

# Synthetic Images of Longitudinal Cracks in Stainless Steel Slabs via Wasserstein Generative Adversarial Networks Used Toward Unsupervised Classification

Digital technologies are transforming industry at all levels. Steel has the opportunity to lead all heavy industries as an early adopter of specific digital technologies to improve our sustainability and competitiveness. This column is part of AIST's strategy to become the epicenter for steel's digital transformation, by providing a variety of platforms to showcase and disseminate Industry 4.0 knowledge specific for steel manufacturing, from big-picture concepts to specific processes.



## Authors

**Diego Andrade** (top row, left)  
Chief Scientist, ANT, Tampa, Fla., USA  
diego.andrade@ant-automation.com

**Miguel Simiand** (top row, right)  
Senior Project Manager, ANT,  
Pittsburgh, Pa., USA  
miguel.simiand@ant-automation.com

**Javier Barreiro** (bottom)  
Chief Executive Officer, ANT,  
Pittsburgh, Pa., USA  
javier.barreiro@ant-automation.com

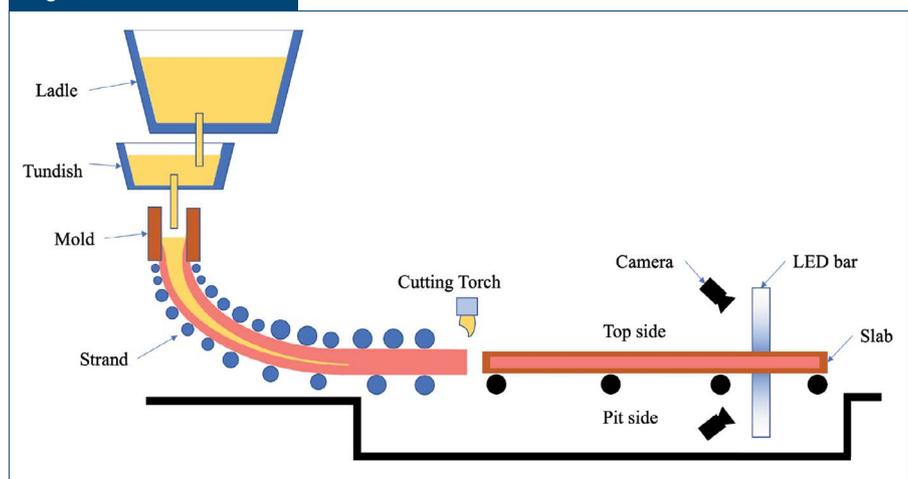
Primary stainless steel production for an integrated steel mill includes ironmaking, steelmaking, casting, rough rolling and product rolling. While all of these steps have their inherent problems, this paper will focus on those manifested in the slabs after the cutting torch (see Fig. 1). Surface defects on continuously cast slabs require treatment by grinding. This new phase in the process causes lower throughput of the final product and additional energy costs. The defects during rolling are carried out through the entire line of production when not detected early. The problems do not arise immediately; the fractures or dislocation continue with the rolling downstream. Where smaller and smaller thicknesses make them appreciable, they can waste significant time and money for any given production line. A low-tech, low-cost, effective solution is a visual inspection of the slab surfaces during the production along with progressive steps during rolling, which requires

constant conditional awareness by qualified operators.

In recent years, with the advent of machine-learning (ML) algorithms for annotation, classification and object detection, exciting applications have been proven. This project uses two technologies that permit and train models with fewer images of longitudinal crack detection. Synthetic images were created using Wasserstein Generative Adversarial Nets (WGAN)<sup>1-4</sup> and then new defects were captured in real time using You Only Look Once (YOLO) framework.<sup>5,6</sup> A research opportunity is presented here, situated at the convergence point of steel production, computer vision, automation and machine learning.

**Problem Statement** — Finding an adequate number of well-labeled surface defects is a challenge for the construction of well-maintained frameworks for ML classification. In the case of object detection and classification for stainless steel, hot

Figure 1



*The slab production process and image acquisition location.*

rolling will deal with hot surfaces of metals with high reflectivity that are linked directly to light settings, surface structure, temperature and camera settings. Changing any of these parameters affects steel production or data gathering.

Visual inspection of steel slabs by an inspector is a difficult task in most cases. There are many adverse factors such as the ambient temperature, large slab area and access to the bottom face of the slabs, which make the visual inspection unreliable. Also, human factors affect production; for example, working through night shifts, attention drawn to other events, subjectivity in terms of defect classification, and a short time frame to make decisions due to continuous production line speeds.

In practice, gathering proper data of surface defects is time-consuming and cost-prohibitive because of operator's fatigue, and since not that many occur, camera setup does not capture them (i.e., improper lighting and other restrictive effects). It is a tedious job for operators to produce reliable tracking of surface defects. The problem centers on how to effectively apply and gather data from a constrictive number of examples and extrapolate the "meaning" of a steel surface defect to new and unseen examples. From there, another essential piece of this system is real-time defect detection and annotation for future learning.

It is believed that the excessive time allocated for longitudinal crack localization depends on five core limitations that make object detection difficult when using current machine-learning models. First, these patterns in the slabs do not always follow a given position and size across the slab (see Fig. 2). Second, lighting setup can produce extreme cases for any defect in the surface of the steel slabs. Third, the frequency of the defects is minimal, so training takes excessive time (i.e., months) because of the lack of well-labeled data used for object detection. Fourth, real-time object detection of longitudinal cracks is a very domain-specific problem; a handful of applications exist, so picking the right model is essential. Finally, the fifth limitation is ground truth selection from a set of real images, which is not a trivial task and requires patience and knowledge.

This paper will show how to create a countless number

of examples to train a ML model when the training examples are limited.

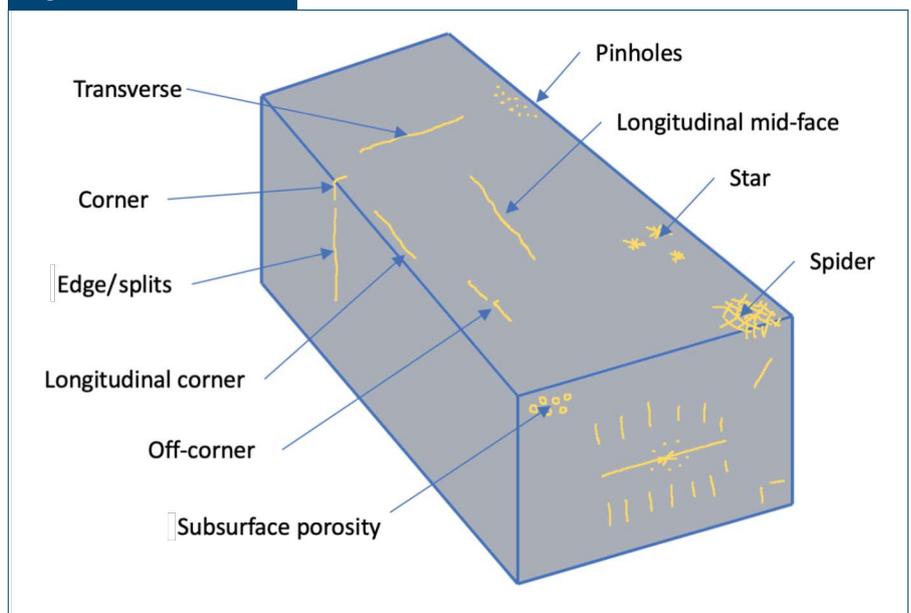
## Related Work

Classification and object detection for steel production need excellent imaging and properly labeled data in the search space. Imaging techniques used in the industry are one of two methods. The first method deals with gray-level intensity imaging, while the second method uses range imaging.<sup>8</sup> These techniques capture the failures of the steel surfaces and create a library of defects for later use.

Several groups have tackled the problem of object detection using Bayesian network classifiers.<sup>9–12</sup> Computer vision application in surface inspection systems is essential, and they are utilized for defect detection and classification in manufacturing. The downside of some of these classification systems is that they are feature dependent, requiring extensive domain knowledge to perform well. They are unique and do not translate well to other domains. Deep learning-based methods (e.g., convolutional neural networks) are feasible for use in surface inspection systems and outperform traditional methods in accuracy and inference time by considerable margins.<sup>12</sup>

Friedman et al. discussed supervised learning with naive Bayesian classifiers, where a strong assumption of independence among features is contentious with state-of-the-art classifiers. Bayesian networks are

Figure 2



Major crack types found on "as-cast" semi-processed steel slab (Source: Veitch-Michaelis et al.<sup>7</sup>).

representations of probability distributions that generalize the naive Bayesian classifier while representing statements about independence. They based their studies in the use of Tree Augmented Naive Bayes (TAN), which outperforms naïve Bayes, due to its computational simplicity (not requiring search) and the robustness of naïve Bayes.<sup>13</sup>

Franz Pernkopf proposed an approach for detecting surface defects with three-dimensional characteristics on scale-covered steel blocks. In flawless surfaces, their reflection properties change sharply. A technique called light sectioning was used as part of the range imaging capture for steel blocks. A depth map is obtained, and then segments of the surface are classified following a set of extracted features utilizing Bayesian network classifiers. For establishing the structure of these Bayesian networks, a search algorithm was applied that tackles the issues of performance and efficiency for structure learning, which achieves a good trade-off between classification performance and computational efficiency for structure learning. Their experiments show that their selective unrestricted Bayesian network classifier outperforms the naïve Bayes and the tree-augmented naïve Bayes decision rules in the case of classification rate.

Yushi Jing et al. presented a framework to combine discriminative data-weighting with generative training of intermediate models; they call it Boosted Bayesian Network Classifiers. They show that this type of classifier includes the basic generative models in isolation while improving classification performance for the suboptimal model.<sup>10</sup>

**Convolutional Neural Networks** — Convolutional neural networks (CNN) are specialized kinds of multi-layer neural networks, devised to categorize visual patterns straight from raw images with minimal pre-processing as well as for image recognition, achieving impressive recognition rates in image classification tasks. The main advantage of CNN compared to its predecessors is that it automatically detects the essential features without any human supervision. Also, generally, CNN are computationally efficient, using convolution and pooling operations and performs parameter sharing. In turn, they are enabling CNN models to run on any device, making them universally attractive and performing automatic feature extraction to achieve superhuman accuracy.

CNNs are in use for demanding large-scale projects but lack real-time applications, and they are also domain-specific (because of the lack of data sets). Even though CNNs are used in object detection and image classification tasks, industrial surface inspection systems barely utilize their potential.<sup>12</sup>

Masci et al. presented a Max-Pooling Convolutional Neural Network approach for supervised steel defect classification. On a classification task with seven

defects collected from a real production line, a low error rate of 7% was obtained. They had good results in comparison with support vector machine (SVM) methods; when also using the train nets, their solution was deployed directly on raw images, which translates into a time reduction.<sup>11</sup>

**Generative Adversarial Networks** — Generative adversarial networks (GANs) are a class of deep generative models that aim to learn a target in an unsupervised fashion.<sup>14</sup> It appears that deep generative models are likely to represent the world around us from labeled data, similar to how humans developed complex mental models in an unsupervised way, directly from sensory experience.<sup>15</sup> Deep generative models are a dominant class of unsupervised machine-learning models. The robust models are applied in a variety of applications, including image generation, super-resolution, text to image, text to image synthesis, image in-painting, texture synthesis, image editing, object detection, music generation, medical anomaly detection and learned compression. GANs provide a way to learn deep representations without extensively annotated training data.<sup>16</sup>

GANs, as first described by Goodfellow,<sup>2</sup> are one of the most popular approaches to learning in a fully unsupervised fashion. In GANs, one network produces a rich, high-dimensional vector that is used as input in another network, and attempts to choose an input that the other network does not know how to process. This framework also has been derived by minimizing a divergence between the model distribution and the correct distribution.<sup>4</sup>

A GAN framework consists of a two-player game where the first player, the generator, is learning to transform some simple input distribution (usually a standard multi-variate normal or uniform distribution) to a distribution on the image space (see Fig. 3) such that the second player, the discriminator, cannot tell where the samples belong or until neither player can improve their loss unilaterally. The discriminator examines real images (training process) while generating images independently. Then there is a judgment whether the input images are real or generated. The framework concludes with an output probability  $P(x)$ , where  $P$  is the probability distribution that the image  $x$  is real or not. If the input is real  $P(x) = 1$ . If the input image is generated, the results are 0. The discriminator finds the proper allocation of features through the process of identifying the real images. At the same time, we want to create images that are close to the distribution  $P(x) = 1$ ; in other words, match the real images. According to Reference 17, there is no evidence that new GAN algorithms in use and tested outperform the non-saturating GAN introduced by Goodfellow,<sup>2</sup> so the same architecture was used in the system.

Learning the probability distribution means learning the probability density. This is often done by defining a parametric family of densities  $(P_\theta)_{\theta \in \mathbb{R}^d}$  and finding the one that maximized the likelihood on our data: if we have real data examples

$$\{x^{(i)}\}_{i=1}^m, \text{ then the problem is to solve } \max_{\theta \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^m \log P_\theta(x^{(i)}).$$

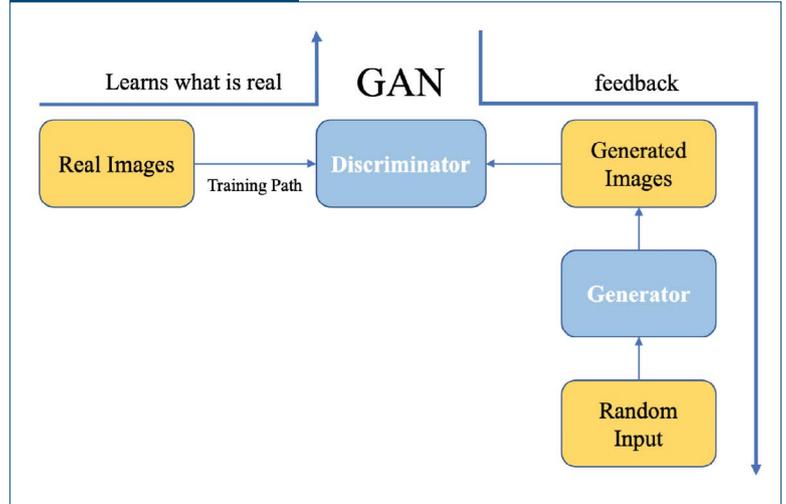
The advances in GAN have been for the most part empirically driven, making it essential for the use of high-quality evaluation metrics.<sup>15</sup> Wasserstein is a distance function defined between probability distributions on a given metric space  $M$ . Intuitively, if each distribution is viewed as a unit amount of “dirt” piled on  $M$ , the metric is the minimum “cost” of turning one pile into the other, which is assumed to be the amount of dirt that needs to be moved times the mean distance it has to be moved. Because of this analogy, the metric is known in computer science as the earth mover’s distance.

The fact that the  $M$  distance is continuous and differentiable means that the critic till optimality can be trained. The argument is simple: the more the critic is trained, the more reliable the Wasserstein gradient, which is actually useful by the fact that Wasserstein is differentiable almost everywhere. For more in-depth details, refer to Arjovsky et al.<sup>4</sup>

It is the case that the model manifold and the correct distribution’s support have a non-negligible intersection, the typical remedy is to add noise term to the model distribution. This is why, in the literature, a noise component exists. The optimal standard deviation of the noise added to the model when maximized is around 0.1 to each pixel in the generated image, after normalizing the pixels between 0 and 1.<sup>4</sup>

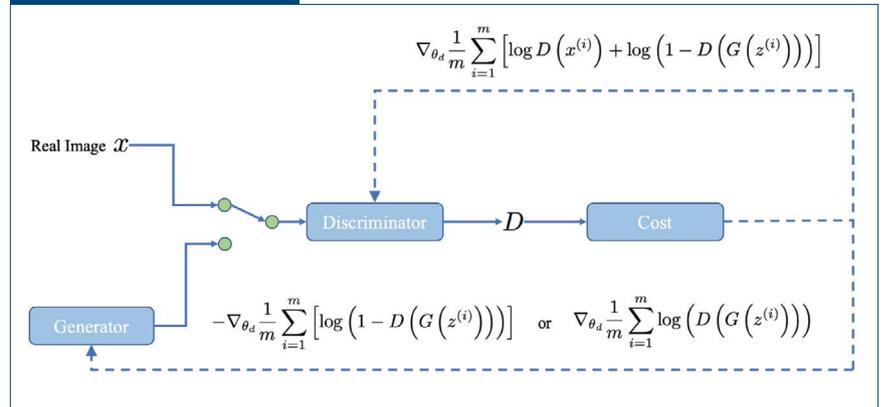
**Backpropagation:** The goal of the system is to maximize the probability of recognizing real images as real and generated images as fake images. The measurement uses cross-entropy  $p \log(q)$ . For real images,  $p$  (the true label for real images) equals 1. For generated images, the label is reversed (i.e., one minus label). So the objective becomes:

Figure 3



Example of a generative adversarial network (GAN) architecture.

Figure 4



Summarizes the data flow and the gradients when backpropagation is applied.

$$\max_D V(D) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

recognizes real images                      recognizes generated images

where  $D$  and  $G$  are the probability distribution for the discriminator and the generator, respectively.

On the generator side, its objective function wants the model to generate images with the highest possible value of  $D(x)$  to fool the discriminator,

$$\min_G V(G) = E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

Optimizes G to trick the discriminator

as a minmax game in which  $G$  wants to minimize  $V$  while  $D$  wants to maximize it.

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

Gradient descent is used to optimize both objective functions (see Fig. 4). The iterations in the model are

made in a staggered fashion; first, the generator model parameters are locked and perform a single iteration on the discriminator using real images. After this step, the focus shifts to the generator locking parameters on the discriminator side. The generator training stage uses backpropagation that targets the discriminator into thinking the images are real. These two networks are trained in a staggered fashion, fighting to improve themselves.

The pseudo-code in Algorithm 1 shows how a WGAN is trained.

#### Generator Diminished Gradient:

However, a gradient diminishing problem is encountered for the generator. The discriminator usually wins early against the generator. It is always easier to distinguish the generated images from real images in early training. That makes  $V$  approaches 0, i.e.,  $\log(1 - D(G(z))) \rightarrow 0$ . The gradient for the generator will also vanish which makes the gradient descent optimization very slow. To improve that, the GAN provides an alternative function to backpropagate the gradient to the generator.

#### Proposed Approach

Surface inspection and the defect detection problem can be generalized into the combination of feature extraction and classification problems.<sup>12</sup> The techniques shown above make use of hundreds to thousands of examples. In contrast, this method uses a couple hundred well-labeled images to create new data, using generative adversarial networks, from there the new data is used to train the YOLO framework to have a real-time application finding surface errors, this two techniques are well used in other applications.<sup>5,6</sup>

The method used occurs in four steps: (1) collecting data, (2) data analysis, (3) data synthesis and (4) data classification, as shown in Fig. 5.

Developing automatic detection and classification of surface defects for stainless steel slabs has been a challenging problem for the steel manufacturing industry. The block diagram shown in Fig. 5 represents the steps to fulfill this task automatically.

#### Algorithm 1

WGAN, the algorithm first implemented by Arjovsky.<sup>4</sup> All experiments in the paper used the default values  $\alpha = 0.00005$ ,  $c = 0.01$ ,  $m = 64$ ,  $n_{\text{critic}} = 5$ .

Require:  $\alpha$ , the learning rate.  $c$ , the clipping parameter.  $m$ , the batch size.  $n_{\text{critic}}$ , the number of iterations of the critic per generator iteration.  $w_0$ , the initial critic parameters.  $\theta_0$ , the initial generator's parameters.

Ensure: The batch of real data images was normalized

```

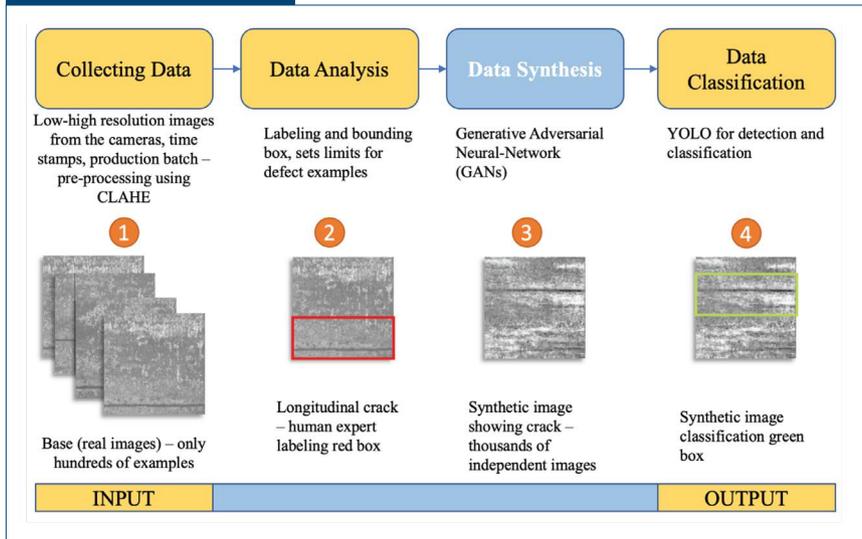
1: while  $\theta_0$  has not converged do
2:   for  $\tau = 0, n_{\text{critic}}$  do
3:     Sample  $\{x^{(i)}\}_{i=1}^m$   $P_r$  a batch from the real data
4:     Sample  $\{z^{(i)}\}_{i=1}^m$   $p(z)$  a batch of prior samples
5:      $g_w = \nabla_w \left[ \frac{1}{m} \sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)})) \right]$ 
6:      $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$ 
7:      $w \leftarrow \text{clip}(w, -c, c)$ 
8:   end for
9:   Sample  $\{z^{(i)}\}_{i=1}^m$   $p(z)$  a batch of prior samples
10:   $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)}))$ 
11:   $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$ 
12: end while

```

A large amount of information is gathered during this process, mainly in the form of images. The system evaluates the images and classifies them by family, type and severity, then suggests actions to be taken to overcome any defects detected. This information, along with the high-definition images, are presented to the supervisor to make the final decision. The automatic system learns over time, using the operator acknowledgment and retraining a classification neural network periodically.

**Data Collection** — The computer vision recognition system uses two ultrahigh-resolution line scan cameras, one for each slab side. The images are taken immediately after the product leaves the cutting machine. The slab produced is marked with identification for tracking purposes. The camera's location after the slab marker has many advantages, as defects can be detected early, straight after the slab identification number is generated. The presence of a pit in this section of the line facilitates the installation of the bottom camera. Both camera housings were designed with refrigeration to maintain operating

Figure 5



The process consists of four steps: (1) the user specifies a given set of examples (hundreds) after image pre-processing is applied; (2) the operator selects and labels the approximate location and dimensions of the longitudinal crack; (3) data synthesis occurs within the generative adversarial network; (4) finally, a set of synthetic images (tens of thousands) is the output ready for classification using the YOLO framework.

temperatures within the specification range. Also, swivel arms are part of the structure to properly align the camera, pointing to the slab surface (see Fig. 6).

The lighting technology used is LED as it has larger mean-time-between-failure (MTBF) than other light sources and lower power consumption and provides uniform illumination. Refrigerated lighting housing was designed and installed to protect light bars from radiation heat coming from the hot slabs and

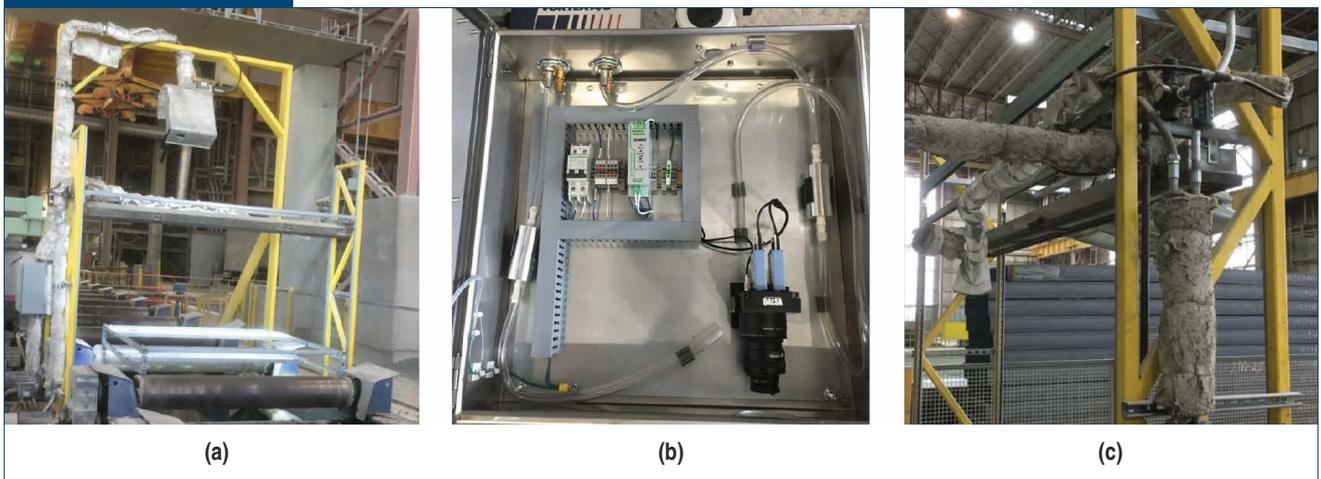
The slab thickness is a value given by the production order. The pixels per millimeters factor is obtained during the camera adjustment process. To do this, a calibration pattern was designed, which is also used to center and align the camera.

After the dimensions are computed, five different features are evaluated:

to enhance light lifetime. A typical slab is between 1.5 and 2.0 m width by approximately 12 m long. In order to get ultrahigh-definition images, line scan cameras with high line rates are used. With the technology used, 36 pixels/mm<sup>2</sup> images were obtained for slabs up to 2 m wide and maximum line speed of 2.5 m/second.

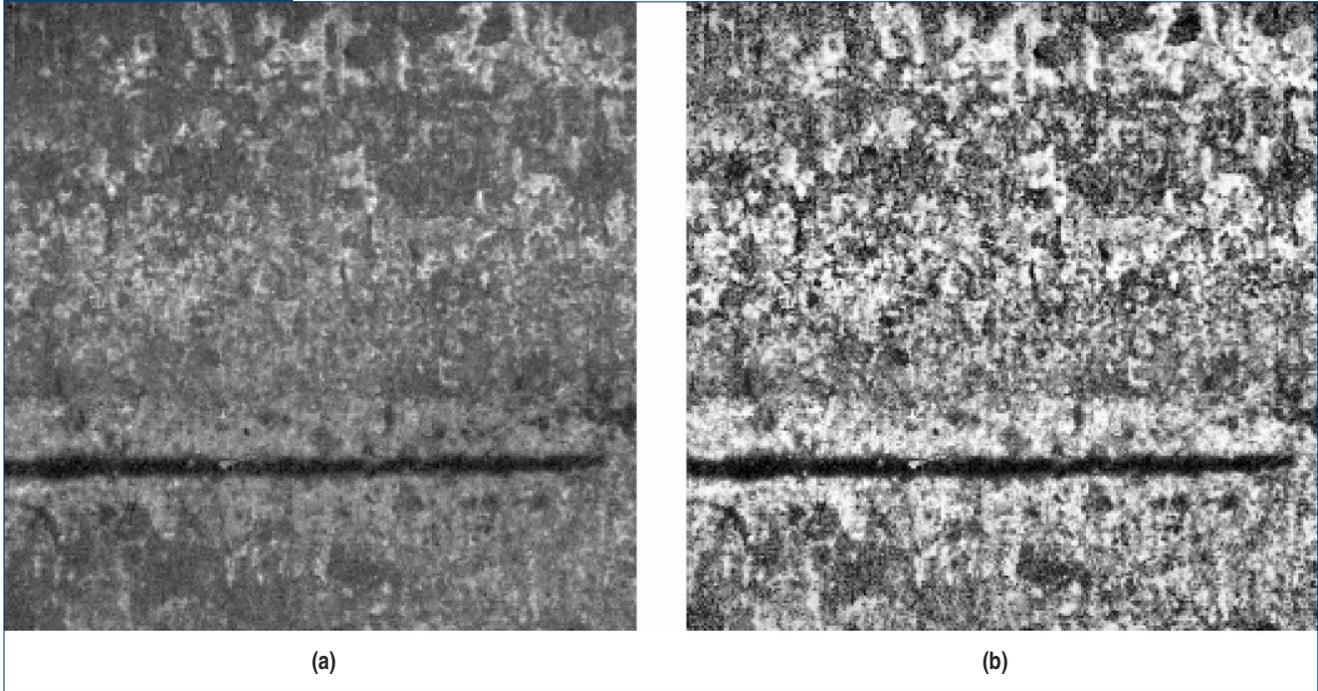
**Data Analysis** — The data analysis is divided into shape or geometric defects and surface or textural defects. Each analysis uses different mathematical tools for detection and classification. The slab's dimensions are calculated in order to ensure compliance with the production order and to guarantee uniformity. After the slab edges are identified in the images, to be able to compute the real dimensions, it is required to know the pixels per millimeters factor and also the slab thickness.

Figure 6



Camera location on the top side of the slab area before the cutting torch (a); camera with a minimum number of components on the field. Internal panel, custom positive pressure and air-cooled enclosures configurations (b); and LED bar, camera heat shield and camera insulation (c).

Figure 7



Original image (a) and Image after applying the contrast limited adaptive histogram equalization pre-processing method (b).

- Average width.
- Tail width.
- Center width.
- Head width.
- Uniformity width.

**Pre-Processing** — The images obtained are in black and white and they require pre-processing. One of the techniques used improves the contrast in the images. Such a technique is known as adaptive histogram equalization (AHE, see Fig. 7). It uses different histograms in the image and combines them to redistribute the lightness values of the image. This technique improves the local contrast, meanwhile enhancing the edge definition on the objects of the image. One of the drawbacks to AHE is noise amplification for homogeneous regions in the image, which is the case for steel production. The most common variant to this technique is the use of contrast limited adaptive histogram equalization (CLAHE), which prevents noise by limiting the amplification.

**Data Synthesis and Classification** — Krizhevsky et al. introduced several data augmentation techniques to artificially increase the data set size using label-preserving transformations.<sup>18</sup> To have more variety in data, rather than only modifying the images, it is desirable to create new samples to expand the data sets. Goodfellow et al. introduced GANs to use neural networks to generate new samples using adversarial training.<sup>1-3</sup> Using the conditional GAN, compared

to other domain-specific methods, Isola et al. introduced a general-purpose paired image translation method also known as pix2pix.<sup>19</sup> Because obtaining paired image data is expensive and difficult, Zhu et al. introduced a cycle-consistent adversarial network architecture called CycleGAN for unpaired image translation problems.<sup>20</sup> With the advancements of synthetic image generation, it has become a common practice to use generated images in training neural networks to avoid the high cost of creating large data sets with real images. Shrivastava et al. introduced an improved approach to image generation with Simulated+Unsupervised learning (SimGAN) which uses synthetic images rather than random vectors as inputs to their GAN.<sup>21</sup> By using a self-regularization term and a local adversarial loss, SimGAN converts synthetic renderings into realistic images without using any labeled data.<sup>21</sup> Their method is able to achieve local changes without altering the global structure of the image. In contrast, a data augmentation method is proposed for altering global scene composition in the image WGAN.<sup>4</sup>

Local defects are limited in space but may appear in a discontinuous fashion on different places and different shapes on the surface (i.e., scratches, cracks, ruptures, blisters and bruises). On the other hand, distributed defects are spread over the large area of the surface and may appear in a continuous pattern.<sup>22</sup>

**Features Detection:** A fast objection approach is utilized that was used by Redmon in his YOLO

implementations. It uses a regression problem to separate especially bounding boxes and its associate class probabilities. A single neural network assigns the bounding boxes and its classification from full images in only one pass. YOLO is extremely fast, working in real time between 45 frames per second (fps) and 155 fps. More details can be found directly in Redmon's references.<sup>5,6,23,24</sup>

**User Interface** — The graphical user interface (GUI) is an important feature designed with simplicity, performance and usability in mind. The system has a full and complete interaction with the operators/users, utilizing a GUI implemented in responsive HTML5 + CC3 + JavaScript. It works efficiently across all major desktop platforms using any of the best-known internet browsers on the market. The GUI allows the user to access real-time and historical data, with slab tracking identification and inspect in ultrahigh-definition imaging (see Fig. 8). The operator is able to mark slabs, mark region of interest and defects, as well as to pre-classify defects when detected, zoom in and zoom out slabs to visualize from 1 mm to meters. In real time after any slab is produced, the system evaluates it and presents at least the following data to the operator:

- Shape information (width and length calculated).
- Defects (classified and action required).
- High-definition image of both slab faces.

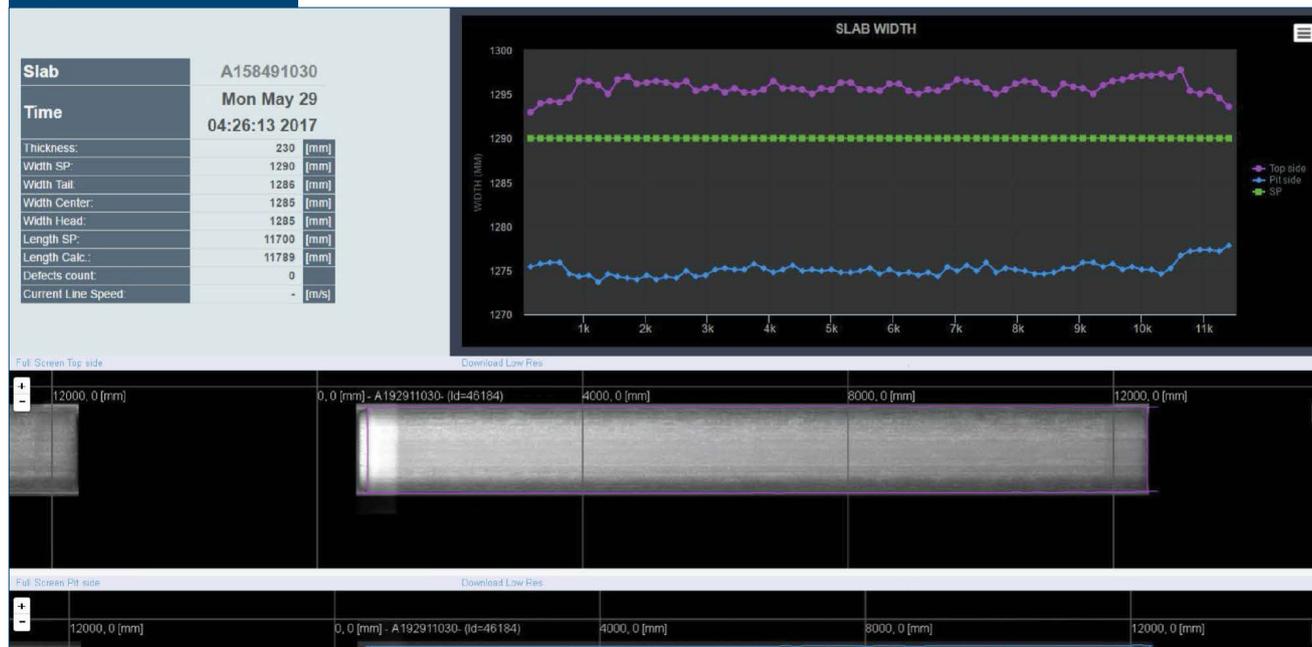
## Results and Discussion

In this project, a visual analysis tool designed to classify longitudinal cracks and to capture sets of images to train a GAN was developed. The final goal is on-line for real-time automatic defect detection and classification. The whole system uses a server running the Windows 10 operating system with a 3.9 GHz Intel Core i3-7100 and 16 GB of RAM. The lighting/camera rig was designed and manufactured by ANT. The camera is a line scan camera — Teledyne Dalsa P3-8X-12K40 Camera Link.

The system was trained with images acquired through linear cameras with a resolution of 12,288 x 1 pixels, and a pixel size of 5  $\mu\text{m}$ , maximum data rate of 8 x 40 Mhz and a maximum line rate of 33.7 kHz. The responsivity of this camera is presented in Fig. 9. The significant advancement in camera technology in recent years along with the reduction of its cost will allow for the use of more sensitive cameras with high dynamic range and multi-line technology in future applications with this same platform to improve further the quality of the images that feed the GAN network presented in this paper.

The current linear cameras with technology multi-line CMOS Time Delay & Integration (TDI) delivers the best performance, a combination of high speed and responsivity with low noise. Over a multi-line CMOS sensor, the image is integrated over multiple adjacent lines as it moves over the sensor. All these lines are combined to produce a highly responsive

Figure 8



Screen capture of the user interface in real time.

output. Thus, to achieve a correct summing, the image motion across the sensor must be synchronized to the sensor time integrator. When the speed is not constant, a synchronizing external trigger pulse — derived from an encoder that generates one pulse for one object pixel of motion — must be provided. The encoder signal must be connected to the encoder input of the frame grabber. Better responsivity requires less light power, which makes it easier to achieve uniformity of light under different ambient conditions. It is a fundamental factor that high acquisition frequency requires much light power; the frequency is linearly related to the camera resolution.

Ultrahigh-resolution cameras, image analysis and current internet tools were combined to build a fully versatile system for defect detection, classification and visualization. The full system uses two cameras and two light sources — the minimum hardware requirement to inspect both slab sides. Other companies have systems with similar capabilities but are made up of up to eight cameras, which implies more significant infrastructure, higher maintenance, more wiring, maintenance and more processing units. The system enabled digital supervised inspection, achieving resolutions of 36 pixels/mm<sup>2</sup>, which is an adequate resolution to detect small stainless steel surface defects. Currently, and during a six-month period, the system collected information to build the first image data set to be used for the GAN supervised learning. The GUI allows the plant operators and quality team to add new markers to continuously retrain to the system with surface defects in order to improve accuracy in the detection. Nowadays, the hardware capabilities are incredibly powerful, which enables obtaining and processing of high-resolution images in real time. As per the resolution achieved and due to the flexibility

of the supervised learning of the GAN, this technology could be applied to carbon steel slabs, where the size of the defects is more significant than in stainless steel slabs.

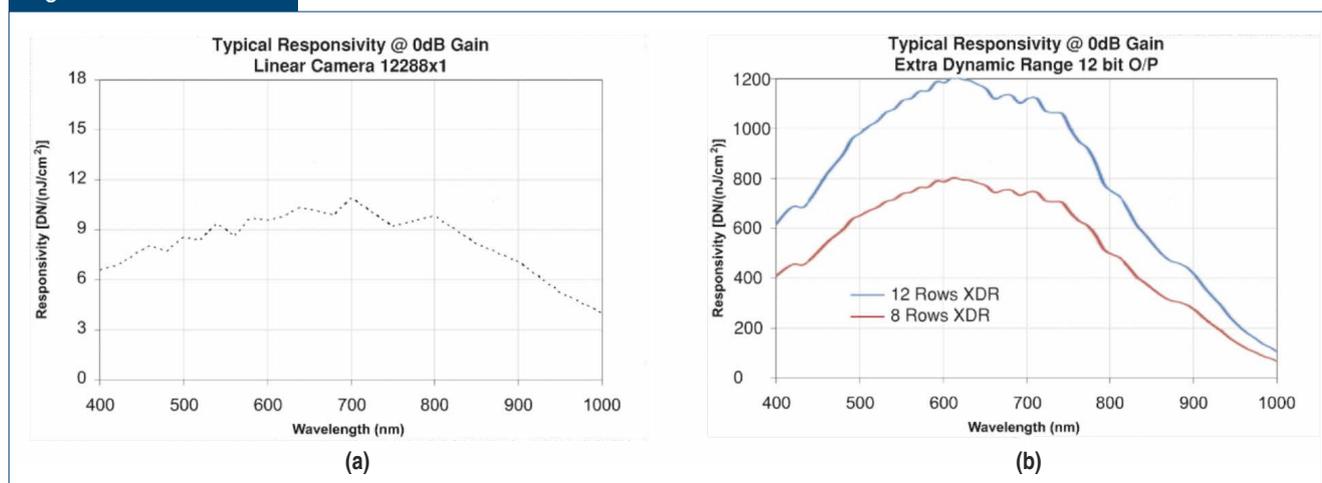
Detecting defects upon the creation of stainless steel slabs at continuous casting can avoid future product degradation and product losses. If the defect is detected early, countermeasures are available to fix it on time. Ultrahigh-resolution images on both sides of the slab are processed at high speed, taking advantage of modern GPU and parallel programming. This system designing goals and main objectives are:

- Decrease non-quality.
- Improve quality control's efficiency while reducing its cost.
- Anomalies detection, defect location and classification.
- Avoid defective slabs to be processed in upstream lines.
- A smart system – able to add and learn defects.

## Conclusions

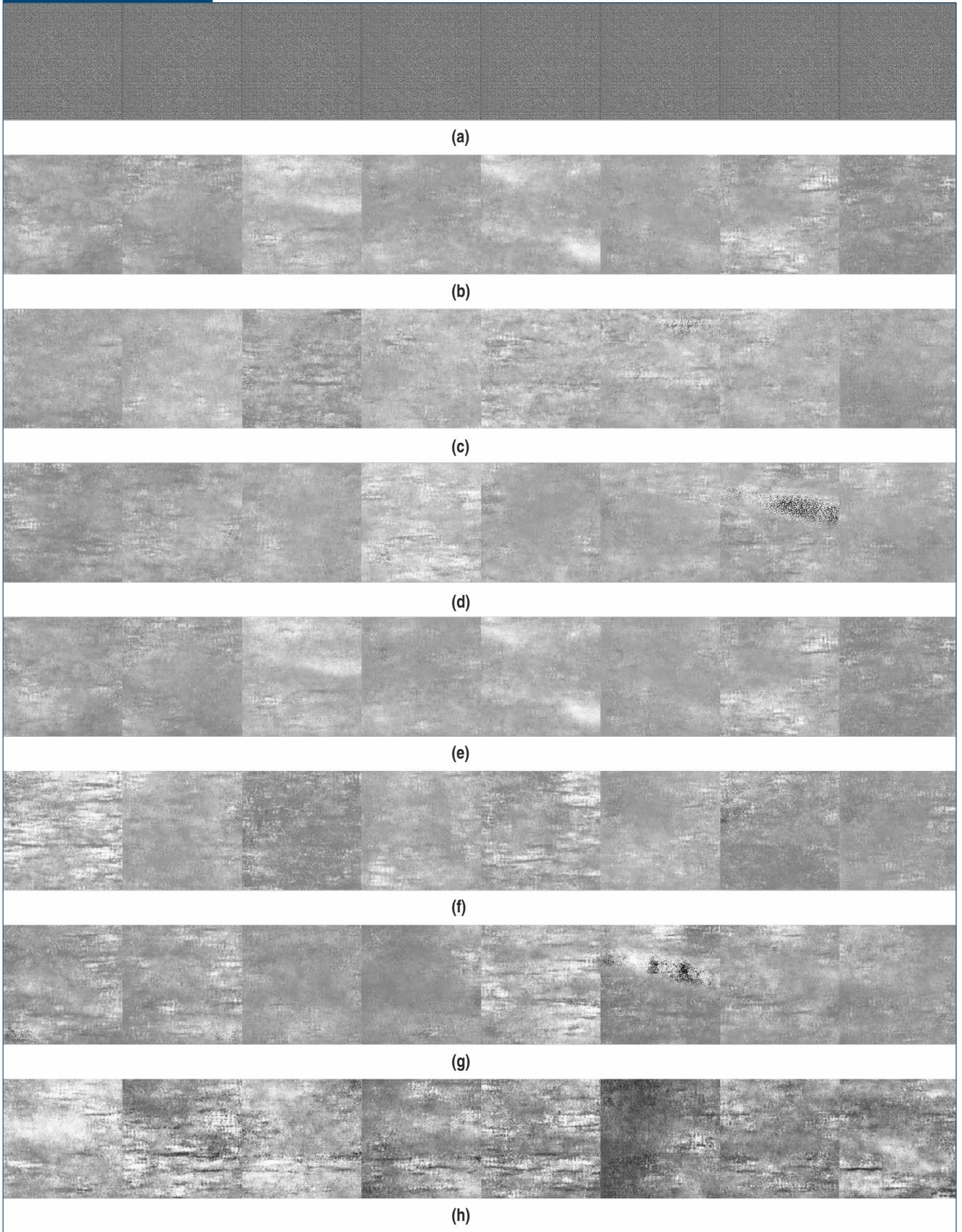
Wasserstein GAN (WGAN) is an essential alternative to traditional GAN training. They showed improved stability for learning, contain problems like mode collapse, and produce meaningful learning curves helpful for debugging and hyperparameter searches, and showed that the corresponding optimization problem is reliable. In the long run, the discriminator can identify tiny differences between the real images and the generated, while the generator shows images that the discriminator cannot discern if they are real ones

Figure 9



Camera responsivity comparison for linear cameras and extra dynamic range cameras with CMOS Time Delay & Integration.

Figure 10



*The stages for random outputs of the GAN: (a) is the latent space (i.e. noise), (b) is the first batch of solutions at 1,000 epochs, (c) shows solutions at 3,000 epochs, (d) shows solutions at 5,000 epochs, (e) shows solutions at 7,000 epochs, (f) shows solutions at 12,000 epochs, (h) shows the final solutions reached after 15,000 epochs.*

or not. Finally, WGAN models, as described, converge into natural-looking images, as shown in Fig. 10h.

Despite a widespread recognition that generative models lie at the frontier of artificial intelligence research, it remains notoriously challenging to evaluate solutions with the inception score metric that has gained popularity to assess the quality of generative models to create synthetic images.<sup>25</sup> In other words, there needs to be a way of effectively evaluating results, which remains a weakness in this approach, not because a solution is not converging but because it needs a ground truth beyond visually similar longitudinal cracks to the GAN output. A collaboration with experts in metallurgy and steelmaking is envisioned, in which the system can be deployed and tested.

Using this method, strong modeling performance and stability were displayed across a variety of hyperparameters. Now that there is a more stable algorithm for training GANs, it is hoped that this work will open the path for more robust modeling performance on large-scale image data sets.

Even though classifications were achieved using YOLO in real time, this classification is not above human operators. There are two main ideas with respect to the future direction of this work: first, to consider better data image sets, which are only possible with the industry participation in the form of process engineers, operators and other experts; and second, to keep trying YOLO on these new sets of synthetic images, achieving better and better fidelity on real-time classification.

As per the resolution achieved and due to the flexibility of the unsupervised learning of the CNNs, this technology can also be applied to carbon steel slabs, where the defect sizes are larger than in stainless steel slabs.

## References

1. I. Goodfellow, "Nips 2016 Tutorial: Generative Adversarial Networks," 2017.
2. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Nets," *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.
3. I. Goodfellow, J. Shlens and C. Szegedy, "Explaining and Harnessing Adversarial Examples," arXiv preprint arXiv:1412.6572, 2014.
4. M. Arjovsky, S. Chintala and L. Bottou, "Wasserstein Generative Adversarial Networks," *International Conference on Machine Learning*, 2017, pp. 214–223.
5. J. Redmon and A. Farhadi, "Yolo9000: Better, Faster, Stronger," 2016, arXiv preprint arXiv:1612.08242.
6. J. Redmon and A. Farhadi, "Yolov3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.
7. J. Veitch-Michaelis, Y. Tao, D. Walton, J. Muller, B. Crutchley, J. Storey, C. Paterson and A. Chown, "Crack Detection in "As-Cast" Steel Using Laser Triangulation and Machine Learning," *13th Conference on Computer and Robot Vision*, IEEE, 2016, pp. 342–349.
8. F. Pernkopf and P. O'Leary, "Image Acquisition Techniques for Automatic Visual Inspection of Metallic Surfaces," *NDT & E International*, Vol. 36, No. 8, 2003, pp. 609–617.
9. F. Pernkopf, "Detection of Surface Defects on Raw Steel Blocks Using Bayesian Network Classifiers," *Pattern Analysis and Applications*, Vol. 7, No. 3, 2004, pp. 333–342.
10. Y. Jing, V. Pavlović and J.M. Rehg, "Boosted Bayesian Network Classifiers," *Machine Learning*, Vol. 73, No. 2, 2008, pp. 155–184.
11. J. Masci, U. Meier, D. Ciresan, J. Schmidhuber and G. Fricout, "Steel Defect Classification With Max-Pooling Convolutional Neural Networks," *The 2012 International Joint Conference on Neural Networks*, IEEE, 2012, pp. 1–6.
12. S. Arikan, K. Varanasi and D. Stricker, "Surface Defect Classification in Real Time Using Convolutional Neural Networks," arXiv preprint arXiv:1904.04671, 2019.
13. N. Friedman, D. Geiger and M. Goldszmidt, "Bayesian Network Classifiers," *Machine Learning*, Vol. 29, No. 2-3, 1997, pp. 131–163.
14. K. Kurach, M. Lucic, X. Zhai, M. Michalski and S. Gelly, "The GAN Landscape: Losses, Architectures, Regularization, and Normalization," 2018.
15. S. Barratt and R. Sharma, "A Note on the Inception Score," arXiv preprint arXiv:1801.01973, 2018.
16. A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta and A.A. Bharath, "Generative Adversarial Networks: An Overview," 2017.
17. M. Lucic, K. Kurach, M. Michalski, S. Gelly and O. Bousquet, "Are GANs Created Equal? A Large-Scale Study," 2017.
18. A. Krizhevsky, I. Sutskever and G.E. Hinton, "Imagenet Classification With Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.
19. P. Isola, J.-Y. Zhu, T. Zhou and A.A. Efros, "Image-to-Image Translation With Conditional Adversarial Networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.
20. J.-Y. Zhu, T. Park, P. Isola and A.A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.
21. A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang and R. Webb, "Learning From Simulated and Unsupervised Images Through Adversarial Training," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2107–2116.
22. M. Selvi and D. Jenefa, "Automated Defect Detection of Steel Surface Using Neural Network Classifier With Co-Occurrence Features," *International Journal*, Vol. 4, No. 3, 2014.
23. J. Redmon and A. Angelova, "Real-Time Grasp Detection Using Convolutional Neural Networks," *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 1316–1322.
24. J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.
25. T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen and X. Chen, "Improved Techniques for Training GANs," *Advances in Neural Information Processing Systems 29* (D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon and R. Garnett, eds.), Curran Associates Inc., 2016, pp. 2234–2242. ◆

## Disclaimer

This was the sole labor and work of ANT Corp.



This paper was published in the AISTech 2020 Conference Proceedings. AIST members can access the AISTech 2020 Conference Proceedings in the AIST Digital Library at [digital.library.aist.org](http://digital.library.aist.org).