

# A NOVEL METHOD OF EXTRACTING DATA IN STONE INSCRIPTION

Mrs.R.REENA ROY<sup>1</sup>, ARUN KUMAR.P<sup>2</sup>, HABIB DHULFIKHAR.H<sup>3</sup>, CAROLINE.J<sup>4</sup>

<sup>1</sup>(ASSISTANT PROF), Department of Information Technology, Easwari Engineering College, Chennai, India

<sup>2</sup>Easwari Engineering College, Department of Information Technology, Chennai, India

<sup>3</sup>Easwari Engineering College, Department of Information Technology, Chennai, India

<sup>4</sup>Easwari Engineering College Department of Information Technology, Chennai, India

**Abstract**—Documents of the ancient time gift several opposition for standalone gesture recognition systems, among them, the division and classification steps. Fastidiously gloss wordings square measure is required to coach a system. In some eventualities, written document square measure solely offered at the subdivision level. During this activity, we have a tendency to demonstrate the way to train the system with few tagged information. We have a tendency to additionally propose a model-based social control theme that considers the variability within the writing scale at the popularity section. We have a tendency to apply this approach to the publically offered browse dataset. Our system achieved the competitor result. Humans have distinctive handwriting designs that prove to be an obstacle for handwriting recognition algorithms. To date, multiple researches are done to acknowledge these totally different handwriting designs, most notable mistreatment the synthetic neural network (ANN) with back propagation algorithms that has additionally been verified to relinquish adequately high accuracies. By mistreatment real time method image capturing, this technique and algorithmic rule will be enforced to use multiple written entry information for faculties and universities, wherever the written information of a regular score sheet from totally different people will be transferred to a computer program.

**Keywords**—Image processing, Python, Matlab, Matplotlib ,Language Independent

## I. INTRODUCTION

The motive of Dravidian language (Tamil) letters examination and identification has continuously been a crucial branch of analysis for tamil leaders.

CNN mechanically performs text orientation correction on every scanned image, thus though your documents square measure placed sideways or the wrong way up, CNN can perpetually be able to browse and show them within the correct reading orientation. CNN has 3 layers. Every square measure connected by Neurons. The primary Layer is Input Layer, the center layer is Hidden Layer and also the Last Layer is Output Layer.

Ancient characters square measure totally different from gift text. Thus trendy peoples can't able to establish the information's in inscription.so we've planned techniques to extract the data in stones victimization image process.

Convolution Neural Network is performed domestically on your system, creating the method quicker and additional convenient. This conjointly makes it potential for you to browse documents in things wherever you've got a network affiliation.

This method cannot provide 100 percent economical as a result of most of the written characters could disagree from each other written characters thus we've to see millions of characters. It's inconceivable to see millions of characters thus it'll provide Associate in nursing prediction of script and Tamil characters.

## II. LITERATURE SURVEY

Ling-Qun Zuo [2] extracts each character using his proposed system. The extracted character is processed using Bi-LSTM algorithm and then encoding and decoding is performed to identify each character.

Xiaoqing Ding [1] proposes a system where a class is defined for every individual character. The script is scanned and converts into discrete values. Gaussian function is used to generate Eigen values, based on which co variance is calculated for each character.

Rohan Katoch [3] proposes a system based on character reconstruction. The character is captured as an image and then the pixel path is traced. The pixel spectral analysis is done to reconstruct the character.

Jun Chen [4] has developed a Captcha recognition system that is text based. The system uses a DCNN to perform classification. Confusion matrix is generated using the DCNN algorithm, the result is compared with a template.

This paper depicts [9] a survey analysis targeted on Indian epitaph letter identification by grasping the languages within the pictures.

This paper [13] has conducted an enquiry on Natural gradient-based versatile freelance part Analysis supported digitisation of Ancient epitaph pictures. It also addresses the issues present throughout digitisation and preservation of epitaph. It also has a technique to reinforce the least distinction between words and also the surroundings of the epitaph image.

The authors [15] have done analysis to develop OCR pipeline for written type areas. This analysis framework was done below the 2 main phases. First is concentrated on planning written text to represent gesture-image knowledge specifically for that process.

## III. Related works

### A. Overview

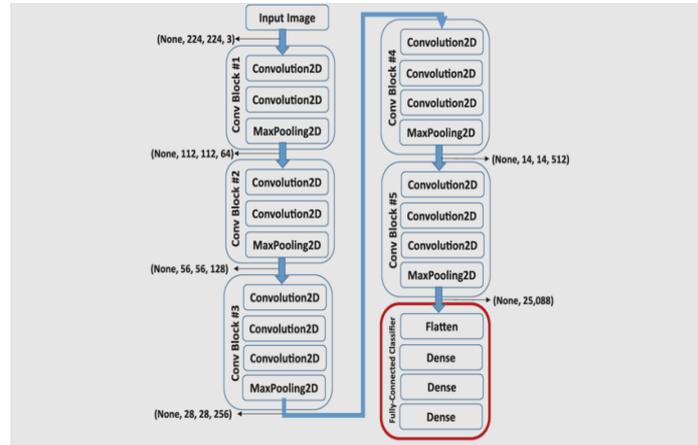
Systems were trained with some depicted word written data. An increase instructing workflow was used with only 0.01 of manually depicted word written data. This work showed how to interlink python and Matlab by Correlation. Generally in matlab there are certain certain algorithms to correlate the strength of a linear relationship between two variables whereas in python an .exe file is called to correlate with the matlab. A work-based scheme was used in the put down balance to the test data.

### B. Word corpus

The info comprises 154 Dravidian letters with one hundred tests in every letter. Every letter consisting of one hundred tests is saved in several directories, therefore a group of 154 directories ar created. Every representation is of various layouts, since we have a tendency to use the VGG sixteen rule the pictures ar resized into 224x224. To get purposeful options throughout the coaching method in logical type ar reborn into images in unsigned number type. This can be implemented out in MATLAB.

c. VGG16

VGG (Visual Geometry Group) is intricacy network that's instructed on over 1,000,000 pictures from the representation web database. The layout is sixteen groups and may split into a thousand entity classes.



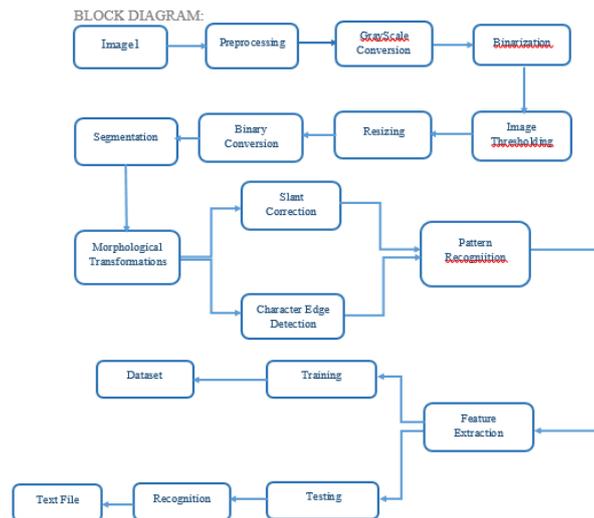
**Fig 1 VGG16 Layers**

The network has a picture size of 194X194. The representation on top of provides the main points regarding the thirteen intricacy groups. These 2 groups are reassigned by the several groups for the various standards. Because the range of groups will increase the amount of netting will increase hence a lot of entity are often removed. Because the range of groups will increase the diameter is reduced.

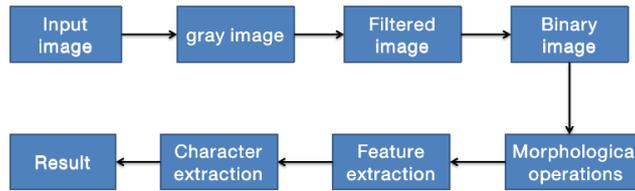
IV. EXPERIMENTAL ANALYSIS

A. System Design

Given below diagram depicts the flow of image recognition using CNN.



**Fig 2 Proposed Model for Character Recognition**

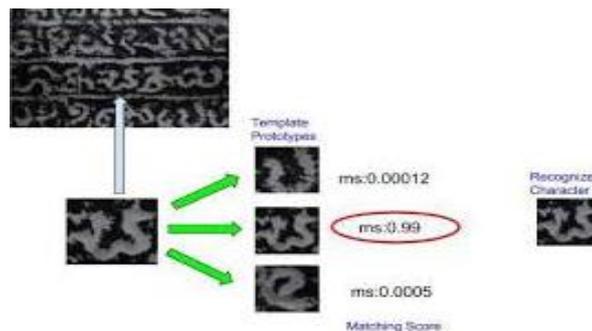


**Fig 3 Experimental model for Character Extraction**

**Workflow:**

- The input image analyzed and filtered then it is converted to gray scale image format.
- Then the image is filtered and is now sampled to convert the analog signal to Binary image format.
- The Binary image is operated with morphological operations for structuring the image.
- From the structured image the features required for further operations are extracted.
- The characters are then recognized from the extracted data which is the required output.

**B. Model**



**Fig 4 Matching Score**

Step1: Take the image manuscripts data set that needs to be converted into characters are stored in database.

Step2: The stored image are then processed by filtering the noise in the data and then prepared for sampling.

Step3: The pre processed images are then fed to morphological transformations for segmentation and individual characters are recognized from the images scanned.

Step4: The recognized characters are checked for correctness of the identified values.

- i) Accuracy
- ii) Precision
- iii) Recall

**C. Modules Description**

**A. Pre-Processing**

This is sweetening function to filter noise from input representation. The input representation is of color type, the input image is pre-processed victimization median filtering technique for noise removal as a result of, it removes noise whereas conserving edges.

## 1. IMAGE THRESHOLDING:

If component price is larger than a limen price, it is appointed one price (may be white), else it's appointed another price (may be black). The operate used is cv2.limen. 1st argument is that the supply representation, that ought to be a grayscale representation. Second argument is that the limen price that is employed to differ the component points. Third argument is that the maximumVal that represents val to tend if component price is quite (sometimes less than) the brink val. OpenCV provides totally different kinds of thresholding and it's determined by the fourth parameter of the operate. Differing kinds are:

cv2.THRESH\_BINARY

- 1) cv2.THRESH\_BINARY\_INV
- 2) cv2.THRESH\_TRUNC
- 3) cv2.THRESH\_TOZERO
- 4) cv2.THRESH\_TOZERO\_INV

Code :

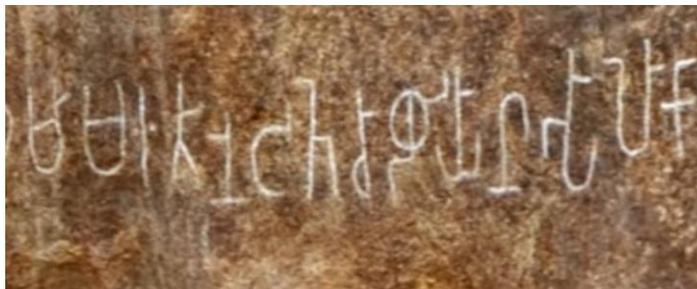
```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('gradient.png',0)
ret,thresh1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(img,127,255,cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(img,127,255,cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO)
ret,thresh5 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO_INV)

titles = ['Original Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]

for i in xrange(6):
    plt.subplot(2,3,i+1),plt.imshow(images[i],'gray')
    plt.title(titles[i])
    plt.xticks([],plt.yticks([]))

plt.show()
```

**Result is given below:**  
**ORIGINAL IMAGE:**



**Fig 5 Stone Characters**

**BINARY:**



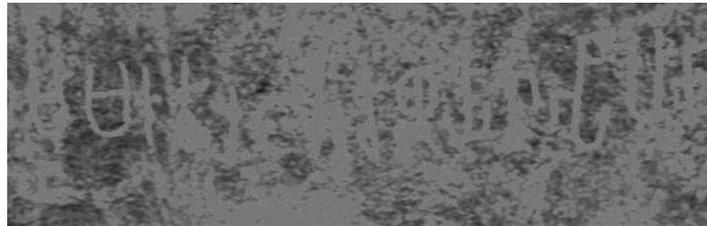
**Fig 5.1 Binary Conversion**

**BINARY\_INV:**



**Fig 5.2 Binary Inversion**

**TRUNC:**



**Fig 5.3 Truncate Conversion**

**TOZERO:**



**Fig 5.4 Binary ToZero Conversion**

**TOZERO\_INV:**



**Fig 5.5 Binary ToZero Inverison**

**B. Segmentation:**

Segmentation victimization the bar chart leveling works great for supporting and up precision. The pre-processed representation is split into different areas with relevancy pixelates. The input representation is on par with the reference representation and pixelate area unit metameric adore reference representation. That is,  $n*n$  pixelates specified each the representation have same variety of pixelates, on par between the

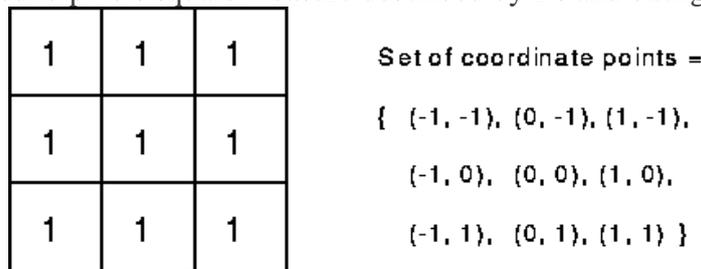
pictures is formed for value removal through picture element by picture element for shrewd the distinction between the representation. Bar chart leveling reads the images from the input data, computes the bar chart, calculate the normalized add, remodel the input image into the output image. Bar chart will offer a visible show of enormous quantity of information that area unit troublesome to know. The goal of image segmentation is to alter the illustration of a picture into one thing that's a lot of purposeful and easier to investigate.

**MOPHOLOGICAL TRANSFORMATIONS:**

Morphological transformations ar some straightforward operations supported the image form. It's ordinarily performed on binary pictures. It wants 2 details, one is our original representation and other is named component that decides the character of execution.

**1. Erosion:**

The mathematical explanation for grayscale erosion is identical except within the approach during which the set of groupings related to the input representation springs. Additionally, these coordinates square measure 3D instead of 2-D. As associate example of binary erosion, suppose that the structuring part may be a 3x3 sq., with the origin at its center as shown in Figure one. Note that during this and resultant diagrams, foreground pixels square measure described by 1's and background pixels by 0's.



**Fig.6 3x3 square structuring element**

To cipher the erosion of a binary input image by this structuring part, we tend to think about every of the foreground pixelates within the input representation successively. For every pixelate we tend to set the part on prime of the input representation so the start of the element concedes with the input pixelates. For each pixelate within the object element, the next component within the representation beneath may be a front component, then the initial component is left because it is.

Code:

```
import cv2
import numpy as np
img = cv2.imread('j.png',0)
kernel = np.ones((5,5),np.uint8)
erosion = cv2.erode(img,kernel,iterations = 1)
```



**Fig 7 Erosion Transformation**

**2. Dilation:**

This is simply negate of abrasion. Here, a representational element is '1'. Therefore it will increase the plain area within the representation or size of frontend entity will increase. Because, erosion wipes out plain hearings, however it conjointly shorts our entity. Therefore we tend to remove it.  
 dilation = cv2.dilate(img,kernel,iterations = 1)



Fig 7.1 Dilation Transformation

**3. Opening:**

Opening is simply another name of abrasion followed by dilation. It's helpful in removing noise, as we tend to explained higher than. Here we tend to use the perform, cv2.morphologyEx()  
 opening = cv2.morphologyEx(img, cv2.MORPH\_OPEN, kernel)

**4. Morphological Gradient:**

This differs between dilation and erosion of representation. The result is shown in the representation.

gradient=cv2.morphologyEx(img,cv2.MORPH\_GRADIENT, kernel)

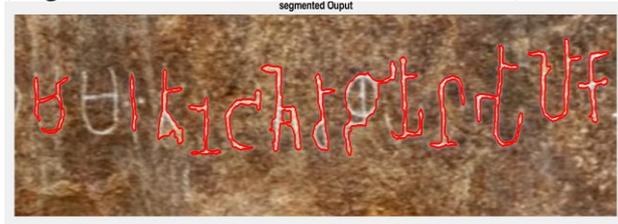


Fig 8 Segmented Output

**STRUCTURING ELEMENT:**

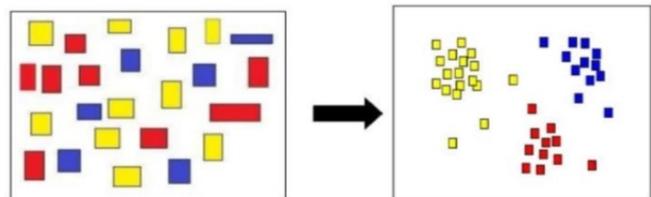
We manually created a structuring parts within the previous examples with facilitate of Numpy. it's rectangular form. however in some cases, you will want elliptical/circular formed kernels. thus for this purpose, OpenCV includes a operate, cv2.getStructuringElement(). you only pass the form and size of the kernel, you get the required kernel.

# Rectangular Kernel

```
>>> cv2.getStructuringElement(cv2.MORPH_RECT,(5,5))
array([[1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1]], dtype=uint8)
```

# Elliptical Kernel

```
>>> cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(5,5))
array([[0, 0, 1, 0, 0],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [0, 0, 1, 0, 0]], dtype=uint8)
```



```
# Cross-shaped Kernel  
>>> cv2.getStructuringElement(cv2.MORPH_CROSS,(5,5))  
array([[0, 0, 1, 0, 0],  
       [0, 0, 1, 0, 0],  
       [1, 1, 1, 1, 1],  
       [0, 0, 1, 0, 0],  
       [0, 0, 1, 0, 0]], dtype=uint8)
```

### C. Multi-Scale Template Matching:

Template Matching could be a translation invariant. The extension we have a tendency to area unit proposing currently will facilitate to create it additional strong to changes in scaling. however in model Matching if you're making an attempt to match the turned objects or objects that exhibit non-affine transformations. In this type of transformations we can able to match the keypoint matching. From this Template Matching we can easily recognize the characters by comparing two images. So if we set the Accuracy level to 70%, it will detect better than the accuracy level upto 90%.

Steps to match Templates:

- Take the Input Image and convert it to gray scale image.
- Take the Converted Gray Scale Input Image as Template
- The Template Slides over the Actual Image and find the relatable image in the given dataset and find the location where accuracy level matches.
- When the result is greater than the accuracy level it will be marked as an output images

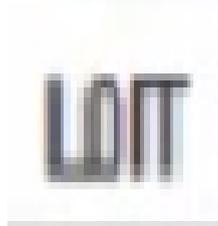
### Fig 9 Process of Template Matching

**Input Image:**



**Fig 10 Input Image**

**Matched Output Image:**



**Fig 10.1 Matched Character**

### D. Feature Extraction

Characteristic removal is one in the entire foremost necessary procedure during this process. Characteristic removal techniques area unit applied to induce options which will be helpful in classifying and recognition of pictures. Feature outlines the bearing of the representation, potency in differing and clearly in time utilization conjointly. Characteristic could be a piece of data extracted from the image. GLCM feature is employed to convert RGB image to grey scale representation to enhance the potency. Background removal process is undertaken in GLCM representation for background removal and eventually to seek out the seed and rate of growth of tumour region growing technique is employed.

**Region growing technique:**



𑀀	𑀁	𑀂	𑀃	𑀄	𑀅	𑀆	𑀇	𑀈	𑀉
a	ā	i	ī	u	ū	ṛ	ṛī	ḷ	ḷī
[ə]	[a:]	[i]	[i:]	[u]	[u:]	[r]	[r:]	[l]	[l:]
𑀀	𑀁	𑀂	𑀃	𑀄	𑀅	𑀆	𑀇	𑀈	𑀉
ka	kā	ki	kī	ku	kū	kr	kṛī	kl	klī
𑀀	𑀁	𑀂	𑀃	𑀄	𑀅	𑀆	𑀇	𑀈	𑀉
e	ai	o	au	añ	aṇ	aḥ			
[e/ɛ]	[əy]	[o/ɔ]	[a.u]	[aŋ]	[ə]	[əh]			
𑀀	𑀁	𑀂	𑀃	𑀄	𑀅	𑀆	𑀇	𑀈	𑀉
ke	kai	ko	kau	kañ	kaṇ	kaḥ	k		

**Fig 12 Brahmi Vowels Literature**

**Consonants:**

𑀀	𑀁	𑀂	𑀃	𑀄	𑀅	𑀆	𑀇	𑀈	𑀉
ka	kha	ga	gha	ña	ca	cha	ja	jha	ña
[kə]	[kʰə]	[gə]	[gʰə]	[ɲə]	[cə]	[cʰə]	[jə]	[jʰə]	[ɲə]
𑀀	𑀁	𑀂	𑀃	𑀄	𑀅	𑀆	𑀇	𑀈	𑀉
ta	tha	ḍa	ḍha	ṇa	ta	tha	da	dha	na
[tə]	[tʰə]	[ḍə]	[ḍʰə]	[ɲə]	[tə]	[tʰə]	[də]	[dʰə]	[nə]
𑀀	𑀁	𑀂	𑀃	𑀄	𑀅	𑀆	𑀇	𑀈	𑀉
pa	pha	ba	bha	ma					
[pə]	[pʰə]	[bə]	[bʰə]	[mə]					
𑀀	𑀁	𑀂	𑀃	𑀄	𑀅	𑀆	𑀇	𑀈	𑀉
ya	ra	la	va	śa	ṣa	sa	ha	ḷa	
[jə]	[rə]	[lə]	[və]	[ʃə]	[ʂə]	[sə]	[ɦə]	[l̪ə]	

**Fig 12.1 Brahmi Consonants Literature**

**Numerals:**

𑀀	𑀁	𑀂	𑀃	𑀄	𑀅	𑀆	𑀇	𑀈	𑀉
0	1	2	3	4	5	6	7	8	9

**Fig 12.2 Brahmi Numerals**

**Tamil Characters:**



**REFERENCES:**

- [1] Halwform Tekle Weldegebriel., Han Liu., Anwar ul Haq., Emmanuel Bugingo., Defu Zhang “A New Hybrid Convolutional Neural Network and Extreme Gradient Boosting Classifier for Recognizing Handwritten Ethiopian Characters”.
- [2] Wanwei Wang ; Jun Yang ; Min Chen ; Peng Wang “A Light CNN for End-to-End Car License Plates Detection and Recognition” Year: 2019 | Volume: 7 | Journal Article | Publisher: IEEE
- [3] Saad Bin Ahmed ; Ibrahim A. Hameed ; Saeeda Naz ; Muhammad Imran Razzak ; Rubiyah Yusof, “Evaluation of Handwritten Urdu Text by Integration of MNIST Dataset Learning Experience” Year: 2019 | Volume: 7 | Journal Article | Publisher: IEEE
- [4] Yanliang Jin ; Jinfei Xie ; Weisi Guo ; Can Luo ; Dijia Wu ; Rui Wang LSTM-CRF Neural Network With Gated Self Attention for Chinese NER, Year: 2019 | Volume: 7 | Journal Article | Publisher: IEEE
- [5] Syed Yasser Arafat ; Muhammad Iqbal, Two Stream Deep Neural Network for Sequence-Based Urdu Ligature Recognition, Year: 2019 | Volume: 7 | Journal Article | Publisher: IEEE
- [6] Liang Xu ; Yuxi Wang ; Xiuxi Li ; Ming Pan, Recognition of Handwritten Chinese Characters Based on Concept Learning, Year: 2019 | Volume: 7 | Journal Article | Publisher: IEEE
- [7] Ling-Qun Zuo ; Hong-Mei Sun ; Qi-Chao Mao ; Rong Qi ; Rui-Sheng Jia, Natural Scene Text Recognition Based on Encoder-Decoder Framework, Year: 2019 | Volume: 7 | Journal Article | Publisher: IEEE
- [8] Yue Yin ; Wei Zhang ; Sheng Hong ; Jie Yang ; Jian Xiong ; Guan Gui, Deep Learning-Aided OCR Techniques for Chinese Uppercase Characters in the Application of Internet of Things, Year: 2019 | Volume: 7 | Journal Article | Publisher: IEEE
- [9] Jiahui Qiu ; Yangming Zhou ; Qi Wang ; Tong Ruan ; Ju Gao, Chinese Clinical Named Entity Recognition Using Residual Dilated Convolutional Neural Network With Conditional Random Field, Year: 2019 | Volume: 18, Issue: 3 | Journal Article | Publisher: IEEE
- [10] Huijun Gao ; Ming Yi ; Jinyong Yu ; Junbao Li ; Xinghu Yu, Character Segmentation-Based Coarse-Fine Approach for Automobile Dashboard Detection, Year: 2019 | Volume: 15, Issue: 10 | Journal Article | Publisher: IEEE
- [11] Jun Chen ; Xiangyang Luo ; Yingying Liu ; Jinwei Wang ; Yuanyuan Ma, Selective Learning Confusion Class for Text-Based CAPTCHA Recognition, Year: 2019 | Volume: 7 | Journal Article | Publisher: IEEE
- [12] S. K. S. Raja, R. Rishi, E. Sundaresan and V. Srijit, "Demand based crop recommender system for farmers," 2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR), Chennai, 2017, pp. 194-199, doi: 10.1109/TIAR.2017.8273714.
- [13] V. Nair, and G.E. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines, Machine Learning -International Workshop Then Conference, 2010, pp. 807-814.
- [14] S. Acharya, A.K. Pant, and P.K. Gyawali, Deep learning based large scale handwritten Devanagari character recognition, 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 2015, pp. 1-6.
- [15] J. Bai, Z. Chen, B. Feng, and B. Xu, Image character recognition using deep convolutional neural network learned from different languages, 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 2560-2564.