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)

This work 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).
Continuous casting

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

Breakout

- Sticker
- Liquid steel
- Solidified steel
- Breakout
- Mold
- Roll support

€250 000
The HD mold monitoring system

Application Server

Process Data Archiving (iba PDA) with OPC-Client

Level 1 Interface with OPC-Server

Drive PLC

SMS.FlexOPC Server

OpcCopy

HMI with OPC-Client

Sticker Detection System

Interrogator

Models with OPC-Client

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

576 fiber optic sensors

Heat map
Sticker

Temperature changes during continuous casting

Sticker area

Casting mold heat map

Sensor 1
Sensor 2
Sensor 3
An example of false alarm

Each false alarm is assessed by an expert.

Damage €1000+
Sticker detection system

**Sticker Detection System**

**Training module**
- Preparation of training set
- Learning of neural network

**Work module**
- Data analysis in real time
- Sticker alarms

**Convolutional neural network**

*Learning* → **Convolutional neural network** → *Analysis*
Data analysis

Data reading (in real time)

Data preprocessing

Data analysis (Neural Network)

Data postprocessing

\[
P_{stickers} = P_1, P_2, P_3, ..., P_N
\]
Data reading (in real time)

Casting mold

Left side

Loose side

Right side

Fixed side

C°
Preprocessing: data cleaning

Smooth:

\[ T = \min(\max(T, T_{avg} - \delta), T_{avg} + \delta) \]
Preprocessing: normalization

\[ T_{\text{norm}} = \frac{T - T_{\text{min}}}{T_{\text{max}} - T_{\text{min}}} \]

Normalized data
Preprocessing: slicing

The Use of Deep Learning for Sticker Detection During Continuous Casting
Data sample as an input for CNN

Before preprocessing

\[ F = 30, \text{ frames} \]
\[ W = 32, \text{ sensors} \]
\[ H = 18, \text{ sensors} \]

After preprocessing

\[ F = 30, \text{ frames} \]
\[ h = 9, \text{ sensors} \]
\[ w = 5, \text{ sensors} \]

\[ F = 30 – \text{ number of frames where gap between frames is 0.25 sec} \]
Structure of CNN

- **Input data** are preprocessed samples of 7.5 sec
- **Output data** are probabilities of “sticker” and “not sticker” events
Training set (Archive of SMS group)

- **14** sticker cases
- **103** false alarm cases
- **∞** regular work cases
Data augmentation

**Transferring**

+ 6307 samples \( W \times H \)

**Mirroring**

\( \times 2 \) samples \( w \times h \)
### Augmented training set

<table>
<thead>
<tr>
<th>Case</th>
<th>Cases</th>
<th>Samples for CNN</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Real</td>
<td>Synthetic</td>
<td></td>
</tr>
<tr>
<td><strong>Sticker</strong></td>
<td>11</td>
<td>6,307</td>
<td>~25,000</td>
</tr>
<tr>
<td><strong>False alarm</strong></td>
<td>88</td>
<td>2,384</td>
<td>~7,000</td>
</tr>
<tr>
<td><strong>Regular work</strong></td>
<td>11,701</td>
<td>0</td>
<td>~30,000</td>
</tr>
</tbody>
</table>
If $\alpha$ is greater than (empirically found) threshold then sticker is detected.
Results

Test set:
- 3 real sticker cases
- 15 false alarm cases
- 9,567 regular work cases

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>BPS</th>
<th>BPS+SDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stickers detected</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Stickers missed</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>False alarms signaled</td>
<td>15</td>
<td>8</td>
</tr>
</tbody>
</table>

Saving €50,000+ per year