Putting Humans Back in the Loop: A Study in Human-Machine Cooperative Learning

Milan Gnjatović[1,2], Nemanja Maček[2], Saša Adamović[3]

[1]Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia, e-mail: milangnjatovic@uns.ac.rs

[2]Faculty of Computer Science, Megatrend University, Bulevar maršala Tolbuhina 8, Belgrade, Serbia, email: nmacek@megatrend.edu.rs

[3]Faculty of Informatics and Computing, Singidunum University, Danijelova 32, Belgrade, Serbia, email: sadamovic@singidunum.ac.rs

Abstract: This paper introduces a novel approach to human-machine collaborative learning that allows for the chronically missing human learnability in the context of supervised machine learning. The basic tenet of this approach is the refinement of a human designed software model through the iterative learning loop. Each iteration of the loop consists of two phases: (i) automatic data-driven parameter adjustment, performed by means of stochastic greedy local search, and (ii) human-driven model adjustment based on insights gained in the previous phase. The proposed approach is demonstrated through a real-life study of automatic electricity meter reading in the presence of noise. Thus, a cognitively-inspired non-connectionist approach to digit detection and recognition is introduced, which is subject to refinement through the iterative process of human-machine cooperation. The prototype system is evaluated with respect to the recognition accuracy (with the highest digit recognition accuracy of 94%), and also discussed with respect to the storage requirements, generalizability, utilized contextual information, and efficiency.

Keywords: human-machine cooperative learning; digit recognition; stochastic search

1 Introduction

An important aspect of digital education relates to supporting the learner to acquire software development competencies. The main lines of research in this field include cost-effective simulation of programming environments [8, 20], dynamic adaptation of e-training [9, 36], human-machine interaction [31, 34] and collaborative learning [21, 22]. Recently, the research attention has been also devoted to developing specific digital teaching paradigms [5]. This paper makes a novel contribution to the field. It introduces an approach to human-machine

collaborative learning that allows for the chronically missing human learnability in the context of supervised machine learning.

Deep learning has undoubtedly made a significant breakthrough in many scientific domains. One of the main reasons of the enormous popularity of neural networks is that they – at least in the manner usually practiced – do not require considerable domain expertise or human engineering [26]. The very term "learning" is somewhat misused in this context. Deep learning relates to the process of encoding statistical regularities from the training corpora into parameters, which operate by very different principles from those underlying human learning [38]. Modern deep neural networks may contain up to hundreds of millions of automatically adjusted parameters [26], and derive high-dimensional representations that are not interpretable by human. Therefore, although deep learning may result in very useful software artifacts, it does not contribute to human learnability of domain expertise.

This paper[1] considers the question of bringing the human back into the learning loop. It proposes an approach to making the process of software development more explanatory to the practitioner, while keeping some of the existing advantages specific to supervised learning. The proposed approach is demonstrated through a real-life study of automatic electricity meter reading in the presence of noise.

1.1 Main Idea and Outline

This paper makes contributions along two research lines. First, it introduces a two-stage approach to digit detection and recognition (cf. Section 2). The approach is cognitively-inspired to the extent that it integrates the dichotomy between the pre-attentive processing and the attentive processing that is present in the theories of human attention [7, 13, 24, 31, 32, 35]. It is also parameterized, and the number of free parameters is small enough that the model is analytically tractable by human. To this extent, this approach is non-connectionist.

However, it cannot be assumed that this approach is *per se* generalizable to the target domain. This leads to the second research line – iterative refinement of the approach through human-machine cooperation. This research line derives from iterative and incremental software development practices that belong to the folklore of computer science [25]. However, it is particularly focused on supervised learning, and on what we refer to as "iterative learning loop", represented diagrammatically in Fig. 1. Each iteration of the loop consists of two phases (cf. Section 3):

[1] This paper is a significantly extended version of [18].

- Automatic data-driven parameter adjustment, performed by means of stochastic greedy local search over a training corpus.
- Human-driven model adjustment based on insights gained in the previous phase.

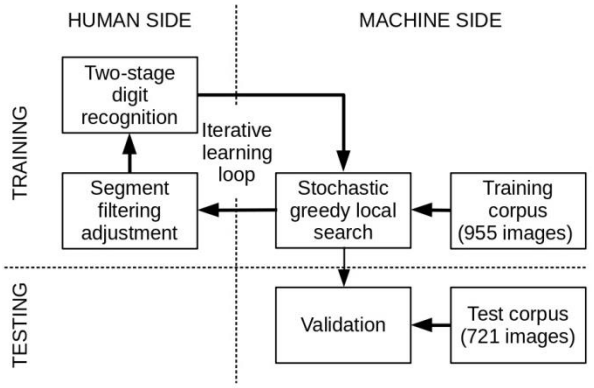

Figure 1
Diagrammatic representation of human-in-the-loop supervised learning

The main idea is to allow the practitioner to take advantage of automatic data-driven supervised learning – not only with respect to parameter adjustment (which is the usual case), but also to gain additional qualitative insights into the target domain. Thus, the iterative refinement of the approach to digit detection and recognition is both machine-driven and human-driven. The number of iterations is not predefined, i.e., the model is refined until it satisfies external requirements (in this paper, we describe two iterations).

After the iterative learning loop is completed, the model is validated on a test corpus. The paper ends with the discussion and conclusion (cf. Section 4).

2 Two-Stage Approach to Digit Recognition

The theories of human attention acknowledge that selective processing of sensory information has an important role in human cognition [2, 6]. However, they do not agree on the processing stage in which information are selected [32, 35]. The *early selection* view assumes that incoming sensory information are filtered in a processing stage prior to the stage of semantic interpretation, and that only the selected information is interpreted [7, 24]. The *late selection* view assumes that all sensory information is semantically interpreted, and that selection occurs in a later stage on the level of interpreted information [13]. A significant body of evidence supporting one or other of these views has been presented in this long-standing

debate, which may imply that the early and late selections of sensor information do not necessarily exclude each other [32].

In line with this, we assume the view that selection of sensory information occurs in two processing stages (cf. [31]):

- Pre-attentive processing serves as the basis for perceptual grouping,
- Attentive processing allows for semantic integration.

In the considered real-life study of automatic electricity meter reading, the pre-attentive processing stage is devoted to detecting relevant numbers, i.e., rows of digits that represent rates, while ignoring irrelevant digits (e.g., an electricity meter serial number) and symbols. Each number detected in the pre-attentive processing stage separately undergoes the attentive processing stage, devoted to recognition.

Two-stage processing has already been applied in various approaches to the research problem of object detection and recognition. At the methodological level, these approaches usually apply data-driven statistically-based techniques, such as the Markov random field theory and maximum a posteriori principle [30], the AdaBoost algorithm [15, 29], the support vector machine [1], neural networks [28], etc. (a more extensive overview is given in [15]). Symbolic approaches are applied significantly less often [17].

However, at the practical level, the production of representative and balanced training corpora is a challenging task, especially in such cases when surface manifestations of noise in image vary significantly. In contrast to the dominant trend, the approach introduced in this section is feature-based, but still non-connectionist in the sense that it does not require an extensive training corpus.

To extract feature vectors that represent image segments, we refer to the normalized histogram of oriented gradients [10]. To estimate the similarity between the feature vectors, we apply the cosine similarity [12].

2.1 Feature Extraction

Let f be an image segment, and let $f(x,y)$ be the intensity of segment *f* at pixel (x,y). To compute the gradient of *f* at pixel (x,y), we apply the horizontal and vertical Sobel filters:

$$\nabla f(x,y) = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} g_x(x,y) \\ g_y(x,y) \end{bmatrix} = g(x,y) \tag{1}$$

This gradient vector can be equivalently represented by its magnitude $\|g(x,y)\|$ and gradient direction $\theta(x,y)$, i.e.:

$$g(x,y) = (\|g(x,y)\|, \theta(x,y)) \tag{2}$$

where the magnitude is approximated by:

$$\|g(x,y)\| = \sqrt{g_x(x,y)^2 + g_y(x,y)^2} \approx |g_x(x,y)| + |g_y(x,y)| \tag{3}$$

and the gradient direction is calculated as:

$$\theta(x,y) = \begin{cases} \arctan \dfrac{g_y(x,y)}{g_x(x,y)}, & \text{if } g_x > 0, \\ \arctan \dfrac{g_y(x,y)}{g_x(x,y)} + \pi, & \text{if } g_x < 0, \\ \dfrac{\pi}{2} \cdot \operatorname{sgn}(g_y), & \text{if } g_x = 0. \end{cases} \tag{4}$$

Gradient vector g(x, y) is further decomposed along n chain code directions, i.e., it is decomposed along a set of n elementary vectors rotated in increments of $2\pi/n$ (where n is an input parameter), having its gradient direction $\theta(x,y)$ approximated to the closet chain code direction. More formally, gradient vector g(x, y) is mapped onto a feature vector:

$$\alpha(x,y) = (a_0(x,y), a_1(x,y), \ldots, a_{n-1}(x,y)) \tag{5}$$

in which all elements but one are equal to zero, and the value of the nonzero element is equal to the gradient vector magnitude $\|g(x,y)\|$:

$$a_i(x,y) = \begin{cases} \|g(x,y)\|, & \text{if } i = \left\lfloor \dfrac{\theta(x,y) \cdot n}{2\pi} \right\rfloor, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

for $0 \le i \le n-1$. Thus, a pixel at (x,y) is represented by an n-dimensional feature vector $\alpha(x,y)$. To represent entire segment f, it is partitioned into an $M \times N$ grid of rectangular cells:

$$B_0(f), B_1(f), \ldots, B_{M \times N - 1}(f), \tag{7}$$

where M and N are input parameters. Each cell $B_i(f)$ is represented by an n-dimensional feature vector:

$$\beta_i(f) = (b_{i,0}(f), b_{i,1}(f), \ldots b_{i,n-1}(f)), \tag{8}$$

calculated as the sum of all feature vectors that represent the pixels belonging to the given cell, i.e.:

$$b_{i,j}(f) = \sum_{(x,y)\in B_i(f)} a_j(x,y), \tag{9}$$

where $0 \le i \le M \times N - 1$ and $0 \le j \le n-1$

Feature vector $\hat{\chi}(f)$ that represents segment f is generated in two steps. First, all cell feature vectors $\beta_0(f), \beta_1(f), \ldots, \beta_{M\times N-1}(f)$, are concatenated:

$$\chi(f) = (c_0(f), c_1(f), \ldots, c_{n\times M\times N-1}(f)), \tag{10}$$

where

$$c_i(f) = b_{\left\lfloor \frac{i}{n} \right\rfloor, (i \bmod n)}(f), \tag{11}$$

and then each element in vector $\chi(f)$ is normalized with respect to the entire segment, i.e.:

$$\hat{\chi}(f) = (\hat{c}_0(f), \hat{c}_1(f), \ldots, \hat{c}_{n\times M\times N-1}(f)), \tag{12}$$

where:

$$\hat{c}_i(f) = \frac{c_i(f)}{\sqrt{\sum_{k=0}^{n\times M\times N-1} c_k(f)^2}}, \tag{13}$$

and $0 \le i \le n \times M \times N - 1$. An image segment is represented by a feature vector whose size (i.e., $n \times M \times N$ elements) is constant and does not depend on the size of the segment. This enables the comparison of segments of different sizes.

2.2 Feature Vector Similarity

Let $\hat{\chi}(f_1)$ and $\hat{\chi}(f_2)$ be feature vectors that represent image segments f_1 and f_2, respectively. As a measure of segment similarity, we apply the cosine similarity:

$$sim(f_1, f_2) = \frac{\hat{\chi}(f_1) \cdot \hat{\chi}(f_2)}{\|\hat{\chi}(f_1)\| \cdot \|\hat{\chi}(f_2)\|} = \frac{\sum_{i=0}^{n\times M\times N-1} \hat{c}_i(f_1)\hat{c}_i(f_2)}{\sqrt{\sum_{i=0}^{n\times M\times N-1} \hat{c}_i(f_1)^2}\sqrt{\sum_{i=0}^{n\times M\times N-1} \hat{c}_i(f_2)^2}} \tag{14}$$

Since all elements in feature vectors are nonnegative, $sim(f_1, f_2)$ will always be in range $[0,1]$ – the higher the score, the more similar the feature vectors.

2.3 Pre-attentive and Attentive Processing

The electricity meter reading is conducted in the following stages: (i) preparatory stage (i.e., image pre-processing), (ii) pre-attentive processing (i.e., number detection), and (iii) attentive processing (i.e., number recognition). The image pre-processing includes:

- Color to grayscale conversion, according to the ITU-R recommendation BT.709-6 (06/2015) [23],
- Contrast stretching, i.e., image enhancement by means of increasing the dynamic range of its gray levels [19, pp. 85-86],
- Adaptive global thresholding [33, pp. 120-121], i.e., binarization of the image by means of separation of light and dark regions.

The other two stages are described in more detail in the following subsections.

2.3.1 Pre-attentive Processing: Number Detection

The introduced approach is feature-based. Each digit $d \in \{0,1,\ldots,9\}$ is described by a feature vector $\hat{\chi}(d)$ extracted from a binarized image of digit d, as described in Subsection 2.1. Let T be the set of all ten ground-truth feature vectors:

$$T = \{\hat{\chi}(0), \hat{\chi}(1), \ldots, \hat{\chi}(9)\}. \tag{15}$$

In the pre-attentive processing stage, a sliding window is used to search through the image and perform early selection of relevant image segments. For each sliding window segment f, its feature vector $\hat{\chi}(f)$ is generated and compared to the ground-truth feature vectors in T. If the maximum similarity value of $\hat{\chi}(f)$ with each one of the ground-truth feature vectors is greater than the predefined threshold value λ_1 (which is also an input parameter), i.e.,

$$\max_{t \in \mathrm{T}} sim(\hat{\chi}(f), t) > \lambda_1, \tag{16}$$

segment f is marked as potentially containing a digit, and added to set P.

It should be noted that the mapping of relevant digits in the input image onto segments in set P is not intended to be bijective. The size and step of the window are input parameters. Depending on their values, segments stored in P may overlap (i.e., they may contain the same digit), or a digit in the input image can

remain undetected. To illustrate this, we deliberately selected non-optimal values of the size and step of the sliding window (i.e., the window size is smaller than optimal, and the step is greater than optimal). The result is shown in Fig. 2. For the purpose of presentation, the width and height of all images are scaled up by the factor of two, and the rectangles that designate segments are automatically generated by the prototype system. The grayscale input image is given in Fig. 2(a). The segments in P are depicted in Fig. 2(b). Some segments are incorrectly marked as containing a digit, while some digits were not detected at all. However, a segment that contains the entire number can be determined as the *minimum rectangular segment* that contains all segments in P, as depicted in Fig. 2(c).

Figure 2
Number detection in the pre-attentive processing stage: (a) the grayscale input image, (b) the segments detected as containing digits, (c) the segment containing the entire number (images adopted and adjusted from [18])

In addition, since an electricity meter may contain more than one row of relevant digits, after the window slides through the entire image, set P is partitioned so that each subset $\mathrm{P}_i \subseteq \mathrm{P}$ contains digits that belong to a separate number. For the purpose of this contribution, we assume that rows of digits are presented one below the other (which is often the case). We consider that segments p_i and p_j from set P are related (i.e., they contain digits belonging to the same number) if they overlap along y-axis or if there is another segment $p_k \in \mathrm{P}$ such that p_k is related both to p_i and p_j. This relation between segments is an equivalence relation, and it is used to partition set P:

$$(\mathrm{P} = \bigcup_i \mathrm{P}_i) \wedge (\forall \mathrm{P}_i, \mathrm{P}_j \in \mathrm{P})(\mathrm{P}_i \neq \mathrm{P}_j \Rightarrow \mathrm{P}_i \cap \mathrm{P}_j = \emptyset), \tag{17}$$

where each subset $\mathrm{P}_i \subseteq \mathrm{P}$ relates to a separate number. This is illustrated in Fig. 3(a,b). The grayscale input image given in Fig. 3(a) contains two relevant numbers, i.e., the electricity meter has two rates. The set of segments detected by the sliding window is partitioned into two subsets: P_1 and P_2 (cf. Fig. 3(b)). The image segment containing the first number is determined as the minimum rectangular segment that encapsulates all segments in P_1. Similarly, the minimum rectangle segment that encapsulates all segments in P_2 relates to the second number.

In general, after the completion of the pre-attentive processing stage, a set of segments containing relevant numbers in the input image is extracted:

$$\{f_{num_1}, f_{num_2}, \ldots, f_{num_k}\}. \tag{18}$$

Each of these segments separately undergoes the attentive processing stage.

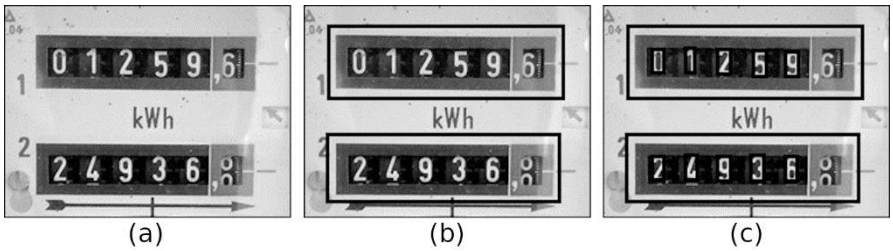

Figure 3

(a) The grayscale input image, (b) number detection in the pre-attentive processing stage, (c) number recognition in the attentive processing stage (images adopted and adjusted from [18])

2.3.2 Attentive Processing: Number Recognition

In the attentive processing stage, the late selection performed over the segments containing relevant numbers includes the following steps:

(i) Segmentation. Each segment f_{num_i} is further segmented by applying the graph-based image segmentation algorithm introduced in [14]. This segmentation algorithm is adapted only with respect to the threshold value used to merge segments – we use a fixed threshold value μ passed as an input parameter.

(ii) Segment filtering. We discard subsegments whose dimensions (relative to the size of the containing segment f_{num_i}), black/white pixel ratios or height-to-width ratios are not in the expected ranges for a digit. It should be noted that the segment filtering conditions are not algorithmically-driven but rather based on the authors' qualitative and inherently limited insights into the domain problem. Therefore, they are subject to further refinement, as described in Section 3.

(iii) Segment classification. For each remaining subsegment f', its feature vector $\hat{\chi}(f')$ is extracted and compared to the ground-truth feature vectors in T (cf. Eq. (15)). If the maximum similarity value of $\hat{\chi}(f')$ with each one of the ground-truth feature vectors is greater than the predefined threshold value λ_2, i.e.,

$$\max_{t \in \mathrm{T}} sim(\hat{\chi}(f'), t) > \lambda_2, \tag{19}$$

the segment is recognized as containing digit $dig(f')$, where:

$$dig(f') = \arg\max_{t \in \mathrm{T}} sim(\hat{\chi}(f'), t). \quad (20)$$

E.g., when two segments designated in Fig. 3(b) are passed to the attentive processing stage, the recognition results are indicated in Fig. 3(c). The digits after the decimal points are deliberately ignored, in accordance with the external requirements.

3 Iterative Learning Loop

The approach to digit detection and recognition introduced in Section 2 is subject to refinement through the iterative process of human-machine cooperative learning, combining automatic data-driven parameter adjustment with human-driven model adjustment (cf. Subsection 1.1).

The image corpora used in the iterative learning loop (and in the subsequent evaluation of the system, cf. Section 5) contain real-life electricity meter images with significant noise and incompleteness from various sources. Electricity meters are inconsistently illuminated, physically damaged (e.g., scratched glass) and obscured by dirt or dust. All the images were captured by naïve operators using standard Android-based phones. The *training corpus* consists of 955 images containing only one rate (i.e., one row of relevant digits and the surrounding context, similarly as in image given in Fig. 2). This corpus is used for automatic data-driven parameter adjustment. The *test corpus*, used for the purpose of evaluation, is described in Section 4.

Table 1

Parameters – marked with * if they are actually optimized

Parameter	
n - number of elements in a feature vector representing a pixel, cf. Eq. (5)	*
$M \times N$ - dimension of a grid of rectangular cells, cf. Eq. (7)	*
Dimension of the sliding window	
Steps of the sliding window along the x and y axes	
λ_1 - threshold for digit detection in the pre-attentive processing stage, cf. Eq. (16)	*
μ - threshold for graph-based segmentation in the attentive proc. stage, Sec. 2.3.2	*
λ_2 - threshold for digit recognition in the attentive processing stage, cf. Eq. (19)	*

3.1 Stochastic Greedy Local Search

The approach to digit detection and recognition is parameterized by a set of ten parameters (cf. Table 1) which are subject to data-driven optimization. The space of possible parameter assignments is too large for an exhaustive search. However, states in the search space are full assignments to all the parameters, which allows for applying a stochastic greedy local search algorithm. Similarly, as in deep learning approaches, the objective function is seen as a hilly landscape in the multidimensional space of parameter values [26], and therefore we apply an adapted hill-climbing algorithm with random restart [11, 37]. The algorithm is specified in Fig. 4 as a higher-order function, and its main idea may be described as follows.

Hill-climbing with random restart

Input: number or random restarts: r,
parameter domains: $D_0, D_1, \ldots, D_{n-1}$,
fitness function $h: D_0 \times D_1 \times \cdots \times D_{n-1} \to \Re$,
mutation oper.: $g: (D_0 \times D_1 \times \cdots \times D_{n-1}) \to \wp(D_0 \times D_1 \times \cdots \times D_{n-1})$.

Output: optimized instantiation of parameters for the fitness function:
$(p_0, p_1, \ldots, p_{n-1})$, where $(\forall\, 0 \le i < n)(p_i \in D_i)$.

Algorithm:

$E = \emptyset$
for $i = 0$ *to* r *do*
  $x = (rand(D_0), rand(D_1), \ldots, rand(D_{n-1}))$
  $search = true$
  while $(search)$ *do*
    $x' = \arg\max_{y \in g(x)} h(y)$
    if $(h(x) < h(x'))$ $x = x'$
    else $search = false$
  end_while
  $E = E \cup \{x\}$
end_for
return $\arg\max_{x \in E} h(x)$

Figure 4
Hill-climbing with random restart

The hill-climbing:

1. The greedy local search starts from a randomly chosen instantiation of parameters. We refer to it as the current instantiation x.

2. For the given current instantiation x, a set $g(x)$ of neighboring instantiations is generated. Let x' be the most fit instantiation in set $g(x)$.

3. If x' is more fit than x, then x' becomes the current instantiation, and the algorithm goes back to Step 2. Otherwise, the current instantiation is selected as a candidate for optimal solution.

Multiple random restart: To reduce the probability of getting stuck in a local extremum, the hill-climbing is restarted from finite number of different, randomly selected instantiations or parameters. Each run of the hill-climbing algorithm generates a candidate for optimal solution. After multiple restarts of the algorithm are completed, the most-fit candidate is selected as optimal solution.

Mutation (local changes in the search space)

Input: instantiation of parameters: $x = (p_0, p_1, \ldots, p_{n-1})$,
parameter domains: $D_0, D_1, \ldots, D_{n-1}$.

Output: set of n instantiations of parameters.

Algorithm:

$G = \emptyset$
for $i = 0$ *to* $n-1$ *do*
  for $j = 0$ *to* $n-1$ *do*
    if $(i \neq j)$ $x'[j] = x[i]$
    else $x'[j] = v \,|\, v \in D_j \wedge v \neq x[j]$
  end _ for
  $G = G \cup \{x'\}$
end _ for
return G

Figure 5
Mutation (local changes in the search space)

We use the following input arguments to the hill-climbing algorithm.

(i) Number of random restarts is set to five.

(ii) Parameters. Not all parameters given in Table 1 are actually optimized. The dimension of the sliding window and the steps of the sliding window along both the axes were predefined in line with the external requirements (cf. the discussion point on efficiency in Section 4). In addition, the threshold value for digit detection in the pre-attentive processing stage is considered to be equal to the threshold value for digit recognition in the attentive processing stage. The complete instantiation of the fitness function is represented as a set of parameters marked with * in Table 1.

(iii) The fitness h of a given instantiation of parameters is defined as the number of training samples that are recognized correctly and completely.

(iv) The mutation operation is defined in Fig. 5. Mutation of a given instantiation of parameters $x=(p_0, p_1,\ldots, p_{n-1})$ generates a set of n new instantiations, each of which differs from initial instantiation x in only one parameter value, i.e., mutation involves only local moves in the search space.

3.2 Learning through Human-Machine Cooperation

Each iteration of the learning loop includes automatic data-driven parameter adjustment, followed by human-driven model adjustment. We describe two iterations conducted in the study.

Iteration 1 – Automatic data-driven parameter adjustment: In the first iteration, the parameters are automatically optimized based on the introduced hill-climbing algorithm and the training corpus. At the digit level, 94% digits are correctly recognized, 3.16% incorrectly recognized, and 2.84% not detected. The confusion matrix is given in Table 2. At the number level, 67.02% images are completely recognized.

Table 2
Confusion matrix in the first iteration

	0	1	2	3	4	5	6	7	8	9	ND	Total
0	628	2	0	0	0	0	4	0	7	1	22	664
1	22	452	0	0	0	0	1	0	6	5	17	503
2	0	0	399	0	8	0	0	7	1	0	11	426
3	0	0	1	434	2	0	0	0	10	3	16	466
4	0	0	0	0	446	0	0	3	2	0	17	468
5	1	0	0	1	0	404	3	0	4	4	14	431
6	8	0	0	0	0	1	430	0	10	0	8	457
7	0	0	3	0	0	0	0	440	0	0	10	453
8	2	0	0	5	0	1	8	0	442	3	10	471
9	1	0	0	4	0	1	2	0	4	422	11	445
INS	78	19	15	15	29	7	23	6	95	34	-	321

INS – segment incorrectly recognized as a digit; ND – digit not detected

Iteration 1 – Human-driven model evaluation and adjustment: The analysis of the system's performance shows that, out of 4784 digits, 136 digits (i.e., 2.84%) are not detected. On the other hand, the number of segments incorrectly recognized as digits is more significant – 321 segments (which equals to 6.71% of all digits) – and affects the accuracy of the system more intensively. Thus, a new

insight into the domain problem emerges, which would otherwise remain hidden because it is not directly related to the parameter adjustment: the segment filtering (cf. point (ii) in Subsection 2.3.2) is the critical point of the system. It is clear that the segment filtering conditions should be adapted accordingly. However, their modification is a trade-off. By tightening the segment filtering conditions, the number of segments incorrectly recognized as digits will decrease, but it can be also expected that the number of digits that are not detected will increase. Therefore, the conditions were modified by the authors with the intention to achieve a balance between these two confronting factors. In addition, this modification may affect the appropriateness of the parameter values automatically derived in the previous phase. To address these issues, the second iteration of the learning loop is started.

Iteration 2 – Automatic data-driven parameter adjustment: In the second iteration, the parameters are again automatically optimized. The optimization resulted in the same instantiation of parameters as in the first iteration, and the confusion matrix is given in Table 3.

Table 3

Confusion matrix in the second iteration

	0	1	2	3	4	5	6	7	8	9	ND	Total
0	626	2	0	0	0	0	3	0	7	1	25	664
1	16	449	0	0	0	0	1	0	2	3	32	503
2	0	0	399	0	8	0	0	6	1	0	12	426
3	0	0	1	434	2	0	0	0	9	3	17	466
4	0	0	0	0	446	0	0	3	2	0	17	468
5	1	0	0	1	0	404	2	0	2	4	17	431
6	6	0	0	0	0	1	429	0	5	0	16	457
7	0	0	1	0	0	0	0	436	0	0	16	453
8	2	0	0	5	0	0	5	0	442	3	14	471
9	1	0	0	4	0	0	0	0	4	422	14	445
INS	29	12	6	4	19	7	8	2	45	17	-	149

INS – segment incorrectly recognized as a digit; ND – digit not detected

Iteration 2 – Human-driven model evaluation and adjustment: From Tables 2 and 3, it can be derived that the total number of segments that undergone classification in the attentive processing stage was reduced from 4969 in the first iteration to 4753 in the second. I.e., the number of processed segments decreased for 216 ($4969-4753$). This set of 216 omitted segments can be divided into two subsets. The first subset contains 172 segments that were incorrectly detected as digits in the first iteration, but correctly rejected in the second iteration. The second subset contains 44 segments that were correctly detected as digits in the first iteration, but not detected in the second iteration (cf. Table 4). The first subset is dominant, which implies that the modification of the segment filtering

conditions in the first iteration resulted in a more balanced relationship between the number of segments incorrectly recognized as a digit, and the number of digit that were not detected.

In addition, the recognition accuracy at the digit level slightly decreased: from 94% in the first iteration to 93.79% in the second. However, the recognition accuracy at the number level increased from 67.02% to 72.46%. This is not contradictory – the number recognition accuracy is increased due to the fact that the number of segments that were incorrectly detected as digits was significantly decreased as a result of the modification of the segment filtering conditions. These conditions can be further optimized in subsequent iterations, but the first two described iterations suffice to illustrate the proposed approach.

Table 4

Filtered segments

Iterative learning loop	# segments in the attentive stage	Average # segments per image	St. dev.	INS	ND
1st iteration	4969	5.20	1.02	321	136
2nd iteration	4753	4.98	0.87	149	180
Abs. difference	216	-	-	172	44

INS – segment incorrectly recognized as a digit; ND – digit not detected

4 Evaluation and Discussion

The automatically calculated instantiation of parameters and the human-adjusted segment filtering conditions are evaluated on the *test corpus* containing 721 images containing only one rate (and the surrounding context). To avoid bias (e.g., training on the test data), the training corpus, described in Section 4, and the test corpus do not overlap. More precisely, it is not only that the training and test corpora do not include images of the same electricity meters, but they also do not include images of the same electricity meter types.

Table 5

Confusion matrix in the evaluation phase

	0	1	2	3	4	5	6	7	8	9	ND	Total
0	406	6	0	1	0	0	2	0	8	2	10	435
1	1	340	0	0	1	0	2	0	3	0	36	383
2	0	0	320	0	1	0	0	10	1	0	7	339
3	1	0	0	272	0	3	1	0	14	15	13	319
4	0	0	2	0	339	0	0	1	1	0	14	357
5	2	0	1	1	0	316	12	0	1	3	18	354

6	10	0	0	4	1	5	321	0	7	5	9	362
7	0	0	2	1	0	0	0	328	2	0	12	345
8	2	1	0	0	0	1	5	0	344	3	11	367
9	3	1	0	1	0	0	0	0	5	317	9	336
INS	8	40	7	9	10	6	24	15	57	10	-	186

INS – segment incorrectly recognized as a digit; ND – digit was not detected

The confusion matrix is given in Table 5. The obtained results are comparable to the results from the training phase. At the digit level, 91.83% digits are correctly recognized, 4.31% incorrectly recognized, and 3.86% not detected. At the number level, 61.03% images are completely recognized. The number of filtered segments per image is 5.05, with standard deviation of 0.95. For the obtained digit recognition rate (i.e., p_d is equal to 93.79% in the training phase, and 91.83% in the testing phase), the reported five-digit number recognition rate is close to the expected value (which, for the illustration purposes, can be approximated as p_d^5). We recall that this accuracy was obtained for images with significant noise and incompleteness, and emphasize, in addition, the following points.

(i) Reduced storage requirements. The recognition process rely only on a set of ten ground-truth feature vectors describing digits in set $\{0,1,\ldots,9\}$ (cf. set $\top$ in Eq. (15)). The recognition accuracy would increase with the number of the ground-truth feature vectors, but we wanted to reduce the storage requirements, in order to make this approach applicable for embedded devices such as mobile phones.

(ii) Generalizability and contextual information. Most aspects of the proposed approach to digit recognition are not domain-specific, including the pre-processing, feature extraction, feature vector comparison, segmentation, and segment classification. A small domain-specific part of the approach includes the ground-truth feature vectors, conceptualization of number as a horizontal pattern of digits (cf. Fig. 2(c)), and segment filtering (cf. point (ii) in Subsection 2.3.2). However, the domain-specific information just encodes the properties of the ground-truth templates, and are thus adaptable to other object recognition domains. The proposed approach does not utilize any additional contextual information that might improve the recognition accuracy (e.g., the expected number of digits per number, etc.). That was an intentional decision, in order to additionally support our statement on the generalizability of the approach.

(iii) Efficiency. Special attention was devoted to the efficiency of the prototype system. Searching through an image with a sliding window in order to conduct early selection of relevant image segments (cf. Subsection 2.3.1) is time consuming operation. Therefore, as already mentioned in Section 3.1, we decided to use only one predefined dimension of the sliding window, and the steps of the window along the axes were also predefined and constant. If we had applied more sliding windows of different dimensions and with different steps, it would have additionally increased the recognition accuracy. However, we decided to adopt a

trade-off between the accuracy and the efficiency of the system. The average processing time per image is 0.57 s (with standard deviation of 0.22 s) in the training phase, and 0.59 s (with standard deviation of 0.17 s) in the testing phase (measured on a standard personal computer).

(iv) Single-frame recognition. In the reported experiment, each electricity meter was represented by one single image. In a practical application of this technology (e.g., using an Android-based phone), multiple frames of an electricity meter would be captured and processed. Since each recognized digit is assigned a similarity score (cf. Eq. (19)), the captured frames can be evaluated, and the most appropriate candidate selected – which would additionally increase recognition accuracy.

Conclusions

This paper identified two separate but related contributions. First, we introduced a cognitively-inspired, non-connectionist approach to digit detection and recognition, in the presence of noise. Second, we proposed a novel approach to human-machine collaborative learning. The basic tenet of this approach is the refinement of a human designed software model, through the iterative learning loop, combining automatic data-driven parameter adjustment with human-driven model adjustment. This approach is demonstrated through a real-life study of automatic electricity meter reading in the presence of noise.

In the terminology of cognitive info-communications, automatic object recognition is referred to as an elementary cognitive capability [3, 4], in contrast to the higher level cognitive capabilities such as affective computing [37], human augmentation and health monitoring [16]. However, the proposed approach is relevant to the field of cognitive info-communications in two respects. One of the fundamental cognitive capabilities that remained under-investigated in this field is learning from small sets of prior experiences. Our approach to digit detection and recognition tends to meet, although only partially, this desideratum – its advantage is that it does not require significant training data, which is demonstrated in [18]. On the other hand, machine learning-based systems are usually developed in an extrinsic manner, i.e., a system is trained as a whole. The automatic adjustment of a large number of parameters leaves the practitioner out of the learning loop – it neither allows for human learnability in the training phase, nor does it provide the practitioner with insights into the performance of individual subsystems. In contrast to this, we proposed an approach to human-in-the-loop supervised learning. To illustrate human learnability, in Subsection 3.2, we discussed the idea that the iterative learning loop enables the practitioner to recognize the segment filtering, as critically important and extends understanding. More generally, the iterative learning loop is intended to make the process of software development more explanatory to a practitioner, by enabling them to intrinsically develop and evaluate individual subsystems, while keeping the advantages specific to supervised learning. In the dominant trend of ever more complex systems, based

on black-box machine learning techniques, making the underlying computational models more human-interpretable, is an important requirement for computer-aided education in computer science.

Acknowledgement

The presented study was funded by the Ministry of Education, Science and Technological Development of the Republic of Serbia (research grant III44008) and within the framework of the ERA.Net RUS Plus program (research grant ID 99). The test images are provided by the courtesy of the TCOM doo Beograd. The authors have the right to publish the material, and declare no conflict of interest.

References

[1] M. Anthimopoulos, B. Gatos, I. Pratikakis: A two-stage scheme for text detection in video images, Image and Vision Computing, 28(9):1413-1426, 2010

[2] A. Baddeley: Working Memory and Language: An Overview, Journal of Communication Disorders, 36(3):189-208, 2003

[3] P. Baranyi, A. Csapó: Definition and Synergies of Cognitive Infocommunications, Acta Polytechnica Hungarica, 9(1):76-83, 2012

[4] P. Baranyi, A. Csapó, G. Sallai: Cognitive Infocommunications (CogInfoCom), Springer International Publishing, 2015

[5] K. Bubnó, V. L. Takács: Cognitive aspects of 'Mathematics aided computer science teaching', Acta Polytechnica Hungarica, 16(6) :73-93, 2019

[6] C. Bledowski, J. Kaiser, B. Rahm: Basic Operations in Working Memory: Contributions from Functional Imaging Studies, Behavioural Brain Research, 214(2):172-179, 2010

[7] D. E. Broadbent: Perception and communication, Pergamon Press, 1958

[8] J. O. Cadenas, R. Simon Sherratt, D. Howlett, C. G. Guy, K. O. Lundqvist: Virtualization for Cost-Effective Teaching of Assembly Language Programming, IEEE Transactions on Education, 58(4) , pp. 282-288, 2015

[9] K. Chrysafiadi, M. Virvou: Dynamically Personalized E-Training in Computer Programming and the Language C, IEEE Transactions on Education, 56(4), pp. 385-392, 2013

[10] N. Dalal, B. Triggs: Histograms of oriented gradients for human detection, Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 8 pages, no pagination, 2005

[11] R. Dechter: Constraint Processing, Morgan Kaufmann, 2003

[12] M. M. Deza, E. Deza: Encyclopedia of Distances, Springer-Verlag, 2009

[13] J. A. Deutsch, D. Deutsch: Attention: Some theoretical considerations, Psychological Review, 70(1):80-90, 1963

[14] P. F. Felzenszwalb, D. P. Huttenlocher: Efficient Graph-Based Image Segmentation, International Journal of Computer Vision, 59(2):167-181, 2004

[15] P. Forczmański, A. Markiewicz: Two-stage approach to extracting visual objects from paper documents, Machine Vision and Applications, 27(8):1243-1257, 2016

[16] Ilona Heldal, Carsten Helgesen: The Digital HealthLab: Supporting Interdisciplinary Projects in Engineering and in Health Education, Journal of Applied Technical and Educational Sciences, Vol. 8, No. 4, 2018, pp. 4-21

[17] R. Gerdes, R. Otterbach, R. Kammüller: Fast and robust recognition and localization of 2-D objects, Machine Vision and Application, 8(6):365-374, 1995

[18] M. Gnjatović, N. Maček, S. Adamović: A Non-Connectionist Two-Stage Approach to Digit Recognition in the Presence of Noise, Proc. of the 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Naples, Italy, pp. 15-20, 2019

[19] R. C. Gonzalez, R. E. Woods: Digital Image Processing, 2nd Edition, Prentice Hall, 2002

[20] Y. Guo, S. Zhang, A. Ritter, H. Man: A Case Study on a Capsule Robot in the Gastrointestinal Tract to Teach Robot Programming and Navigation, IEEE Transactions on Education, 57(2), pp. 112-121, 2014

[21] I. Horváth: The Edu-coaching Method in the Service of Efficient Teaching of Disruptive Technologies, In: R. Klempous, J. Nikodem, P. Baranyi (eds) Cognitive Infocommunications, Theory and Applications, Springer, Cham, pp. 349-363, 2019

[22] I. Horváth: Innovative engineering education in the cooperative VR environment, Proc. of the 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Wroclaw, Poland, pp. 359-364, 2016

[23] International Telecommunication Union: Recommendation ITU-R BT.709-6 (06/2015): Parameter values for the HDTV standards for production and international programme exchange, International Telecommunication Union – ITU Radiocommunication Sector, [Online] Available: www.itu.int/rec/R-REC-BT.709-6-201506-I/en, [Accessed: 26-May-2019]

[24] D. Kahneman: Attention and effort, Englewood Cliffs, Prentice Hall, 1973

[25] C. Larman, V. R. Basili: Iterative and incremental developments: a brief history, Computer, IEEE, 36(6):47-56, 2003

[26] Y. LeCun, Y. Bengio, Ge. Hinton: Deep learning, Nature, 521, pp. 436-444, 2015

[27] S. Z. Li, J. Hornegger: A two-stage probabilistic approach for object recognition, In: H. Burkhardt, B. Neumann (eds) Computer Vision – ECCV'98, Lecture Notes in Computer Science, Vol. 1407, Springer, Berlin, Heidelberg, pp. 733-747, 1998

[28] H-K. Lu; P-C Lin: Effects of interactivity on students' intention to use simulation-based learning tool in computer networking education, Proc. of the 14th International Conference on Advanced Communication Technology (ICACT), 4 pages, no pagination, 2012

[29] O. Matei, P. C. Pop, H. Vălean: A Robust Approach to Digit Recognition in Noisy Environments, In: H. Jiang, W. Ding, M. Ali, X. Wu (eds) Advanced Research in Applied Artificial Intelligence. IEA/AIE 2012, LNCS, Vol. 7345, Springer, Berlin, Heidelberg, pp. 606-615, 2012

[30] T. Mitsui, H. Fujiyoshi: Object detection by joint features based on two-stage boosting, Proc. of the 12th International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1169-1176, 2009

[31] U. Neisser: Cognitive psychology, NY: Appleton-Century-Crofts, 1967

[32] C. Pohl, A. Kiesel, W. Kunde, J. Hoffmann: Early and late selection in unconscious information processing, Journal of Experimental Psychology, 36(2):268-285, 2010

[33] F. Y. Shih: Image Processing and Pattern Recognition: Fundamentals and Techniques, Wiley-IEEE Press, 2010

[34] T. Ujbanyi, G. Sziladi, J. Katona, A. Kovari: Pilot Application of Eye-Tracking to Analyze a Computer Exam Test, In: R. Klempous, J. Nikodem, P. Baranyi (eds) Cognitive Infocommunications, Theory and Applications, Springer, Cham, pp. 329-347, 2019

[35] A. H. Van der Heijden, R. Hagenaar, W. Bloem: Two stages in postcategorical filtering and selection, Memory & Cognition, 12 (5):458-469, 1984

[36] J. Villalobos, N. Calderón, C. Jiménez: Developing programming skills by using interactive learning objects, Proc. of the 14th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ACM SIGCSE Bulletin - ITiCSE '09, 41(3), pp. 151-155, 2009

[37] X-S Yang: Nature-Inspired Optimization Algorithms, Elsevier, 2014

[38] A. M. Zador: A critique of pure learning and what artificial neural networks can learn from animal brains, Nature Communications, 10:3770, 7 pages, no pagination 2019