International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 06 | June 2019 www.irjet.net p-ISSN: 2395-0072

TEXT LINE DETECTION IN CAMERA CAPUTERD IMAGES USING MATLAB GUI

Safvana Sajeev[1], Ojus Thomas Lee[2]

[1]*Post Graduation Student, Dept. of CSE, College of Engineering Kidangoor, Kerala, India*
[2]*Associate Professor & HOD, Dept. of CSE, College of Engineering Kidangoor, Kerala, India*

Abstract - *In today's digital world, the camera based text pre-processing get an important task that finds applications in many fields like Text2Speech conversion, information retrieval etc... So, this led to the development of many pre-processing methods. This paper proposes a system that combines text line identification from images containing printed and handwritten text. The Maximally Stable Extremel Region (MSER) algorithm is used to determine the character regions in the image. Based on the geometric property of the region, unwanted or dissimilar regions are removed. The Optical Character Recognition (OCR) is used to recognize each character and detect the word region. After detection, the text can be used to search for information on the internet. In addition, the detected text can be converted to text to assist visually impaired peoples. Our proposed method are tested with data sets based on scene text, street view data set and handwritten text. The percentage of exact text identification our method in scene dataset is more than 50% in any situation (Indoor, Outdoor, Book Cover, Light, Shadow etc..) , in case of street view dataset is 74% and that for handwritten dataset is 68%.*

Keywords: **Text Detection, Text2Speech Conversion, Information Gathering, MSER, OCR**

1. INTRODUCTION

The importance of image processing has increased day by day due to the availability of cost effective devices for capturing photos and videos. In today's digital world, people captured photos and videos and passed to the social media. Text is the most expansive means of communication. Text is embedded into documents or scenes for communication. It can be observable and readable by others. The narrative in image contain valuable information and can be used for numerous image and video based applications like image search web site, information recovery, reviewing text and identify text in mobile devices. The size, font, style of text and the back ground with snow, rock make text detection in camera captured images more difficult.

Text detection is the major task in text line detection. It is used in Optical Character Recognition (OCR) and preprocessing algorithms. The English language text in the images will be captured.

The main text detection challenges are divided into three groups:

1) Natural image properties: The characters in natural images are in different font style, different size, unique color, and unique font alignment.

2) Backgrounds of an image: The background includes grasses bricks, rocks, are leads to more complexity for identifying text.

3) Interference factors: The main inference factor is blurring. From a blurred image text can't be detected.

The Main Contributions of our work are

1) We use MSER algorithm to determine character regions in images.
2) OCR is used to recognize characters.
3) Provide methods to convert the detected text to voice.
4) Add methods to do internet search of identified text.
5) We use scene text, street view and handwritten data sets to test the method.

The rest of the document is organized as follows,

Section 2 describes the related works done in this paper, Section 3 and 4 explain the system architecture and its working. The Section 5 describes the implementation steps and Section 6 describes the results obtained. Finally Section 7 is conclusion.

2. RELATED WORK

The emergence of smart phones and digital cameras has made the text line detection from images in demand. In this scenario we have made a study on available methods for the text line identification from images. The indent of the study was identify the existing methods and to suggest improved methods for the text line identification. After reviewing the literature in the existing domain we have found that there exists the following methods for solving the problem.

1) MSER with Ada boost
2) State Estimation
3) Texture Based SVM
4) Chain Level Classification
5) CMSER Algorithm

6) Graph Model

In the case of MSER with Ada boost, [1] [2] [3] propose a connected component procedure that extraction the connected component and form a group of individual components that contain text. [4] [5] are suggesting procedure to detecting character content from natural images. Under the State Estimation category, [6] [7] [8] initiate state estimation procedure for the extraction of text in natural images. Here algorithm has two phases. First phase using the connected components, interline spacing is calculated. The next phase of the procedure identifies the text block using the estimated states. [9][10][11] Proposed texture based SVM method. Texture classification is like a score image. Each pixel is denoted as the possibility of the part of region which contain text. [12] Propose a connected component with SVM method to identify the characters from the image. [13] Propose arbitrary text detection in natural images using chain level classifier. Chain Level Classifier is a system that produces robust result for detecting text. [14] Suggest a method to detect road sign text detection. In addition, [15] [16] Proposed an identification mechanism for traffic sign boards.

3. SYSTEM ARCHITECTURE

The purpose of this paper is to recognize text from images. The images are scene images at shadow, light, indoor, outdoor, book cover condition. Next street view images that contain the shop name, posters. Next one is handwritten text images real time. Here we are using an OCR classifier for the purpose of text detection. The detected text is converted to speech for visually impaired peoples. And also the text is passed to Google for gather information about the text. The MSER is used to detect the regions in the image. After detect the regions unwanted regions are removed based on their geometric property. Using OCR texts/words are detected.

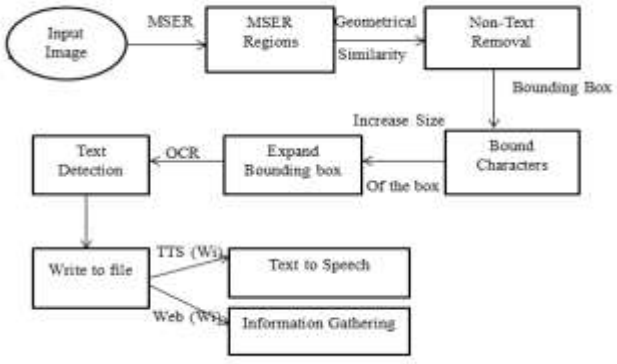

Fig -1: System Architecture

Initially upload the image and by using MSER regions are identified. The detected MSER regions are taken and remove the non-text region based on their geometrical similarity. Using bounding property each character in the text region is bounded. Then expand the bounding box to get the words/text. The OCR detects text in the bounding box. After the detection detected text is converting to speech and it is also sent to Google browser for gather information about the text content.

In the case of handwritten text, the detected regions are taken and the unwanted regions are removed. Here also the detected region contain characters are bounded. And using OCR the text is detected. Then detected text is converting to speech and sent to Google browser.

4. WORKING

Initially login to the system by using username and password. After login a main window is displayed. The project works in two streams, for the extraction of two general categories of text extraction.

The two text extractions are,

1) Text Extraction-1

2) Text Extraction-2

The **Text Extraction-1** is for scene and street view image database. The **Text Extraction-2** handle handwritten image database.

In the case of **Text Extraction-1** after we input the image, the process involved is as follows.

1) *Convert image in to MSER regions*: Convert the image into regions by using MSER algorithm.

2) *Remove Non-text region based on their geometric property*: Based on the geometric property of the regions remove the unwanted/dissimilar region.

3) *Characters are bounded using bounding box*: Using bounding box property of the MSER regions, characters are bounded. The bounding boxes are represented by using rectangle.

4) *Expand the bounding box*: Increase sizes of the bounding box for represent full word/texts. The size is increased by taking length and breadth values.

5) *Text Detection*: The bounding box contains texts/words are detected by using OCR. The OCR read the entire bounding box and recognizes text content and shown it.

6) *Text is display in file and converts it to speech*: The detected text content in the bounding boxes is read and writes to a file. Then file content is read and convert it into speech.

7) *Text is sent to Google browser for searching/gather information*: Here also read the file content and it sent to a browser for collect the information about the text content.

For **Text Extraction-2** after we input the image, the process involved is explained below,

1) *Convert image in to MSER regions*: Convert the image into regions by using MSER algorithm.

2) *Extract each region and shown in figures*: The regions contain characters and they are represented by using bounding box property. The bounding boxes are shown in rectangle shape and read all bounding boxes and display as a figure.

3) *Compare the bounding box character with previously stored character dataset*: Compare the characters with previously stored character dataset. If any match is found then character is detected and store to a file. If character is not detected then take next bounding box character. This process is iterating until all bounding box characters are detected. The detected characters are form text/words.

4) *Text is display in file and converts to speech*: The detected text/words are read and display in a file and convert it into speech.

5) *Text is sent to Google browser for gather information*: Detected text is sent to a browser for collect information about text content.

5. IMPLEMENTATION

The project was implemented using MATLAB and was used for the detection of text from images. MATLAB stands for *Matrix Laboratory* it is a programing language. It was developed by MathWorks. MATLAB provide operations like matrix manipulations operation, plotting graphs and data, creating user interface, and interfacing with programs written in other languages.

The GUI required for project was done by using MATLAB GUI. GUIs stands for graphical user interfaces and it provide an easy way to run the program or application without any type commands. The GUI contains controls like menus, toolbars, buttons, and sliders [17].

The testing of the proposed method was carried out using different kinds of images having text embedded in it. The category of images used involves scene images with text, images with street sign boards and images with handwritten text. The images of these categories have been used by taking following data sets.

1) KAIST_Scene_Text_Database [18]

2) The_Street_View_Text_Dataset[19]

3) Handwritten Text Database

The scene text database is again subdivided into two,

1) Image taken by using mobile phone

2) Image taken by using digital camera

The image taken by using mobile phone is again subdivided based on their condition. The conditions are,

1). Indoor

2). Outdoor

3). Book Cover

The image taken by using digital camera is again subdivided based on their condition. The conditions are,

1). Indoor

2). Outdoor

3). Shadow

4). Light

The Street View database contains images like shop front, posters, sign boards, headlines etc… The Handwritten database contains upper case handwritten text images.

6. RESULTS

Using the implementation plans discussed in the previous section we run the testing of the data set and obtains a couple of results. A comparison of the results is represented as pie chart in each of the case. We classify the results as correct, partial and wrong results. A result is said to be correct if the proposed method is able to identify the text in the image with >90% accuracy. In the case of partial results, the text identified is >60% and <90%. If the number of characters identified from the image is <60% accurate then, the result is said to be wrong.

6.1 Mobile Camera Image - Outdoor Condition (MCI-OC)

We use the scene dataset here with images taken using mobile camera. The image is taken under an outdoor condition is considered. Here some results are shows partial and wrong output. The partial output is due to the difference in fonts and styles. For example, in some font '***ℓ***' is treated as '***u***'. The wrong output is due poor edge detection. The comparisons of accuracy of the results are given in fig 2. Table 1 gives a statistics of the images in the data set considered under this category.

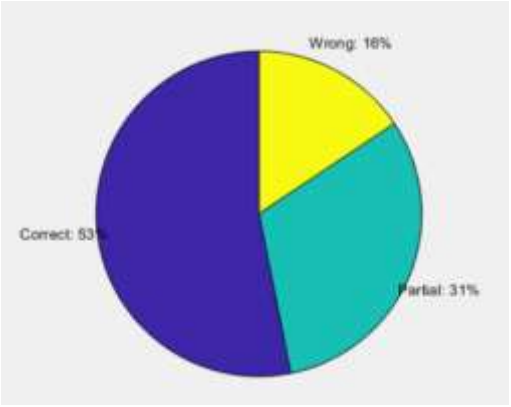

Fig- 2: Graph for Image from mobile camera under outdoor condition (Scene Text Database)

6.2 Mobile Camera Image - Indoor Condition (MCI-IC)

We use the indoor images shot by mobile camera in the scene dataset. Here we get 48% of images correctly identified. The partial output is due to the difference in font style and background of the image. Some results are shows wrong output, because the edge detection is not work properly. The comparisons of accuracy of the results are given in fig 3. A statistics of the images in the data set considered under this category is shown in table 1.

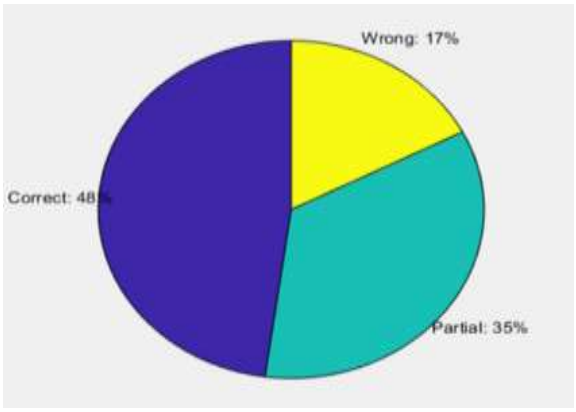

Fig - 3: Graph for Image from mobile camera under indoor condition (Scene Text Database)

6.3 Mobile Camera Image - Book Cover Condition (MCI-BC)

The book cover images captured by mobile camera in the scene data set are used here. Here some results are shows partial and wrong output. The partial output is due to the difference in fonts and styles. For example, in some font 'o's treated as '0'. Some results are shows wrong output, because the edge detection is not work properly. The comparisons of accuracy of the results are given in fig 4. Table 1 gives the number of images in the data set considered under this category.

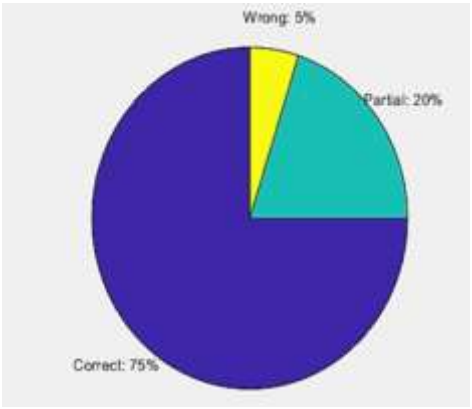

Fig - 4: Graph for Image from mobile camera under book cover condition (Scene Text Database)

6.4 Digital Camera Image - Shadow Condition (DCI-SC)

In this section we considered the images under shadow condition in the scene data set taken by digital camera. Here some results are shows partial and wrong output. The partial output is due to the edge detection. The edge detection is not work properly. Some results are shows wrong output due to the background. The comparisons of accuracy of the results are given in fig 5. A listing of the number of images in the data set for this category is presented in table 1.

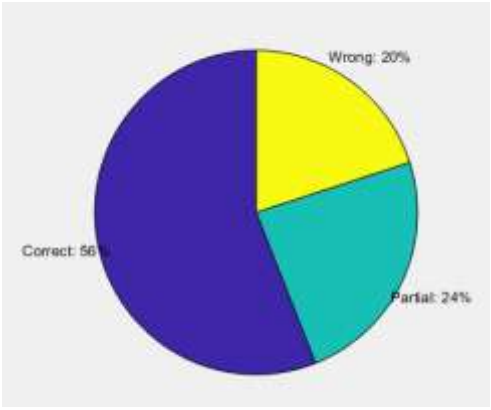

Fig - 5: Graph for Image from digital camera under shadow condition (Scene Text Database)

6.5 Digital Camera Image - Indoor Condition (DCI-IC)

In this section we also use the images taken by digital camera in indoor condition and are part of scene data set. Here some results are shows partial and wrong output. The partial output is due to the difference in font style and edge detection. Some results are shows wrong output, because of the background of the image. The comparisons of accuracy of the results are given in fig 6. The number of images in the data set considered under this category is presented in table 1.

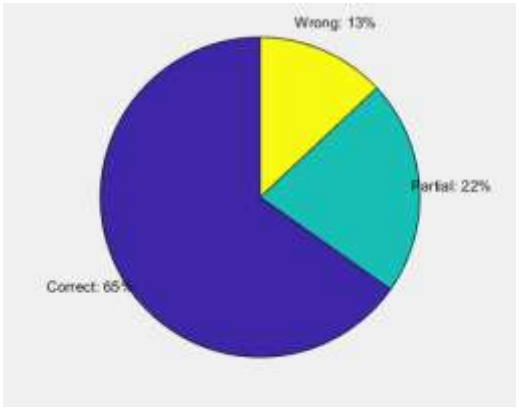

Fig - 6: Graph for Image from digital camera under indoor condition (Scene Text Database)

6.6 Digital Camera Image - Outdoor Condition (DCI-OC)

We use the images shot by digital camera under outdoor condition and are part of the scene data set. The partial output is due to the difference in font. Some results are shows wrong output, because of the improper edge detection. The comparisons of accuracy of the results are given in fig 7. The listing of images in the data set for this category is presented in table 1.

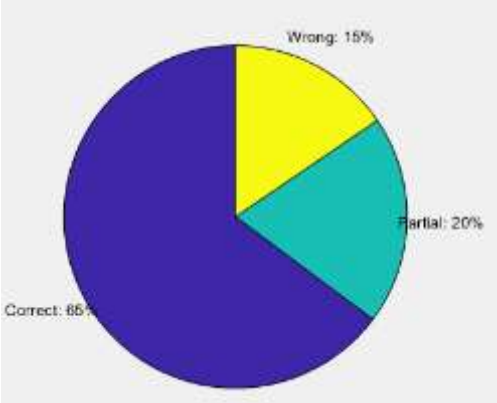

Fig - 7: Graph for Image from digital camera under outdoor condition (Scene Text Database)

6.7 Digital Camera Image - Light Condition (DCI-LC)

Set of digital camera images shot in the light condition which are members of scene data set is used in this section. The results are shows partial and wrong output. The partial output is due to the edge detection. Some results are shows wrong output, because of the image is taken not horizontally. The comparisons of accuracy of the results are given in fig 8. The statistics of images in the data set for this category is presented in table 1.

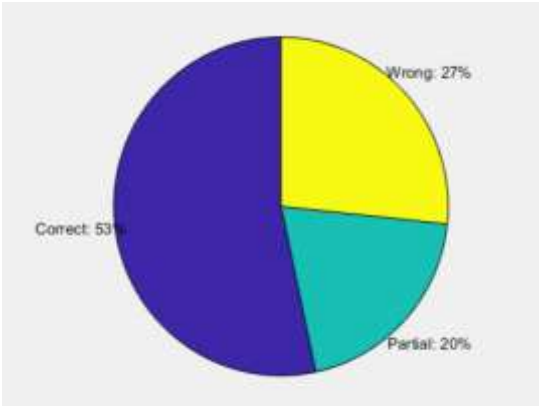

Fig - 8: Graph for Image from digital camera under light condition (Scene Text Database)

6.8 Street View Image (SVI)

The street view text dataset is used in this section. Here the street view images are considered. The street view images contain sign boards, posters, shop front, placards, mile stone etc… The results are shows partial output, because the image is taken from the street so, the quality of the text is less. The wrong output is due to the poor edge detection. The comparisons of accuracy of the results are given in fig 9. A number of images in the data set for this category is presented in table 1.

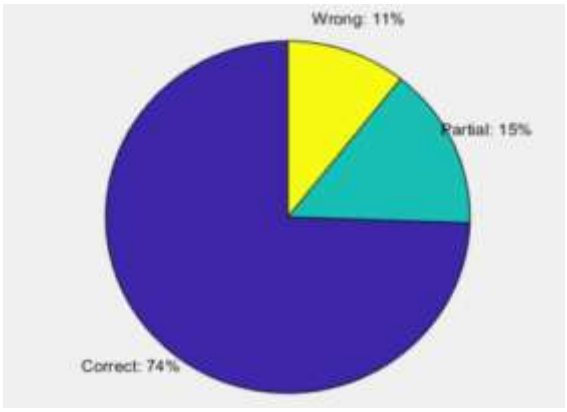

Fig - 9: Graph for Street View Text Database

6.9 Handwritten Image (HI)

Handwritten uppercase letters/characters are used in this section. The real time and existing handwritten characters are considered and checks the accuracy of getting correct output. Here the challenging factor is person writing style of each character. The partial output is due to mismatch in the person writing character and exiting alphabet database. For example sometimes 'c' is treated as 'e'. The wrong results due to worst region detection. The region detection is not proper. The comparisons of accuracy of the results are given in fig 10. A list of number of images in the data set for this category is presented in table 1.

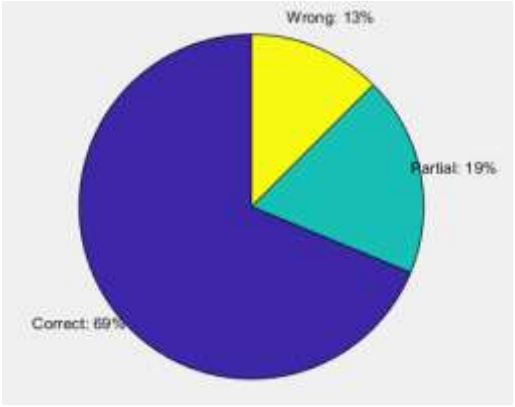

Fig - 10: Graph for Handwritten Text Database

Table -1: Count and results statistics of the categories of images used

Image Details	MCI-OC	MCI-IC	MCI-BC	DCI-SC	DCI-IC	DCI-OC	DCI-LC	SVI	HI
No. of Images	31	23	32	31	46	71	15	47	32
No. of correct output	17	11	24	17	60	46	8	35	22
No. of partial output	10	8	6	10	10	14	3	7	6
No. of wrong output	5	4	2	5	6	11	4	5	4

7. CONCLUSIONS

As technology progress day by day text line detection become an important task. Here we focus on detecting the text from image, convert to speech and also have the provision to gather information the text from internet. The system is implemented using MATLAB. The text regions are detecting using MSER region detector. The OCR classifier detects the text from MSER regions. Here we are using three types of datasets--Scene Database, Street Database and Handwritten Database. The results shows that in scene database we get more than 40% output under any conditions like if the image is click in a shadow, light, indoor or outdoor situation. In street database get 74% output as correct. In handwritten database we get 68% output as correct.

In future develop an OCR for other languages and apply some machine learning techniques that convert language sentences from one language to another.

REFERENCES

1. **H. I. Koo, D. H. Kim**, Scene text detection via connected component clustering and non-text filtering, IEEE transactions on image processing 22 (6) (2013) 2296–2305.

2. **X. Chen, A. L. Yuille**, Detecting and reading text in natural scenes, in: Computer Vision and Pattern Recognition, 2004, CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on Vol.2, IEEE, 2004, pp, II–II.

3. **M. S**, detecting text in natural scenes with connected components clustering and non-text filtering, in: International journal of Engg and technology (IRJET2016), IRJET, 2016, pp. 725–728

4. **X.-C. Yin, X. Yin, K. Huang, H.-W. Hao**, Robust text detection in natural scene images, IEEE transactions on pattern analysis and machine intelligence 36 (5) (2014) 970–983.

5. **X. Yin, X.-C. Yin, H.-W. Hao, K. Iqbal**, Effective text localization in natural scene images with mser, geometry-based grouping and adaboost, in: Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), IEEE, 2012, pp. 725–728

6. **H. I. Koo, N. I. Cho**, State estimation in a document image and its application in text block identification and text line extraction, in: European Conference on Computer Vision, Springer, 2010, pp. 421–434.

7. **H. I. Koo, N. I. Cho**, Text-line extraction in handwritten chinese documents based on an energy minimization framework, IEEE Transactions on Image Processing 21 (3) (2012) 1169–1175.

8. **Y.-F. Pan, X. Hou, C.-L. Liu, et al.**, A hybrid approach to detect and localize texts in natural scene images, IEEE Transactions on Image Processing 20 (3) (2011) 800.

9. **K. I. Kim, K. Jung, J. H. Kim**, Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (12) (2003) 1631–1639.

10. **K. Jung, K. I. Kim, A. K. Jain**, Text information extraction in images and video: a survey, Pattern recognition 37 (5) (2004) 977–997.

11. **L. G´omez, D. Karatzas**, Mser-based real-time text detection and tracking, in: Pattern Recognition (ICPR), 2014 22nd International Conference on, IEEE, 2014, pp. 3110–3115.

12. **S. Unar, A. Hussain, M. Shaikh, K. H. Memon, M. A. Ansari, Z. Memon**, A study on text detection and localization techniques for natural scene images, IJCSNS 18 (1) (2018) 100.

13. **C. Yao, X. Bai, W. Liu, Y. Ma, Z. Tu**, Detecting texts of arbitrary orientations in natural images, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 1083–109.

14. **M. S. Hossain, A. F. Alwan, M. Pervin**, Road sign text detection using contrast intensify maximally stable extremal regions, in: 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), IEEE, 2018, pp. 321–325.

15. **K. Sumi, M. Arun Kumar**, Detection, and recognition of road traffic signs-a survey, International Journal of Computer Application (0975-8887) 160 (3).

16. **Y. Tang, X. Wu**, Scene text detection and segmentation based on cascaded convolution neural networks, IEEE Transactions on Image Processing 26 (3) (2017) 1509–1520.

17. https://in.mathworks.com/discovery/matlab-gui.html

18. http://www.iaprtc11.org/mediawiki/index.php?title = KAIST_Scene_Text_Database

19. http://www.iaprtc11.org/mediawiki/index.php?title= The_Street_View_Text_Datase

BIOGRAPHIES

Safvana Sajeev, She is a post-graduation student in College of Engineering Kidangoor. Specialization in Computer and Information Science. She received the graduation in Computer Science and Engineering from Gurudeva Institute of Science and Technology. Her areas of interest are Image Processing and Databases.

Ojus Thomas Lee, is an Associate Professor in College of Engineering Kidangoor. His areas of interest are Databases, Cloud Computing, Sensor Networks and Image Processing.