ISSN:0975-9646

Mohit K.Chauhan et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 8285-8292

Automatic Text Detection and Information Retrieval on Mobile

Mohit K.Chauhan[#1],Gaurav Kumar[*2]

[#]*Department of Computer Science & Engineering, Shobhit University*

NH- 58,Modipuram, Meerut,Uttar Pradesh, India

[*]*mCarbon Tech Innovation Pvt Ltd*

Sector 2 ,Noida Uttar Pradesh,India

***Abstract*--The automatic text detection within a natural image is an important problem in many applications. Text detection in natural images has gained much attention in the last years as it is a primary step towards fully autonomous text reorganization. This paper tackles the problem of recognizing text in images captured by camera from mobile Phone. After recognize the text we hit the query for getting the information of text like Songs. If we receive the songs list in the response of query then we show the list to the user. User can Download/Play these songs in Mobile Client. Text Detection is performed in the four step these are Image labeling, Finding boundary and Generating X, Y coordinate pixel array, Matching connected pixels with learned set, Forming words. The performance of the approach is demonstrated by presenting promising experimental results for a set of images captured by camera from mobile.**

***Keywords*-- Text detection, Text segmentation, Text recognition, Information retrieval, Songs Download, Songs Play, Search song, information of text, Client- server, Mobile Client.**

I. INTRODUCTION

Now the Time has been changed. Everybody has the mobile and there are many applications available to make the task easy through mobile, same like that this paper describes an application. Everybody knows, TEXT in any form or place has high purpose and contains more information related to the place and helps us to understand the objective more easily. The rapid growth in digital technologies and gadgets outfitted with megapixel cameras and invention of latest touch screen method in digital devices like mobile, PDA, etc., increase the demand for information retrieval and it leads to many new research challenges. Text detection and segmentation from natural scene images are useful in many applications. the primary property of scene text such as, high contrast against background, uniform colors are difficult to preserve in real application. When the system scans whole image for texts, text pixels with low contrast and non uniform lighting could be confused as background due to similar colors. Automatic text recognition from natural images receives a growing attention because of potential applications in image retrieval, robotics and intelligent transport system. Camera-based document analysis becomes a real possibility with the increasing resolution and availability of digital cameras.

The main objective tackles the problem of recognizing text in images captured by camera from mobile Phone. After recognize the text we hit the query for getting the information of text like Songs. If we receive the songs list in the response of query then we show the list to the user. User can Download/Play these songs in Mobile Client.

For successfully execution of this project we use many Algorithms. One of them is Character Recognition Algorithm. This Algorithm relies on the set of learned character. It compares the character in the captured image file to the character in this learned set. Generating the learned set is quite simple. Learned set requires an image file with the desired characters in the desired font be created, and a text file representing the character in this image file.

There are four basic algorithms describe in this paper to achieve the task.

1. Image labeling Algorithm: It uses the Two-pass algorithm, which relatively simple to implement and understand, the two-pass algorithm iterates through 2-dimensional, binary data. The algorithm makes two passes over the image: one pass to record equivalences and assign temporary labels and the second to replace each temporary label by the label of its equivalence class.

2. Finding boundary and Generating X, Y coordinate pixel array: This is done from the labels from the above algorithm, then it's merely adding all the connected X, Y coordinates in the connect component list.

3. Matching Character: Finally this most easy task. We match the connect component bit array with the .xml data. Each pixel are matched according the X, Y coordinates. The fully matched pixels coordinates are the matched character from the xml.

4. Forming Words: In this algorithm, maintain the Left index and Right index for each character. The Left index represents the left most index of the character in the bitmap specified initially in the blog. The Right index represents the right most X coordinate of the character. When the difference

coordinates of current character and previous character is less than 3 pixels then they are joined. This algorithm is quite simple. But you can extend to join words according to grammar in the dictionary.

The remaining part of our paper is organized as follows: In section II we will discuss the related work done in field of image to text detection and in section III we discuss the four basic algorithms. These algorithms are the heart of this project. Tools and technology will be discussed in section IV and finally in section V we conclude the paper.

II. RELATED WORK

In this section, we review existing methods for text detection and text recognition. These two problems are often addressed separately in the literature. Several approaches for text detection in images have been proposed in the past. Techniques for automatic detection and translation of text in images have been proposed. Most of these methods aim to detect the characters based on general properties of character pixels. Many researchers working on text detection and thresholding algorithm with various approaches have achieved good performance based on some constraints.

A. Text detection

Text can be detected by exploiting the discriminate properties of text characters such as the vertical edge density, the texture or the edge orientation variance. One early approach for localizing text in covers, assumed that text characters were contained in regions of high horizontal variance satisfying certain spatial properties that could be exploited in a connected component analysis process. These methods are fast but also produce many false alarms because many background regions may also have strong horizontal contrast. The method of Wu et al. for text localization is based on texture segmentation. Texture features are computed at each pixel from the derivatives of the image at Di Herent scales. Using a K-means algorithm, pixels are classified into three classes in the feature space. The class with highest energy in this space indicates text while the two others indicate non-text and uncertainty. However, the segmentation quality is very sensitive to background noise and image content and the feature extraction is computationally expensive. More recently, Garcia et al. proposed a new feature referred to as variance of edge orientation. This relies on the fact that text strings contain edges in many orientations. Variation of edge orientation was computed in local area from image gradient and combined with edge features for locating text blocks. The method, however, may exhibit some problems for characters with strong parallel edges characteristics such as “i” or “1”.

The above manually designed heuristic features usually perform fast detection but are not very robust when the background texture is very complex. As an alternative, a few systems considered machine learning tools to perform the text detection [10,15]. These systems extracted wavelet [10] or derivative features [15] from 3xed-size blocks of pixels and classi3ed the feature vectors into text or non-text using arti3cial neural networks. However, since the neural network based classi3cation was applied to all the possible positions of the whole image, the detection system was not efficient in terms of computation cost and produced unsatisfactory false alarm and rejection rates.

B. Text recognition

Since commercial OCR engines achieve high recognition performance when processing black and white images at high resolution, almost all the methods in the literature that addressed the issue of text recognition in complex images and videos employed an OCR system. However, these OCR systems cannot be applied directly on regions extracted by a text localization procedure. Experience shows that OCR performance in this context is quite unstable [6], and significantly depends on the segmentation quality, in the sense that errors made in the segmentation are directly forwarded to the OCR.

Some bottom-up based-techniques addressed the segmentation problem for text recognition. For instance, Lienhart [6] and Bunke [7] clustered text pixels from images using standard image segmentation or color clustering algorithm. Although these methods can avoid explicit text localization, they are very sensitive to character size, noise and background patterns. On the other hand, most top-down text segmentation methods are performed after text string localization. These methods assume that the gray-scale distribution is bimodal and that characters a priori correspond to either the white part or the black part, but without providing a way of choosing which of the two possibilities applies. Great eHorts are thus devoted to performing better binarization, combining global and local thresholding [16], M-estimation [17], or simple smoothing [9]. However, these methods are unable to 3lter out background regions with similar gray-scale values to the characters. If the character gray-scale value is known, text enhancement methods can help the binarization process [18]. However, without proper estimation of the character scale, the designed 3lters cannot enhance character strokes with diHerent thickness [19]. In videos, multi-frame enhancement [10] can also reduce the in Nuence of background regions, but only when text and background have diHerent movements.

III. FOUR BASIC ALGORITHMS

There are four basic Algorithms describe completely in this paper. These Algorithms are the heart of this paper. These are given below.

1. Image Labeling Algorithm.
2. Finding boundary and Generating X, Y Coordinate pixel array
3. Matching Character
4. Forming words

A. Image Labeling Algorithm:

It uses the Two-pass algorithm, which relatively simple to implement and understand, the two-pass algorithm iterates through 2-dimensional, binary data. The algorithm makes two passes over the image: one pass to record equivalences and assign temporary labels and the second to replace each temporary label by the label of its equivalence class.

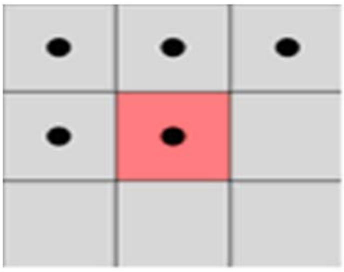

Fig 1 Image with Pixels

Connectivity checks are carried out by checking the labels of pixels that are North -East, North, North - West and West of the current pixel (assuming 8-connectivity). 4-connectivity uses only North and West neighbors of the current pixel. The following conditions are checked to determine the value of the label to be assigned to the current.

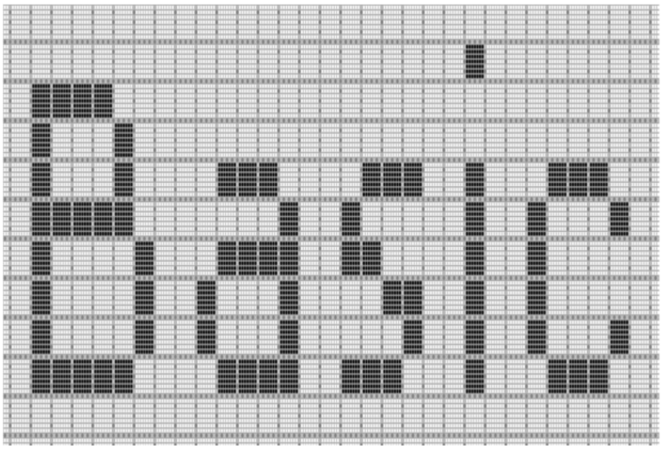

Fig 2 Pixels representation in the captured image

The above-enlarged image is a pixel representation, which serves purpose for our discussion. Every pixel in the bitmap image is represented by its X and Y coordinates. The letter "B" in the above example shows how all the pixels are connected. The image-labeling algorithm will label the entire connected pixel with the same label. The UML diagram below illustrates the flow of the algorithm.

B. Finding boundary and Generating X, Y Coordinate pixel array:

Finding boundary and generating X, Y coordinate pixel array from the labels from the above algorithm, then its merely adding all the connected X, Y coordinates in the connect component list. The above image shows all the connected component boundary, which marked in yellow. I have highlighted the boundary (X, Y) coordinates of the connected component "a".

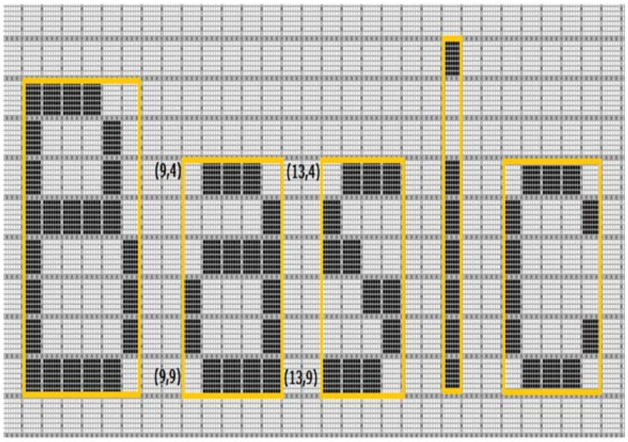

Fig 3 Sample representation of X,Y Coordinated

- LeftXCor: - Starting left X coordinate of the connected component. For the connected component "a" it is 9.
- RightXCor: - Ending left X coordinate of the connected component. For the connected component "a" it is 13.
- TopYIndex: - Starting or the lowest Y coordinate of the connected component. For the connected component "a" it is 4.
- BottomYIndex: - Ending or the highest Y coordinates of the connected component. For the connected component "a" it is 9.
- Width: - Width of the connect component will be RightXCor – LeftXCor.In case of "a" it will be 13 - 9 = 4.But add one since it start from zero so the width will be 5. Height: -Similarly height of the connected component will be BottonYCor – TopYCor.In this case for "a" the height will be 6.
- PixelCoordinate [,]: - As per the height and width of the connected component initialize the two-dimensional array. For "a" it will be [5, 6]. For the appropriate connected pixel coordinate set the bit high. For e.g. for the connected component "a" (9, 4) coordinate there is no connected pixel so set [0,0] to false. Since (9, 4) is the starting X, Y coordinate, so it is (0, 0). Similarly for (13, 9) there is a connected coordinate so [4, 5] is set to true. Similarly for the entire connected component X and Y Coordinates.

C. Matching Character Algorithm:

Finally this most easy task. We match the connect component bit array with the xml data. Each pixel are matched according the X, Y coordinates. The fully matched pixels coordinates are the matched character from the xml.

	min	max
Height	6	460
Width	2	320

TABLE 1 (Font size in Pixel)

Up until the grouping each character has been treated separately, and the context in which each character appears has usually not been exploited. However, in advanced optical text recognition problems, a system consisting only of single-character recognition will not be sufficient. Even the best recognition systems will not give 100% percent correct identification of all characters, but some of these errors may be detected or even corrected by the use of context.

Height	# of chars	Chars detected
<=10	705	55.2%
11-30	7571	90.0%
30>	13754	95.2%
>=6	22030	92.1%

TABLE 2

The classification is the process of identifying each character and assigning to it the correct character class. In the following sections two different approaches for classification in character recognition are discussed. First decision-theoretic recognition is treated.

These methods are used when the description of the character can be numerically represented in a feature vector. We may also have pattern characteristics derived from the physical structure of the character, which are not as easily quantified. In these cases the relationship between the characteristics may be of importance when deciding on class membership. For instance, if we know that a character consists of one vertical and one horizontal stroke, it may be either an “L” or a “T”, and the relationship between the two strokes is needed to distinguish the characters. A structural approach is then needed.

D. *Forming words Algorithm:*

As per the above example "Basic". We maintain the Left index and Right index for each character. The Left index represents the left most index of the character in the bitmap specified initially in the blog. The Right index represents the right most X coordinate of character. When the difference coordinates of current character and previous character is less than 3 pixels then they are joined. This algorithm is quite simple. But you can extend to join words according to grammar in the Dictionary .

The result of plain symbol recognition on a document is a set of individual symbols. However, these symbols in themselves do usually not contain enough information. Instead we would like to associate the individual symbols that belong to the same string with each other, making up words and numbers. The process of performing this association of symbols into strings, is commonly referred to as grouping. The grouping of the symbols into strings is based on the symbols’ location in the document. Symbols that are found to be sufficiently close are grouped together.

For fonts with fixed pitch the process of grouping is fairly easy, as the position of each character is known. For typeset characters the distance between characters are variable. However, the Distance between words are usually significantly larger than the distance between characters, and grouping is therefore still possible. The real problems occur for handwritten characters or when the text is skewed.

IV. TOOLS AND TECHNOLOGYS

A. *iOS Technologies:*

iOS is the operating system that runs on iPhone, iPod touch, and iPad devices. The operating system manages the device hardware and provides the technologies required to implement native apps. The operating system also ships with various system apps, such as Phone, Mail, and Safari, that provide standard system services to the user.

The iOS Software Development Kit (SDK) contains the tools and interfaces needed to develop, install, run, and test native apps that appear on an iOS device’s Home screen. Native apps are built using the iOS system frameworks and Objective-C language and run directly on iOS. Unlike web apps, native apps are installed physically on a device and are therefore always available to the user, even when the device is in Airplane mode. They reside next to other system apps and both the app and any user data is synced to the user’s computer through iTunes.

The iOS SDK provides the resources you need to develop native iOS apps. Therefore, understanding a little about the technologies and tools that make up this SDK can help you make better choices about how to design and implement your apps.

B. *iOS Architecture is Layered:*

At the highest level, iOS acts as an intermediary between the underlying hardware and the apps that appear on the screen. The apps you create rarely talk to the underlying hardware directly. Instead, apps communicate with the hardware through a set of well-defined system interfaces that protect your app from hardware changes. This abstraction makes it easy to write apps that work consistently on devices with different hardware capabilities.

The implementation of iOS technologies can also be viewed as a set of layers, which are shown in Figure I-1. At the lower layers of the system are the fundamental services and technologies on which all apps rely higher-level layers contain more sophisticated services and technologies.

As you write your code, you should prefer the use of higher-level frameworks over lower-level frameworks whenever possible. The higher-level frameworks are there to provide object-oriented abstractions for lower-level constructs. These abstractions generally make it much easier to write code because they reduce the amount of code you have to write and encapsulate potentially complex features, such as sockets and threads. Although they abstract out lower-level technologies, they do not mask those technologies from you. The lower-level frameworks are still available for developers who prefer to use them or who want to use aspects of those frameworks that are not exposed by the higher layers.

C. Basic Syntax for a Class:

Most object oriented languages have something called a class which encapsulates data and provides access to it. In Objective-C a class can contain instance variables, methods and/or properties. Let us look at an example for a class called Book.

All classes are declared in two parts:

.h – contains the interface which is a declaration of the class structure

.m – contains the implementation of all the methods

The member variables by default are set to private hence you have to write accessor methods which are your getter and setter methods. In the example above, the class Book has getAuthor and setAuthor methods to allow getting and setting the author of a book externally.

D. Methods:

The method signature starts with either a plus or minus, the method name and then followed by arguments.

There are two types of methods in Objective-C:

- Class methods
- Instance methods.

Book *book = [Book createBookWithTitle:@"A Brave New World"];

A Class method is like a static method in Java or C#. It is not tied to an instance of a class and has no knowledge of the instance variables. Class methods should be mainly used for creating a new object, utility or shared instance for a singleton class. The above method returns an object which is the instance of the class Book.

Instance methods as the name implies are scoped to a particular instance of a class.

[book title]; // method call without any arguments

Instance methods can also refer to self or super. self refers to its own instance whereas super refers to the parent class.

E. Properties:

When you have too many member variables then having to sit and write getter/setter methods for them can become tedious. Not to mention, the code looks bloated. That is why Objective-C has feature called property. The property declaration automatically declares getter/setter methods. Let's look at a modified version of the Book class with property

The property syntax is very basic starts with @property followed by keywords which allow you define whether you want only a getter method or both a getter and setter methods.

In your implementation you want to synthesize your properties which will actually create the accessor methods.

@synthesize title, author;

You can always create your own custom getter and setter methods in your implementation.

1) Implementation: All the methods in the implementation must be defined between @implmentation and @end

The @synthesize generates the getter and setter methods as needed.

Both the init methods first make a call to the parent class [super init] so that it can initialize and then return an instance.The if statement checks to see if self is a valid instance.

You must be wondering what is id? id allows you to create a loosely or weakly typed variable where the type or class of the variable is determined at runtime. Even though id does not have an asterisk, it denotes a pointer to an object therefore you cannot use it for a scalar type such as int, float, double or char.

In Objective-C the way to release an object is to use the method release or autorelease. There are a whole set of rules around memory management but to sum it up anytime you create or allocate you own the object and hence must release it.

Both the create methods use the method autorelease which basically notifies the system that the object book should be released in the immediate future to release ownership.

F. Convention:

The two create methods in the Book class are known as convenience constructors because they combine the two steps of allocating and initializing to return instances of the class. Convention dictates that such methods not start with the word create but with the name of the class. So below is the appropriate way to define those methods .

G. Methodology of Image Retrieval System:

To solve the defined handwritten character recognition problem of classification we used Methodology of image retrieval system.

Pre-processing:

The raw data, depending on the data acquisition type, is subjected to a number of preliminary processing steps to make it usable in the decimates Retrievaliptive stages of character analysis. Preprocessing aims to produce data that comes from server or local machine. The main objectives of preprocessing are:

1. Noise reduction,
2. Normalization of the data

The sub-images have to be cropped sharp to the border of the character in order to standardize the sub-images. Finding the maximum row and column with 1s and with the peak point, increase and decrease the counter until meeting the white space, or the line with all 0s does the image standardization. This technique is shown in figure below where a character “S” is being cropped and resized.

Algorithm	Test-62	Sample-62	Sample - 36
A.K. Jain and B. Yu,1998 [1]	67.0%	-	-
Yi-Fang Pan; Xinwen Hour,2008 [20]	-	81.4	-
S. P. Chowdhury, S. Dhār, 2009 [15]	-	-	84.54
This Paper	81.7%	81.4%	85.5%

TABLE 3 Test Recognition Accuracy on ICDAR

Segmentation:

Segmentation is an important stage, because the extent one can reach in separation of words, lines or characters directly affects the recognition rate of the image Retrievaliptive.

The two types of segmentation are:

-External Segmentation

-Internal Segmentation

This work is a contribution of the Semantic gap between the high level concept of the textual details and the low level features of the image. The thesis highlights the importance of reasoning and contextual knowledge in the image understanding process, emphasizes the limitations of current approaches and tries to provide solutions that can overcome these limitations. We have evaluated that the result retrieved by taking the semantic approach gives better and more precise results as compared to the normal keyword based search, by introducing the concept of the low level features of the image in the ontology further refines the search.

Representation:

Image representation plays one of the most important roles in a recognition system. In the simplest case, gray-level or blue-level are fed to a recognizer. However, in most of the recognition systems, in order to avoid extra complexity and to in Image Retrieval ease the accuracy of the algorithms, a more compact and characteristic representation is required.

A good survey on feature extraction methods for character recognition can be found in [13]. We show all the images in machine and show in a grid view:

Training and Recognition Techniques:

IMAGE RETRIEVAL systems extensively use the methodologies of four basic algorithm, which assigns an unknown sample into a predefined class. Matching connected pixels with learned set is the most importent algorithms , as suggested as

-Character Matching,

-Word Matching

-Statistical Techniques,

-Structural Techniques

Post Processing:

It is well known that humans read by context up to 60% for careless handwriting. While preprocessing tries to clean” the document in a certain sense, it may remove important information, since the context information is not available at this stage. The review of the recent image retrival research indicates minor improvements, when only shape recognition of the character is considered. Therefore, the incorporation of context and shape information in all the stages of image retrival systems. This is done in the post processing stage with a feedback to the early stages of image retrival. The simplest way of incorporating the context information is the utilization of a dictionary for correcting the minor mistakes of the image retrival systems.

When user click the details of song then this song download automatically from the server which contain the binary data of all songs. It already provide the meta data , we hit to download the binay data of song. When we get the binary data ,we pass it to the Player.

Player play the song. we can controll the sound of player.User can play or pause the music. This player screen showes the song information with sound.

V. CONCLUSION AND FUTURE SCOPE

In the End of this paper we conclude that we can solve the problem of automatic text detection within a natural image is an in many applications. This paper tackles the problem of recognizing text in images captured by camera from mobile Phone. After recognize the text we hit the query for getting the information of text like Songs. If we receive the songs list in the response of query then we show the list to the user. User can Download/Play these songs in Mobile Client. Text Detection is performed in the four step these are Image labeling, Finding boundary and Generating X, Y coordinate pixel array, Matching connected pixels with learned set, Forming words. The performance of the approach is demonstrated by presenting promising experimental results for a set of images captured by camera from mobile.

How the Problem is being extended For Dissertation:

Extract text from image (JPG, JPEG, BMP, TIFF, GIF) and convert into editable Word output formats. Converted documents look exactly like the original word.

The large number of research publications in the field of text-based information retrieval especially in recent years shows that it is very active and that it is starting to get more attention. This will hopefully advance the field as new tools and technologies will be developed and performance will increase. Text-based information retrieval retrieval definitely has a large potential in the text domain. The amount of visual data produced shows the importance of developing new and alternative access methods to complement text. . Text-based methods can be used on a large variety of images and in a wide area of applications. Still, much work needs to be done to produce running applications and not only research prototypes. When looking at most current systems, it becomes clear that few to none of them are actually in routine use.

	Total	Recognized	% over Total
Char	16664	15656	83%
Word	3304	2926	72.4%

TABLE 4

The previous proposed systems focused more on the text when it came to semantic web; the work was very limited in the field of images. Our system gave a balanced approach to the text as well as the image generating a result in a consistent format. The most important module, which is introduced in our system, is the concept of extraction of low level features and storing them in a text format hence making it easier to access. The approach proposed in this thesis will benefit the text community to a large extent as large collections of text based information can be indexed and retrieved semantically.

Future Work:

To implement a full-fledged image retrieval system with high-level semantics requires the integration of salient low level feature extraction, effective learning of high-level semantics, friendly user interface, and efficient indexing tool. Most systems understandably limit their contributions to one or two of these components. A TBIR framework providing a more balanced view of all the text components is in need.

We are currently working on improving the efficiency by integrating the feature extraction mechanism within the application as of now the user has to manually enter the details of the low level features extracted by the tool (image) so to make the use of the application more user-friendly and easy, automatic annotation can be introduced in the same system. Future improvements to this implementation also include automating the image segmentation and feature extraction phase and using some learning and optimization techniques to make the system more efficient with different -2 font .

To sum up, while this work focuses on making semantic text of images explicit, the approach discussed can be broadly applicable to a number of domains, where representation layer demands special attention.

We are also currently working on improving the limitation of this project. There are following limitations

- The text should be in Arial font.
- The text color should be black.
- The background contain single color(not black).

ACKNOWLEDGMENT

Before going into thick of things I would like to add a few heartfelt words for the people who were part of this project in numerous ways. A large number of teachers, students and computer lovers have made valuable suggestions which have been incorporated in this work. It is not possible to acknowledge all of them individually. I take this opportunity to express my profound gratitude and ineptness to them.

I am gratefully thankful to Mr. Gaurav Kumar**,** iOS Apps Developer in mCarbon Tech Innovation Pvt Ltd for his kind cooperation, excellent suggestions, moral support and guidance.

The love, affection and encouragement that I got from my family members is unforgettable, they are my strength, my support and my pride. How can I ever thank to my dearest Family Members for always cheering up and spicing up my life

At least we thank and praise the Almighty for his constant blessing and grace through every step of our life.

REFRENCES

[1] A.K. Jain and B. Yu, “Automatic Text Location in Images”, Pattern Recognition, vol. 31, no. 12, pp. 2,055-2076, Nov. 1998.

[2] V. Wu, R. Manmatha, and E.M. Riseman, “Finding Text In Images”, Proc. the Second Int'l Conf. Digital Libraries, pp. 1-10, Philadelphia, PA, July 1997.

[3] Mohamed Kamel and Aiguo Zhao, “Extraction of Binary Character/ graphics Images from Gray scale Documents Images”, Computer Vision, graphics and Image Processing, 55(3), pp. 203-217, May. 1993.

[4] L.A. Fletcher and R. Kasturi, “A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images and Audio”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 10, no. 6, pp. 910-918, Nov. 1988.

[5] A. Carzaniga, D. S. Rosenblum and A. L. Wolf "Design and Evaluation of a Wide-Area Event Notification Service", ACM *Trans. Computer Systems*, vol. 19, no. 3, pp.332 -383 2001

[6] P. Deolasee "Adaptive Push-Pull: Dissemination of Dynamic Web Data", IEEE *Trans. Computers*, vol. 51, no. 6, pp.652 -668 2001

[7] M. Capriccio, A. Carzaniga and A. L. Wolf "Design and Evaluation of a Support Service for Mobile, Wireless Publish/Subscribe Applications", IEEE *Trans. Software Eng.*, vol. 29, no. 12, pp.1059 – 1071 2003

[8] C. Intanagonwiwat "Directed Diffusion for Wireless Sensor Networking", IEEE*/ACM Trans. Networking*, vol. 11, no. 1, pp.2 -16 2003

[9] P. Bellavista, A. Corradi and C. Stefanelli, "Protection and Interoperability for Mobile Agents: A Secure and Open Programming Environment," *IEICE Trans. Comm.,* May 2000, pp. 961-972.

[10] Al-Jobouri, L., Fleury, M., Ghanbari, M., "Multicast and unicast video streaming and audio with rate less channel-coding over wireless broadband", Consumer *Communications and Networking Conference (CCNC), 2012 IEEE,* On page(s): 737 – 741

[11] Pedersen, M.V., Fitzek, F.H.P., "Mobile Clouds: The New Content Distribution Platform", Proceedings *of the IEEE,* on page(s): 1400 - 1403 Volume: 100, Issue: Special Centennial Issue, May 2012

[12] P. Bellavista, A. Corradi and C. Stefanelli, "Protection and Interoperability for Mobile Agents: A Secure and Open Programming Environment," *IEICE Trans. Comm.,* May 2000, pp. 961-972.

[13] R. Kasturi, D. Golgof, P *International Workshop on,* On page(s): 74 - 78. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova and J. Zhang "Framework for performance evaluation of face, text and vehicle detection and tracking in video: Data, metrics, and protocol", IEEE *Trans. Pattern Anal. Mach.*

Intell, vol. 31, no. 2, pp.319 -336 2009

[14] J. Zang and R. Kasturi "Extraction of text objects in video documents: Recent progress", Proc. *Int. Workshop DAS*, pp.5 -17 2008

[15] S. P. Chowdhury, S. Dhar, A. K. Das, B. Chanda and K. McMenemy "Robust extraction of text from camera images", Proc. *ICDAR*, pp.1280 -1284 2009

[16] Y. M. Y. Hasan and L. J. Karam "Morphological text extraction from images", IEEE *Trans. Image Process,* vol. 9, no. 11, pp.1978 – 1983 2000

[17] B. Epshtein, E. Ofek and Y. Wexler "Detecting text in nature scenes with stroke width transform", Proc. *IEEE Conf. Comput. Vis. Pattern Recognition,* pp.2963 -2970 2010

[18] Chucai Yi, YingLi Tian "Text String Detection From Natural Scenes by Structure-Based Partition and Grouping", Image *Processing, IEEE Transactions on,* on page(s): 2594 - 2605 Volume: 20, Issue: 9, Sept. 2011

[19] Shivakumara, P.; Sreedhar, R.P.; Trung Quy Phan; Shijian Lu; Tan, C.L. "Multioriented Video Scene Text Detection Through Bayesian Classification and Boundary Growing", Circuits *and Systems for Video Technology, IEEE Transactions on,* On page(s): 1227 - 1235 Volume: 22, Issue: 8, Aug. 2012

[20] Yi-Feng Pan; Xinwen Hou; Cheng-Lin Liu, "A Hybrid Approach to Detect and Localize Texts in Natural Scene Images", Image *Processing, IEEE Transactions on,* On page(s): 800 - 813 Volume: 20, Issue: 3, March 2011

[21] Sharma, N. Shivakumara, P. Pal, U. Blumenstein, M. Tan, C.L. "A New Method for Arbitrarily-Oriented Text Detection in Video", Document *Analysis Systems (DAS), 2012 10th IAPR*

[22] Sharma, N. Pal, U. Blumenstein, M. Tan "Recent Advances in Video Based Document Processing: A Review", Document *Analysis Systems (DAS), 2012 10th IAPR International Workshop on,* on page(s): 63 – 68.

[23] D. Chen, J.-M. Odobez, A new method of contrast normalization for text veri3cation in complex backgrounds, IDIAP-RR-02 16, IDIAP, April 08, 2002.

[24] S. Katz, Estimation of probabilities from sparse data for the language model component of a speech recognizer, IEEE Trans. Acoust. Speech Signal Process. 35 (2005) 400–401.

[25] Y. Zhong, K. Karu, A.K. Jain, Locating text in complex color images, Pattern Recognition 10 (28) (2009) 1523–1536.

[26] V. Wu, R. Manmatha, E.M. Riseman, Finding text in images, in: Proceedings of ACM International Conference on Digital Libraries, 2010, pp. 23–26.

[27] H. Li, D. Doermann, Text enhancement in digital video using multiple frame integration, in: ACM Multimedia, 2008, pp. 385–395.

[28] C. Garcia, X. Apostolidis, Text detection and segmentation in complex color images, in: International Conference on Acoustics, Speech and Signal Processing, 2000, pp. 2326–2329.

[29] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, PAMI 6 (2000) 721–741.

[30] B. Chalmond, Image restoration using an estimated Markov model, Signal Process. 15 (2) (2005) 115–129.

[31] Shivakumara, P.; Sreedhar, R.P.; Trung Quy Phan; Shijian Lu; Tan, C.L. "Multioriented Video Scene Text Detection Through Bayesian Classification and Boundary Growing", Circuits *and Systems for Video Technology, IEEE Transactions on,* On page(s): 1227 - 1235 Volume: 22, Issue: 8, Aug. 2012

[32] Yi-Feng Pan; Xinwen Hou; Cheng-Lin Liu, "A Hybrid Approach to Detect and Localize Texts in Natural Scene Images", Image *Processing, IEEE Transactions on,* On page(s): 800 - 813 Volume: 20, Issue: 3, March 2011

[33] Sharma, N. Shivakumara, P. Pal, U. Blumenstein, M. Tan, C.L. "A New Method for Arbitrarily-Oriented Text Detection in Video", Document *Analysis Systems (DAS), 2012 10th IAPR International Workshop on,* On page(s): 74 - 78

[34] Sharma, N. Pal, U. Blumenstein, M. Tan "Recent Advances in Video Based Document Processing: A Review", Document *Analysis Systems (DAS), 2012 10th IAPR International Workshop on,* on page(s): 63 – 68.