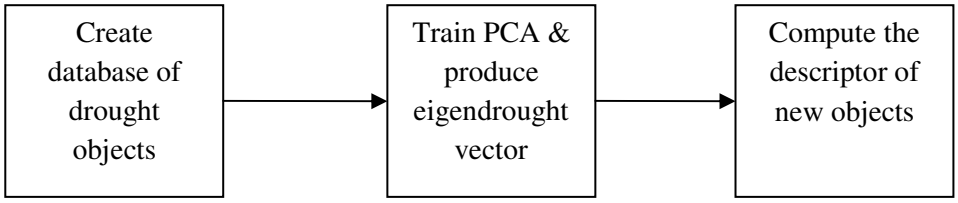


Fig 5.12 Steps in Dimensionality Reduction

Following the procedure in Fig 5.12, the dimension of objects in the series which were supposed to be used for the training and testing of the neural networks was reduced and a lower dimensional representation of the each object was created.

5.4 Prediction Using ANN

The prediction section in this thesis employed neural networks. Neural networks are potential candidates for the forecasting domain because of the advantages such as nonlinear learning and noise tolerance as compared to conventionally used methods, which estimate only linear data [38-42].

Works in neural networks has concentrated on forecasting future developments of a time series from values in the past up to the current time. Formally, this can be stated as: find a function *f:*

$R^N \rightarrow R$ such as to obtain estimate of x at time $t+d$, from the N time steps back from time t such that [40]

$$x(t+d) = f\big(x(t), x(t-1), \ldots, x(t-N+1)\big) \quad \rightarrow \quad x(t+d) = f(y(t))) \tag{5.4}$$

where $\mathbf{y}(t)$ is the N - ary vector of lagged x values. Normally, d will be one, so that f will be forecasting the next value of x.

The standard neural network method of performing time series prediction is to induce the function f using any feed forward function approximating neural network architecture, such as, a standard MLP, a radial basis function (RBF) architecture, or a Cascade correlation model [39], using a set of N-tuples as inputs and a single output as the target value of the network. This method is often called the sliding window technique as the N-tuple input slides over the full training set. Fig 5.13 gives the basic architecture of a network with three time lags such as *x(t-2), x(t-1)*, and *x(t)*; and prediction of the next value of *x* (i.e., *x(t+1)*) in the series.

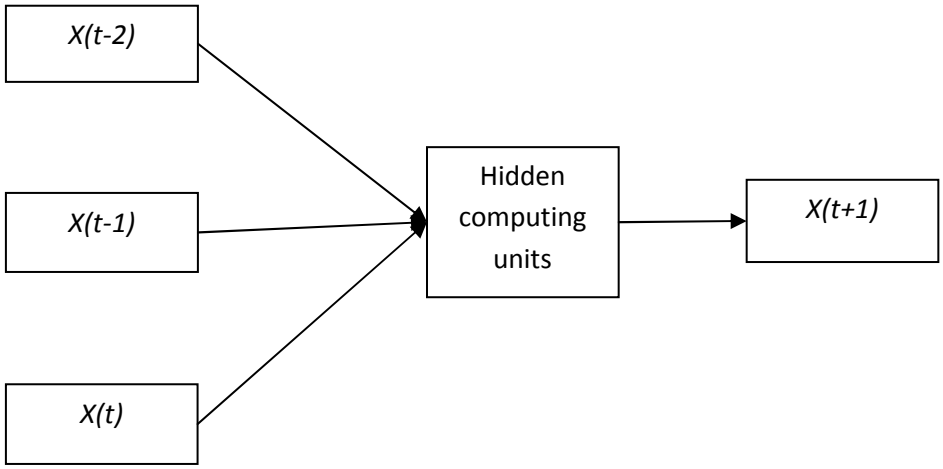


Fig 5.13 The Standard Method of Performing Time Series Prediction Using a Sliding Window of Three Time Steps [40].

In this thesis work, the prediction unit was a neural network using back propagation algorithm. The structural make up of the network is the same as the one shown in Fig 5.13 with slight change. This was because of the fact that the data nature, which can be modeled as a multivariate function. A virtual drought object at a given time stamp has a descriptor vector resulting from the PCA output. The descriptor vector posses a number of components

depending on the number of PCs considered as shown in equation (3.16). This makes the model a multivariate predictor.

The predictor takes the descriptor vectors of different time lags (steps) and produces the descriptor vector of the next time step in the series. As there are multiple parameters (descriptors) per time lag, it can be treated as a multivariate time series. For example, every object at a given time step (dekadal) in the series is a function of its descriptors $d_1, d_2, d_3,\ldots..,d_n$. Therefore, considerations have to be made on how to order these descriptors for input and output of the predictor network. Accordingly, this work employed two approaches in order to incorporate these facts. These are combined and separate network architectures.

Combined network: This approach assumes a drought object as a point in the time series and gives all of its descriptors to the network to determine all of the descriptors of an object in the next time step in the future. Here, the number of input units is the same as the product of the number of time lags considered and the number of elements of every descriptor vector. For example, if we take three time steps as an input to the ANN model with the size of descriptor vector of an object at a given time step being 15, then the number of input nodes of the predicting model will be 45. Furthermore, the number of output nodes is the same as the size of descriptor vector as we want to know the descriptor of an object in the next time step in the series.

The model shown in Fig 5.14 represents the architecture of the neural network using the first approach. The descriptor vectors at time *t, t-1*, and *t-2* of the object for three time lag are given as an input to the neural network and the output is the descriptor vector of the object at *t+1*. This is a kind of object which has a changing feature over time. That is if there are some patterns, like that shown in Fig 5.11, in the past, the next state of can be predicted using this model.

The number of hidden layer and the corresponding number of neurons were determined depending on the output. Thus, the result with those different network parameters is presented in the next chapter.

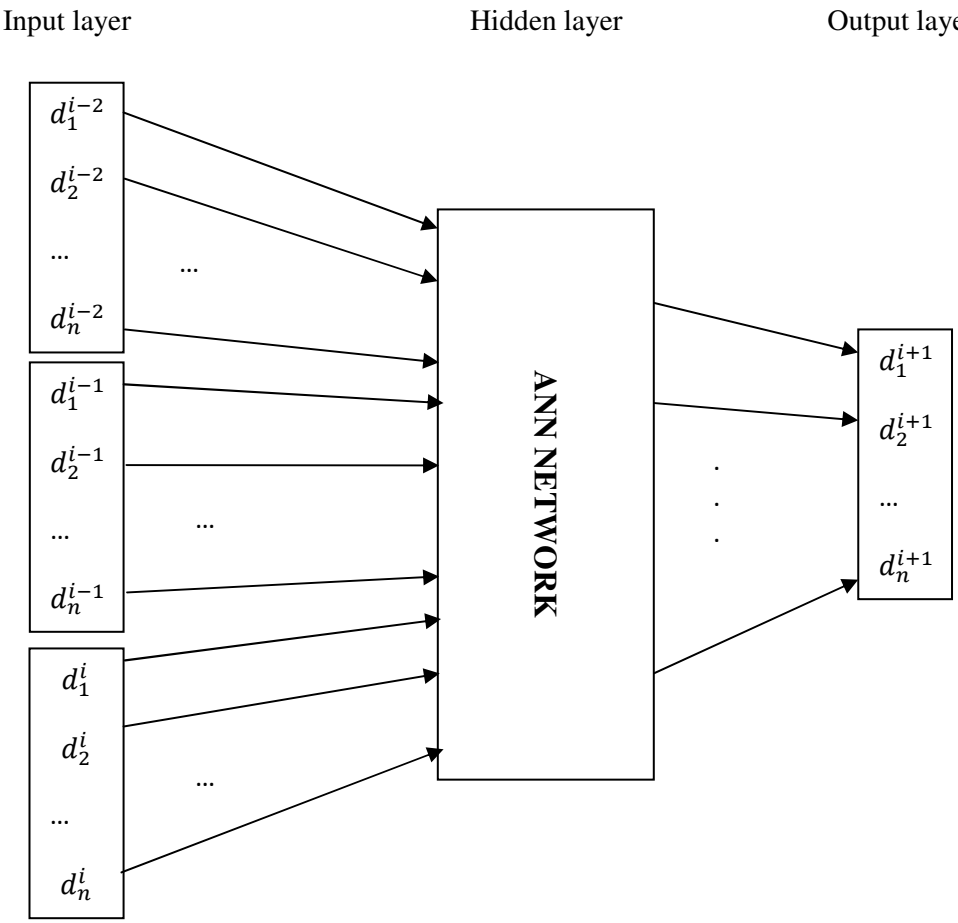


Fig 5.14 A Combined ANN Predicting Model Using Three Time Step Virtual Drought Object [42]

Separate network: This is an approach in which every descriptor of an object in the next time step in the series is predicted separately. Here, every descriptor d_1, d_2, d_3,….., and d_n of the object were determined separately using corresponding descriptors in the time lags. For example, to predict d_1 component of the descriptor vector of the future time, corresponding d_1's in the time lags were used. The descriptors are predicted separately; that is why separate network. The network architecture is shown in Fig 5.15.

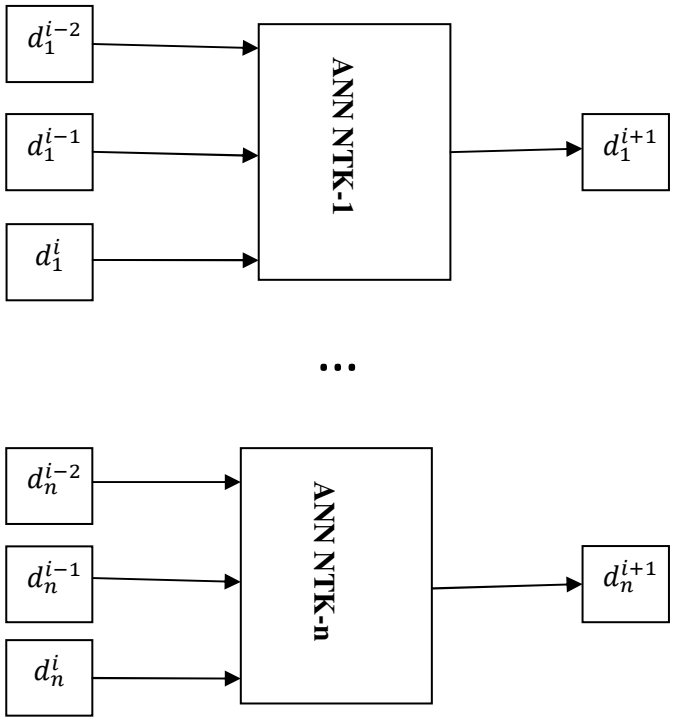


Fig 5.15 A Separate ANN Predicting Models for Three Time Steps [42]

For this special work, provided that there are some past patterns of vegetation condition in a given area, their state in the future can be determined using this model. Depending on the future states predicted, early measures can be taken to alleviate or minimize the consequences during the production season.

Performance metrics

To evaluate the performance of the proposed predictor ANN model; the computational time in terms of number of epochs and a statistical test called Root Mean Square Error (RMSE) was carried out. The following is the mathematical formula for RMSE test [43]:

$$RMSE = \sqrt{(\frac{1}{n}\sum_{t=1}^{n}(actual_t - predicted_t)^2)} \quad (5.5)$$

where n is the number of pixels in the object, $actual_t$ is the actual time series pixel values and $predicted_t$ is the forecasting time series pixel values.

CHAPTER 6

RESULTS AND DISCUSSIONS

The design described in chapter five is implemented using Matlab software. Therefore, the results and analysis of each step towards prediction is presented in this chapter.

6.1 Noise Removal

It is the application of anisotropic diffusion to an image from the AOI selected according to criteria mentioned in section 5.1.1. For example, Fig. 6.1 shows the image before and after denoising for an area selected from the North Eastern part of Ethiopia as shown in Fig 5.3.

a) Original image

b) Image after removing sharp false values

c) Denoised image using anisotropic diffusion

Fig 6.1 Noise Removal from Deviation of NDVI Image

The masking step (step (b)) is required because there were missing values for water bodies like Tana as shown in the original image of Fig 6.1. Thus, pixels with sharp very small values (Eg: multiple of 10^{-38} as observed here) are set to background (equal to zero). This results in the second image in Fig 6.1. Finally, the anisotropic diffusion filter is applied to the image.

The above procedure is repeated for every image in different dekadals of a given study region to accompany for the time series analysis. The series was the growing season of a given year which in this study includes the months of June to September. The sample series of denoised images is shown in Fig 6.2.

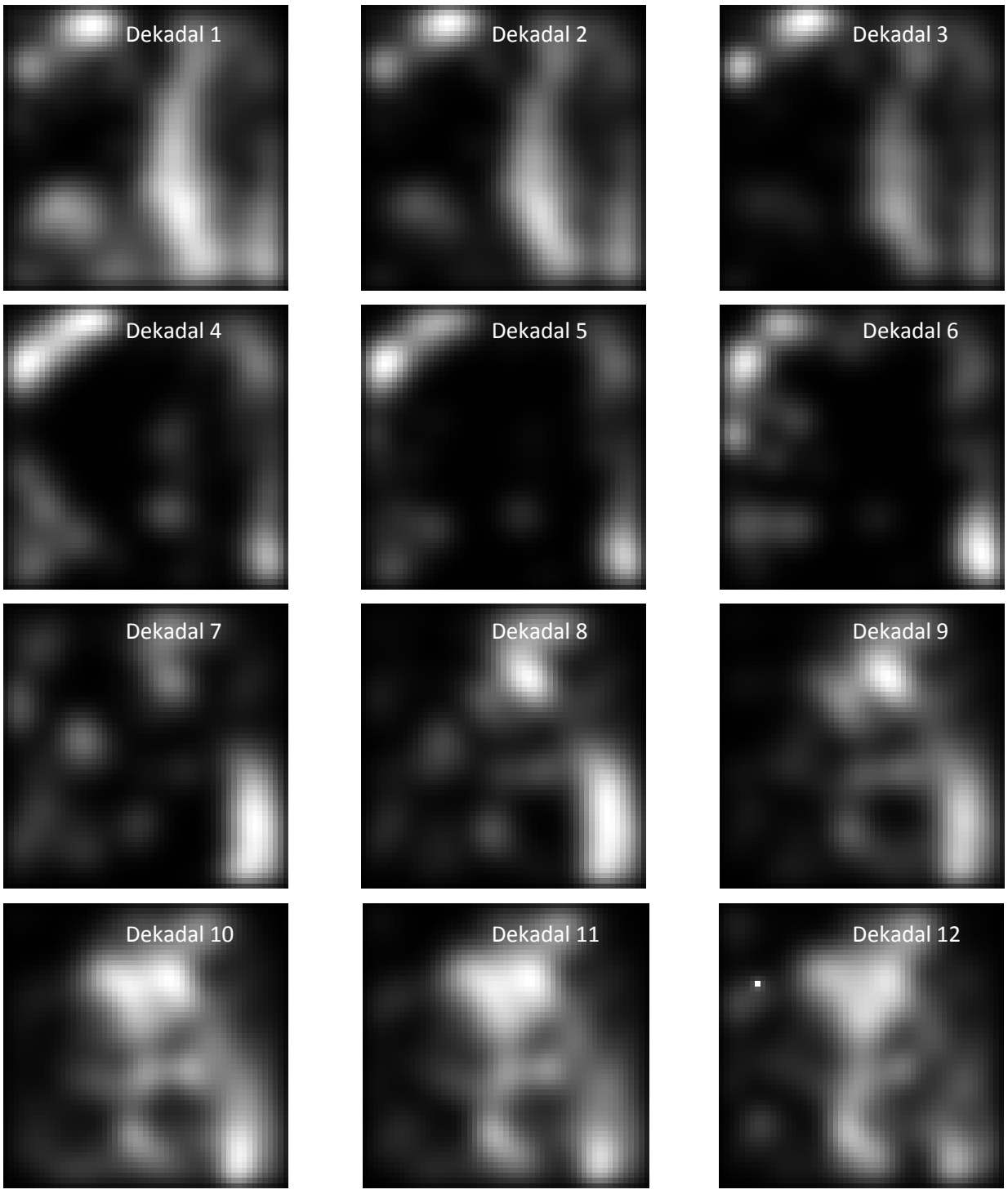


Fig 6.2 Denoising Results on AOI Identified in Fig 5.4

The diffusion filter enhances the contrast and smoothes the image at the same time. In addition, neighboring pixels are connected defining an object (drought, in this case).

6.2 Segmentation

After noise was removed/minimized, segmentation using the k-mean clustering approach was applied. This method requires the number of clusters k as an input. Therefore, the performance metric used was the one evaluated in terms of a simple validity measure based on the intra-class and inter-class distance measures. This evaluation was applied to different images in a given series and most of them yielded a value similar to that shown in table 6.1.

Table 6.1 Intra to Inter Cluster Ratio for Different Number of Clusters

No. clusters	2	3	4	5	6	7	8	9	10
intra/ inter	0.0867	0.0594	0.0416	0.0356	0.0488	0.0382	0.0398	0.0454	0.0461

According to the experimental output of table 6.1, the optimum number of clusters that minimizes the intra-to-inter cluster ratio was found to be five (5). The classification in table 5.1 was based on this number of clusters.

In general, this number of clusters (five) was used for segmentation of an image into different classes of drought objects and background. As a result, the sample segmented images produced to be overlaid on the original images of Fig 6.2 for object extraction are shown in Fig 6.3.

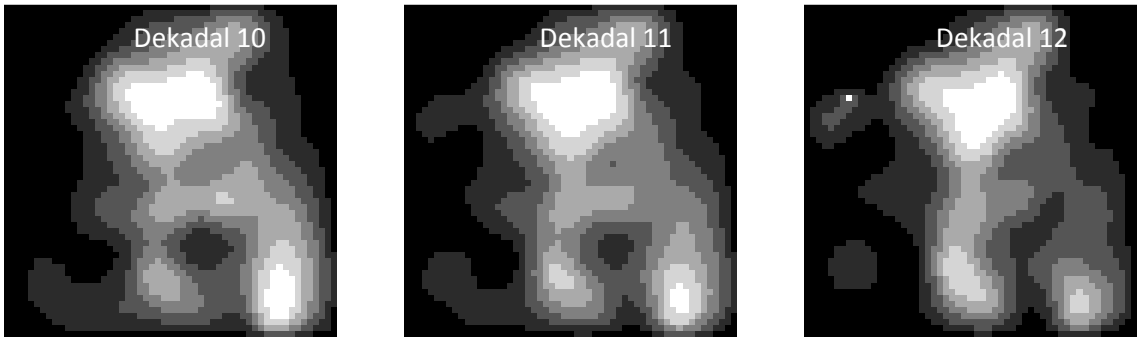


Fig 6.3 Sample Segmentation Results

The segmentation procedure in Fig 6.3 was applied to all the images in a series. Using those uniform regions in the segmentation results, the objects from the original images were extracted. For example, using the region identified by rectangular section in Fig 5.8, the following series of objects can be extracted from the AOI in Fig 6.2 for that specific sub-region.

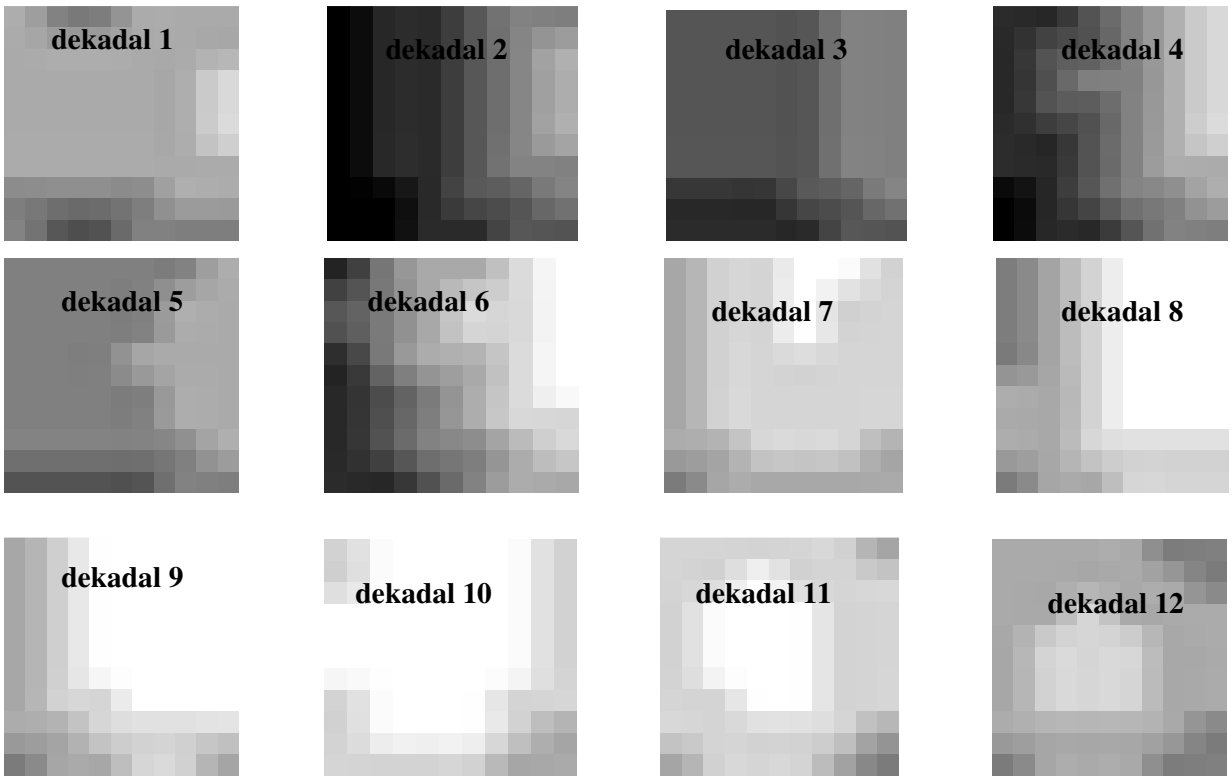


Fig 6.4 Extracted Objects from the Series of Images in Fig 6.2 for Location Shown in Fig 5.9.

The pattern of intensity means of these objects was shown in Fig 5.10 and Fig 5.11. Once the objects were made available, the next step was dimensionality reduction of each object in the series.

6.3 Feature Extraction and Selection

Given a set of pixels defining an object, PCA finds the linear lower-dimensional representation of the object such that the variance of the reconstructed data is preserved. Therefore, instead of the large number of pixels in an object, some components characterizing the object are produced using PCA.

To produce these components, the database of images of drought objects after segmentation was produced for training the PCA. The database was established in such a way that a representative of every drought object from the near normal to the very severe were included. Accordingly, 36 drought objects from different regions were included in the database. Hence, a maximum of 35 PCs can be extracted for every object. For example, the 35 PCs for the first object in Fig 6.4 are shown in table 6.2.

Table 6.2 Sample Descriptors for Drought Object of Dekadal 1 in Fig 6.4

No.	1	2	3	4	5	6	7	8	9	10
PC	405	29.7	20.6	5.8	5.32	3.39	1.99	1.03	0.412	0.325
No.	11	12	13	14	15	16	17	18	19	20
PC	.265	.107	.085	.059	.0469	.045	.0438	.0436	.0265	-.01
No.	21	22	23	24	25	26	27	28	29	30
PC	-.016	-.022	-.193	-.483	-.598	-1.02	-2.39	-2.78	-2.82	-14.3
No.	31	32	33	34	35					
PC	-30.8	-35	-77	-77.2	-130					

But all of these components are not important for defining a given object. As a result, the most PCs were selected using some criterion, which was the image reconstruction error. The reconstruction error on an image is defined as the sum of RMSE of all pixels within the object. The results on reconstruction errors are shown in Fig 6.5, corresponding to using different number of PCs.

The reconstruction error criterion was used in this work for evaluation of the performance of PCA as a tool for dimensionality reduction and feature construction. The measure using reconstruction error was applied to all the images in the series and the image with the worst reconstruction error was used to select the number of PCs.

Most studies show that the principal components found by standard PCA is optimal in terms of reconstruction error minimization [34, 35]. However, this optimality is true only on the training examples, i.e., on the data where the PCA is learned. Thus, the evaluation for selecting optimum number of components was performed on images from training set and outside the training set.

In the Fig 6.5, the reconstruction error was measured on both training images and unseen test images. The sample graph shows that the number of principal components required for optimum reconstruction error is greater for the unseen test image than an image from the training sample.

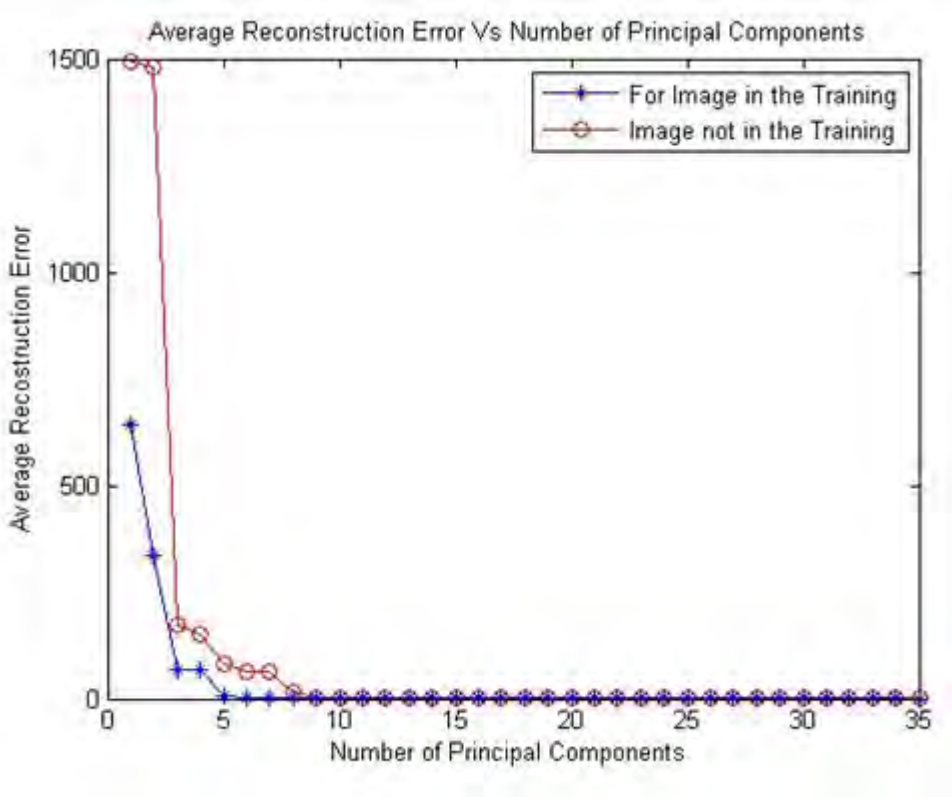


Fig 6.5 Average Reconstruction Error Versus Number of Principal Components for Sample Images from Training Set and Outside the Training Set.

Finally, 15 principal components were selected with RMSE reconstruction error of 0.1 and 0.9 for images from training set and for images outside the training set, respectively. The sample reconstructed images with this number of components are shown in Fig 6.6 and Fig 6.7, with the numbers on the axes indicating pixel coordinates.

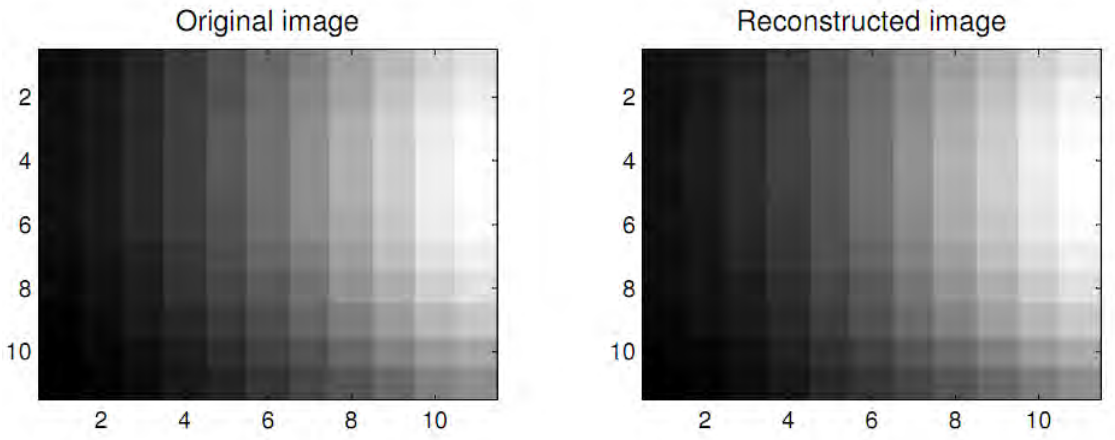


Fig 6.6 Sample Reconstruction for an Image from the PCA Training Database (PC=15)

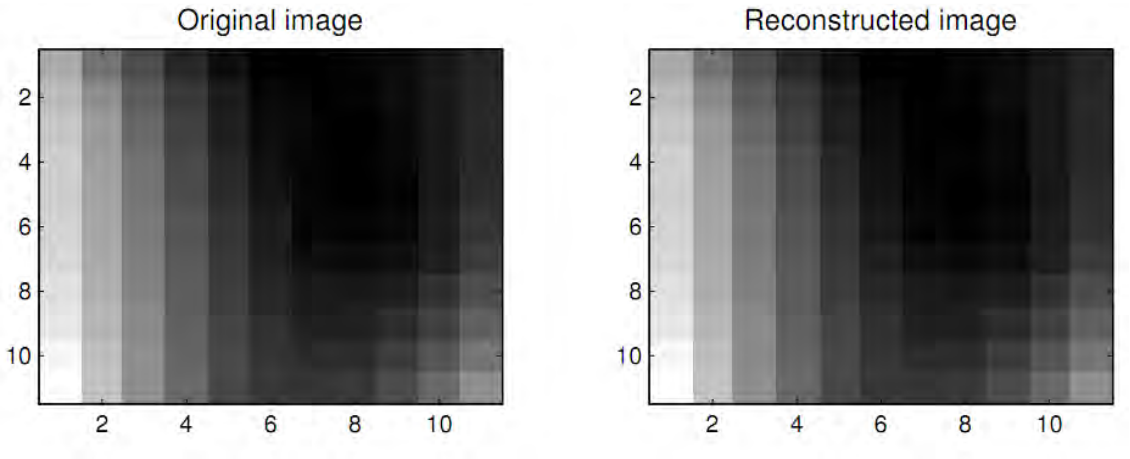


Fig 6.7 Sample Reconstruction for an Image Outside the PCA Training Database (PC=15)

The descriptors of objects used in the training and testing set with 15 principal components are shown in Appendix B.

6.4 Prediction

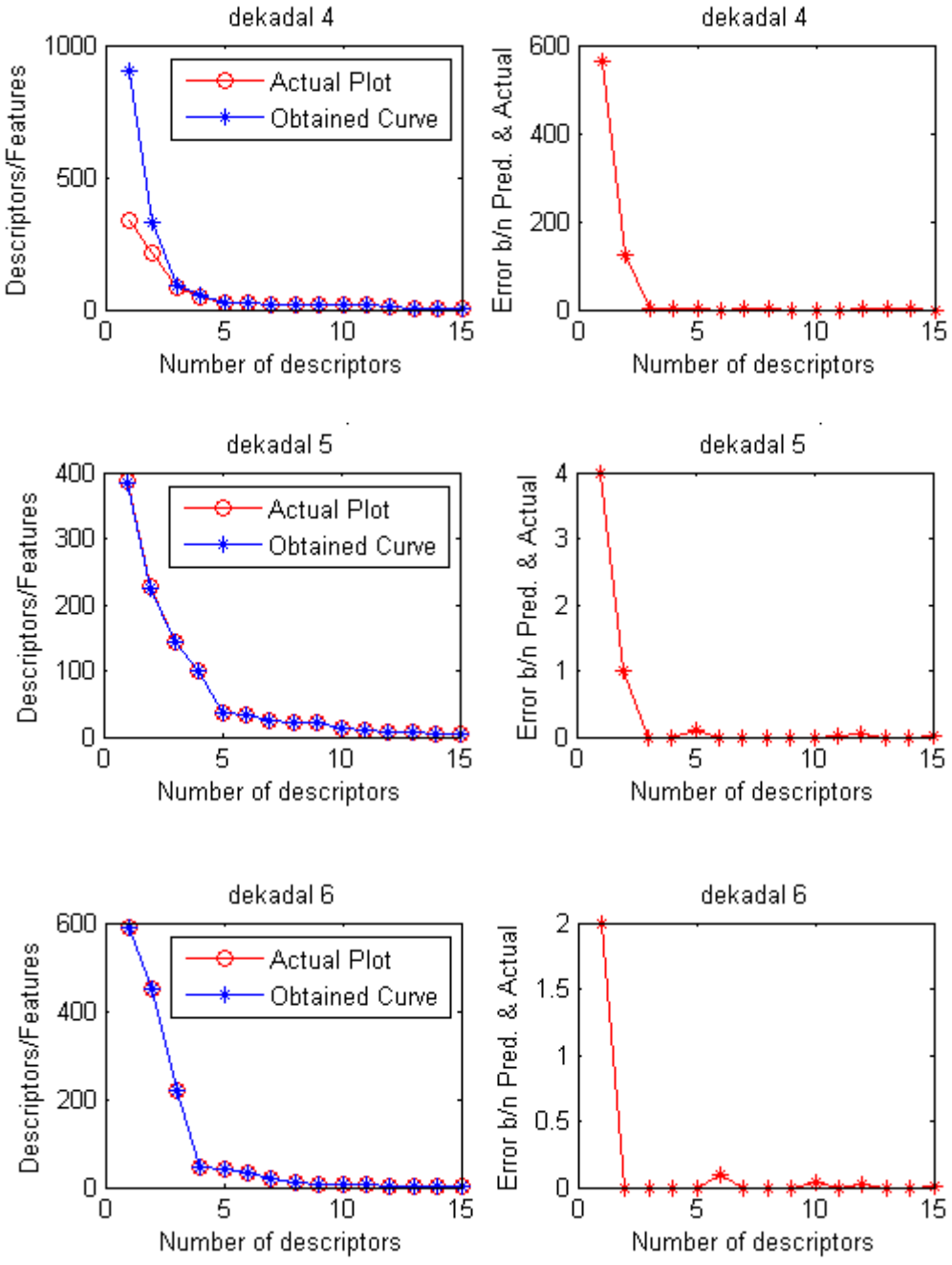
Prediction was performed using the two network approaches described in section 5.4. Thus, the description and performance analysis of each network approach is given in detail as follows.

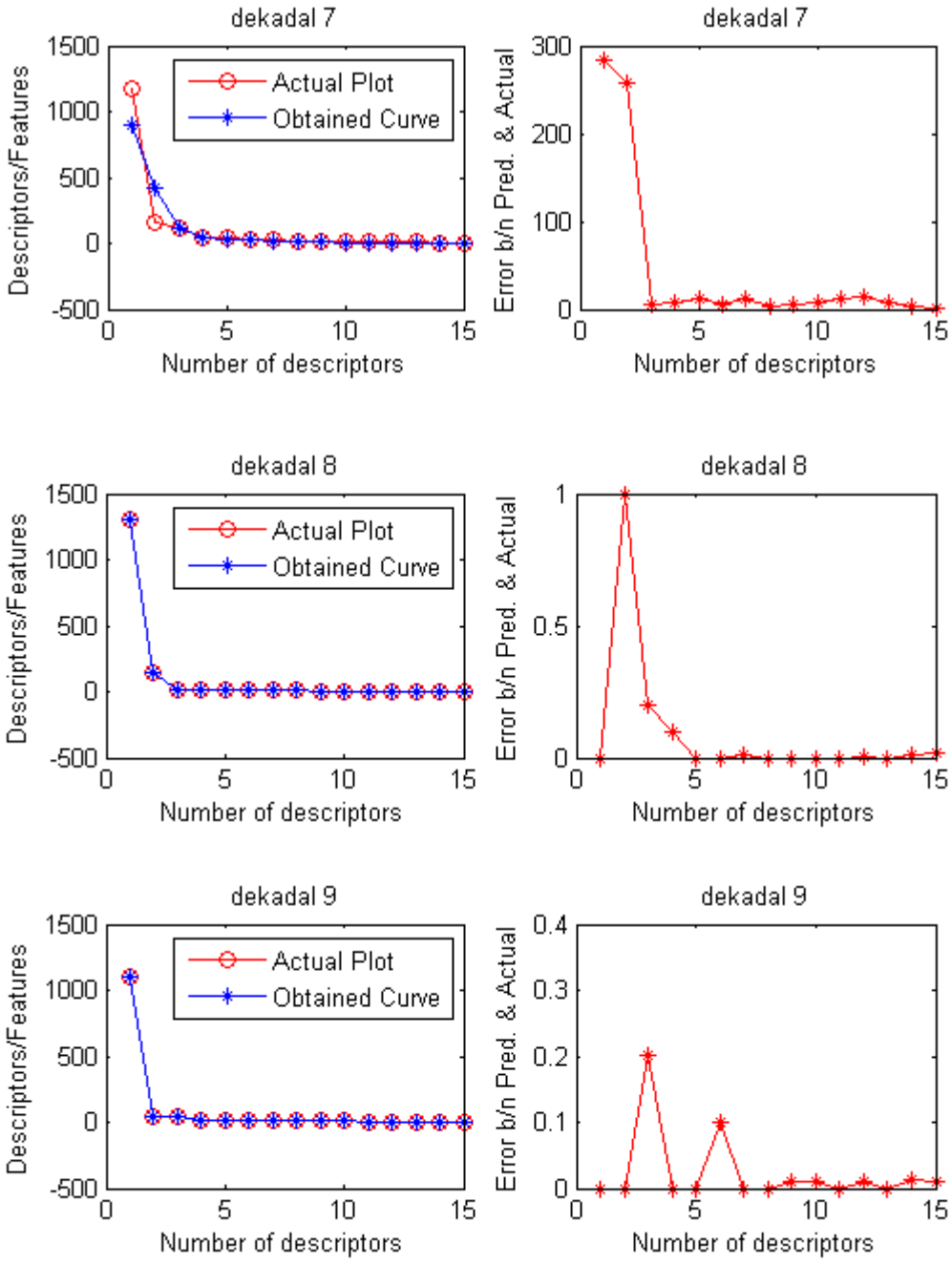
a) Combined Network Approach

With number of descriptors selected according to the discussion in section 6.3, the network structure in Fig 5.12 was set up. The series of objects from different locations were divided in to training and testing set. Out of 9 series, 7 of them were used for training and the rest for testing. After the network was trained with the training set, the performance of the network was evaluated using the data in the testing set in terms of RMSE.

The descriptor vectors shown in the tables of section B.3 of appendix B were used as inputs to the ANN. The 15 descriptors for an object in the next time step of three and four time lags were predicted all at a time. For example, in the three lag network, the descriptors of the first three objects in the series were used to predict the 15 descriptors of the fourth object. Hence, there were 45 input nodes and 15 output nodes in the ANN structure. Fig 6.8 and 6.9 shows the result of prediction for each object using three and four time steps, respectively, in terms of the descriptors using this network. The descriptors are shown being sorted in decreasing order. For

clear view of the difference between predicted and the actual descriptor values, the graph of the deviation of the predicted values of each component is shown to the right of every prediction result. The deviation graph is simply the absolute difference between the predicted and actual descriptor values.




dekadal 7
Descriptors/Features
Actual Plot
Obtained Curve
Number of descriptors
dekadal 7
Error b/n Pred. & Actual
Number of descriptors
dekadal 8
Descriptors/Features
Actual Plot
Obtained Curve
Number of descriptors
dekadal 8
Error b/n Pred. & Actual
Number of descriptors
dekadal 9
Descriptors/Features
Actual Plot
Obtained Curve
Number of descriptors
dekadal 9
Error b/n Pred. & Actual
Number of descriptors

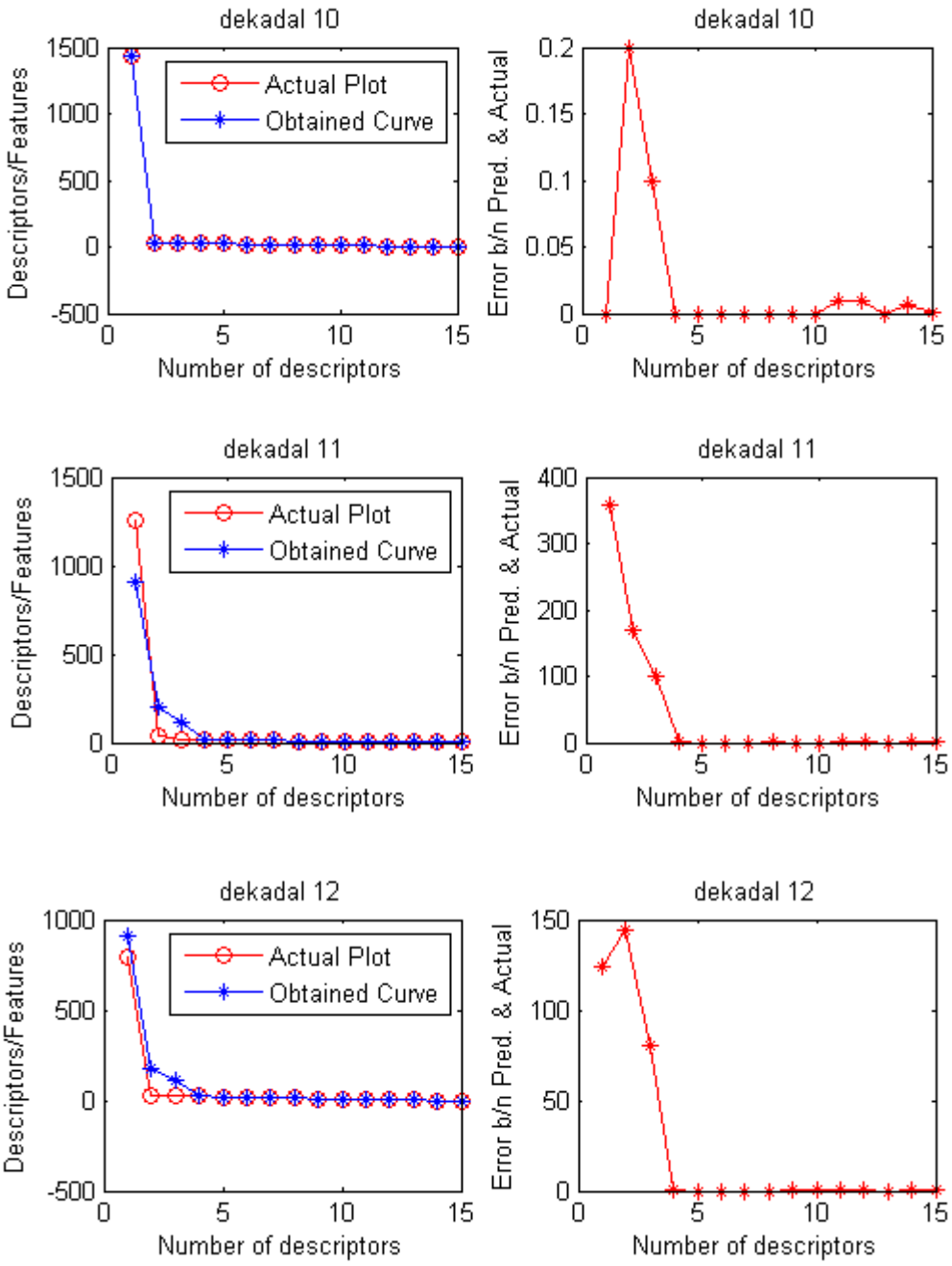
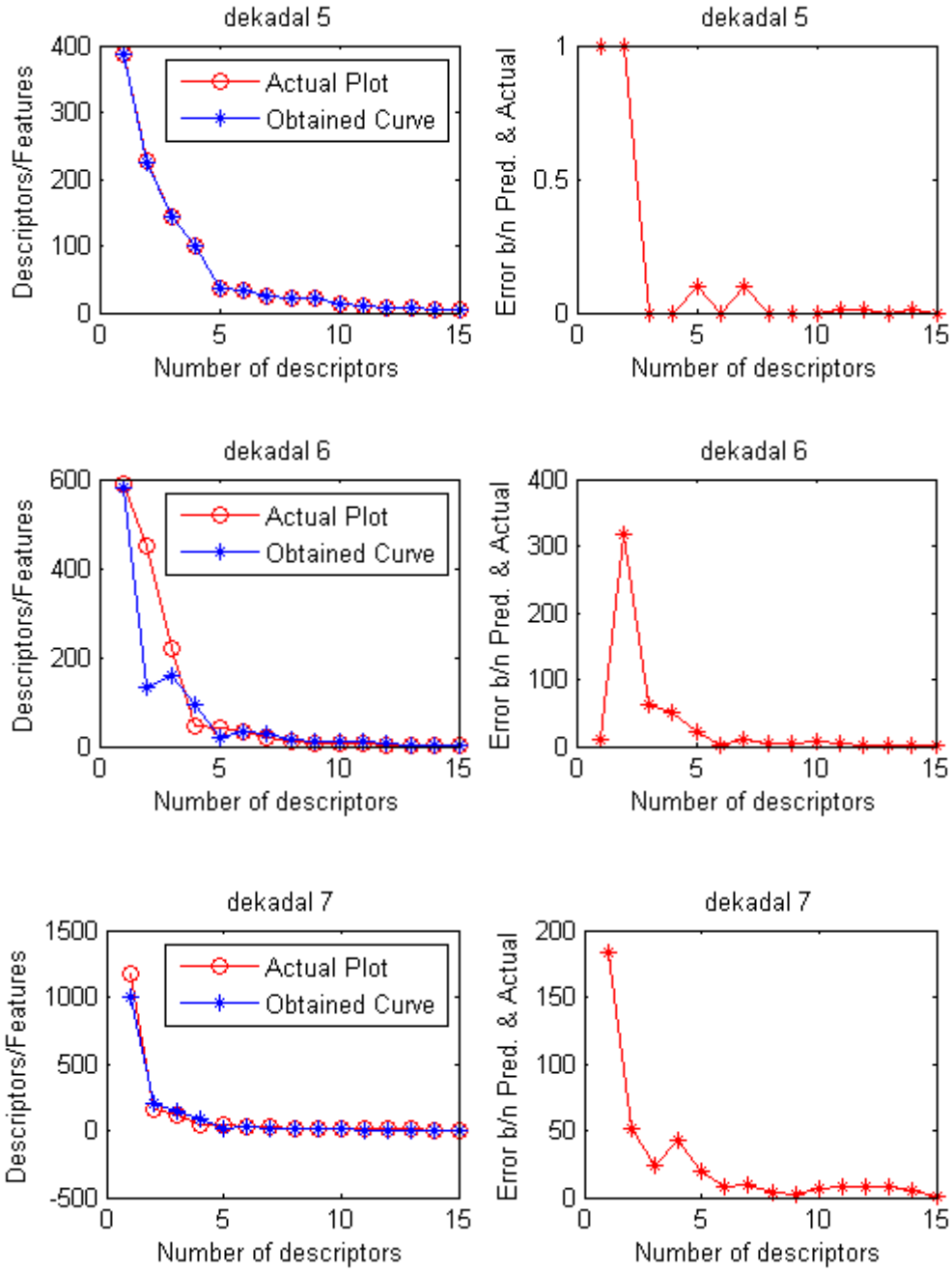
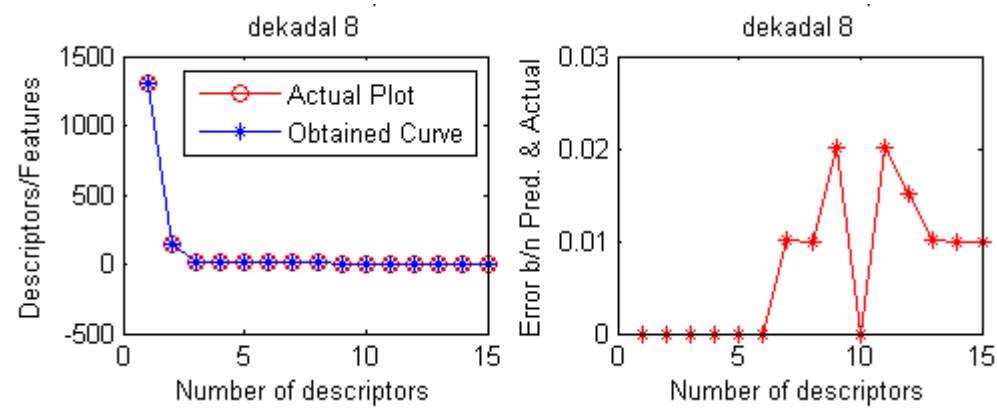
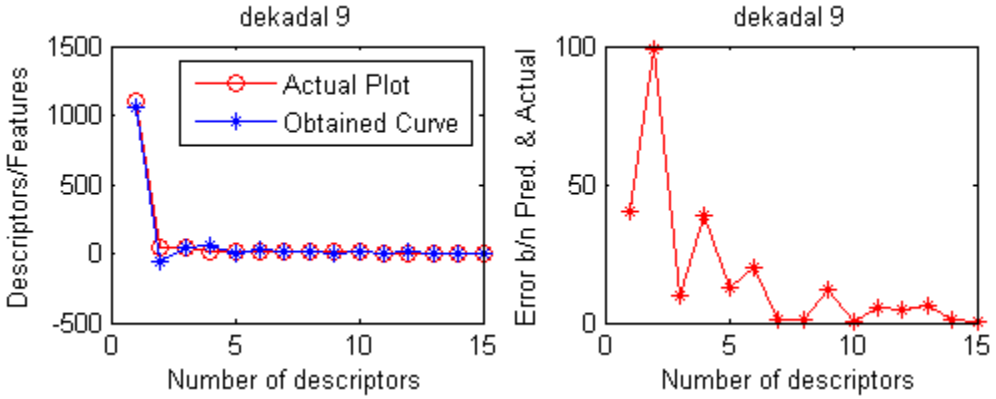


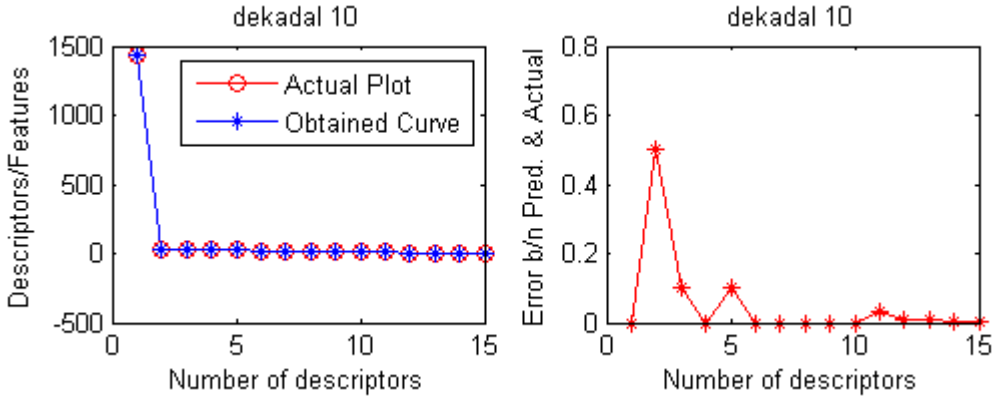
Fig 6.8 Actual & Predicted Values of Descriptors for Each Dekadal for Three Time Lags Using the Combined Network Approach

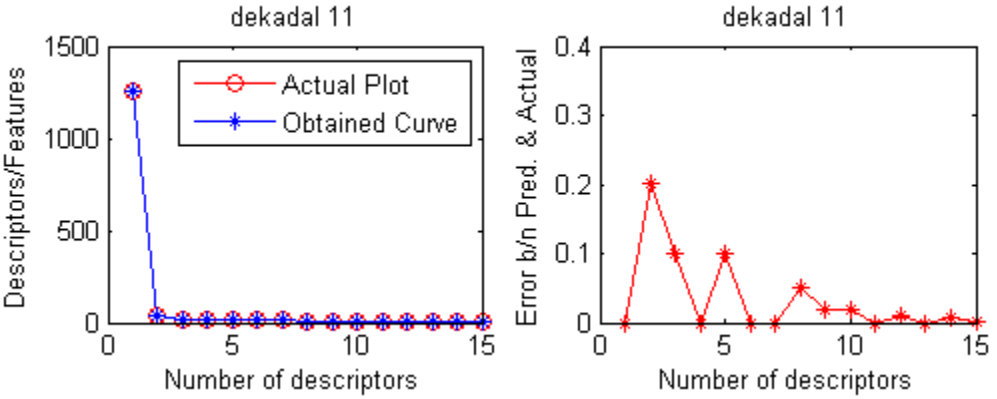
The following figures show the prediction results of the combined network approach using the four time lag. As a result, of the number of lags the prediction starts from the 5th dekadal.




dekadal 8
Actual Plot
Obtained Curve
Descriptors/Features
Number of descriptors
dekadal 8
Error b/n Pred. & Actual
Number of descriptors


dekadal 9
Actual Plot
Obtained Curve
Descriptors/Features
Number of descriptors
dekadal 9
Error b/n Pred. & Actual
Number of descriptors


dekadal 10
Actual Plot
Obtained Curve
Descriptors/Features
Number of descriptors
dekadal 10
Error b/n Pred. & Actual
Number of descriptors


dekadal 11
Actual Plot
Obtained Curve
Descriptors/Features
Number of descriptors
dekadal 11
Error b/n Pred. & Actual
Number of descriptors

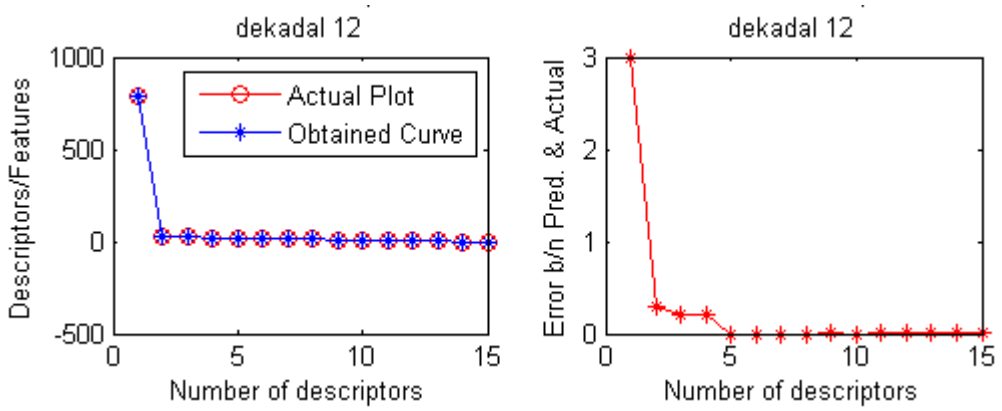


Fig 6.9 Actual and Predicted Values of Descriptors for Each Dekadal for Four Time Lags Using Combined Network Approach

The prediction was made using every three and four dekadals in the series to predict the next object's descriptors. Thus, from June to September, we have the prediction of dekadals 4 to 12 and 5 to 12 for three and four time lags, respectively. Therefore, there were overlaps in the input descriptors for prediction of the next dekadal in the time series.

Both time lags produced reasonably good results though with four time lags the prediction yielded slightly better overall accuracy. In addition, the network architecture was found to affect the prediction results. In general, the overall performance with different network parameters and time lags are summarized in table 6.3.

Table 6.3 Performance Analysis of Different Parameters for Prediction

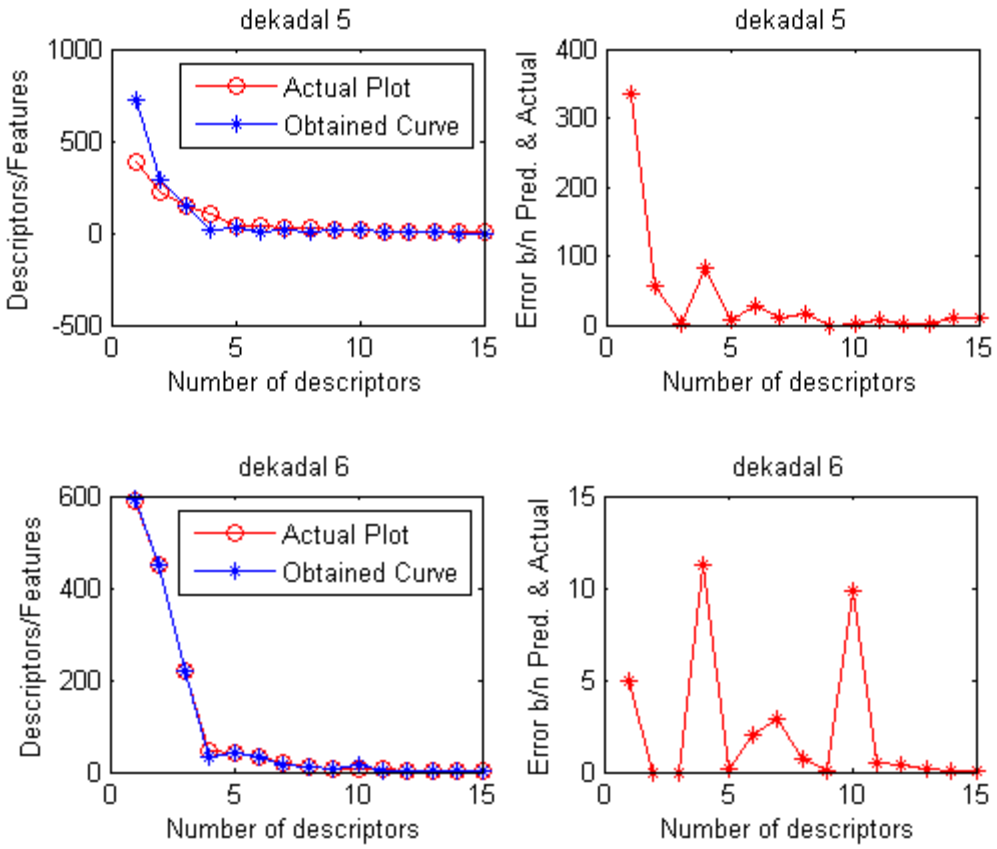
No. of time lags	ANN Architecture			No. of epochs	Learning Rate	RMSE
	Input	Hidden	Output			
3	45	47	15	500	0.01	15.8610
	45	55	15	1200	0.02	20.8010
4	60	45	15	1711	0.01	16.4277
	60	55	15	1923	0.02	14.5863

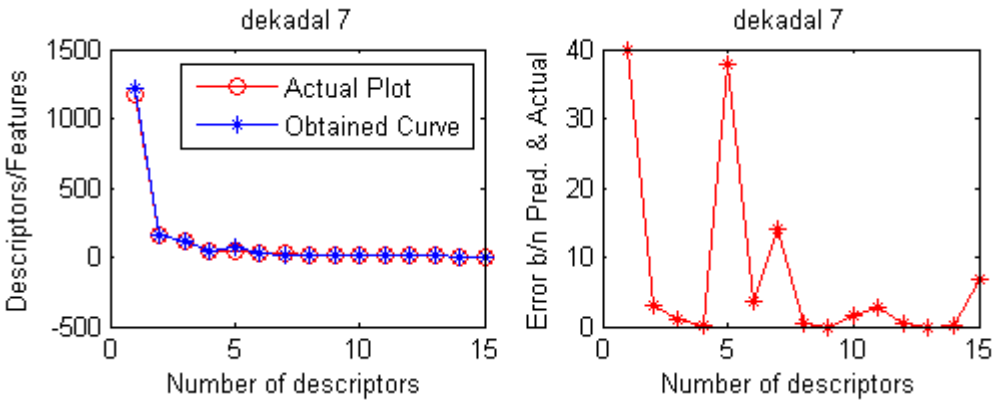
The ANN parameters and the number of lags described in table 6.3 were because of their better result compared to others. Furthermore, the number of lags was found to be three and four as lower number of lags like two didn't produce promising output. And for lags of above four the

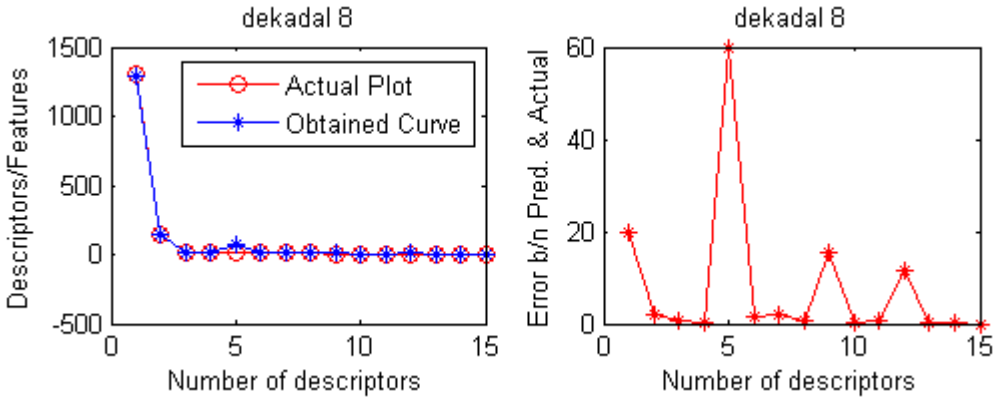
prediction will be so late to take early precautions within the growing season. The results in the above table were generated using stopping criteria of 5000 epochs or maximum allowable RMSE of 0.001 for the normalized descriptors during training of the ANN. As shown in the table lower RMSE was produced with 4-lag, 55 hidden layer neurons and learning rate of 0.02; though it was at a cost of computation time.

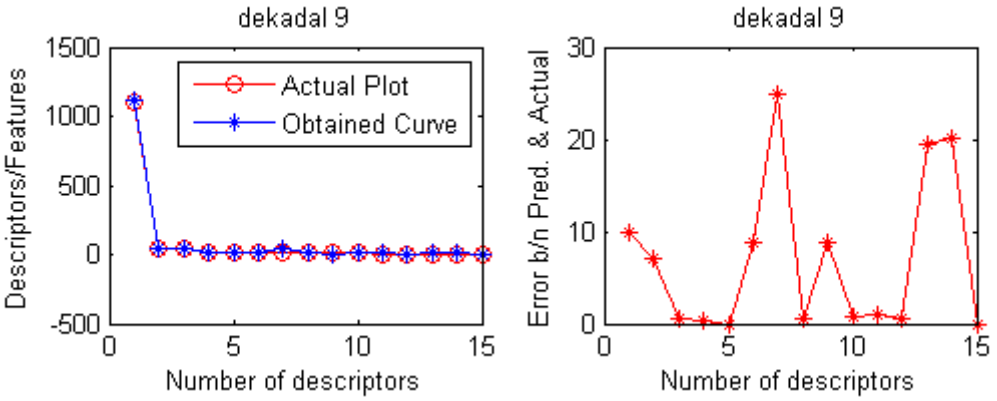
b) Separate Network Approach

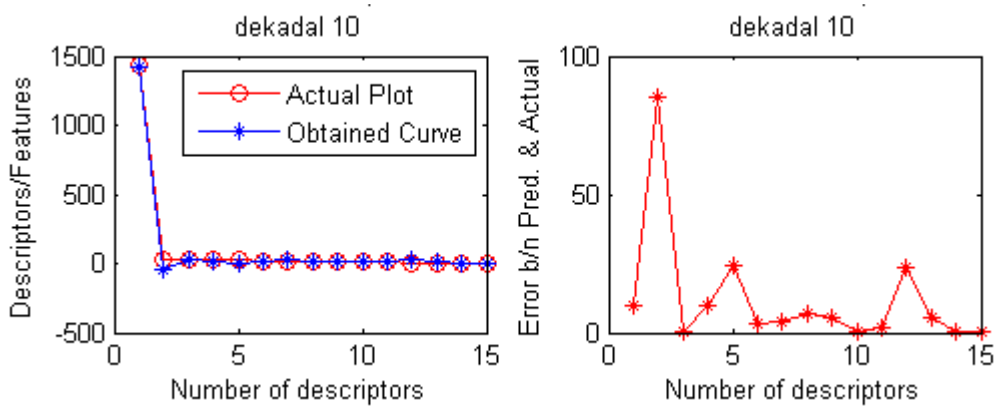
In this case a separate network was set for each of the 15 components in the descriptor vector of an object in the time series. Thus, prediction was made component wise. For instance, the first principal components of every four objects in the time lag were used to predict the first principal component of the next object in the series and, so on. Finally, the components were combined to make up the descriptor of an object predicted. Different network parameters were used and the prediction using four time lags was found to yield better result. The outputs of prediction using this four lag network for the descriptors of objects in the series are shown in Fig 6.10.




dekadal 7
Actual Plot
Obtained Curve
Descriptors/Features
Number of descriptors
dekadal 7
Error b/n Pred. & Actual
Number of descriptors


dekadal 8
Actual Plot
Obtained Curve
Descriptors/Features
Number of descriptors
dekadal 8
Error b/n Pred. & Actual
Number of descriptors


dekadal 9
Actual Plot
Obtained Curve
Descriptors/Features
Number of descriptors
dekadal 9
Error b/n Pred. & Actual
Number of descriptors


dekadal 10
Actual Plot
Obtained Curve
Descriptors/Features
Number of descriptors
dekadal 10
Error b/n Pred. & Actual
Number of descriptors

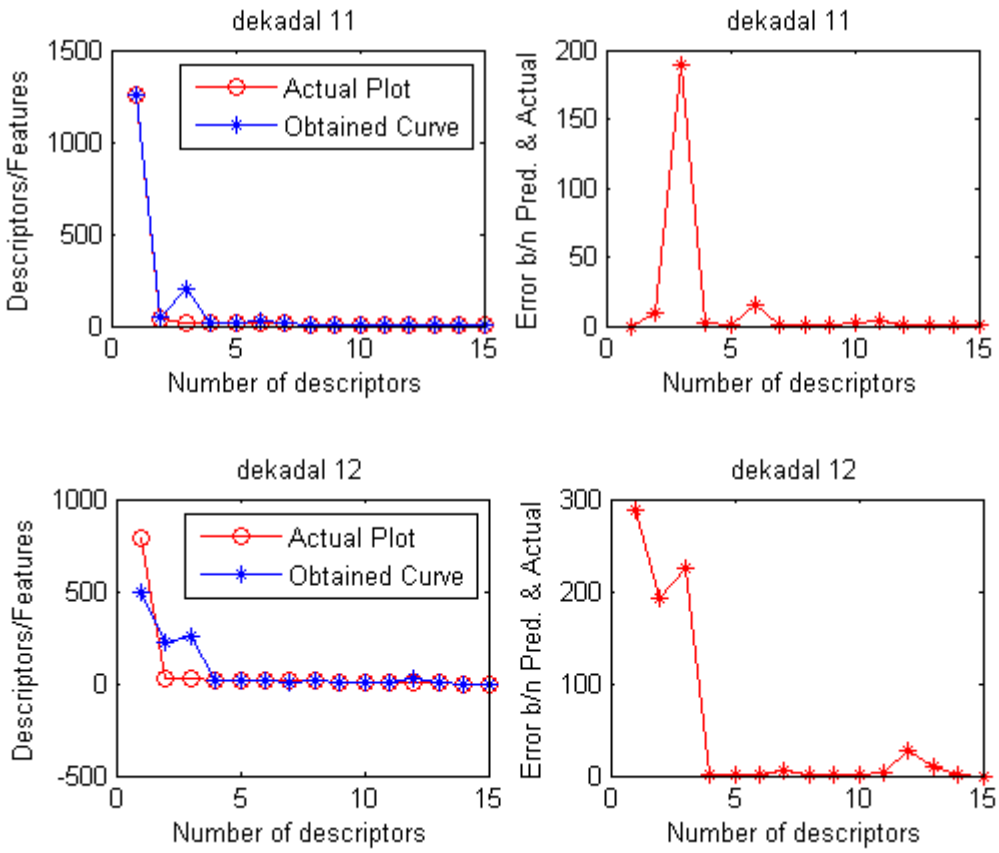


Fig 6.10 Actual and Predicted Values of Descriptors for Four Time Lags Using Separate Network Approach

The prediction results were fine but not as good as that of combined network approach. This is because of the fact that separate network approach doesn't incorporate the dependences of components with in an object. In addition, the overall computational efficiency is lower and RMSE error of the final object's descriptors was higher compared to the first approach. The performance of the network with best results in this approach is shown in the table 6.4.

Table 6.4 Performance Analysis of Separate ANN

ANN Architecture			Average no. of Epochs	RMSE
Input	Hidden	Output		
4	3	1	5000	36.9175

As the prediction provided in here was in terms of PCs, post processing is required to recover the objects and show the time series prediction and its actual value on the same axis.

Post-processing: The PCA eigendroughts are used to reconstruct the drought objects from the predicted descriptors. Fig 6.11 and Fig 6.12 shows the actual versus predicted values for the 5th and 10th dekadals, respectively, after reconstruction to manifest the difference.

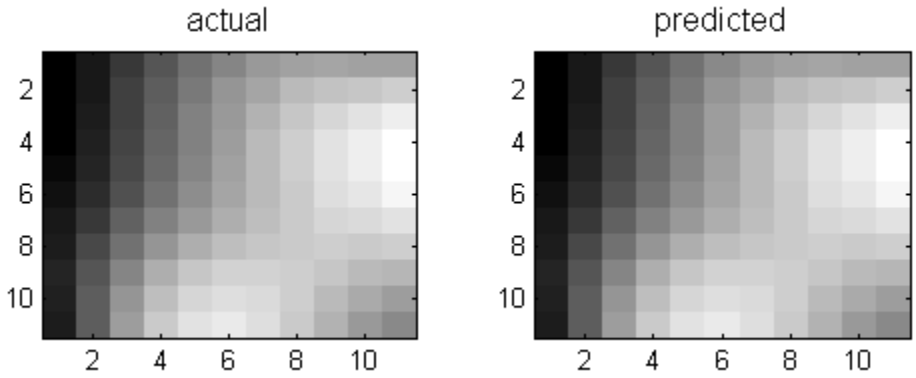


Fig 6.11 Sample Reconstruction of an Object for 5th Dekadal

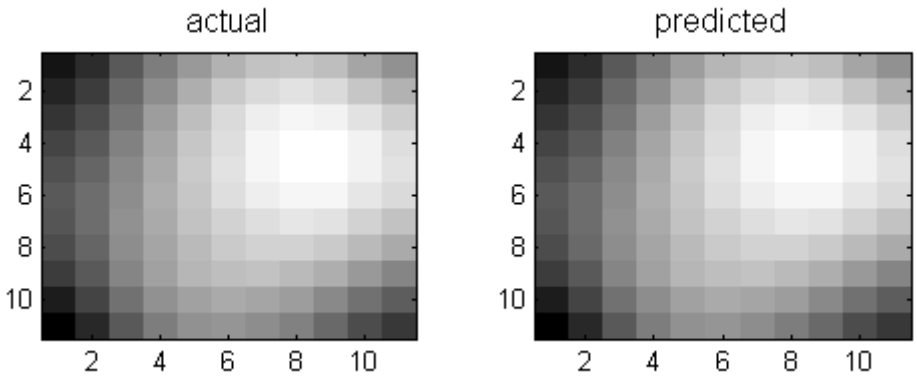


Fig 6.12 Sample Reconstructed Object for the 10th Dekadal

In general, the predicted drought objects for the best network were reconstructed. And the mean value of intensity of the drought objects in the test set after prediction over time from 5th to 12th dekadals is shown in Fig 6.13.

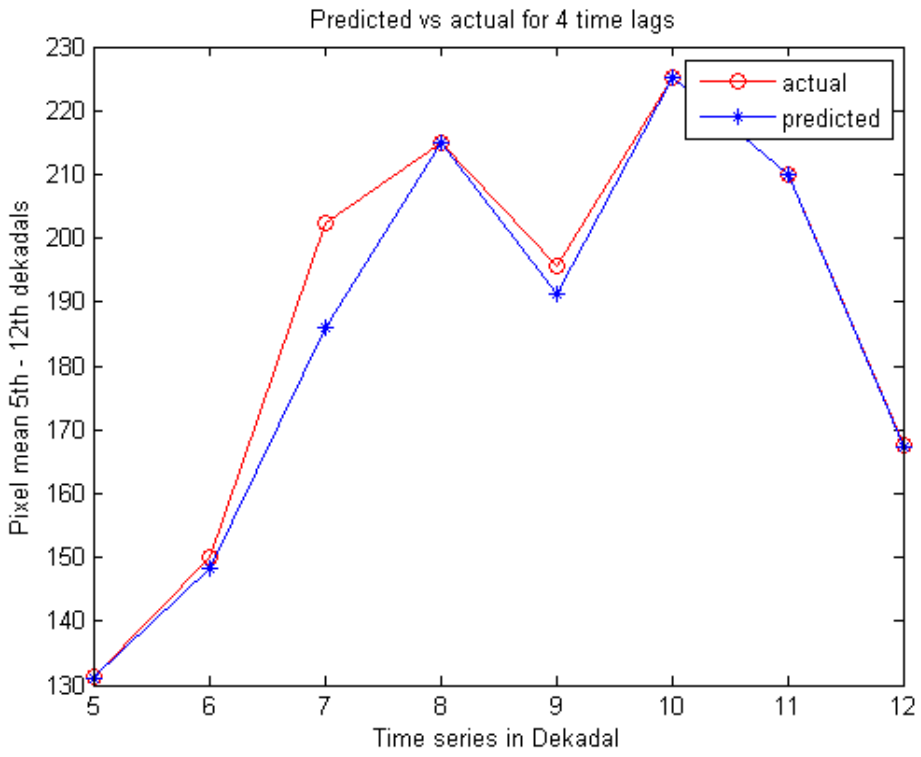


Fig 6.13 Actual and Predicted Intensity Means of Drought Objects for Four Lag

The graph in Fig 6.13 is for the best network with four time lags. Thus, the first four objects (Jun 10 to July 10) were not included.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusion

Developments in remote sensing image analysis for drought monitoring for early warning are an ongoing research area. Most of the works so far have been based on the classification of the image region in to drought and non- drought categories at pixel level. And prediction of the probable decrease in crop yield during harvesting is based on this classification results. Though there are certain patterns that drought occurrences could manifest, as shown in this work, there were little considerations. Furthermore, this thesis also gave an insight to the necessity of time series analysis to take required action early while within the growing season itself.

Therefore, the time series analysis of satellite images for prediction using ANN produced promising results. Given, a good preprocessing and feature extraction steps, the neural model for prediction from four time lag points yielded a lower overall RMSE at cost of increase in convergence time. As prediction accuracy was given a priority in this work, the combined model approach with one hidden layer of 55 neurons was selected as predictor.

The system was experimentally analyzed with 108 virtual drought objects making up 9 time series for the identified growing season of June to September. Of these series, around 78% were used for training and the rest for testing. After the extraction of descriptor vectors of each object in the series, the prediction of features of the object at the next time step was performed using both combined and separate ANN models. As a result, the combined approach presented a higher prediction accuracy compared to the separate network approach. Especially, the three and four time lag combined ANN prediction model produced an overall RMSE of 20.80 and 16.43, respectively, which was a better result compared to a 36.92 RMSE for separate ANN approach. It is understood that this work will give new views for ways in drought prediction for early warning and crop condition monitoring at near real-time.

7.2 Recommendations

This work introduced the idea of object level analysis of drought from an image. But detail research work is required to group pixels to an object based on the inherent nature of the crop. Thus, the segmentation approach will include the study results of this kind.

Furthermore, this study was based on limited historical image data for drought occurrences. As a result, the forecasting was based only on the patterns that manifested during those times. But for accurate prediction, the model should incorporate patterns of drought obtained from detail investigation of its historical incidences with different severity levels at different times. In addition, the combination of other parameters like amount of rain fall, duration of sunshine, etc., with the NDVI images is essential for more reliable prediction. Hence, future studies should put this fact in to considerations.

More importantly, the time series data used in this work was for ten day intervals, which makes the series coarse. Hence, decreasing the series interval will give smooth data and will even help the prediction to be with in fewer days.

REFERENCES

[1] Rulinda, C.M., Bijker, W., and Stein, A., "Image Mining for Drought Monitoring in Eastern Africa Using Meteosat SEVIRI data." International Journal Appl. Earth Observ. Geoinform., 2009.

[2] Virat Shukla, "Modelling Spatio-Temporal Pattern of Drought using Three-Dimensional Markov Random Field", M.Sc. thesis, International Institute for Gio-information Science and Earth Observation, the Netherlands, January 2007.

[3] Thiruvengadachari, S., Gopalkrishana, H.R., "An integrated PC Environment for the Assessment of Drought", International Journal of Remote Sensing, 14: 3201 – 3208, 1993.

[4] Aditi Sharma, "Spatial Data Mining for Drought Monitoring: An Approach Using temporal NDVI and Rainfall Relationship", M.Sc thesis, International Institute for Gio-information Science and Earth Observation, The Netherlands, March 2007.

[5] FEWSNET, "United Nations' FEWS (Famine Early Warning Systems) program", http://www.fews.net/Pages/default.aspx?l=en, verified on 8 December 2009.

[6] T. Blaschke, "Object Based Image Analysis for Remote Sensing", ISPRS Journal of Photogrammetry and Remote Sensing, 2009.

[7] Huang, B., and Jiang, B., Li, H. "An Integration of GIS, Virtual Reality and the Internet for Visualization, Analysis and Exploration of Spatial Data", International Journal of Geographical Information Science, 15 (5), pp. 439 – 456, 2001.

[8] Wilhite, D. A., "Drought as a natural hazard: Concepts and Definitions, Drought: A Global Assessment", D.A. Wilhite, Ed., Vol. 1, Routledge, 3–18, 2000.

[9] Hayes, M., M. Svoboda, D. A. Wilhite, and O. Vanyarkho, "Monitoring the 1996 drought using the Standardized Precipitation Index", Bull. Amer. Meteor. Soc., 80, 429–438, 1999

[10] Tadesse, T., Brown, J.F., and Hayes, M.J.H., "A New Approach for Predicting Drought Related Vegetation Stress: Integrating Satellite, Climate, and Biophysical Data Over the U.S. Central Plains", ISPRS Journal of Photogrammetry and Remote Sensing 59 (4), 244–253, 2005.

[11] Musaningabe Rulinda Coco, "Mining Drought from Remote Sensing Images", M.Sc thesis, International Institute for Gio-information Science and Earth Observation, The Netherlands, March 2007

[12] Legesse Hadush et al., "Drought Risk Assessing Using Remote Sensing and GIS: A Case Study in Southern Zones, Tigray Region, Ethiopia", M.Sc. thesis, GIS of Addis Ababa University, Ethiopia, June 2010.

[13] Wilhite, D.A. & Sovoda, M.D., "Drought Early Warning System in the Context of Drought Preparedness and Mitigation", Paper Presented at the Early Warning Systems for Drought Preparedness and Drought Management, Lisbon, Portugal, 2000.

[14] Boken, V.K., Cracknell, A,.P., & Heathcote, R.L., "Monitoring and Predicting Agricultural Drought: A Global Study", Oxford University Press, 2005.

[15] Siqi ding, "Predicting Dynamics of Vegetative Drought Classes Using Fuzzy Markov Chains", M.Sc. thesis, Geo Information Science and Earth Observation, Enschede, The Netherlands, March 2011

[16] Nagarajan, R., "Drought Assessment", pp 23-45, Springer, 2009

[17] S. Neiyemer, "New Drought Indices", European Commission, DG Joint Research Center, Institute of Environment and Sustainability, Italy.

[18] Bayarjargal, Y., Karniel, A., Bayasgalan, M.Khudulmar, S.,Ganush, C. and Tucker, C.J., "A Comparative Study of NOAA-AVHRR Derived Drought Indices Using Change Vector Analysis", Remote Sens. Environ., 105:9-22, 2006.

[19] Tucker, C.J. and Choudhary, B.J., "Satellite Remote Sensing of Drought Conditions",Remote Sens. Environ., 23:243-251, 1987.

[20] Peters, A.J., Walter Shea, E.A., Vina, A, Hayes, M.J., and Svoboda, M.D., "Drought Monitoring with NDVI-Based Standardized Vegetation Index", Photogramm. Eng. Remote Sens., 68:71-75, 2000.

[21] Bajgiran, P.R., Darvishsefat, A. A., Khalili, A., & Makhdoum, M. F., "Using AVHRR-Based Vegetation Indices for Drought Monitoring in the Northwest of Iran" Journal of Arid Environments, 72:1086-1096, 2008.

[22] Begumkultur, "Pre-Processing in Image Analysis of Satellite Pictures", power point presentation, http://web.itu.edu.tr/~denizali/remote_sensing, Istanbul Teknik Universitesi

[23] Coppin, P., and M. Bauer, "Digital Change Detection in Forest Ecosystems with Remote Sensing Imagery", Remote Sensing Reviews, 13:207–234, 1996.

[24] S. K. Weeratunga and C. Kamath, "PDE-based Non-linear Diffusion Techniques for Denoising Scientific and Industrial Images: an Empirical Study", Image Processing: Algorithms and Systems Conference , December 1, 2001

[25] G.Padmavathi et al., "Performance Analysis of Non Linear Filtering Algorithms for Underwater Images", Sensors, Signals, Visualization, Imaging, Simulation and Materials, pp 179-184.

[26] Dr.G.Padmavathi et al., "Comparison of Filters used for Underwater Image Pre-Processing", January 2010.

[27] Jiying Yuan and Guojin He., "Application of an Anisotropic Diffusion Based Preprocessing Filtering Algorithm for High Resolution Remote Sensing Image Segmentation", Congress on Image and Signal Processing, 2008.

[28] Perona P., Malik J., "Scale-Space and Edge Detection Using Anisotropic Diffusion", IEEE Trans. Pattern Anal. Machine Intell., 1990, 12:629-639

[29] Leen-Kiat Soh and Costas Tsatsoulisy "Segmentation of Satellite Imagery of Natural Scenes Using Data Mining", IEEE Trans. On Geoscience and Remote Sensing, Vol. 37, No. 2, March 1999.

[30] Vincenzo BARRILE et al., "Object-Oriented Analysis Applied to High Resolution Satellite Data ", WSEAS trans. on Signal Processing.

[31] Umamaheshwaran Rajasekar, "Image Mining within Meteosat Data: A Case Study Of Modeling Forest Fire", M.Sc. Thesis, International Institute for Geo-information Science and Earth Observation, Enschede, The Netherlands, March 2005.

[32] Ujjwal Maulik and Indrajit Saha, "Differential Evolution Based Fuzzy Clustering Technique: Application to Satellite Image Segmentation", International Conference on Systemics, Cybernetics and Informatics, 2009, West Bengal, India.

[33] Dr.G.Padmavathi and Mr.Muthukumar, "Image Segmentation Using Fuzzy C-Means Clustering Method with Thresholding for Underwater Images", Int. J. Advanced Networking and Applications, Volume: 02, Issue: 02, Pages:514-518, 2010.

[34] Charles Bouveyron, Stéphane Girard and Cordelia Schmid., "Dimension Reduction and Classification Methods for Object Recognition in Vision", 5th French-Danish Workshop on Spatial Statistics and Image Analysis in Biology, 10-13 may 2004, Saint-pierre de Chartreuse - France.

[35] Aleix M. Martinez et al., "PCA versus LDA", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.23, Feb 2001

[36] P.E. Robinson et al., "Comparison of Principal Component Analysis and Linear Discriminant Analysis for Face Recognition", IEEE trans., March 2007

[37] A.K. Mishra and V.R. Desai, "Drought Forecasting Using Feed-Forward Recursive Neural Network", Department of Civil Engineering, Indian Institute of Technology, Kharagpur 721302, India.

[38] Tae-Woong Kim and Juan B. Valdés, "A Nonlinear Model for Drought Forecasting Based on Conjunction of Wavelet Transforms and Neural Networks", ASCE, Journal of Hydrologic Engineering, 2003.

[39] Granger, C.W.J., Newbold, P., "Forecasting Economic Time Series", Academic Press 2nd ed., 1986.

[40] Md. Khairul Hasan et al., "An Efficient Hybrid Model to Load Forecasting", IJCSNS Int. Journal of Computer Science and Network Security, VOL.10, No.8, Aug. 2010.

[41] Min Hen, Jianhui Xi, Shiguo Xu, Fu-Liang Yin, "Prediction of Chaotic Time Series Based on the Reccurent Predictor Neural Network",vol52.No.12.2004

[42] Holger R. Maier, Graeme C. Dandy, "Neural Networks for the Prediction and Forecasting of Water Resources Variables: A Review of Modeling Issues and Applications", Environmental Modeling & Software 15 (2000) 101–124.

[43] G.A. Adepoju, S.O.A. Ogunjuyigbe, and K.O. Alawode, "Application of Neural Network to Load Forecasting in Nigerian Electrical Power System", Department of Electronic and Electrical Engineering, Ladoke Akintola Univ. of Technology, Nigeria.

[44] Holben, B.N., "Characteristics of Maximum-Value Composite Images from Temporal Data", International Journal of Remote Sensing, 7:1417-1434, 1986.

[45] EUMETSAT, "Meteosat Second Generation in Orbit", MSG.02 Version 1, Am Kavalleriesand 31, D-64295 Darmstadt, Germany, 2005, Web: http://www.eumetsat.int.

[46] Smith, P.M., Kalluri, S., Prince, S., Defries, R., "The NOAA NASA Pathfinder AVHRR 8km Land Data Set", Photogrammetric Engineering and Remote Sens., 63: 12-31, 1997.

[47] Salem Saleh Al-amri, N.V. Kalyankar and Khamitkar S.D., "A Comparative Study of Removal Noise from Remote Sensing Image", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue. 1, No. 1, Jan. 2010.

[48] Suman Tatiraju & Avi Mehta, "Image Segmentation Using K-means Clustering, EM and Normalized Cuts", Department of EECS University Of California, Irvine.

[49] J. Arenas-Garc´ıa, K. B. Petersen, and L. K. Hansen, "Linear Dimensionality Reduction for Multi-label Classification", In Twenty-first International Joint Conference on Artificial Intelligence, 2009.

[50] I.T. Jolliffe, "Principal Component Analysis", 2nd ed., Springer series in Statistics.

[51] Rumelhart, D.E., Hinton, G.E. and Williams, R.J., "Learning Representations by Back-Propagating Errors", Nature (London), 323, 533–536, 1986.

[52] McCulloch, W.S. and Pitts, W., "A Logical Calculus of the Ideas Immanent in Nervous Activity", Bulletin of Mathematical Biophysics, 5, 115–137, 1943.

[53] Ajith Abraham, "Artificial Neural Network", Oklahoma State Univ., Stillwater, USA

[54] NDMC , "Understanding and Defining what is Drought", Release of 2006, Retrieved 16, Nov., 2011 from http://www.drought.unl.edu/whatis/concept.htm

[55] Getachew, B., Tsegaye, T., Solomon, A., Shawndra, H., "Drought Monitoring in Food Insecure Areas of Ethiopia by Using Satellite Technologies", on the book: "Experiences of Climate Change Adaptation in Africa" (editor: Prof. Walter Leal Filho), 2010.

[56] Michael Negnevitsky, "Artificial Intelligence: A Guide to Intelligent Systems", 2nd ed., 2005.

[57] Siddheswar Ray and Rose H. Turi, "Determination of Number of Clusters in K-Means Clustering and Application in Colour Image Segmentation", School of Computer Science and Software Engineering Monash University, Wellington Road, Clayton, Victoria, 3168, Australia.

APPENDIX A: BACK-PROPAGATION ALGORITHM

The back-propagation algorithm is used in layered feed-forward ANNs [51]. This means that the artificial neurons are organized in layers, and send their signals "forward", and then the errors are propagated backwards. The network receives inputs by neurons in the *input layer*, and the output of the network is given by the neurons on an *output layer*. There may be one or more intermediate *hidden layers*. The back-propagation algorithm uses supervised learning, which means that we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error (difference between actual and expected results) is calculated.

Considering the multi layer perceptron (MLP) in Fig 4.3, the back propagation algorithm is described below. A unit in the output layer determines its activity by following a two step procedure.

First, it computes the total weighted input x_j, using the formula [53]:

$$X_j = \sum_i y_i W_{ij} \tag{A.1}$$

where y_i is the activity level of the jth unit in the previous layer and W_{ij} is the weight of the connection between the ith and the jth unit.

Next, the unit calculates the activity y_j using some function of the total weighted input. Typically we use the sigmoid function:

$$y_j = \frac{1}{1+e^{-x_j}} \tag{A.2}$$

Once the activities of all output units have been determined, the network computes the error E, which is defined by the expression:

$$E = \frac{1}{2}\sum_i (y_i - d_i)^2 \tag{A.3}$$

Where, y_i is the activity level of the jth unit in the top layer and d_i is the desired output of the jth unit.

The back-propagation algorithm consists of four steps [51, 53]:

1. Compute how fast the error changes as the activity of an output unit is changed. This error derivative (EA) is the difference between the actual and the desired activity.

$$EA_j = \frac{\partial E}{\partial y_j} = y_j - d_j \tag{A.4}$$

2. Compute how fast the error changes as the total input received by an output unit is changed. This quantity (EI) is the answer from step 1 multiplied by the rate at which the output of a unit changes as its total input is changed.

$$EI_j = \frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j} \times \frac{dy_j}{dx_j} = EA_j y_j (1 - y_j) \tag{A.5}$$

3. Compute how fast the error changes as a weight on the connection into an output unit is changed. This quantity (EW) is the answer from step 2 multiplied by the activity level of the unit from which the connection emanates.

$$EW_{ij} = \frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial x_j} \times \frac{\partial x_j}{\partial W_{ij}} = EI_j y_i \tag{A.6}$$

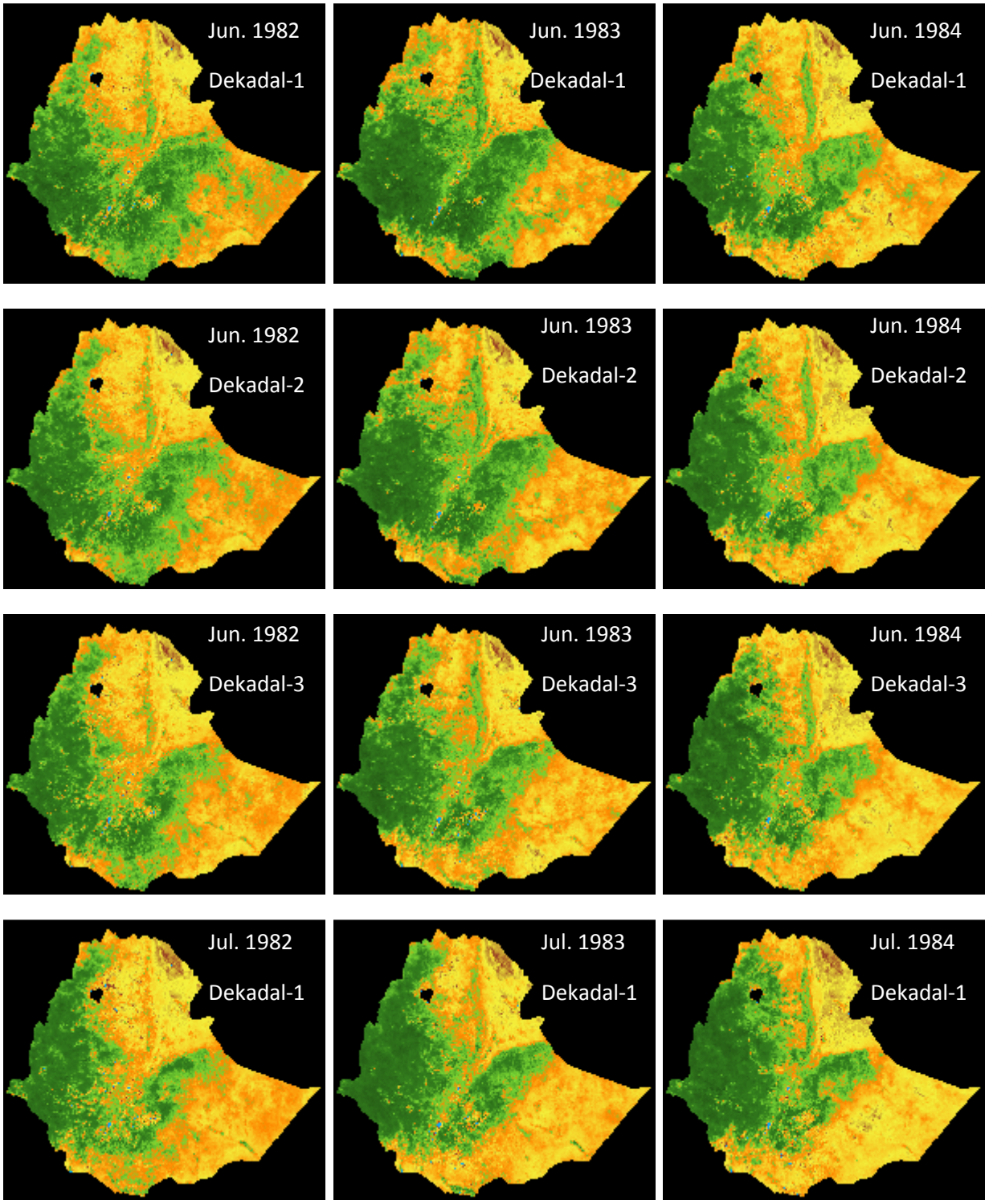
4. Compute how fast the error changes as the activity of a unit in the previous layer is changed. This crucial step allows back propagation to be applied to multilayer networks. When the activity of a unit in the previous layer changes, it affects the activities of all the output units to which it is connected. So to compute the overall effect on the error, we add together all these separate effects on output units. But each effect is simple to calculate. It is the answer in step 2 multiplied by the weight on the connection to that output unit.

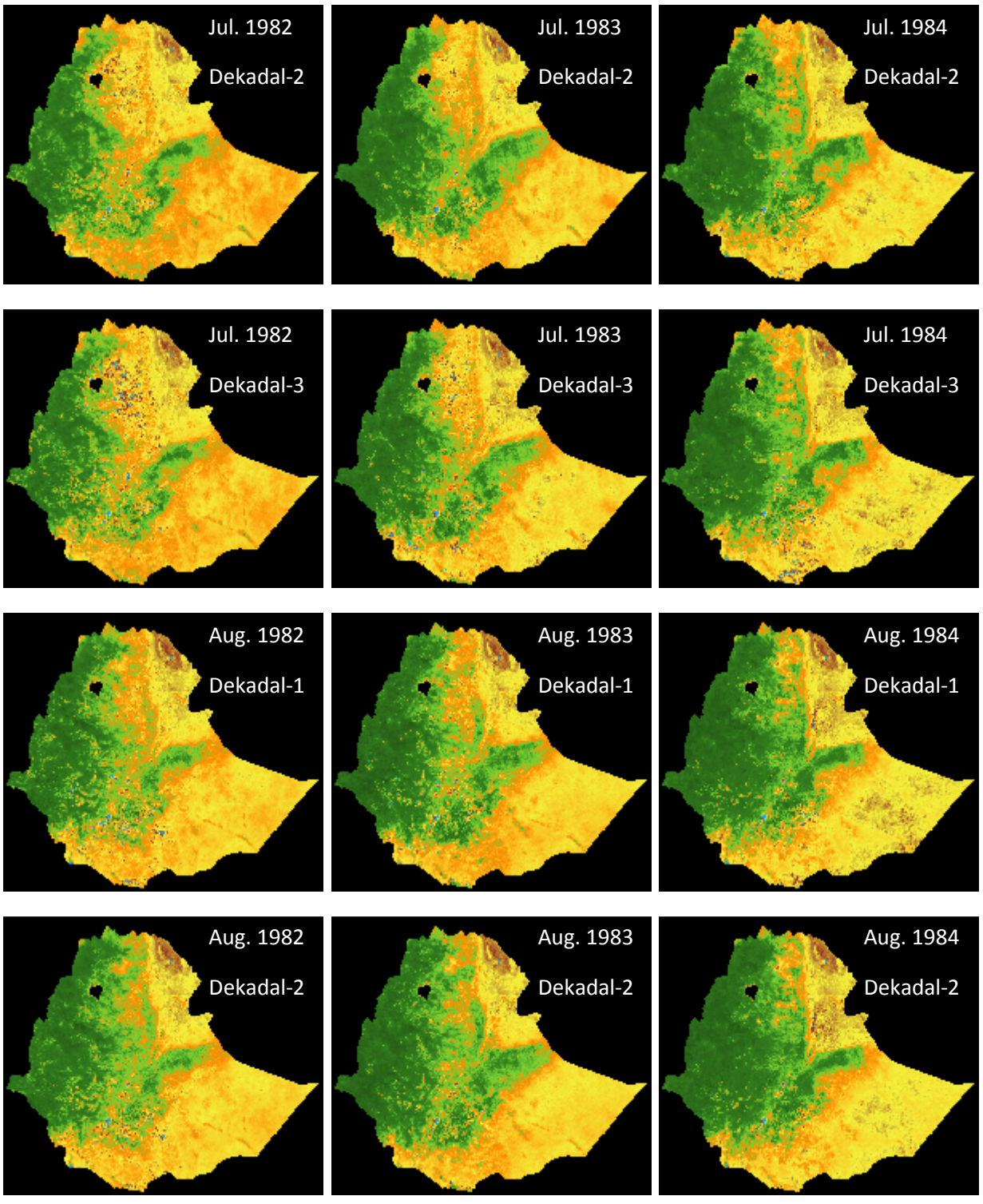
$$EA_{ij} = \frac{\partial E}{\partial y_i} = \sum_j \frac{\partial E}{\partial x_j} \times \frac{\partial x_j}{\partial y_i} = \sum_j EI_j \times W_{ij} \tag{A.7}$$

By using steps 2 and 4, we can convert the *EAs* of one layer of units into *EAs* for the previous layer. This procedure can be repeated to get the *EAs* for as many previous layers as desired. Once we know the *EA* of a unit, we can use steps 2 and 3 to compute the *EWs* on its incoming connections.

APPENDIX B: SAMPLE IMAGES AND EXTRACTED DESCRIPTORS

B.1: Sample NDVI Images




Jul. 1982
Dekadal-2
Jul. 1983
Dekadal-2
Jul. 1984
Dekadal-2
Jul. 1982
Dekadal-3
Jul. 1983
Dekadal-3
Jul. 1984
Dekadal-3
Aug. 1982
Dekadal-1
Aug. 1983
Dekadal-1
Aug. 1984
Dekadal-1
Aug. 1982
Dekadal-2
Aug. 1983
Dekadal-2
Aug. 1984
Dekadal-2

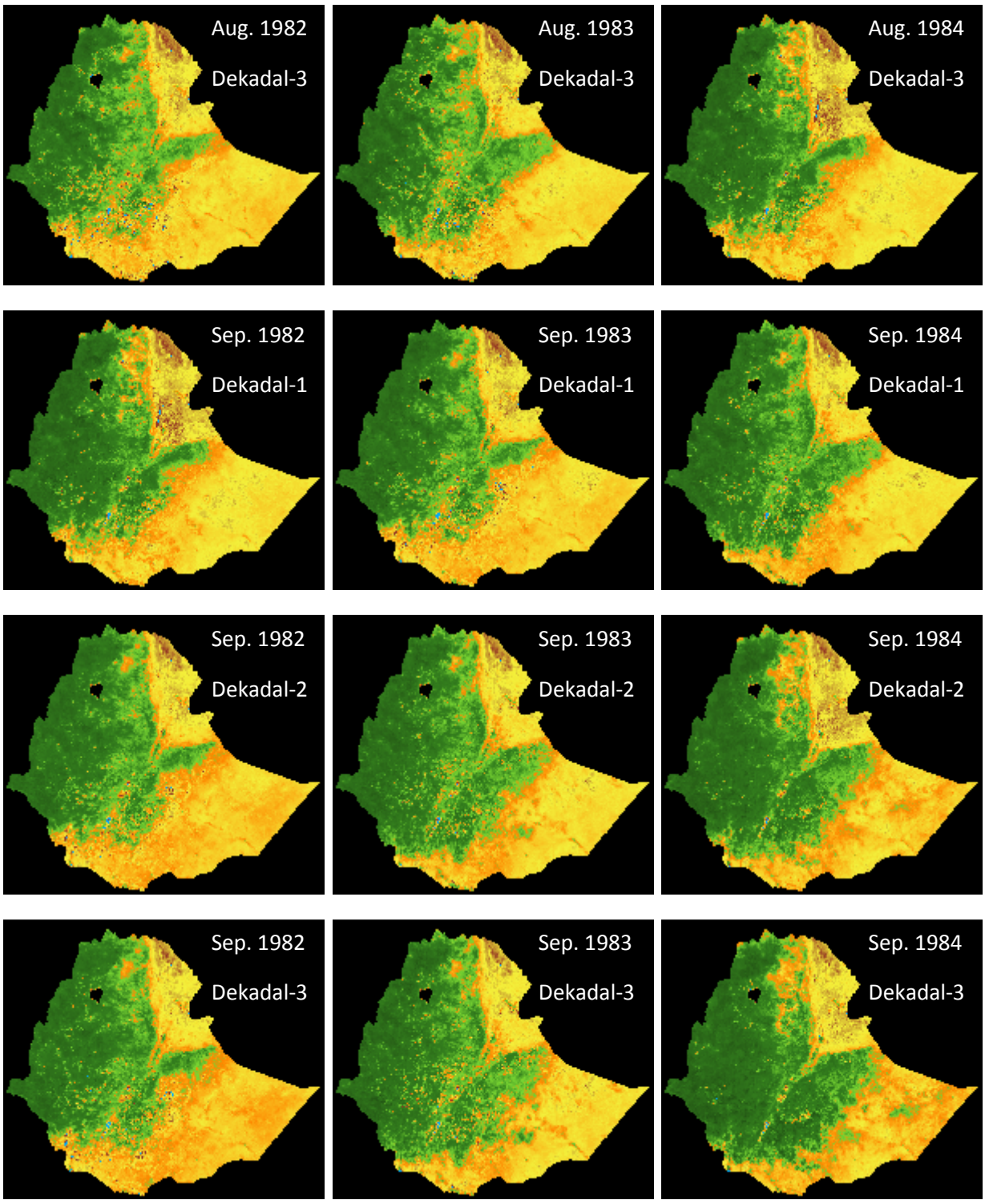


Fig B.1 NDVI Images of the Years 1982, 1983, and 1984. Dekadal Numbers Indicate the Ten Day Intervals in Each Month (June - September)

B.2 Sample NDVI Deviation Images

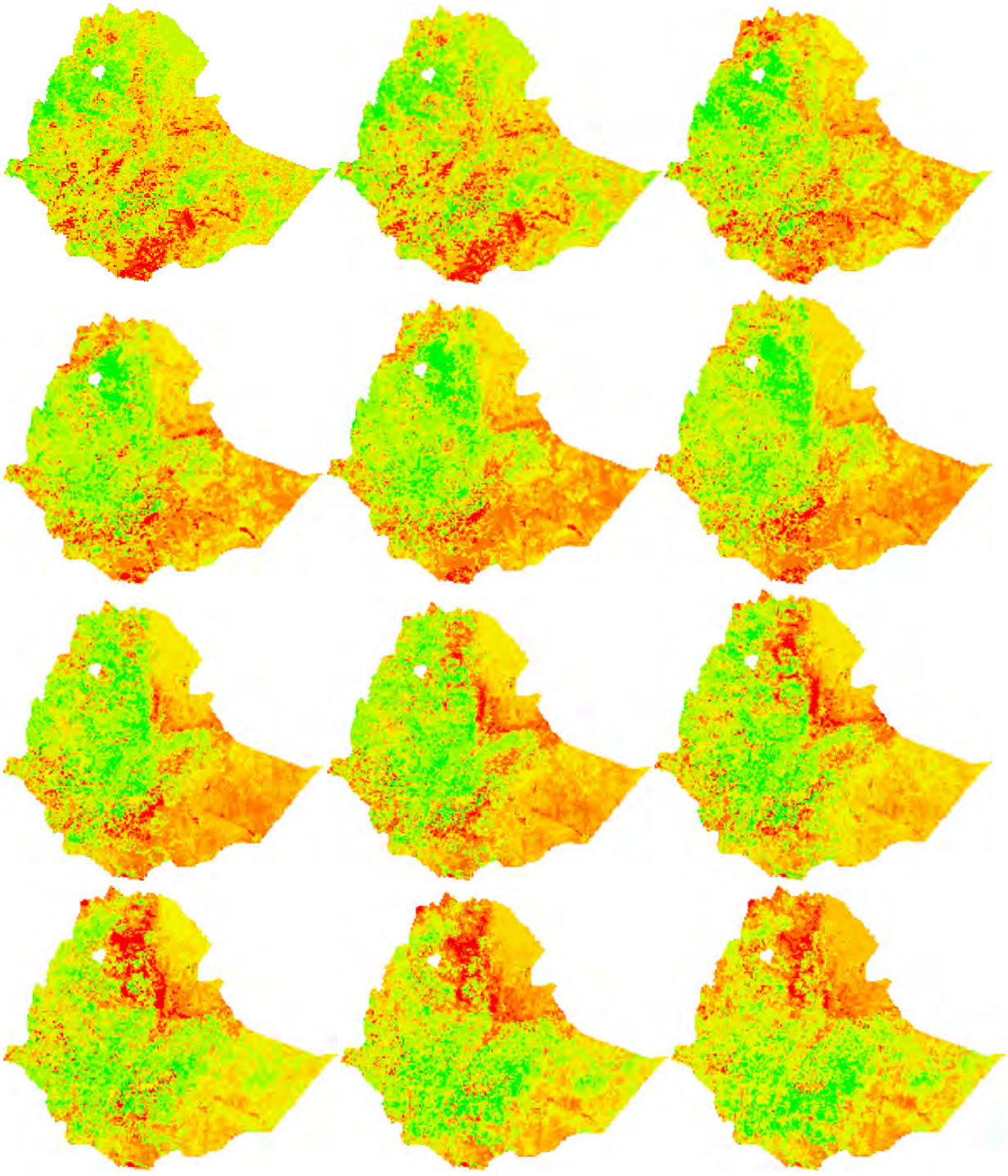


Fig B.2 Deviation of NDVI derived from NDVI's in Fig B.1 for 1984

B.3 Sample AOI for Drought Object Extraction

The extraction of objects was performed on the AOI shown in Fig 5.4. The locations of some these objects are shown in Fig B.3. This locations are used over all the images in the growing season to produce a series of object characterstics over time. Accordingly, the descriptors of the objects over time are generated usin PCA as shown in section B.4.

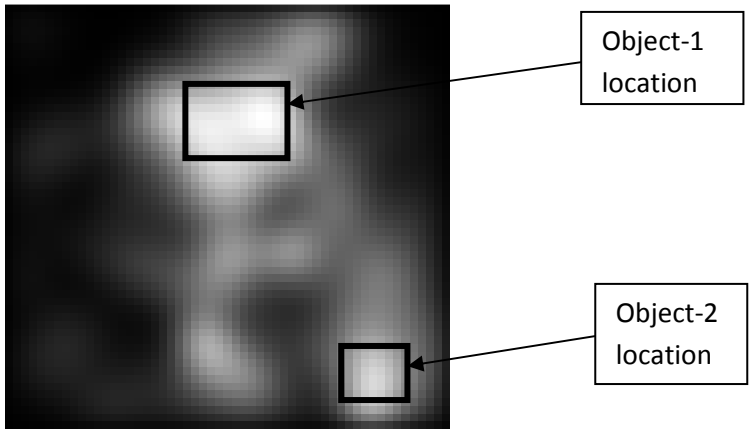


Fig B.3 Sample Object Locations for which Descriptors are Produced

B.4 Features of Sample Drought Objects Used for Training and Testing

The following tables indicate the features of drought objects extracted from different regions of the Ethiopian map. Forexample, tables B.1 and B.2 are for objects extracted over the growing season of 1984 from locations shown in Fig B.2. The remaining features are for objects from different locations extracted in similar to that shown in Fig B.2.

The namings object-1, object-2,…, are to indicate the different series of objects extracted from different locations. The 15 descriptors for every object at a given time instance (dekadal) are shown by the variables d_1, d_2, d_3,……, d_{15}. Thus, the columns indicate the series in dekadals over a growing season. For example, the first dekadal by Jun10 because it includes the days of the June month from 1 to 10, the second dekadal by Jun20, …, etc. until all the dekadals from beginning of June to end of Septmber are covered. Generally, a column indicates the descriptors of object at that time instance and a row indicates the descriptor number.

Table B.1 Descriptors of Object-1 Over Time for One Growing Season

Descriptor no.	Features of drought object represented in terms of descriptors for 12 dekadals (1984)											
	Jun1	Jun2	Jun3	Jul1	Jul2	Jul3	Aug1	Aug2	Aug3	Sep1	Sep2	Sep3
d1	41.1	30.2	24.1	103	140	162	308	358	179	230	52.2	57.4
d2	41.1	20.2	2.54	30.4	31.9	35.7	294	328	138	40.9	21.1	7.11
d3	10.8	6.81	2.03	7.14	13.6	20.9	6.97	222	30.6	37.5	8.79	5.07
d4	9.47	6.1	1.76	2.81	12	14.8	6.63	11.5	25.5	20	4.53	2.19
d5	8.74	1.36	0.465	1.3	11.6	9.46	5.81	6.65	24.2	16.5	3.76	1.82
d6	3.41	1.02	0.45	1.21	2.33	4.72	3.66	6.37	20.8	4.75	2.76	1.31
d7	1.81	0.945	0.439	1.12	1.42	4.6	1.37	3.73	13.4	4.38	2.13	1.07
d8	1.11	0.587	0.355	.857	1.07	4.13	1.14	1.91	7.57	3.95	0.889	0.634
d9	0.723	0.575	0.327	.662	1.04	3.21	0.626	1.36	4.86	2.38	0.812	0.447
d10	0.592	0.573	0.325	.659	.952	3.12	0.53	0.948	2.62	1.26	0.685	0.44
d11	0.413	0.486	0.192	.554	.914	1.62	0.452	0.621	0.976	0.935	0.574	0.329
d12	0.383	0.281	-.015	.538	.851	1.29	0.329	0.453	0.802	0.846	0.534	0.319
d13	0.285	0.183	-.275	.433	.619	.883	0.262	0.407	0.603	0.735	0.419	.0361
d14	0.262	0.181	-.288	0.41	.497	.855	0.211	0.407	0.519	0.626	0.372	.0115
d15	0.262	.0393	-.411	0.38	.162	0.78	0.156	0.314	0.487	0.519	0.274	-

Table B.2 Descriptors of Object-2 Over Time for One Growing Season

Descriptor no.	Features of drought object represented in terms of descriptors for 12 dekadals (1984)											
	Jun1	Jun2	Jun3	Jul1	Jul2	Jul3	Aug1	Aug2	Aug3	Sep1	Sep2	Sep3
d1	217	181	164	98.9	167	203	448	627	501	537	249	94.5
d2	81.8	72.8	44.6	82.8	138	163	181	337	118	46.7	42.4	72.1
d3	39.7	28.5	35.3	37.5	25.8	56	111	69.4	50.4	46.2	36.5	32.9
d4	32.6	24.2	33.9	19.4	25.7	42.2	58.1	61.7	50.4	45.9	34.8	32.2
d5	28.1	22.2	33.8	17	25.2	36.5	51	37.6	36.2	31.1	33.3	30.9
d6	23.5	21.4	27	12.3	20	22.4	30.3	23	25.6	20.7	22.9	24.7
d7	22.3	18.9	22.5	11.2	15.5	13.9	13.4	22.4	18.5	18.5	20	18.7
d8	20	17.5	22.5	7.61	13.6	12.5	11.2	21.3	14.8	16.9	19.4	10.7
d9	19.9	16.3	20.2	6.9	11.5	9.7	9.56	11.8	12.4	15.4	12.2	10.3
d10	19.1	12	15.6	6.55	8.33	7.76	9.51	11.5	8.81	13.2	11.6	9.57
d11	11.3	9.22	13.1	6.47	5.99	7.66	7.55	10.9	6.81	11.7	9.48	6.1
d12	9.43	7.67	8.37	5.6	4.89	6.58	6.58	8.36	6.59	7.66	8.01	4.49
d13	6.9	7.52	5.48	5.49	3.94	3.79	4.79	5.52	3.53	7.61	7.78	4.1
d14	5.7	5.51	1.94	5.47	3.44	3.36	3.16	3.2	3.31	7.28	5.07	-
d15	3.89	0.297	1.12	5.01	1.62	1.06	0.373	2.27	2.24	7.02	3.01	-

Table B.3 Descriptors of Object-3 Over Time for One Growing Season

Descriptor no.	Features of drought object represented in terms of descriptors for 12 dekadals (1984)											
	Jun1	Jun2	Jun3	Jul1	Jul2	Jul3	Aug1	Aug2	Aug3	Sep1	Sep2	Sep3
d1	405	318	127	330	398	529	1000	1110	826	1220	957	474
d2	29.7	43.1	20.4	40.9	49.1	135	258	220	84.6	119	103	85.5
d3	20.6	14.3	13.6	39.7	44.5	75.3	118	91.7	63.5	26.4	13.3	12.8
d4	5.8	4.36	3.7	6.39	36.6	74.2	100	47.1	61.7	25.7	12.4	8.09
d5	5.32	4.35	1.56	3.7	4.39	61.1	60.8	31.7	11.7	13.5	4.45	4.27
d6	3.39	2.39	1.09	2.15	2.43	5.25	12.3	3.4	8.95	12.2	3.22	2.66
d7	1.99	1.39	1.02	2.05	1.6	4.4	8.96	2.24	5.21	8.84	1.52	1.9
d8	1.03	.876	.869	1.97	1.47	3.24	3.27	2.11	4.41	4.16	1.08	1.29
d9	.412	.473	.547	1.36	1.38	1.97	2.61	1.98	4.39	2.81	.721	1.23
d10	.325	.447	.499	1.06	1.29	1.63	1.55	1.11	1.98	1.44	.646	0.421
d11	.265	.314	.412	1.02	1.08	1.61	1.28	.733	1.1	1.3	.573	0.348
d12	.107	.138	.392	.849	1.04	1.53	.99	.718	.453	.837	.508	0.252
d13	.085	.126	.345	.181	.956	1.09	.608	.474	.451	.834	.482	0.163
d14	.059	-.133	.205	-	.767	1.08	.271	.241	.205	.592	.369	0.048
d15	.047	-.155	.079	-	.632	1.01	.205	.147	.192	.485	.30	-.023

Table B.4 Descriptors of Object-4 Over Time for One Growing Season

Descriptor no.	Features of drought object represented in terms of descriptors for 12 dekadals (1984)											
	Jun1	Jun2	Jun3	Jul10	Jul20	Jul31	Aug1	Aug2	Aug3	Sep1	Sep2	Sep3
d1	255	167	103	38.3	50.3	60.4	312	563	750	1300	1360	1130
d2	168	113	49.2	34.9	34.8	33	83.6	446	285	28.6	36.5	41.8
d3	61.8	57.3	40	12.8	21.9	17.5	72.1	159	174	21	26.7	23.4
d4	18.4	24.4	19.5	9.1	6.85	12.2	28.7	40.1	14.5	4.21	2.44	12.5
d5	15.8	21.4	5.79	7.13	1.77	7.4	5.47	7.28	11.2	2.1	2.22	4.63
d6	13	6.97	2.52	4.25	1.68	6.58	1.42	5.42	5.08	1.01	2.13	3.64
d7	4.27	1.86	1.89	2.07	0.97	5.49	1.21	3.45	4.53	0.698	1.36	1.88
d8	4.14	1.4	1.48	1.84	0.91	2.89	1.05	3.15	2.94	0.647	1.18	1.21
d9	2.01	1.22	1.44	1.73	0.48	2.73	0.729	1.44	1.4	0.641	1.1	0.932
d10	0.949	0.991	0.673	1.7	0.47	2.34	0.708	0.86	1.27	0.496	0.989	0.686
d11	0.846	0.615	0.586	1.01	0.32	1.01	0.571	0.552	0.975	0.476	0.867	0.42
d12	0.818	0.386	0.402	0.35	0.29	0.95	0.513	0.423	0.937	0.454	0.735	0.397
d13	0.433	0.209	0.349	0.26	.194	.863	-.0133	0.354	0.692	0.349	0.655	0.331
d14	0.353	-.135	0.214	0.15	0.18	.706	-0.122	0.281	0.532	0.142	0.46	0.241
d15	0.136	-.201	0.2	.017	0.11	0.45	-0.126	0.204	0.448	0.111	0.458	0.235

Table B.5 Descriptors of Object-5 Over Time for One Growing Season

Descriptor no.	Features of drought object represented in terms of descriptors for 12 dekadals (1984)											
	Jun1	Jun2	Jun3	Jul1	Jul20	Jul31	Aug1	Aug2	Aug3	Sep1	Sep2	Sep3
d1	420	282	113	35.5	27.7	18.4	376	808	818	1570	1620	1580
d2	150	134	103	21.1	13.7	17.1	73.7	443	307	53.4	69.9	72.6
d3	65.7	119	75.7	16.9	8.15	15.5	60.8	175	186	29.2	34.5	40.2
d4	55.5	37	50.8	16.1	6.04	15.5	33.8	108	92.1	26.9	32.4	38.5
d5	39	30.2	31.3	12.5	5.2	15.1	28.3	52.5	34.6	15.7	29.2	32.8
d6	33.2	29.6	26	10.5	4.92	13.3	19.6	34	24.8	14.8	25.5	29.9
d7	30.4	29.2	15	8.53	4.54	9.39	12.5	26.7	18.8	11.3	19.6	29.4
d8	25.7	20.2	10.2	5.63	3.89	6.72	8.96	21.1	17.8	8.88	16.8	27.3
d9	22.6	18.5	7.29	5.5	2.54	5.36	7.48	20.6	16.9	8.28	10.3	26
d10	13.7	16.7	7.08	4.83	2.08	4.39	7.11	19.1	12.5	6.06	9.29	20.3
d11	11.9	9.81	6.54	4.54	1.85	4.38	6.18	18.8	11.8	4.77	5.84	18.7
d12	9.47	8.88	6.34	4.5	1.74	3.81	4.61	17.3	10.1	4.73	5.02	16.2
d13	5.21	7.58	3.89	4.41	0.93	2.79	3.02	16.4	8.3	4.06	4.14	12.3
d14	3.39	5.86	1.24	2.29	0.17	1.51	2.78	15.1	8.11	-.244	3.43	11.2
d15	1.92	-.319	0.296	2.02	-.304	0.17	2.64	6.72	7.06	-1.1	0.833	10.9

Table B.6 Descriptors of Object-6 Over Time for One Growing Season

Descriptor no.	Features of drought object represented in terms of descriptors for 12 dekadals (1984)											
	Jun1	Jun2	Jun3	Jul1	Jul2	Jul3	Aug1	Aug2	Aug3	Sep1	Sep2	Sep3
d1	700	426	161	352	407	621	1240	1380	1160	1510	1330	830
d2	89.6	99.7	99.5	222	239	476	165	150	49	36.7	38.6	33.3
d3	60.1	77.1	31.1	89.4	150	229	121	21.6	39.3	30.6	14.9	31.2
d4	58.1	54.8	26.7	53.4	104	48.3	52.3	21.4	21.6	24.4	14.7	22.8
d5	23.8	46.5	20	25.3	37.2	42.2	42.9	17.6	18.2	23.5	12.9	19.6
d6	17.7	32.5	18.8	23.8	35.6	32.8	28.9	15.7	12.9	21.8	12.7	16.7
d7	17.6	31.2	12.6	21.2	26.3	19	28.9	8.22	12.2	18.8	12.1	12.1
d8	16.6	19.6	10.8	20.8	23.3	11.5	17.4	7.92	11.8	15.6	9.89	11.9
d9	14.7	18.9	7.87	20.2	22.4	7.4	16.8	6.68	10.3	13.7	6.01	9.11
d10	13.2	18.3	7.76	19.7	14.5	5.05	16.3	6.34	8.8	11.1	5.37	6.24
d11	10.5	14.3	7.1	16.2	10.4	5.03	16.2	1.07	6.62	8.91	4.88	3.68
d12	9.93	12.2	6.19	7.82	6.99	3.85	15.9	0.783	2.58	5.75	4.62	2.42
d13	5.57	11.7	6.04	6.84	6.74	3.8	10.4	-1.67	1.46	4.97	3.4	1.73
d14	4.25	11	3.25	6.77	4.41	3.79	4.5	-3.44	-.0842	-.377	0.734	-.639
d15	3.08	9.54	3.07	5.7	4.1	2.93	1.73	-5.1	-1.54	-.479	0.403	-.726

Table B.7 Descriptors of Object-7 Over Time for One Growing Season

Descriptor no.	Features of drought object represented in terms of descriptors for 12 dekadals (1984)											
	Jun1	Jun2	Jun3	Jul1	Jul20	Jul31	Aug1	Aug2	Aug3	Sep1	Sep2	Sep3
d1	420	282	113	35.5	27.7	18.4	376	808	818	1570	1620	1580
d2	150	134	103	21.1	13.7	17.1	73.7	443	307	53.4	69.9	72.6
d3	65.7	119	75.7	16.9	8.15	15.5	60.8	175	186	29.2	34.5	40.2
d4	55.5	37	50.8	16.1	6.04	15.5	33.8	108	92.1	26.9	32.4	38.5
d5	39	30.2	31.3	12.5	5.2	15.1	28.3	52.5	34.6	15.7	29.2	32.8
d6	33.2	29.6	26	10.5	4.92	13.3	19.6	34	24.8	14.8	25.5	29.9
d7	30.4	29.2	15	8.53	4.54	9.39	12.5	26.7	18.8	11.3	19.6	29.4
d8	25.7	20.2	10.2	5.63	3.89	6.72	8.96	21.1	17.8	8.88	16.8	27.3
d9	22.6	18.5	7.29	5.5	2.54	5.36	7.48	20.6	16.9	8.28	10.3	26
d10	13.7	16.7	7.08	4.83	2.08	4.39	7.11	19.1	12.5	6.06	9.29	20.3
d11	11.9	9.81	6.54	4.54	1.85	4.38	6.18	18.8	11.8	4.77	5.84	18.7
d12	9.47	8.88	6.34	4.5	1.74	3.81	4.61	17.3	10.1	4.73	5.02	16.2
d13	5.21	7.58	3.89	4.41	0.93	2.79	3.02	16.4	8.3	4.06	4.14	12.3
d14	3.39	5.86	1.24	2.29	0.17	1.51	2.78	15.1	8.11	-.244	3.43	11.2
d15	1.92	-.319	0.296	2.02	-.304	0.17	2.64	6.72	7.06	-1.1	0.833	10.9

Table B.8 Descriptors of Object-8 Over Time for One Growing Season

Descriptor no.	Features of drought object represented in terms of descriptors for 12 dekadals (1984)											
	Jun1	Jun2	Jun3	Jul1	Jul2	Jul3	Aug1	Aug2	Aug3	Sep1	Sep2	Sep3
d1	665	405	153	334	387	590	1180	1310	1100	1430	1260	789
d2	85.1	94.7	94.5	211	227	452	157	143	46.5	34.9	36.7	31.6
d3	57.1	73.2	29.5	84.9	143	218	115	20.5	37.3	29.1	14.2	29.6
d4	55.2	52.1	25.4	50.7	98.8	45.9	49.7	20.3	20.5	23.2	14	21.7
d5	22.6	44.2	19	24	35.3	40.1	40.8	16.7	17.3	22.3	12.3	18.6
d6	16.8	30.9	17.9	22.6	33.8	31.2	27.5	14.9	12.3	20.7	12.1	15.9
d7	16.7	29.6	12	20.1	25	18.1	27.5	7.81	11.6	17.9	11.5	11.5
d8	15.8	18.6	10.3	19.8	22.1	10.9	16.5	7.52	11.2	14.8	9.4	11.3
d9	14	18	7.48	19.2	21.3	7.03	16	6.35	9.79	13	5.71	8.65
d10	12.5	17.4	7.37	18.7	13.8	4.8	15.5	6.02	8.36	10.5	5.1	5.93
d11	9.97	13.6	6.74	15.4	9.88	4.78	15.4	1.02	6.29	8.46	4.64	3.5
d12	9.43	11.6	5.88	7.43	6.64	3.66	15.1	0.744	2.45	5.46	4.39	2.3
d13	5.29	11.1	5.74	6.5	6.4	3.61	9.88	-1.59	1.39	4.72	3.23	1.64
d14	4.04	10.4	3.09	6.43	4.19	3.6	4.27	-3.27	-0.08	-.358	0.697	-.607
d15	2.93	9.06	2.92	5.42	3.89	2.78	1.64	-4.84	-1.46	-.455	0.383	-0.69

Table B.9 Descriptors of Object-9 Over Time for One Growing Season

Descriptor no.	Features of drought object represented in terms of descriptors for 12 dekadals (1984)											
	Jun1	Jun2	Jun3	Jul10	Jul2	Jul3	Aug1	Aug2	Aug3	Sep1	Sep2	Sep3
d1	387	590	1180	1310	1100	1430	1260	789	387	590	1180	1310
d2	227	452	157	143	46.5	34.9	36.7	31.6	227	452	157	143
d3	143	218	115	20.5	37.3	29.1	14.2	29.6	143	218	115	20.5
d4	98.8	45.9	49.7	20.3	20.5	23.2	14	21.7	98.8	45.9	49.7	20.3
d5	35.3	40.1	40.8	16.7	17.3	22.3	12.3	18.6	35.3	40.1	40.8	16.7
d6	33.8	31.2	27.5	14.9	12.3	20.7	12.1	15.9	33.8	31.2	27.5	14.9
d7	25	18.1	27.5	7.81	11.6	17.9	11.5	11.5	25	18.1	27.5	7.81
d8	22.1	10.9	16.5	7.52	11.2	14.8	9.4	11.3	22.1	10.9	16.5	7.52
d9	21.3	7.03	16	6.35	9.79	13	5.71	8.65	21.3	7.03	16	6.35
d10	13.8	4.8	15.5	6.02	8.36	10.5	5.1	5.93	13.8	4.8	15.5	6.02
d11	9.88	4.78	15.4	1.02	6.29	8.46	4.64	3.5	9.88	4.78	15.4	1.02
d12	6.64	3.66	15.1	0.74	2.45	5.46	4.39	2.3	6.64	3.66	15.1	0.744
d13	6.4	3.61	9.88	-1.59	1.39	4.72	3.23	1.64	6.4	3.61	9.88	-1.59
d14	4.19	3.6	4.27	-3.27	-0.08	-.358	0.697	-0.607	4.19	3.6	4.27	-3.27
d15	3.89	2.78	1.64	-4.84	-1.46	-.455	0.383	-0.69	3.89	2.78	1.64	-4.84

Table B.10 Result of Prediction for the Best Network Using Object-9

Descriptor no.	Features of drought object represented in terms of descriptors for 12 dekadals (1984)											
	Jun1	Jun2	Jun3	Jul10	Jul20	Jul3	Aug1	Aug2	Aug3	Sep1	Sep2	Sep3
d1	386	581	997	1310	1060	1430	1260	786	386	581	997	1310
d2	226	134	209	143	-52	34.4	36.5	31.9	226	134	209	143
d3	143	157	139	20.5	47.3	29	14.3	29.4	143	157	139	20.5
d4	98.8	95.2	92.2	20.3	59.4	23.2	14	21.5	98.8	95.2	92.2	20.3
d5	35.4	18	21	16.7	4.42	22.4	12.2	18.6	35.4	18	21	16.7
d6	33.8	33.8	34.9	14.9	32.2	20.7	12.1	15.9	33.8	33.8	34.9	14.9
d7	25.1	29.6	17.9	7.82	10.3	17.9	11.5	11.5	25.1	29.6	17.9	7.82
d8	22.1	15.8	19.8	7.51	10.4	14.8	9.35	11.3	22.1	15.8	19.8	7.51
d9	21.3	12.3	14.2	6.37	-2.32	13	5.69	8.66	21.3	12.3	14.2	6.37
d10	13.8	12.5	8.38	6.02	7.68	10.5	5.12	5.93	13.8	12.5	8.38	6.02
d11	9.89	9.59	6.98	1	0.82	8.49	4.64	3.51	9.89	9.59	6.98	1
d12	6.65	6.51	6.84	0.72	7.41	5.47	4.38	2.29	6.65	6.51	6.84	0.729
d13	6.4	4.13	1.82	-1.6	-4.61	4.73	3.23	1.65	6.4	4.13	1.82	-1.6
d14	4.18	3.44	-.704	-3.28	-1.01	-.359	0.704	-.604	4.18	3.44	-.704	-3.28
d15	3.89	2.51	.796	-4.85	-1.15	-0.45	0.384	-.696	3.89	2.51	.796	-4.85

APPENDIX C: MATLAB IMPLEMENTATION OF THE SYSTEM

C.1 Functions used for each step

%Anisotropic Diffusion for Smoothing

```
function diffused_image = anisotropicdiff(image, num_iter, delta_t,
kappa, option)
%
%
% A 2-Dimensional network structure of 8 neighboring nodes is

% considered for % diffusion conduction.

%IMAGE    - gray scale image (MxN).
%NUM_ITER - number of iterations.
%DELTA_T  - integration constant (0 <= delta_t <= 1/7).
%KAPPA    - gradient modulus threshold that controls the conduction.
%OPTION   - conduction coefficient functions of by Perona & Malik:
%             1 - c(x,y,t) = exp(-(nablaI/kappa).^2),
%
%             2 - c(x,y,t) = 1./(1 + (nablaI/kappa).^2),
%
%DIFFUSED_IMAGE - (diffused) image with scale-space parameter.
%

%

% Convert input image to double.
image = double(image);

% PDE (partial differential equation) initial condition.
diffused_image = image;

% distance between center of pixels.
dx = 1;
dy = 1;
dd = sqrt(2);

% 2-dimension convolution masks - finite differences.
hN = [0 1 0; 0 -1 0; 0 0 0];
hS = [0 0 0; 0 -1 0; 0 1 0];
hE = [0 0 0; 0 -1 1; 0 0 0];
hW = [0 0 0; 1 -1 0; 0 0 0];
hNE = [0 0 1; 0 -1 0; 0 0 0];
hSE = [0 0 0; 0 -1 0; 0 0 1];
hSW = [0 0 0; 0 -1 0; 1 0 0];
hNW = [1 0 0; 0 -1 0; 0 0 0];

% Anisotropic diffusion.
for t = 1:num_iter
```

```
        % Finite differences.
        nablaN = conv2(diffused_image,hN,'same');
        nablaS = conv2(diffused_image,hS,'same');
        nablaW = conv2(diffused_image,hW,'same');
        nablaE = conv2(diffused_image,hE,'same');
        nablaNE = conv2(diffused_image,hNE,'same');
        nablaSE = conv2(diffused_image,hSE,'same');
        nablaSW = conv2(diffused_image,hSW,'same');
        nablaNW = conv2(diffused_image,hNW,'same');

        % Diffusion function.
        if option == 1
            cN = exp(-(nablaN/kappa).^2);
            cS = exp(-(nablaS/kappa).^2);
            cW = exp(-(nablaW/kappa).^2);
            cE = exp(-(nablaE/kappa).^2);
            cNE = exp(-(nablaNE/kappa).^2);
            cSE = exp(-(nablaSE/kappa).^2);
            cSW = exp(-(nablaSW/kappa).^2);
            cNW = exp(-(nablaNW/kappa).^2);
        elseif option == 2
            cN = 1./(1 + (nablaN/kappa).^2);
            cS = 1./(1 + (nablaS/kappa).^2);
            cW = 1./(1 + (nablaW/kappa).^2);
            cE = 1./(1 + (nablaE/kappa).^2);
            cNE = 1./(1 + (nablaNE/kappa).^2);
            cSE = 1./(1 + (nablaSE/kappa).^2);
            cSW = 1./(1 + (nablaSW/kappa).^2);
            cNW = 1./(1 + (nablaNW/kappa).^2);
        end

        % Discrete PDE solution.
        diffused_image = diffused_image + ...
                  delta_t*(...
                  (1/(dy^2))*cN.*nablaN + (1/(dy^2))*cS.*nablaS + ...
                  (1/(dx^2))*cW.*nablaW + (1/(dx^2))*cE.*nablaE + ...
                  (1/(dd^2))*cNE.*nablaNE + (1/(dd^2))*cSE.*nablaSE +
...
                  (1/(dd^2))*cSW.*nablaSW + (1/(dd^2))*cNW.*nablaNW);


end
```

%K-means Segmentation

```
function [mea,overlay]=k_segment(image,k)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   image: grey color image
%   k: Number of classes
%   mea: vector of class means
%   overlay: clasification image mask
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% check image
ima=double(image);
copy=ima;          % make a copy
ima=ima(:);        % vectorize image
mi=min(ima);       % deal with negative
ima=ima-mi+1;      % and zero values

s=length(ima);

%create image histogram

m=max(ima)+1;
h=zeros(1,m);
hc=zeros(1,m);

for i=1:s
  if(ima(i)>0)
      h(ima(i)) = h(ima(i))+1;
  end;
end
ind=find(h);
hl=length(ind);

% initiate centroids

mea=(1:k)*m/(k+1);

% start process

while(true)

  oldmea=mea;
  % current classification

  for i=1:hl
      c=abs(ind(i)-mea);
      cc=find(c==min(c));
      hc(ind(i))=cc(1);
  end

  %recalculation of means

  for i=1:k,
      a=find(hc==i);
      mea(i)=sum(a.*h(a))/sum(h(a));
  end

  if(mu==oldmea)
      break;
  end;

end
```

```
% calculate the overlay as mask
s=size(copy);
overlay=zeros(s);
for i=1:s(1),
for j=1:s(2),
  c=abs(copy(i,j)-mea);
  a=find(c==min(c));
  overlay(i,j)=a(1);
end
end

mea=mea+mi-1;             % recover actual values of the mean
```

%PCA for Dimensionality Reduction

%======================create database of image matrix=======================

```
function M = imageMatrix(PathOfTrainObjects)
% Align a set of drought images (the training set M1, M2, ... , MM )
%
%
% PathOfTrainObjects - Path of the training database for PCA
%
% M                  - A 2D matrix, containing all 1D image vectors.
%

% directory and file handling
TrainFiles = dir(PathOfTrainObjects);
Train_Number = 0;
for i = 1:size(TrainFiles,1)
    if
not(strcmp(TrainFiles(i).name,'.')|strcmp(TrainFiles(i).name,'..')|st
rcmp(TrainFiles(i).name,'Thumbs.db'))
      Train_Number = Train_Number + 1; % No of images in training set
    end
end
%%%%%%%%%%%%%%%%%% Construction of 2D matrix from 1D image vectors
T = [];
for i = 1 : Train_Number

%The name of each image in databases was chosen as a corresponding
%number.
    str = int2str(i);
    str = strcat('\',str,'.tif');
    str = strcat(PathOfTrainObjects,str);
    img = imread(str);
    [irow icol] = size(img);
    temp = reshape(img',irow*icol,1); %Reshape 2D images to 1D image
                                      %vectors
    M = [M temp]; % 'T' grows after each turn
end
```

%================determining features of drought objects using PCA==============

```
function [mn, A, Eigendroughts] = EigendroughtCore(M,no)
%
%  M                - A 2D matrix, containing all 1D image vectors.
%  mn               - (M*Nx1) Mean of the training database
%  Eigendroughts    - Eigenvectors of the cov. matrix of the training
%  A                - Matrix of centered image vectors
%  no               - Number of eigenvectors to be chosen


mn = mean(M,2); % Computing the average drought image
Train_Number = size(M,2);
% Calculating the deviation of each image from mean image
A = [];
for i = 1 : Train_Number
    temp = double(M(:,i)) - mn; % Computing the difference images
    A = [A temp]; % Merging all centered images
End

% Eigendrought method calculation using surrogate matrix
L = A'*A; % L is the surrogate of covariance matrix C=A*A'.
[V D] = eig(L);
%Sorting and eliminating eigenvalues
L_eig_vec = [];
D=diag(D);

d1=sort(D,1,'descend');

dir ='C:\Users\BELOVED\Desktop\';
    fmt2='.txt';
    filename='eigenvalue';
    saveFileName=strcat(dir,filename,fmt2);
    myfile = fopen(saveFileName ,'w'); %'w' opens a new file to write
to
    fprintf(myfile,('\r\n\r\n'));
    fclose(myfile);
    d1=d1';%represents in the text file row wise
    dlmwrite(saveFileName,d1, '-
append','delimiter','\t','precision',3);


L_eig_vec=[];
for i=1:size(V,2)
 ind=find(D==d1(i));
 L_eig_vec = [L_eig_vec V(:,ind)];
end
%select only given eigenvalues and corresponding vectors

d1_sum=sum(d1);

df=[]; vf=[];
for j=1:no
     %if d1(j) > 1
```

```
        df(j)= d1(j);
        vf=[vf L_eig_vec(:,j)];
     % end
end

for i=1:size(vf,2)       %access each column
   kk=vf(:,i);
   temp=sqrt(sum(kk.^2));
   vf(:,i)=vf(:,i)./temp;
end


% calculating the normalized eigenvectors(eigendroughts)

Eigendroughts = A * vf; % A: centered image vectors
for e=1:size(Eigendroughts,2)
    dsum=sqrt(sum(Eigendroughts(:,e).^2));
    Eigendroughts (:,e)= Eigendroughts(:,e)./dsum;
end
```

%====================Descriptor vector producing step ==================

```
function d = FeatureVector(InImage, mn, Eigendroughts)
%
% computes feature vector(d) by projecting the images into
droughtspace
%
% InImage        - Path of the input test image
% mn             - Mean value from 'EigendroughtCore' function.
% Eigendroughts  - Eigenvectors computed in ‘EigendroughtCore' fun.
%
% d              - Descriptor vector of the newly input image
%

%%%%%%%%%%%%%%%% Projecting centered image vectors into droughtspace

%%%%%%%%%%%%%%%%%%%%%%%% Extracting the PCA features from test image
InputImage = imread(InImage);
temp = InputImage(:,:,1);
[irow icol] = size(temp);
InImage = reshape(temp',irow*icol,1);
Difference = double(InImage)-mn; % Centered test image
d = Eigendroughts'*Difference; % Test image feature vector
d=sort(d,1,'descend');
```

%Prediction Using ANN

%==================== back-propagation algorithm==========================

```
%=====================================================================
%This implements the back propagation alg. for descriptor of objects.
%
```

```
%=================================================================
function BackPropagation(trno,l,no,Emax)
%trno    - the training number of the total objects
%l       - number of hidden layers
%no      - number nodes of a layer in the ANN architecture
%Emax    - maximum allowable error
E=10; %initially assumed error to continue a while loop
d=0;

[a,b]=readData(trno,1); %reads the descriptor vectors from file
[ar,ac]=size(a);
[br,bc]=size(b);

%These three lines checks if corresponding o/p is set for every input
if ac~=bc
   error('no of input test vectors not equal to no of output test
vectors ');
end

no(1)=ar; %intitialize no of nodes of input layer to no of input
vectors
no(l)=br; %no of nodes of output layer to no of desired output
vectors
[W,B,delw]=WBinitialise(l,no); %initializes weights and biases


%Generate Random Test Vector of length ac2
ac2=ceil(ac);
r=randint(1,ac2,[1,ac]);
%load matrices r;

for i=1:ac2
   ra(:,i)=a(:,r(i)); %selects input and corresponding o/p randomly,
   rb(:,i)=b(:,r(i)); %w/c helps to randomize input sequence
end


figure('BackingStore','off');
hlr =
uicontrol('Style','slider','value',.04,'Min',0,'Max',0.2,'SliderStep'
,[0.01 0.45],'Position', [75 7 150 20]);
N = get(hlr,'value');
edr=uicontrol('Style','text','FontSize', 12,'string',
num2str(N),'Position',[230 7 50 20]);

%Training Loop using backpropagation algorithm
%=================================================================
while E>Emax && d<10000
    N = get(hlr,'value');
    set(edr,'string',num2str(N));
    E=0;
    d=d+1;
    r=randint(1,ac2,[1,ac2]); %generates random number to select i/p
```

```
    for c=1:ac2

        %Enter training vectors randomly
        for i=1:no(1)
            Y(i,1)=ra(i,r(c));
            V(i,1)=ra(i,r(c));
            der(i,1)=1; %derivative of the transfer function used
        end

        for i=1:no(l)
            D(i)=rb(i,r(c));
        end
%-------------------------------------------------------------------
   %computes the output of hidden layers using logsig transfer
function
        for k=1:l-2
            for j=1:no(k+1)
                w=W(1:no(k),j,k);
                y=Y(1:no(k),k);
                V(j,k+1)=y'*w;
                Y(j,k+1)=logsig(V(j,k+1)-B(j,k+1));
                der(j,k+1)=dlogsig(V(j,k+1)-B(j,k+1),Y(j,k+1));
            end
        end
 %-------------------------------------------------------------------
   %compute the output of output-layer using linear transfer function
        k=l-1;
        for j=1:no(l)
            w=W(1:no(k),j,k);
            y=Y(1:no(k),k);
            V(j,l)=y'*w;
            Y(j,l)=purelin(V(j,l)-B(j,l));
            der(j,l)=dpurelin(V(j,l)-B(j,l));
        end
   %-------------------------------------------------------------------
   %Computation of Error
        for i=1:no(l)
            e(i)=D(i)-Y(i,l);
            E=E+0.5*(e(i)*e(i));
            del(i,l)=der(i,l)*e(i);
        end

        for k=l-1:-1:1
            for i=1:no(k)
                w=W(i,1:no(k+1),k);
                dd=del(1:no(k+1),k+1);
                de=w*dd;
                del(i,k)=de*der(i,k);
            end
        end
   %-------------------------------------------------------------------
   %Adjusts Weights & Bias after every input impression
        for k=l-1:-1:1
```

```
            for i=1:no(k)
                for j=1:no(k+1)
                    delw(i,j,k)=N*del(j,k+1)*Y(i,k)+0.4*delw(i,j,k);
                    W(i,j,k)=W(i,j,k)+delw(i,j,k);
                end
            end
        end

        for k=2:l
            for i=1:no(k)
                delb(i,k)=N*(-1)*del(i,k);
                B(i,k)=B(i,k)+delb(i,k);
            end
        end
    end
   %-----------------------------------------------------------------
    err(d)=E;

    %this saves the network parameters after training
     if rem(d,100)==0              %plot after every 100 epochs
        plot([1:d],err,'blu-',[1:d],Emax,'gre-');

dir1='E:\shortcuts&files\recentdocs\May31\Back_Prop_Alg\ntkParam\';
        filename1='weightmatrix';
        stri=num2str(trno);
        fmt='.mat';
        savefile=strcat(dir1,filename1,stri,fmt);
        save(savefile, 'W', 'B', 'delw');
        pause(0.05);

    end;
end
%End of Training Loop

dir2='E:\shortcuts&files\recentdocs\May31\Back_Prop_Alg\ntkParam\';
filename2='norms';
stri=num2str(trno);
fmt='.mat';
savefile=strcat(dir2,filename2,stri,fmt);

save(savefile, 'no', 'l');

plot(err),title('Learning Curve'),xlabel('   No. of Epochs -----> ');
ylabel('Mean Square Error -----> ');
hold on;
plot([0:d],Emax,'-*gre');
grid on;
```

%====================== reading descriptor vectors==========================

```
%this is used to read input for both traing & testing

function [a,b]=readData(trno,option);
```

```
% option   -to specify ac choice between training & testing
dir1='E:\shortcuts&files\recentdocs\AOI\features35\';
strd=['low    ';'medium '; 'medium2'; 'severe1'; 'severe2';
'severe3';'severe4'];
fmt='.txt';
if (option==1)
for z=1:7  %to choose the directory of a feature vector
  trainingData=strcat(dir1,strd(z,:),fmt);
  im=dlmread(trainingData);

  %normalization of input & target data
    m1 =mean2(im);
    v1=std2(im);

  for g=1:size(im,2)
          m2(g)=max(im(:,g));
          im(:,g)=im(:,g)/m2(g);
      end

  % input vector management for input-this is for 3 lags
  count=0; cnt=0;
 for j=1:size(im,1)-3
         count=count+1;
         inpt(count:count+2,1)=im(j:j+2,trno);
         count=count +2;
         cnt=cnt+1;
         outpt(cnt,:)=im(j+3,trno);     % 4th data in the series-
output

  end
end

elseif ( option==2) % for testing phase
 testingData=strcat(dir1,strd(7,:),fmt);
 im=dlmread(testingData);

      for g=1:size(im,2)
          m2(g)=max(im(:,g));
          im(:,g)=im(:,g)/m2(g);
      end


 count=0; cnt=0;
 for j=1:size(im,1)-3
         count=count+1;
         inpt(count:count+2,1)=im(j:j+2,trno);
         count=count +2;
         cnt=cnt+1;
         outpt(cnt,:)=im(j+3,trno);

  end
else
    error('Sorry! Invalid Choice! ');
```

```
end

%This line vectorizes series of consecutive inputs to one ANN input
in=[];
for o=1:3:size(inpt,1)
     in = [in reshape(inpt(o:o+2,:)',3*size(inpt,2),1)];
end

a=in;
b=outpt';
```

%======================== weights & bias initialization========================

```
%=====================================================================
%This function initializes the weights of each layer
%---------------------------------------------------------------------
function [W,B,delw]=WBinitialise(l,no)

for k=1:l-1
     for i=1:no(k)
         for j=1:no(k+1)
             W(i,j,k)=rand-0.5;
             delw(i,j,k)=0;
         end;
     end;
 end;
 disp('initial Weight Matrix W= ');
 %disp(W);

for k=2:l
    for i=1:no(k)
          B(i,k)=rand;
          delb(i,k)=0;
     end
end
```

%============== impressing test inputs on the trained network==================

```
%=====================================================================
%this is the testing phase through impression of test inputs
%---------------------------------------------------------------------
function main(trno)
%Reads the test vectors from files
[a,b]=readData(trno,2);
ain=a;
[ar,ac]=size(a);

%Loads the stored weights after training & structure of the ANN
dir1='E:\shortcuts&files\recentdocs\May31\Back_Prop_Alg\ntkParam\';
filename1='weightmatrix';
filename2='norms';
stri=num2str(trno);
```

```
fmt='.mat';
loadfileW=strcat(dir1,filename1,stri,fmt);
loadfileN=strcat(dir1,filename2,stri,fmt);

load( loadfileW, 'W', 'B');
load (loadfileN, 'no', 'l');

%impresses the test inputs on the network
for c=1:ac
    for i=1:no(1)
        Y(i,1)=a(i,c);
        V(i,1)=a(i,c);
    end

    for k=1:l-2
        for j=1:no(k+1)
            w=W(1:no(k),j,k);
            y=Y(1:no(k),k);
            V(j,k+1)=y'*w;
            Y(j,k+1)=logsig(V(j,k+1)-B(j,k+1));
        end
    end

    k=l-1;
    for j=1:no(l)
        w=W(1:no(k),j,k);
        y=Y(1:no(k),k);
        V(j,l)=y'*w;
        Y(j,l)=purelin(V(j,l)-B(j,l));
        O(j,c)=Y(j,l);
    end
end
x=1:size(a,2);
%plot of predicted & actual descriptor vectors
figure;
plot(x,b(1,:),'-*gre',x,O(1,:),'*blu-');
```

C.2 Implementation each step using the functions in B.1

%Denoising and/or Smoothing

```
strc2='0';
dir1='C:\Users\BELOVED\Desktop\ao1_unmasked\';
fmt='.tif';
pwd % PWD-displays the current working directory.

%=================================================================
for i=6:9
    for j=1:3
        strc1='dev_84';
        stri=num2str(i);
        strj=num2str(j);
```

```
        strc=strcat(dir1,strc1,strc2,stri,strc2,strj,fmt); %directory
        u=imread(strc);


num_iter = 10;
delta_t = 1/5;
kappa = 45;
option = 1;
ad = anisotropicdiff(u,num_iter,delta_t,kappa,option); %diff.filter

r = mat2gray(ad); %converts matrix to gray scale image
dir2='C:\Users\BELOVED\Desktop\denoised\';
output=strcat(dir2,strc1,strc2,stri,strc2,strj,fmt);
imwrite(r,output,'tif'); %writes the denoised image to file

    end
end
```

%Segmentation

```
close all;
clear all;
strc2='0';
dir1='C:\Users\BELOVED\Desktop\denoised\'; %denoised image directory
fmt='.tif';
pwd % PWD-displays the current working directory.

%=========================================================================
for i=6:9
    for j=1:3
        strc1='dev_84';
        stri=num2str(i);
        strj=num2str(j);
        strc=strcat(dir1,strc1,strc2,stri,strc2,strj,fmt);
        u=imread(strc);
k=5;
[mn,overlay]=k_segment(u,k);
r = mat2gray(overlay);
dir2='C:\Users\BELOVED\Desktop\segmented\';
output=strcat(dir2,strc1,strc2,stri,strc2,strj,fmt);
imwrite(r,output,'tif');

  end
end
figure;
imshow(output);
colormap(jet);
```

% Feature/Descriptor Vector Extraction

```
close all
%-------------------------------------------------------------------
TrainingObjectsPath =
uigetdir('C:\Users\BELOVED\Desktop\ROI\pca_data_segm', 'Select
training database path' );
%=======produce descriptor (feature) vector for each drought object
strc2='0';
dir1='C:\Users\BELOVED\Desktop\ROI\';
s='\';
strd=['low_segm    ';'medium_segm ';'severe1_segm';'severe2_segm'];
strf=['_low   ';'_medium';'_sev1  ';'_sev2  '];

fmt='.tif';
pwd   % PWD-displays the current working directory.
errmax=0.0;
%===================================================================
for z=1:4  % to choose the directory of a class
    count=0;
for i=6:9
    for j=1:3
        strc1='dev_840';
        stri=num2str(i);
        strj=num2str(j);

InImage=strcat(dir1,strd(z,:),s,strc1,stri,strc2,strj,strf(z,:),fmt);
        im=double(imread(TestImage));

    T = ImageMatrix(TrainingObjectsPath);
    [m, A, Eigenobjects] = EigendroughtCore(T);
    d = FeatureVector(InImage, m, Eigenobjects);
    filename=strd(z,:);
    dir2 ='C:\Users\BELOVED\Desktop\ROI\features_segm35\';
    fmt2='.txt';
    saveFileName=strcat(dir2,filename,fmt2);
    myfile = fopen(saveFileName ,'a'); %'a' opens a new file
    fprintf(myfile,('\r\n\r\n'));
    fclose(myfile);
    d=d'; % just to represent in the text file row wise
    dlmwrite(saveFileName,d, '-append', 'delimiter', '\t',
    'precision',3);

   end
  end
end
```

% Prediction of the descriptor vectors in the time series

```
clc;
dir1='E:\shortcuts&files\recentdocs\AOI\'; %dir. of test features
filename='features';
strd='\severe1';
fmt='.txt';
sel=input('enter the number no of features [20 25 30 35]');
choice=num2str(sel); %selection of the feature vectors
filedir=strcat(dir1,filename,choice,strd,fmt);
featur=dlmread(filedir);
choice=input('Training or Testing [1 or 2]?');

%specify the number of hidden layers and corresponding nodes
if choice==1
l=input('Enter no. of hidden layers ');
l=l+2;
for i=2:l-1
    no(i)=input(sprintf('Enter no of nodes in hidden layer %d: ',i-1
));
end
Emax=input('Enter max expected error Emax ');
tic; %starts a stop watch timer to measure elapsed time
for q=1:3
    BackPropagation (q,l,no,Emax); % call for training algorithm
end
toc;  %end of stop watch for elapsed time during training
else
 for p=1:3
    main(p);   %call for testing function
end
end
```

% Post-processing to reconstruct the drought objects from the descriptors

```
clc;
close all;
clear all;
TrainDatabasePath = uigetdir('C:\Users\BELOVED\Desktop\ROI\data',
'Select training database path' );
dir2 ='C:\Users\BELOVED\Desktop\ROI\features35\';
fmt2='.txt';
filename='severe1';
d=dlmread('E:\shortcuts&files\ROI1\results\for separate_four
lag\actual.txt');
TestImage1='E:\shortcuts&files\ROI1\severe2\dev_840701_sev2.tif';
TestImage2='E:\shortcuts&files\ROI1\medium\dev_840601_medium.tif';
e1=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0];
for choose=1:2
M = imageMatrix(PathOfTrainObjects);
```

```
[mn, A, Eigendroughts] = EigendroughtCore(M,15);

%=================================================================

RecoveredImage = mn + Eigendroughts*d;
%mn is the mean image, Eigendroughts is the eigenvector

 N=irow*icol;

% Reconstruction mean squared error
e1(choose, no) = sum(sum((InImage-RecoveredImage).^2))/N;

ReshapedImage = reshape(RecoveredImage,11,11);
figure;
subplot(2,2,1 )
imagesc(InputImage);
title('Original image','fontsize',12);
subplot(2,2,2)
imagesc(ReshapedImage'); colormap('gray');
title('Reconstructed image','fontsize',12)
end
end
x=1:size(e1,2);
figure;
plot(x,e1(1,:),'o-r',x,e1(2,:),'blu-*');
xlabel('Number of Principal Components '),ylabel('Average
Recostruction Error');
legend('For Image in the Training ','Image not in the Training');
title('Average Reconstruction Error Vs Number of Principal Components
');
```