A Novel OCR System Based on Rough Set Semi-reduct

Ushasi Chaudhuri(✉), Partha Bhowmick, and Jayanta Mukherjee

Indian Institute of Technology Kharagpur, Kharagpur, India
ushasi.cdry@gmail.com, {pb,jay}@cse.iitkgp.ernet.in

Abstract. Most of the well-known OCR engines, such as Google Tesseract, resort to a supervised classification, causing the system drooping in speed with increasing diversity in font style. Hence, with an aim to resolve the tediousness and pitfalls of training an OCR system, but without compromising with its efficiency, we introduce here a novel rough-set-theoretic model. It is designed to effectuate an unsupervised classification of optical characters with a suboptimal attribute set, called the semi-reduct. The semi-reduct attributes are mostly geometric and topological in nature, each having a small range of discrete values estimated from different combinatorial characteristics of rough-set approximations. This eventually leads to quick and easy discernibility of almost all the characters irrespective of their font style. For a few indiscernible characters, Tesseract features are used, but very sparingly, in the final stages of the OCR pipeline so as to ensure an attractive run time of the overall process. Preliminary experimental results demonstrate its further scope and promise.

Keywords: OCR · Geometric features · Combinatorial features · Approximate reasoning · Rough set · Semi-reduct

1 Introduction

Optical character recognition (OCR) continues to remain a demanding subject in the field of document digitization [12]. It has a multitude of connections with many text- and image-related applications, and to name a few, these are editing, searching, and formatting of text for a better recognition model [7–9].

With growing demand of OCR, designing of an efficient OCR system is gradually becoming more challenging and cumbersome. The challenge, in fact, shoots up to an inordinate level when the optical characters are scripted using atypical and complex font styles, thus making the datasets huge in volume and diversity. Training the OCR system becomes a natural way out to meet this challenge, but this has several pitfalls. One is the immense time and tenacity required to selectively prepare the training set. Another is the slowdown of the OCR engine owing to too much dependence on the training-set prototypes for getting a reasonable solution.

B.U. Shankar et al. (Eds.): PReMI 2017, LNCS 10597, pp. 263–269, 2017.
https://doi.org/10.1007/978-3-319-69900-4_33

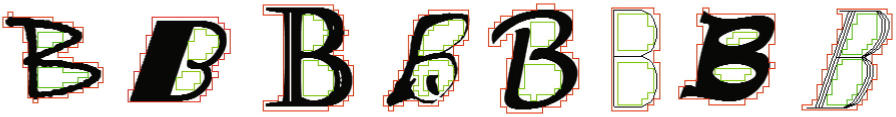


Fig. 1. Different instances of ‘B’ where (approximate) Euler number remains invariant as a semi-reduct attribute (red = outer polygon, green = hole polygons). (Color figure online)

Clearly, with increasing volume and diversity of datasets, it is required that we perform the recognition of characters in the least computational time possible. There exist several algorithms implemented and tested for performing this task. We refer to [5,6] and the bibliographies therein for their comparative study.

The Google Tesseract, an open-sourced OCR [13], is recognized as a powerful model since a long time. It uses various geometric features for its OCR engine. However, it requires a tedious training process to improve the efficiency of the character recognition. The training set, when large in size, also reduces the speed of the OCR engine quite drastically. Hence, to strike a balance, up to 32 trained data samples can be provided to the Tesseract after which its performance starts deteriorating.

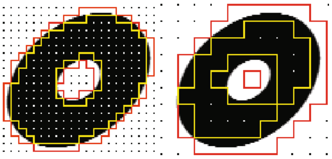
In order to circumvent the pitfalls of training and supervised classification in case of large datasets with rich and diverse scripting styles, we address the OCR problem with a new perspective of rough set. Each optical character is treated as a digital object, laid on a cellular grid, and approximated by its tightest cover called *rough-set cover* [14]. In order to define the reduct, a small set of attributes is considered, which are mostly geometric and topological in nature and defined in a combinatorial way in the discrete domain of rough set. As an introductory example, we have shown in Fig. 1 how different complex instances of the English optical character ‘B’ get associated with the same value of (approximate) Euler number (discussed in Sect. 2) when its rough-set cover is considered. Notice that this is not feasible by a usual image analysis, wherein lies the importance of rough set.

We aim to create a rough-set semi-reduct for an alphanumeric character set so as to design an efficient OCR pipeline. As the dataset we have taken up is quite complex and challenging, the reduct attributes are sometimes not enough in discriminating two characters with a high confusion. Hence, as a reinforcement, we use Tesseract features, although very occasionally and only towards the final stages of the pipeline. This not only improves the overall performance of the system but also significantly gains in the average runtime, as shown in Sect. 3.

2 Rough Set Reduct

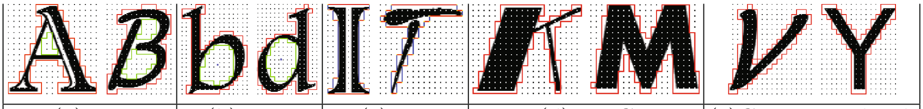
We use the concepts of rough set mostly from [10,11]. We use them in two stages—first for construction of upper and lower approximations of a 2D digital object and next for defining the approximations of their attributes comprising the reduct.

Let S be a 2D digital object and $\mathbb{G}$ a cellular grid. We denote by $\overline{\mathcal{P}}_{\mathbb{G}}(S)$ and $\underline{\mathcal{P}}_{\mathbb{G}}(S)$ the respective *tight upper approximation* and *tight lower approximation* of S induced by $\mathbb{G}$. Each of them essentially consists of one or more polygons with axis-parallel edges induced by $\mathbb{G}$.



Each polygon has two types of vertices, one of 90^0 and another of 270^0 interior angle, which we denote by '+' and '−', respectively. Depending on the grid resolution, the *accuracy* of the rough-set representation of S is given by $\alpha_{\mathbb{G}}(S) = \frac{\text{area}(\underline{\mathcal{P}}_{\mathbb{G}}\text{S})}{\text{area}(\overline{\mathcal{P}}_{\mathbb{G}}\text{S})}$. In the inset figure, there are two such approximations for cell size 6 × 6 and 12 × 12; the upper approximation is shown in red and the lower one in yellow.

Since each digital object S corresponds to a specific optical character, we first apply an isotropic scaling on S so that S fits inside a box of predefined height (128 in our experimental setup), and then set the grid by cell-size 4×4. We use the algorithm in [1] for construction of $\overline{\mathcal{P}}_{\mathbb{G}}(S)$. From the vertex sequence of the polygon(s) in $\overline{\mathcal{P}}_{\mathbb{G}}(S)$, we compute the values of the semi-reduct attributes (i.e., features), as discussed next.



(a) EN | (b) PoH | (c) ER | (d) VDC | (e)Concavity depth

Fig. 2. Some typical examples on the discriminating power of rough-set attributes.

1. Euler number. The upper approximation $\overline{\mathcal{P}}_{\mathbb{G}}(S)$ consists of one or more polygons. The largest among them is the *outer polygon*, and it tightly circumscribes S. Each other polygon tightly inscribes a hole or cavity of S, and is treated as a *hole polygon*. To capture this information, we consider approximate Euler number (EN) as an important attribute, and define it as $2 - n$, where n is the total number of polygons in $\overline{\mathcal{P}}_{\mathbb{G}}(S)$. In Fig. 2a, the character images 'A' and 'B' have $n = 2$ and $n = 3$, whereby EN $= 0$ and 1, respectively, thus discriminating them. Notice that without rough-set interpretation, the instance of 'B' would have EN $= 1$ by conventional image processing, which would produce erroneous result in subsequent analysis.

2. Hole positions. 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 $\overline{\mathcal{P}}_{\mathbb{G}}(S)$. In Fig. 2b, we see how the characters 'b' and 'd' are differentiated by this attribute: c lies right of v_0 for 'b', and left in case of 'd'. We assign '−' and '+' to denote left and right lateral halves, and '1' and '2' for respective upper and lower halves; hence, the hole polygon in ‘b’ has PoH $= +2$ and that in ‘d’ has PoH $= -2$.

3. Edge ratio. For each polygon in $\overline{\mathcal{P}}_{\mathbb{G}}(S)$, we define horizontal perimeter component (HPC) as the sum of lengths of its horizontal edges and vertical perimeter component (VPC) as that corresponding to its vertical edges. The ratio VPC:HPC, discretized to the nearest value in $\{\frac{1}{2}, 1, 2\}$, is called edge ratio (ER). As clear from Fig. 2c, this attribute really comes handy for discriminating characters like ‘I’ and ‘T’.

4. U-turns. While traversing along the boundary of the outer polygon, the number of ‘U-turns’ along the vertical direction is defined as vertical direction change (VDC). Each U-turn is defined by a vertex sequence where two consecutive vertices are of type $\langle +, + \rangle$; and for each such U-turn, we also consider their relative positions similar to PoH. Figure 2d shows how two characters are discriminated by VDC; here, VDC(‘K’) $= 6$ and VDC(‘M’) $= 8$. A similar measure along the horizontal direction gives horizontal direction change (HDC), which, however, is not found to be a strong discriminating attribute as VDC. It is hence omit-able from the reduct, while keeping the classification preserved, as inferred from our experimentation and hence not included in the semi-reduct.

5. Concavities. As shown in [2], concavity serves as an important characteristic of any shape. Hence, we use concavity as an attribute and define it as two consecutive vertices of type $\langle -, - \rangle$. We classify a concavity depending on its orientation: left (L), right (R), upward (U), and downward (D). Further, as a rough-theoretic measure, we discretize the relative depth of each concavity to the nearest value in $\{1, 2, 3\}$. It is represented by a 3-tuple of the form $\langle$concavity direction, region, depth$\rangle$. In Fig. 2e, the characters ‘V’ and ‘Y’ have similar concavities (i.e., U) but have respective depths 2 and 1, and hence get discriminated.

In Table 1, we have shown the composition of reduct attributes for a subset of the English alphanumeric set. Notice that the attribute tuples are well-discernible, which justifies their merit in playing a decisive role in our OCR system. Figure 3 shows the pipeline in stages where each stage is based on a particular semi-reduct attribute. Observe that in the initial stages of the pipeline, the average number of objects per equivalence class is more, and the equivalence classes gradually get smaller in size down the pipeline until each character gets uniquely recognized. The characters in red-colored nodes are discernible using the semi-reduct attributes only, while those in yellow nodes are discriminated using Tesseract features on top of the semi-reduct towards the final stage of the pipeline.

3 Experimental Results

For testing, we have checked several datasets and finally have picked up `Chars74k` [3,4] to report here the test result. We select this dataset for its challenging scripting styles to OCR design. It contains images of 26 capital letters, 26 small

Table 1. Sample information table (shown partial) containing the object properties against the semi-reduct.

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

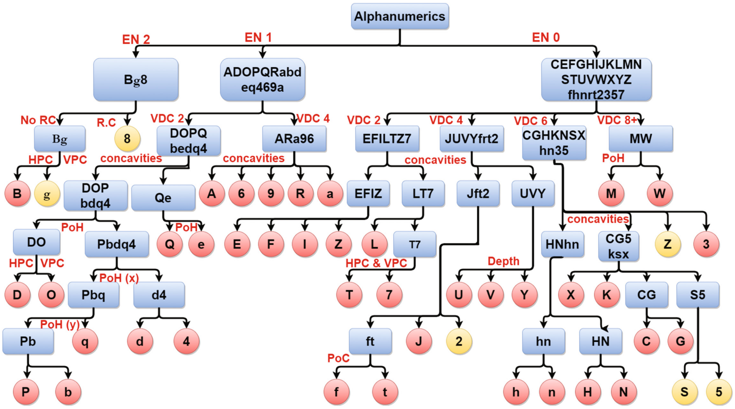


Fig. 3. Semi-reduct attributes working down the pipeline leads to decomposition of equivalence classes. (Color figure online)

letters, and ten numeric digits in English, written with 1016 different font styles. Each image has a resolution of 128 × 128 pixels.

We get an average CPU time of 0.051 s for the recognition of a character using our OCR engine. This is computationally attractive w.r.t. Google Tesseract engine that takes 0.203 s per character. This CPU time is achieved on a 64-bit Intel® 2-Core™ i5 processor, with 4 GB RAM, DELL machine. As shown in Table 2, we get a result of 88.98% accuracy using our model, while Google Tesseract, version 3.02.02, gives 64.79% with `eng.trainneddata` training set.

Since the characters are isolated objects, the classification is context-free; as a result, some character images are not mutually discernible. Hence, we categorize them in the same class: (0/o/O), (i/l/I/1), (C/c), (J/j), (K/k), (M/m),

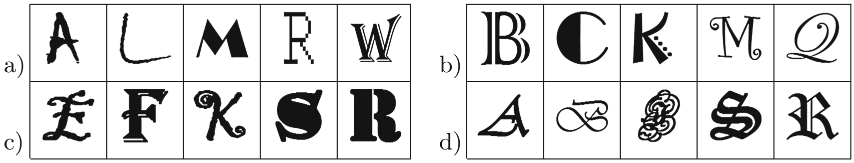


Fig. 4. Some typical instances of test cases to adjudge the quality of the proposed rough-set approach. (a) Semi-reduct and Tesseract are independently successful. (b) Semi-reduct is successful, Tesseract is not. (c) Semi-reduct combined with Tesseract is successful. (d) None is successful.

Table 2. Comparison by accuracy

Rough set	Letters	Tesseract
Above 90%	CEIJKLMSVXYZf83	75.49–90.84%
80–90%	ABDFHNOPQRTUW	4.90–87.00%
70–80%	Gbdem247	5.01–80.70%
60–70%	anqrt56	0.78–52.85%
50–60%	gh9	1.08–60.33%
Average 88.98%	–	64.79%

(P/p), (S/s), (U/u), (V/v), (W/w), (X/x), (Y/y), (Z/z). Also, other than these, there are a few characters which bear very close resemblance with each other over a varied font style, e.g., (z/2), (s/5), and (g/8/9). When the font style is complex, such as the ones used in scripting the letters shown in Fig. 4, there might be erroneous result owing to erratic mapping of the attribute values in the discretized space defined for the rough set. With larger dataset and more minute observation of their differences, discernibility of these characters can be targeted.

4 Conclusions

We have shown how a rough-set model with a small-cardinality semi-reduct can indeed be useful for quick and efficient discernibility of optical characters over varying font style. It has a significant operational difference with the existing techniques and can be designed to an efficient OCR with less runtime. The semi-reduct attributes used in our model are found to have strong discriminating power and can extend the concept further for OCR design in scripts other than English. Additional attributes can be explored and tested in different combination with these attributes to downsize a suboptimal semi-reduct to an optimal reduct, especially when a script has a large alphabet size.

References

1. Biswas, A., Bhowmick, P., Bhattacharya, B.: Construction of isothetic covers of a digital object: a combinatorial approach. JVCIR **21**(4), 295–31 (2010)
2. Bag, S., Bhowmick, P., Harit, G.: Detection of structural concavities in character images–a writer-independent approach. In: Kundu, M.K., Mitra, S., Mazumdar, D., Pal, S.K. (eds.) Perception and Machine Intelligence. LNCS, vol. 7143, pp. 260–268. Springer, Heidelberg (2012). doi:10.1007/978-3-642-27387-2_33
3. The Chars74K dataset: EnglishFnt (2009). http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/. Accessed 27 Mar 2017
4. de Campos, T.E., Babu, B.R., Varma, M.: Character recognition in natural images. In: Proceedings of the International Conference on Computer Vision Theory & Applications, Portugal (2009)
5. Fujisawa, H.: Forty years of research in character and document recognition—an industrial perspective. Pattern Recogn. **41**(8), 2435–2446 (2008)
6. Govindan, V.K., Shivaprasad, A.P.: Character recognition—a review. Pattern Recogn. **23**(7), 671–683 (1990)
7. Kumar, A., Jawahar, C.V., Manmatha, R.: Efficient search in document image collections. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007. LNCS, vol. 4843, pp. 586–595. Springer, Heidelberg (2007). doi:10.1007/978-3-540-76386-4_55
8. Laroum, S., Béchet, N., Hamza, H., Roche, M.: Hybred: an OCR document representation for classification tasks. Int. J. Comput. Sci. Issues **8**(3), 1–8 (2011)
9. Pati, P.B., Ramakrishnan, A.G.: Word level multi-script identification. Pattern Recogn. Lett. **29**(9), 1218–1229 (2008)
10. Pawlak, Z.: Rough sets. Int. J. Comput. Inf. Sci. **11**, 341–356 (1982)
11. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishing, Boston (1991)
12. Sarkar, P.: Document image analysis for digital libraries. In: Proceedings of the IWRIDL 2006, pp. 12:1–12:9 (2007)
13. Smith, R.: An overview of the Tesseract OCR engine. In: Proceedings of the ICDAR 2007, pp. 629–633 (2007)
14. Yao, Y.: Probabilistic rough set approximations. Int. J. Approx. Reason. **49**, 255–271 (2008)