Recognition of distorted characters printed on metal using fuzzy logic methods

Vilém Novák, Petr Hurtík, Hashim Habiballa
Centre of Excellence IT4Innovations, division of the University of Ostrava
Institute for Research and Applications of Fuzzy Modeling,
30. dubna 22, 701 03 Ostrava 1, Czech Republic
E-mail: {Vilem.Novak,Petr.Hurtik,Hashim.Habiballa}@osu.cz

***Abstract*—This paper is based on real application whose task was to recognize characters printed on metal ingots. The problem is that the surface of ingots is very uneven — ingots are either hot, or cut by rough instrument, the printing machine can be worn down, etc. Therefore, the first difficulty was to separate characters from the background and the second one was the fact that the separated characters are heavily distorted. It is a challenge to recognize the letters correctly. In this paper, we will present two approaches: recognition based on application of mathematical fuzzy logic and a special human behavior inspired approach.**

I. Introduction

Character recognition is a unique ability of human brain that is trained from childhood and continuously improved in changing environment. Though functioning of the human brain is not precisely known, it seems that a given pattern is associated with its meaning so that if an unknown character is figured out then people are able to guess its meaning and to find the the most similar one.

The problem of automatic recognition of characters has been solved already for more than 100 years (for example, the OCR system was proposed already 1914 by Emanuel Goldberg, cf. [1]). Recognition is usually based on comparison of a given character or word with a dictionary for a specified language defined before. Another approach called “Completely Automated Public Turing test to tell Computers and Humans Apart” (CAPTCHA) is based on a human recognition of a damaged character due to a computer script and it is usually used for security web forms. Quality CAPTCHA solvers achieves about 60-100 % of successful reading for precisely defined type or set of types.

In this paper, we describe our solution of the character recognition problem for the case when characters are printed on metal surface that is very uneven — ingots with the printed characters are either hot, or cut by rough instrument, the printing machine can be worn down, etc. Consequently, the characters are usually distorted in high degree and moreover, it is difficult to distinguish them from the background. Therefore, very robust methods are needed. We present results of two methods: the methods based on application of mathematical fuzzy logic and a special method that we call “human behavior inspired approach”.

This paper was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

II. Problem definition

Our task was to recognize code printed both on hot as well as on cold metal ingots. The code consists of characters from the standard alphanumeric alphabet and is printed in two lines consisting of up to eight characters. The characters can be mutually shifted, or distorted. Dimension (width and height) of each character is known with a certain small tolerance. Image resolution is the same for one ingot type. Fig 1 contains examples of both cases.

We face two problems in this task. First, to separate the code from its background and secnd, to recognize the characters correctly because of the fact that they can be very much distorted.

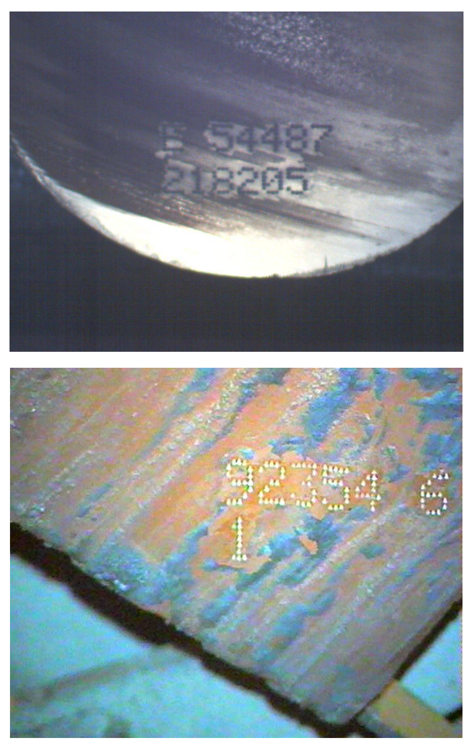

Fig. 1. Example of code printed on metal ingots. Up: Hot ingot. Down: Cold ingot.

We applied two methods for code recognition. The first one is based on the theory of mathematical fuzzy logic and is very robust, so that even very distorted characters can be correctly recognized. This method is convenient especially for cold ingots. The second one, developed especially for hot ingots, makes it possible to identify the placement of the code on the ingot and then to recognize correctly all the characters.

Let us also mention that we actually tried also third recognition method using neural networks with one inner layer ([2]). The results, however, were not convincing especially due to the problem with separation of characters from their background and insufficient number of learning patterns.

III. Recognition method based on mathematical fuzzy logic

The first method is based on the theory of mathematical fuzzy logic with evaluated syntax $\mathrm{Ev}_{Ł}$ presented in detail in [3]. The idea is to split the image into a grid of parts and to characterize each part both by its content as well as intensity. The content is characterized by a certain logical formula whose meaning can be, for example, that the given part is "a house, a pipe, a tree, a branch, a steep curve", etc. or, of course, simply "a pixel". This opens the possibility to evaluate similarity of images on the basis of content of its parts. Since the the basic concept of $\mathrm{Ev}_{Ł}$ is that of *evaluated formula* a/A where A is a formula and a is its evaluation being element of the algebra of truth degrees, we have a natural means at disposal which can be used for the described task; simply A represents the content of the given part of the image and a is its degree of truth. The procedure was first published in [4] as a method using which we can tentatively recognize in some sense "suspicious" images that could later on be passed to a more detailed analysis using other sophisticated methods. In this paper we present a simplified application which that is very effective in recognition of even severely distorted letters.

A. Fundamental notions

First, we consider a special language J of first-order $\mathrm{Ev}_{Ł}$. We suppose that it contains a sufficient number of terms (constants) $t_{i,j}$ which will represent *locations* in the two-dimensional space (i.e., selected parts of the image). Each location can be whatever part of the image, including a single pixel or a larger region of the image. The two-dimensional space will be represented by matrices of terms taken from the set of closed terms M_V:

$$M = (t_{i,j})_{\substack{i\in I\\ j\in J}} = \begin{pmatrix} t_{11} & \dots & t_{1n} \\ \vdots & \vdots & \vdots \\ t_{m1} & \dots & t_{mn} \end{pmatrix} \tag{1}$$

where $I = \{1, \dots, m\}$ and $J = \{1, \dots, n\}$ are some index sets. The matrix (1) will be called the *frame of the pattern*. In other words, the frame of the pattern is the underlying grid of parts of the given image. The pattern itself is the letter which we suppose to be contained in the image and which is to be recognized. A vector $t_i^L = (t_{i1}, \dots, t_{in})$ is a *line* of the frame M and $t_j^C = (t_{1j}, \dots, t_{mj})$ is a *column* of the frame M. The simplest content of the location is the *pixel* since pixels are points of which images are formed. A *pixel* is represented by a certain designated (and fixed) atomic formula $P(x)$ where the variable x can be replaced by terms from (1), i.e., it runs over the locations. Another special designated formula is $N(x)$. It will represent "nothing" or also "empty space". We put $N(x) := \mathbf{0}$. The algebra of truth values is the standard Łukasiwicz MV-algebra

$$\mathscr{L} = \langle [0,1], \vee, \wedge, \otimes, \to, 0, 1\rangle \tag{2}$$

where

$$\begin{aligned} &\wedge = \text{minimum}, && \vee = \text{maximum}, \\ &a \otimes b = \max(0, a+b-1), && a \to b = \min(1, 1-a+b), \\ &\neg a = a \to 0 = 1-a. \end{aligned}$$

Formulas of the language J are *properties of the given location* (its content) in the space. They can represent whatever shape, e.g., circles, rectangles, hand-drawn curves, etc. As mentioned, the main concept in the formal theory is that of *evaluated formula*. It is a couple a/A where A is a formula and $a \in [0,1]$ is a syntactic truth value. In connection with the analysis of images, we will usually call a *intensity* of the formula A. Note that $0/\mathbf{0}$, i.e., "nothing" has always the intensity 0. Let M_Γ be a frame. The *pattern* Γ is a matrix of evaluated formulas

$$\Gamma = \left(a_{ij}/A_x[t_{ij}]\right)_{\substack{i\in I_{M_\Gamma}\\ j\in J_{M_\Gamma}}} \tag{3}$$

where $A(x) \in \Sigma(x)$, $t_{ij} \in M_\Gamma$ and I_{M_Γ}, J_{M_Γ} are index sets of terms taken from the frame M_Γ. A *horizontal component* of the pattern Γ is

$$\Lambda_i^H = \left(a_{ij}/A_x[t_{ij}] \in \Gamma \mid t_{ij} \in t_i^L\right), \qquad j \in J_{M_\Gamma} \tag{4}$$

where t_i^L is a line of M_Γ. Similarly, a *vertical component* of the pattern Γ is

$$\Lambda_j^V = \left(a_{ij}/A_x[t_{ij}] \in \Gamma \mid t_{ij} \in t_j^C\right), \qquad i \in I_{M_\Gamma} \tag{5}$$

where t_i^C is a column of M_Γ. If the direction does not matter, we will simply talk about *component*. The *empty component* is $E = (0/\mathbf{0}, \dots, 0/\mathbf{0})$. Hence, a component is a vertical or horizontal line selected in the picture which consists of some well defined elements represented by formulas. *Dimension* of the component $\Lambda = (a_1/A_1, \dots, a_n/A_n)$ is $\dim(\Lambda) = k^{max} - k^{min} + 1$, $1 \le k^{min}, k^{max} \le n$, where

$$k^{min} = \min_{i=1,\dots,n}\{i \mid \vdash_a \neg(A_i(x) \Leftrightarrow \mathbf{0}), a > 0\}, \tag{6}$$

$$k^{max} = \max_{i=1,\dots,n}\{i \mid \vdash_a \neg(A_i(x) \Leftrightarrow \mathbf{0}), a > 0\} \tag{7}$$

This definition takes into account that there may be "holes" in the component which, however, should be considered to contribute to the component itself and hence to its dimension. On the other hand, the outer empty parts of the component are *not* included in the computation of its dimension. A pattern is *horizontally (vertically) normalized* if Λ_1^H (Λ_1^V) contains at least one evaluated formula a/A such that $\vdash \neg(A \Leftrightarrow \mathbf{0})$. A pattern is *normalized* if it is both horizontally and vertically normalized. This definition means that there is at least one location in the given line where the truth evaluation of the respective formula A is equal to 1 (for example, at least one pixel with the intensity 1). *Intensity* of a pattern is the matrix

$$Y_\Gamma = (a_{ij})_{\substack{i\in I_{M_\Gamma}\\ j\in J_{M_\Gamma}}} . \tag{8}$$

The intensity $Y_{\Lambda_i^H}$ ($Y_{\Lambda_j^V}$) of a horizontal (vertical) component is defined analogously. Intensity of a pattern Γ is said to be *normal* if $\check{H}_\Gamma = 1$. The pattern with normal intensity is called *normal*.

B. Comparison of patterns

Let us consider two patterns

$$\Gamma = \left({}^{a_{ij}}/A_x[t_{ij}]\right)_{\substack{i\in I_{M_\Gamma}\\ j\in J_{M_\Gamma}}}$$

and

$$\Gamma' = \left({}^{a'_{ij}}/A'_x[t_{ij}]\right)_{\substack{i\in I_{M_{\Gamma'}}\\ j\in J_{M_{\Gamma'}}}} .$$

Without loss of generality, we will assume in the sequel that $I_{M_\Gamma} = I_{M_{\Gamma'}}$ and $J_{M_\Gamma} = J_{M_{\Gamma'}}$. The patterns will be compared both according to the content as well as intensity of the corresponding locations. Hence, we will consider a bijection $f : M_\Gamma \longrightarrow M_{\Gamma'}$ $f(t_{ij}) = t'_{ij}$, $i \in I, j \in J$ between the frames M_Γ and $M_{\Gamma'}$. Let two components, $\Lambda = ({}^{a_1}/A_1, \dots, {}^{a_n}/A_n)$ and $\Lambda' = ({}^{a'_1}/A'_1, \dots, {}^{a'_n}/A'_n)$ be given. Put $K_1 = \min(k^{min}, k'^{min})$ and $K_2 = \max(k^{max}, k'^{max})$ where k^{min}, k'^{min} are the corresponding indices defined in (6) and k^{max}, k'^{max} are those defined in (7), respectively. Note that K_1 and K_2 are the left-most and right-most indices of some nonempty place which occurs in either of the two compared patterns in the direction of the given components. Furthermore, we put

$$\begin{aligned} n^C = \sum\{ & b_i \mid \vdash_{b_i} A_i(x) \Leftrightarrow A'_i(x), {}^{a}/A_{i,x}[t] \in \Lambda, \\ & {}^{a'}/A'_{i,x}[f(t)] \in \Lambda', K_1 \le i \le K_2\}, \end{aligned} \tag{9}$$

$$\begin{aligned} n^I = \sum\{ & b_i = a_i \leftrightarrow a'_i \mid {}^{a_i}/A_{i,x}[t] \in \Lambda, {}^{a'_i}/A'_{i,x}[f(t)] \in \Lambda', \\ & K_1 \le i \le K_2\}. \end{aligned} \tag{10}$$

The number n^C represents the total degree in which the corresponding places in both patterns tally *in the content*. This extends the power of the procedure as from the formal point of view, A'_i may differ from A_i but they still may represent the same object; at least to some degree b_i. The number n^I is similar but it reflects the compared intensity of the objects residing in the respective locations. The components Λ and Λ' are said to *tally in the degree* q if

$$q = \begin{cases} \frac{n^C + n^I}{2(K_2 - K_1 + 1)} & \text{if } K_2 - K_1 + 1 > 0 \\ 1 & \text{otherwise.} \end{cases} \tag{11}$$

We will write

$$\Lambda \approx_q \Lambda'$$

to denote that two components Λ and Λ' tally in the degree q. When $q = 1$ then the subscript q will be omitted. It can be demonstrated that if all formulas A in ${}^{a}/A$, for which it holds that $\vdash \neg(A \Leftrightarrow \mathbf{0})$ and which occur in Λ and Λ', are the same then we can compute q using the following formula:

$$q = 1 - \frac{1}{2} \frac{\sum_{K_1 \le i \le K_2} |a_i - a'_i|}{K_2 - K_1 + 1}. \tag{12}$$

It can be proved that the components Λ and Λ' tally in the degree 1 iff they are the same, i.e. their content as well as their intensity coincide. The pattern Γ can be viewed in two ways:

(a) From the *horizontal view*, i.e., as consisting of horizontal components

$$\Gamma = \left(\Lambda_i^H\right)_{i\in I} = (\Lambda_1^H, \dots, \Lambda_m^H). \tag{13}$$

(b) From the *vertical view*, i.e., as consisting of vertical components

$$\Gamma = \left(\Lambda_j^V\right)_{j\in J} = (\Lambda_1^V, \dots, \Lambda_n^V). \tag{14}$$

If the distinction between horizontal and vertical view of the pattern is inessential, we will simply use the term *pattern* in the sequel. A *subpattern* (horizontal or vertical) $\Delta \subseteq \Gamma$ of $\Gamma = (\Lambda_1, \dots, \Lambda_p)$ is any connected sequence

$$\Delta = (\Lambda_{j_1}, \dots, \Lambda_{j_k}), \qquad 1 \le j_1, j_k \le p \tag{15}$$

of components (horizontal or vertical, respectively) from Γ. If $\Lambda_{j_1} \ne E$ and $\Lambda_{j_k} \ne E$ then Δ is a *bare subpattern* of Γ and the number k is its *dimension*. Δ is a *maximal bare subpattern* of Γ if $\bar{\Delta} \subseteq \Delta$ for every bare subpattern $\bar{\Delta}$. The dimension of a maximal bare subpattern of Γ is the *dimension of* Γ and it will be denoted by $\dim(\Gamma)$. Recall from our previous agreement that, in fact, we distinguish horizontal ($\dim_H(\Gamma)$) or a vertical dimensions ($\dim_V(\Gamma)$) of the pattern depending on whether a pattern is viewed horizontally or vertically. Note that both dimensions are, in general, different.

Suppose now that two patterns Γ and Γ' are given and let $\Delta \subseteq \Gamma$ have a dimension k. Let q_0 be some threshold value of the the degree (11) (according to experiments, it is useful to set $q_0 \approx 0.7$). We say that Δ *occurs in* Γ' with the degree q_0 (is a q_0*-common subpattern* of both Γ and Γ') if there is a subpattern $\Delta' \subseteq \Gamma'$ of dimension k for which the property

$$\Lambda_i \approx_q \Lambda'_i, \qquad q \ge q_0 \tag{16}$$

holds for every pair of components from Δ and Δ', $i = 1, \dots, k$, respectively. If $q_0 = 1$, then Δ is a common subpattern of Γ and Γ'. A q_0-common subpattern of Γ and Γ' is *maximal* if every subpattern $\bar{\Delta} \subseteq \Gamma$ such that $\bar{\Delta} \supseteq \Delta$ is not a q_0-common subpattern of Γ and Γ'. It can be proved that different q_0-common subpatterns are disjoint.

Let $\Delta_1, \dots, \Delta_r$ be all maximal q_0-common subpatterns of Γ and Γ'. The *degree of matching* of Γ and Γ' is the number

$$\eta(q_0) = \frac{1}{2} \left(\frac{\sum_{j=1}^r \dim(\Delta_j)}{\dim(\Gamma)} + \frac{\sum_{j=1}^r \dim(\Delta_j)}{\dim(\Gamma')} \right) \tag{17}$$

where $\dim(\Delta_j) \ge 2$ for all $j = 1, \dots, r$. We must compute (17) separately for horizontal and vertical view so that we obtain horizontal η_H as well as vertical η_V degrees of matching, respectively. Then the *total degree of matching* of the patterns Γ and Γ' is the number

$$\bar{\eta}(q_0) = \frac{\eta_H(q_0) + \eta_V(q_0)}{2}. \quad (18)$$

We can prove that two patterns match in the degree 1 iff they are the same, i.e., that

$$1 = \bar{\eta}(1) = \eta_H(1) = \eta_V(1) \quad \text{iff} \quad \vdash A_i \Leftrightarrow A'_i \text{ and } a_i = a'_i$$

holds for every $a_i/A_{i,x}[t] \in \Gamma$ and $a'_i/A'_{i,x}[t] \in \Gamma'$.

IV. Human behavior inspired recognition method

Because human brain is able to localize and recognize characters in the picture, we can think about the way, how does this process work? According to our tests it seems that it is a sequence of subprocesses: first we focus on a small area of the whole image, verify that it contains characters and finally, we recognize the character. Similar idea is used in SIFT [?] where interesting location is detected and then, comparison with patterns is computed. However, implementation of this technique did not bring expected results.

In this section, we describe a simple way how this process can be mimicked. An important criterion that we, at the same time, had to fulfill was simplicity so that implementation of the method can work very fast. The method described below was developed especially for recognition of characters (and the whole code) printed on hot ingot.

The image is represented by a function $f : D \longrightarrow [0,1]$ where $D = \{1,\dots,W\} \times \{1,\dots,H\}$ is the domain of image for some width W and height H. The recognition process consist of three steps: (1) preprocessing, i.e., we begin with f and transform it into a new function f' whose properties are more suitable for further processing; (2) location of the area of interest; (3) character extraction and recognition.

A. Preprocessing of the image

Because colors of characters and background can have various combination of their hue, saturation, lightness or they even have the same color to each other, we will form a new image function f' for which all these problems are reduces and which is partially independent on the color.

Formation of the new image function consist of the following steps:

1) *Forming a grayscale image:* This is done using the common approach[*)] by setting:

$$f_{gray}(x,y) = \frac{1}{32}(11 f_{red}(x,y) + 16 f_{green}(x,y) + \\ + 5 f_{blue}(x,y)), \qquad (x,y) \in D.$$

It can be demonstrated that we can obtain good results by using only red channel, the subsequent steps proceed over this this channel only.

2) *Histogram adjustment:* Each intensity value n_j in the histogram is replaced by a new intensity value $n'_j = \frac{1}{W \cdot H}\sum_{i=0}^{j} n_i$.

The reason for this adjustment is the following: there can be areas in the hot ingot in which the intensity value is almost maximal and so, some important details can be lost. By histogram adjustment we can increase intensity steps in these areas.

3) *Standard deviation based voting:* Let us set a square mask $S \subset D$ of the dimension $(2p+1)\times(2p+1)$ with the center (x_0, y_0) where $p > 0$ is a small number. We compute a standard deviation over this mask:

$$c_{S,(x_0,y_0)} = \\ = \sqrt{\frac{\sum_{x=-p}^{p}\sum_{y=-p}^{p}(f(x_0+x, y_0+y) - a_{(x_0,y_0)})^2}{(2p+1)^2}} \quad (19)$$

where $a_{(x_0,y_0)} = \frac{\sum_{x=-p}^{p}\sum_{y=-p}^{p} f(x_0+x,y_0+y)}{(2p+1)^2}$ is the arithmetic mean. Then the transformed image function becomes

$$f'(x,y) = \max\{f(x,y) - 2c_{S,(x,y)}, 0\}, \qquad (x,y) \in D.$$

In our case, we use the deviation based voting twice, with different radius $p = 1$ and $p = 2$.

4) *Convolution:* since the character can be subject to noise and distortion, we need more intensive blurring. Therefore, we apply convolution with the standard arithmetic matrix (instead of the Gauss kernel).

B. Finding area containing the characters

To find the area containing characters, we utilize the information that the characters are organized in two lines with fixed size and number of characters in each line. Hence, the search area $E_{(x,y)} \subset D$ is defined as two dimensional subspace of D with the dimension $(W' \times H')$ and center (x, y).

Furthermore, values of the function $f'(x,y)$ when considering only pixels inside characters apparently have higher standard deviation than that for pixels in the background. Therefore, we will compute a ratio r between the "dark" (background) pixels and the other ones in area E as follows:

$$r(E) = \frac{\#\{(x,y) \in E \mid f'(x,y) < T\}}{W' \cdot H'}$$

for a certain threshold T defined for the dark color. Then, since the dimensions W' and H' are known, we can determine the area E as such area for which

$$c_{E,(x,y)} + \tau \cdot r(E)$$

is maximal, where $c_{E,(x,y)}$ is the standard deviation (19) over the area E and τ is a certain normalization coefficient. Let us remark that this method is very fast.

C. Preprocessing of the area E

After finding the area E, we must again realize its preprocessing, i.e., we must realize the following steps:

1) Local histogram adjustment: the same step as in Subsection IV-A.

[*)] See, e.g., http://doc.qt.digia.com/stable/qcolor.html

2) Removing background: all pixels with the sum of intensity values smaller than a given threshold are removed.
3) Local histogram adjustment in cells of rows.

D. Character recognition

The third step of the human behavior inspired recognition algorithm is the proper character recognition. We can apply the logical method described in Section III. However, we faced in solution of the real problem the necessity to realize the character recognition in less than 1 second on a common PC. Since the previous steps are still quite computationally expensive, the recognition must be very quick. Therefore, we proposed the following simplified method.

Each character C is decomposed into vertical and horizontal lines L, so $C = \{L_1, \ldots, L_n\}$. Each line L_i, $i = 1, \ldots, n$ is a set of pixels P, $L = \{P_1, \ldots, P_m\}$. For each line we compute two values: α is coordinate of the beginning of the given line and β is its ending. Both values are determined with respect to certain thresholds of intensity determining difference in intensity of neighboring pixels and background. Then we determine the degree of truth that we have a given character using the formula

$$P(C) = \frac{\sum_{i=1}^{n}(\alpha_i - \beta_i)}{\sum_{i=1}^{n} \#\{L_i\}}$$

where $\#\{L_i\}$ is number of pixels forming the line L_i. We choose character with the highest value of $P(C)$.

V. Results

A. Hot ingot

We tested our methods on 27 images of characters printed on hot ingot. Each image consists of 12 characters and so, we had a set of 324 characters at disposal. For comparison, we applied also the well known OCR software†). Both of them failed, because they were not able to separate characters from the background and to recognize them. Then we tested our human behavior inspired algorithm. According to a ratio $R = \frac{true\ recognized}{all\ characters} \times 100$ we obtained 95.6 % of successful recognition. This means that only in three of 27 images our recognition failed.

Another available software is CAPTCHA. These are various kinds of algorithms that, however, work only for small image area containing characters and so, they were not able to find character area in a bigger image.

To test robustness of the behavior inspired algorithm we take an easy-readable image and different types of distortion to it — see Figure 3. The R ratio is 100 % on each image shown below:

The algorithm is very fast. For comparison, using notebook Dell x13 with Intel i7-2637@1.7 GHz processor, the average computing time to obtain 1 result using the human behavior inspired algorithm is 0.7 sec. This includes all preprocessing steps, displaying and logging results and about 50,000 character comparisons with predefined patterns. The system should be soon put to commercial operation.

†)SimpleOCR [TODO] and TopOCR[TODO].

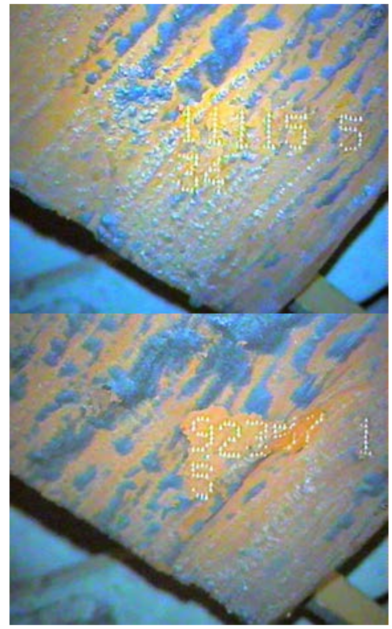

Fig. 4. Example of characters printed on cold ingot

B. Cold ingot

As mentioned in the introduction, part of our task was to recognize characters printed on cold ingot. Examples of the images are in Fig. 3. The problem here is that it is very difficult to distinguish the characters from the background. Therefore, after all the characters become very much distorted. Therefore, we had to use the fuzzy logic method described in Section III because this technique turned out to be very robust. Example

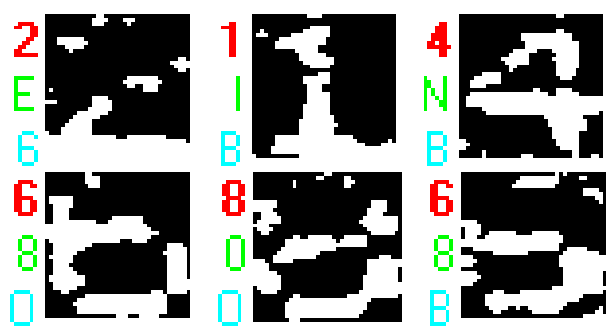

Fig. 5. Example of robustness the fuzzy logic technique. On the left of each image are three best recognized letters.

demonstrating how characters look like after their separation is in Fig. 5. This figure also demonstrates robustness of our technique.

For testing, we had 39 images at disposal with 7 or 8 characters on each image. That is, we had 302 characters in total. The success rate of the recognition using fuzzy logic was 80 %. Improvement of this result depends on effectiveness of the method for separation of characters from the background. This problem is the focus of further research.

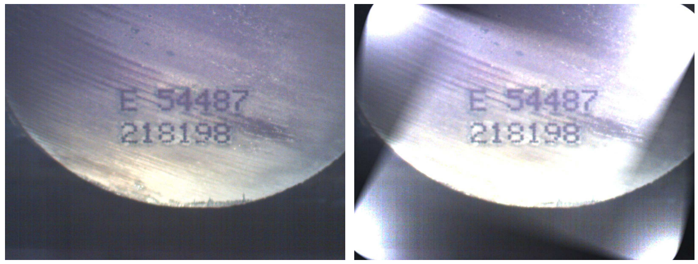

Fig. 2. Left: Original image; Right: The same image in different light

Fig. 3. Left: Image damaged by hand painting; Center: Damaged characters. Right: Blurred image

VI. Conclusion

In this paper, we presented results of solving practical task contracted us by a Czech company. The task was to recognize characters printed both on hot as well as cold metal ingots. We faced two problems: The first problem was to separate characters from their background and the second problem was to recognize them correctly. It was very difficult to distinguish the background from the characters because the characters on hot ingots have color very similar to the color of the background. The surface of cold ingots, on the other hand, is very uneven, damages by scratches of various size. Moreover, the combination of possible wear of the printing machine and unevenness of the surface of ingots caused that the separated characters can be distorted in high degree. Therefore, we had to find efficient methods for separation of the background that is, at the same time, very robust. This led us to develop an application on the theory of fuzzy logic. The results are quite successful and are going to be implemented in the practice.

References

[1] K. Singh and A. Bhasin, "A survey of OCR applications," *Int, J. of Machine Learning and Computing*, vol. 3, pp. 314–318, 2012.

[2] R. Rojas, *Neural Networks – A Systematic Introduction*. Berlin: Springer, 1996.

[3] V. Novák, I. Perfilieva, and J. Močkoř, *Mathematical Principles of Fuzzy Logic*. Boston: Kluwer, 1999.

[4] V. Novák, A. Zorat, and M. Fedrizzi, "A simple procedure for pattern prerecoginition based on fuzzy logic analysis," *Int. J. of Uncertainty, Fuzziness and Knowledge-Based systems*, vol. 5, pp. 31–45, 1997.