The Use of Deep Learning for Sticker Detection During Continuous Casting*

Artur Faizullin, Mikhail Zymbler
South Ural State University
Chelyabinsk, Russia

Dirk Lieftucht, Felix Fanghänel
SMS group GmbH
Düsseldorf, Germany

Breakout is the most expensive and dangerous issue of continuous casting, which causes loss of production time and significant yield penalties. The common cause of breakout is sticker, that is a part of strand shell, which adheres to a mold surface. Stickers can be detected by a temperature pattern in a mold heat-map. SMS group GmbH (Germany) develops HD mold, a cyber-physical system for sticker detection by monitoring and analysis of the temperature data from the fiber optical sensors during casting. Currently, HD mold exploits an analytical sticker detection algorithm that gives a large number of false alarms. This leads to significant loss time and quality overheads. We design Convolutional Neural Network (CNN) that recognizes sticker pattern and can be employed as a full-fledged substitute or an assistant of the current algorithm. Experiments show that being an assistant, CNN reduces the number of false alarms of the current algorithm by 47%.

Keywords—continuous casting; breakout prevention; sticker detection; machine learning; convolutional neural network; HD mold

I. Introduction

Continuous casting of steel is a method of solidification processing, which now approaching to 90% of crude steel output in the world [1]. During the casting, the liquid steel flows via preheated submerged nozzles into the water-cooled copper mold. In the output, there is a solidified steel strand, which can be divided to plates of required length.

Breakout is the most expensive and dangerous problem of continuous casting, when red-hot steel poured out of a mold damaging a caster and interrupts casting process. In practice, typical breakout for a conventional slab caster leads to losses more than 250,000€ [10]. The most common cause of breakouts during casting are stickers. *Sticker* is a part of strand shell, which adheres to a mold surface.

Fig. 1 shows how sticker leads to breakout, if no actions are taken. By measuring the temperatures in the copper plates attached outside to the mold, those events can be detected beforehand to avoid breakouts.

SMS group GmbH (Germany) develops *HD mold^{FO}*, a cyber-physical system aiming to avoid such events by real time monitoring and analysis of the temperature data from the fiber optical sensors installed on a mold. Sticker can be detected by a temperature pattern in the mold heat-map [1]. Because of high cost of breakout, current algorithm tries to minimize sticker probability by signaling an alarm as much earlier as possible. In case of the sticker, it is necessary to slow down the casting process for a while to prevent breakout and then gradually increase casting speed to a nominal value. Thus, a large amount of false alarms leads to significant time and quality overheads during continuous casting [8]. In practice, one false alarm case costs more than 1,000€ [10].

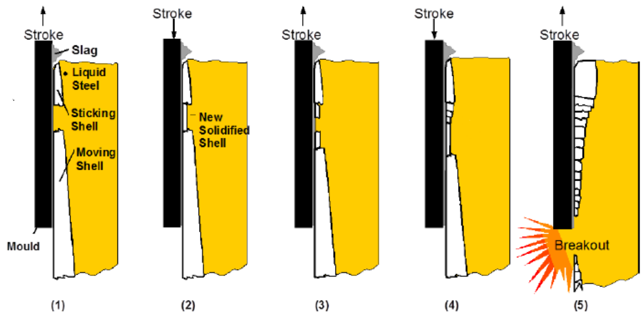

Fig. 1. Sticker breakout.

In this paper, we address the problem of reduction the number of false sticker alarms during casting. The paper makes the following contributions. We design Convolutional Neural Network (CNN), which recognizes sticker pattern in the mold heat-map. We implement CNN in MATLAB as a part of the Sticker Detection System (SDS). SDS can be employed in HD mold^{FO} as a full-fledged substitute or an assistant of the current algorithm. The latter means that SDS starts after the sticker warning from the current algorithm, and analyzes only suspicious situations. We evaluate our development on a real data from archives of SMS group GmbH. Experiments showed that having played a role of an assistant of the current algorithm, SDS reduces the number of false alarms by 47%.

The rest of the paper is organized as follows. Sect. II describes HD mold^{FO} and project background. In Sect. III, we present design and implementation of SDS. We give the results of the experimental evaluation of the developed approach in Sect. IV. Sect. V briefly discusses related works. Finally, in Sect. VI, we summarize the obtained results and draw directions for further research.

II. Background

Our development is a part of the HD mold^{FO} cyber-physical system [2]. HD mold^{FO} is the SMS group technology package, which uses fiber optical sensors installed in the caster copper

* The reported research was financially supported by SMS group GmbH (Germany), by Act 211 of the Government of the Russian Federation (contract No. 02.A03.21.0011), and by the Ministry of Education and Science of the Russian Federation (government order 2.7905.2017/8.9).

mold to provide fast, reliable feedback of real-time casting conditions. The HD mold^FO system provides a wide variety of models, which exploit the temperature feedback to assist the caster operator in troubleshooting problems and optimizing production. This assures increased plant availability, increased yield, protects the casting machine, enhances product quality, and frees the caster operator to focus their time on other important caster duties.

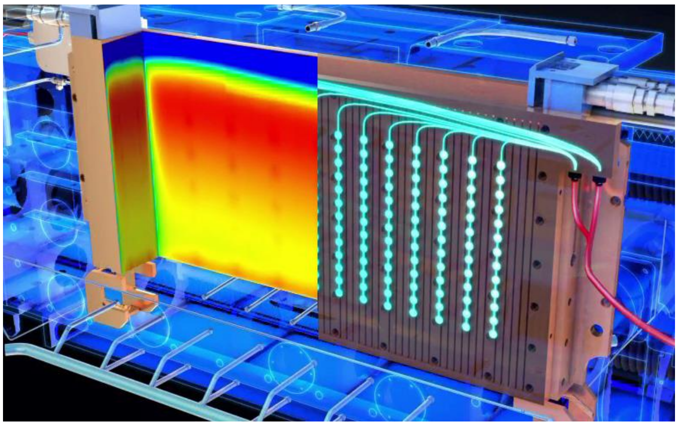

Fig. 2. Fiber optical sensors (blue points) on a broad side of a mold.

On a mold surface up to 576 fiber optical sensors are placed (cf. Fig. 2), which send temperature values every 0.25 second to a server in real time. Server is to add information on casting speed and set slowdown status where this status shows if casting speed was decreased because of true or false sticker. Then, server transmits *frames* (all the measured values associated with the same time moment) to HD mold^FO monitoring system. HD mold^FO makes decisions based on frames and information on mold size, positions, type and number of sensors.

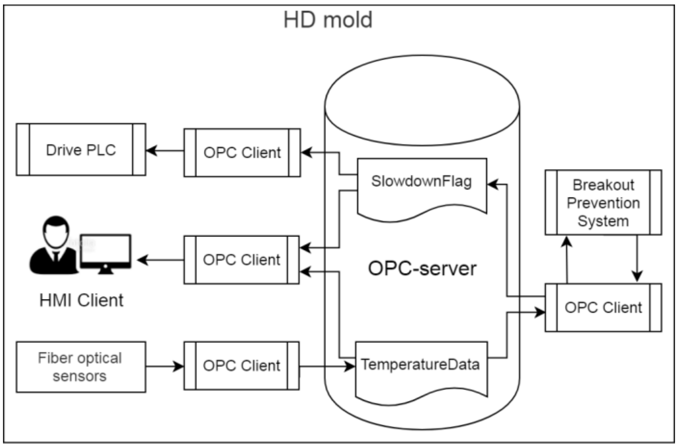

Fig. 3. Structure of the HD mold system.

Fig. 3 depicts the structure of HD mold^FO. *OPC-server* is responsible for data storage and communications between different modules through *OPC-clients*. *Fiber optical sensors* transmit measured values to OPC-server. The *Breakout Prevention System (BPS)* is the most important module of HD mold^FO. BPS evaluates temperature gradients in the mold to detect stickers and sets the *SlowDown flag* in that case. OPC-server sends alarm signal to *Drive PLC (Programming Logical Controller)*. Finally, the controller decreases the casting speed temporarily to let steel cool down and grow a new strand shell to close the gap. An operator monitors casting process including temperature values and the SlowDown flag through the *HMI Client*.

Typical sticker's temperature pattern can be detected as follows. Temperature sensors next to broken strand shell will show higher temperatures than their neighbours, which have more isolating strand shell in front.

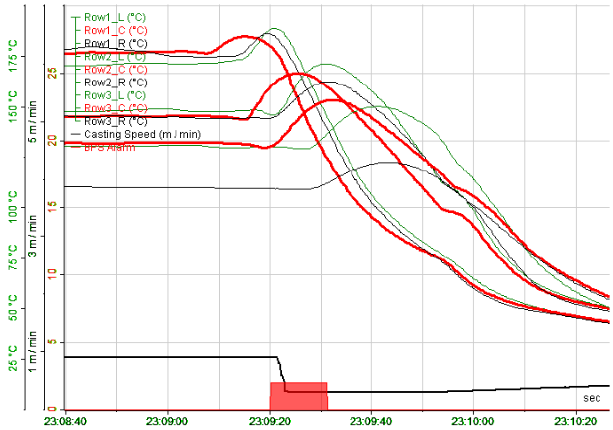

Fig. 4. Typical temperature pattern of the sticker.

Fig. 4 shows the temperatures of neighbourhood of 3 columns with 3 rows of sensors over time. Thick lines reflect the center column, which is the closest to the origin of the sticker. Thin lines reflect neighbouring columns. At the bottom of the graph, the line shows casting speed, and the rectangle highlights the moment, when the sticker was detected and casting speed was decreased [5]. Stickers have one particular origin and can be identified as a hotspot traveling from top to bottom through the mold. Furthermore, propagation of the hotspot to the left and to the right can be identified, since the hotspot always reaches the neighbour columns a little bit later [7].

Current version of BPS exploits an analytical sticker detection algorithm, which implements an idea discussed above by a set of condition checks. This handles any suspicious situation as a potential sticker and generates warning. After warning, BPS runs a sequence of additional sub-algorithms where each sub-algorithm is to confirm or deny some typical predetermined sticker case. If a sticker is confirmed, BPS sets the SlowDown flag. However, despite the additional checks of sub-algorithms, such an approach leads to large number of false alarms where each false alarm decreases production both speed and quality. In this regard, we address the problem of reduction the amount of false alarms by embedding CNN for sticker detection into HD mold^FO.

III. Design and Implementation

In this research, we design and implement the *Sticker Detection System (SDS)*, which recognizes stickers using CNN. SDS may play one of the two possible roles, namely a full-fledged substitute or an assistant of BPS. The former means that BPS is inactive and SDS works all alone. The latter means that SDS starts after the sticker warning from BPS, and analyzes only suspicious situations. Fig. 5 depicts the SDS structure.

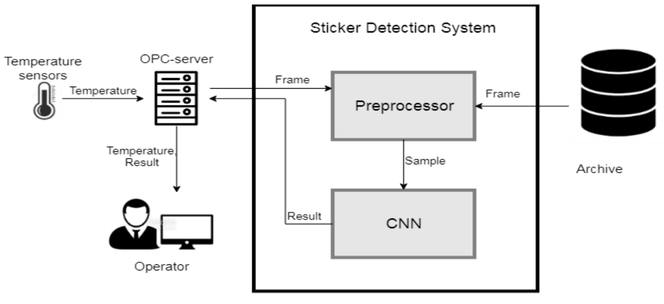

Fig. 5. Structure of the Sticker Detection System.

Preprocessor manipulates frames transmitted by OPC-server and prepares data for CNN input. *Archive* is supported by SMS group GmbH to accumulate frames of previous castings. SDS is implemented by means of the following software stack: MATLAB, Neural Network Toolbox and OPC ToolBox for MATLAB, ibaAnalyzer.

Further, we present CNN for sticker detection during continuous casting. Design and implementation of a CNN include the following typical steps: selection of the data to learn CNN, preprocessing of the data selected, data augmentation, and construction of CNN layers. We consider these matters below, in Subsect. A, B, C, and D, respectively. In addition, in Subsect. E, we describe a scenario of CNN application.

A. Data Selection

Since HD mold^{FO} used fiber optical sensors less than two years ago and exploits overly cautious breakout prevention algorithm, the SMS group GmbH archive contains data regarding pretty small number true and false stickers, namely frames corresponding to 14 real stickers and 103 false alarms, respectively. In order to construct the training set, we took frames corresponding to 11 real stickers and 88 false alarms. The rest frames (corresponding to 3 real stickers and 15 false alarms) were used to construct the test set.

As a sample of the training set and the test set, we consider F sequential frames instead of one frame where parameter F is chosen empirically. In this study, we took F=30, i.e. a sample corresponds to consecutive 7.5 seconds of the casting process. Thus, a sample is an array of size $F \times S$ where S is the number of mold sensors.

Next, to generate "sticker" samples of the training set, we choose *landmarks*, i.e. frames where the SlowDown flag was set (regardless of whether it was true or false alarm). Finally, to generate "not sticker" samples of the training set, we choose random frames that are far from landmarks. As a result of data selection, we have 11 "sticker" landmarks and 11,701 "not sticker" landmarks.

B. Data Preprocessing

Next, we preprocess the selected data through the following steps.

- *Data reshaping.* Since sensors on a mold are placed on rectangular area, we convert the 2-dimensional array to 3-dimensional tensor of $F \times H \times W$ form where H is the number of sensors in one column and W is the number of sensors in one row.
- *Data reduction.* Since stickers can appear in the top and in the middle part of a mold, in order to accelerate processing, we exclude the fruitless data of all the sensors from the lower third of the mold. Further, we work with array of size $F \times h \times W$ where $h = \left\lceil \frac{2}{3} \cdot H \right\rceil$.
- *Data cleaning.* Since some sensors on a mold can be damaged during the casting and transmit invalid signal of -273°C, we replace such a value with minimal valid value of temperature among current F frames.
- *Data normalization.* In order to improve prediction quality and convert all the data to the same interval, we use max-min normalization the data to bring all values to [0;1].
- *Data slicing.* At first, we slice the data into arrays with width w. Since typical sticker is of 3 sensor in width [7], we set $w = 5$, so as a sticker takes up most of a heat-map image. Next, we move a sliding window of length w along each array with step of one sensor and create a new sample at each step. This results in up to $W - w + 1$ samples from one landmark.

After the steps above, we get each sample as a 3-dimensional array of $F \times h \times w$ size containing real values from 0 to 1 to be used by our CNN. Finally, if a sample contains center of a real sticker, we mark the sample as "sticker", otherwise—as "not sticker".

C. Data Augmentation

By construction, our training set is imbalanced because of large number of "not sticker" samples. In order to increase the number of "sticker" samples, we generate artificial data by means of the following methods.

- *Sticker transferring.* For each frame of a sample, we calculate temperature difference with the first frame and store result as an array. Then we add this difference array to a "not sticker" sample and get a new "sticker" sample. Note that, this step is made before the data preprocessing.
- *Data biasing.* We consider only two-thirds of the rows of mold sensors. When a sticker appears, but not in the top row of sensors, we can create a sample for CNN with bias from the top row. For example, if a sticker appears at 4th row, we can generate three "sticker" samples from the 1st, 2nd, and 3rd row, and all the samples will include the case of real sticker.
- *Time biasing.* Typical sticker takes 12 seconds from its appearance to breakout [1]. We generated samples for each 0.5 second starting from the 1st to the 10th second after appearance of the sticker.
- *Data mirroring.* Since typical sticker has V-shape, "sticker" samples can be mirrored without loss of quality [6].

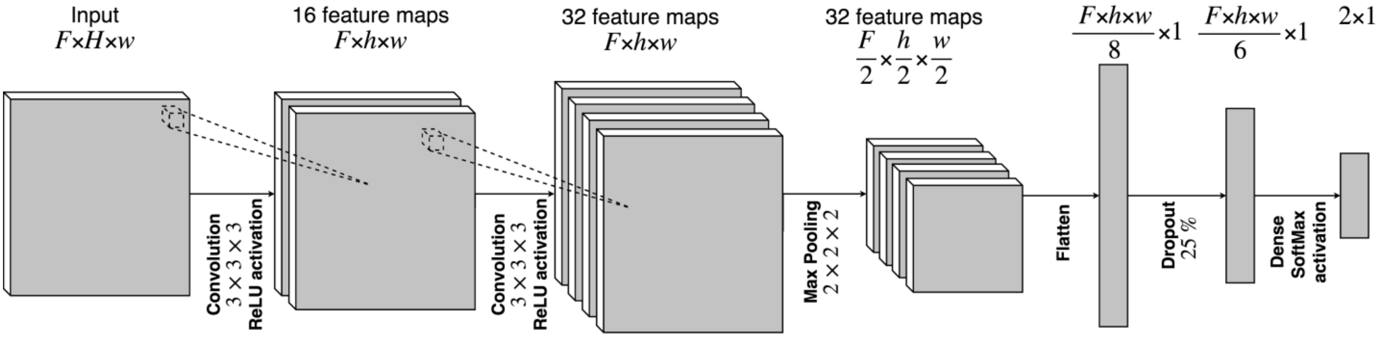

Fig. 6. CNN structure.

After the steps above, we increase the number of "sticker" samples to 6,318. Note that, we apply augmentation only to real stickers and false alarms, and not to samples of regular work because of their large amount, they do not need augmentation.

D. *Design of CNN*

We choose CNN as an approach for sticker detection during continuous casting. CNN is a deep neural network with convolutional layer and pooling layer in addition to standard hidden perceptron layers. These layers are not fully-connected and have significantly fewer parameters to learn. Thus, CNN learns faster in comparison with the traditional neural networks. In addition, convolutional layers allow for efficiently learning important image characteristics [9].

As an input, our CNN takes 3-dimensional array, which includes F preprocessed sequential frames. CNN outputs two positive real values giving a total of 1, which are probabilities of "sticker" and "not sticker" events for the given input.

Fig. 6 depicts the structure of the designed CNN. It consists of the input layer and output layer, and the following hidden layers.

- *Convolutional layers*. These two layers have similar structure and perform convolution of the input data. By doing so, we get a feature map. In most cases, sticker area includes three sensors in width and two sensors in height [7]. Thus, we chose 3×3×3 as a size of convolutional kernel. We use rectified linear unit (*ReLU*) as an activation function.
- *Max-pooling layer*. This layer reduces size of a feature map discarding areas without any key feature. In addition, this increases CNN's robustness to scaling. We chose pooling size 2×2×2 and 2 strides.
- *Flatten layer*. This layer transforms 3-dimensional tensor to 1-dimensional vector.
- *Dropout layer*. This layer randomly deactivates 25% of neurons to prevent overfitting of CNN.
- *Dense layer*. This is fully-connected layer, which applies *SoftMax* activation to input data and transforms it to two signals. Values of this signals represent probabilities of "sticker" and "not sticker" events.

We learned our CNN on the Tornado SUSU supercomputer node [4] using stochastic gradient descent with a batch size of 16 examples and binary cross-entropy as a loss function.

E. *Scenario of CNN Application*

SDS exploits CNN to process data in frame-by-frame manner according to the following scenario.

Firstly, SDS receives a frame with size $H \times W$ from the OPC-server. Then, by means of preprocessing, we transform the frame to the $W - \mathrm{w} + 1$ (at most) samples with size $F \times h \times w$. After that, for each sample CNN outputs the probability of "sticker" and "not sticker" event. Next, we compute mean value of all the probabilities of "sticker" event. Finally, if the mean value is greater than prespecified threshold α, then SDS generates sticker alert. Here α is a subject of empirical choice, and we show the respective experimental results below in Sect. IV.

IV. Experimental Evaluation

In the experiments, we study three approaches, namely BPS, SDS, and BPS+SDS. BPS implements an existent analytical algorithm. SDS is an approach when CNN substitutes BPS. Finally, BPS+SDS means that SDS works as an assistant of BPS. As a test set, we used the part of the archive remaining after learning, which corresponds to the 9,585 real cases of continuous casting consisting of the 3 sticker cases, 15 false alarms, and 9,567 examples of regular work.

For the aforementioned approaches, we compared the following characteristics: number of stickers detected, number of stickers missed, and number of false alarms signaled. Tab. I depicts the experimental results. Note that, for SDS, numbers of false alarms signaled are out of overall number of cases in the whole archive. At the same time, for each of the approaches BPS and BPS+SDS, number of false alarms is out of number of those cases of the archive where BPS either detected sticker or signaled false alarm.

We can see that each approach successfully recognizes stickers not missing even one. In false alarms, SDS performs worst. However, BPS+SDS performs best among all other approaches to the number of false alarms. Finally, combination of the existent analytical algorithm and CNN decreases number of false alarms by 47% in comparison with the existent system (i.e. by 7 out of 15 cases of false alarms).

TABLE I. COMPARSION OF APPROACHES

Characteristics	SDS	BPS	BPS+ SDS
Stickers detected (out of 9,585 cases in the whole archive)	3	3	3
Stickers missed (out of 3 cases of real sticker)	0	0	0
False alarms signaled (lower is better)	43*	15**	8**

* Out of 9,585 cases in the whole archive.
** Out of 18 cases: 3 real stickers and 15 real false alarms.

The preceding experimental results were obtained after an empirical research was carried out to choose optimal values of the α parameter (cf. Sect. III, Subsect. E). For the approaches SDS and BPS+SDS, the optimal values α_{SDS} and $\alpha_{BPS+SDS}$ were determined as follows.

We conducted experiments with the approaches SDS and BPS+SDS employing the test set above for various α values, and measured the numbers of stickers detected and false alarms. We choose the value of α as an optimal if it meets the following two requirements: 1) an approach must successfully recognize all stickers not missing even one, and 2) an approach must provide minimal number of false alarms among all values of α.

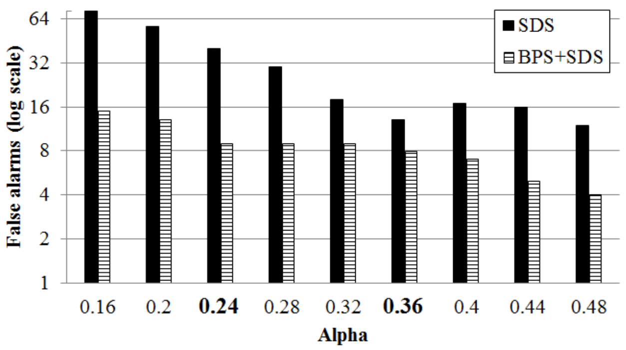

Fig. 7. Number of false alarms depending on the threshold $\boldsymbol{\alpha}$ (lower is better).

Experiments showed that SDS meet the first requirement only if $\alpha \leq 0.24$, and same that for the BPS+SDS approach if $\alpha \leq 0.36$. Then, according to Fig. 7, the approaches SDS and BPS+SDS meet the second requirement if $\alpha_{SDS} = 0.24$ and $\alpha_{BPS+SDS} = 0.36$, respectively.

V. RELATED WORKS

The problems of breakout prevention and sticker detection have been extensively studied in recent decade. Basic works include the following.

In [11], author describes the metallurgical background of defects in continuous slab casting, and gives the relation between the defect on the slab and the temperature, and heat flux signal. Conventional state of the art approaches to sticker detection and breakout prevention discussed in [13] in detail. This work also provides a description of rules and settings to detect sticker related breakouts. The effects of different casting parameters on sticker appearance are given in [7].

In [11], an adaptive approach to sticker detection is presented. In practice, a neural network (NN) based system is seldom applied successfully because it requires enough learning data, which is not given in the initial stage of the system [13]. In [12], two different neural networks exploiting temperature data from thermocouples are applied to sticker detection. Results in [12] look promising but necessity of a large amount of data to train NN limits the application of the approach in practice.

In this work, we showed how deep convolutional neural network can be successfully applied to the sticker detection problem with a lack of training data (what was done for the first time, to the best of our knowledge).

VI. CONCLUSIONS

Nowadays, continuous casting is the most commonly used technology of steel processing. Breakout is the most expensive and dangerous problem of continuous casting, which causes loss of production time and significant yield penalties. Typically, breakout occurs because of a sticker, a part of strand shell, which adheres to a mold surface. Stickers are detected by a temperature pattern in a mold heat-map.

Our study was carried out as a part of HD mold^{FO} cyber-physical system that detects stickers by real time monitoring and analysis of the temperature data from the fiber optical sensors installed on a mold. Current algorithm tries to minimize sticker probability by signaling an alarm as much earlier as possible. This leads to a large amount of false alarms with significant time and quality overheads during continuous casting.

In this paper, we presented an approach to sticker detection during continuous casting based on Convolutional Neural Network (CNN) to reduce amount of false sticker alarms. We described selection of the data to learn CNN, preprocessing of the data selected, data augmentation, construction of CNN layers. Such an approach was implemented as the Sticker Detection System (SDS) in MATLAB.

As a training sample, we took all the values measured by sensors for consecutive 7.5 seconds of the casting process. Next, we preprocessed the samples through the following steps, namely conversion of the 2-dimensional array to 3-dimensional tensor, exclusion of the fruitless data of all sensors from the lower third of a mold, replacement of invalid signals from damaged sensors, max-min normalization, and slicing to smaller parts to fit in sticker.

Since HD mold^{FO} starts less than two years ago and exploits overly cautious algorithm, we have pretty small number of real sticker cases to train our CNN. We employed sticker transferring, time biasing, data biasing, and data mirroring to generate extra artificial "sticker" samples and get significantly less imbalanced training set.

We designed CNN with the input layer and output layer, and five hidden layers, namely two Convolutional layers, Max-pooling layer, Flatten layer, Dropout layer, and Dense layer. We learned our CNN using stochastic gradient descent with a batch size of 16 examples and binary cross-entropy as a loss function.

In the experiments, we compared three following approaches. BPS is an analytical algorithm embedded into the HD mold^{FO} system. SDS is an approach when CNN works alone, and BPS is inactive. The BPS+SDS approach means that SDS analyzes only suspicious situations after the sticker warning from BPS. All the approaches successfully detected all stickers. The BPS+SDS approach decreases the number of false alarms of BPS by 47% and, thus, gives a saving of 47,000€ on every hundred of false alarms.

Our further studies might elaborate on the following topics: reducing number of false alarms of CNN to make it possible to use Sticker Detection System alone (without BPS), and development of CNN for detection of longitudinal face cracks in the steel during continuous casting.

ACKNOWLEDGMENT

Authors thank Alexey Porozov and Yana Kraeva (South Ural State University) for their valuable comments.

REFERENCES

[1] R. Bartos, S. Brockmann, R. Fandrich, G. Endemann, S. Heinzel, and C. Keul, "Steel Manual," Transl. P. Knighton, Verlag Stahleisen, Düsseldorf, 218 p., 2015.

[2] S. Bouhouche, M. Zoheir, Z. Slimane, and B. Jurgen, "Inferential sensor-based adaptive principal components analysis of mould bath level for breakout defect detection and evaluation in continuous casting," *Appl. Soft Comput. J.*, vol. 34, pp. 120-128, 2015.

[3] M. Korzi, M. Arzberger, F. Fanghänel, D. Lieftucht, M. Langer, and M. Reifferscheid, "HD mold - A wide variety of caster assistance tools for enhanced product quality,“ AISTech 2016 Iron and Steel Technology Conference, Pittsburgh, United States, 16-19 May 2016, vol. 3, pp. 3095-3103.

[4] P. Kostenetskiy and A. Safonov, "SUSU supercomputer resources," PCT'2016, International Scientific Conference on Parallel Computational Technologies, Arkhangelsk, Russia, March 29-31, 2016,. CEUR Workshop Proceedings. vol. 1576, pp. 561-573, 2016.

[5] A. Krasilnikov, F. Fanghänel, D. Lieftucht, M. Reifferscheid, and J. Laughlin, "Local heat transfer through mold flux film and optimal narrow face taper adjustment," AISTech, 2015.

[6] Y. Liu, X.D. Wang, F.M. Du, L.W. Kong, M. Yao, and X.B. Zhang. "Visual detection based on computer vision for sticker breakou in slab continuous casting“, Ironmaking and Steelmaking vol. 42, pp. 417-423, 2015.

[7] Y. Liu, X.D. Wang, F.M. Du, L.W. Kong, M. Yao, and X.B. Zhang. "Effects of casting parameters on sticker breakout and its propagation behaviour dusing slab continuous casting“, Ironmaking and Steelmaking vol. 41, pp. 748-755, 2014.

[8] M. Oquab, L. Bottou, I. Laptev, J. Sivic, "Leraning and transferring mid-level image representations using convolutional neural networks", in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1717-1724.

[9] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for image classification : a comprehensive review," Neural Comput., vol. 29, no. 9, pp. 2352-2449, 2017.

[10] SMS group, "HD mold. High definition – high performance," https://www.sms-group.com/sms-group/downloads/download-detail/22203/, no. A-345E, pp. 2, 2016.

[11] B.G. Thomas, "On-line Detection of Quality Problems in Continuous Casting of Steel," in *Materials Science and Technology 2003 Meeting*, 2003, pp. 29-45.

[12] B. Zhang, R. Zhang, G. Wang, L. Sun, Z. Zhang, and Q. Li, "Breakout prediction for continuous casting using genetic algorithm-based back propagation neural network model," *Int. J. Model. Identif. Control*, vol. 16, no. 3, pp. 199-205, 2012.

[13] F. He, D.-F. He, Z.-H. Deng, A.-J. Xu, and N.-Y. Tian "Development and application of mould breakout prediction system with online thermal map for steel continuous casting," in *Ironmaking and Steelmaking*, vol. 42, no. 3, 2015.