Performance Optimization of Automated Surface Inspection Systems




In the production of flat-rolled steel sheet, surface quality is a key factor in distinguishing steel products between automotive and other applications. Due to the growth of market competitiveness, which requires the product to be checked on-line in terms of defect detection and classification, steel producers have increasingly turned to the use of commercially available automated surface inspection systems (ASIS) to aid in the detection and classification of surface defects. High defect detection and classification performance of an ASIS is an important prerequisite for efficient use of ASIS for improved surface quality control and assurance. This is especially useful for the hot rolling process since it is the earlier rolling process, and ASIS results with good classification performance could be used to avoid defect crisis and reduce costs in the following processes. This paper focuses on classification performance optimization using ArcelorMittal Cleveland's hot strip mill ASIS as an example.

Authors

Zhelin Chen (top row, left)
Ph.D. candidate, University of Illinois at Urbana-Champaign, Urbana, Ill., USA
zchen61@illinois.edu

Jue Li
formerly of ArcelorMittal Cleveland 84-Inch Hot Strip Mill, Cleveland, Ohio, USA

Richard J. Pavlosky (top row, right)
senior quality engineer, ArcelorMittal Cleveland 84-Inch Hot Strip Mill, Cleveland, Ohio, USA

Lane Galey (bottom row, left)
systems engineer, ArcelorMittal Cleveland 84-Inch Hot Strip Mill, Cleveland, Ohio, USA
lane.galey@arcelormittal.com

Liwei Zhang (bottom row, right)
lead engineer, processing and control, ArcelorMittal Global R&D – East Chicago, East Chicago, Ind., USA
liwei.zhang@arcelormittal.com

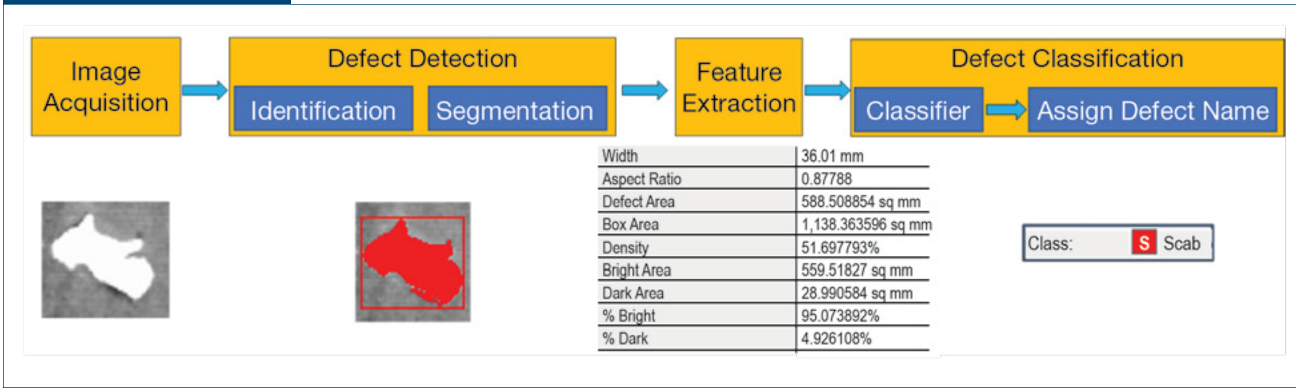
Automated surface inspection technology has been researched and developed during the last few decades. In its earliest phases, surface inspection systems[1] employed the measurement of reflected light from a point detector or a set of photodetector arrays. In recent years, automated surface inspection technology has been significantly improved and commercialized automated surface inspection systems (ASIS) have been applied to many different steelmaking processes, including hot rolling,[2] pickling,[3] tinning and hot-dip galvanizing,[4] for surface quality control and assurance.

There are three main ASIS vendors in the market: Cognex/Ametek, Siemens VAI/Primetals Technologies and ISRA-Parsytec.[5] All three types of ASIS are camera-based inspection systems that include four functions: image acquisition, defect detection/segmentation, feature extraction and defect classification.[6] These ASIS have both defect detection and classification abilities, and the classification results could be used in near real time to assist in surface quality control and assurance. Each ASIS vendor applies different technologies to detect and classify defects. In order to filter image noise and detect defects, each ASIS may have applied different detection algorithms. For defect classification, they all have used supervised pattern recognition techniques, but different algorithms are used in different ASIS as main classification tools.[5]

ArcelorMittal Cleveland's hot strip mill (HSM) has installed and developed the use of a Cognex/Ametek ASIS, which has been optimized to meet the growing demands of providing hot bands to meet critical surface quality requirements since its initial installation in May 2011.[7] The Cognex/Ametek system uses a decision tree algorithm as the main classification tool.[8] This paper starts with the introduction of ASIS functions and its classification tuning for the Cognex/Ametek system, then presents and analyzes the C5.0 decision tree theory. Finally, it describes best practice studies on C5.0 decision tree (DT) classification optimization, using the Cleveland HSM Cognex/Ametek ASIS image data as examples. The best practices are very useful to optimize an ASIS classification performance based on

This article is available online at AIST.org for 30 days following publication.

Figure 1



Block diagram of automated surface inspection systems (ASIS) process.

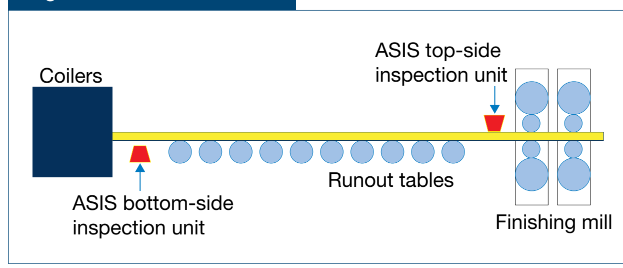
the DT model. The general methodology can also be applied to classification performance optimization of any other ASIS.

ASIS Functions and Its Classification Tuning

ASIS Functions — ASIS is a camera-based vision system enabling the on-line detection, localization and classification of surface defects while the strip is running. As shown in Fig. 1, ASIS includes four processes: image acquisition, defect detection/segmentation, feature extraction and defect classification. Image acquisition, one of the key processes, is the process of translating optical signals to a stored digital image by the camera sensor and electronics. If the image has not been satisfactorily acquired, the intended tasks may not be achievable, even with the aid of some forms of image enhancement. The defect detection process includes two steps: defect identification and processing, and defect segmentation. The defect identification process is to find and process defects from normal strip while the defect segmentation process is to segment the defects as single blobs and merge related defects as a complete defect. Feature extraction is the process of deriving mathematical measurements from the detected defects. These measurements are very similar to defects in the same category and very different for defects in distinct categories. During the defect classification process, the classifier will assign the defect to a category based on the feature vectors provided by the feature extraction process. The whole surface defect inspection process is a closely forward-linked chain. The performance of defect classification will depend on the performance of all the previous processes in the chain. The performance of any process will affect the final performance of an ASIS.

Cognex/Ametek ASIS at the Cleveland HSM — Fig. 2 shows the Cognex/Ametek ASIS setup at the Cleveland

Figure 2



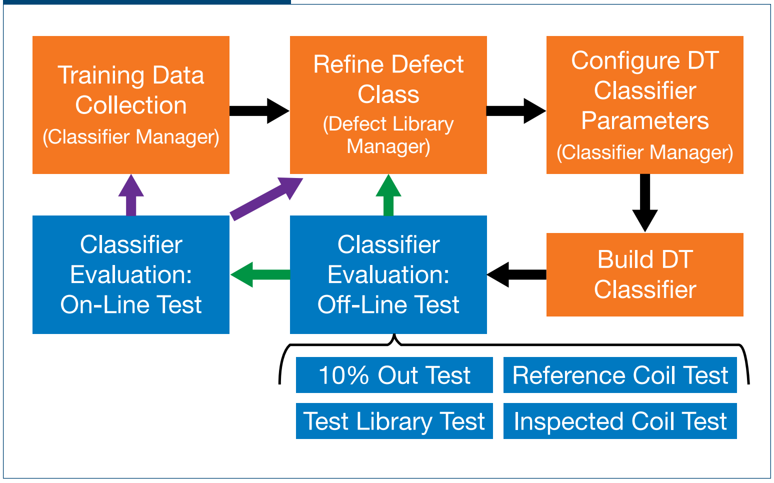
Cognex/Ametek ASIS installation location at the Cleveland hot strip mill (HSM).

HSM.[7] The top-side surface inspection unit is installed at the end of the 7-stand finishing mill and the bottom-side surface inspection unit is installed before the downcoilers.

The Cognex/Ametek ASIS is used to detect and classify surface defects that adversely affect the surface quality of hot-rolled strips. Then, the outputs of ASIS classification for the defects are fed into proper ASIS data application for surface quality control and assurance in near real time. These defects include, but are not limited to, roll marks, drag scales, scales, furnace tears, laminations, scabs and slivers. Therefore, the classifier plays an essential role to achieve efficient surface quality control and assurance in near real time. Cognex/Ametek ASIS deploys a C5.0-type DT classifier as the main classification tool. The C5.0-type DT classifier has been widely used in industry because of advantages such as quickly classifying unknown records, being able to handle both continuous and discrete attributes, and its robust effect of outliers.[9]

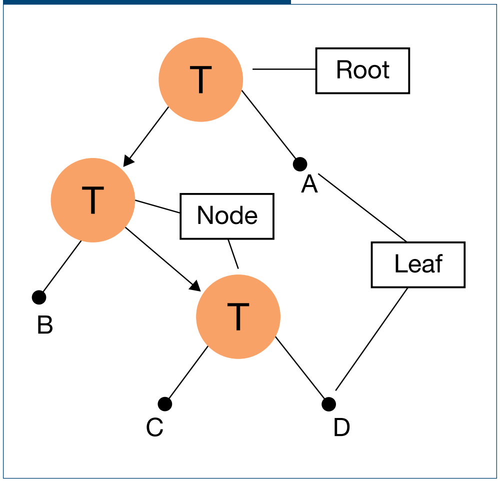
Classification Tuning for Cognex/Ametek ASIS — The classification performance for a certain defect is evaluated by two parameters: accuracy rate and confidence rate.

Figure 3



Flowchart of general classifier tuning procedure.

Figure 4



Decision tree classifier example.

The accuracy rate defines the percentage of the defect correctly classified by the classifier. The confidence rate defines the percentage of true sample numbers in the classified samples for a certain defect. A good classifier needs to have as high accuracy rate and as high confidence rate as possible. Therefore, a classifier needs to classify all the real gross and important defects and reduce false positive ones. Confusions between gross and real defects, including important and less important ones, are normally allowed.

Fig. 3 shows the flowchart of a general classifier tuning procedure. The first step is to prepare a training set, which is the defect library that contains different defect samples. Classifier Manager (Cognex/Ametek software) is used for collecting defect samples from an online database and build a defect library. Defect Library Manager (Cognex/Ametek software) is then used to refine the defect library. After choosing classifier parameters, Classifier Manager is used to build a DT classifier. Finally, the classifier is tested both off-line and on-line. Based on the test results, the whole procedure will be repeated several times until a satisfactory classifier is created.

Decision Tree Algorithm

Decision Tree Classifier — A decision tree, which is based on the "divide and conquer" strategy,[10] is one of the inductive learning algorithms that generates a classification tree using the training data/samples. An example of a DT classifier is shown in Fig. 4. Leaves of a tree are class names, and each node represents a feature-based test with branches as possible outcomes. To classify a sample, one starts at the root of the tree, evaluates the test and takes the branch appropriate to the outcome. The process continues until a leaf is encountered, at which time the sample is asserted whether it belongs to the class named by the leaf. The tree is expanded until every sample is classified into one of the leaves.

Decision tree classifiers are sometimes more interpretable than other classifiers such as neural networks and support vector machines because they combine simple questions about the data in an understandable way.[10] The decision tree approach has substantial advantages for surface quality classification problems because of its flexibility and ability to handle nonlinear relations between features and classes at a faster speed. Hence it can easily achieve near-real-time classification and improve the classification accuracy to a great extent.

The generalized method for constructing a decision tree for an arbitrary collection S of samples can be summarized as follows:

- If S is empty or only contains samples of one class, then the simplest DT is a leaf labeled with that class.
- Otherwise, let T be any test on a sample that will produce possible outcomes of $\{O_1,O_2,...,O_k\}$. Each sample in S will give one of these outcomes for T. Therefore, T portioned S into subsets $\{S_1,S_2,...,S_k\}$ S_i containing all the samples with the outcome O_i. The same method is applied recursively to each subset S_i of S to build a tree.

C4.5/C5.0 Classification Algorithm — C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan,[11] and C5.0 is the latest commercialized version of C4.5. Being multi-threaded, C5.0 is faster, more memory sufficient than C4.5, but the selecting criteria used to build the tree is still the same.

C4.5 builds decision trees using the concept of information entropy. Suppose there are n classes, denoted as $\{C_1, C_2, ..., C_n\}$. The entropy at a given node t is:

$$entropy(S_t) = -\sum_{j=1}^{n} \frac{freq(C_j, S_t)}{|S_t|} \times \log_2\left(\frac{freq(C_j, S_t)}{|S_t|}\right)$$

(Eq. 1)

where

S_t = the subset of samples in node t,
$|S_t|$ = the number of samples in S_t and
$freq(C_j, S_t)$ = the frequency of class C_j at node t.

$entropy(S_t)$ measures the amount of information needed to identify the class of a sample in S_t. Now considering a parent node p with a subset S_p, after applied test, T it is partitioned into k subsets $\{S_1, S_2, ..., S_k\}$, the information gain of test T is defined as:

$$info\_gain(T) = entropy(S_p) - \sum_{i=1}^{k} \frac{|S_i|}{|S_p|} entropy(S_i)$$

(Eq. 2)

info_gain(T) measures the information gained by portioning S_p in accordance with test T. By analogy with the definition of *info_gain(T)*, we have:

$$split\_info(T) = -\sum_{i=1}^{k} \frac{|S_i|}{|S_p|} \times \log_2\left(\frac{|S_i|}{|S_p|}\right)$$

(Eq. 3)

This represents the potential information generated by dividing S_p into k subsets, whereas the information gain measures the information relevant to the classification that arises from the same division. Then, gain ratio

$$gain\_ratio(T) = \frac{info\_gain(T)}{split\_info(T)}$$

(Eq. 4)

expresses the portion of information generated by the split that is used for classification.

At each node, C4.5 chooses one feature that most effectively splits its set of samples into subsets. There are two possible splitting criteria:

- Gain criterion selects a test T to maximize the $info\_gain(T)$.
- Gain ratio criterion selects a test T to maximize the $gain\_ratio(T)$.

Although giving quite satisfactory results, the gain criterion has a serious deficiency, which is to have a strong bias in favor of tests with many outcomes.[11] However, since Cognex/Ametek ASIS uses binary tests, this deficiency is not a problem. In general, the gain ratio criterion is more robust and typically gives a consistently better result than gain criterion.[12]

The recursive partitioning method of constructing decision trees will continue to subdivide the set of training samples until each subset contains samples of a single class, or until no test offers any improvement. The result is often a very complex tree that overfits outliers, mislabeled, noisy data resulting in the inference of more structures than being justified by the training samples/set.[11] In recent decades, many methods for simplifying trees have been proposed. Among them, pruning techniques are probably the most popular. C4.5 uses error-based pruning[11] to discard unnecessary subtrees and replaces them with leaves.

Model Overfitting — The errors committed by a classification model are usually divided into two categories: training error and generalization error. Training error is the number of misclassification errors committed on the training set, whereas the generalization error is the expected error of the classification model on unseen data, i.e., test set. A good classification model must have both low training error and low generalization error. Besides the training set, there is another independent data set called the test set, which will be used to test the performance of the resulting DT classifier from the training set. The error of the training set (percentage of the correctly classified samples in the training set) is called the training error, while the error of the test set is called the test error. The test error is a generalization error since the model has not seen the test data.

When a tree is too simple, both the training error and the test error are large, which is called underfitting. Underfitting occurs when the model has not yet learned the true structure of the training set. As a result, the classification error is large on both the training set and previously unseen data. When the number of nodes increases, the tree grows larger and both errors decrease. However, once the tree becomes too large, its test error starts to increase while its training error continuously decreases. This phenomenon is called model overfitting. The reason is that when the tree is too complex, some nodes that accidentally fit the outliers or noisy data may not generalize well to test set and downgrade the performance of a tree.

For the Cognex/Ametek system, besides the DT algorithm itself with the pruning and boosting capabilities to minimize the underfitting and overfitting problems, underfitting can be solved by increasing the training set size, to increase the tree complexity. Overfitting may be a more important problem. Overfitting can be caused by variety of reasons like presence of noise, lack of representative samples and most importantly the complexity of the model.[13]

Factors Affecting Classifier Performance Based on DT Theory

When building a DT classifier, the effects of its algorithm parameters on the classification performance need to be optimized, but it is not the focus of this paper. This paper focuses on mainly the effects of classification performance using a different training set (with the same algorithm parameters). The training set has significant effects on the classifier's performance because small variations in the training data can result in very different-looking trees.[14]

One of the requirements for the C4.5 DT model is sufficient data. The amount of data required is affected by factors such as the numbers of features and classes and the complexity of the classification model; as these increase, more data will be needed to construct a reliable model.[11] It has been proven that the model accuracy tends to increase at a decreasing rate with increases in the sample size.[15] However, if the sample size is too large, this might also hurt the model's performance. Oates et al. found that training set size has a direct impact on tree size. Even if C4.5 is pruning unnecessary subtrees, overfitting is in fact occurring, and it gets worse as the sample size increases. Research shows that there is a nearly linear relationship between training set size and tree size, even after accuracy has ceased to increase.[16]

Besides the sample size, the number of classes and features will also have effects on the performance. If there are too many classes, then the number of training samples becomes smaller quickly in a tree with many levels/branches. This problem is called error-prone.[17] For a given sample size used in training a classifier, there exists an optimal feature size. This result is true for both two-class problems and multiclass problems. In tasks where more features than the "optimal" are available, decision tree quality is affected by redundant and irrelevant features.[18]

For Cognex/Ametek ASIS, there are two types of defect library: training set and test set. The training set defect library will affect the classification performance in many ways. The following discusses in detail the specific factors in the training set, which might affect a final DT classifier performance:

1. Subclass separate training vs. combined class training — Suppose one has two defect classes: A and B, which are separated into different subclasses (A1, A2, A3, B1, B2) based on size, darkness or severity. Separate training means that a classifier is trained for five different classes (A1, A2, A3, B1, B2), but only two classes (A and B) are considered while evaluating the performance. Combined training means that a classifier is trained for two classes (A and B). For the same data set, separate training may have several drawbacks: first, it will have bigger entropy due to more classes, and more information is needed to correctly classify the data set; second, a sample in one subclass may also fill into another subclass (a dark sample may also be a small sample).
2. Sequence of classes when training classifier — Based on the DT algorithm, there should be no difference with different class sequence put in the classifier training. The practical study will test if there is improved classification performance with the Cognex/Ametek DT classifier regarding the class sequence.
3. Total class categories — As mentioned in the previous section, more data will be needed to construct a reliable model with more class categories,[11] which means more time and efforts need to be devoted to sampling collection. For a specific training set, if there are too many classes, the number of training samples becomes smaller quickly in a tree with many levels/branches,[17] which may lead to overfitting.
4. Data balancing for each class/category — A data set is considered "imbalanced" if one class (the majority class) vastly outnumbers the others (minority classes) in the training data. One of the weaknesses of decision trees is to deal with imbalanced data sets.[19] This is because when learning under highly imbalanced training data, classifying all instances as the majority class will result in high classification accuracy. Therefore, a training data set consisting of an importunately high number of examples from one class will result in a classifier that is biased toward this majority class.[20]
5. Feature number vs. sample number — Recall that model accuracy tends to increase at a decreasing rate with increases in training set size.[15] However, if the training set size is too large, this might also hurt the model's performance. What's more, for a given sample size used in training a classifier, there exists an optimal feature size. If more features than the "optimal" are available, the decision tree quality is known to be affected by the redundant and irrelevant features.[18] Then for a given feature size, there should also be an optimum sample size.

Practical Study

This section presents the practical studies regarding the above five specific aspects affecting a DT classifier's performance.

Test Methodology — Software tools in Cognex/Ametek ASIS are used to build and refine a training set. The final training set includes 65 subclasses, 21 classes and a total of 6,648 samples. Two test sets were prepared. The first one consists of 10 verified coil inspection data under the normal and good production conditions, which is shown in Table 1, and will be referred to as 10-coil test set. The second test set consists of the previously collected samples (not used in the training), as shown in Table 2. These samples are a good test set to be used for studies, where large numbers of gross defects are needed. The classifier model is created in the Cognex/Ametek ASIS environment. Several classifiers were built using different training sets under the same DT algorithm parameters with all 132 available features in Cognex/Ametek.

The C5.0 DT algorithm builds a classifier to maximize overall accuracy rate for all defect classes in the training set, and normally the performance of a classifier is evaluated using the accuracy rate of all classes. In this paper, the classifier performances on the gross/important defects (SkidTear, SkidTearD, Scab, and Lamn) are used for evaluation and comparison.

Table 1

Reference Coil Defect Samples

Total samples \ Defect types	SkidTear	SkidTearD	Scab	Lamn	Other
25,049	28	10	23	2	25,031

Table 2

Test Library Defect Samples

Total samples \ Defect types	SkidTear	SkidTearD	Scab	Lamn	Other
2,853	630	300	755	225	913

Results and Discussion

Separate Training and Combined Training: Two classifiers are studied here: the first classifier was built using the separate training option with 65 different classes since all subclasses were treated as independent classes; the second classifier was built under the combined training option with 21 classes. The classification results of these two classifiers are shown in Table 3.

The test results show that the classifier with the combined training has better performance (better accuracy and confidence rates, and lower unclassified rate) than the separate training. This is consistent with the theoretical study.

Defect Class Sequences: In a Cognex/Ametek classifier building environment, users can choose the sequence of defect classes when training a classifier. Typically, the defect classes should be in the order of difficulty to classify. To study the effect of the sequence while training a classifier, two classifiers with reversed sequence were trained: the first classifier uses the sequence from pseudo to gross defects while the second classifier uses the sequence from gross to pseudo defects. The classifier test results are shown in Table 4.

The test results show that the unclassified rate is about the same for two different classifiers. However, the classifier with gross defects first in the sequence has better defect classification accuracy and confidence rates. Therefore, while training a classifier in Cognex/Ametek ASIS environment, it's better to put gross defects first, then pseudo defects.

Table 3

Performance of Classifiers

Training methods \ Classifier performance	Accuracy rate (%)	Confidence rate (%)	Unclassified rate (%)
Combined training	93.65	78.67	6.16
Separate training	74.60	75.81	21.61

Note: 10-coil test set is used for classifier performance test (10 verified coil inspection data under the normal and good production conditions). The gross defect classification performance was evaluated.

Table 4

Performance of Classifiers

Sequence \ Classifier performance	Accuracy rate (%)	Confidence rate (%)	Unclassified rate (%)
Gross first	93.65	78.67	6.16
Pseudo first	82.54	67.53	6.41

Note: Pseudo defects mean unimportant defects, which do not affect surface quality. 10-coil test set is used for classifier performance test. The gross defect classification performance was evaluated.

Table 5

Performance of Classifiers

Classifier \ Classifier performance	Accuracy rate (%)	Confidence rate (%)	Unclassified rate (%)
A	93.65	78.67	6.16
B	85.71	60.67	49.80
C	71.43	68.18	6.58

Note: 10-coil test set is used for classifier performance test. The gross defect classification performance was evaluated.

Total Class Categories: Based on the discussion from the previous section, the number of total classes in the training set has effects on a classifier's performance. In this section, three classifiers with different defect class categories are studied using the same training set (65 subclasses, and a total of 6,648 samples) as for the previous study:

- Classifier A: This classifier combines 65 subclasses into 21 classes.
- Classifier B: This classifier combines 65 subclasses into 30 classes. Compared to Classifier A, the gross defect classes are the same, but more pseudo defect classes are added based on their shapes, sizes or brightness differences. For example, Friction class in Classifier A is separated into FrictionBlock, FrictionDiagonal, FrictionVshape and so on.
- Classifier C: This classifier combines 65 subclasses into 27 classes. Compared to Classifier A, the pseudo defect classes are the same, but more gross defect classes are added based on their shapes, sizes or brightness differences. For example, Tear class in Classifier A is separated into TearLarge, TearMedium and TearSmall.

These three classifiers use the same defect samples and classifier parameters; the only difference is the number of defect classes to be classified. As a result, comparing the performance of these three classifiers will reveal the effects of total class categories for the specific training set on classifier performance. The 10-coil test set is used to test the performance and the test results are shown in Table 5.

The test results show that Classifier A has the best performance overall. Comparing the performance of Classifiers A and B, one can see that when more pseudo defect classes are added to the training set, the accuracy and confidence rates of the gross/important defects are dropped. The unclassified rate is increased and most of the newly unclassified samples are pseudo defects. Compared with Classifier A, the results of Classifier C show that when adding more gross defect classes, the accuracy and confidence rates of gross/important defect classification are drop significantly although the unclassified rate was not seriously affected (increase about 0.4%). Therefore, one can conclude that adding more classes based on their shapes, sizes and brightness differences will downgrade the performance of a classifier.

Data Balancing Between Classes: To study the effects of the data balancing between defect classes on the classifier's performance, six classes are used (SkidTear, SkidTearD, Scab, Lamn, Stain and ScalePits). The second test set (samples not used in the classifier training) is used for performance test instead of on-line coil inspection data since the on-line coil inspection data does not contain enough gross defect samples for this study. SkidTear, SkidTearD, Scab and Lamn are categorized as gross defects while the rest are categorized as pseudo defects. Note that ScalePits are important defects. However, the Cognex/Ametek system is not able to distinguish them from stains or pseudo dots (unimportant defects) well. Therefore, the ScalePits defects are put into the pseudo category in this study.

Two different studies are performed:

- Low gross defect and high pseudo defect cases: Each gross defect class has 50 samples. The samples in each pseudo defect class vary according to different ratios.
- High gross defect and low pseudo defect cases: Each gross defect class has 20 samples, while the samples in each gross defect class vary. The reason 20 samples per class is chosen is that there are not enough gross defect samples to allow a larger number under 10-to-1 ratio for gross defects over pseudo defects.

Random defect samples of the six classes were selected from the second test set to form a new subtest set, with total 1,940 gross samples and 913 pseudo samples. The test results are shown in Tables 6 and 7.

Results in Table 6 show that, when there are more pseudo defects, the classifier's performance increases as the pseudo defect sample number decreases (i.e., the data set is transferring from unbalanced to balanced). When learning under pseudo defect dominated unbalanced training data, C5.0 DT algorithm tends to bias toward the pseudo classes (majority classes). Therefore, the accuracy rate of gross defect is increased when the pseudo samples are decreased, while the confidence rate acts the opposite.

The case is similar when there are much more gross defect samples than pseudo defect samples. C5.0 DT algorithm tends to classify pseudo defect samples as gross defects, which may lead to high accuracy but low confidence rates for gross defects for the resulting classifier. Results in Table 7 show that when the training set is highly unbalanced and dominated by gross defects, the resulting classifier can achieve higher accuracy rate, but lower confidence rate on gross defects.

Both the theoretical study and experimental results show that a balanced training set, such as 1:1 ratio for gross and pseudo defects, is better for DT algorithm.

Feature Size and Training Set Size: It is shown in the literature that for a given sample size, there exists an optimal feature size and for a given feature size, there is an optimal sample size.[15,18] This section explores the effect of sample number per class on the classifier performance. Classifiers studied in this section uses 132 total features. Six classes (SkidTear, SkidTearD, Scab, Lamn, Stain and Scale) are selected, and the second test set is used for classifier performance test. The sample number for all classes is kept the same intentionally to form a balanced training set. Different training data set sizes (30, 60, 120, 180, 250 samples per class) are used and the test results are shown in Table 8.

Results in Table 8 show that both accuracy and confidence rates are lower when the training set is smaller than the feature size. The accuracy and confidence rates are increased with a decreasing rate while training data set size is increased, which matches the DT theory analysis results. The literature review also shows that, if the training set size is bigger than the "optimum" value, the classifier's performance will be decreased due to an overfitting problem. Although more studies need to be performed to find the optimal sample size, the results in Table 8 show that the 250 samples per class seem to be a good choice for this particular study.

Table 6

Performance of Classifiers

Gross : Pseudo \ Classifier performance	Accuracy rate (%)	Confidence rate (%)	Unclassified rate (%)
1:12	65.88	99.22	3.22
1:10	66.55	99.54	3.89
1:8	70.26	99.20	2.70
1:6	73.66	99.17	3.05
1:5	76.75	98.35	2.49
1:4	80.10	97.67	2.49
1:3	85.15	96.83	1.86

Note: The new sub-test set (1,940 gross samples and 913 pseudo samples) is used for classifier performance test. The gross defect classification performance was evaluated.

Table 7

Performance of Classifiers

Gross : Pseudo \ Classifier performance	Accuracy rate (%)	Confidence rate (%)	Unclassified rate (%)
10:1	99.74	72.20	0.46
8:1	99.33	78.69	0.46
6:1	99.43	79.42	0.46
5:1	99.48	79.69	0.60
4:1	99.28	81.47	0.53
1:1	96.19	91.70	0.67

Note: The new sub-test set (1,940 gross samples and 913 pseudo samples) is used for classifier performance test. The gross defect classification performance was evaluated.

Table 8

Performance of Classifiers

Sample per class \ Classifier performance	Accuracy rate (%)	Confidence rate (%)	Unclassified rate (%)
250	95.74	90.80	0.81
180	95.62	89.88	0.73
120	94.73	91.02	0.96
60	92.25	87.83	0.92
30	90.30	75.77	1.38

Note: The new sub-test set (1,940 gross samples and 913 pseudo samples) is used for classifier performance test. The gross defect classification performance was evaluated.

Conclusion

This paper studied the best practices of robust tuning of a decision tree classifier using Cognex/Ametek ASIS classifier training as an example. The studies show that when building the classifier, combined training and putting gross defects before a pseudo defect will improve the classifier model's performance. When building the training set, unnecessary classes (like separate classes based on size, brightness) and an unbalanced data set need to be avoided. The studies also show that an optimal feature size/training size relation exists, and the optimal sample size appeared to be 250 samples per class with the used training set.

The classification performance is the key to maximizing ASIS use benefit. Although the study and test results are from Cognex/Ametek ASIS, the best practices can be applied to any ASIS where a DT classifier is deployed.

Acknowledgment

This work was supported by ArcelorMittal Global R&D and ArcelorMittal Cleveland. Special thanks are given to Cleveland Hot Strip Mill management and global ArcelorMittal Global R&D – East Chicago management for their great support and encouragement to this work. Special thanks are also given to Joseph Bentsman from University of Illinois at Urbana-Champaign and B.G. Thomas from Colorado School of Mines for ideas and suggestions.

References

1. E.J. Klimas, F. Popovich and R.J. Pavlosky, "The Practical Application of Automatic Inspection Instrumentation to Cold-Rolled Steel Strip," SAE Technical Paper, 1984.
2. M. Baileul, "Dynamic Surface Inspection at the Hot Strip Mill," *AISTech 2005 Conference Proceedings,* Vol. II, 2005, pp. 289–293.
3. T. MacDougall and E. Dillon, "Implementation of Surface Inspection Technology on Noisy Surfaces," *AISTech 2007 Conference Proceedings,* Vol. II, 2007, pp. 629–637.
4. M. Gonzalez, A. Treviño, M. Bailleul and F. Nlome-Nze, "Automatic Surface Inspection for Galvanized Products at IMSA MEX," *AISTech 2006 Conference Proceedings,* Vol. II, 2006, pp. 199–201.
5. L. Zhang, H. Long and J. Chapko, "Tuning and Deployment of a Surface Inspection System," *Iron & Steel Technology,* Vol. 7, No. 1, 2010, pp. 38–46.
6. L. Zhang, J. Franklin, T. Pisanelli, R.J. Pavlosky and H. Long, "Optimization of Surface Inspection System Performance," *AISTech 2008 Conference Proceedings,* Vol. II, 2008, pp. 869–879.
7. R.J. Pavlosky, L. Galey, M.B. Assar and L. Zhang, "Practical Applications of Surface Inspection System at the Hot Mill," *AISTech 2015 Conference Proceedings,* Vol. II, 2015, pp. 2734–2744.
8. C. Corp., "SmartView ICN System Setup and Reference."
9. S. Miertschin, "Decision Tree," University of Houston.
10. R. Safavian and D. Landgrebe, "A Survey of Decision Tree Classifier Methodology," *IEEE Transactions,* Vol. 21, No. 3, 1991, pp. 660–674.
11. J.R. Quinlan, C4.5: Programs for Machine Learning, Elsevier, 2014.
12. J.R. Quinlan, "Decision Trees and Multi-Valued Attributes," *Machine Intelligence 11,* New York, N.Y., USA, 1988, pp. 305–318.
13. P-N. Tan, M. Steinbach and V. Kumar, *Introduction to Data Mining,* Addison-Wesley Longman Inc., 2006, pp. 172–186.
14. J.R. Quinlan, "Induction of Decision Trees," *Machine Learning,* Vol. 1, No. 1, 1986, pp. 81–106.
15. J. Morgan, R. Daugherty, A. Hilchie and B. Carey, "Sample Size and Modeling Accuracy of Decision Tree Based Data Mining Tools," *Academy of Information and Management Science Journal,* Vol. 6, No. 2, 2003, pp. 71–99.
16. T. Oates and D. Jensen, "The Effects of Training Set Size on Decision Tree Complexity," *Fourteenth International Conference on Machine Learning,* San Francisco, Calif., USA, 1997.
17. A. Nordman, "Classification: Decision Trees [Lecture 3]," TNM033: Data Mining.
18. H. Almuallim and T.G. Dietterich, "Learning Boolean Concepts in the Presence of Many Irrelevant Features," *Artificial Intelligence,* Vol. 69, No. 1, 1994, pp. 279–305.
19. H. He and E.A. Garcia, "Learning From Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering,* Vol. 21, No. 9, 2009, pp. 1263–1284.
20. N. Japkowicz, "The Class Imbalance Problem: Significance and Strategies," *Proceedings of the International Conference on Artificial Intelligence,* 2000. ✦

This paper was presented at AISTech 2018 — The Iron & Steel Technology Conference and Exposition, Philadelphia, Pa., USA, and published in the Conference Proceedings.

Did You Know?

The 9th Steelie Award winners announced

The winners of the 9th Steelie Awards were announced at the annual dinner of the World Steel Association (worldsteel) on 16 October 2018. The Steelie Awards recognize member companies or individuals for their contribution to the steel industry over a one-year period. The Steelies are awarded in seven categories, two of which are new this year.

Excellence in Digital Communications expands upon the highly coveted Website of the Year award to now recognize all digital communications efforts from websites to mobile applications to social media.

Excellence in Communications Programs recognizes investment, creativity and innovation in major communications programs (internal and external) that promote the steel industry or steel as a competitive material. The award covers all channels from advertising, print, marketing promotion and digital.

The winners for 2018 are:

- Excellence in Digital Communications: **Tenaris.**
- Innovation of the Year: **Acciaierie Bertoli Safau S.p.A.,** Rotoforgia innovative project.
- Excellence in Sustainability: **ArcelorMittal Brazil,** Water Master Plan.
- Excellence in Life Cycle Assessment: **ArcelorMittal,** Use of LCA to support the Steligence® project.
- Excellence in Education and Training: **Tata Steel Europe,** Product and process fundamentals.
- Journalist of the Year: **Chris Davis,** content director, EMEA Metals, S&P Global Platts.
- Excellence in Communications Programs: **Tata Steel Ltd.,** #DoorsofIndia — A journey by Pravesh.