Pre-Trained CNN for Classification of Time Series Images of Anti-Necking Control in a Hot Strip Mill

Samuel Latham^{a,*}, Cinzia Giannetti^b

^aFaculty of Science and Engineering, Swansea University, Fabian Way, Swansea, SA1 8EN, Wales

^bFaculty of Science and Engineering, Swansea University, Fabian Way, Swansea, SA1 8EN, Wales

*Corresponding Author: 865767@swansea.ac.uk

Abstract

The steel industry is highly competitive, and companies must make the most of their current resources to maximise efficiency and, therefore, profitability. In the age of Industry 4.0, data is one of the most valuable resources available and, with appropriate processing and analysis, can improve the quality and adaptability of various applications. One such application is the monitoring and classification of Anti-Necking Control in a Hot Strip Mill. This paper proposes a deep learning approach to this application through the use of a pre-trained Convolutional Neural Network. The proposed system binarily classifies the timing of Anti-Necking Control strokes and has been optimised using grid search optimisation in conjunction with k-fold cross validation to determine an optimal time series image classification model.

Keywords: Deep learning, time series, image classification.

1. Introduction

A steel mill can produce tens of thousands of tonnes of steel strip in a single day through numerous convoluted processes such as furnacing, casting, and hot rolling. To maintain prominence in the steel industry, it is imperative for companies to stay up to date on the latest technologies in all sectors, including mechanics, chemistry, and computing to make their product as feasible as possible while maintaining its expected high quality. To achieve this, they must make use of every resource at their disposal, both physical and digital. Data is abundant in most manufacturing processes and it is often not used to its full potential when it comes to process control and optimisation. Until recently, particularly before the fourth industrial revolution (Industry 4.0)⁽¹⁾, processes were predominantly monitored and controlled

manually, posing numerous problems including human error and slow decision making⁽²⁾.

The application of Industry 4.0 in manufacturing involves the utilisation of recent technologies including big data, cloud computing, and advanced robotics to create highly adjustable and customisable production lines dubbed “smart factories”⁽³⁾. The concept of smart factories has become increasingly popular in recent years as they can mitigate, or even eliminate, the previously mentioned issues while increasing productivity and improving maintenance and product quality^(3,4). There are two approaches to installing Industry 4.0 technologies in a manufacturing setting; complete decommission and reinstallation from scratch which can be high in cost but immediately effective; and gradually integrating new technologies without any downtime which can be slower overall but is more feasible in the short term. With many newer, smart steel plants being commissioned around the world, older, less developed sites could benefit greatly from the integration of these technologies to maintain their high standards in such a competitive industry.

While a huge number of operations are involved in the steelmaking process this study focusses on a specific operation in the Hot Strip Mill (HSM) which will be integrated individually into the HSM process with no down time; The HSM is one of the final stages in the production of steel strip. The Roughing Mill (RM), a process within the HSM, is the final point of direct width control of the steel. ‘Necking’ and ‘flare’ are two common width defects in steel strip products. Measures referred to as ‘Anti-Necking Control’ (ANC) and ‘Anti-Flare Control’ (AFC) quickly adapt the position of the RM’s rollers to counteract their effects. However, the timing of these adaptive movements is crucial as a mistimed ANC or AFC ‘stroke’ can have the

opposite effect than intended, sometimes encouraging necking and flare in the strip.

By providing immediate feedback to operators of the timing of these strokes, it may be possible to mitigate the effects of necking and flare in the HSM such that the underlying cause of the mistime can be identified as soon as possible once a mistimed stroke is recognised. In this paper, a pre-trained Convolutional Neural Network (CNN) has been proposed for the classification of time series images to determine ANC stroke timing; a deep learning approach in combination with big data to provide immediate feedback for quick process adaption.

2. Background

2.1 Use of Neural Network Technologies in the Steel Industry

Neural network technologies have been used to optimise hot rolled steel processes for several decades. Early applications were heavily based on numerical data which attempted to predict and control general properties of steel strip, such as width and temperature, during the rolling process^(5,6). Although computing technology was still rapidly developing during this time, it was difficult to implement changes suggested by these applications quickly the way real-time systems are able to today^(7,8). It has since become more common to use neural networks to process various datatypes, including time series data and images, which has led to a broader scope of applications such as prediction of roll force and other mechanical properties^(9,10), gearbox fault diagnosis⁽¹¹⁾, and temperature control⁽¹²⁾. In the last decade, Convolutions Neural Networks (CNNs) and transfer learning have become increasingly popular in image classification: In the last decade, many applications of this technology in the steel industry have focused on the classification of steel surface defect images^(13–16).

2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are one of the most commonly used neural network models for image classification⁽¹⁷⁾ and are designed to recognise patterns in image data, much like the human eye⁽¹⁸⁾. Statistical models can also be used to solve image classification problems although they are limited by their ability to reuse contextual information when deriving patterns in data, which CNNs are designed to retain. This is not to say that CNNs are without their drawbacks. Due to the amount of data processing

necessary to train such a model, CNNs can be computationally expensive and difficult to implement with challenges such as class imbalances and overfitting being present. These can however be solved with more computational power and data pre-processing. Once data is transformed into the required format, CNNs use multiple hidden layers to extract features and its final layers to output a label for the data based on the identified features. This process is completed over several layers which make up the CNN's structure⁽¹⁹⁾. In image classification, each layer acts as a filter which changes the network's perspective on the image in order to learn particular features.

The first layer is an input layer where image data enters the network. Next is a convolution layer which reveals features by applying filters to the image, converting it into a readable format. Third is a rectified linear unit or activation layer (ReLU) which eliminates redundant data by changing negative values in the image to zero. The fourth layer is a pooling layer which reduces data dimensionality, making features easier to recognise in image sections. The second to last layer is a fully connected layer which outputs the probabilities of the image belonging to each class. The last layer of the network is a softmax layer which outputs the classification decided by the network. This however is the basic structure of a CNN. There are many architectures which use a mix of combinations tailored to specific classification problems. No one CNN architecture is best for every classification problem⁽²⁰⁾.

Fig. 1. Diagram of a simple CNN

In the last few years, CNNs have become widely popular in a range of industries including manufacturing, medicine, and security; this paper will focus on examples within the manufacturing industry. CNNs are now more commonly used for condition monitoring and defect identification, often trained using historical data in order to process real-time data. Many use case studies have been completed to demonstrate these capabilities. CNNs have been used to determine the condition of tools in a CNC machine using time series data in the form of plot images⁽²¹⁾. Similarly, they have been used to classify power quality disturbances based on voltage

signals⁽²²⁻²³⁾. Transfer learning and other neural network architectures can be combined with CNNs to create hybrid architectures which attempt to utilise the properties and benefits of both architectures.

2.3 Transfer Learning

In traditional machine learning, models are trained from scratch. This can take a long time when large datasets are used. It is however possible to use a pre-trained network for a chosen classification problem given that this network has been trained on relevant data. This is known as Transfer Learning and is used when a pre-trained network can boost model training times and accuracy when learning new data where the old data (which the pre-trained network was trained on) is similar to the new data. For example, if a model is to be trained on a new dataset to classify images of vegetables, it may be possible to use a network pre-trained with images of fruit as it may extract features similarly between these two datasets. The pre-trained component of a network is usually used for more fundamental feature extraction such as edge and shape detection. “The objective of Transfer Learning is to take advantage of data from the first setting to extract information that may be useful when learning or even when directly making predictions in the second setting”⁽²⁰⁾.

To achieve this, a pre-trained network is chosen and, unless already specifically trained for the new problem, is amended with new layers for further feature extraction more specific to the new data including output layers to suit the machine learning problem. This means that only the new layers must be trained using the new data, saving massively on training times and computational power since it is no longer necessary to train the network from scratch.

2.4 Necking and Anti-Necking Control (ANC)

Necking is a shape defect in steel bars and strips in which the head and/or tail end of the bar is narrower than the main body. It is important to first check the capability of the bar to determine whether a ‘dog bone’ effect can be rolled throughout the rest of the HSM process. The most probable causes of necking are related to the Anti-Necking Control (ANC); a force that reduces or eliminates necking by edging the head and tail ends of the bar in the roughing mill at a particular point on the bar.

The factors of ANC which can cause necking are the stroke being too early or too late, and the magnitude of the stroke. If the stroke is too quick, the ANC will encourage further necking by edging the sides of the bar too early. If the

stroke is too late, it will miss the end of the bar and encourage flare; the body of the strip is narrower than the head and/or tail end in this case. Early or late strokes may be caused by sensor irises being dirty or by steel being accidentally detected in the mill, meaning a false presence is detected and ANC begins too early or the bar is more difficult to see and detected too late.

3. Methodology

3.1 Dataset

Since coils affected by early ANC strokes account for a much smaller percentage of steel coils, the overall dataset is unbalanced. Over 6000 samples were collected from a six-month time period, 262 of which had early ANC strokes. 262 samples without ANC problems were handpicked to ensure variation and were combined with the 262 early samples to create a balanced training dataset. Over 2000 samples were collected from the following two months, which included 77 early samples, and were combined with the remaining samples without ANC problems to be included in the final test dataset. As there is a large class imbalance in the test dataset, selected subsets have been created from this to simulate multiple balanced test datasets.

	Training Dataset	Testing Dataset	Balanced Testing
Early	262	77	77
Okay	262	8671	77
Total	524	8748	154

Fig. 2. Full training and testing dataset split

Each sample consists of two variable-length time series collected directly from the steel mill. The two time series describe edger force in the roughing mill and roller capsule position for a single pass of a steel bar through the roughing mill. Due to the ambiguity and fluctuation of values in raw time series data, the features necessary to determine whether an ANC stroke is early are more easily recognisable when visualized in image form. This feature, or characteristic, is based on the position and proximity of the sudden change in capsule position which occurs when the ANC is activated and there is a steady, but not consistent, rise in edger force. The dataset is private and was provided by an existing steel company.

3.2 Pre-Processing

It is impossible to accurately determine the exact roughing mill entry and exit points of the steel bar in these time series. This is because the only signal that can be used for this is a binary metal detection signal which can be early or late if there is debris in the mill or if there is a faulty or dirty sensor. These issues which cause metal detection to be early or late directly affect whether the ANC stroke is early or late, thus the classification of an early or late ANC stroke can help to determine the existence of the underlying issues. Therefore, the two remaining signals alone, edger force and capsule movement, must instead be used as they currently are when manually checking ANC stroke timing.

The ANC stroke itself contains a binary signal which allows a beginning and end time to be specified. Edger force time series are variable length but have a consistent pattern in which they rise when the bar enters the mill and lower when it exits. It is important to keep in mind that this pattern cannot be used to find the exact roughing mill entry and exit points of the bar due to variance in the climb caused by other factors including debris in the mill and oddly-shaped bars. In conclusion, the most informative image that can be generated from these time series is a full edger force time series combined with a capsule position time series which has been filtered with an ANC binary signal time series.

Fig. 3. Okay (left) and Early (right) examples of ANC time series images (axis labels are only shown in this figure for illustration)

3.3 Network Architecture

GoogleNet is the chosen pre-trained network for this experiment due to its ability to maintain a low computational expense when training regardless of its width and depth⁽²⁴⁾. More specifically it uses smaller convolutional filters to reduce the total parameters within the network, allowing for a larger architecture. The main feature of GoogleNet is its ‘inception module’ which contains sequential, fixed-sized convolutional filters that enable it to generalize features

more easily and avoid overfitting. GoogleNet is trained on the ImageNet dataset which contains over 14 million images.

Fig. 4. GoogleNet inception module

The final two layers used for classification in the standard GoogleNet architecture have been replaced by a custom fully-connected layer followed by a classification output layer to enable further training using the collected ANC time series images, repurposing the network to binarily classify ANC stroke timing.

3.4 Grid Search Optimisation and K-Fold Cross Validation

To find the optimal and best performing model, a range of training options has been considered. The chosen options can largely affect the accuracy, recall, and precision of a classification model. The cartesian product of the table shown in Fig. 5 is equal to the set of test cases in this grid search optimisation, totaling 540 test cases. This method enables me to visualize and determine which model performs best and any trends shown between training options.

Max Epochs	Mini Batch Size	Initial Learn Rate	Learn Rate Drop Factor	L2 Regularization
5	2	0.01	0.2	0.0001
10	4	0.001	0.3	0.0002
20	8	0.0001	0.5	0.0003
-	16	0.00001	-	-
-	32	-	-	-

Fig. 5. The training options used for grid search optimisation in this experiment

Grid search optimisation has been carried out in conjunction with k-fold cross validation to determine how well each model generalizes when training on a subset of the original model. A k of 5 has been used such that 80% of the data was used in each run; For 5 runs, one split of 20% is not used for training.

	No. of Samples
Split 1	105
Split 2	105
Split 3	105
Split 4	105
Split 5	104
Training Dataset Total	524

Fig. 6. The training dataset splits used for k-fold cross validation in this experiment

While k-fold cross validation and grid search optimisation are used in conjunction, grid search optimisation has also been conducted on the complete training dataset to determine which model(s) perform best overall.

4. Experiments and Results

4.1 Grid Search Optimisation of Balanced Datasets

The four balanced datasets have been used to test each of the 540 cases produced during grid search optimisation. The 540 cases are formed by the complete set of combinations of training options described in the methodology. Fig. 7 and Fig. 8 show the variance in accuracy and F-score between all of these combinations which is calculated through recall and precision. Precision describes the percentage of samples of one class that are classified correctly while recall describes the percentage of samples that were correctly identified. These are used to calculate the F-score which describes the overall performance of the model with regards to precision and recall. Accuracy is considered alongside F-score as it describes the overall percentage of correctly classified samples over all classes.

From these results, it is possible to choose a set of training options whose model shows both a high accuracy and F-score as well as low variance between results. It is important to consider anomalies and large variations which can be the result of combinations of other training options. Despite this, a low variation means that a particular training option is favoured.

$$P = \frac{\text{True Positives}}{(\text{True Positives} + \text{False Positives})} \quad (1)$$

$$R = \frac{\text{True Positives}}{(\text{True Positives} + \text{False Negatives})} \quad (2)$$

$$F_1 = 2 * \left(\frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \right) \quad (3)$$

The results of grid search optimisation show that a max epochs of 20 and a mini batch size of 8 are the optimal options for the ANC stroke timing classification model. Generally, more epochs lead to a higher accuracy due to the larger number of training iterations. In this case, a higher mini batch size does not necessarily equal a higher accuracy or F-score. A mid-range mini batch size is optimal here since the training dataset is relatively small. Another option that can be clearly chosen from these results is an initial learning rate of 0.001. A lower initial learn rate may cause a model to over-generalise meaning that particular image features are learned, resulting in a high training accuracy. A lower initial learn rate however can cause a model to learn the wrong features too quickly and, in this case, update weights at a rate which leads to failure during training.

Changes in learn rate drop factor and L2 regularization appear to have a lesser impact on the performance of these classification models as those formerly mentioned. Lastly, a L2 regularization value of 0.0001 and a learn rate drop factor of 0.3 have been chosen for the ANC classification model as they show the least variance in accuracy. F-score is very similar for both of these training options meaning they do not majorly affect training compared to others.

4.2 K-Fold Cross Validation

To determine the repeatability and adaptivity of the chosen model, k-fold cross validation with a k of 5 has been used to determine its performance on each balanced dataset when training on each cross validation split.

The boxplots shown in Fig. 9 and Fig. 10 show that the model performs relatively well over each balanced dataset. The maximum range of variance in accuracy and F-score between each training split is approximately 10%. The median and both quantiles in each case is relatively high, all with an accuracy and F-score of above 90%, the highest of which reaching above 98%.

Fig. 9. Boxplot showing the accuracy distribution of each cross validation split for each balanced test run

Fig. 10. Boxplot showing the F-score distribution of each cross validation split for each balanced test run

Fig. 11. Subplots of confusion matrices showing the classification results of the chosen model when tested using the selected dataset (top) and a randomised dataset (bottom)

The model, when tested with the selected balanced dataset,

does not perform as well as the randomized balanced sets. This is due to the hand-picked variability in image features of the selected balanced dataset compared to the randomized balanced datasets. The chosen model therefore performs effectively even when presented with a varied testing dataset. It is important to note that a slight drop in accuracy and F-score is expected in cross validation testing since, in this case, only 80% of the total training data is used per model. Using the complete training dataset also guarantees maximum variability of image features.

5. Conclusion

A sufficient model has been produced for the purpose of binarily classifying ANC stroke timing using time series images. Although only a small dataset was available, grid search optimisation has been used to identify the optimal training options for the pre-trained CNN used. The chosen model achieves an accuracy and F-score of greater than 97% and does not need frequent retraining in the given scenario.

The results of this experiment support the theory that images can be used to classify image time series data. This is however heavily dependent upon the use case and, more specifically, the features used to determine classes. In this case, only a small number of features are needed to determine whether an ANC stroke is early.

In this paper, a CNN pre-trained using GoogleNet has been discussed. The classification model trained from this network has sufficiently classified ANC stroke time series images, achieving an accuracy and F-score of more than 97%.

Acknowledgment

The authors would like to acknowledge the M2A funding from the European Social Fund via the Welsh Government (c80816) and Tata Steel Europe that has made this research possible.

References

- (1) D. Klitou, J. Conrads, and M. Rasmussen, "Germany: Industrie 4.0," 2017.
- (2) H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus. Inf. Syst. Eng.*, vol. 6, no. 4, pp. 239–242, 2014.
- (3) N. Jazdi, "Cyber physical systems in the context of Industry 4.0," *Proc. 2014 IEEE Int. Conf. Autom. Qual. Testing, Robot. AQTR 2014*, pp. 1–4, 2014.

- (4) E. A. Lee, "Cyber Physical Systems: Design Challenges," *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pp. 363–369, 2008.
- (5) M. Valle, D. Baratta, and D. D. Caviglia, "Application of hierarchical neural networks to pattern recognition for quality control analysis in steel-industry plants," *Proceedings of International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing, NICROSP*, pp. 246–252, 1996.
- (6) T. Fechner, D. Neumerkel, and I. Keller, "Adaptive neural network filter for steel rolling," *IEEE International Conference on Neural Networks - Conference Proceedings*, vol. 6, pp. 3915–3920, 1994.
- (7) X. Yao, A. K. Tieu, X. D. Fang, and D. Frances, "Neural network application to head & tail width control in a hot strip mill," *IEEE International Conference on Neural Networks - Conference Proceedings*, vol. 1, pp. 433–437, 1995.
- (8) Y. S. Kim, B. J. Yum, and M. Kim, "Robust design of artificial neural network for roll force prediction in hot strip mill," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 4, pp. 2800–2804, 2001.
- (9) M. Bagheripoor and H. Bisadi, "Application of artificial neural networks for the prediction of roll force and roll torque in hot strip rolling process," *Appl. Math. Model.*, vol. 37, no. 7, pp. 4593–4607, 2013.
- (10) Z. W. Xu, X. M. Liu, and K. Zhang, "Mechanical Properties Prediction for Hot Rolled Alloy Steel Using Convolutional Neural Network," *IEEE Access*, vol. 7, pp. 47068–47078, 2019.
- (11) P. Liang, C. Deng, J. Wu, Z. Yang, J. Zhu, and Z. Zhang, "Compound Fault Diagnosis of Gearboxes via Multi-label Convolutional Neural Network and Wavelet Transform," *Comput. Ind.*, vol. 113, pp. 103132, 2019.
- (12) A. G. Levykina, A. A. Chabonenko, V. V. Shkatov, and I. P. Mazur, "The study of the thermal state of the metal in the production of the hot rolled strips in Deform 3D," *Journal of Physics: Conference Series*, vol. 1134, no. 1, pp. 12034, 2018.
- (13) S. Zhou, Y. Chen, D. Zhang, J. Xie, and Y. Zhou, "Classification of surface defects on steel sheet using convolutional neural networks," *Mater. Tehnol.*, vol. 51, no. 1, pp. 123–131, 2017.
- (14) W. Chen, Y. Gao, L. Gao, and X. Li, "A New Ensemble Approach based on Deep Convolutional Neural Networks for Steel Surface Defect classification," *Procedia CIRP*, vol. 72, pp. 1069–1072, 2018.
- (15) D. He, K. Xu, and D. Wang, "Design of multi-scale receptive field convolutional neural network for surface inspection of hot rolled steels," *Image Vis. Comput.*, vol. 89, pp. 12–20, 2019.
- (16) Y. Liu, K. Xu, and J. Xu, "Periodic surface defect detection in steel plates based on deep learning," *Appl. Sci.*, vol. 9, no. 15, 2019.
- (17) H.-C. Shin *et al.*, "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning.," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.
- (18) A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- (19) J. Wang, L.-C. Yu, K. Lai, and X. Zhang, "Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model," *Proc. 54th Annu. Meet. Assoc. Comput. Linguist.*, pp. 225–230, 2016.
- (20) I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- (21) Y. C. Liang, W. D. Li, X. Lu, and S. Wang, "Fog computing and convolutional neural network enabled prognosis for machining process optimization," *J. Manuf. Syst.*, vol. 52, pp. 32–42, 2019.
- (22) S. Khokhar, A. A. Mohd Zin, A. P. Memon, and A. S. Mokhtar, "A new optimal feature selection algorithm for classification of power quality disturbances using discrete wavelet transform and probabilistic neural network," *Meas. J. Int. Meas. Confed.*, vol. 95, pp. 246–259, 2017.
- (23) S. Wang and H. Chen, "A novel deep learning method for the classification of power quality disturbances using deep convolutional neural network," *Appl. Energy*, vol. 235, pp. 1126–1140, 2019.
- (24) C. Szegedy *et al.*, "Going deeper with convolutions," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015.

Fig. 7. Subplots of boxplots showing the accuracy distribution of each model by each grid search optimisation training option for each balanced test run (including a legend illustrating the balanced test run)

Fig. 8. Subplots of boxplots showing the F-score distribution of each model by each grid search optimisation training option for each balanced test run (including a legend illustrating the balanced test run)