MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION
Federal State Autonomous Educational Institution of Higher Education
“South Ural State University (national research university)”
School of Electrical Engineering and Computer Science
Department of Computer Science

THESIS IS CHECKED

Reviewer,
Head of Component Development
__________ D. Lieftucht
“___”___________ 2019

ACCEPTED FOR THE DEFENSE

Head of the department, Dr. Sci., Prof.
__________ L.B. Sokolinsky
“___”___________ 2019

USING A CONVOLUTIONAL NEURAL NETWORK FOR STICKER DETECTION IN SLAB CONTINUOUS CASTING

GRADUATE QUALIFICATION WORK
SUSU–02.04.02.2019.308-154.GQW

Supervisor,
Assoc. Prof., Cand. Sci.
__________ M.L. Zymbler

Author,
student of the group CE-220
__________ A.R. Faizullin

Normative control
_____________ O.N. Ivanova
“___”___________ 2019

Chelyabinsk–2019

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION
Federal State Autonomous Educational Institution of High Education
“South Ural State University (National Research University)”
School of Electrical Engineering and Computer Science
Department of Computer Science

APPROVED

Head of the department,
Dr. Sci., Prof.

__________ L.B. Sokolinsky

“___”___________ 2019

TASK
of the master graduate qualification work
for the student of the group CE-220
Faizullin Artur Rushanovich
in master direction 02.04.02
“Fundamental Informatics and Information Technologies”
(master program “Technologies of high-load systems development”)

1. **The topic** (approved by the order of the rector from 25.04.2019 No. 899)

Using a convolutional neural network for sticker detection in slab continuous casting.

2. **The deadline for the completion of the work:** 05.06.2019.
3. **The source data for the work**

3.1. Lieftucht D. HD mold - A Wide Variety of Caster Assistance Tools for Enhanced Product Quality / D. Lieftucht, M. Korzi, M. Arzberger, F. Fanghänel, M. Langer et al. – 2016.
3.2. Bartos R., Brockmann S., Fandrich R., Endemann G., Heinzel S., Keul C. Steel Manual. – Verlag Stahleisen, Düsseldorf, 2015. – P. 218.
3.3. He F., Zhang L. Mold breakout prediction in slab continuous casting based on combined method of GA-BP neural network and logic rules // The International Journal of Advanced Manufacturing Technology. – 2018. – Vol. 95.
3.4. Meng Q. Using GA-BP Neural Network for Sticking Breakout Prediction in Continuous Slab Casting / Q. Meng, B. Li, J. Qi, C. Yao. – 2017.

4. **The list of the development issues**

4.1. Review existing software methods for sticker detection.
4.2. Design a sticker detection system, including the design of a convolutional neural network.
4.3. Implement a sticker detection system, prepare training data, and train the designed CNN.

4.4. Conduct experiments to examine the effectiveness of the implemented system and compare the results with other existing methods.

5. Issuance date of the task: 08.02.2019.

Supervisor

Assoc. Prof., Cand. Sci.	M.L. Zymbler
The task is taken to perform	A.R. Faizullin

CONTENTS

INTRODUCTION

Topicality

Continuous casting of steel is a rapidly developing method of steel processing. Currently, it accounts for almost 90 % of the overall crude steel output [19]. *Breakout* is one of the most detrimental and dangerous issues in continuous casting, causing loss of production time and significant drops in yield. *Stickers* are a common cause of breakouts. A *sticker* is a part of the strand shell that adheres to the *mold* surface. Stickers can be detected by a peculiar temperature pattern in a *mold heat map*. *SMS group GmbH* (Germany) has developed *HD mold* [10], a cyber-physical system for sticker detection by monitoring and analyzing temperature data from fiber optic sensors during casting. At present, *HD mold* exploits an analytical sticker-detection algorithm that gives a large number of false alarms [9]. This leads to significant losses in production time and substantial economic damage. In this thesis, we suggest a *convolutional neural network* (CNN) that recognizes sticker patterns and can be employed as a full-fledged substitute or an assistant to the currently used algorithm [5].

Goals and objectives

This Master's thesis documents the development of a sticker-detection algorithm based on a convolutional neural network. This new method of sticker detection should be able to detect all stickers and reduce the number of false alarms.

This research has the following objectives:

1) review existing software methods of sticker detection;
2) design a sticker-detection system, including the design of a convolutional neural network;
3) implement sticker detection system, prepare training data and train the designed CNN;
4) conduct experiments to assess the effectiveness of the implemented system and compare the results with those for existing methods.

Structure of the thesis

The Master's thesis consists of an introduction, five main chapters, conclusions, references, and an appendix. The work contains 40 pages; the References section lists 20 entries.

The First Chapter, **«Background»**, gives a background of continuous casting, describes breakouts and stickers, and provides information on the *HD mold* monitoring system.

The Second Chapter, **«Related work»**, contains a review and analysis of previous works on this topic.

The Third Chapter, **«The Design»**, presents the design of the proposed sticker detection system.

In the Fourth Chapter, **«The Implementation»**, we explain the details of implementation of the sticker detection system, including the following: data preparation, augmentation algorithms, data processing, and training of the convolutional neural network.

In the Fifth Chapter, **«Experimental evaluation»**, we describe the experiments and results obtained during the work of the developed system, and a comparison thereof with the *HD mold Breakout Prevention System*.

The **«Conclusions»** section finishes the thesis and offers a brief overview of the task, its solution, and results.

The **«Appendix»** contains various lists: a list of abbreviations, a list of illustrations, and a list of tables.

1. BACKGROUND

Continuous casting

Continuous casting (CC) of steel, as an industrial solidification process, has a relatively short history of some 50 years, not much longer than oxygen steel-making. The CC ratio in the world steel industry is now approaching 90 % of the overall crude steel output, in spite of the fact that it accounted for a mere 4 % in 1970 [19]. Figure 1 shows a scheme of a typical modern casting machine with mold.

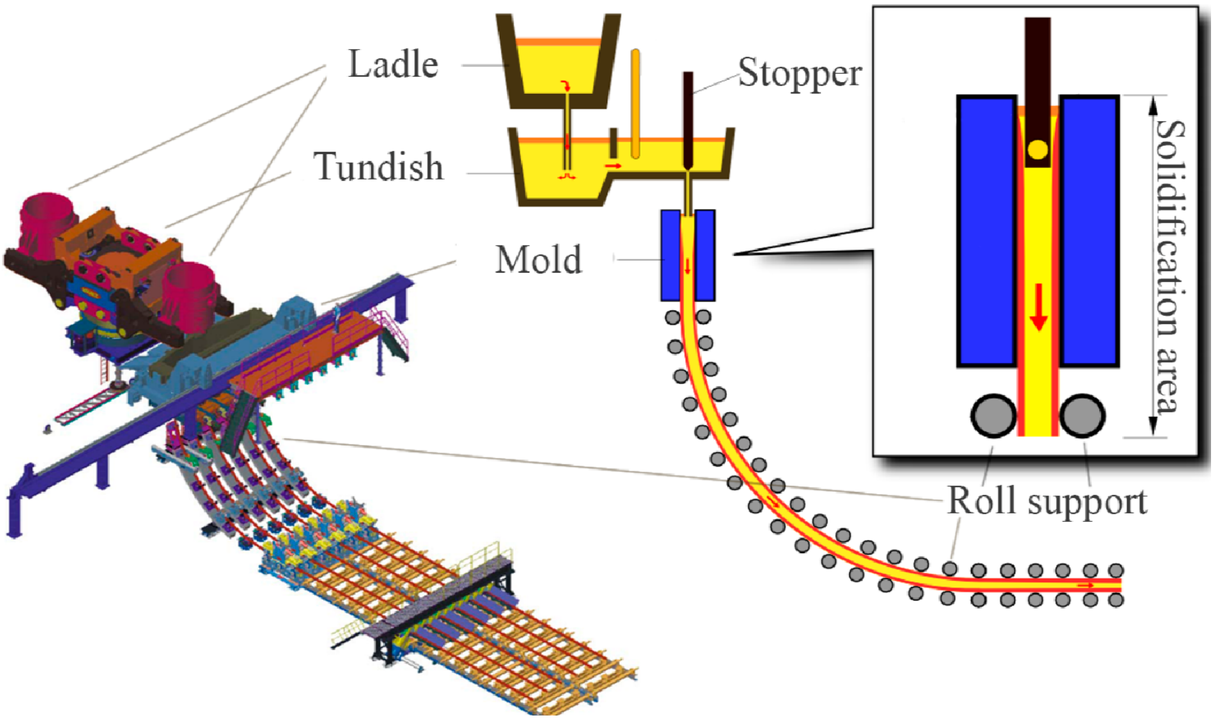

Fig. 1. Scheme of a caster and mold

Steel moves through a casting machine according to the following scenario. Molten metal is tapped from a furnace into a *ladle*. Then, the ladle is transported to the top of the casting machine. From the ladle, the liquid metal is poured via a refractory shroud (pipe) to a holding bath called a *tundish*. The tundish is a reservoir of metal that feeds the casting machine while ladles are switched, thus acting as a buffer of hot metal, smoothing out the flow, and regulating the amount of metal fed to the mold.

Metal is poured from the tundish through another shroud into the top of an open-base copper *mold*. The depth of the mold can range from 0.5 to 2 metres, depending on the casting speed and section size. The mold is water-cooled to solidify the liquid metal directly in contact with it; this is the *primary cooling* process. In addition, the mold oscillates vertically to prevent the metal sticking to the mold walls (i.e. to prevent *stickers*). A lubricant (either powders that melt in contact with the metal or liquids) is added to the metal in the mold to prevent sticking and to trap any slag particles, including oxide particles. The shroud is fitted in such a manner that the hot metal flows out below the surface of the slag layer in the mold.

In the mold, a thin shell of metal next to the mold walls solidifies before the middle section, which is now called a *strand*, comes out from the base of the mold into a spray chamber. The bulk of metal contained within the walls of the strand is still molten. The strand is immediately supported by closely spaced water-cooled rollers to prevent deformations of the wall of the strand that might be caused by the ferrostatic pressure of the liquid still solidifying in the strand. To increase the rate of solidification, the strand is sprayed with large amounts of water as it passes through the spray-chamber; this is the *secondary cooling* process. The final solidification of the strand takes place after the strand has come out of the mold. The result of the work of the casting machine is a steel bloom, a billet, or a slab of any length [3].

Sticker breakout

A *breakout* of liquid metal is a major issue that may occur in continuous casting. It happens when the solid shell of the strand breaks and allows the still molten metal contained within it to spill and solidify over parts of the machine. In most industrial environments, this event is very costly as it leads to an interruption of the strand and typically requires an extended stoppage involving removal of the spilled material from the strand equipment and/or replacement of damaged machinery. A typical breakout for a conventional slab casting machine leads to losses of more than € 250,000 [6]. The most common cause of breakouts during continuous casting are stickers (79 % of all breakouts [4]). A *sticker* is a part of the strand shell which adheres to the mold surface. Under a sticker, the shell wall becomes too thin to support the liquid column above it. In this case, when the thin part of the wall comes out of the mold, a breakout happens.

Figure 2 shows how a sticker leads to a breakout if no action is taken. By measuring the temperatures on copper plates attached to the outside of the mold, stickers can be detected before they can cause breakouts [18].

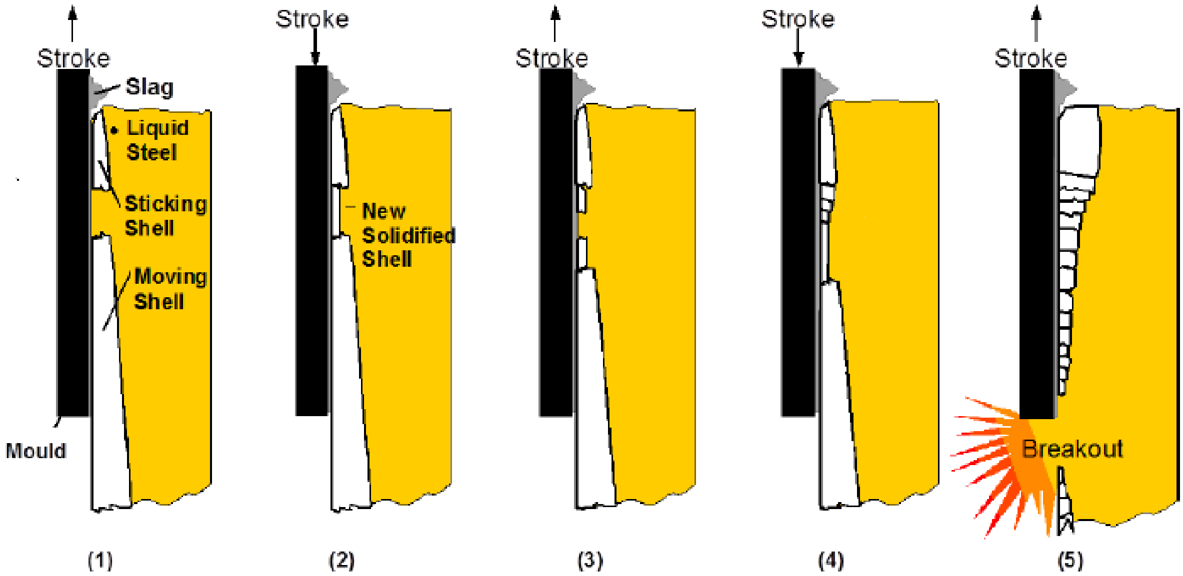

Fig. 2. Sticker breakout

It is possible to prevent sticker breakouts. If a sticker is detected, it is necessary to slow down the casting process for a while to prevent the breakout and then gradually increase the casting speed to a nominal value.

HD mold

HD mold^{FO} is an *SMS group* technology package that uses fiber optic sensors installed in the casting-machine copper mold to provide fast and reliable feedback of real-time casting conditions. The HD mold^{FO} system provides a wide variety of models that exploit temperature feedback to assist the casting machine operator in troubleshooting problems and optimizing production. This assures increased plant availability, growth in yield, protects the casting machine, enhances product quality, and frees the casting machine operator to focus on other important casting machine duties [10].

HD mold^{FO} includes a *Breakout Prevention System* (BPS) which aims to avoid breakouts by real-time monitoring and analyzing temperature data from the fiber optic sensors installed on the mold. A sticker can be detected by a specific temperature pattern in the mold heat map [1, 2]. Given the high cost of breakouts, the current algorithm tries to minimize sticker probability by raising

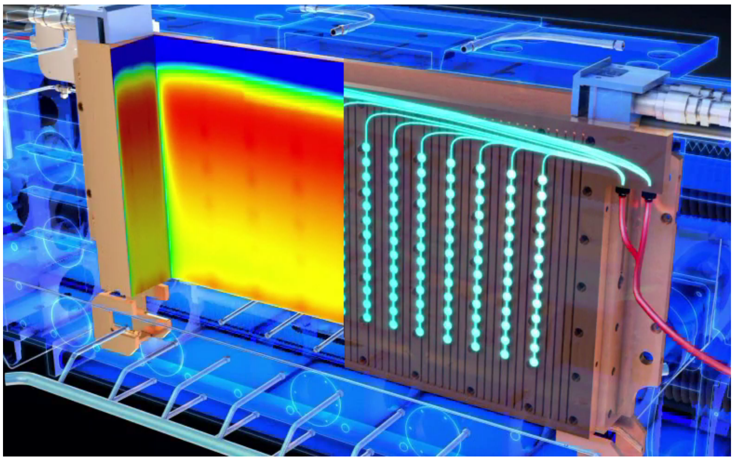

Fig. 3. Fiber optic sensors (blue points) on a broad side of the mold

an alarm as early as possible. This strategy leads to large numbers of false alarms, thereby causing a significant loss in production time and substantial economic damage during continuous casting [15]. In practice, a false alarm can cause a loss of more than € 1000 [6].

Up to 576 fiber optic sensors are placed on the mold surface (see Fig. 3). These sensors send temperature values every 0.25 second to a server in real time. The server adds information on casting speed and sets the slowdown status which shows whether casting speed was decreased due to a true or a false sticker. Then the server sends *frames* (a frame consists of all the values measured at the same time) to the HD mold^{FO} monitoring system. HD mold^{FO} makes decisions based on the frames and information on mold size, position, type, and number of sensors.

Figure 4 depicts the structure of HD mold^{FO}. The *OPC-server* is responsible for data storage and communication between different modules through *OPC-clients*. *Fiber optic sensors* send the measured values to the OPC-server. The *Breakout Prevention System* (*BPS*) is one of the modules that compose HD mold^{FO}. The BPS evaluates temperature gradients in the mold to detect stickers and generates an alarm in this case. The OPC-server sends an alarm signal to the *Drive PLC* (*Programming Logical Controller*) and the operator. Finally, the controller reduces the casting speed temporarily to let the steel cool down and grow a new shell to close the gap. The operator monitors the casting process, including temperature values and sticker alarms, through the *HMI Client*.

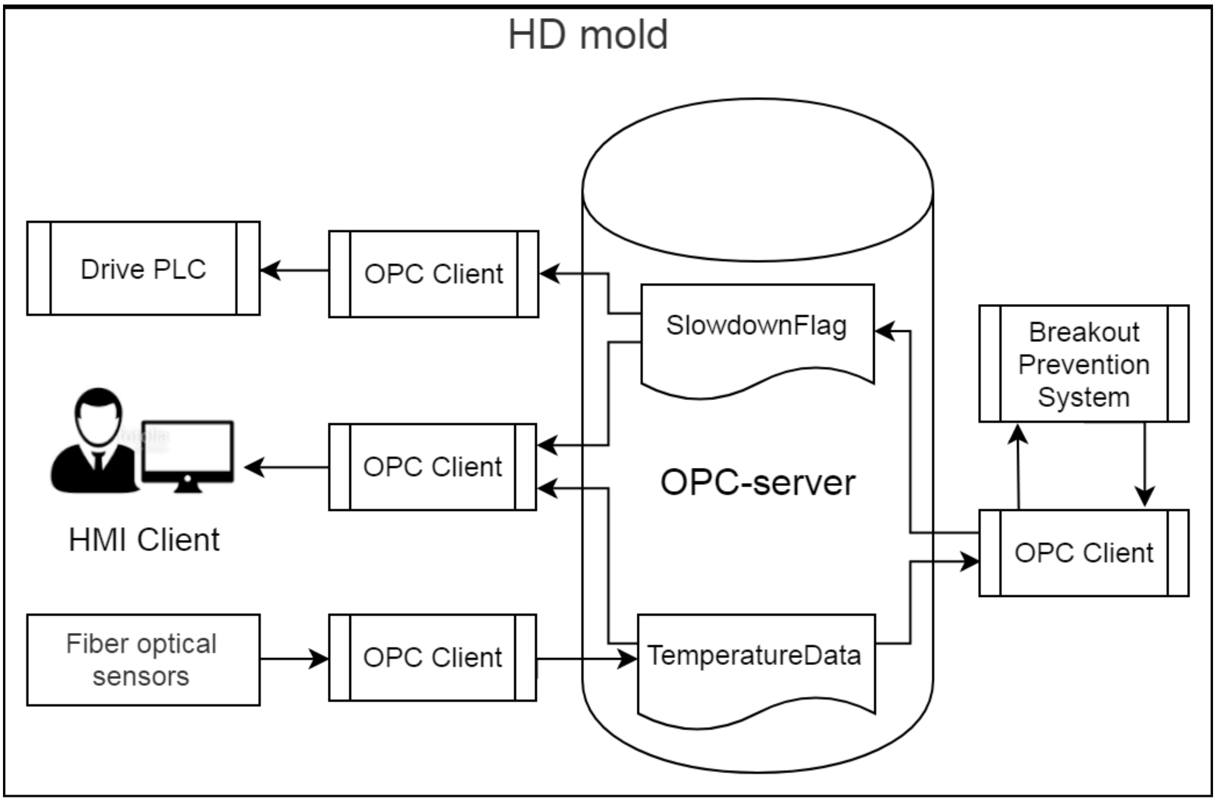

Fig. 4. Structure of the HD mold system

A typical sticker temperature pattern can be detected as follows. Temperature sensors next to the broken strand shell show higher temperatures than their neighbors, which are located against a more isolated part of the strand.

Figure 5 shows the temperatures over time in the neighborhood of three columns by three rows of sensors. The thick lines correspond to the central column, which is the closest to the origin of the sticker. The thin lines reflect the neighboring columns. The line at the bottom of the graph shows the casting speed; the rectangle indicates the moment when the sticker was detected and the casting speed was reduced [9]. Stickers have one particular origin and can be identified as a hotspot moving from top to bottom along the mold. Furthermore, the propagation of the hotspot to the left and to the right can be identified since the hotspot always reaches neighbor columns a little bit later [12].

The current version of the BPS exploits an analytical sticker detection algorithm that uses a set of condition checks to implement the idea discussed above. It handles any suspicious situation as a potential sticker and generates a warning. After the warning, the BPS runs a sequence of additional sub-algorithms; each of them confirms or disproves some typical predetermined

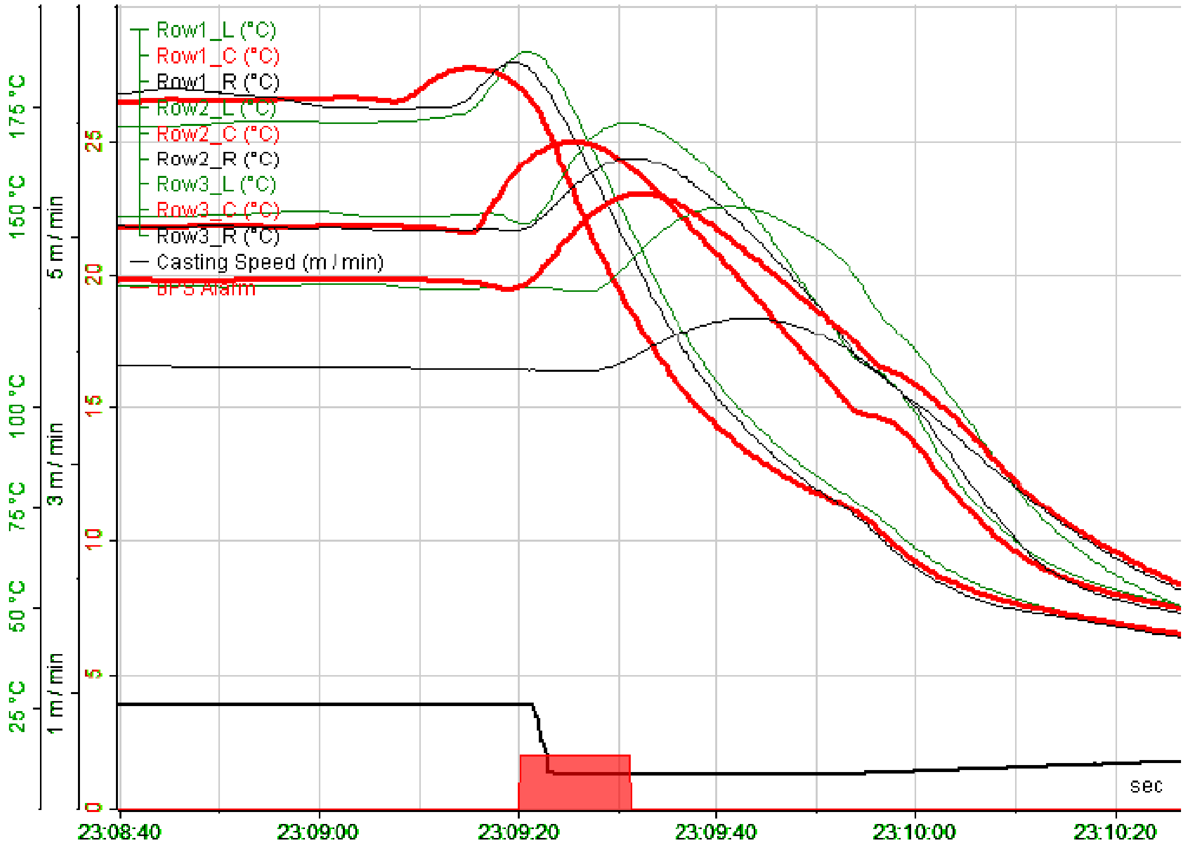

Fig. 5. Typical temperature changes during a sticker event

sticker case. If a sticker is confirmed, the BPS sends an alarm. However, despite the additional checks conducted by the sub-algorithms, this approach leads to a large number of false alarms, which has a negative impact on both the speed and the quality of the production. In this regard, we address the problem of reducing the number of false alarms by embedding a CNN for sticker detection into HD mold^{FO}.

The Data

SMS group GmbH provided a data archive for training the neural network. The archive contains a few years of records of continuous-casting processes from various plants. The records can be divided according to the type of transmitting sensor: a thermocouple or a fiber optic sensor. In this research, we developed data processing algorithms only for fiber optic sensors.

All data in the archive are stored as files with '*.iba' extension. Each file contains about 60 minutes of continuous casting records. Each record consists of *frames*.

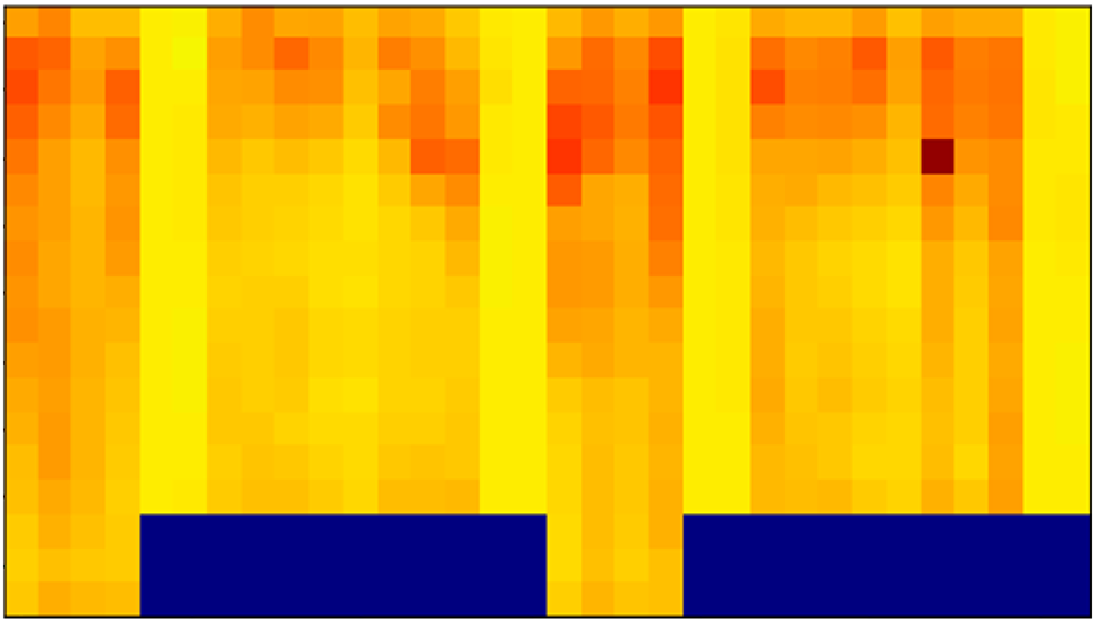

Fig. 6. Frame of temperature data. Each pixel is a temperature value provided by a fiber optic sensor

Figure 6 portrays a *frame*. A frame is a structure containing all the data about a continuous casting process at a particular moment in time. Data for a frame can be obtained in different ways. Temperature values from the sensors are available in real time. Some data, such as the speed of casting or the slab size, must be configured by a casting machine operator. The time of casting, the amount of steel used, and some other parameters are calculated based on other data. Each second, HD mold adds four new frames to the record. All new frames contain current data at the time of writing.

For this research, we analyzed the following data:

Temperature of a sensor. If a sensor works correctly, it sends valid temperature values of the mold surface. Otherwise, it sends $-273\,°C$. Sometimes, temperature values can be distorted while being transferred to the server. In this case, the temperature values of a sensor may differ significantly from those of surrounding sensors.

SlowDown status. The SlowDown status indicates whether casting was slowed down due to detection of a sticker. If the SlowDown status is equal to 0, then the casting process is in normal mode. If the SlowDown status changes to 1, then a sticker was detected and HD mold slowed down the casting process automatically. The SlowDown status is controlled by the HD mold system and changes automatically. After some time, the detected sticker is considered to be removed. Consequently, the SlowDown status changes to 0 and the casting speed is returned to the configured value.

The rest of the data in the archive is not needed for the work of the proposed sticker detection method.

Metallurgical experts from SMS group GmbH assess all the alarms generated by the Breakout Prevention System (an integral part of HD mold) and classify each as false or real. This information is not available either in the frames or in the records. Therefore, we had to categorize all the alarms as false or real manually with the assistance of experts.

Since HD mold^{FO} has been using fiber optic sensors for less than two years and, in addition, exploits an overly cautious breakout-prevention algorithm, the SMS group GmbH archive contains data regarding a pretty small number of true and false stickers, namely frames corresponding to 14 real stickers and 103 false alarms. In order to construct the training set, we took frames corresponding to 11 real stickers and 88 false alarms. The rest of the frames (corresponding to 3 real stickers and 15 false alarms) were used to construct the test set. In addition to “alarm” frames, we have a quite large number of frames corresponding to ordinary work of casting machines.

During real-time work, the server at a plant with HD mold installed and configured generates a new frame four times a second and sends it to all subscribed modules (e.g., Breakout Prevention System) via Open Platform Communications (OPC). Real-time frames and frames in the archive have the same structure.

2. RELATED WORK

Problems related to breakout prevention and sticker detection have been extensively studied. Both technical and software solutions have been suggested to solve these problems.

Technical solutions can be divided into two groups: *prevention* and *monitoring*. Prevention methods work without any software and prevent sticker at a physical level. These methods include such means as mold oscillation and lubricants applied between steel and mold surface. Monitoring systems collect data for further analysis. The main types of monitoring systems are described below.

Monitoring hardware

Friction monitoring. First friction monitoring technologies were introduced in the late 1970s. The concept of monitoring mold friction to detect and thus prevent breakouts stems from the basic relation that increased friction between the strand and the mold is caused by poor lubrication. This concept can be implemented with *accelerometers*, *load cells*, or *strain gauges*. For all implementations, the detection rate for stickers is 60 to 70 %, while the false alarm rate exceeds 30 % [4].

Thermal monitoring. The relationship between sticker-type breakouts and increased heat flux through the mold wall contiguous to the sticker was recognized in 1954. However, it was not until the 1970s, with the widespread development of continuous casting and the availability of sophisticated microprocessors for collecting and analyzing specific temperature inputs, that sticker breakout detection by thermal analysis became a reality. The temperature-based detection system works on the principle that a localized heat-transfer variation can be easily recognized. For this method, one needs to install thermal sensors around the mold surface and analyze incoming temperature data. The thermal sensors used are of two types: *heat flux sensors* and *embedded thermocouples*. Heat flux sensors are mounted on the cold face of the mold, and therefore the drilling of holes in the mold assembly is not required. However, heat flux sensors have a poor sensitivity, and this fact reduces sticker detection rate to 90 %. In contrast to heat flux sensors, thermocouples are installed into specially drilled holes in the mold. This means that thermocouples are closer to the hot surface of the mold. In this case, sticker detection rate can reach 100 %, whereas false alarm rate per heat drops to 0.04–0.08 % [4].

Fiber optic sensors. These sensors were suggested in the late 1990s, it is the newest method of thermal sticker detection. The principle of work is the same as in the case of thermocouples: sensors are installed into the mold and provide temperature data. Fiber optic sensors, however, are more robust to electrical distortions and can provide more accurate temperature values. Also, they are smaller, which makes it possible to install 576 sensors in a mold, compared to only 176 thermocouples, and produce a more detailed heat map of the mold surface. Thanks to this improvement, fiber optic sensors can detect all stickers and reduce the number of false alarms by 50 % compared to thermocouples.

All monitoring methods we have described require special software for analysis.

Monitoring software

Historically, the first sticker detection and breakout prevention solutions relied on analytical algorithms, for example, *HD mold^{FO}* (see Subsect. 1). Another monitoring system was presented in [7]. This system also exploits analytical algorithms ("logic judgement", according to the authors). Both monitoring systems achieve 100 % sticker detection rate, but generate a large number of false alarms: 0.012 % and 0.015 % false alarm rate per heat, respectively. Such numbers of false alarms lead to costs approaching € 100,000 per year [6].

Machine learning solutions for sticker detection have been widely used in the last five years. One well-known problem for machine learning algorithms is the lack of training data. For example, the authors of [14] used 300 breakout samples in their study. According to them, it is easy for the error function of the neural network to fall into a local extreme point instead of reaching a real minimum. This reduces the quality of detection. Another problem connected with the small amount of data can be *overfitting*. In this case, the neural network works very effectively on the training set but shows bad results on the test set.

In [11], the mold temperature and its velocity thermographs were constructed on the basis of mold temperature readings measured by thermocouples during slab continuous casting, which allowed the authors to devise a visual detection method for sticker breakouts. They found important visual features of stickers and computed statistics for some sticker parameters, such as geometry, temperature, and propagation velocity. This is a valuable work for further research but it contains no results on sticker detection.

The effectiveness of a genetic algorithm-based back propagation (GABP) neural network model and its application to breakout prediction in continuous casting were investigated in [20]. The authors analyzed the formation of sticking-type breakouts and the prediction principle of the thermocouple thermometry method. Then, they provided a genetic algorithm-based back propagation neural network model by fusing the genetic algorithm and an error back propagation neural network to offset the demerits of one paradigm by the merits of the other. Finally, the GABP neural network model was applied to breakout prediction in continuous casting and the feasibility of the model was verified in a test which showed a sticker detection rate of 100 % and 2 false alarms generated out of 120 regular (no sticker) samples.

In [8], the authors used a genetic algorithm (GA) and a back propagation (BP) neural network to construct a time series model for recognizing the temperature change waveform of a single thermocouple in a mold breakout process. Based on the breakout mechanism, they resorted to logical rules to construct a spatial model of multi-thermocouples for identifying the two-dimensional (2D) propagation behavior of the sticker. The time series model based on a GA-BP neural network and the spatial model based on logical rules form a new breakout prediction method. On field tests, this method showed 100 % sticker detection rate and generated 4 false alarms out of 2,911 regular samples.

A single thermocouple time sequence network and a T-shaped four-thermocouple space network were constructed in [14] for sticking breakout prediction. Moreover, a genetic algorithm (GA) was used to determine the initial values of the network weights required to make the BP neural network converge to the global optimum more quickly, while avoiding its plunging into a local extreme. As a result, the suggested neural network achieved 100 % sticker detection rate and generated 2 false alarms out of 400 regular samples.

Adaptive principal component analysis (APCA) was applied in [2] as a predictor of inputs-outputs models which are defined by the mold bath level and casting speed. The authors conducted simulations of continuous casting and compared the mean squared error of the suggested algorithm with the principal component analysis method, non-linear system identification based on neural network, and the support vector regression method. The method suggested in the paper gives the best mean squared error but the authors do not conducted experiments with real data, only with simulated data.

Discussion

As we can see, the topic of our research has been considered extensively in the literature, and there is a large number of new studies that touch upon both the software and the hardware aspects of the problem. For the time being, fiber optic sensors are the most advanced devices used for mold temperature measurement. However, all reviewed software exploits signals from thermocouples.

Currently used industrial monitoring systems take advantage of analytical algorithms for sticker detection and breakout prevention problems. Nevertheless, the majority of new methods are implemented through different machine learning solutions. In the papers we have reviewed, the authors use back propagation neural networks with different input signals. Trying to overcome the lack of data, they resort to genetic algorithms. It should be noted that the results they have achieved are outstanding, which demonstrates the benefits of neural networks and other machine learning methods.

In the present work, we use data from fiber optic sensors. This is a new technology, so we had only 14 breakout samples available, which is less than other authors had at their disposition. However, owing to the designed and implemented augmentation methods, we managed to achieve better results.

3. THE DESIGN

3.1. Sticker Detection System

Sticker Detection System (SDS) is the name of this project. The system includes the suggested algorithm for sticker detection, scripts for training-data preparation, scripts for CNN training, and the CNN itself [5].

In the beginning, the convolutional neural network in the SDS has to be trained. The SDS connects to a *data archive* for this purpose. Further, the SDS works with the *OPC-server* for real-time data analysis (see Fig. 7).

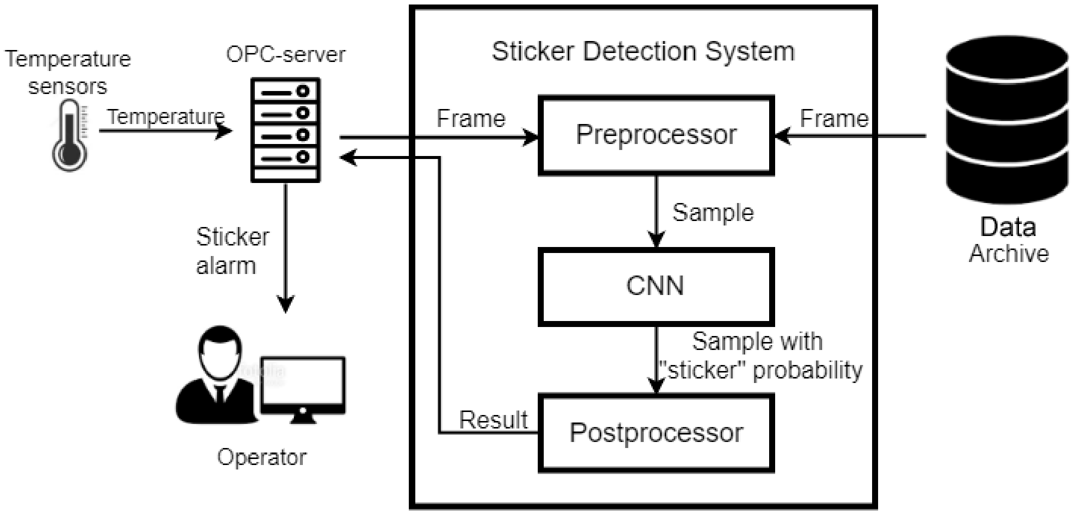

Fig. 7. Use of the Sticker Detection System

The *Preprocessor* performs data preprocessing (cleaning, normalization) before the analysis with the CNN or CNN training. Input data for the preprocessor comes from a data archive or from the OPC-server in real time.

The *convolutional neural network* (CNN) performs the data analysis. It handles the output of the preprocessor as input. The output of the CNN is the probability of a sticker event for each input sample.

The *Postprocessor* performs the final processing of the analysis results and the compilation of the results if necessary. If the system works in training mode, then the predicted sticker type (real or false) is compared with a real sticker type (determined by experts). During real-time work, the result of the analysis is sent back to the OPC-server. In case of a real sticker, the HD mold^{FO} monitoring system automatically runs the breakout prevention algorithm and notifies the operator of the casting machine.

In case of work with the OPC-server, the SDS is integrated with HD mold^{FO}. The OPC-server can be deployed together with the HD mold^{FO} software at a steel plant and connected directly to a casting machine. The OPC-server performs a basic processing of data from temperature sensors. This processing involves handling of electrical signals and their conversion to numbers in a given format. The frequency of data acquisition is four times per second.

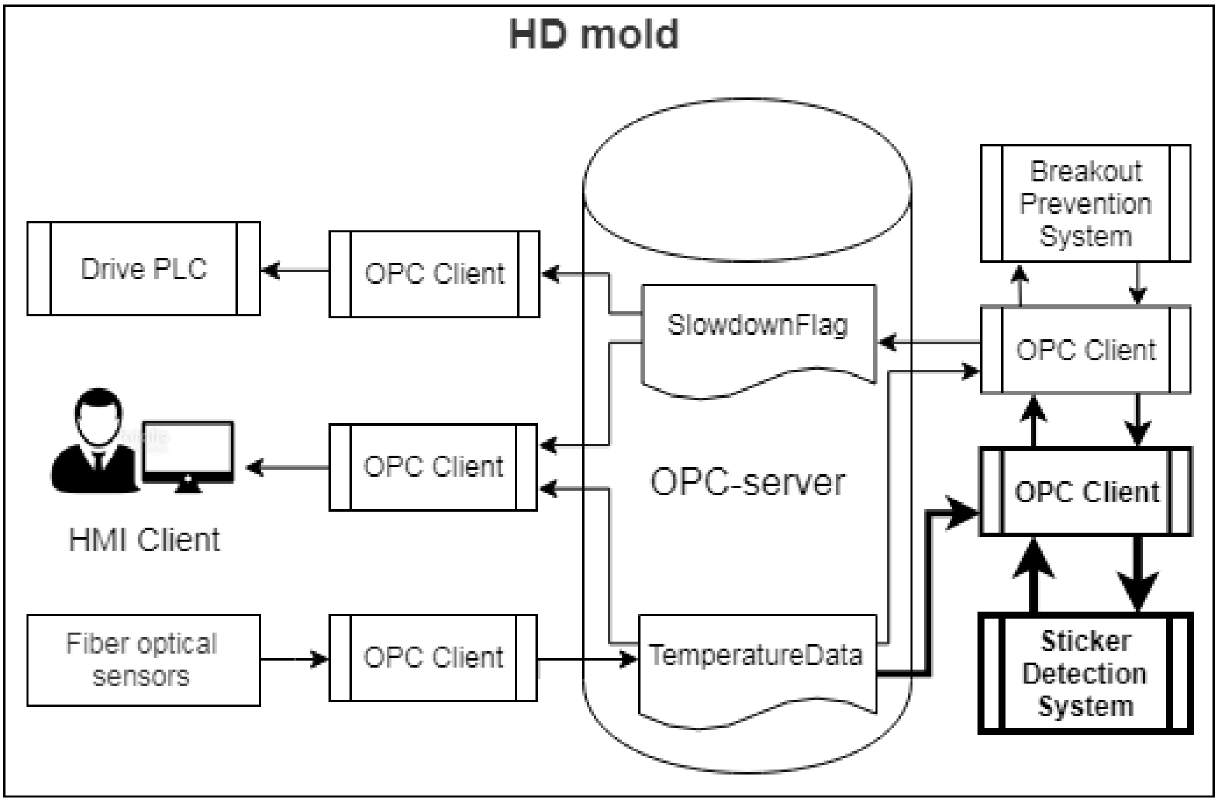

Fig. 8. Sticker Detection System in the HD mold^{FO} environment

Figure 8 depicts the work of the Sticker Detection System in the HD mold^{FO} environment. There are two possible ways of using the SDS. In the first case, the SDS works as an independent module of HD mold^{FO}. The SDS receives all data directly from the OPC-server and decides whether there is a sticker. Another option is to use the SDS as a sub-algorithm of the BPS. Data from the OPC-server enters the BPS. In case of some suspicious activity, the BPS runs additional checks including the SDS. The final decision is the result of voting of sub-algorithms.

The data archive is available only at SMS group GmbH premises. We used these data for a primary configuration of the SDS and training of the CNN included in the SDS.

3.2. Use case

A use-case diagram of the SDS is shown in Figure 9.

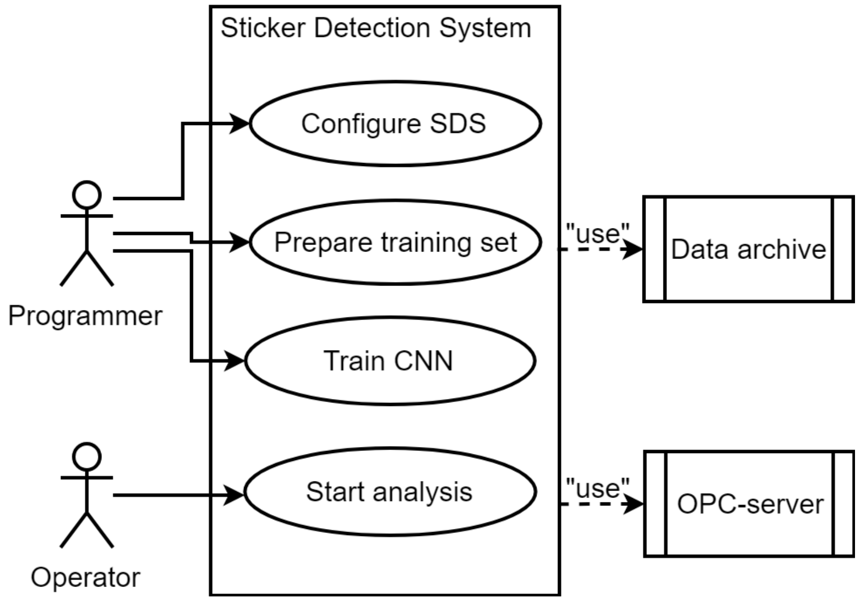

Fig. 9. A use-case diagram for the Sticker Detection System

The SDS have two actors, namely: the *programmer* and the *operator*.

The *programmer* role is played by the developer. The programmer is responsible for preparing the SDS for work. The programmer prepares the training set for the CNN, trains the CNN, or configures the SDS.

The *operator* role is played by the operator of the casting machine. Since the HD mold^{FO} monitoring system works in automatic mode, the operator can only start or stop the data analysis.

4. THE IMPLEMENTATION

In this section, we describe the details of the Sticker Detection System implementation. In the beginning, in 4.1, we give details on the preparation of raw data for further processing. In 4.2, we describe methods of data augmentation.

Subsections 4.3, 4.4, and 4.5 deal with data preprocessing, training of the convolutional neural network, and data postprocessing, respectively. The scheme in Figure 10 shows these three steps.

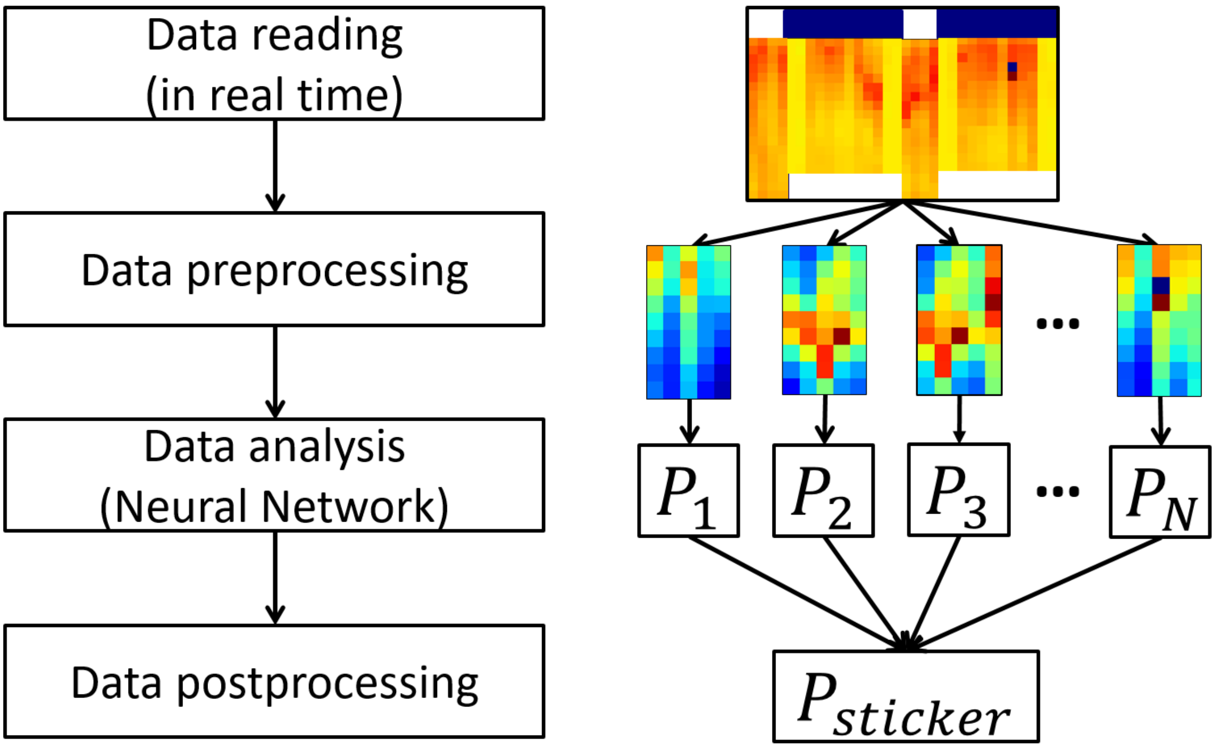

Fig. 10. Data analysis steps

Finally, Subsection 4.6 provides information on the deployment of the Sticker Detection System.

4.1. Data preparation

This section describes data preparation for both the training set and real-time work.

The training set consists of samples classified as "sticker" or "not sticker". The first class contains only samples of real sticker events. The second class includes samples of both false alarms and regular work (no stickers, no alarms).

A sample from the training set consists of data contained in 30 frames: one *key frame* and 29 frames preceding it. These frames correspond to 7.5 seconds of continuous casting real time. We chose this time according to expert estimations of the time it takes from the emergence of a sticker to its detection by the existing method (implemented in the Breakout Prevention System).

We choose the key frames as follows.

First, we mark as key frames all frames at which HD mold$^{\mathrm{FO}}$ generated an alarm (real or false). We use the SlowDown status to find all alarms automatically. A sticker is a process extended in time. Therefore, we consider 240 frames before an alarm (60 seconds of casting) and 1,200 frames after it (5 minutes of casting) as a sample of irregular work. Other frames can be considered as samples of regular work.

Each alarm (real or false) provides a sample of the corresponding class. A real alarm is a sample of the “sticker” class, whereas a false alarm is a sample of the “not sticker” class.

Further, we mark random frames of regular work as key frames. Since HD mold$^{\mathrm{FO}}$ is constantly running, we can generate as many key frames of regular work as we need.

During real-time work, we mark the last received frame as a key frame and add it to a special queue. In this queue, we store the last 30 frames for analysis.

Thus, a sample of the training set is an $F \times S$ rectangular matrix in which $F = 30$ is the number of frames and S is the number of sensors in the current mold configuration. Each cell of the matrix stores a temperature value from the respective sensor at a particular time. The rest of data in the frames are not used.

As a result of the data preparation, we created a training set containing 14 samples of real stickers for the “sticker” class, 103 samples of false alarms for the “not sticker” class, and 11,701 samples of regular work for the “not sticker” class. All samples of false alarms are actually samples of regular work in which the Breakout Prevention System made a mistake during the analysis. Samples from the training set are the input for the Sticker Detection System.

4.2. Data augmentation

SMS group GmbH data archive contains only 14 records with real sticker alarms, all of them produced by fiber optic sensors. This number of “sticker” samples is not enough for the proper training of the convolutional neural network. In order to increase the number of “sticker” samples, we generate artificial data by means of the following methods.

Sticker transferring (see Fig. 11). First of all, we have to extract the pattern of a real sticker. By a *pattern*, we mean the difference between temperatures during a sticker event and temperatures of regular work under the same conditions. To extract a pattern, we compute the temperature difference between the first and all other frames. As a result, we obtain an *extracted sticker pattern* which is an array of the same size as the original sticker and contains only temperature changes during time. Next, we add this sticker pattern to a “not sticker” sample of regular work and get a new “sticker” sample. When the constructed artificial “sticker” sample was shown to experts, they marked it as a real sticker. This means that the artificial “sticker” sample can be used as a training set. Note that this augmentation step is performed before data preprocessing. By this augmentation method, we obtained 6,307 additional “sticker” samples out of 11 original “sticker” samples and 2,348 “not sticker” samples from 88 samples of false alarms.

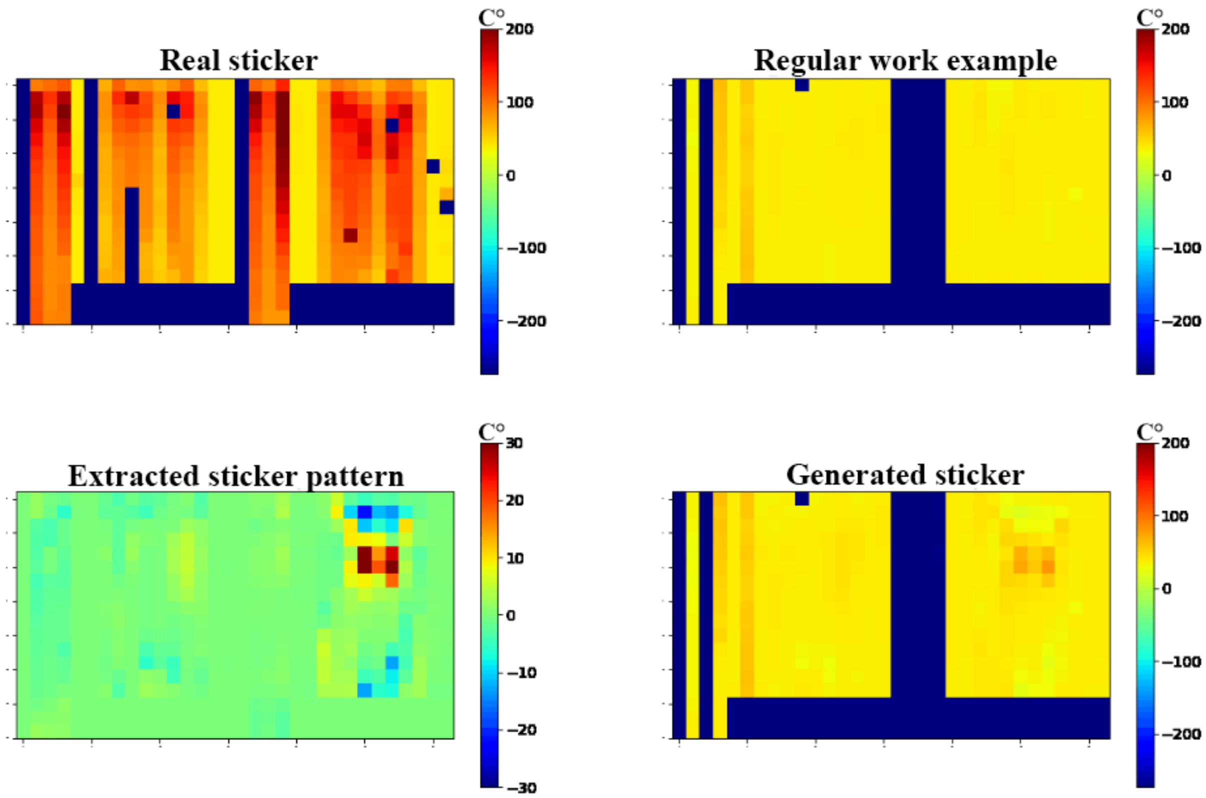

Fig. 11. Augmentation by sticker transferring. In this sample, there is a sticker on the right part of the mold

Sticker mirroring (see Fig. 12). Stickers are symmetrical and V-shaped [11]. It is owing to this feature that "sticker" samples can be mirrored without loss of quality. This augmentation step is performed after preprocessing. By this method, we can double the number of samples if we need.

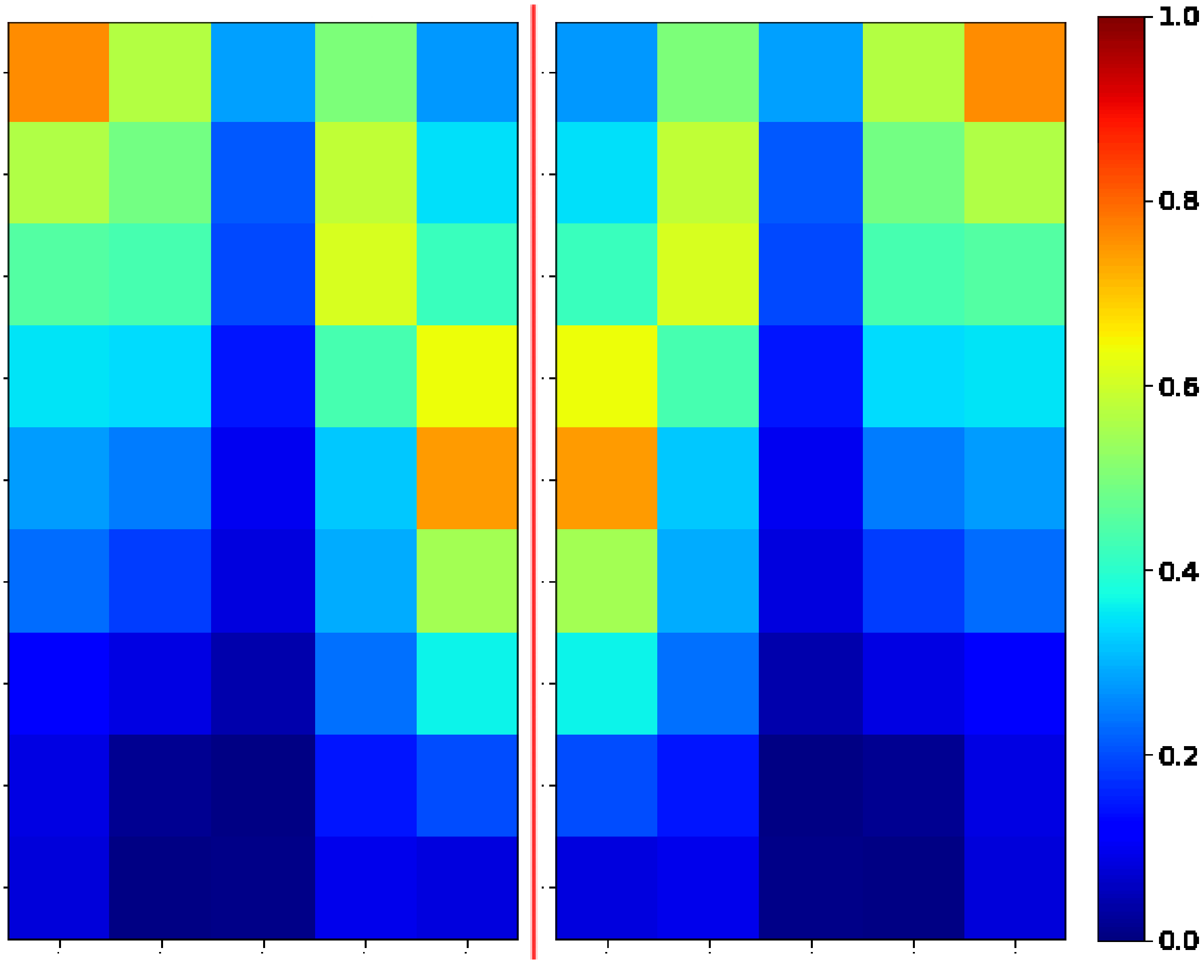

Fig. 12. Augmentation by sticker mirroring

The augmentation results are given in Table 1. Note that we apply augmentation only to real stickers and false alarms, and not to samples of regular work since there is a considerable number of the latter and they do not need to be augmented. We call all the samples after preprocessing *samples for CNN*. The number of samples increases in the course of preprocessing.

4.3. Data preprocessing

During the preprocessing step, we prepare data for input to the convolutional neural network.

Table 1. Augmentation results

Case	Cases		Samples for CNN	Class
	Real	Artificial		
Sticker	11	6,307	25,000	Sticker
False alarms	88	2,348	7,000	Not sticker
Regular work	11,701	0	30,000	Not sticker

The number of active sensors on the mold surface can change during the casting process. It depends on the current mold configuration. Also, some of the sensors can get damaged during casting. Therefore, the input to the SDS (the rectangular matrix of temperatures) can have various sizes. In the preprocessing step, we reduce all the data to a unified size and clean invalid data from damaged or deactivated sensors.

Figure 13 depicts the preprocessing of input data. Preprocessing consists of the following steps:

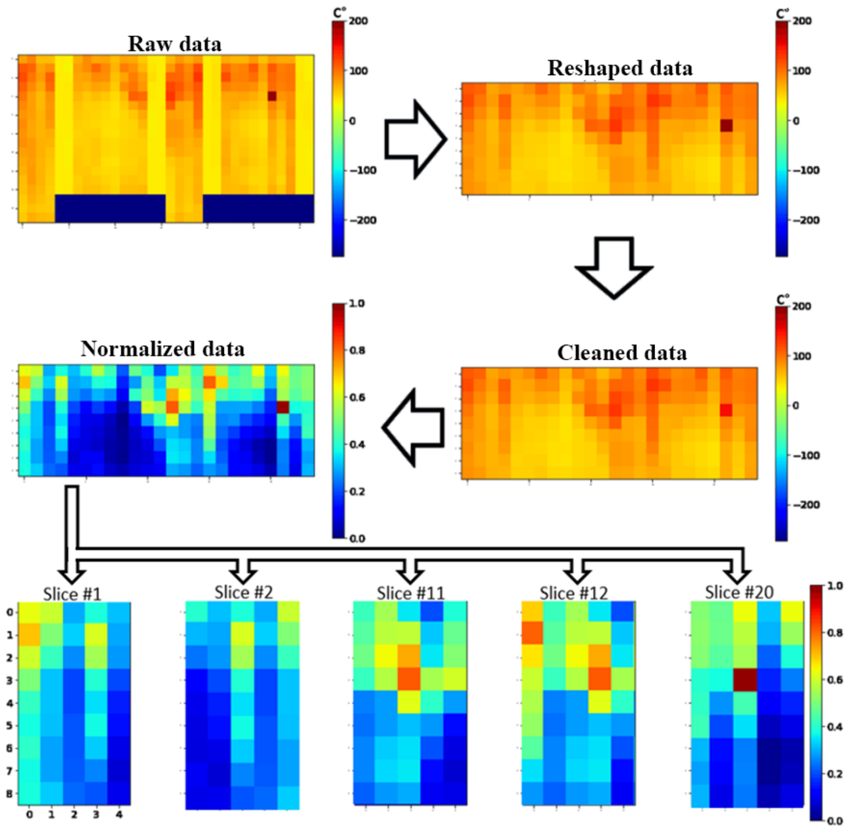

Fig. 13. Preprocessing steps: reshaping, cleaning, normalizing, slicing

Data reshaping. The relative positions of temperature values is important for the proper work of the convolutional layers of the neural network. In case of incorrect relative positions, the convolutional kernel will find nonexistent sticker characteristics. Since the sensors are placed on a rectangular area of the mold, we convert the two-dimensional array to a three-dimensional tensor of the form $F{\times}H{\times}W$, where H is the number of sensors in a column and W is the number of sensors in a row.

Data reduction. Depending on mold configuration, some temperature sensors can be positioned above the liquid-steel level. Also, stickers can appear on the top and in the middle part of the mold. To accelerate processing, we get rid of ineffective data from all sensors on the lower part of the mold. From this moment, we will handle an array of size $F{\times}h{\times}W$ with $h = 9$.

Data cleaning. Since some sensors on the mold can get damaged during casting, thereby sending invalid signals with strong deviations from those of neighboring sensors, we replace such values with the average value of the temperature for the current F frames. Temperature deviation is considered high if it is more than 65 °C from the mean temperature value.

Data normalization. A sticker can be detected by a characteristic temperature change compared with neighboring sensors. Also, absolute temperature values can significantly vary from one mold to another, depending on both the mold configuration and the type of steel. Thus, we are only interested in relative temperature values and temperature changes over time. This ensures that handled data is valuable under different casting conditions, thereby improving the quality of detection. For this purpose, we use max-min normalization of data to scale all values to the interval $[0; 1]$.

Data slicing (see Fig. 14). The size of input data for the convolutional neural network must be unified (this is prescribed by the specificity of its work). During previous steps, we determined F and h values. To unify the widths of the input matrices, we slice the data into arrays of width w. Since a typical sticker has three sensors in width [12], we set $w = 5$, so that a sticker takes up most of a heat map image. Next, we move a sliding window of length w along each array, with a step of one sensor, and create a new sample at each step. This results in up to $W - w + 1$ samples generated from one key frame. Each sample will be marked as “sticker” or “not sticker” for training purposes.

After the steps above described, we obtain each sample as a

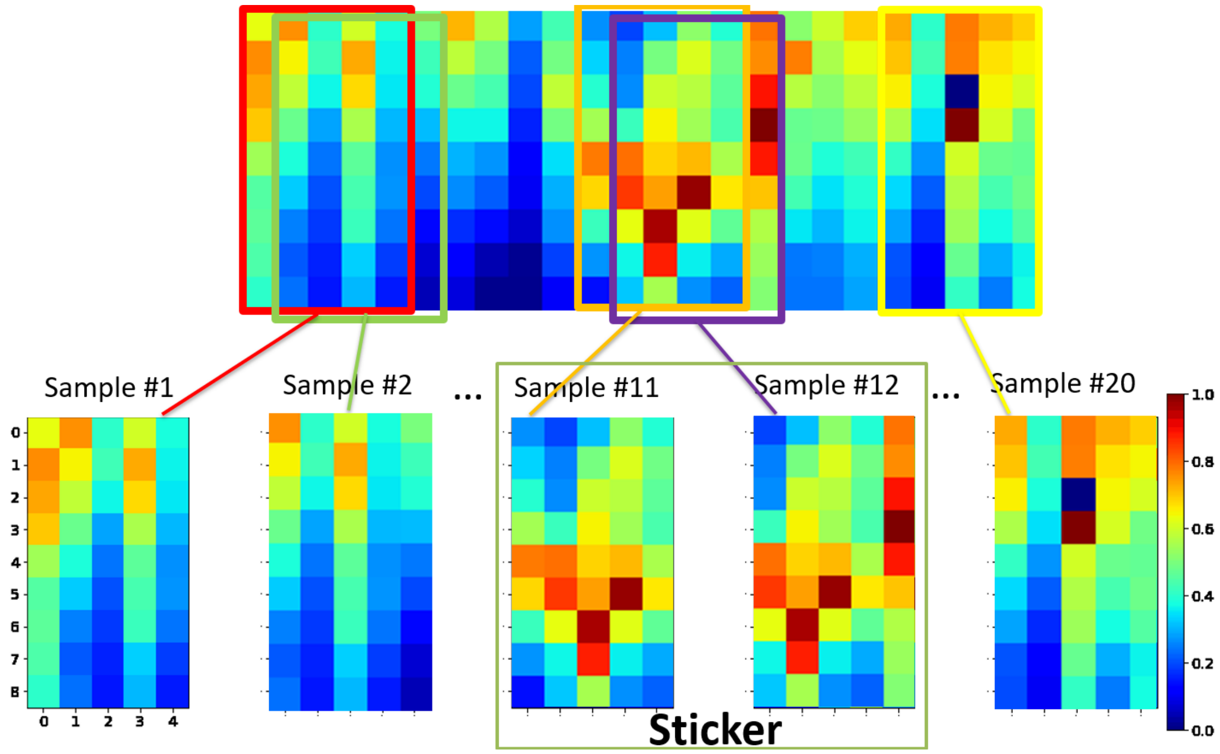

Fig. 14. Data slicing

three-dimensional array of $F \times h \times w$ size (see Fig. 15) containing real values between 0 and 1. These arrays are used by our CNN. Finally, if a sample contains the central position of a real sticker, we mark it as "sticker"; otherwise, we mark it as "not sticker".

4.4. CNN training

Our approach to sticker detection in continuous casting involves the use of a CNN. A CNN is a deep neural network with a convolutional layer and a pooling layer in addition to standard hidden perceptron layers. These layers are not fully connected and have significantly fewer parameters that must be learned. Thus, a CNN learns faster in comparison with traditional neural networks. In addition, convolutional layers allow for efficiently learning important image characteristics [17]. We can consider our three-dimensional samples as images because the relative position of the sensors in time and space plays an important role.

After data preparation, we had the following samples as SDS input: 14 real stickers, 103 false alarms, and 11,701 samples of regular work. We divided all the data into two parts: a training set and a test set. The training set is used

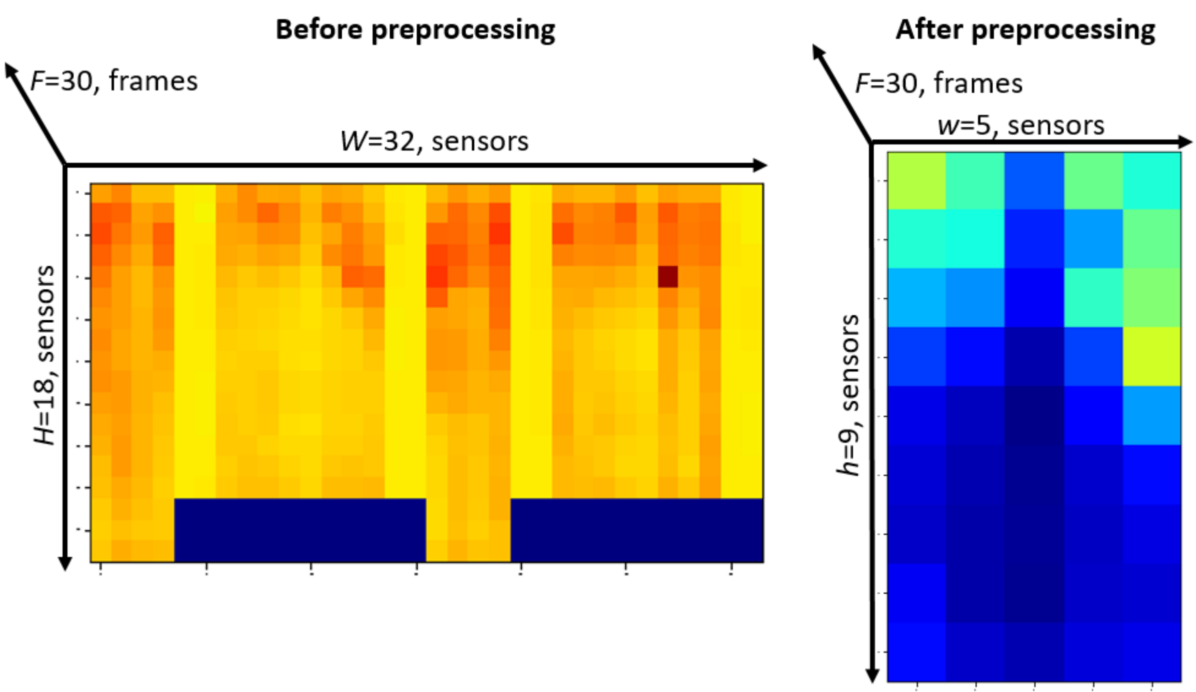

Fig. 15. Preprocessing results

for CNN training. After training, we used the test set to estimate training results.

The augmentation and preprocessing steps gave the following data for the training set as input to the CNN: 25,000 "sticker" samples and 37,000 "not sticker" samples.

Our CNN takes as input a three-dimensional array containing F preprocessed sequential frames. Each frame is a two-dimensional array of size $h \times w$ with real values ranging from 0 to 1. The CNN outputs two positive real values whose sum is 1, namely the probabilities of "sticker" and "not sticker" events for the given input.

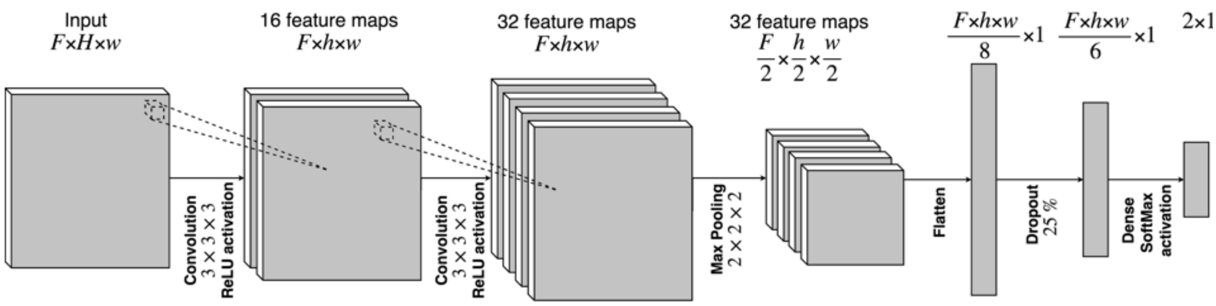

Fig. 16. CNN structure

Figure 16 shows the structure of our CNN. It consists of an input layer, an output layer, and several hidden layers which we consider in more detail below.

Convolutional layers. These two layers have similar structures and perform the convolution of the input data. The output of the convolutional layer is a feature map. In most cases, the sticker area spans three sensors in width and two sensors in height.

Accordingly, we chose 3×3×3 as the size of the convolutional kernel. We use a rectified linear unit (ReLU) as activation function since it works faster and yields better results in such tasks.

Max-pooling layer. This layer reduces the size of a feature map discarding areas containing no key features. In addition, this increases CNN's robustness to scaling. We chose a pooling size of 2×2×2 and 2 strides. The output of the max-pooling layer is a feature map of smaller size (half the size of the input feature map in this case).

Flatten layer. This layer transforms a three-dimensional tensor into a one-dimensional vector.

Dropout layer. This layer randomly deactivates 25 % of neurons to prevent overfitting of the CNN. Overfitting is a training error which leads to good prediction results on the training set and bad prediction results on other datasets.

Dense layer. This is a fully connected layer which applies SoftMax activation to input data and transforms them into two signals. The values of these signals represent the probabilities of "sticker" and "not sticker" events.

The probabilities of "sticker" and "not sticker" events add up to 1. Therefore, we will consider below only the probability of "sticker".

After the augmentation step, we obtained for the CNN 25,000 samples marked as "sticker" and 37,000 samples marked as "not sticker". This data set was divided into two parts: a training set and a test set. The training set consists of 22,000 "stickers" and 32,000 "not stickers". The test set contains 3,000 "stickers" and 5,000 "not stickers".

During this step, the answer of the CNN was marked as correct if the probability of "sticker" was greater then 0.5. The trained CNN achieved 73 % accuracy. This is not enough for proper sticker detection. However, this prediction was generated for a small part of the mold surface. We ran the CNN on each small overlapped part and got various probabilities as the outcome. To get one final result, we implemented the next step: postprocessing.

4.5. Data postprocessing

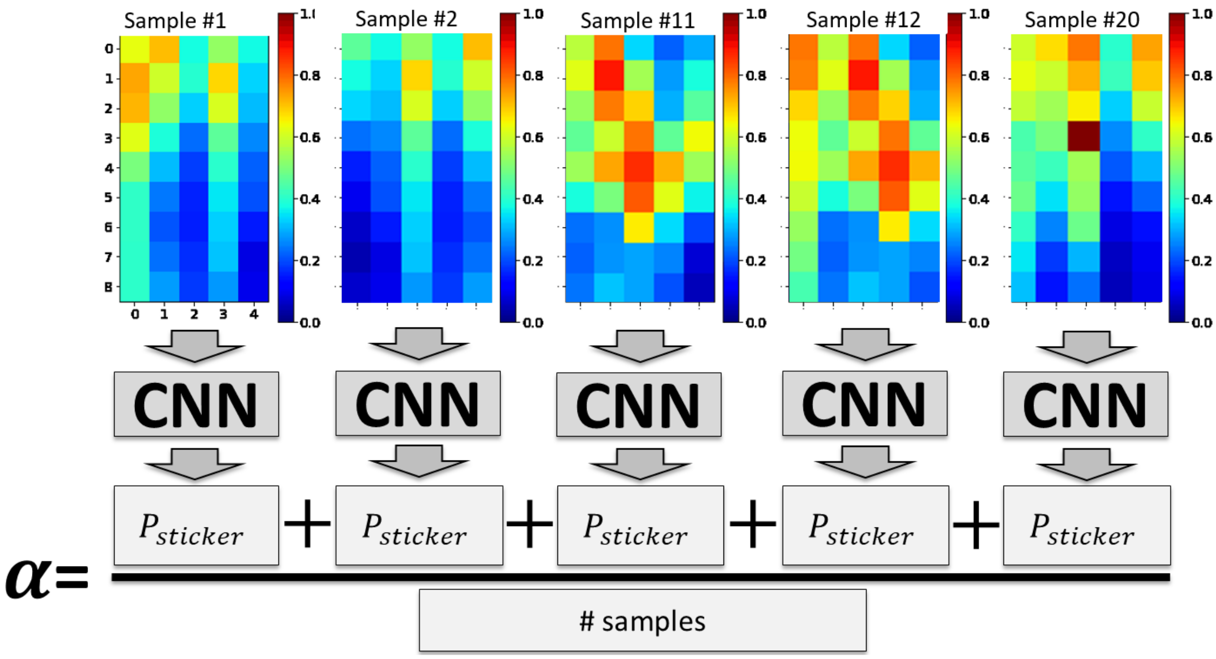

Fig. 17. Data postprocessing

At the postprocessing step (see Fig. 17), we compiled the final answer from several probabilities calculated by the CNN. For this, we computed the mean value α of all the probabilities:

$$\alpha = \text{mean}(P_{\text{stickers}}). \tag{1}$$

Next, we compared α with the threshold value α_{SDS} or $\alpha_{\text{BPS+SDS}}$. If α exceeded the threshold value, then we got a sticker. We computed α_{SDS} and $\alpha_{BPS+SDS}$ experimentally. In section 5.1, we offer a description of the experiments performed.

4.6. The Deployment

The Sticker Detection System consists of three individual parts: the training module, the working module, and the convolutional neural network. The training module is a program that includes functions for data preparation, data augmentation, and CNN training. The working module is a program responsible for real-time data analysis and transmission of notifications about detected stickers to a server. The convolutional neural network is, in the case considered, a file containing all weights and parameters of the trained neural network model.

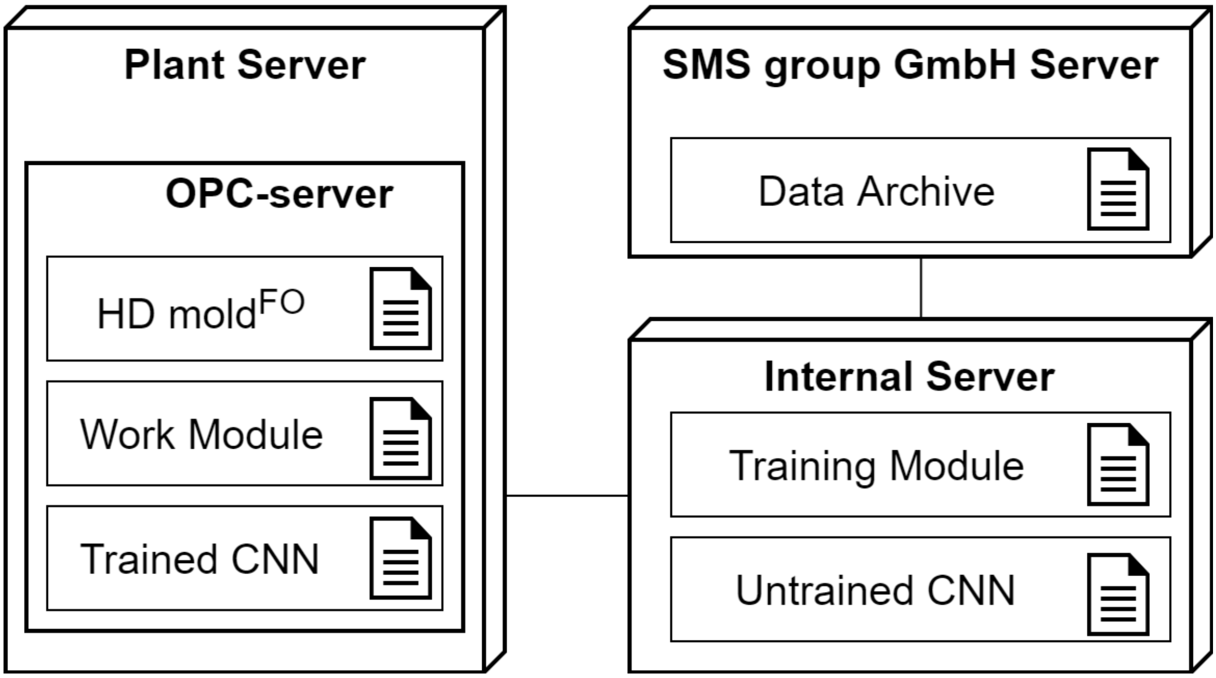

Fig. 18. Deployment of the Sticker Detection System

The training module is only for internal use within a company. For data preparation, it is necessary to partition real and false alarms manually. The rest of the work on training set preparation, data augmentation, and CNN training is fully automated.

The training module works as follows. First, we perform the data preparation step to generate input data for the SDS. Then, we run the first step of data augmentation, namely sticker transferring. After data augmentation, we process all real and artificial samples in the preprocessing step to prepare them for the CNN. All samples are divided into two classes: "stickers" and "not stickers". We train the CNN on the obtained samples and save the trained model to a file. After that, we test the trained model on a test set which includes only real samples. We test our system in real-time mode performing all processing steps sequentially: preprocessing, analysis with CNN, postprocessing. The CNN is ready for work when the test result satisfies the requirements of the customer. Otherwise, we repeat the training process from the beginning.

The working module is a part of the HD mold^{FO} monitoring system. It is deployed on the customer's OPC-server along with other software. The working module requires a trained CNN for its work. When a sticker is detected, the working module generates an internal signal for HD mold^{FO} which automatically runs breakout prevention algorithms.

The working module of the SDS takes advantage of the CNN to process the data in a frame-by-frame manner according to the following scenario. First, the SDS receives an $H{\times}W$ frame from HD mold$^{\mathrm{FO}}$. Then, by means of preprocessing, we transform the frame into (at most) $W - w + 1$ samples of size $F{\times}h{\times}w$. After that, the CNN outputs for each sample the probability of a "sticker" and a "not sticker" event. Next, in the postprocessing step, the SDS evaluates the "sticker" event and generates a sticker alert if needed.

The SDS is implemented by means of the following software stack: MATLAB [13], Deep Learning Toolbox and OPC ToolBox for MATLAB, Python [16], and Keras with Tensorflow backend.

5. EXPERIMENTAL EVALUATION

We conducted two experiments during the research. The purpose of the first experiment was to find an optimal value for the postprocessing parameter α. In the second experiment, our approach was compared with other existing methods.

We used only real data for the test set. This part of the data archive corresponds to 9,585 real cases of continuous casting, namely 3 sticker cases, 15 false alarms, and 9,567 examples of regular work. These data were not used for the CNN training.

We also used *MatrikonOPC Explorer* during the test. This application reads files from a data archive and then transmits the data as a real OPC-server. Owing to this feature, we can test the work of the SDS under real conditions. Moreover, MatrikonOPC Explorer allows to change the speed of data translation and provides new frames more often than four times per second. This means that we can run the test faster and without real hardware.

5.1. Optimal postprocessing

The preceding experimental results were obtained after an empirical research carried out to choose the optimal values of the α parameter (see Subsect. 4.5). The optimal values α_{SDS} and $\alpha_{\text{BPS+SDS}}$ for the SDS and BPS+SDS approaches were determined as follows.

We conducted experiments for both the SDS and the BPS+SDS approaches employing the test set constructed above for various α values, and counted the numbers of detected stickers and false alarms. We regarded a value of α as the optimal one if it met the following two requirements:

- the approach must be able to successfully recognize all stickers without missing even one;
- the approach must provide the minimum number of false alarms among all values of α.

The experiments showed that the SDS meets the first requirement only if $\alpha \leq 0.24$, whereas the BPS+SDS approach satisfies it if $\alpha \leq 0.36$. Then, according to figure 19, the SDS and the BPS+SDS approaches meet the second requirement if $\alpha_{SDS} = 0.24$ and $\alpha_{BPS+SDS} = 0.36$, respectively.

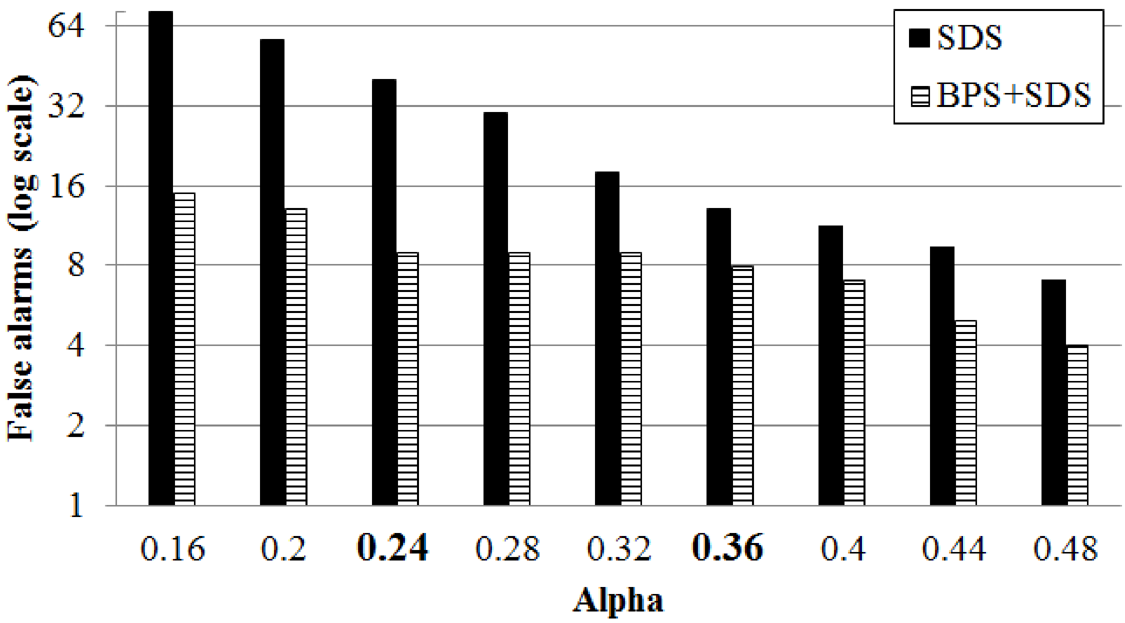

Fig. 19. Number of false alarms depending on the threshold α (lower is better)

Table 2. Comparison of approaches

Characteristics	SDS	BPS	BPS+SDS
Stickers detected	3	3	3
Stickers missed	0	0	0
False alarms activated	43	15	8

5.2. Comparison of detection results

In the experiments, we studied three approaches, namely BPS, SDS, and BPS+SDS. BPS implements an existing analytical algorithm. SDS is the approach in which the CNN substitutes the BPS. Finally, BPS+SDS means that the SDS works as an assistant of the BPS.

For the aforementioned approaches, we compared the following characteristics: the number of stickers detected, the number of stickers missed, and the number of false alarms. Table 2 contains the results of the experiments. Note that, for the SDS, the number of false alarms activated exceeds the overall number of cases in the whole archive. At the same time, for each of the approaches BPS and BPS+SDS, the number of false alarms exceeds the number of those cases in the archive in which the BPS either detected a sticker or gave a false alarm.

We can see that each approach successfully recognizes stickers without missing even one. In false alarms, the SDS performs worse. However, as for the number of false alarms, BPS+SDS performance is the best among all other approaches. Finally, the combination of the existing analytical algorithm and the CNN reduces the number of false alarms by 47 % compared with the existing system (namely by 7 out of 15 cases of false alarms).

CONCLUSIONS

In this thesis, we presented an approach to sticker detection in continuous casting. This approach is based on the use of a convolutional neural network (CNN) and was devised to reduce the number of false sticker alarms. We described the selection of data to train the CNN, preprocessing of selected data, data augmentation, and the construction of the CNN layers. The approach was implemented as a Sticker Detection System (SDS) in MATLAB. After that, the SDS was integrated with the HD mold^{FO} monitoring system.

We conducted experiments in which we compared three approaches: BPS, SDS, and BPS+SDS. BPS is an analytical algorithm embedded in the HD mold^{FO} system. SDS is the approach in which the CNN works alone while BPS remains inactive. In the BPS+SDS approach, the SDS analyzes only suspicious situations after sticker warnings received from BPS. All the approaches successfully detected all stickers. The BPS+SDS approach reduces the number of false alarms from BPS by 47 %. Throughout the tests, we got 8 false alarms out of 9,567 samples. Note that our solution outperforms the results obtained by similar methods. In contradistinction to previous studies, we implemented augmentation algorithms to expand the training set and improve results.

Here are the basic results we have achieved:

1. Analysis of continuous casting and monitoring methods.
2. Overview of breakout prediction and sticker detection algorithms.
3. Design of a convolutional neural network for sticker detection.
4. Design of a sticker detection system.
5. Implementation of the sticker detection system, which features the following functions:
   - automatic data preparation;
   - automatic data augmentation;
   - preprocessing of data;
   - automatic training of the CNN;
   - postprocessing of data.
6. Experiments to evaluate the effectiveness of the created system:
   - computation of optimal postprocessing parameters;

- comparison of detection results.

The results achieved in dissertation work were reported at the 2018 Global Smart Industry Conference (GloSIC 2018, November 13–15, 2018, Chelyabinsk, Russia) and published in the conference proceedings [5] (indexed in Scopus).

Currently, Sticker Detection System is in a trial operation phase in SMS group GmbH.

Further research might focus on such topics as the reduction of the number of false alarms of the CNN (this would enable the use of the Sticker Detection System independently of the BPS) and the development of a CNN for detection of longitudinal face cracks in steel during continuous casting.

REFERENCES

1. Bartos R. Steel Manual / R. Bartos, S. Brockmann, R. Fandrich, G. Endemann, S. Heinzel et al. – Verlag Stahleisen, Düsseldorf, 2015. – P. 218.

2. Bouhouche S. Inferential sensor-based adaptive principal components analysis of mould bath level for breakout defect detection and evaluation in continuous casting / S. Bouhouche, M. Zoheir, Z. Slimane, B. Jurgen // Applied Soft Computing. – 2015. – Vol. 34. – P. 120–128.

3. Brand T. Future-Oriented Modernization Concepts With Proven Technology for Continuous Casting Plants / T. Brand, I. Knopp, H.G. Braß, I. Hüllen, M. Rzepczyk et al. – 2018.

4. Emling W.H. Breakout Prevention // Modeling of Continuous Casting. – 11 ed. – AISE Steel Foundation, Pittsburg, PA, 2003.

5. Faizullin A. Use of Deep Learning for Sticker Detection During Continuous Casting / A. Faizullin, M. Zymbler, D. Lieftucht, F. Fanghänel // Proceedings of 2018 Global Smart Industry Conference. – GloSIC 2018. – Chelyabinsk, Russia : IEEE, 2018. – November 13–15. – P. Article no. 8570155.

6. HD mold. High definition – high performance. – 2016. – [Source link] URL: https://www.sms-group.com/sms-group/downloads/download-detail/22203/ (date of the request: 28.01.2019).

7. He F. Development and application of mould breakout prediction system with online thermal map for steel continuous casting / F. He, D.F. He, Z.H. Deng, A.J. Xu, N.Y. Tian // Ironmaking and Steelmaking. – 2015. – Vol. 42. – No. 3. – P. 194–208.

8. He F., Zhang L. Mold breakout prediction in slab continuous casting based on combined method of GA-BP neural network and logic rules // The International Journal of Advanced Manufacturing Technology. – 2018. – Vol. 95. – P. 4081–4089.

9. Krasilnikov A. Local heat transfer through mold flux film and optimal narrow face taper adjustment / A. Krasilnikov, F. Fanghänel, D. Lieftucht, M. Reifferscheid, J. Laughlin. – 2015. – Vol. 2. – P. 2485–2493.

10. Lieftucht D. HD mold - A Wide Variety of Caster Assistance Tools for Enhanced Product Quality / D. Lieftucht, M. Korzi, M. Arzberger, F. Fanghänel, M. Langer et al. – 2016.

11. Liu Y. Visual detection based on computer vision for sticker breakout in slab continuous casting / Y. Liu, X. D. Wang, F. M. Du, L. W. Kong, M. Yao et al. // Ironmaking and Steelmaking. – 2015. – Vol. 42. – P. 417–423.

12. Liu Y. Effect of casting parameters on sticker breakout and its propagation behaviour during slab continuous casting / Y. Liu, X. D. Wang, M. Yao, X. B. Zhang, H. Ma et al. // Ironmaking and Steelmaking. – 2014. – Vol. 41. – P. 748–755.

13. MATLAB Documentation. – 2019. – [Source link] URL: https://uk.mathworks.com/help/matlab/ (date of the request: 15.02.2019).

14. Meng Q. Using GA-BP Neural Network for Sticking Breakout Prediction in Continuous Slab Casting / Q. Meng, B. Li, J. Qi, C. Yao. – 2017.

15. Oquab M. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks / M. Oquab, L. Bottou, I. Laptev, J. Sivic // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. – 2014. – 06.

16. Python Docs. – 2019. – [Source link] URL: https://www.python.org/doc/ (date of the request: 09.02.2019).

17. Rawat W., Wang Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review // Neural Computation. – 2017. – Vol. 29. – P. 1–98.

18. Thomas B. On-line Detection of Quality Problems in Continuous Casting of Steel // Materials Science and Technology 2003 Meeting. – 2003.

19. Wolf M.M. Historical Aspects and Key Technologies // Modeling of Continuous Casting. – 11 ed. – AISE Steel Foundation, Pittsburg, PA, 2003.

20. Zhang B. Breakout prediction for continuous casting using genetic algorithm-based back propagation neural network model / B. Zhang, R. Zhang, G. Wang, L. Sun, Z. Zhang et al. // International Journal of Modelling. – 2012. – Vol. 16.

APPENDIX

Abbreviations

BPS	Breakout Prevention System
CC	Continuous Casting
CNN	Convolutional Neural Network
FO	Fiber Optic
HMI	Human-Machine Interface
NN	Neural Network
OPC	Open Platform Communications
PLC	Programming Logical Controller
ReLU	Rectified Linear Unit
SDS	Sticker Detection System

List of illustrations

List of tables