Addis Ababa University

Addis Ababa institute of Technology

School of Electrical & Computer Engineering

Multilingual Text Detection and Script Recognition from Video Scene using Deeplearning

By: Kirubel G/hiwot Salfore

October 10, 2019

Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa institute of Technology

School of Electrical & Computer Engineering

Multilingual Text Detection and Script Recognition from Video Scene using Deeplearning

By: Kirubel G/hiwot Salfore

Advisor: Menore Tekeba

A thesis Submitted to the School of Electrical and Computer Engineering in partial fulfulment of the requirement for the Degree of Master of Science in Computer Engineering

October 10, 2019

Addis Ababa, Ethiopia

Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

Multilingual Text Detection and Script Recognition from Video Scene using Deeplearning

By: Kirubel G/hiwot Salfore

Approval by: Board of Examiners

Dr. Yalemzewd Negash	_____
Dean, School of Electrical and Computer Engineering	Signature
Menore Tekeba	_____
Advisor	Signature
_____	_____
Internal Examiner	Signature
_____	_____
External Examiner	Signature

October 10, 2019

Addis Ababa, Ethiopia

Abstract

Scene Texts occur more frequently in most videos which may contain crucial information. The information may have contents such as location and time. In Ethiopia most information on the streets are posted using Ethiopic (Geez) and Latin Scripts. In our Research work we have studied Multilingual Text Detection, Script Identification and Character Recognition from Video Scene using Deep Learning Neural Network Model.

The Videos being captured by the digital camera are processed and Keyframes are extracted using Keyframe Selection Algorithm, Text regions are detected by using Trained Convolutional Neural Network and those text regions which are found by bounding box regression are cropped out by taking their bounding box values. The use of Faster R-CNN that consists of dropout layer for text detection has achieved a 91% of precision, 92.9% recall and an execution time of 7.5 sec during testing the network. After taking those cropped text blocks, scripts are classified or identified by using a trained network through transfer learning into their script classes. Following the script identification Line Segmentation, Word segmentation and Character Segmentation using Horizontal and Vertical Projection profile are performed which are the preprocessing steps for Optical Character Recognition, where script identification has achieved 88.5% of accuracy without the use of dropout layer and 93.3% of accuracy with the use of dropout layer. The final phase of this work includes character recognition which lies on the previous text detection, and script identification phases, different epochs were considered during training the network to maximize the efficiency of the network to recognize characters. The network that was trained with an epoch size of 200 has achieved 0.0076% of error during testing. This shows that maximizing the number of epochs during setting the training options improves the character recognition performance while decreasing the error value to the minimum value.

Keywords: Faster R-CNN, Deep Learning Neural Network, Optical Character Recognition, Alexnet

Declaration

I, the undersigned, certify that research work titled by Multilingual text detection and script Recognition from Video Scene using Deeplearning is my own work. The work has not been presented elsewhere for assessment. Where material has been used from other sources, it has been properly acknowledged.

Kirubel G/hiwot Salfore:

Signature: ———

Date of submission:

Place: Addis Ababa

This thesis has been submitted for examination with my approval as a university advisor.

Advisor: Menore Tekeba:

Signature: ————

Acknowledgment

I prioritize to thank My God for being with me all the time throughout my life. As the Holy book Says “In all the ways acknowledge him, and he shall direct the paths”.

My deepest gratitude also goes to Mr. Menore Tekeba, without his help and support what would have happened to My Thesis paper? His doors are always open whenever I dig a problem or had some question, limitless valuable ideas, friendly treatment and Supportive Advice as a Father.

I will not stand here without the help and Support of My Mother. Without her support and full of love I would have never been ‘me’.

Contents

List of Acronyms

CAMSHIFT: Continuously Adaptive Mean Shift Algorithm

CNN: Convolutional Neural Network

FN: False Negative

FP: False Positive

ICDAR: International Conference on Document Analysis and Recognition

MSER: Maximally Stable Extermal Regions

OCR: Optical Character Recognition

R-CNN: Region based Convolutional Neural Network

RPN: Region Proposal Network

SWT: Stroke Width Transform

SVM: Support Vector Machine

tanh: hyperbolic tangent

List of Figures

List of Tables

List of Algorithms

Chapter One

Introduction

1.1. Background

The availability of Digital Electronics such as Smartphones, Webcams, and Digital Cameras also electronic social media such as YouTube and Facebook changes and influences the way we retrieve and analyze information. Nowadays the total number of people who are using YouTube are about 1, 300, 000, 000 where 5 billion videos are being watched on YouTube every single day by 3 million visitors per day. As we can see from the statistics the number of videos being captured and uploaded to the internet is growing very large. However, extracting one of the useful information (Text) from those video remains a challenging task. The text that is found in such videos holds the information like the name of a person, places, organization, date and time.

Considering the access and utility of Digital Video Cameras, it is important to understand and extract text within videos. Generally, text can be divided into two classes, scene text and overlay text (Caption) [1]. Captions are an artificially superimposed text which are embedded during post-editing of the video while texts which are embedded naturally in images or videos that are captured during video recording are Scene texts.

In most video scenes, naturally occurring texts (Scene text) appear frequently than Caption Texts that hold information such as name of place, street, Organization, Brand. Extracting such information is a little bit difficult than Caption text as a result of arbitrary orientation, Size, Background Color, Font style, different light intensity and font color.

Scholars in this field have proposed different mechanisms to detect, localize, segment, identify and recognize scripts from Videos and Images for different purposes such as Recognition of text and its translation into a user-defined language [2], License plate detection & recognition [3], Video indexing and retrieval [4].

Different algorithms such as MSER, SWT, connected component-based methods, Hybrid approaches (texture and edge-based) and Regression-based [5] [6] [7] [8] [9] [10] [11] were proposed and implemented for text detection.

Text detection and extraction problem can be subdivided into the following sections: (1) detection, (2) localization, (3) tracking, (4) extraction and enhancement (5) OCR.

In order to save time and memory, redundant frames are removed from the extracted frame through a method known as Keyframe selection. Video signal is basically a composition of multiple Frames and extracting such frames would enable us to extract crucial information found within the video.

In this paper, we have proposed a text detection approach by using R-CNN (Region-based Convolutional neural network) which is inspired by the work of [11] [12]. There is a big difference between Objects and Texts and this difference can be viewed in terms of Diversity, Size and Aspect Ratio [11].

1.2. Statement of the Problem

There are several challenges in detecting scene text from video. Diversified orientation of texts in different location of the video makes it difficult to detect text exactly. In some cases, color of scene text and their background become similar which makes it hard to differentiate text from the background. Also, complex backgrounds cannot be isolated easily from text which reduces the efficiency of text detection algorithms. Low resolution images, blurred images and Noise can be seen as factors that may introduce degradation in text detection due to text detection algorithms extracts noise and non-text features as text features which results in poor detection performance.

Most information in Ethiopia now a days are posted on boards that consist of two types of scripts which are Latin and Ethiopic. Recognizing characters from information posted on boards is difficult as a result of optical character recognition systems are suited for recognizing characters which are found in a specific script category.

Even if different researches have been conducted on the recognition of Ethiopic or Geez characters there is still a gap Text detection performance from videos.

Video Search is a problem that needed to be solved. In video uploading sites sometimes the name of video files is not associated with the content of the video. If text found with in a video can be extracted and indexed search results would be returned efficiently.

1.3. Research Questions

Question of interests that has to be answered by this Thesis are

- The Use of CNN for Text Detection with preprocessing
  Does the usage of Convolutional neural network and preprocessing the input images improve the detection performance from videos?
- The Application of Transfer Learning in Script Classification
  Can we achieve a comparative performance improvement of Script identification from Multilingual scripts by implementing Transfer learning?
- Can we perform character recognition from scene videos with a better performance with Deep Learning models?

1.4. Objectives

1.4.1. General Objective

The objective of this research work is to develop Text detection, Script Identification and Character Recognition method based on a Deeplearning artificial neural network model.

1.4.2. Specific Objective

- Measuring the Performance of Deeplearning neural network in Text Detection.
- Implementing Transfer learning Technique in Script Identification and Measuring the implementation in Script identification.
- Evaluating the Performance of recognizing Ethiopic scripts from Videos using OCR.
- Comparing our system with other system that extracts textual feature from Video frames specially with rule-based systems.
- Compare the identification metrics of this work which uses CNN with SVM.

1.5. Significance of the Study

- This study can offer a research output as a performance metrics in each Testing phase that includes Text detection, script identification and Character Recognition for Researchers in the field.
- Besides the numerical outputs it will play a role in explaining the challenges and stages of Text detection, script classification and character recognition from Video.
- Serves as an opening door for real time OCR Application Development which is useful for blind people with Multilingual Characteristics.

1.6. Contribution of the Thesis

As Researches should focus towards bringing solutions to the lives of our society, this research has the following contributions. This work is an extension of text detection, script identification and Character Recognition from Video with the following contributions. This work explores Deep Learning Neural Network Architectures and Proposes a method by including image preprocessing techniques which are suitable for text detection and character recognition. The proposed method includes a different approach other than the previous works conducted for script classification which achieves a very good script classification accuracy than the baseline research works. We have enabled character recognition from scene videos by implementing a deep learning neural network model and achieved a very good character recognition performance.

1.7. Methodology

The Technical Steps followed in this Thesis work is shown in Figure 1.1. The Evaluation metrics that are going to be used in this research work is Precision and Recall, Accuracy and Classification Error for Text Detection, Script Identification and Character Recognition Respectively.

1.7.1. Data Collection

The dataset used in this research work is collected for two phases: Training the Neural Network to detect text within videos and the second phase is for training the Artificial Neural Network to recognize characters from the detected text blocks. In the first phase training data and test data

were prepared from Recorded Videos using digital camera from streets of Addis Ababa while in the remaining phase, that is recognition, the training data and test data were used from the work of [1].

1.7.2. Designing and Implementation Tools

- Literature Review: Reading Books, Journals, Articles, Conference Publications, and materials that helps to conduct this thesis work successfully.
- Preprocessing Data: Making the collected data appropriate for Feature detection and extraction using Convolutional Neural network. MATLAB 2018a is used for preprocessing the extracted frames from the videos. The preprocessing step involves the conversion of RGB to Greyscale color format in order to extract text regions easily from the video frames.
- Training each of the Deep Learning Neural Networks (Faster R-CNN, Transferred Network and LeNet-5) to detect, classify scripts and to classify characters in to their letter candidate.
- Implementation: MATLAB 2018a software has been used to implementing the proposed system.
- Experimentations and Discussion: Discussion on the Results, conclusions and the future directions.

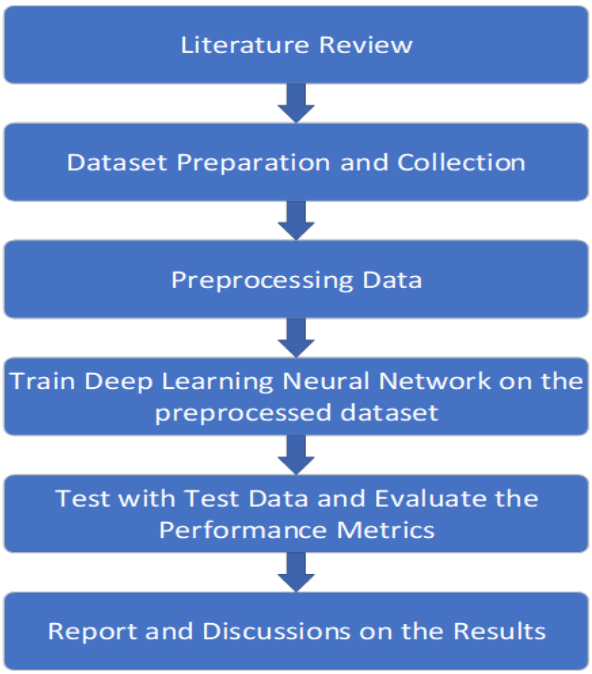


Figure 1. 1 Methodology Followed to conduct the Research

1.8. Thesis Outline

The thesis is organized as follows: Chapter two is about literature review on text detection, feature extraction, Script identification and Character Recognition from images and videos that were done both in local and abroad by different scholars. Chapter Three is about detailed understanding of different Text detection mechanisms used in detecting text from video frames and images, script identification techniques and models used for character recognition tasks. Chapter Four is about the design and implementation of this research. Chapter Five talks about the dataset preparation, evaluation metrics and experimentation. The final chapter provides conclusion and future work.

Chapter Two

Literature Review

Previous Research works in Text Detection, script identification and character recognition are presented under this section. Text Detection mechanisms are categorized under the following categories as Texture Based, Connected Component-Based, Region-based, Hybrid based and Regression based. Researches on script identification have also reviewed in three categories namely; SVM based script identification, LBP based script identification and Neural Network-based script identification. Finally, research works conducted for character recognition has been reviewed in which uses Neural Network models to recognize characters.

2.1. Texture Based Approach

Texture based method utilizes the observation that text regions have more distinct textural properties than the background region. In [2] the researchers have proposed a method which uses SVM to analyze texture feature from images and classifies the pixel located at the image into text and non-text by and later CAMSHIFT algorithm is applied to identify text regions as it is shown in the figure below.

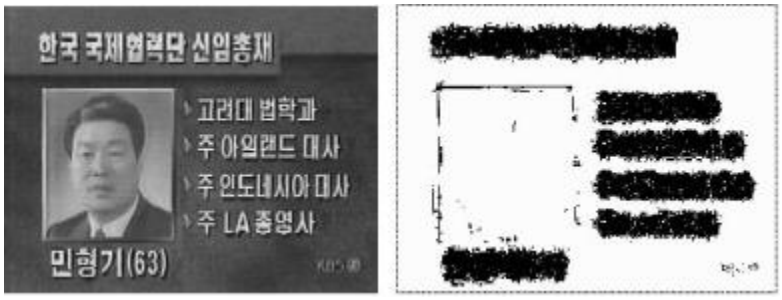


Figure 2. 1 Example of Texture Classification with input image on the left and classification result on the right

The shortcoming of this approach was the computational time increases when to classify the pixels on image size of 640x480 resolution and when this approach is compared to another classification approach that classifies pixels located within the search windows involved in CAMSHIFT that

only takes 0.45 seconds. Also, the approach is not efficient in classifying text with very small contrast.

2.2. Connected Component Based Approach

The connected Component-based approach extracts regions from the image and uses geometric constraints to rule out non-text candidates [3]. In [3] Maximally Stable Extermal Regions are efficiently extracted as a basic letter candidate and enhanced using Canny Edges obtained from the Greyscale image as shown in the figure below.

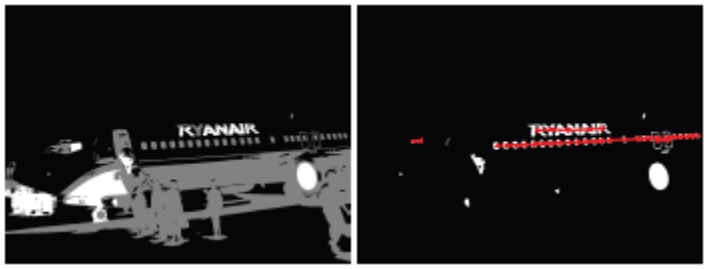


Figure 2. 2 Detected MSER on the left and Letter Candidate on the Right

After extracting enhanced MSER Geometric filtering has been conducted by rejecting CC which has a very large and very small aspect ratio, later SWT was applied to the binary image of its stroke width image. Finally, text line formation was done using clustering of pairwise connected letter candidate. The shortcoming of this approach is that MSER has been sensitive to blur so that text detection performance can be lowered if the blur is introduced into the input image, due to this we have left this algorithm and search for another efficient method for text detection. The excellent feature they come up with their work is that they have tried to improve text detection performance on images with blur by exploiting complementary properties of MSER and Canny Edges.

2.3. Region Based Approach

Region-based methods use the properties of the color or gray-scale in a text region and then group small components into successive larger components until all regions are identified in a video image [4]. In the work of [5] a coarse to fine strategy which employs detection of candidate text regions

using stroke map and morphological operation following by Text localization to identify text regions accurately. Their work focuses on motion pictures where method achieves good text detection metrics. However, there is no clear point that makes the comparison with other works which can be considered as a point at which their work can be a better one than the previously conducted works. Here we cannot see that it outfits another state of the art method in a text detection task.

Stroke width Transform (SWT) is a local image operator which finds per pixel the width of most likely stroke containing the pixel [6]. In [6] they have been used Stroke Width Transform (SWT) to find stroke width information for each representative pixel and merge neighboring pixels with similar stroke width into connected components which forms letter candidates.

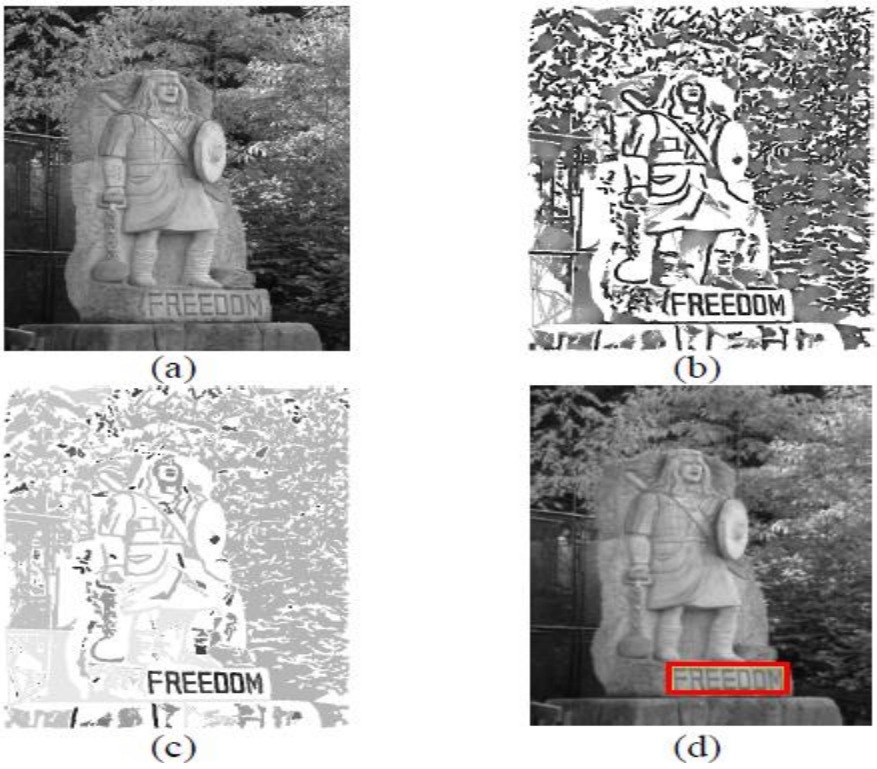


Figure 2. 3 The SWT converts the image *[6]* (a) from containing gray values to an array containing likely stroke widths for each pixel (b). This information suffices for extracting the text by measuring the width variance in each component as shown in (c) because text tends to maintain fixed stroke width. This puts it apart from other image elements such as foliage. The detected text is shown in (d)

Although their method achieves better detection metrics it cannot detect text with large font size and it doesn't show its capacity on how to detect natural scene texts with different orientation. Due to this shortcoming we have skipped to use this method in our research work.

2.4. Hybrid Approach of Texture and Edge based Approach

The hybrid feature-based method uses the combination of two or more feature together to detect text in the image. In [4] texture and edge-based method together have been used to detect scene and overlay text in videos. After extracting hybrid features each window was classified as Text window or background window by employing SVM as classifier. Morphological Operation has been done to precisely locate text the text regions since there are false detections which can't be removed. The shortcoming of this work is that the proposed method cannot localize the non-Horizontally (Vertical) Aligned texts which appear in Chinese and Japanese News Video. Therefore, we left out this method due to it is not capable of detecting vertically aligned texts and of course, our focus is not on detecting Chinese and Japanese Texts.

2.5. Regression Based Text Detection Approach

Region-based Convolutional Neural network (R-CNN) has been implemented with multiple Region Proposal network (RPN) to detect text from images in [7]. This work has made a difference from the previous original Faster R-CNN model that generates ROI by one RPN, but the new feature introduced in this work is that, they have proposed and implemented a method which generates ROI by multiple RPN. One of the big advantages of using R-CNN network is the use of RPN in the network architecture that enabled the detection of texts regardless of their size.

Although it is not familiar with this thesis Faster R-CNN has been used in object detection task in the work conducted by [8]. In this work, it was observed that convolutional feature maps were used by the Faster R-CNN as well by RPN where the R-CNN network used to detect the regions and RPN proposes region proposals. The effectiveness of their work has been shown by implementing RPN which is designed to effectively predict a wide variety of region proposals with different aspect ratio and scales. Their work was different than techniques that use pyramids of images that run on at all scales and pyramids of filters with multiple scales which runs on feature

maps to detect text due that they employed pyramids of reference boxes in the regression function as shown in figure 2.4.

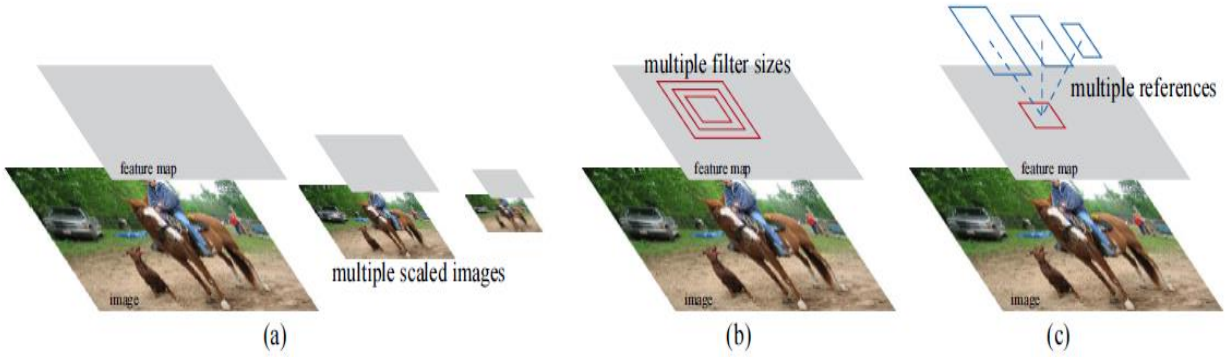


Figure 2. 4 Different schemes for detecting text [8] (a) Pyramids of images and feature maps. (b) Pyramids of filters with multiple scales/sizes (c) pyramids of reference boxes

Even though this scheme is used for object detection we are inspired to deploy the network with preprocessing techniques to detect text as objects from scene images as it is efficient in detecting objects with different aspect ratio and scales.

Another work carried out by [9] has shown the use of corner response feature map to detect candidate text regions. By employing corner feature map technique and using high resolution and high quality images high precision and F1 measure score on Microsoft common test set, high recall, Precision and F1 measure metrics on TV News Test set and finally high Precision, Recall and F1 Measure metrics on YouTube Test set were achieved than other technique which uses the TV news test set and YouTube test set. If the low-resolution dataset is introduced into their work, the system couldn't achieve good text detection metrics as a result of their training set is dependent on high resolution and high-quality images.

Arbitrary oriented scene texts have also been detected with superior performance by [10] where they have been using Rotation Region Proposal Networks (RRPN) to detect text at arbitrary positions. The RRPN is designed to generate inclined proposals with texts having orientation angle information in bounding box regression. The orientation angle information is used to make proposals which are accurately fit into the text region in terms of orientation. Finally, the Rotation Region of Interest Pooling layer (RROI) proposed text regions which are arbitrary and these proposals are used by text region classifier. The powerfulness of this work has been shown by

ensuring the computational efficiency of the arbitrary scene text detection when compared to previous works conducted for text detection. This work has been evaluated on three publicly available datasets known as MSRA-TD500, ICDAR2015, and ICDAR2013 and it was found out to be that they have achieved maximum Precision, recall, and F-Measure metrics when compared to [11] [12] [13]. From the different works conducted so far for text detection, this one is the best approach for scene text detection and should be followed by others as a technique for text detection.

2.6. Script Identification

In [14] Maximally Stable Extermal Regions (MSER) and Stroke Width Transform (SWT) algorithms are used to extract text regions from video and images. After extracting texture features are computed using Local Binary Pattern (LBP), along with this SVM has been employed to classify text region from non-text regions. In the segmentation phase of this work, the horizontal projection profile was used followed by a vertical projection of words. Finally, in his work, the resulting text words are categorized into their respective classes using SVM. The use of LBP with SVM has achieved a good identification accuracy but in the research work, it is mentioned that if some other classification mechanisms can be exploited a better result can be found.

LBP has been employed in the work of [15] to describe the script stroke structure. Since stroke directions became clearer in white and black in its preprocessing stage the images are converted into Greyscale image using Otsu Threshold. Horizontal projection was used to segment text lines horizontally. Finally, SVM or Least Square Support Vector Machine (LS-SVM) was employed to make binary decision and multiclass classification for script identification.

In [16] employed a novel Deep Neural Network structure to efficiently identify scripts from image. Their design was targeted to exploit two factors known as image representation and spatial dependencies within text lines. To make the script identification they have brought Convolutional Neural Network and Recurrent Neural Network into one end to end trainable network. While the Convolutional Neural Network used to generate rich image representation the Recurrent Neural Network efficiently analyzes long term spatial dependencies on the other end. Several experiments have been conducted on different datasets including SIW-13 and CVSI2015 and their approach

achieves a potential performance when compared with the previous works. In their work the strength of using CNN has been shown in the task of image representation as a result of CNN made up of several layers of non-linear feature extractors where the network is controlled by varying the width and breadth of stacked structure. And the use of discriminative model known as Recurrent Neural Networks in their work achieves a better result than the previous works which uses conventional models for identification tasks, where previous models produce non-normalized likelihood while discriminative models are normalized.

SVM Classifier has been exploited in script identification work of [17]. Feature extraction techniques, namely Zernike moments, Gabor and gradient features were used to extract features which are going to be used for script identification task. Super resolution and skeletonization techniques were employed to tackle the problem that comes with the Video. The application of preprocessing techniques to identify scripts has shown a comparative performance improvement rather than the identification method that doesn't incorporate preprocessing technique. But their script identification performance gets lower when fewer characters appear in the image.
Bag of Local convolutional Triplets were employed for script identification from scene images in [18]. These feature triplets are created by combining feature descriptors extracted from the input image. Their work was evaluated on three publicly available datasets for script identification in Video captions and it has shown outperformance than the base line works carried out and evaluated on the same datasets. The effectiveness of their approach in discriminating scripts lies on the use of triplets of local Convolutional features extracted from the trained convolutional Neural Network however their result shows that the classification error rate will decrease as a result of more number of triplets usage in identification of scripts and it is dependent on smaller number of triplets to gain an optimal result in script identification.

2.7. Character Recognition

A Deep learning artificial neural network was employed in Ancient Ethiopic Manuscript recognition in [19]. In this work Restricted Boltzmann Machine (RBM) was used as a Deeplearning neural network and trained with Greedy layer wise unsupervised strategy. The

developed system consists of Image Acquisition, Preprocessing, Character segmentation, classification and recognition. In this work, they have achieved an Excellent Character recognition performance however the method focuses on ancient manuscript characters and the trained network was very suitable for recognizing such characters so that we have searched for another Deeplearning architecture to recognize characters from scene images.

Another model in the deep learning architecture known as LeNet-5 has been exploited in the work [20] to recognize handwritten characters.by varying the different activation function known as Relu(Rectified Linear Unit) and tanh they have obtained the highest recognition performance than the previous works conducted using other techniques [21] [22] [23] [24]. By looking at the powerfulness of LeNet-5 deep learning neural network architecture we are inspired to conduct our character recognition using it.

The Effectiveness of Convolutional Neural Networks in the task of Character recognition has been clearly shown in the work conducted [25]. When the result presented by [25] compared with previous works that uses multilayer perceptron which has one hidden layer the error rate was found out to be 4.5%, the same Multilayer perceptron with two hidden layer achieved an error rate of 3.05%. Another models LeNet-1 as well as LeNet-4 achieved an error rate of 1.7% and 1.1% respectively, boosting technique with several instances has been introduced into LeNet-4 to maximize character recognition performance that achieves an error rate of 0.7%. LeNet-5 has achieved an error rate of 0.95% which is somehow error prone but computationally better than LeNet-4 architecture which would be three times computation as compared to LeNet-5.

Finally we are inspired by the work conducted by [7] [8] and to employ Faster R-CNN network to detect text from Video Frames as they are powerful in computation and reduces the computational time to detect text. In [16] the use of Convolutional Neural network has shown a significant improvement in the scene text script classification task than other works conducted to classify scripts. By looking at this performance we are going to employ Convolutional Neural Network Model by using a new scheme known as Transfer Learning which shares the weights of the previously trained network to classify millions of images. For the character recognition phase, the use of LeNet-5 increases the recognition performance as it is shown in the work conducted by [26] so we propose the network model to be LeNet-5 to recognize characters with high performance metrics.

Chapter Three

Related Works

3.1. Text Detection

Text detection is the process of detecting and locating those regions that contain text from a given image and is the first step in obtaining textual information [2]. As a tool for transmitting information from time to time and generation to generation writing is very important in our daily life. In most of images and videos, texts are found which carry crucial information about places, street address, date, house number, organization names, brandings and related. Considering the information contained within the images and videos we are interested to detect, extract and Recognize Ethiopic (Geez) texts from Videos.

3.1.1. Scene Text

Scene Texts are texts which we can find naturally and captured by the camera. They are textual information captured by a camera as a part of a video scene and image describing the name of organization, brands, locations, date and time [27] [28] [29]. Scene texts are difficult to detect and hard to extract their feature as a result of characteristics of the Scene such as Movement, Affine Transformation, Lighting and Occlusion [30].

Major characteristics of Scene Texts are dynamicity of size, font, color and orientation as well alignment. Some portion of the text may be occluded [30].

Figure 3. 1 a Scene Text Example

As shown in figure 3.1, the textual information is embedded within the scene image which shows the name of the department. As indicated in the figure 3.1, there are different light intensities around every corner of the image and complex background associated with the text.

3.1.2. Multilingual Text Characteristics

Considerations that are being taken into account when studying the nature of Multilingual Text are Contrast, Color, Orientation, Stationary Location, Stroke Density, Font Size, Aspect Ratio, and Stroke Statistics according to [31]. And these characteristics are Categorized under two sections as Language Independent and Language Dependent Characteristics [31].

3.1.2.1. Language Independent Characteristics

- Contrast: It assumes text is characterized by high contrast against its complex background [31] [3].

- Color: When the color of text is compared to its background texts become lighter or Darker than the background Color [31].
- Orientation: Texts in scene images or videos can be found at any orientation. An arbitrary orientation of texts in scene image and videos can be considered as a major challenge to train neural networks for character recognition and text classification [32].
- Stationary Location: Scene text found with in video frames and images are stationary through time by their nature [14].

3.1.2.2. Language Dependent Characteristics

Both Latin and Ethiopic scripts belong to Alphabetic literals based on linguistic classification [14]. The differences in the following language dependent characteristic has an impact on Video Text Processing [31].

- Stroke Width Density: The number of pixels in a stroke that constitutes a measurement of its length, strokes in vertical, horizontal, left and Right diagonal directions of the image can be defined as Stroke Density [14]. The stroke density of that of English text is roughly Uniform [31].
- Font Size: There is an assumption that there may be a font size variation [14].
- Aspect Ratio: To form a meaningful word in both Ethiopic and Latin there must be a combination of characters found in both Ethiopic and Latin. But there are exceptional characters in Ethiopic script such as “nu” and “na” which are considered as a single letter and have meaning by themselves [14]. Therefore, there is no aspect ratio difference in both Ethiopic and Latin characters as a result of the difference in their scripting system except the type of text [14].

- Stroke Statistics: Both Ethiopic and Latin Scripts do not contain much intersections excluding i and j in English and all family of “ve” for Ethiopic much difference in stroke statistics is not observed in both Ethiopic and Latin Scripts [14].

3.2. Key Frame Selection

A video signal is basically viewed as a sequence of frames. A Keyframe is an image frame to represent a video by summarizing the content which is found in the video [33]. It reduces the time and memory usage also the computational cost during video data processing [33]. Keyframe algorithm has been used in [29] to avoid the redundancy of frames which have similar contents.

3.3. Text Detection using Faster R-CNN Deep Learning Network

3.3.1. Artificial Neural Network

An artificial Neural Network is an interconnection of single neurons where each neuron is used for information processing which is inspired from the trend of how Biological Neural Networks process information [34].

The configuration of an Artificial Neural Network differs in its application such as Pattern Recognition, object classification. In any Artificial Neural Network, a neuron or a processing element is known as Unit or Node. Each modeled neuron as shown in Figure 3.2 has a set of connecting weights (corresponding to synapses in biological neurons), a summing unit, and an activation function. The output of each neurons y can be computed as the weighted combination of input signals as it is given in equation 3.1. [19].

$$Y = \sum_{i=1}^{d} w_i X_i + b \qquad (3.1)$$

Where w_i is the weight of each connection between the input signal X_i and the hidden nodes in the network and b is the bias. The following figure illustrates a single neuronal model.

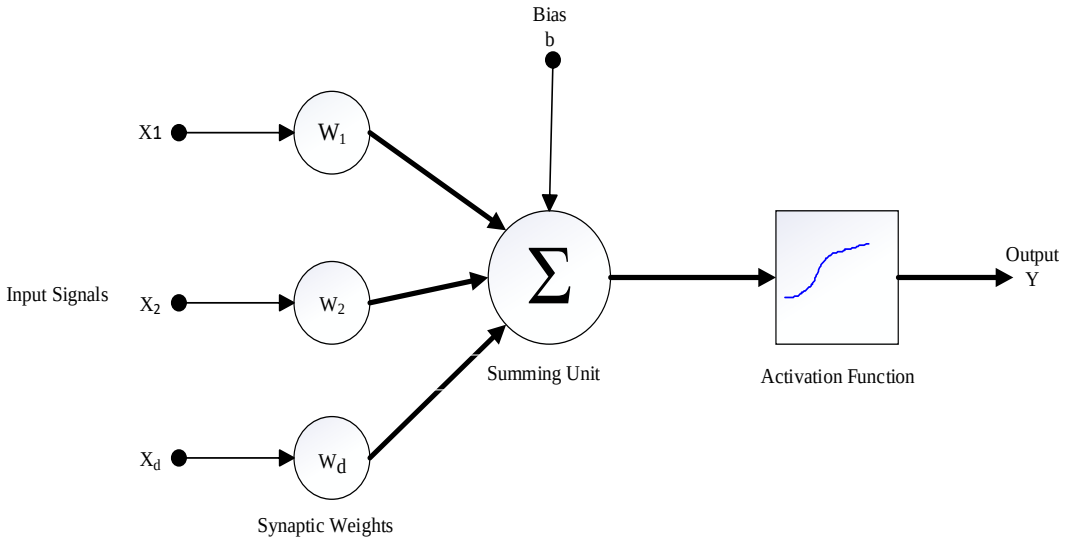


Figure 3. 2 A single Neuron Model

A given artificial neural network is defined by three types of parameters [34]:

1. Interconnection pattern between different layers of neuron.
2. The learning process for updating the weights of each connection.
3. The activation function used to convert the neuron's weighted input to its output activation.

Main characteristics of ANN are [34],

- they can perform tasks which linear program cannot perform,
- during processing of information, the network will not be stopped doing its task if an element of the neural network fails because of their parallel nature.
- What a neural network learned before or earlier will not be required to be learned again by the network.
- NN can be implemented for any application without any problem.
- NN needs training to be functional.
- Requires high processing time for large neural networks.

Multilayer Neural Network which is shown graphically in Figure 3.3 is one of the ANN architecture capable of classifying most patterns with an appropriate parameter. Character Recognition uses the backpropagation model or Multilayer Perceptron model which employs

supervised learning technique [19]. The following figure shows the architecture of Multilayer network.

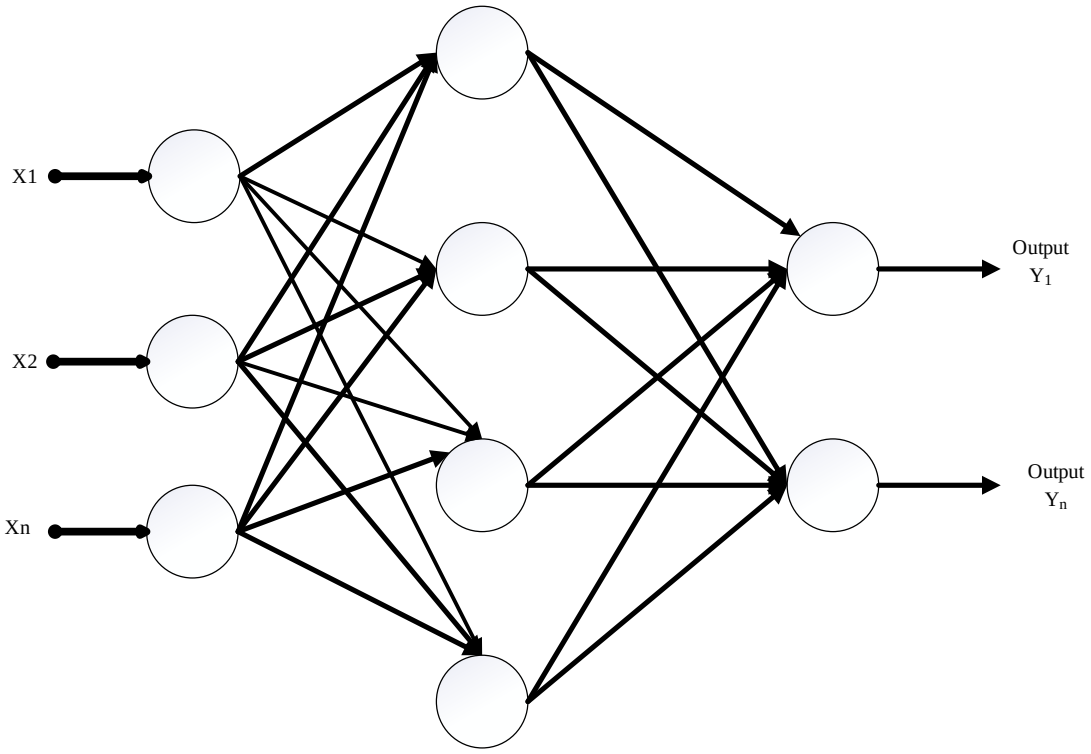


Figure 3. 3 Multilayer Neural Network Model

3.3.2. Deep Learning Neural Network

Deep learning, a very efficient technique for learning in depth networks. Which is derived from the Machine Learning discipline that is based on a set of algorithms that attempt to learn and model high-level abstraction in data by implementing multiple processing layers and composition of multiple non-linear transformations [5]. The main difference between deep learning and the traditional pattern recognition is that deep learning automatically learns features from big data rather than adapting handcrafted features [18]. A good example for such case is that a given image can be represented by a vector of intensity values per pixel, as a set of edges, regions of particular shape or etc [5].

Deeplearning as one of the class of Machine Learning Algorithms, it has been characterized in number of ways which are [19]:

- The application of many cascaded layers of non-linear processing nodes for feature extraction and transformation. Each of the successive layers takes the output from the previous layer as an input and process it. Algorithms can be seen as supervised or unsupervised which depends on the task the network is assigned, for pattern analysis unsupervised algorithm will be exploited and supervised algorithm for classification.
- Uses Unsupervised learning of multiple level of features or representations of data. It forms a hierarchical representation of feature where higher-level feature are derived from lower level features.
- Is one of the broad algorithms of Machine Learning field in the learning representations of data.
- It can learn multiple level of data representations which corresponds to different level of abstraction.

Deep learning neural network is a technical name for a collection of neural networks composed of several layers. Each of the layers are made of nodes [5]. A node or unit is a place where computation happens based on mathematical modeling. This network accept the data to be trained by combining Weights which simplify or dampen the weight, then the weights of the input are multiplied with the input values and summed up where the sum is passed through the nodes activation function to determine whether the signal progress to which extend in the network and affect the whole outcome.

3.3.3. Faster R-CNN for Text Detection

R-CNN is a neural network model based on Convolutional Neural network proposed by [8]. And it is widely used in Object recognition and other fields. R-CNN Architecture contains two networks inside, known as RPN and CNN. The usage of the RPN network is to distinguish all proposed bounding boxes on the extracted feature map that is computed by the convolutional layer [35]. The feature extraction in convolutional neural network is made by Convolution kernel, each neuron and the receptive field of the previous layer which makes it different from the traditional feature extraction method [35]. According to [35] Faster R-CNN network for object detection works in such a way that, first RPN proposes region proposals for labeled objects. It does this by having two separate outputs for each of the bounding boxes. The first one is the probability that

the detected box is an object which is referred as objectness score as this is used for filtering out bad predictions for object in the second stage. And the second output is the bounding box regressions for adjusting the proposed boxes to fit the object it is predicting.

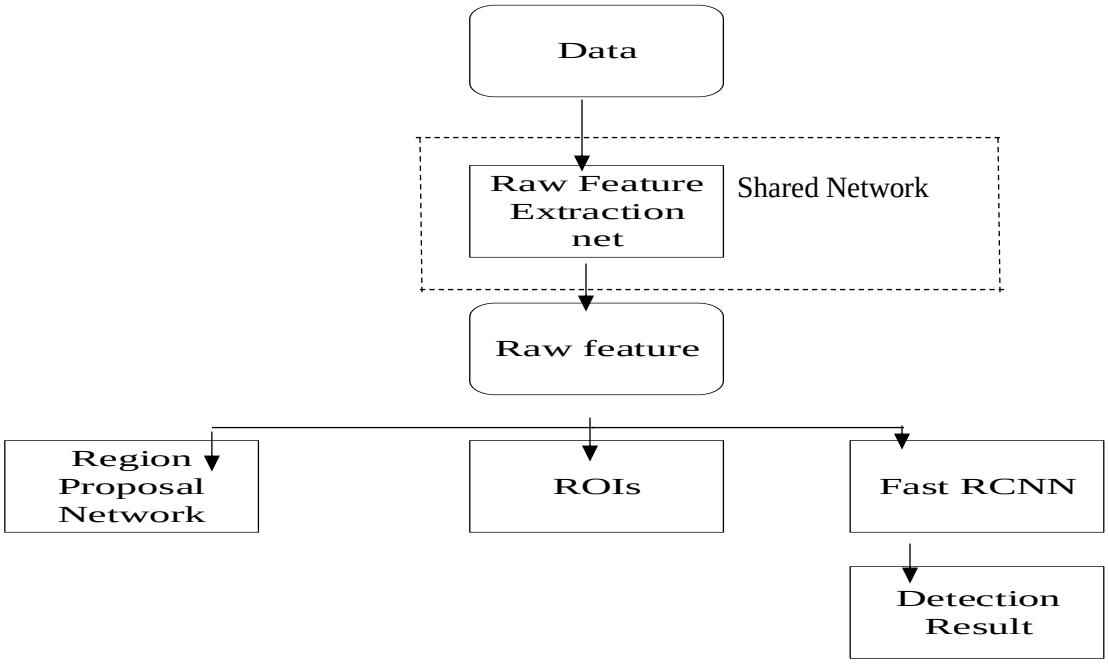


Figure 3. 4 Fast RCNN Training Flowchart *[35]*

To make sure the Network learns an accurate noise model, rather than fitting the deep network to the noisy label erroneously, an aggressive dropout regularization added to the softmax layer in [36] . Dropout regularization is used to overcome the problem of overfitting by implementing a dropout layer that randomly sets input elements with zero with a given probability. The probability for dropping out input elements(neurons) during training time is varying scalar in the range between 0 and 1. Inspired by the work [7] which achieves a good detection metrics we have proposed Faster R-CNN Network to detect text objects from scene images.

3.4. Preprocessing

Preprocessing on training image is the task of enhancing image quality and rendering the image to make it comfortable for better text detection, script identification and segmentation stages. The various preprocessing techniques we have implemented in our work are Greyscale conversion and Binarization.

3.4.1. Greyscale conversion

The training images that are used during the training of deep learning neural network were originally RGB in their color format. Greyscale conversion is used to convert RGB color format into greyscale format by forming the weighted sum of each of the RGB color format [19]. In the phase of character recognition, a preprocessing step should be included in order to recognize characters effectively. The use of preprocessing technique known as Binarization has been included in [37].

3.4.2. Binarization

Textual information is embedded within scene video which has crucial information about Places (Location), Organization names, Brands, date and time of events and so on. In video frames those texts appear against non-uniform and complex background and their color will vary due to uneven illumination the text [14]. In the work [14] they have implemented image binarization by using Niblack's binarization algorithm which calculates several thresholds for every pixel by applying specific formulae.

3.4.3. Skew Detection and Correction

The occurrence of skew in a given image can be viewed as two perspectives, one is skew can be occurred during capturing the video or Image using Digital camera and the second is, during writing the texts [19]. The reason for skew correction is primarily to have a better character recognition performance and secondly during segmentation text lines can be found effectively. From various skew detection and correction algorithm radon transform has found to be the best algorithm to detect and correct skewness of an image with high accuracy in the work conducted by [38] as it was less sensitive to noise . The application of Radon Transform on an image f(x,y) works by computing the projection of the image specified by theta angle. The resulting image is the sum of radon transform of each individual pixel, R(ρ,θ) as given in equation 3.2.

$$R(\rho,\theta)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dx\,dy \tag{3.2}$$

3.5. Segmentation

The major phase Preceding recognition of characters is Segmentation of Texts into individual Characters or Words. A well processed segmentation of each and individual symbols achieves a better recognition performance.

After detecting and correcting the skewness of a given image, in order to classify and recognize the characters, segmentation should be takes place. In segmentation of characters there are three steps, line segmentation, word segmentation and character segmentation. The first one involves the separation of each lines found within the detected bounding box which is followed by word segmentation where words which are found during line segmentation are separated in to a single unit and finally each character is separated using character segmentation [19].

3.6. Classification

The main aim of our research work in the classification phase is to classify Latin and Ethiopic scripts according to their script class. The task of classifying scripts into their script classes is divided into two parts in our research work. The primary task is to identify in end to end system where the texts are detected using trained convolutional neural network, preprocessed, noises are reduced, line segmentation and Text word segmentation are done, finally we have used a pretrained deep neural network on a different dataset for classification task and use the final layers of the pretrained network to classify our own word images by extracting features and classify them accordingly to their script classes. The secondary task is to classify cropped word images where the word images are cropped manually using cropping tools and the transferred work is used to classify those cropped word images.

Machine Learning Algorithms were known for classification of different researches related with Script identification [14] [39]. But their work can be improved if a deep learning network is implemented and tested on the same dataset as a result of it contain a depth network composed of millions of neurons. We have neglected Machine learning algorithm known as Support Vector Machine which was employed in identification tasks carried out by [14] [39].

3.6.1. Support Vector Machine -Based Methods

SVM algorithm was used to classify for both Text detection and Script Identification from Images and Video frames [14] . In their work they have implemented steps recommended by [40].

3.7. Optical Character Recognition

OCR is a technique related with translating handwritten, type written or printed Text characters into Machine Editable Text. Now a days OCR is widely used as a form of data entry from passports, invoices, bank statements, computerized receipts, mail printouts, business cards. Optical Character recognition Techniques extensively uses the methodologies of pattern recognition which uses unknown sample to classify into predefined class. There are four general approaches of pattern recognition namely (1). Template Matching (2). Statistical Technique (3). Structural Technique (4). Artificial Neural Network Technique.

1. Template Matching: this is the simplest way of Optical Character Recognition which matches the stored prototypes against the character to be recognized [41]. Technically the matching operation decides or determines the degree of similarity between two vectors such as Group of pixels, curvature, shape, etc. As well it is the way of finding the location of sub image called template inside an image [41]. After finding several corresponding templates their centers is going to be used to determine the registration parameter.
2. Statistical Technique: Usually, there are assumptions which are used as a basis for Statistical Technique, the prior one is the distribution of the feature set that is Gaussian or in the worst-case Uniform. The second one is there are sufficient statistics available for each class and the last one is, given ensemble of images {I} one is able to extract a set of features $\{f_i\} \epsilon$ F, i= {1, …., n} which represents each distinct class of pattern [41].

   Measurements taken from n features of each character unit can be thought to represent an n dimensional vector space and the vector whose coordinates corresponds to the measurements taken represents the original character unit [41].

3. Structural Technique: this technique can be seen as the recursive description of a complex pattern interms of simpler patterns based on the shape of the object [41]. It is also the representation of characters as the union of Structural Primitives. In this technique there is an assumption that the character primitives extracted from writing is quantifiable and one can find relation among them.
4. Neural Network Technique: Artificial Neural Network is composed of a number of parallel interconnections of Neural Processors it can do computation faster than the classical techniques [41] .

   In its nature it can adapt to the changes in data and learns the characteristics of the input signal.

3.7.1. Character Recognition Using Neural Network

Neural Networks have been considered as a good solution to resolve recognition problems, using this approach a large number of characters known as Training data are fed into the algorithm in order to infer rules automatically to recognize characters [42].

The invent of Deeplearning neural networks such as convolutional Neural Networks makes it easier to extract features from the training images using repeated convolutional layers specified in the architecture of the network. Training images are fed into the image input layer of the network by specifying the size of the input data and passed to the corresponding convolutional2d Layer to convolve the input image with filters and extract feature map from the given image. Among various convolutional Neural Network architectures LeNet-5 achieves highest recognition performance in [26]. In such network convolutional filters are initialized randomly and the cross entropy is used to measure the difference between true class and predicted class.

Chapter Four

Design and Implementations

4.1. System Description

The proposed Multilingual Text Detection and Script Recognition mtehod from Video involves training a deep convolutional neural network using Greyscale frames that were extracted and preprocessed from Videos and following this, scripts are classified into Ethiopic and Latin, finally another deep convolutional neural network is trained and recognize characters into predefined set of Geez (Ethiopic) Characters.

The detection of text from Videos, classifying scripts into their respective class and recognizing characters accompanied by a series of steps. This work is illustrated in the following figure 4.1.

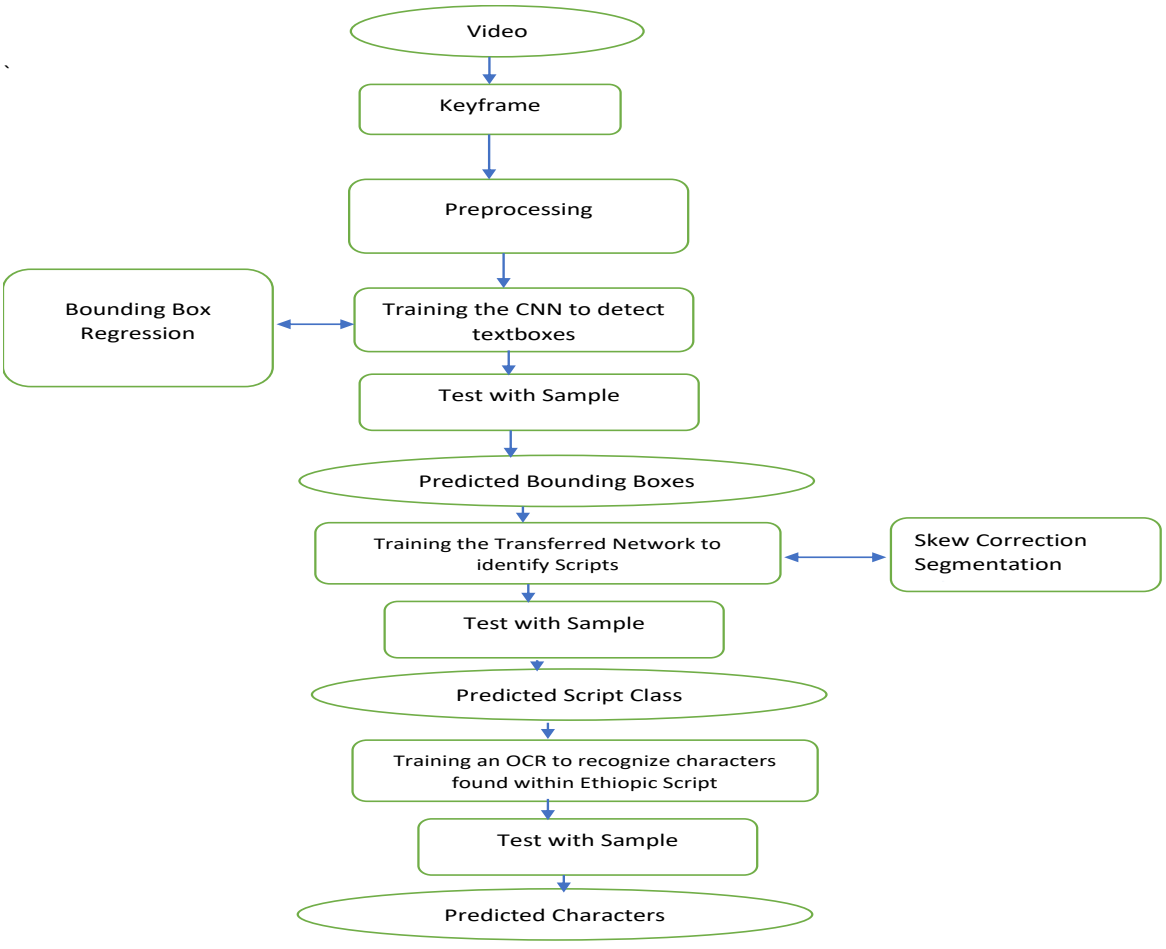


Figure 4. 1 Block Diagram of the Proposed System

4.2. Key Frame Selection

A single video is decomposed into multiple frames to process the required data frame by frame. Using Video Reader function in MATLAB the number of frames found within the video is calculated and frames are extracted. As it is proposed in the previous chapter keyframes are selected to reduce memory and time usage, this is done using Histogram difference algorithm. Frames that were extracted continuously are put together and histogram difference algorithm used to calculate the difference of the preceding and succeeding frames, if the difference of the two frames is greater than the Threshold value which is obtained by the following formula (4.1), the successor of the two frame is selected as a key frame.

Algorithm 1 Key frame selection using histogram difference *[43]*.

input: Total number of frames

output: Key frames

for i = 1 to Number of frames **do**

I ← k;

J ← k+1;

S← abs difference[i; j];

end

mean ← mean(s);

standard deviation std(s);

threshold ← standard deviation+ (mean _ k);

if S < threshold **then**

write image J as a key frame

end

$$\text{Threshold} = \text{Standard Deviation} + \text{mean} * b \; ; \qquad (4.1)$$

Where b is constant, most of the time the value of b is set to 4 to get the desired key frames.

4.3. Preprocessing

4.3.1. Greyscale Conversion

The extracted frames from the video are RGB in their color, in order to process them in the neural network they should be converted to Greyscale color format. We have converted the RGB frames into Greyscale using rgb2gray() MATLAB function.

Figure 4. 2 Greyscale image conversion

4.4. Text Detection by Faster R-CNN

Inspired by [7] the Deeplearning architecture selected for text detection from videos is Faster R-CNN which is based on region of Interest. As it is shown in figure 4.3 a predefined set of training images are prepared during the dataset preparation using Training image labeler, where those training images are extracted as a keyframe from the Video. The implementation of this Neural network starts with video frames from which we want to obtain a list of bounding boxes, a label assigned to each bounding box and a probability for each label and bounding box. The input layer of the network is defined as 227 x 227 x1 and those input frames from the video are passed through a pretrained Convolutional Neural Network ending up with a convolutional feature map.

Before conducting the training, we have configured the training options for the Network. MATLAB built in function known as trainingOptions is used for specifying the conditions used in training the network. The Faster RCNN Object detector trains the object detector in 4 steps. The primary two steps train the region proposal and detection networks used in the network architecture. The remaining two steps combine the networks from the last two steps where a single network is created for detection. In such network Checkpoint path is defined to have temporary location for all the training options. This path is crucial for fault tolerance during the training, if training is interrupted by power outage or if the system is failed, we can resume the training later from this path.

Region Proposal Network (RPN) in Faster R-CNN Network is used to get a predefined number of region proposals that contain Text by taking the features extracted from the previous Convolutional Neural Network. We have been putting anchors throughout the image to generate a variable length list of bounding boxes. Based on those anchors which placed throughout the image, we model the problem into two parts as: Does this anchor contain a relevant text object? How would we configure this anchor to better get the exact Text object? Therefore, the RPN takes all the reference boxes and outputs a set of good proposals for objects by having two different sets of output for every anchors, these outputs are; the probability that the predicted object is a text object (an objectness score) and the second output is the bounding box regression for adjusting the anchor to exactly detect the object it's predicting.

After the getting the region proposals our next task is to assign a class for a bunch of object proposals by taking bounding boxes. Region of interest pooling has been done so far in our work to extract anchors for each object proposals. These anchors are used by the R-CNN to classify them into a fixed number of classes. Only two classes are modeled in our research work those are text class and background, our intention is to retrieve only the text class objects.

At the final stage of the neural network architecture the R-CNN is used to classify each object and output scores for each possible object class. The goal of R-CNN is to classify proposals into one of the predefined classes plus background class and better adjust bounding boxes for proposals according to the predicted class.

During the evaluation of the network performance to detect text objects the Computer Vision Toolbox™ is used which provide best object detector evaluation functions to measure standard

metrics such as average precision that is known as *evaluateDetectionPrecision* in the MATLAB environment. This MATLAB built in function outputs a number that indicates the ability of the network to make correct classification, technically known as Precision and the ability of the network to find all relevant objects known as Recall.

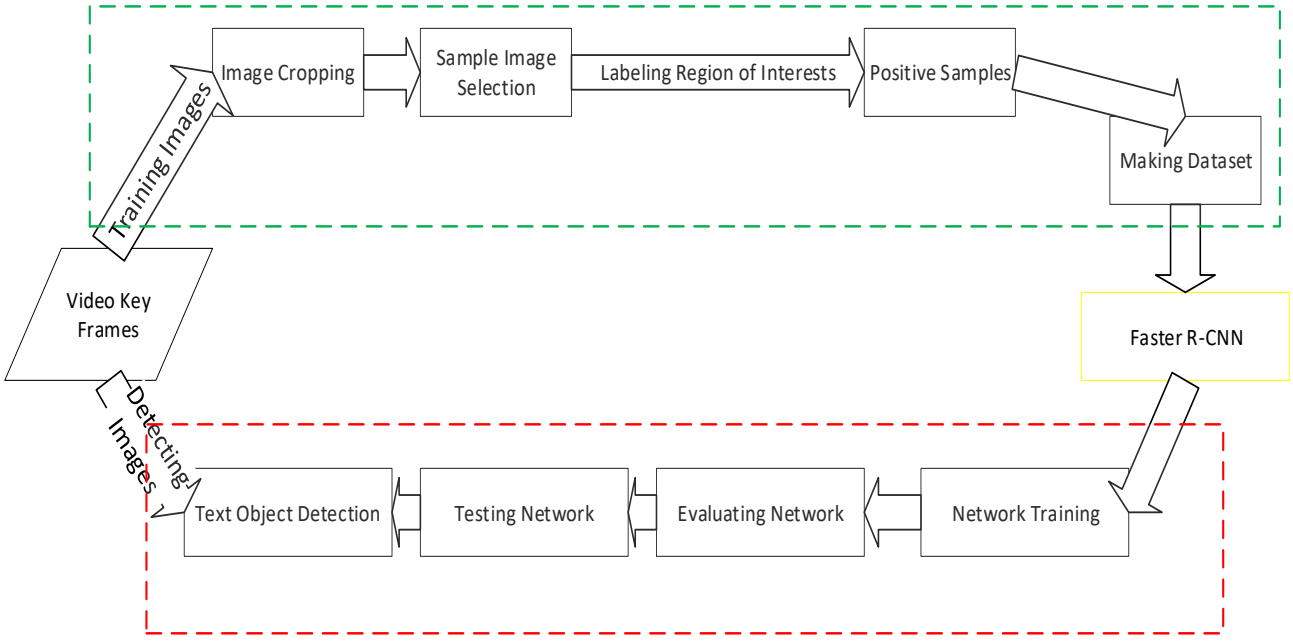


Figure 4. 3 Faster R-CNN Training and Text detection Flowchart

4.4.1. Network Architecture

The Faster R-CNN network defined as *vision.cnn.FastRCNN* object, consists of layers that define the convolutional neural network used in Fast R-CNN detector. It is a series Network composed of layers which arranged one after the other for deep learning. The series of Network object that was being used in our work is Alexnet which is a pretrained convolutional Network.

Alexnet has an input image size of 227 x227 x3, and the hidden layer is composed of 5 convolutional Layers following with corresponding Rectification Linear Unit, Cross channel normalization and maximum pooling layers. The Final Layers originally have a dropout layer which is used to tackle the problem of overfitting during training the network. But to see the comparative performance in the detection of text objects, we customize the training process to have a dropout layer for a single training phase and in the other phase we have ignored to insert

the dropout layer in the final layer. The remaining sublayers in the final layer of the network are three fully connected layers and one classification output layer.

The following steps are the precondition to start training the network to detect text objects from scene images

1. Provide the training data with an image datastore

An image datastore is used to manage a set of training images. A MATLAB built in function (*imageDatastore*) is used to create an image datastore object.

2. Specify the Network Architecture that is going to be trained to detect text objects.
3. Set the Training Options. These includes setting the solver name with its parameters. The solver name used in our work is Stochastic Gradient Descent with Momentum, which has Mini Batch Size, Max Epochs, Initial Learn Rate, Checkpoint Path parameters.

The parameters found with in Stochastic Gradient Descent with Momentum (sgdm) solver name used in our work are:

Mini Batch Size: the number of training images that are a divided into smaller batches. Since it is too large to feed into the network at one we need to divide the Training data into smaller batches. A mini batch is used to evaluate the gradient of the loss function. By using mini batch size we can update the weights at each iterations.

Max Epochs: one epoch can be considered as when an entire dataset is passed forward and backward through the Network once. To update the weight a greater number of times to converge the minimum error value More Number of Epochs are specified in the training options that is known as Max Epochs.

Initial Learn Rate: the rate at which the network is learned to do the task it is assigned. If the rate is low, it will take long time to train the network and if the rate is very high the training will reach a suboptimal result.

Checkpoint Path: it is the path where check point networks are saved for resuming training the network from where it was stopped before.

4. Start training the network using the MATLAB built in function(trainFasterRCNNObjectDetector) provided for Faster R-CNN with provided training data, network architecture and Training options.

4.4.2. Training Algorithm

To train deep learning Networks the algorithm should be defined in a well-organized manner so as to find the desired outputs correctly. The solver name used in the network is Stochastic gradient Descent with Momentum, minibatch size of 100, maximum epoch of 50 and Initial Learning rate of 0.00001. Also, the algorithm contains a check point path as temporary directory to resume training at later time.

4.4.3. Activation Function

The activation function used by Alexnet deep learning neural network is Relu(Rectified Linear Unit) activation. Mathematically this function is defined as;

$$y = \max(0, x) \tag{4.1}$$

4.4.4. Evaluating the trained Network

The trained Network has to be evaluated in order to know how much it is precise to detect text objects from Video frames. The standard evaluation metrics used in text detection works are Precision and Recall which are presented in [14] [44] [3] [29]. Therefore, we used Precision and Recall to evaluate our work. During evaluation we have provided the Ground Truth data used in training the Network and compared with the detection results by the network.

The MATLAB built in function (evaluateDetectionPrecision) has been used to evaluate the Precision and recall metrics of the trained network which takes two input arguments that are the Ground Truth data and the detection results.

Precision is the ratio true positive instances to the all positive instances of objects in the detector based on the Ground Truth.

$$Precision = \frac{TP}{TP+FP} \quad 4.2$$

Where,

TP stands for True positive and is defined as an outcome where the model correctly predicts the positive class (Text),

TN stands for True Negative and is defined as an outcome where the model correctly predicts the negative class (Background),

FP stands for False Positive and is defined as an outcome where the model incorrectly predicts the positive class (Text),

and FN stands for False Negative and is defined as an outcome where the model incorrectly predicts the negative class (Background).

Average Precision is a vector of average precision scores for each object class in multi class detector.

Recall is the ratio of true positive to the sum of true positive and false negative in the detector based on the Ground Truth.

$$Recall = \frac{TP}{TP+FN} \quad 4.3$$

Where TP refers true positive and FN refers to False Negative.

Testing the Trained Network

Test data is provided with its Ground Truth value with a structure array and the detection results were found by running the trained network over the Test data and saved as a structure array. The structure array of the detection results are converted to table data type and compared with the Ground truth value which is loaded from the disk.

4.5. Binarization

In our research, binarization involves the process of converting greyscale frames into black and white. The newly created image is known as binary image where black and white pixels are defined over the image representing the foreground and the background of the image respectively. In the implementation we have been using imbinarize MATLAB built in function to binarize greyscale images, where this function by default uses Otsu thresholding. Otsu Thresholding technically involves the iteration over all the possible threshold values and calculates a measure of spread for all the pixel level on each side of the threshold, i.e. pixels that lies on the foreground and the background of the image. The goal of otsu thresholding is to find the threshold value where the sum of all the foreground and the background spreads is at its minimum.

In Otsu thresholding, first the within-class variance is defined as the weighted sum of the variance of each cluster.

$$\sigma^2_{\text{Within}}(T) = n_B(T)\sigma^2_B(T) + n_O(T)\sigma^2_O(T) \tag{4.4}$$

where,

$$n_B(T) = \sum_{i=0}^{T-1} p(i) \tag{4.4.1}$$

where P(i) is the probability distribution each pixel represented as i.

$$n_O(T) = \sum_{i=T}^{N-1} p(i) \tag{4.4.2}$$

$\sigma^2_B(T)$ = the variance of the pixels in the background (below threshold) (4.4.3)

$\sigma^2_O(T)$ = the variance of the pixels in the foreground (above threshold) (4.4.4)

and $[0, N-1]$ is the range of intensity levels.

Instead of computing the within class variance for the two class we can use another way of computation to find the within class variance for the remaining cluster (either foreground or background). By subtracting the within class variance from the total variance we can get the between-class variance, which is defined as given in equation 4.4.4,

$$\sigma 2\text{Between}(T) = \sigma 2 - \sigma 2\ \text{Within}(T) = nB(T)\ [\mu B(T) - \mu]2 + nO(T)\ [\mu O(T) - \mu]2 \quad 4.4.4$$

Finally, for each Threshold T, the following procedures should takes place in Otsu Thresholding

1. Separate the pixels into two clusters according to the threshold.
2. Find the mean of each cluster.
3. Square the difference between the means.
4. Multiply by the number of pixels in one cluster times the number in the other.

Figure 4. 4 Result of Binarization using Otsu Thresholding

4.6. Skew Detection and Correction

During recording the Video texts can be skewed by some angle or the position of the digital camera may not be aligned with the text. The detected images which are found in the bounding boxes are therefore exposed to skewness and this will create a low performance on line segmentation and

character segmentation as well as Optical Character Recognition. In order to tackle this problem, we implemented skew detection and correction algorithm using Radon Transform.

4.7. Segmentation

4.7.1. Text line Segmentation

Segmenting the image into text line requires horizontal projection computed by a row wise black pixel. The gap found between two consecutive horizontal projections plus the histogram height is minimum denotes the boundary line. Finally using the boundary lines the image is segmented into different text lines.

Algorithm 2 Algorithm of Text line Segmentation *[45]*

input: image with more than one line of sentence

output: Each line segmented

plot the projection profile and scan the image horizontally where the value of pixel is zero and background 1, where there are letters.

Letterlocations= horizontal projection >0

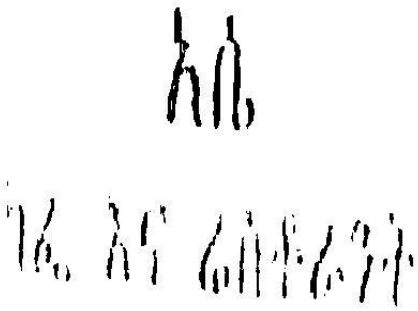
StartingRow= find(the value of a pixel>0)

EndigRow= find(the value of a pixel=0)

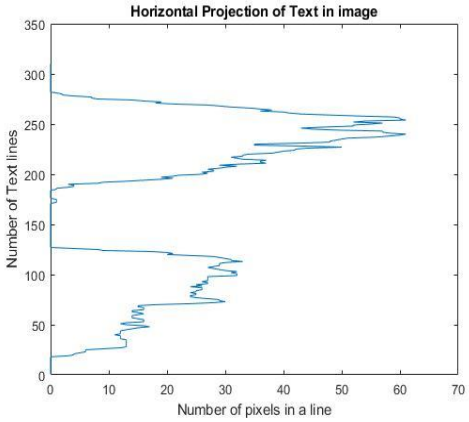
for K=***1: length(projectionprofile)*** **do**

| crop out each line using StrtingRow and EndingRow

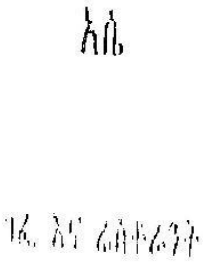
End



(a) Detected text image



(c) Horizontal Projection of text image



(b) Segmented Text Lines

Figure 4. 5 Text Line Segmentation

4.7.2. Text word Segmentation

One of the essential part of an optical character recognition system is text word segmentation. The algorithm that works behind this word segmentation is Vertical Projection of segmented text lines. After profiling the vertical projection of each of the text lines we have changed the greyscale image into binary image by thresholding the grey image and perform image dilation to connect all the letters found within the text line. Blobs which have less than 200 pixels are removed and find the area and bounding boxes of each word and finally each of the word images are cropped using the area and bounding box measurements.

Algorithm 3 Text word Segmentation *[45]*

input: image more than one word sentence

output: each word segmented

plot the projection profile and scan the image vertically and count the number of pixels in the grey level

Binary image= greyImage < 140

Perform Image Dilation

Remove Blobs < 200 Pixels

Measurements= regionprops('binaryImage', 'Area', 'BoundingBoxes')

for blob =1: measurement

Get the Bounding box

crop out each word using BoundingBoxes

end

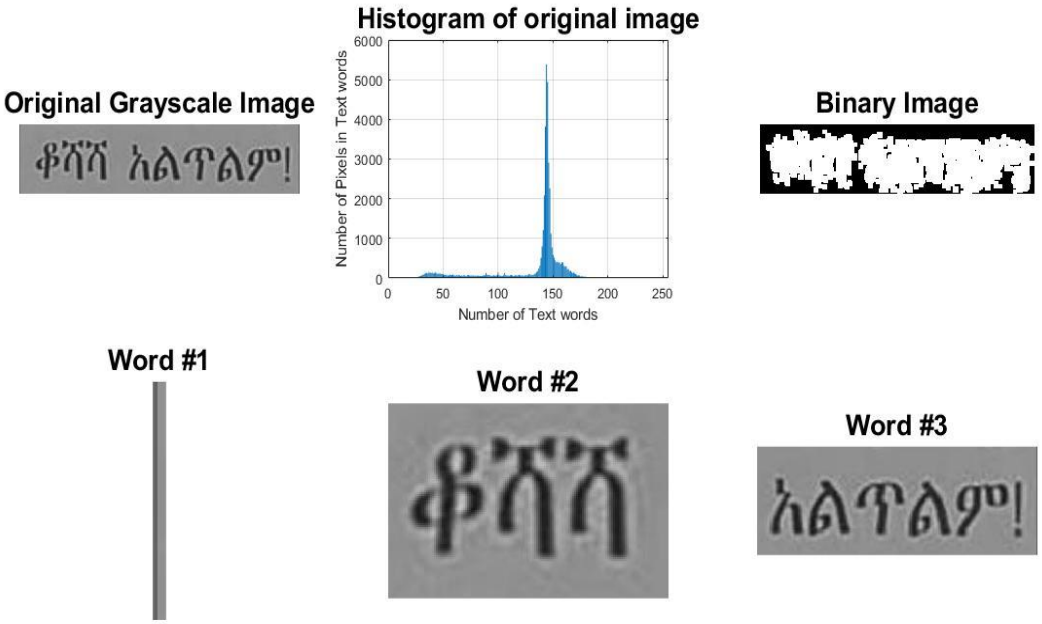
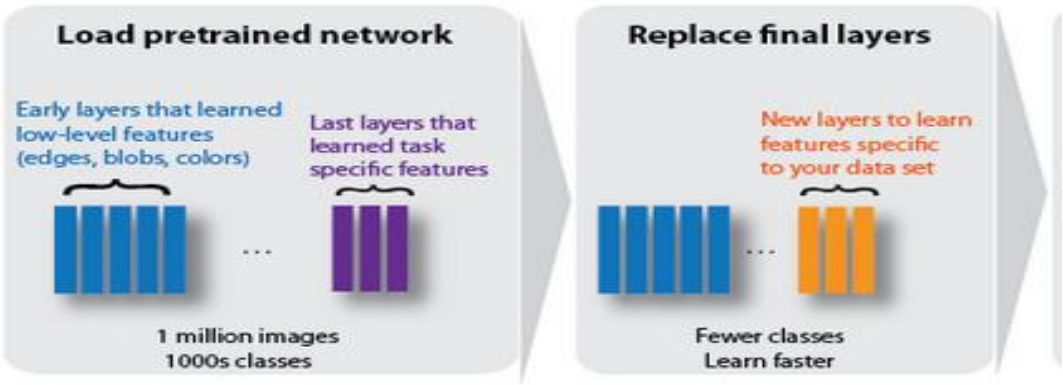


Figure 4. 6 Text word segmentation using vertical projection profile and Bounding Box measurement

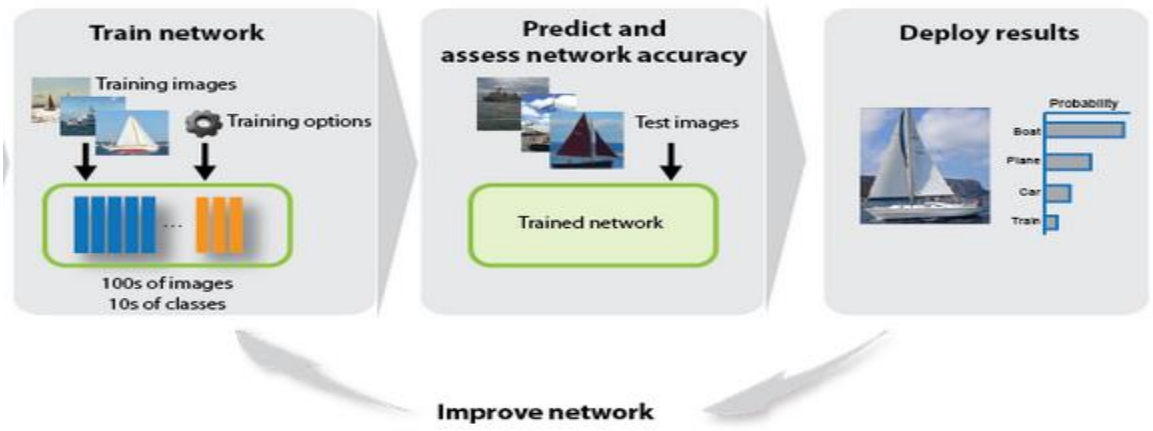
4.8. Classification

After Word segmentation is performed the next task of this research work is Script identification. Since the detected text blocks may contain several lines of text which can hold different types of scripts together, Scripts should be classified into their script class in order to recognize characters using OCR. The classification task can be performed into ways, as a first option it is possible to perform script identification simply on the detected word images, and as another option, as long as this research work is focuses on directly detecting text from Videos(image), preprocess, segment, identify and recognize characters, our script identification includes all the above tasks before identification.

In this research work we have used a pretrained Deep learning neural network architecture known as Alexnet which has trained over one million images and can classify into 1000 object categories. This Deeplearning neural network takes images as an input and outputs a label for the object in the image together with the probabilities each of the object categories. Technically training of a pretrained network such as Alexnet Deeplearning neural network architecture and use it as a starting point to learn a new task is known as Transfer learning. Fine tuning network with transfer learning is much faster and easier than training network with randomly initialized weights from scratch. In this way we can transfer learned features to a new task using a smaller number of training images. Below it is shown how Transfer learning can be achieved for a specific task using an existing pretrained network.



(a)



(b)

Figure 4. 7 Pretrained Network Architecture

In the research conducted by [46], the use of Convolutional Neural network has shown a great improvement on image classification task. The convolutional neural network model they have implemented in their work was Alexnet. This network is originally published by them by making the model to have a larger capacity in order for the machine to learn from Millions of images. This network was trained to classify 1.2 Million images and their result outfits the previous state of the art work in image classification.

In our research work the base network to train the classifier for image classification is Alexnet and the scheme we employed to train the network is Transfer Learning. But it should be noted that our dataset does not have millions of images rather we have thousands of images and we have introduced the number of images to be classified in the image input layer of the Network. And the output expected from the trained network is two class (Binary) outputs which are Ethiopic and Latin Scripts, not thousands of categories. What we have tried to achieve in our classification work is that from out of a combined set of scripts within a video frame we trained the network to bring only two classes of scripts as a standard set of script classes to be identified.

To Reuse Pretrained Network predefined set of steps should be followed that are shown on the figure:

i. Loading the Pretrained Network:

The pretrained network used by our research work is Alexnet which is a type of convolutional neural network. This Deep Learning Neural Network architecture has achieved high performance in classification tasks. In the figure 4.9(a) it is shown that the pretrained network consists of 1000 classes in the last three layers, however in our work we are interested to classify multiple number of word images into Ethiopic (Geez) and Latin. Therefore, we have reduced the last three layers of the network and configured with a new one which has only two classes to be classified, therefore the classification problem takes multiple word images and outputs them in to two script classes (Latin or Ethiopic). In the pretrained network some of the Major layers used are, Image input layer which is defined to be 227 x 227 x3 at the first layer of the network, Convolution2dLayer that is used to extract feature maps from the input image which are going to be given for the following layer in the network architecture, Max pooling Layer which is often used to pass the maximum value from each respective field.

Here is a table showing the layers used from the pretrained Alexnet network

Table 4. 1 Layers of Pretrained Network used by our research work

No	Name	Description of the Layers	
1	Data	Image Input	227x227x3 images with 'zerocenter' normalization
2	conv1	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [0 0 0 0]
3	relu1	ReLU	ReLU
4	norm1	Cross Channel Normalization	cross channel normalization with 5 channels per element
5	pool1	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
6	conv2	Convolution	256 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]
7	relu2	ReLU	ReLU
8	norm2	Cross Channel Normalization	cross channel normalization with 5 channels per element
9	Pool2	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
10	conv3	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
11	relu3	ReLU	ReLU
12	conv4	Convolution	384 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]

13	relu4	ReLU	ReLU
14	conv5	Convolution	256 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
15	relu5	ReLU	ReLU
16	pool5	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
17	fc6	Fully Connected	4096 fully connected layer
18	relu6	ReLU	ReLU
19	drop6	Dropout	50% dropout
20	fc7	Fully Connected	4096 fully connected layer
21	relu7	ReLU	ReLU
22	drop7	Dropout	50% dropout

ii. Replacing the Final Layers

In the pretrained network we left the final three sublayers of the architecture as a result of it was configured for 1000 classes. Originally Alexnet Network has 25 layers arranged seriously but 22 layers out of 25 are loaded as shown in table 4.1. The remaining three layers should be refined for our classification problem which has only two classes since our work is focused on two classes of scripts known as Latin and Ethiopic.

Softmax Layer: it is the commonly used classifier used in popular deep learning architectures at their final layer. Soft max classifier is used to give the probabilities of each class labels in the training data. This classifier is generally used for multiclass classification.

Softmax regression (or multinomial logistic regression) is a generalization of logistic regression to the case where we want to handle multiple classes. In logistic regression we assumed that the labels were binary: $y(i) \in \{0,1\}$. We used such a classifier to distinguish between two kinds of hand-written digits. Softmax regression allows us to handle $y(i) \in \{1,\ldots,K\}$ where K is the number of classes.

Recall that in logistic regression, we had a training set $\{(x(1),y(1)),\ldots,(x(m),y(m))\}$

of *m* labeled examples, where the input features are $x(i) \in \mathrm{R}n$. With logistic regression, we were in the binary classification setting, so the labels were $y(i) \in \{0,1\}$. Our hypothesis took the form:

$$h_\theta(x) = \frac{1}{1+\exp(-\theta^T x)} \tag{4.5}$$

and the model parameters θ were trained to minimize the cost function

$$J(\theta) = -\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(X^{(i)}) + (1 - y^{(i)} \log (1 - h_\theta(X^{(i)}))\right] \tag{4.6}$$

In the softmax regression setting, we are interested in multi-class classification (as opposed to only binary classification), and so the label *y* can take on *K* different values, rather than only two. Thus, in our training set $\{(x(1),y(1)),\ldots,(x(m),y(m))\}$, we now have that $y(i) \in \{1,2,\ldots,K\}$. (Note that our convention will be to index the classes starting from 1, rather than from 0.) For example, in the MNIST digit recognition task, we would have K=10 different classes.

Given a test input *x*, we want our hypothesis to estimate the probability that $P(y=k|x)$ for each value of k=1,…,*K*. I.e., we want to estimate the probability of the class label taking on each of the *K* different possible values. Thus, our hypothesis will output a *K*-dimensional vector (whose elements sum to 1) giving us our *K* estimated probabilities. Concretely, our hypothesis $h\theta(x)$ takes the form:

$$h_\theta(x) = \begin{bmatrix} P(y=1|x;0) \\ P(y=2)|x;\theta) \\ . \\ . \\ . \\ P(y=K|x;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} \exp(\theta^{(J)T} x)} \begin{bmatrix} \exp(\theta^{(1)T} x) \\ \exp(\theta^{(2)T} x) \\ . \\ . \\ . \\ \exp(\theta^{(K)T} x) \end{bmatrix}$$

Here $\theta(1),\theta(2),\ldots,\theta(K) \in \mathrm{R}n$ are the parameters of our model. Notice that the term $1\sum Kj=1\exp(\theta(j)\top x)$ normalizes the distribution, so that it sums to one.

Θ can be written as equation 4.8 to denote all the parameters of our model. When we implement softmax regression, it is usually better to represent θ as a n-by-K matrix obtained by concatenating $\theta(1),\theta(2),\ldots,\theta(K)$ into columns, so that

$$\theta = \begin{bmatrix} | & | & | & | \\ \theta^{(1)} & \theta^{(2)} & \cdots & \theta^{(K)} \\ | & | & | & | \end{bmatrix} \tag{4.8}$$

Cost Function

Now we describe the cost function that we are going to use for softmax regression. In the equation (4.9) below, 1{·} is the "indicator function" so that 1{a true statement}=1, and 1{a false statement}=0. For example, 1{2+2=4} evaluates to 1; whereas 1{1+1=5} evaluates to 0. Our cost function will be:

$$J(\theta) = -\left[\sum_{i=1}^{m}\sum_{k=1}^{K} \mathbf{1}\left\{y^{(i)} = k\right\} \log \frac{\exp(\theta^{(k)\top} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x^{(i)})}\right] \tag{4.9}$$

Note that this equation can be considered as generalization of the logistic regression cost function, and it could also have been written as [47]:

$$\begin{aligned} J(\theta) &= -\left[\sum_{i=1}^{m} (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) + y^{(i)} \log h_\theta(x^{(i)})\right] \\ &= -\left[\sum_{i=1}^{m}\sum_{k=0}^{1} \mathbf{1}\left\{y^{(i)} = k\right\} \log P(y^{(i)} = k|x^{(i)};\theta)\right] \end{aligned} \tag{4.10}$$

The softmax cost function is similar, considering an exception that summing over the *K* different possible values of the class label. It should be noted that in softmax regression, we have that

$$P(y^{(i)} = k|x^{(i)};\theta) = \frac{\exp(\theta^{(k)\top} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x^{(i)})} \tag{4.11}$$

Softmax regression is also known as multinomial regression, or multi-class logistic regression. Train the Network

We have considered to augment the image datastore to automatically resize the training images which can reduce overfitting and utilizing more memory space to have a more detail of images.

A MATLAB built in function trainNetwork is used during training the Transferred Network.

iii. Predicting and Evaluating the Accuracy of the Network

Predicted class labels with scores are returned using classify function by providing image data from augmented validation set. The function takes input arguments, the network that has been trained so far and the image data from the augmented validation set.

Accuracy can be defined as the ratio of the labels that the network predicts which is defined as YPred in the classification task. It is calculated as the mean value of the predicted labels and the labels extracted from the augmented validation set.

iv. Deploying the Result

Displaying the accuracy of the Prediction by the network.

4.8.1. Character Segmentation

After doing line and word segmentation the next step proceeding optical character recognition is Character segmentation. The algorithm works based on vertical projection of segmented word images which are processed during word segmentation.

Algorithm 4 Character Segmentation Algorithm [19]

Input: Segmented word images

Output: Single Characters

Read the segmented word images

Count the black pixel in each column

Find the vertical projection of the black pixel

Using the vertical projection profile find the column containing white pixel which separates the characters

Find location containing single white pixel
Mark the bounding box for each character using single white pixel
Copy the pixel in the bounding box and save in separate file

4.9. Optical Character Recognition

In this research work, we used LeNet-5 for training character recognizer. LeNet-5 is one of the convolutional neural network architectures best suited for character recognition, in our work we fed the training images into the network by splitting the dataset about 0.7 % and the network is trained to classify the images according to their class.

LeNet-5 architecture contains two sets of convolutional layers and average pooling layers with two fully connected layers.

First Layer: at the first layer of the network architecture LeNet-5 has a 32 x 32 input neurons to handle greyscale images.

Middle Layer: the middle layer consists of 2 Convolutional layers and Average pooling layers. The greyscale images which are fed into the input of the network passes through the first convolutional layer with 6 filters having 5 x5 with stride of 1 which produces an output size of 28 X 28.

The following Max pooling layer in the network architecture applies subsampling with a filter size of 2 x2 with a stride of two, therefore the resulting image dimension an be reduced t0 14 x14 x 6. The second convolutional layer with 16 filters of 5 x5 size with a stride of 1 and the final Average pooling layer has a filter size of 2 x 2 with a stride of 2 and will produce an output of 5 x5 x16.

Final Layer: the final layer consists of two fully connected Layers and softmax output layers that is defined based on the possible number of classes that the training data contains. Softmax is a transfer function used by neural networks. This transfer function calculates an output from its net input.

Model Parameters: Before starting the training process to recognize characters training parameters should be set up. The following are the parameters required for the proposed network with the dataset provided.

Number of Convolutional Layers in the Middle or hidden Layer: number of layers to extract feature maps from the input images.

Here is the summary of the LeNet-5 Deep learning neural network used in our work to recognize characters.

Table 4. 2 Summary of LeNet-5 Network used in character Recognition *[48]*

Layer		Feature Map	size	Kernel size	Stride	Activation
Input	image	1	32 x32	-	-	-
1	Convolution	6	28 x 28	5 x5	1	Tanh
2	Average Pooling	6	14 x 14	2x2	2	Tanh
3	Convolution	16	10 x 10	5x5	1	Tanh
4	Average Pooling	16	5 x 5	2x2	2	Tanh
5	Convolution	120	2 x 2	5x5	1	Tanh
6	FC	-	5 x 5	-	-	Tanh
Output	FC	-	-	-	-	Softmax

Learning Rate: training parameter that controls the size of weight and bias changes in learning of the training algorithm.

Batch Size: total number of training examples found within a single batch. Typical Value depends on the size of the training data.

Training Algorithm of LeNet-5:

- Learning rate was set to 0.0001.
- Batch size was set to 500.
- Max Epoch was set to 10.
- The convolution layer of the network used the tanh activation function.

- The output layer of the network uses the softmax activation to classify the characters into their respective character class.

Activation Function:

Tanh Function: the tanh function is the updated version of the sigmoid activation function. In the sigmoid activation the values are snapped between 0 and 1, however in the case of tanh the values are snapped between -1 and 1 which is graphically shown in Figure 4.8. Although the model that uses this activation has achieved high performance in character recognition, still the activation function has a vanishing gradient problem that is found in sigmoid activation function. Since there is hard or soft rule for selecting which activation function to use during training our network, we depend on the activation function that the network model used to be trained.

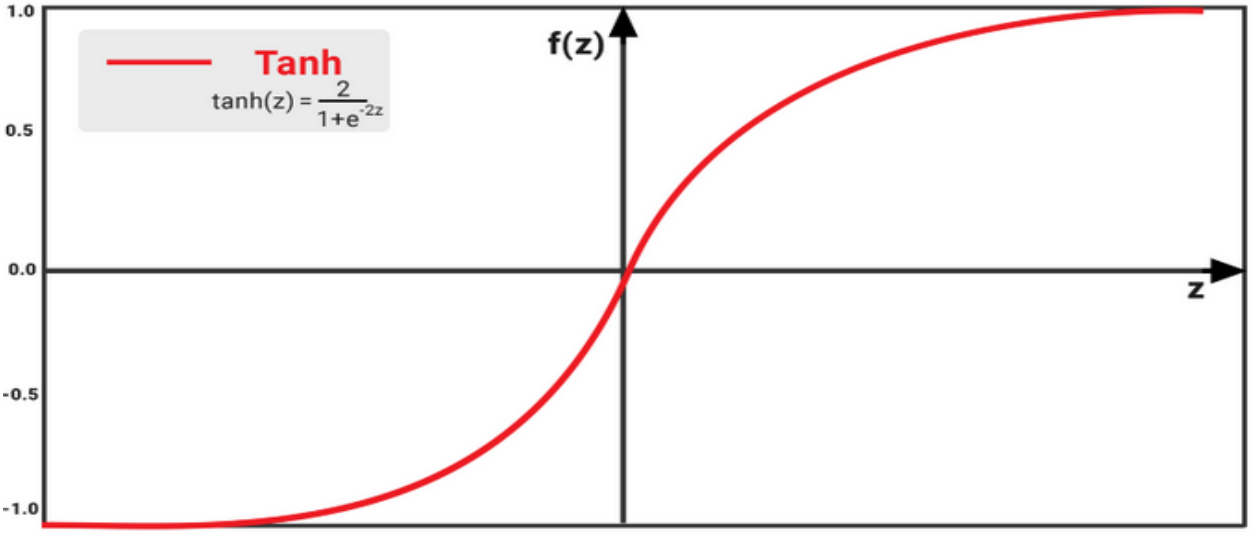


Figure 4. 8 Performance of Tanh activation function

Chapter Five

Experimental Results and Discussions

In this section we describe our evaluation of text detection, script identification and character recognition of our system. Our proposed method is implemented using MATLAB 2018a on a system with Intel Core TM i5 CPU @ 2.3GHz , 8GB RAM and NVIDIA GPU of 2GB.

5.1. Dataset and Evaluation

5.1.1. Dataset

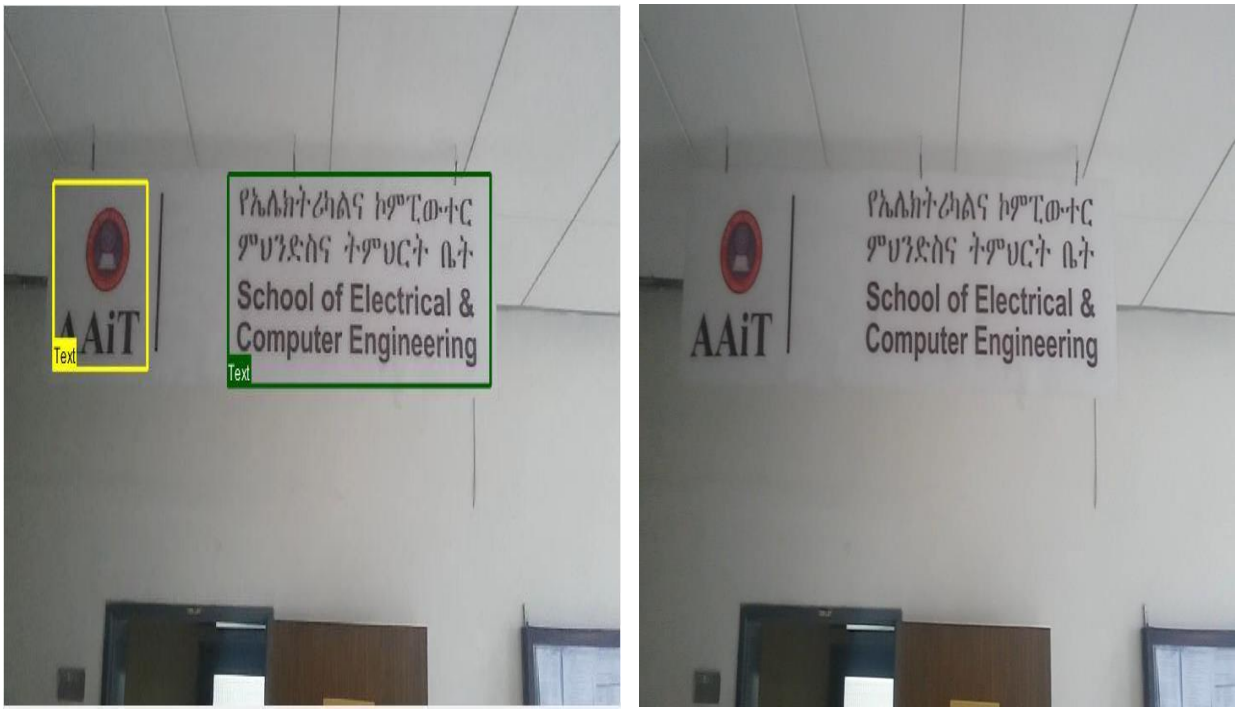
It is known that text detection and script identification work has been conducted by [14], but we couldn't find available Ethiopic scene text image database for conducting our experiment. Due to this we have prepared our dataset which is scene text dataset. The dataset contains 1250 video frames which are taken from the streets of Addis Ababa using digital camera. For each text object the ground truth for text detection is manually labeled and prepared with the use of Training image labeler. The text found in the images are an indication of commonly used text in the daily life of the people living in Addis Ababa which are mainly shops, broker name and address, street guides, café and restaurants, organizations, schools and departments with complex background and orientation. The ICDAR images are taken from different sources that include book covers, street signs, cd covers.

In order to see whether our text detection compromise the quality in its evaluation it should be trained with a dataset that contain a diversity of texts with multilingual schemes, complex background. So, our dataset is prepared to be a standard one.

The second dataset used is the ICDAR2003 dataset which is updating through time from a series of Robust Reading competition held by ICDAR. The dataset consists of 462 images including 229 training images and 233 test images. For each word with in images the ground truth is fully annotated.

5.1.2. Evaluation

In our work we set two types of evaluation, subjective and objective. In the subjective evaluation, the evaluation metrics use the regions which contain text with bounding box as shown in Fig . In the objective evaluation we have prepared the dataset with ground truth as shown in Fig .



(a) Sample ground truth for objective evaluation (b) Sample dataset for Subjective evaluation

Figure 5. 1 A Multilingual Scene Text Example

Our approach for text detection in both objective and subjective evaluation is evaluated by considering its precision, recall and average precision.

5.1.2.1 Objective Evaluation

In this evaluation, we have the ground truth data for each training image where the MATLAB Training image labeler that label the Region of Interests (ROI) was used. And the detection results from the trained network are taken by considering their coordinates or bounding box values. Therefore, we have two sets that are the Ground truth values and the detection results formed by

the network. These values are matched and the performance metrics are calculated. The geometry for these values is x, y, width and height. The experimental evaluation results for text detection are discussed on the next topic of this chapter.

5.2. Text Detection

The Deeplearning network that has been trained to detect scene or Natural texts is Faster R-CNN (Region based Convolutional Neural Network). The network architecture consists of 227 x 227 pixel input layer which takes this size of input images into the network, the middle layer consists of 5 convolutional layers which computes feature map and feed it to the next layer in the network architecture with a specified window of 96x 1, 256x 96, 384 x 256, 384 x 384 and 256 x 384 respectively. Three fully connected layers are constituted in the final layer of the network architecture where the output of the network can be found from this layer. During the training about 1250 training data with 434 training images for Latin script labeled using Training image labeler and 532 training images that contain Ethiopic scripts. But those scripts are appearing together on some of the training images.

The First experiment is conducted on our own dataset. All of the training data are labeled with text within the ROI Label. During testing the Network that consists dropout layer in its architecture we have found a precision of 91% and 92.9% recall metrics as it is shown in table 5.1 which shows a significant value in detecting text objects from scene images. The text regions are effectively detected by the trained network as shown in Figure 5.2.

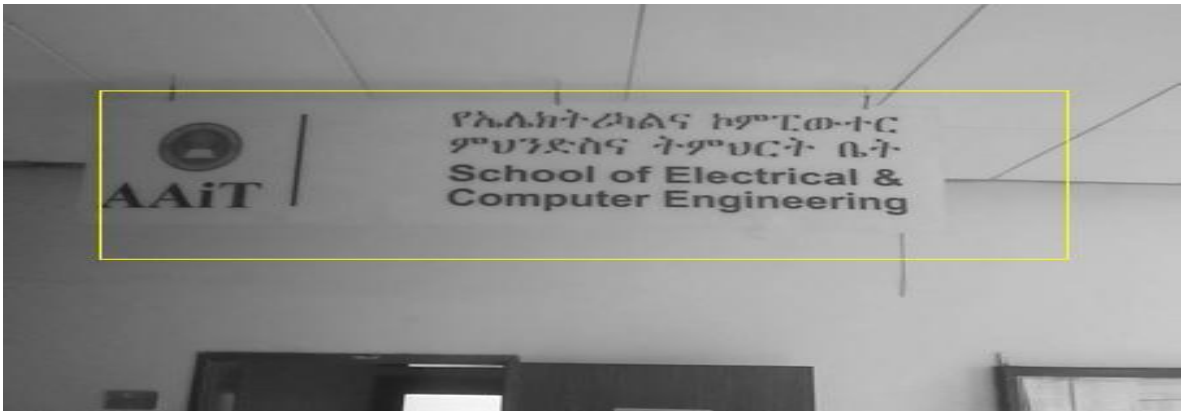


Figure 5. 2 Text Detection using Faster R-CNN

Table 5. 1 Faster R-CNN Detection result with Dropout Layer

Precision	Recall	Time(Sec.)
91%	92.9%	7.5

Another experiment for text detection was carried out using the network that do not consist dropout layer in its architecture, in this experiment it is found that a precision of 79.2% and 76.5% recall metrics. Ignoring dropout layer in the network architecture results in a performance degradation in the text detection task as it is shown in table 5.1.

Table 5. 2 Faster R-CNN detection result without Dropout Layer

Precision	Recall	Time(Sec.)
79.2%	76.5%	6.4

Figure 5. 3 Text Detection using Faster R-CNN without the use of Dropout Layer

Ignoring dropout layer during defining the network to be trained results in degradation of text detection performance as shown in Figure 5.3.

The second Experiment is conducted on ICDAR2003 robust reading competition dataset. Due to there is available Ethiopic or geez dataset for Text detection, we compare our work with the work conducted by [14]. The comparison is done with the results found from our system with preprocessing steps and the use of dropout layer.

Table 5. 3 Text Detection Result in ICDAR dataset

Criteria for Assessment				
Method		Precision	Recall	Time (Sec.)
	Ours	92.33%	91.9%	7.5
	Method [14]	78%	70%	13

The state-of-the-art method exploited in [14] uses MSER Algorithm to detect text. This method considers regions which stays nearly the same through a wide range of thresholds. The area is first scanned and those regions whose variations with respect to the threshold is minimal are defined as Maximally Stable.

5.3. Script Identification

As a second phase of our work script identification has been conducted so far, in this section; two approaches are taken which are End to End Script Identification and Cropped Script Identification. For each script type a feature vector is generated by the neural network based on that feature vector the script is classified into its particular label(class).

5.4. End to End Script Identification

This stage combines all the preprocessing steps that are needed for identifying multilingual scripts. It takes the input video and extract key frame, finds text regions using trained Faster R-CNN network, in this stage it crops out the bounding boxes that contain the text object and each of the text objects are preprocessed and words are segmented in order to feed them into trained network using Transfer Learning that identify each word into their appropriate script class. In the text detection part, the system fails to detect some of the text regions which are found in the video frames with the use of bounding box. However, we have given of the text image from the text detection with bounding boxes and we have evaluated the performance of script identification.

The dataset was prepared for this experimentation comes from the result of text detection on multilingual and monolingual images, and segmentation stages that include line segmentation and word segmentation. We have found 1000 word images and we ignore words that have more than

one script type because during learning the features from the word images needs each of the script classes alone. We have also used the dataset from ICDAR2013 for Latin where majority of the images are with low resolutions, noise and blur. The evaluation metrics used in script identification is accuracy. Which is defined as follows;

$$Accuracy = \frac{Correctly\ identified\ word\ images\ with\ script\ labels}{Total\ number\ of\ word\ images\ with\ script\ labels} * 100 \quad (5.1)$$

Table 5. 4 Accuracy of Script Identification using Transfer Learning

Net	Overall Accuracy
Transfer Learning with Alexnet base network	88.5

5.5. Cropped Script Identification

Practically it is difficult to get an exact bounding box that contain the text from text detectors, to see the performance of Script Identification without preprocessing steps that may degrade the result of identification we have manually cropped word images from of Latin and Ethiopic Training sets. The Validation accuracy using validation data which are found by randomly shuffling the Training data is 100% as shown on the figure 5.4 below.

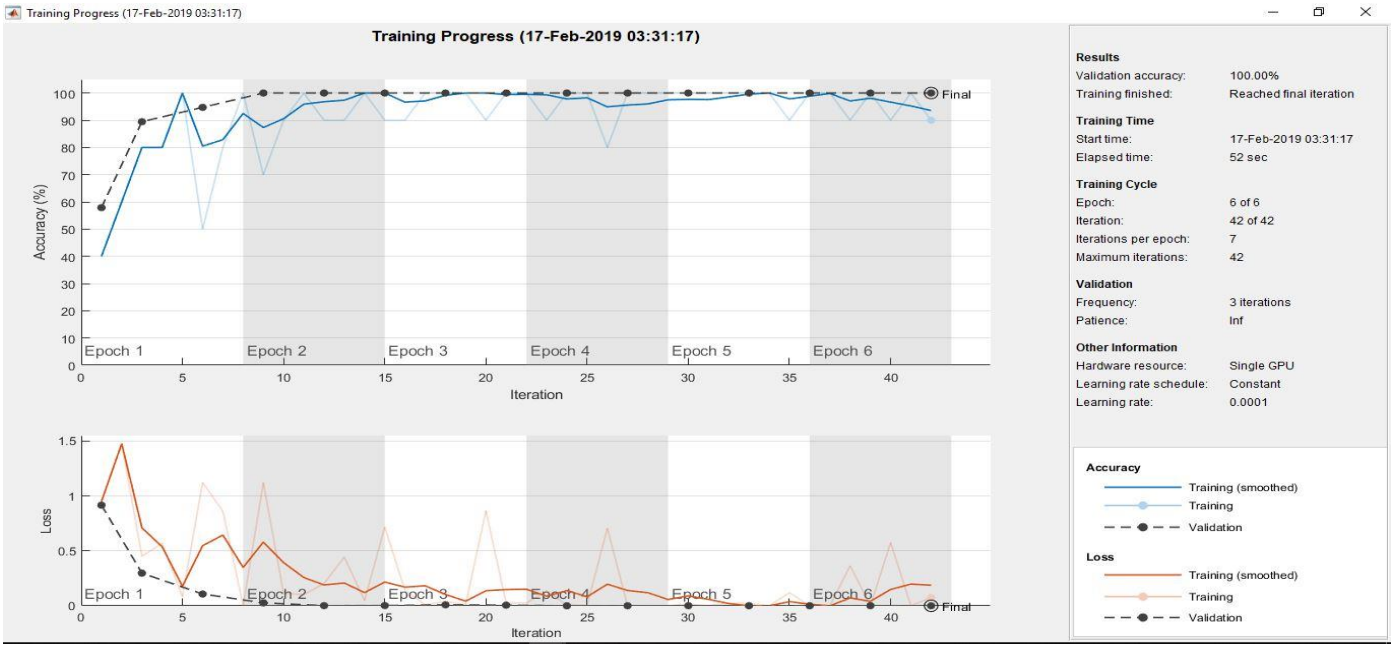


Figure 5. 4 Training and Evaluation of the Transferred Network

Below it is presented the Graphical output of our script identification which is given a validation set of different type of scripts and classify these scripts in two labels(classes) known as Ethiopic and Latin.

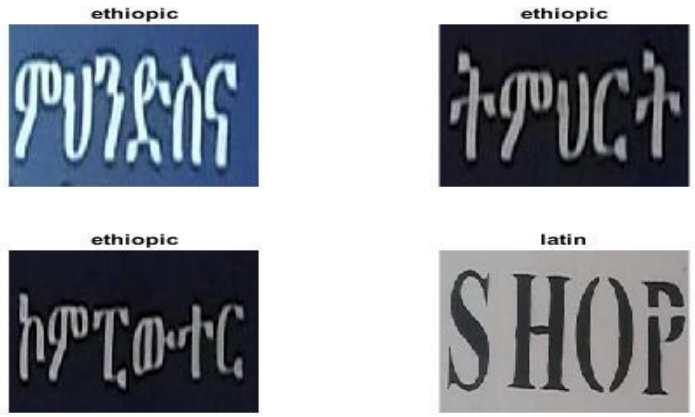


Figure 5. 5 Classification output of the Trained Network

Table 5. 5 Script Identification with Transfer Learning

	Training with Dropout Layer	Training without Dropout Layer
Accuracy	93.33%	81.5%

In the second experiment, ICDAR2003 Robust reading competition dataset has been used to test the network accuracy in identifying scripts with state of the art method [14] which includes all the preprocessing steps carried out previously. Using ICDAR2003 dataset as common data to compare our work with the work conducted in [14] the following table script identification metrics are found. Our script identification phase assumes training with dropout layer.

Table 5. 6 Script identification result in ICDAR 2003 dataset

Methods	Accuracy
Our's	93.33%
SVM [14]	79.9%

5.6. Character Recognition

Our proposed Deep Learning Neural Network was trained and tested under the following conditions:

- The Training data is provided as an input image to the network in 32 x 32-pixel zip file. Some of the randomly selected sample images are shown in fig 5.4.
- Each input image character is 32 x 32 pixels, accordingly the number of input neurons will be 1024.
- Learning rate was set to 0.0001.
- Batch size was set to 500.
- Max Epoch was set to 10.
- Network Model used was LeNet-5.

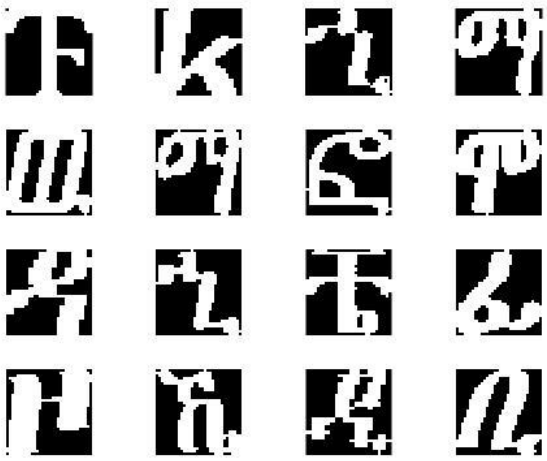


Figure 5. 6 Sample Training Data

Our Focus during training the network for recognition lies on Ethiopic or Geez Characters. Due to so many researches have been conducted for character recognition including English and Geez characters this stage doesn't concern that much about the optical character recognition. However, we have implemented LeNet-5 which is powerful deep learning neural network in the task of character recognition. In the implementation we vary the number of epochs to see their effect in recognizing characters to get a better result. The following table shows the recognition error found in each epoch.

Table 5. 7 Trained LeNet-5 Network with different epochs

Epoch	Classification Error
50	0.061
100	0.045
150	0.012
200	0.0076

As we are looking from table 5.5 it is obvious that a greater number of epochs will reduce the classification error. From the table better result was obtained by using 200 Epochs.

5.7. Challenges

Character Recognition from Video Scene is challenging task due to the following main reasons:

Text in Scene Videos are frequently embedded on complex background, having Different font styles, similarity of Text object and different shapes such as bricks and Fences. The other challenge we have encountered in our work is missing exact text objects which are found on the video frames. Different text object proposals were made by Faster R-CNN network to detect the text class objects. Targets in Faster R-CNN networks are calculated as the offset between the proposal and its corresponding ground truth boxes, in fact some of the detection result have IoU below 0.5. In the script identification task we have found an excellent identification metrics therefore we don't have any worry about this stage. The use of activation function in the hidden network that is tanh activation has a drawback in learning the features from the data as a result of it has a vanishing gradient problem by its nature.

(a)

(b)

Figure 5. 7 Missing Text object during testing the Network

5.8. Answers to the Research Questions

Question of interests has already noted on the Research Question of this thesis work. Our work tries to reflect the impact of deep neural networks for Text Detection and recognition from Video Scene. In this part we have briefly recall the Question of interests that were discussed on Chapter one.

RQ1. The primary and the main question of this thesis work is, Does the usage of Convolutional Neural Networks with preprocessing improve the performance of text detection from video?

The RQ1 implies detecting text blocks from video frames. Major challenges such as Background color and Noise make text detection performance poor, to tackle these problems we have implemented a preprocessing technique which is Grey Scale conversion to detect text block easily with a better performance and Faster R-CNN Network which was earlier used for object detection has been implemented in our work to detect text block as objects. Therefore, using both Grey Scale Conversion together with Faster R-CNN makes the text detection system more Efficient.

RQ2. Can we achieve a comparative performance improvement of Script identification from Multilingual scripts by implementing Transfer learning?

Here we are interested in classifying or identifying Ethiopic and Latin Scripts from the detected text blocks with a better accuracy. We have started our classification task by taking the text blocks from the previous stage that is text detection. Then we have implemented segmentation steps starting from Line Segmentation to Word Segmentation. The task of identifying scripts in our work is based on word features which are found from word segmentation step. A Transferred Network known as Alex net was trained to identify scripts and classify them into their script class. The use of Transferred Network achieves maximum classification accuracy.

5.9. Discussions

1. The results that were found from our experiment has shown the robustness of our proposed method. Noisy video frames are difficult to process and detect text with a good accuracy.

Besides Scene texts in their nature are challenging to be exactly detected by different text detection methods, however the use of Faster R-CNN in our work achieves a better text detection performance. Since the training of the Faster R-CNN is focused on finding region of interests that were labeled during data preparation stages the convolutional layers can easily learns to extract important features which can be considered as text regions. The reason that this work has achieved a greater performance is due to it finds text region proposal rather than pixel wise computation to exactly find text features from the whole image. Further the problem of overfitting can be tackled by the use of Drop out Layer which drops some of the outputs from the previous layer with assigned probability. The probability we have assigned in the dropout layer during training the network for text detection, script identification and character recognition is 0.5. Training Deep Learning neural networks takes longer hours to days and needs an expensive hardware like high performance computer with graphical processing unit for parallel computation. During testing the trained network, we can gain an advantage by getting small amount of time to do the desired task and we can achieve a better performance if the network is trained well with huge data. Due to these reasons we have got a comparative performance improvement than the previous state of the art method [14] in Text detection performance.

2. From the comparison of Our work with state-of-the-art method [14], it is obvious that the performance of our proposed method is better in detection performance and execution time. In our implementation training the network has taken longer time in which the network has learned abstract features from the training data which are prepared using training image labeler with a greater number of Epochs, this could result in a better text detection quality during testing the network. Besides the network takes preprocessed images where noise and other unwanted signals are removed which makes it suitable detecting the text regions easily.
3. Skewness of an image will result in poor character recognition performance. In our work we have implemented Radon Transform to detect skew angle in the detected text bounding boxes. The Radon transform is the projection of the image intensity along a radial line oriented at a specific angle. During Line segmentation we have found more lines are segmented correctly.
4. Image Binarization has played a great role for script identification which uses Otsu thresholding. It classifies the original greyscale image into black and white by choosing a threshold that reduces the intraclass variance of the thresholded black and white pixels.

5. The exploitation of Transfer learning technique in script identification has achieved a greater performance when compared to state-of-the-art method [14]. The base network that is used in transfer learning is a deep learning network which is Alexnet. This is trained over a million of training images and capable of classifying images with high accuracy. Therefore, using this base network architecture and modified weights can be advantageous rather than developing a deep learning network and modifying it frequently to gain a better classification accuracy. State of the Art method [14], script identification has been performed using LBP which compute feature vector from each segmented word. After the computation of the feature vector by the LBP, SVM was exploited to classify words into their script class.
6. Our system is totally composed of Text detection, segmentation, script identification and character recognition. Each of the steps has already achieved quality detection, identification and character recognition performance with a good execution time. We have employed Faster R-CNN deep learning neural network that achieves a better precision and recall metrics than state of the art method [14], in the script identification task another deep learning architecture known as Alexnet was employed for Transfer learning which achieves an excellent classification accuracy of scripts and in the final phase another Deeplearning neural network known as LeNet-5 and has achieved very high recognition performance with an increasing number of Epochs.
7. By using an activation function that is tanh which is shown in figure 4.10, we can not gain maximum performance by the trained network. Although the Network Model that recognize characters uses tanh activation achieves more error rates with minimum number of epochs during testing, we minimized the error value to minimum value by increasing the number of epochs during training the network.

Chapter Six

Conclusion and Recommendation

In this chapter we have concluded our thesis work by discussing the proposed methodology and comparison with baseline works conducted before.

6.1. Conclusion

Generally, the objective of this thesis work is detecting text from Videos, identifying scripts and character recognition. We have found that text detection from scene video is challenging task. The challenge is due to the complexity of Background, orientation of text, font style and varying font size with in the same text. Dynamicity of text location is one of the behaviors often found within Scene text.

The video that contain text object is divided into frames in order to detect text. Histogram difference between consecutive frames are calculated to select key frame from a collection of extracted frames from the video. In order to reduce the noise that exist in video frames, preprocessing techniques were applied on the frames. Faster R-CNN network was used to detect text objects from the preprocessed frames and we have found a better evaluation metrics. Even though Faster R-CNN detects most text objects, also detects regions that do not contain objects.

For developing OCR systems that handle multiclass scripts, script identification is very important. The primary task of this thesis work is to detect the text object found with in video frames, after detecting text objects the results are then goes to segmentation process which segments text lines and words respectively. Transfer Learning technique was used to identify segmented words into two script classes, Ethiopic and Latin.

The last phase of our work is recognizing characters from Ethiopic Script which is found after script identification. Our intention is to make character recognition only from Ethiopic scripts due to recognizing Ethiopic characters by itself is very large and time-consuming task. We have found very less recognition accuracy as we go from minimum to maximum number of epochs. Therefore, the use of Deeplearning neural networks in our system can be seen as a golden technique to make character recognition from videos in Realtime. The character recognition task is limited up to

classifying characters into their character class. But the recognition task does not include post processing which uses character database and encode the classified characters into machine encoded characters.

6.2. Recommendation

This thesis opens opportunities for further research.

- The technique used in extracting key frames is calculation of histogram difference between consecutive frames. Others techniques can be employed to extract key frames from video to enhance the performance of Processing data.
- In text detection, script identification and character recognition Deep learning neural network architectures that are Faster R-CNN, Transfer learning that uses Alex net as the base network and LeNet-5 networks are employed for text detection, script identification and character recognition respectively. An Other Regression based method like Rotation Region Proposal Network which has been employed in [10] can be employed to detect text Scene Text from Video which enhances the performance of text detection as Scene texts can be found in arbitrary position and orientation by their nature.
- Our character recognition task lies only on Ethiopic or Geez characters due to we consider more character recognition works are done for Latin scripts so far. The use of Deeplearning neural network LeNet-5 has shown a very good recognition performance. But other Networks such as Multilayer Perceptron can be incorporated in the Realtime character recognition task which starts from detection and goes to script identification and finally to recognition. Further works should be done to include post processing after classifying characters into their character class and encode into machine encoded character.

References

[1] A. Alemu, "Character Recognition of Bilingual Amharic-Latin Printed Documents," Addis Ababa, Ethiopia, 2018.

[2] K. J. a. J. H. K. Kwang In Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, TAMPI, 2003.

[3] S. S. T. G. S. D. M. C. R. G. a. B. G. H. Chen, "Robust text detection in natural images with edge-enhanced Maximally Stable Extremal Regions," in *International Conference on Image Processing*, Brussels, 2011.

[4] J. W. a. Y.-T. S. Zhong Ji, "Text detection in video frames using hybrid features," in *International Conference on Machine Learning and Cybernetics*, Baoding, 2009.

[5] Y. W. a. X. F. T. Yusufu, "A Video Text Detection and Tracking System," in *IEEE International Symposium on Multimedia*, Anaheim, CA, 2013.

[6] E. O. a. Y. W. B. Epshtein, "Detecting text in natural scenes with stroke width transform," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 2010.

[7] T. M. Y. S. a. S. O. Y. Nagaoka, "Text Detection by Faster R-CNN with Multiple Region Proposal Networks," in *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, 2017.

[8] H. K. G. Ren S, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015.

[9] H. S. J. C. X. H. a. J. Y. W. Lu, "A Novel Approach for Video Text Detection and Recognition Based on a Corner Response Feature Map and Transferred Deep Convolutional Neural Network," *IEEE Access*, vol. 6, pp. 40198-40211, 2018.

[10] W. S. H. Y. L. W. H. W. Y. Z. a. X. X. Jianqi Ma, "Arbitrary-Oriented Scene Text Detection via Rotation Proposals," *CoRR*, vol. abs/1703.01086, 2017.

[11] C. Z. W. S. C. Y. W. L. a. X. B. Z. Zhang, "Multi-oriented Text Detection with Fully Convolutional Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 2016.

[12] W. H. T. H. P. H. a. Y. Q. Z. Tian, "Detecting Text in Natural Image with Connectionist Text Proposal Network," in *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, 2016, p. 56–72.

[13] X. B. N. S. X. Z. S. Z. a. C. Cong Yao, "Scene Text Detection via Holistic, Multi-Channel Prediction," *CoRR*, vol. abs/1606.09002, 2016.

[14] A. Awoke, "Ethiopic and Latin Multilingual Text Detection and Script Identification," Addis Ababa, 2018.

[15] A. M. a. U. P. M. A. Ferrer, "LBP Based Line-Wise Script Identification," in *International Conference on Document Analysis and Recognition*, Washington, DC, 2013.

[16] L. D. B. S. X. B. Jieru Mei, "Scene Text Script Identification with Convolutional Recurrent Neural Networks," in *23rd International Conference on Pattern Recognition (ICPR)*, Cancún, México, 2016.

[17] S. C. U. P. a. M. B. N. Sharma, "Word-Wise Script Identification from Video Frames," in *International Conference on Document Analysis and Recognition*, Washington, DC, 2013.

[18] J. Z. a. H. Nakayama, "Bag of Local Convolutional Triplets for Script Identification in Scene Text," in *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, 2017.

[19] S. Getu, "Ancient Ethiopic Manuscript Recognition Using Deeplearning Artificial Network," Addis Ababa, 2016.

[20] B. A. a. S. Roohi, "Persian handwritten character recognition using convolutional neural network," in *10th Iranian Conference on Machine Vision and Image Processing (MVIP)*, Isfahan, 2017.

[21] Z. S. a. A. Testolin, "Learning representation hierarchies by sharing visual features: a computational investigation of Persian character recognition with unsupervised deep learning," *Cognitive Processing*, vol. 18, no. 3, pp. 273-284, 2017.

[22] K. P. a. M. Soryani, "From machine generated to handwritten character recognition; a deep learning approach," *3rd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, 2017.

[23] a. A. a. Mahdi Toopchi, "Persian Alphabet Recognition Usnig Multilayer Perceptron," in *Conference on New Research Findings of Science, Engineering and Technology*, Turkey-Istanbul, 2016.

[24] F. S. a. P. N. A. Dehghani, "Off-line recognition of isolated Persian handwritten characters using multiple hidden Markov models," in *Proceedings International Conference on Information Technology: Coding and Computing*, Las Vegas, NV, USA, 2001.

[25] D. Bouchain, "Character Recognition Using Convolutional Neural Network," University of Ulm, Germany, 2006.

[26] B. A. Samad Roohi, "Persian Handwritten Character Recognition Using Convolutional Neural Network," in *10th Iranian Conference on Machine Vision and Image Processing*, Isfahan, Iran, 2017.

[27] Q. H. W. G. D. Z. Qixiang Ye, "Fast and robust text detection in images and video frames," *Image and Vision Computing* , vol. 23, no. 6, pp. 565-576, 2005.

[28] T. P. S. T. C. L. W. Lu, "Video Text Detection," Springer, 2014.

[29] S. A. M. B. S. M. V. A. C. A. Thilagavathy, "Tamil Text detection in videos," in *International Journal of Engineering and Innovative Technology (IJEIT)*, Tamil Nadu, 2014.

[30] D. &. L. J. Chen, A Survey of Text Detection and Recognition in Images and Videos., ResearchGate, 2000.

[31] J. S. a. M. C. M. R. Lyu, "A comprehensive method for multilingual video text detection, localization, and extraction," in *IEEE Transactions on Circuits and Systems for Video Technology*, 2005.

[32] W. P. J. Z. a. H. H. X. Yin, "Multi-Orientation Scene Text Detection with Adaptive Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1930 - 1937, 2015.

[33] N. K. N. Sheena C Va, "Key-frame extraction by analysis of histograms of video frames," in *Procedia of Computer Science*, 2015.

[34] M. S. D. M. Madhup Shrivastava, "ARTIFICIAL NEURAL NETWORK BASED CHARACTER RECOGNITION USING BACKPROPAGAT," *International Journal of Computers & Technology*, vol. 3, no. 1, 2012.

[35] J. X. L. C. H. J.-S. P. Boya Wang, "Scene Text Recognition Algorithm Based on Faster RCNN," in *Electronics Instrumentaion & Information Systems (EIIS)*, Harbin, China, 2017.

[36] M. N. X. C. Ishan Jindal, "Learning Deep Networks from Noisy Labels with Dropout Regularization," *16th International Conference on Data Mining*, pp. 967-972, 2016.

[37] P. K. B. I. Gaurav Kumar, "Analytical Review of Preprocessing Techniques for Offline Handwritten Caracter Recognition," *Proceedings of 2nd International Conference on Emerging Trends in Engineering and Management, ICETEM*, 2013.

[38] R. G. U. D. A. S. P. C. Prakash K Aithal, "A Fast and Novel Skew Estimation Approach using Radon Transform," *International Journal of Computer Information Systems and Industrial Management Applications.*, vol. 5, pp. 337-344, 2013.

[39] S. Abebe, "BILINGUAL SCRIPT IDENTIFICATION FOR OPTICAL CHARACTER RECOGNITION OF AMHARIC & ENGLISH PRINTED DOCUMENT," Addis Ababa, 2011.

[40] C.-W. C.-C. C. a. C.-J. L. Hsu, A practical guide to support vector classification, (2003), pp. 1-16.

[41] K. M. P. B. S. K. G. Arindam Chaudhuri, Optical Character Recognition Systems for Different Languages with Soft Computing, Springer, 2017.

[42] M. M. a. J.-H. C. P. D. Gader, "Handwritten word recognition with character and inter-character neural networks," *IEEE*, vol. 27, pp. 158-164, 1991.

[43] G. I. a. D. A. N. Rathod, "An algorithm for shot boundary detection," in *International Journal of Emerging Technology and Advanced Engineering*, 2013.

[44] J. W. Y.-T. S. ZHONG JI, "TEXT DETECTION IN VIDEO FRAMES USING HYBRID FEATURES," in *International Conference on Machine Learning and Cybernetics*, Baoding, 2013.

[45] H. C. S. K. N. a. G. L. A. Vinod, "Detection, Extraction and Segmentation of Video Text in Complex Background.," *International Journal on Advanced Computer Theory and Engineering*, pp. 117-123, 2013.

[46] A. &. S. I. &. H. G. Krizhevsky, "ImageNet Classification with Deep Convolutional Neural Networks," *Neural Information Processing Systems*, vol. 60, no. 6, pp. 84-90, 2012.

[47] S. University, "Stanford University," 2018. [Online]. Available: http://deeplearning.stanford.edu/tutorial/supervised/SoftmaxRegression/. [Accessed July 2019].

[48] M. Rizwan, "engMRK," 2018. [Online]. Available: engmrk.com/lenet-5-a-classic-cnn-architecture/amp/. [Accessed 30 May 2019].

[49] M. B. H. a. A. M. A. Z. Selmi, "Deep Learning System for Automatic License Plate Detection and Recognition," in *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, 2017.

[50] D. C. B. a. A. Aggarwal, "Automatic text recognition in natural scene and its translation into user defined language," in *International Conference on Parallel, Distributed and Grid Computing*, Solan, 2014.

[51] H. L. a. D. Doermann, "Video indexing and retrieval based on recognized tex," in *IEEE Workshop on Multimedia Signal Processing*, 2002 .

[52] X. S. Y. S. Y. G. Hong Liang, "Text feature extraction based on deep Learning: a review," *EURASIP Journal on Wireless Communicastions & Networking* , vol. 211, 2017.

[53] I. Goodfellow, "Deep Learning," in *Adaptive Computation and Machine Learning series*, 2016, p. 290.

Appendix

Appendix A: Sample of Extracted Key Frames from Scene Videos

(a)

(b)

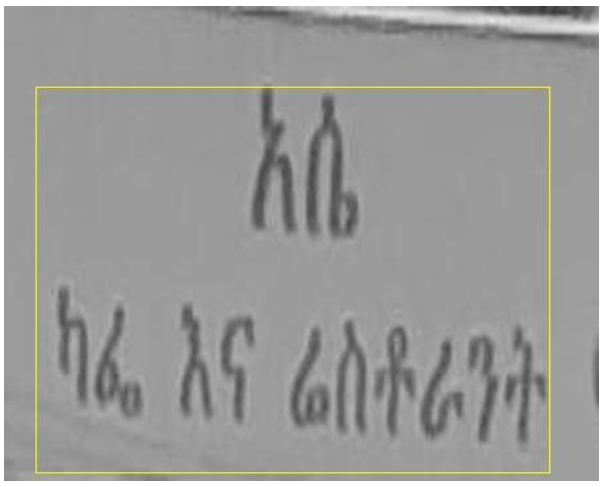
These images are the results of application of Histogram difference Algorithm

(c)

(d)

These images are a result of the application of histogram difference algorithm.

Appendix B: Result of Detected Text Blocks using Faster R-CNN

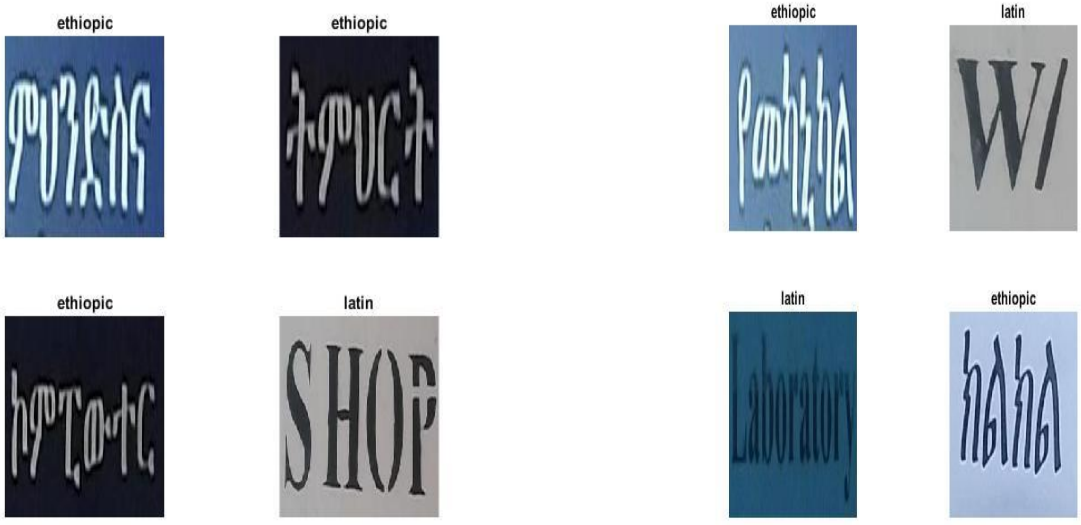


(a)

(b)

Using Faster R-CNN Model we have found these text regions during Testing the Network.

Appendix D: Result of Identified scripts using Transfer Learning



(a) (b)

Applying Transfer Learning on the Previously trained network on millions of training images and configuring the final layer of the network with our task specific classes (Latin & Ethiopic Scripts) we have found the above script identification in MATLAB environment.

Appendix E: Sample Implemented codes

1. Reading Video and Extracted Key Frames

```
clc;
clear all;
V = 'b5.mp4';       %Video Name
xyloObj = VideoReader(V);    %Using video reader reading video

   %Extracting frames
```

```
   T= xyloObj.NumberOfFrames            % Calculating number of
frames
   for g=1:T
           p=read( xyloObj,g);            % Retrieve data from
video
           if(g~=  xyloObj.NumberOfFrames)
                 J=read( xyloObj,g+1);
                 th=difference(p,J);             %To  calculate
histogram difference between two frames
                 X(g)=th;
           end

   end
   %calculating  mean  and  standard  deviation  and  extracting
frames
  % mean=mean2(X)
   %std=std2(X)
   threshold=std+mean*4
   for g=1: T
       p =  read(xyloObj,g);
       if(g~=xyloObj.NumberOfFrames)
        J=   read(xyloObj,g+1);
        th=difference(p,J);
        if(th>mean)    % Greater than threshold select as a key
frame


filename  =  fullfile('Tree',  sprintf('b5laa_%05d.JPG',  g));
%Writing the keyframes
imwrite(J, filename);
```

```
        end
        end
    end
```

2. Text Detection Using Faster R-CNN

```
%%
% Load training data.
data = load('eltrainingData.mat');


%trainingData= data.eltrainingData;
inputLayer = imageInputLayer([227 227 1]);
middleLayers = [
    convolution2dLayer([11  11],  96,'stride',4,  'Padding',  0,
'NumChannels',1 )
    reluLayer()
    crossChannelNormalizationLayer(5)
    maxPooling2dLayer(3,'stride',2, 'Padding',0)
    convolution2dLayer([5  5],  256,'stride',1,  'Padding',  2,
'NumChannels',96)
    reluLayer()
    crossChannelNormalizationLayer(5)
    maxPooling2dLayer(3,'stride',2, 'Padding',0)
```

```
    convolution2dLayer([3    3],    384,'stride',1,    'Padding',
1,'NumChannels',256)

    reluLayer()

    convolution2dLayer([3   3],   384,'stride',1,   'Padding',   1,
'NumChannels',384)

    reluLayer()

     convolution2dLayer([3   3],   384,'stride',1,   'Padding',   1,
'NumChannels',384)

    reluLayer()

    convolution2dLayer([3    3],    384,'stride',1,    'Padding',
1,'NumChannels',384)

    reluLayer()

     convolution2dLayer([3    3],    256,'stride',1,    'Padding',
1,'NumChannels',384)

    reluLayer()

    maxPooling2dLayer(3,'stride',2, 'Padding',0)

    ];

finalLayers = [

    % Add a fully connected layer with 64 output neurons. The
output size

    % of this layer will be an array with a length of 64.

    fullyConnectedLayer(4096)

    reluLayer()

    dropoutLayer(0.5)

    fullyConnectedLayer(4096)
```

```
    reluLayer()
    dropoutLayer(0.5)
      fullyConnectedLayer(width(eltrainingData))
      % Add the softmax loss layer and classification layer.
      softmaxLayer()
      classificationLayer()
  ];
layers = [
    inputLayer
    middleLayers
    finalLayers
    ]
optionsStage1 = trainingOptions('sgdm', ...
    'MiniBatchSize', 100, ...
    'MaxEpochs', 50, ...
    'InitialLearnRate', 1e-5, ...
    'CheckpointPath', tempdir);
% Options for step 2
optionsStage2 = trainingOptions('sgdm', ...
    'MiniBatchSize', 200, ...
    'MaxEpochs', 50, ...
    'InitialLearnRate', 1e-5, ...
    'CheckpointPath', tempdir);
```

```
% Options for step 3.
optionsStage3 = trainingOptions('sgdm', ...
    'MiniBatchSize', 100, ...
    'MaxEpochs', 50, ...
    'InitialLearnRate', 1e-6, ...
    'CheckpointPath', tempdir);
% Options for step 4.
optionsStage4 = trainingOptions('sgdm', ...
    'MiniBatchSize', 200, ...
    'MaxEpochs', 50, ...
    'InitialLearnRate', 1e-6, ...
    'CheckpointPath', tempdir);
options = [
    optionsStage1
    optionsStage2
    optionsStage3
    optionsStage4
    ];
tic
% A trained network is loaded from disk to save time when running
the
% example. Set this flag to true to train the network.
doTrainingAndEval = true %false;
```

```
if doTrainingAndEval
    % Set random seed to ensure example training reproducibility.
    rng(0);
      % Train Faster R-CNN detector. Select a BoxPyramidScale of
1.2 to allow
      % for finer resolution for multiscale object detection.
      detector   =   trainFasterRCNNObjectDetector(eltrainingData,
layers, options, ...
          'NegativeOverlapRange', [0 0.3], ...
          'PositiveOverlapRange', [0.6 1], ...
          'SmallestImageDimension', [500], ...
          'BoxPyramidScale', 1.2);
      % Test the Fast R-CNN detector on a test image.
      %resultsStruct = struct([]);
      %for i= 1: height(trainingData)
%I = imread(trainingData.imageFilename{i});
 % Run the detector.
        %[bbox, score, label] = detect(detector, I);
        % Collect the results.
        %resultsStruct(i).Boxes = bbox;
        %resultsStruct(i).Scores = score;
        %resultsStruct(i).Labels = label;
      %end
```

```
    % Convert the results into a table.
    %results = struct2table(resultsStruct);


%Extract the Ground Truth Data from the Training Data
%expectedResults = trainingData(:, 2:end);
%Evaluate the Precision of The Detection
%[ap, recall, precision] = evaluateDetectionPrecision(results,
expectedResults);
%figure
%plot(recall,precision)
%xlabel('Recall')
%ylabel('Precision')
%grid on
%title(sprintf('Average Precision = %.2f', ap))
%%
% Display detection results.
%detectedImg = insertShape(img, 'Rectangle', bbox);


%figure
%imshow(detectedImg)
  else
      %         'NumStrongestRegions', 500, ...
```

```
%         'MinBoxSizes', [21 21], ...
    % Load pretrained detector for the example.
    detector = data.detector;
end
toc
```

4. Script Identification

```
unzip(fullfile('C:\Users\KIRUBEL\Documents\MATLAB','netrclData.
zip'));


imds = imageDatastore('netrclData',...
    'IncludeSubfolders',true,...
    'LabelSource','foldernames');


[imdsTrain,imdsValidation]                                    =
splitEachLabel(imds,0.7,'randomized');


numTrainingImages=numel(imdsTrain.Labels);
idx= randperm(numTrainingImages,16);
figure
for i=1:16
subplot(4,4,i)
I=readimage(imdsTrain,idx(i));
imshow(I)
```

```
end
net = alexnet;
%displaying the Network Architecture
net.Layers;
%%checking the image input size
inputSize = net.Layers(1).InputSize
layersTransfer = net.Layers(1:end-3);
%%looking the number of labels in the training images
numClasses = numel(categories(imdsTrain.Labels))
%%defining the layers to be trained
layers = [
    layersTransfer

fullyConnectedLayer(numClasses,'WeightLearnRateFactor',20,'Bias
LearnRateFactor',20)
    softmaxLayer
    classificationLayer];
pixelRange = [-30 30];
imageAugmenter = imageDataAugmenter( ...
    'RandXReflection',true, ...
    'RandXTranslation',pixelRange, ...
    'RandYTranslation',pixelRange);
augimdsTrain = augmentedImageDatastore(inputSize(1:2),imdsTrain,
...
```

```
    'DataAugmentation',imageAugmenter);
%automatically resizing validation images
augimdsValidation=augmentedImageDatastore(inputSize(1:2),imdsVa
lidation);
options = trainingOptions('sgdm', ...
    'MiniBatchSize',10, ...
    'MaxEpochs',6, ...
    'InitialLearnRate',1e-4, ...
    'ValidationData',augimdsValidation, ...
    'ValidationFrequency',3, ...
    'ValidationPatience',Inf, ...
    'Verbose',false, ...
    'Plots','training-progress', 'CheckpointPath', tempdir);
%%train the network that consists of the transfered and new layers
netTransfer = trainNetwork(augimdsTrain,layers,options);
%%Classify the Validation images
[YPred,scores] = classify(netTransfer,augimdsValidation);
%%Calculate the accuracy on the Validation set
YValidation = imdsValidation.Labels;
accuracy = mean(YPred == YValidation)
```