A Novel OCR System based on Rough Set Semi-Reduct

Ushasi Chaudhuri[1] **Partha Bhowmick**[2] **Jayanta Mukherjee**[2]

[1]Advanced Technology and Development Center
IIT Kharagpur

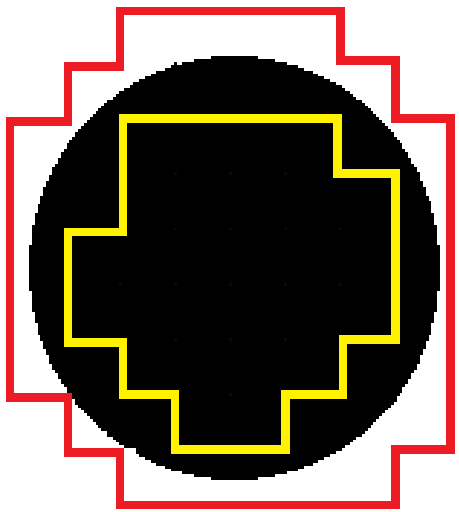
[2]Department of Computer Science and Engineering
IIT Kharagpur

7th International Conference on Pattern Recognition and Machine Intelligence, 2017

Outlines

Introduction : Rough Set

- Rough set is an approximation of a crisp set in terms of a pair of sets for lower and upper approximations of the original set.
- It is easier to handle each of the approximating sets compared to the original crisp set.



Prior Work: Rough-Set

- Rough set — set the basic fundamentals Pawlak[1]
- While many problems has been tackled in rough set[2][3], its application in image processing has been explored less.
- Czajewski[4] use a discernibility matrix built on a reduced database for classification algorithms; average accuracy of 66.6%.

[1] Pawlak & Zdzislaw,"Rough sets", IJCIS-1982

[2] Thangavel,"Dimensionality reduction based on rough set theory", ASC-2009

[3] Thangavel,"Application of rough set theory to feature selection for unsupervised clustering", Chemometrics and Intelligent Laboratory Systems-2002

[4] Czajewski,"Rough Sets in Optical Character Recognition", RSCTC-1998

Objective

- Characters scripted using atypical and complex font styles: datasets huge in volume and diversity. Hence, many supervised classification models require immense time and tenacity for training, resulting in slowdown.
- Google Tesseract[5] uses various geometric features; tedious training process to improve the efficiency; also reduces the speed of the OCR engine.

[5] "http://tesseract-ocr.repairfaq.org/"

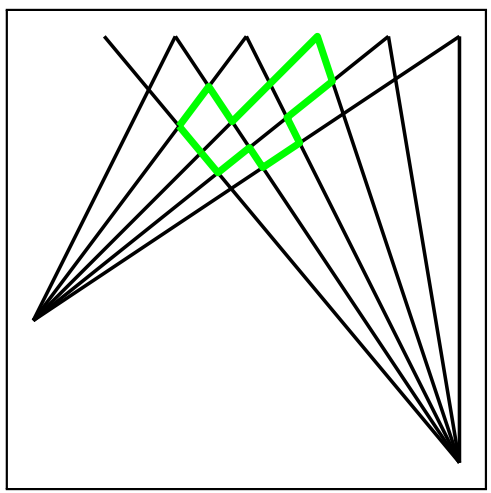
Motivation— Unsupervised OCR System

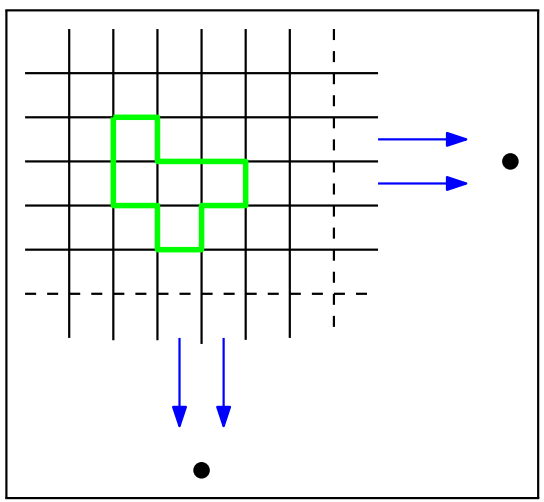
- Very difficult to develop a font independent supervised system.
- Requires re-training to incorporate newer fonts for supervised learning.
- Supervised system is computation and memory intensive.
- However, supervised system is potentially more accurate.
- We attempt to develop an efficient and accurate unsupervised OCR system.

Definition

Definition (Rough-set Polygon)

A *Rough-set polygon* is a polygon which is constructed by two parametric families of straight lines with their centers at two points. Rectilinear polygons are a type of isothetic polygons with their centers lying at infinity.





For computational advantage, we consider only axes parallel polygons.

Object Approximation

Definition (Rough-set Semi-Reduct)

A *Rough-set semi-reduct* is the sub-optimal set of attributes, that are required for any classification system, without having any discrepancies.

Definition (Tight Upper Approximation)

A finite and minimal union of elementary sets, also called a composed set, containing X is called the *best tight upper approximation* of X in $\mathbb{A}$, denoted by $\overline{Apr}_{\mathbb{A}}(X)$.

Definition (Tight Lower Approximation)

The greatest composed set in $\mathbb{A}$ contained in X is called the *best tight lower approximation* of X in $\mathbb{A}$, denoted by $\underline{Apr}_{\mathbb{A}}(X)$.

Object Approximation

Upper and Lower boundaries for an object X.

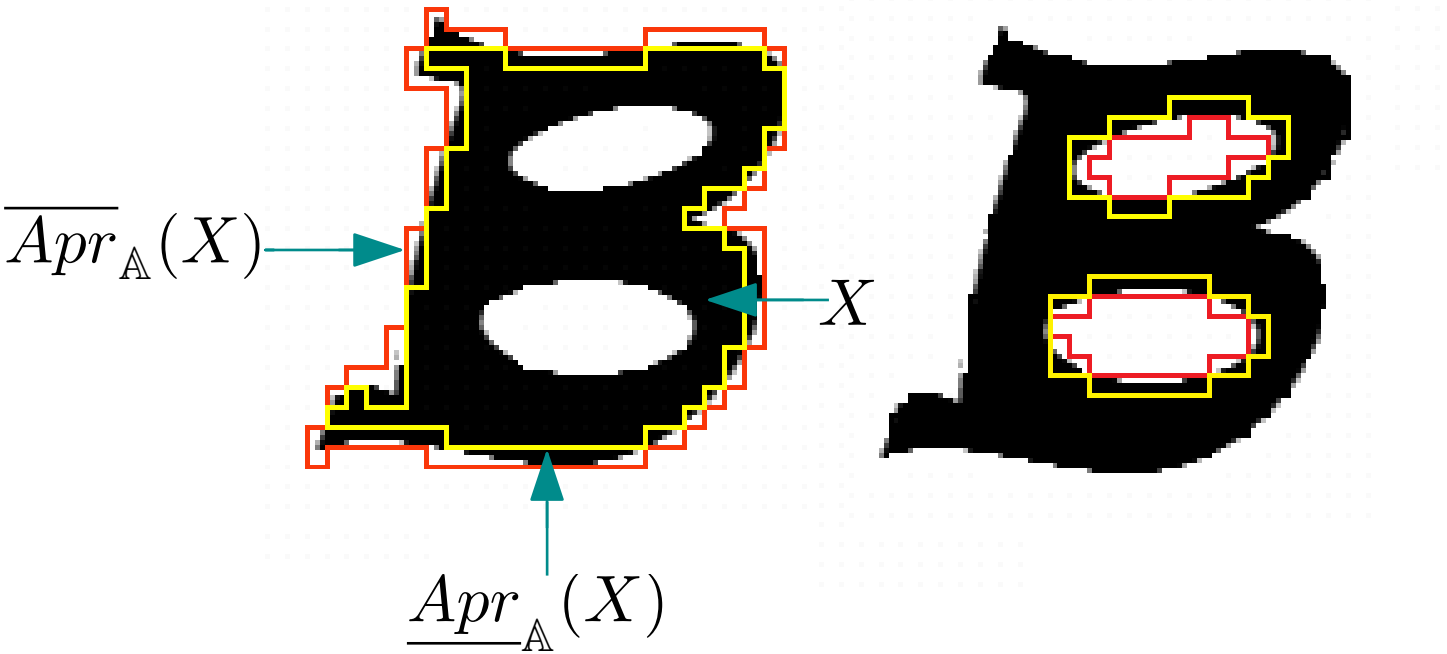


Figure: Boundaries of object X

Tight Upper and Lower Approximations

- S is a 2D digital object and $\mathbb{G}$ a cellular grid.
- *Tight upper approximation* is given by $\overline{\mathcal{P}}_{\mathbb{G}}(S)$, and *tight lower approximation* is $\underline{\mathcal{P}}_{\mathbb{G}}(S)$ of S (induced by $\mathbb{G}$) where $\underline{\mathcal{P}}_{\mathbb{G}}(S)$ and $\underline{\mathcal{P}}_{\mathbb{G}}(S)$ can have multiple polygons.
- *Accuracy*, $\alpha_{\mathbb{G}}(S)$ of S is given by

$$\alpha_{\mathbb{G}}(S) = \frac{\text{area}\left(\underline{\mathcal{P}}_{\mathbb{G}}(\text{S})\right)}{\text{area}\left(\overline{\mathcal{P}}_{\mathbb{G}}(\text{S})\right)} \tag{1}$$

Rought-Set: Dependance on cell size

We use rough-sets in two stages[6]—

- For construction of upper and lower approximations of a 2D digital object.
- Defining the approximations of their attributes comprising the reduct.

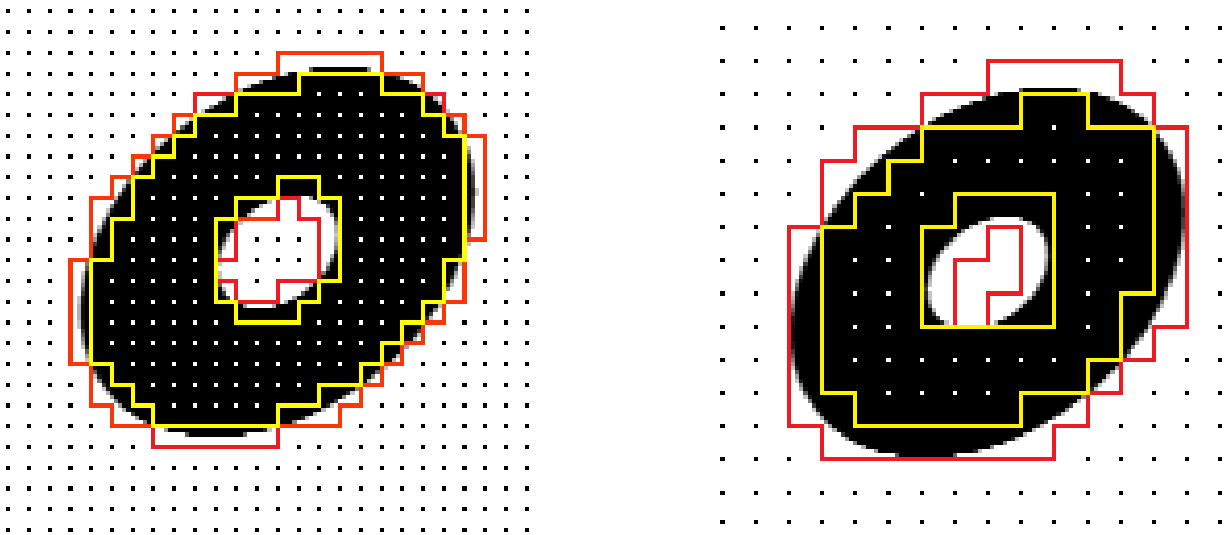


Figure: Cell-size 6×6 and 9×9.

[6] Pawlak & Zdzislaw, "Rough sets", IJCIS-1982

Approximate Euler Number

Definition (Hole polygon)

Some of the characters cover the entire region inside the cover, while some characters have voids inside them. These void covers are called as the lower boundary or a hole polygon.

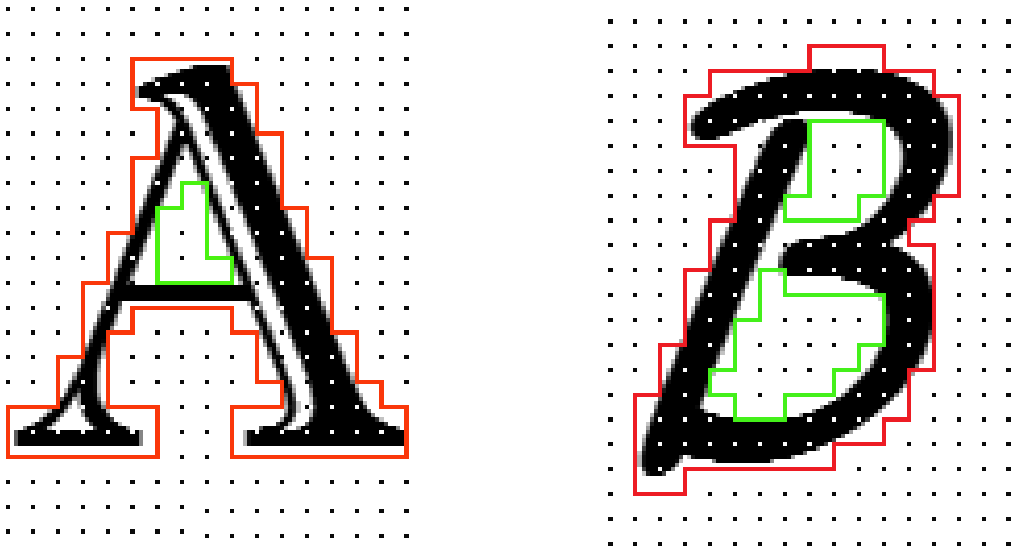
The Euler number(EN) is calculated as—

$$EN = 2 - n \tag{2}$$

where n = total number of polygons in $\overline{\mathcal{P}}_{\mathbb{G}}(S)$.

Characters like `A` or `D` have Euler number 0, while `E`, `F`, etc., have Euler number 1. This is used as the primary attribute in our classification system.

Approximate Euler Number



A — $n = 2$; EN $= 0$

B — $n = 3$; EN $= -1$

Since we are using rough set lower boundaries, we are able to get 2 holes despite having internal connectivities.

Relative Hole Positions

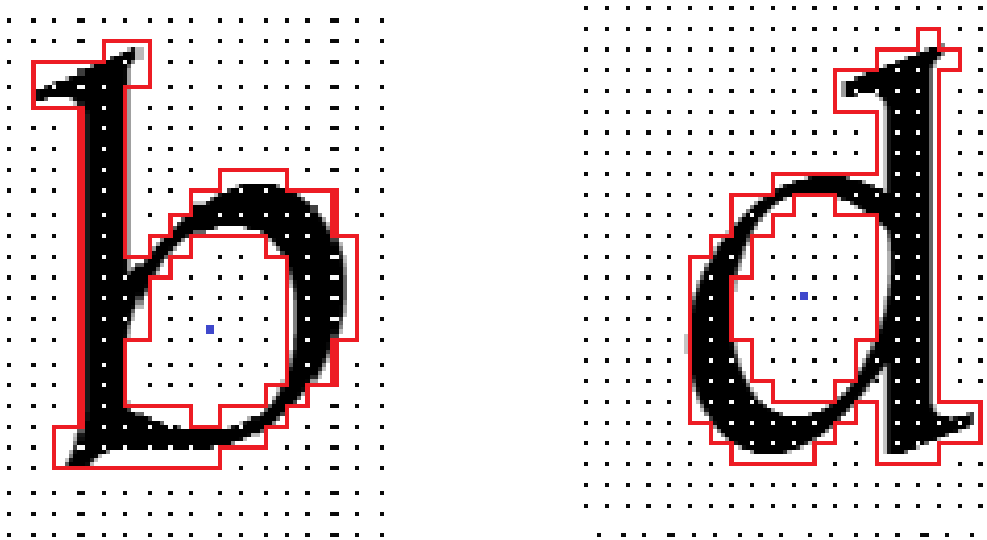
Definition (Hole position)

A hole position is given by the centroid of the hole w.r.t to the centroid of the containing polygon.

The hole centroid (c), is found in the local coordinate with the top left vertex of the character as the reference point.

The relative position (PoH) of each hole polygon is determined by comparing its center c with the top-left vertex v_0 of the outer polygon in $\overline{\mathcal{P}}_{\mathbb{G}}(S)$.

Relative Hole Positions



‘−’ and ‘+’ to denote left and right lateral halves
‘1’ and ‘2’ for upper and lower halves;
b has PoH $= +2$ and d has PoH $= -2$.

	EN	PoH	VDC	Concavity	ER
b	0	$+2$	8	-	2
d	0	-2	8	-	2

Edge Ratio

Definition (Horizontal Perimeter Component (HPC))

For each polygon in $\overline{\mathcal{P}}_{\mathbb{G}}(S)$, we define horizontal perimeter component (HPC) as the sum of the lengths of its horizontal edges.

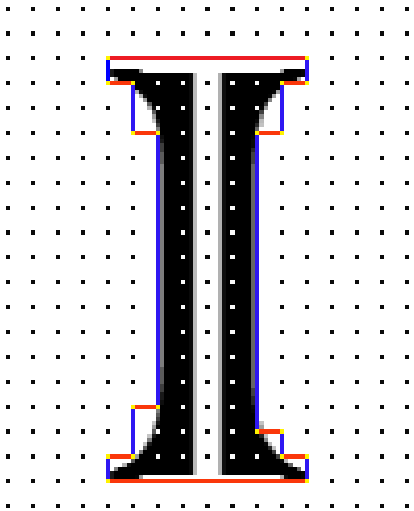
Definition (Vertical Perimeter Component (VPC))

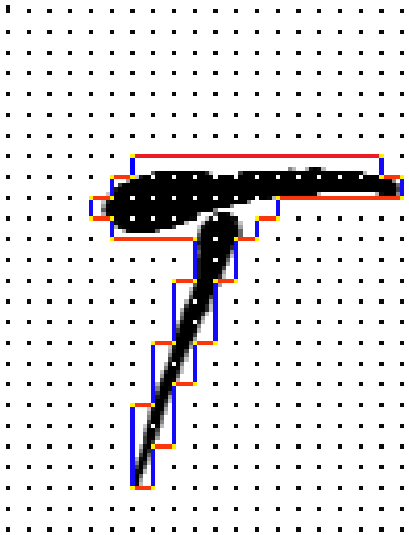
For each polygon in $\overline{\mathcal{P}}_{\mathbb{G}}(S)$, we define vertical perimeter component (VPC) as the sum of the lengths of its vertical edges.

Definition (Edge Ratio (ER))

The ratio VPC:HPC, discretized to the nearest value in $\{\frac{1}{2}, 1, 2\}$, is called edge ratio (ER).

Edge Ratio





By ER conventions, we get an ER of 2 and 1 for `I` and `T`

	EN	PoH	VDC	Concavity	ER
`I`	+1	-	2	-	2
`T`	+1	-	2	-	1

Directional Changes

Definition (U-turn)

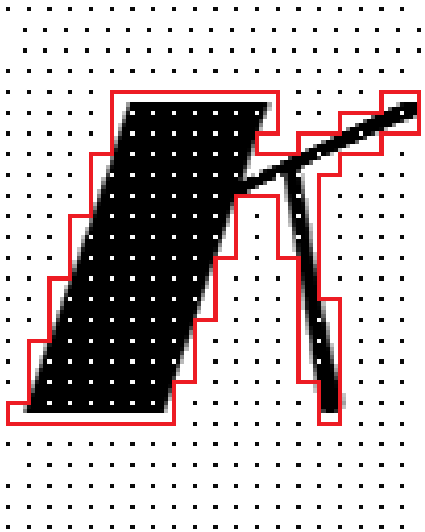
U-turn is defined by a vertex sequence where two consecutive vertices are of type $\langle +, + \rangle$; and for each such U-turn.

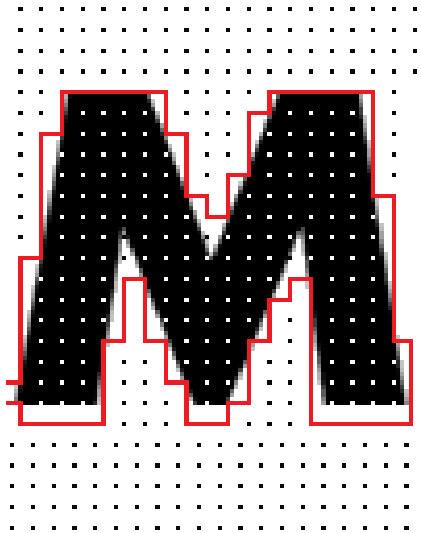
Definition (Vertical Direction Change (VDC))

When we start traversing along the upper boundary of the rough set perimeter of the object from the top left corner, the number of times we encounter a directional 'U-turns' in the vertical direction (ex 1 to 3, or 3 to 1) is defined as vertical direction change (VDC).

Similarly we have Horizontal direction change (HDC).
e.g. `I` has VDC 2; `A` has VDC 4.

Directional Changes





VDC(K)=6 and VDC(M)=8 The two challenges of using VDC —

- Variations due to different fonts — solved using error function.
- For characters with same VDC, like (M/W), we also use their VDC position informations.

Concavity Features

Definition (Concavity)

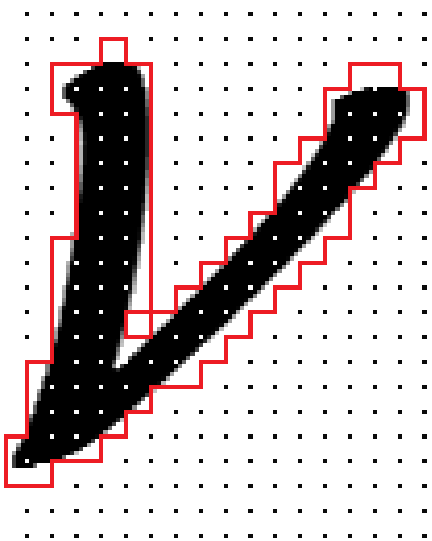
Concavity is defined as two consecutive vertices of type $\langle -, - \rangle$ (without considering the type intermediate 0 vertices).

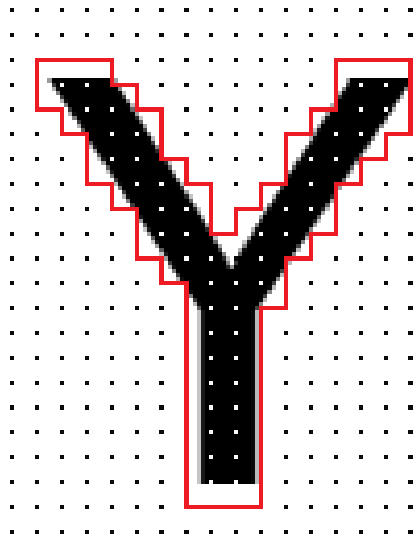
If more than two type -1 vertices occur one after another, then we get a nested concavity. Types of concavities—

- left (L)
- right (R)
- upward (U)
- downward (B)

Concavity Features

U, V and Y, having similar cover, we use *depth* information.
We discretize the relative depth in 1, 2, 3.





⟨concavity direction, region, depth⟩
V=(U,+1, 2) and Y=(U,+1, 1)

	EN	PoH	VDC	Concavity	ER
V	+1	-	8	(U,+1, 2)	2
Y	+1	-	8	(U,+1, 1)	2

Partial Illustration of Attributes

	EN	PoH	VDC	Concavity	ER
B	-1	$+1, +2$	2	(L,$+1, -$)	2
E	$+1$	-	2	(L,$+1, -$),(L,$+2, -$)	$\frac{1}{2}$
I	$+1$	-	2	-	2
M	$+1$	-	8	(D,$-2, -$),(D,$+2, -$),(U,$+1, -$)	1
T	$+1$	-	2	-	1
V	$+1$	-	8	(U,$+1, 2$)	2
Y	$+1$	-	8	(U,$+1, 1$)	2
b	0	$+2$	8	-	2
d	0	-2	8	-	2
3	$+1$	-	8	(R,$+1, -$),(R,$+2, -$)	1

Table: Information table (shown partial)

Illustration of Classifiability Based on Attributes

The attribute space classification of the characters in two stages of the pipeline —

Position of concavity	Type of concavity: L	B	R	U
−2		N	35S	
−1			3	
+1	G5S			HKNY
+2	CHKhn			

Position of hole	
−2	d
−1	q4
0	o
+1	P
+2	bd

Figure: Decomposition of equivalent class CGHKNSXhn35 based on rough-set attributes

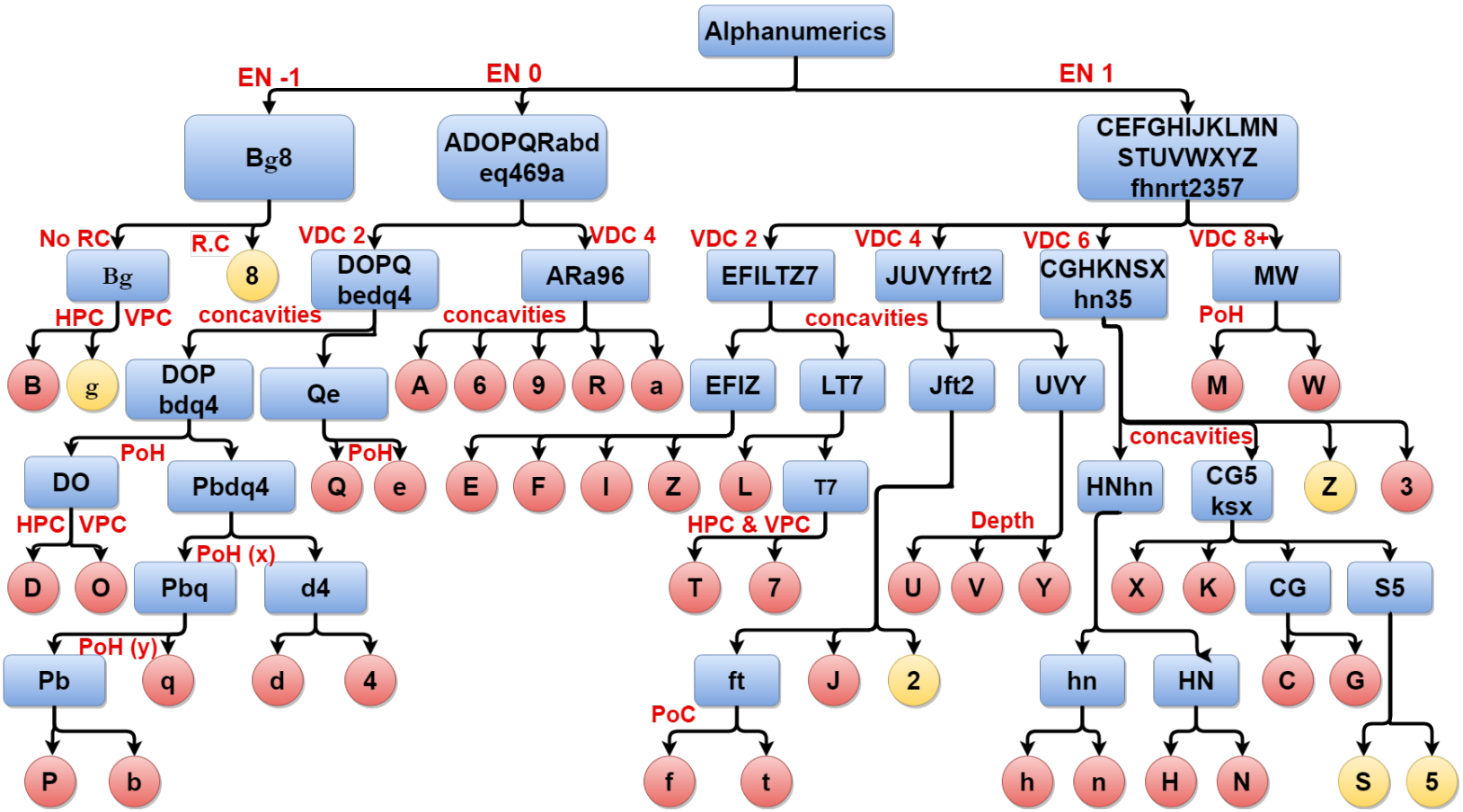
Limited Interface to Supervised Classifier

- Classification is context-free;
  (O/o/0), (i/l/I/1), (C/c), (J/j), (K/k), (M/m), (P/p), (S/s), (U/u), (V/v), (W/w), (X/x), (Y/y), (Z/z).
- Characters which bear structural resemblance with each other over a varied font style;
  B(z/2), (9/g), (s/5), and (g/8).
  These four character pairs are included in the indefinite class.

Tesseract Reinforced Model

- Few characters with high confusion are not well-discernible by the combinatorial attributes in the reduct.
- Certain atypical or idiosyncratic font styles creating confusion in the discretized topological space defined for the rough set.
- We design a hybrid model of character classification wherein we run the Tesseract for all the indefinite class characters.
- Improves the overall performance of the system, significantly gains in the average runtime.

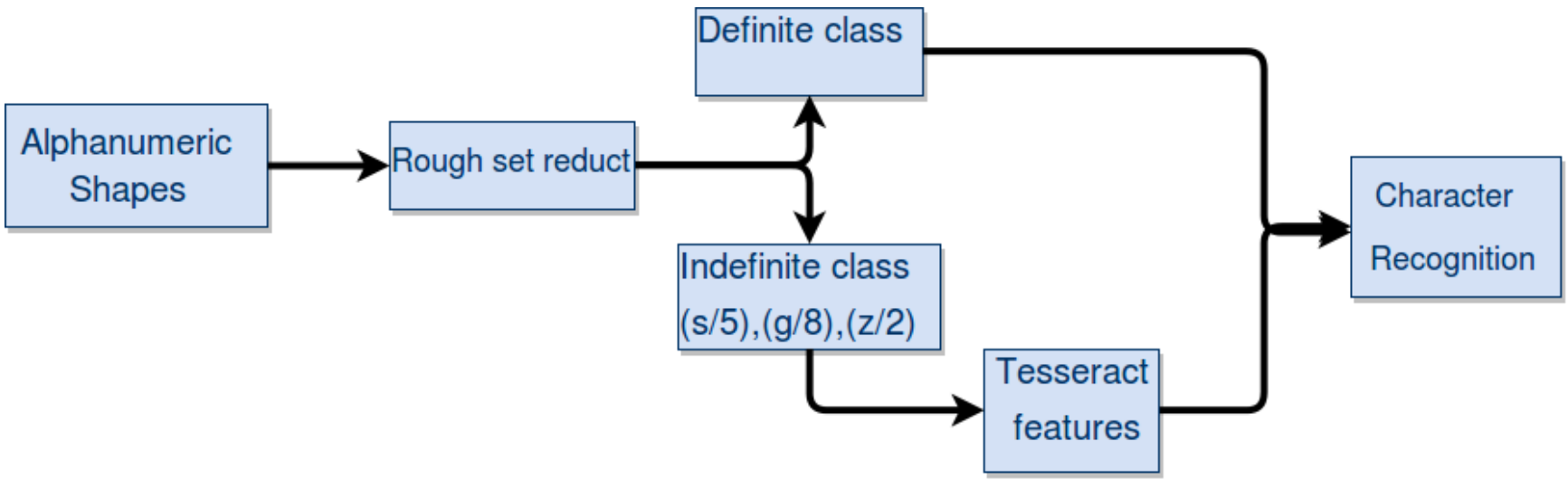
Classification Criteria



Red— Unsupervised, Yellow— Supervised

Block Diagram of the Overall Method

Proposed algorithm with Tesseract reinforcement.



Data Set

For testing, we have used the Chars74k dataset of optical characters [7].
This dataset contains

- images of 26 capital letters
- 26 small letters
- ten numeric digits.

They are written with 1016 different font styles.
Each image has a resolution of 128×128 pixels.

[7] "http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/"

Results

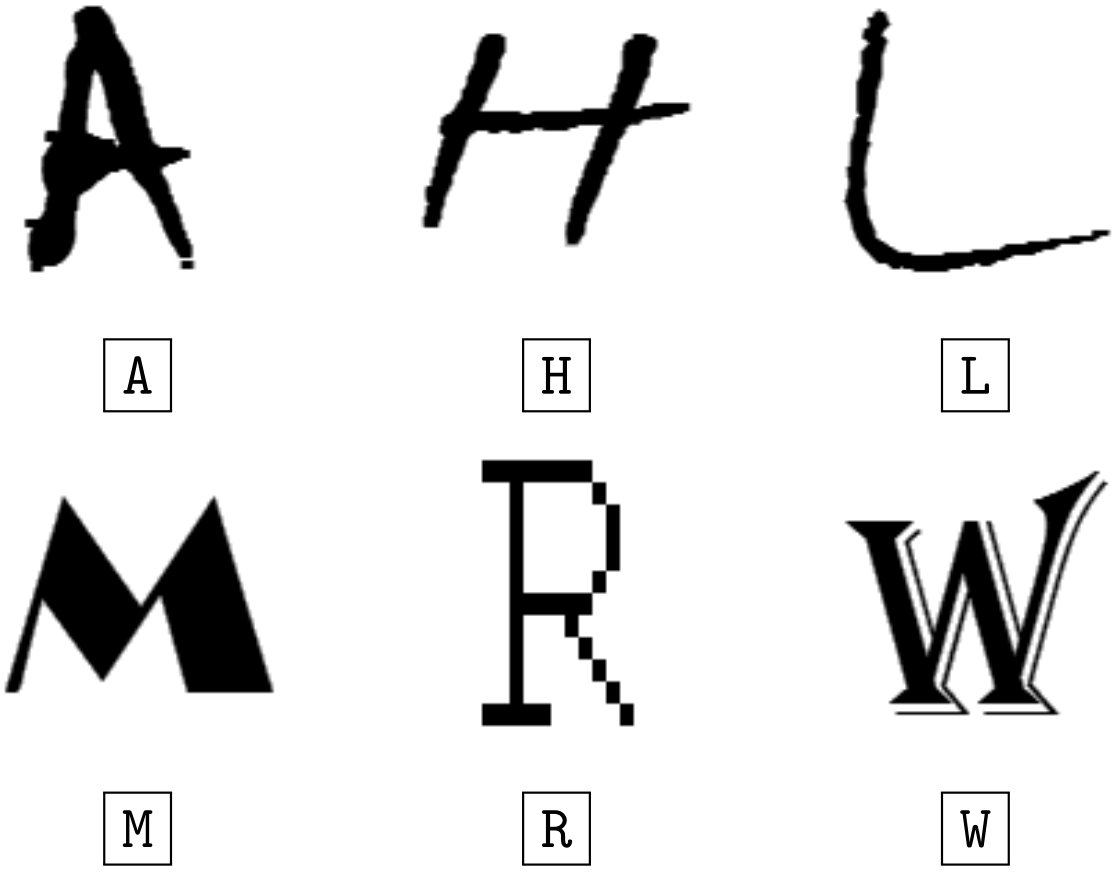


Figure: Reduct and Tesseract are independently successful.

Results

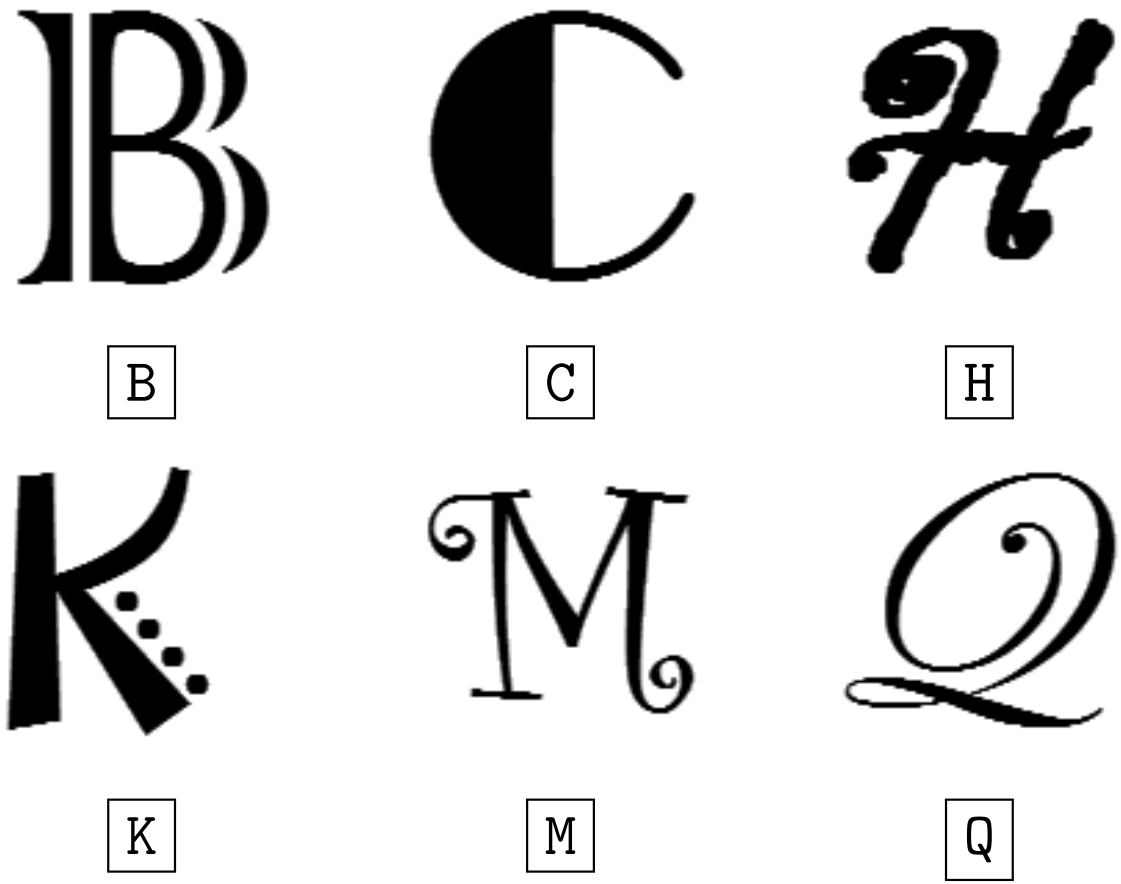


Figure: Reduct is successful; Tesseract fails.

Results

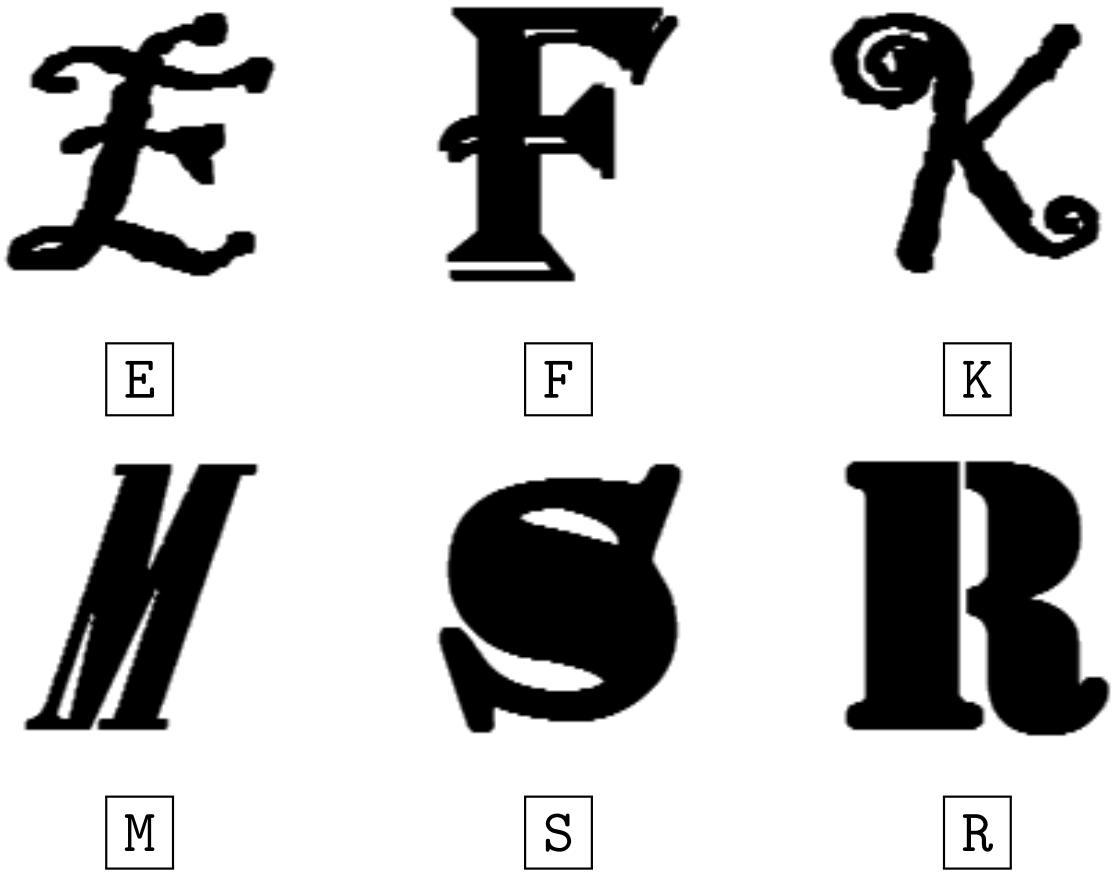


Figure: Reduct combined with Tesseract is successful.

Results

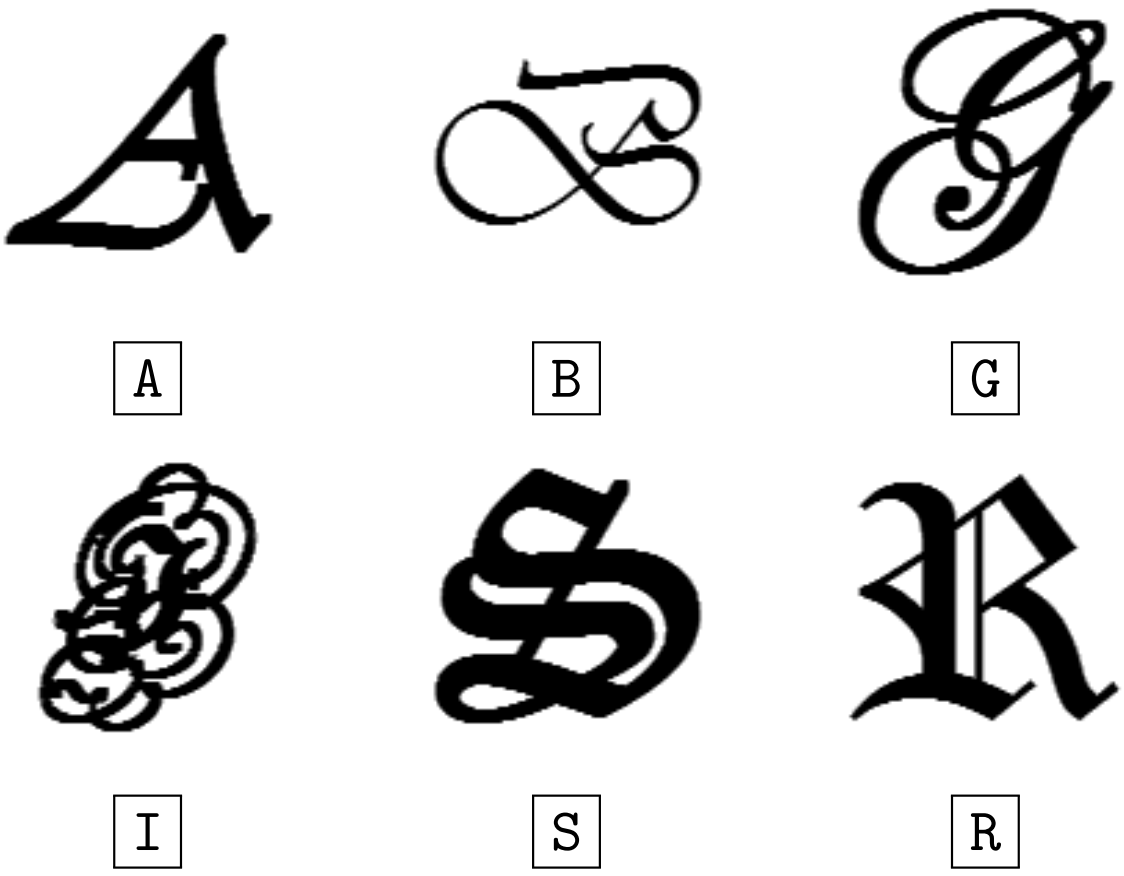


Figure: None is successful.

Results

To improve the accuracy, we group the characters into two classes, definite class and indefinite class. The indefinite class characters are those characters which are either unclassified or the characters having lower accuracy in classification.

- We get a result of about 88.98% accuracy
- The standard Google Tesseract, version 3.02.02, gives 64.79% with the inbuilt *eng.trainneddata* training.

Table: Experimental results.

Model	Runtime	Results
Google Tesseract™ 3.02.02	0.203 secs	64.79%
Proposed Model	0.051 secs	78.29%
Hybrid model	0.079 sec	88.98%

Results

This CPU time is achieved on a 64-bit Intel® 2-Core™ i5 processor, with 4GB RAM, DELL machine.

Accuracy		
Letters	Our Algorithm	Tesseract
CEIJKLMSVXYZf83	Above 90%	75.49 - 90.84%
ABDFHNOPQRTUW	80 - 90%	4.90 - 87.00%
Gbdem247	70 - 80%	5.01 - 80.70%
anqrt56	60 - 70%	0.78 - 52.85%
gh9	below 60%	1.08 - 60.33%
Average 88.98%	-	64.79%

Table: Comparison by accuracy

Conclusion

- Main advantage of the grid polygon approximation method is its reduced time complexity.
- When used for some large scale classification process, it saves us a huge amount of time.
- Also memory efficient as only classification criteria to be stored.
- The Tesseract used here is the untrained Tesseract. It's accuracy can be improved with training but at the cost of reduced speed.
- Reduced number of invocation of Tesseract in hybrid model for computational saving.

Scope for Future Work

- To extend the work to develop a PDF reader. .
- These geometrical features can be further used for various other applications like electrical symbol classification, logo retrieval, etc.
- To investigate the effect of skew on OCR accuracy.
- Possibility of application in Indian language OCR.
- To study robustness of the attributes in presence of noise.

Thank You