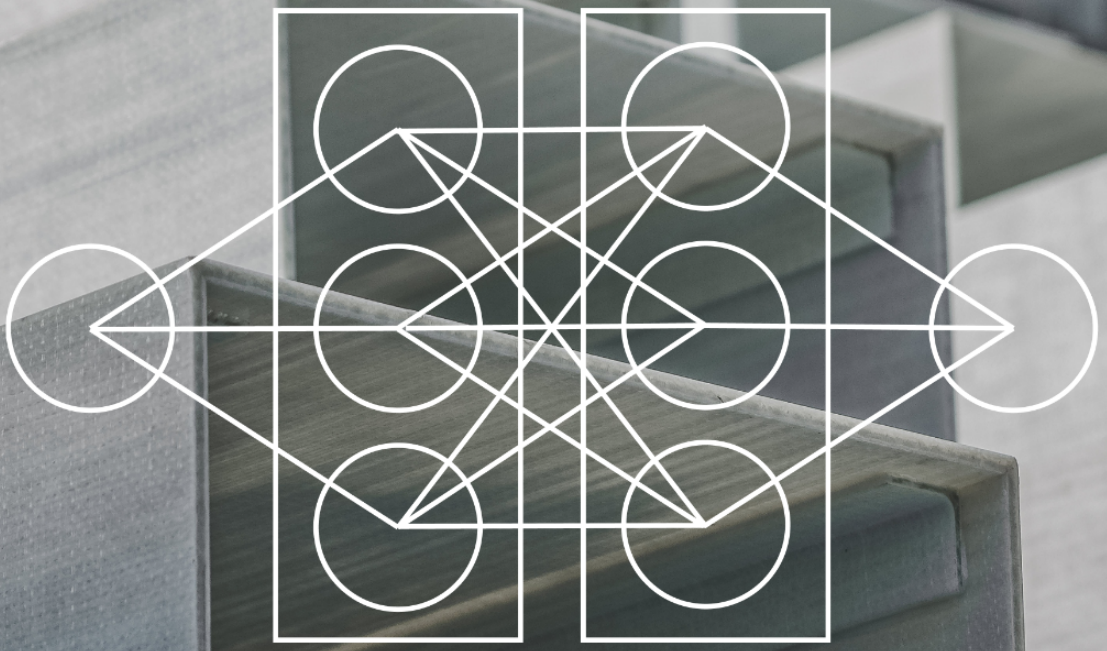
Recognizing metal surface defects using machine learning models

Classifying defects in hot-rolled metal sheets



Authors:
Axel Björlin, Alexander Hansson, Axel Magnusson, Moshteba Qurbani

KTH Royal Institute of Technology, MH1025

Stockholm, 2022-03-04

Abstract

Machine learning in material science is something that has increased rapidly the last ten years thanks to improved algorithms and different models. In this project, this method is investigated to understand if it is possible to use an already existing data set, to identify six different types of metal surface defects with a 98% hit accuracy.

The method used is examined and analyzed to understand the possibility to make it more efficient. Different methods such as the YOLO method and the Sequential method are compared. There is also an estimation of the right amount of epochs which are needed. Using TensorFlow and Keras together with the sequential model, this resulted in a defect recognition accuracy of 98% using 158 epochs during 35 minutes.

Environmental and economical aspects are something crucial in today's society. In this report, the energy consumption which the program required to run (1.68 MJ), will be discussed. The report briefly talks about the social and ethical aspects.

The role of machine learning in material science is discussed and the goal to implement the code into today's industry is estimated.

Sammanfattning

Maskininlärning i material teknik är något som har sett en exponentiell ökning de senaste tio åren på grund av förbättrade algoritmer och nya modeller. I det här projektet undersöks om det är möjligt att använda ett redan existerande dataset för att identifiera sex olika typer av defekter i stål med en 98% säkerhet.

Datasetet kommer att examineras och analyseras ifall det finns någon möjlighet för att göra det mer effektivt. YOLO metoden och Sequential metoden kommer att jämföras. Det kommer även att estimeras hur många epoker som är lämpliga att använda. Genom att använda TensorFlow och Keras tillsammans med sequential metoden uppnåddes en 98% noggrannhet av defekt identifiering efter 158 epoker och 35 minuter.

Miljön och ekonomin är något som är extremt viktigt i dagens samhälle därför kommer det att diskuteras ifall resultatet för att köra själva programmet, vilket är (1.68 MJ), är rimligt hållbarhets mässigt. Det kommer även gås igenom sociala och etiska aspekter.

Rollen som maskininlärning har inom materialteknik kommer att diskuteras och ses över ifall det är rimligt att implementera koden i dagens industri.

Table of contents

1 Introduction

1.1 Background

To make humanoids or human-like robots was the topic of many sci-fi movies and novels. There were many struggles in history to make something like them. From the sci-fi novel "The Wizard of OZ" to the murder robots in the Terminator films, even Leonadeo da Vinci, the great renaissance artist, had design notes of a robot knight in his sketchbooks.

But the modern work on AI happened in 1969 when Shakey the Robot was created at Stanford's Research Institute, which was an autonomous device that could navigate inside a room. The robot was made with a "top-down approach" which means the scientists wrote a program of the rules for how Shakey could find its way in the room and then placed it inside Shakey. This approach was quite good for a simple robot like Shakey and after that made it possible to program robots who were very good at playing chess, solve algebra problems and pick up objects, mainly cubes. But this approach was not good enough to create more complicated robots who could recognize objects like a cup, a scissor or an orange. The robots could see like us and they could do it better than us but they would have a major problem understanding what they see. On the other hand, the rules of doing a simple task like opening a door or walking on the sidewalk was more complicated than what a human could write inside a program. [1]

Because of all the problems which top-down approach had, scientists created a new method called "bottom-up" approach. This approach is simply a mimic of the evolution of children who learn tasks by doing trial and error. The program runs several times and reprograms itself to achieve a satisfactory result. Those tasks could be image recognition or walking and jumping on different roads, this approach can be improved and in a bigger scale it changes to deep learning. Future developments of machine learning made it possible for a human shaped figure to be able to walk, run and even jump only by doing random motions continuously in a simulation. Right now robots and AI can benefit both approaches in different amounts to achieve good results. [10]

1.1.1 Machine learning in material science

Artificial intelligence has been a big thing for a while but in recent years it has become more and more popular in the material science section. Especially the branch of machine learning.

Material science has historically been using a trial and error method based on a large number of experiments with a small number of computer simulations to help. In recent years it has become more popular to combine machine learning with material science. There are plenty of different uses for machine learning in material science. Some of these areas are Accelerated Simulation, Predicting the property of new materials and Synthetic Route Planning. [8]

Upgrading of Characterization Methods is something that is relevant to identifying defects in metal surfaces. The last ten years it has seen an immense development that has led scientists to be able to observe atomic-level structures that can trace atomic-level movements. [8]

1.1.2 Hot Rolled metal sheets

Hot rolled metal sheets have a sizable impact in the automotive industry, appliance manufacturing, bridges and electric motors. These industries are very important for companies and peoples daily life. The surface clearness and quality of the metal sheets are the most important aspects related to the final product, therefore the search for surface defects on the metal sheets is of utmost importance and should be controlled thoroughly and regularly. There are different factors that affect the surface quality of metal sheets and which can be improved upon by a variety of control methods. The various surface defects such as slag inclusion, red iron and surface scratches have different effects on the quality of the products therefore it is important to classify the different types of defects that could appear on the metal sheets to be able to reduce and prevent them impacting the metal sheets. [9]

The metal sheet production lines are equipped with surface defect detections systems but these systems are underperforming when it comes to classifying different types of defects. This shows itself when you compare the theoretical value of a system's accuracy with the actual process of the operations. This led to algorithms not being able to replace the manual labor for classifying defects on the metal sheets, and another reason for focusing on more advanced algorithms to improve the classification accuracy of defects. [1]

1.1.3 Defects in hot rolled metal sheets surfaces

There are plenty of different defects that can occur in hot rolled metal surfaces but the six essentials that are relevant to the dataset used in the report are the following:

- **Rolled in scale:** The phenomenon of rolled in scale is a defect which occurs when the mill scale is rolled towards the metal through the rolling process. [9] ***Figure 1, (1)***

- **Scratches:** A scratch is a mark that is the absorption on a surface. Scratches are usually caused by contact that is not aware of the development of mechanical parts and mill components during rolling. [5][9] ***Figure 1, (2)***

- **Crazing:** Crazing is when cracks arise on the surface of the metal. [9] ***Figure 1, (3)***

- **Inclusion:** Inclusion is a common defect that occurs in metal surfaces. It's not always necessarily something major that affects the metal. Sometimes it's just ignored but there are cases when the area of the inclusion is large which then becomes a concern. [6][9] ***Figure 1, (4)***

- **Pitted surface:** Pitting is a type of corrosion that is seen as pockmarks. The reason for this is poor quality when the metal is rolled and results in different hardnesses on the surface. [7][9] ***Figure 1, (5)***

- **Patches:** A patch is identified as a part of metal that is marked out from the others by a particular property. [9] ***Figure 1, (6)***

1.1.4 Machine learning energy consumption

During recent years machine learning researchers have had the focus to produce highly accurate models without thinking about the energy consumption as a factor. Many times these deep learning algorithms can need megaflops of computational power and memory requirements to account for millions of parameters. [2] This requires a large amount of energy and raises the question if the cost of energy required can be made up from the result of the algorithm.

1.2 Purpose and goals

The aim of the project is to use an already existing dataset with the help of machine learning to identify six different types of defects in hot rolled metal sheet surfaces with at least 98% accuracy. This is chosen because 98% is a reachable goal for our limited hardware within a reasonable time. We hope to come to the conclusion that our method could in best case be applicable in industry to reduce manual labor.

The purpose with the project is to evaluate how machine learning can be implemented into material sciences based on literature and our results.

2 Method

2.1 Dataset

The dataset that was used was taken from the website Kaggle. Kaggle consists of many repositories of datasets uploaded by companies or individuals for people to use and apply machine learning algorithms to. [3]

2.1.1 NEU metal defects data

The dataset that was used comes from Northeastern University in Boston and consists of 1800 grayscale images with a resolution of 128 x 128 pixels of hot rolled metal sheet defects divided into six categories. The dataset was pre divided into 3 directories, train, validation and test which is essential to train the program. The training set is a set of examples that is used for learning and to fit the parameters of the model, the validation set is used for tuning the parameters of the model and the test set is used for assessing the performance of the model. [3]

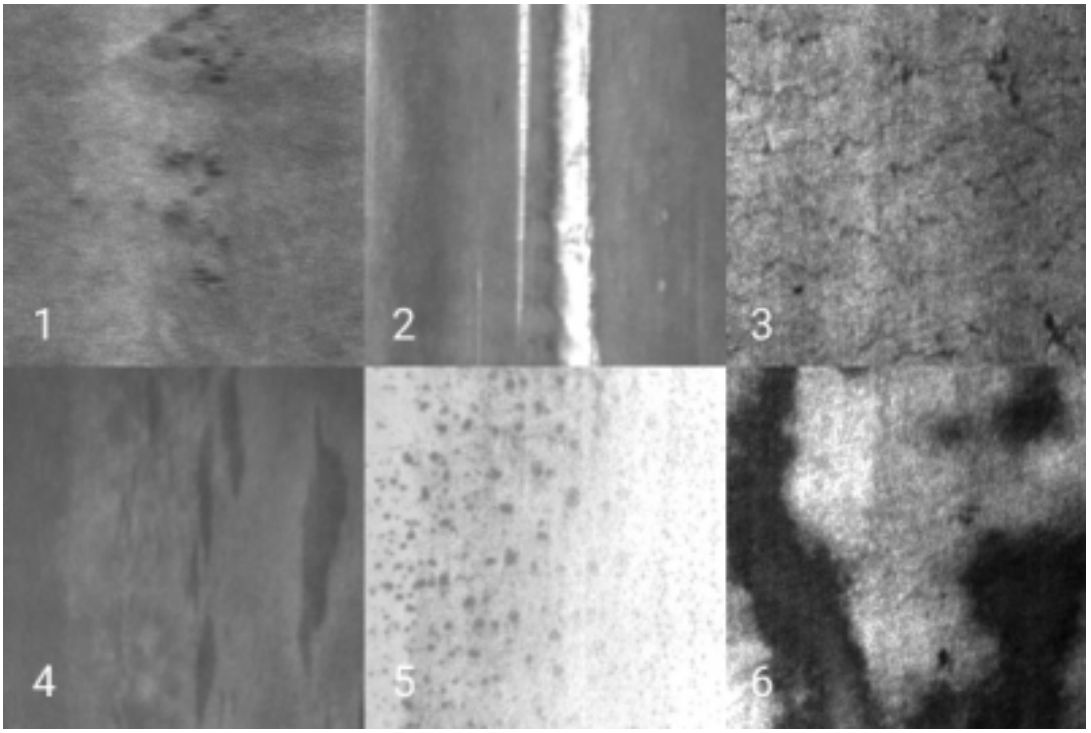


Figure 1. *Samples of the six types of surface defects.*
(1) *Rolled in scale,* ***(2)*** *scratches,* ***(3)*** *crazing,* ***(4)*** *inclusion,* ***(5)*** *pitted surface,* ***(6)*** *patches.*

2.2 Software

2.2.1 TensorFlow and Keras

TensorFlow is a program library with open source code specified for machine learning which is developed and used by Google. It can be used with python and it allows for program training on large neural networks by distributing the computation between many GPU servers. Keras is a deep learning API which runs on top of TensorFlow. It enables fast experimentation, from notion to result which makes it easy to work with. These combined provide the tools and models for image categorization. [14][15]

2.3 Code

With our basic understanding of machine learning and programming the code is taken from an example from kaggle which is then applied to the dataset. [16] The code was manipulated to run multiple epochs for different accuracies. For time efficiency we switched from standard CPU calculation to GPU calculation with the help of Nvidia's CUDA platform. This allowed us to run larger samples as we considerably lowered our computation time when switching from the CPU to the GPU, going from around 45 seconds computation time per epoch to around 13 seconds.

2.3.1 Model

The code utilizes the sequential model. It is a linear training model based on a layer to layer learning. It is appropriate for data where each layer exactly has one output tensor and one input tensor. The sequential model contains three so-called convolution blocks with a max pooling layer in every block. [17]

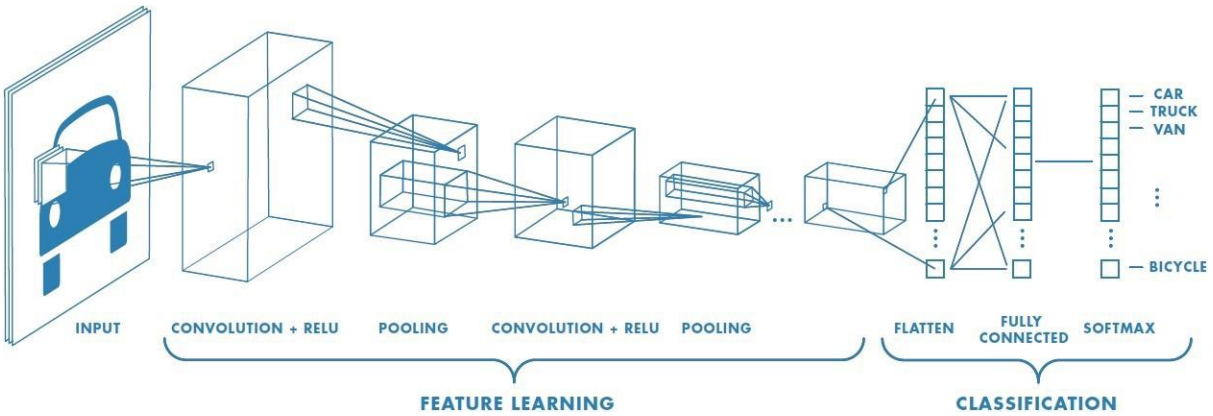


Figure 2. *An example image of the layering in a convolutional neural network* [13]

The two main layering operators the model uses are Conv2D and MaxPooling2D. The Conv2D layer creates a convolution kernel which is a small matrix that is used for image blurring, sharpening, edge detection and much more. The main objective of max pooling in the model is to downscale the images reducing the dimensionality which allows the model to

make assumptions about the features in the images. Stacking these two layers multiple times upon each other is done to improve the models accuracy. An example of the convolution layer and pooling layer can be seen in ***Figure 2***.

Other models that could've been used is a Convolutional Neural Network (CNN) such as the *"You only look once"* or YOLO model which is a deep learning model that is designed for fast object detection. [11] The main difference between our chosen sequential model and a CNN model is that the latter uses multiple inputs instead of a single input.

2.3.2 Accuracy, loss and epoch

When the layering is all done and compiled the model measures accuracy based on the number of correct guesses it's making out of the total amount of images, this can defined as:

$$Accuracy = \frac{n_{correct}}{n_{total}}$$

When the model makes predictions the loss is a number which indicates how bad of a prediction the model made. In other words loss is a measure of how many incorrect guesses were made in an epoch. [18]

The definition of an epoch is something that passes once over the whole dataset. The number of epochs represents how many times the entire dataset will be passed through the model during its training. There are two crucial types of problems with epoch optimization; these are called Overfitting and Underfitting. [4]

Common issues when training a model is making sure it doesn't get overfitted. When a statistical model is overfitted it can not accurately predict outcomes when getting data it has never seen before, this occurs when a model fits exactly against its training data. To combat making an overfitted model a dropout layer may be added which randomly sets inputs to zero at a set rate. Another way to make sure the model does not get overfitted is to stop the training after a certain time, usually when a condition is met.

On the other hand when a model is underfitted it hasn't trained enough and cannot quite capture the significance of the input variables. This also results in a model which cannot comprehend the data it's given and make accurate predictions. Making sure a model isn't getting underfitted is to feed it more data to train on. In ***Figure 3*** an example of a typical underfitted, appropriate-fitted and overfitted model can be seen.

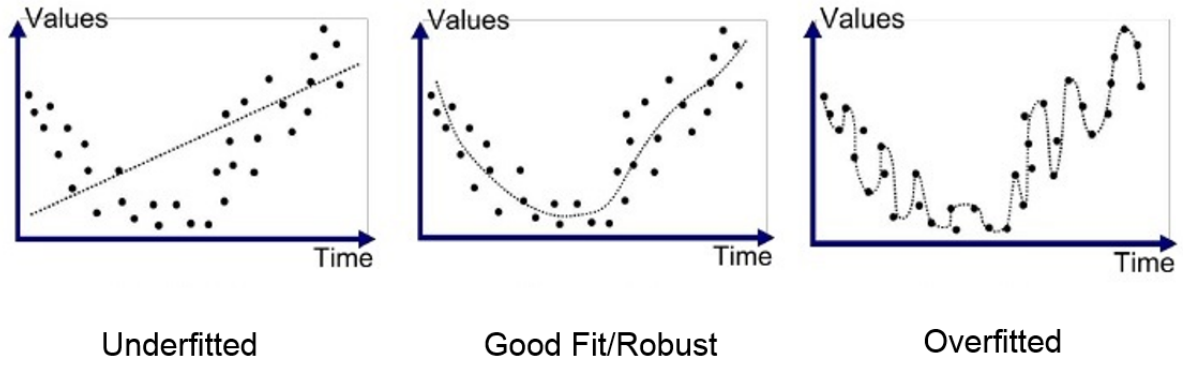


Figure 3. *Examples of a typically underfitted, appropriate-fitted and overfitted model* [12]

2.4 Evaluation

Following the previous steps successfully will provide a basic understanding of how machine learning is applied to a material science based on image categorization. With the help of this we will make conclusions based on our purpose and goals. With the results an evaluation will also be made if our method of identifying hot rolled metal sheets is sufficient.

3 Results

To bring it back to the task at hand, the previously stated goal of the project was: “To identify six different types of defects in hot rolled metal sheet surfaces with at least 98% accuracy.” To give further context on the problem definition the results have been reported in three sections: validation, sample, time estimation and energy consumption.

3.1 Validation

3.1.1 Accuracy

The accuracy condition from the code was to stop training the model when it reached 98% which occurred at 158 epochs for our model as can be seen in ***Figure 4***. Some deviations are easily spotted as the spikes downwards in the graph. They represent the epochs that have identified fewer correct images and are directly correlated to the big upswings seen in ***Figure 5***. The training data had an average accuracy of 94.47% over the 158 epochs and the validation data had an average accuracy of 95.74% over the same period.

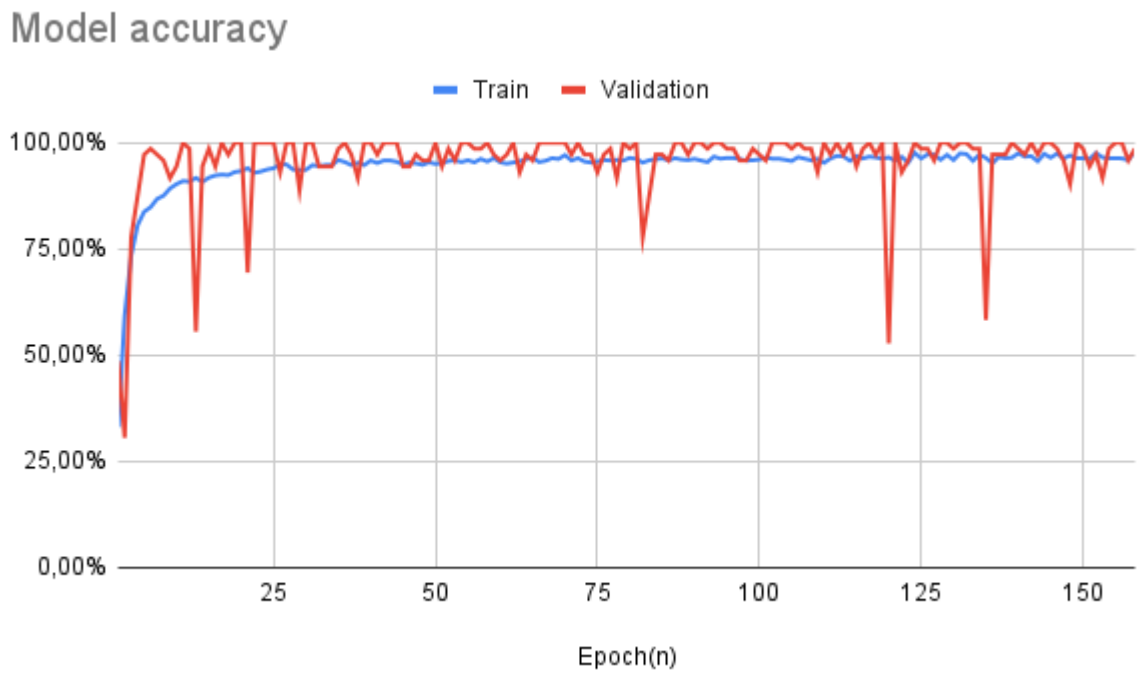


Figure 4. *Accuracy of the model plotted against the number of epochs. The blue curve being the training accuracy and the red curve being the validation accuracy*

3.1.2 Loss

Looking at the results from ***Figure 5*** the big upswings at epoch 120 and epoch 135 are easy to identify as epochs where the loss was substantially bigger.

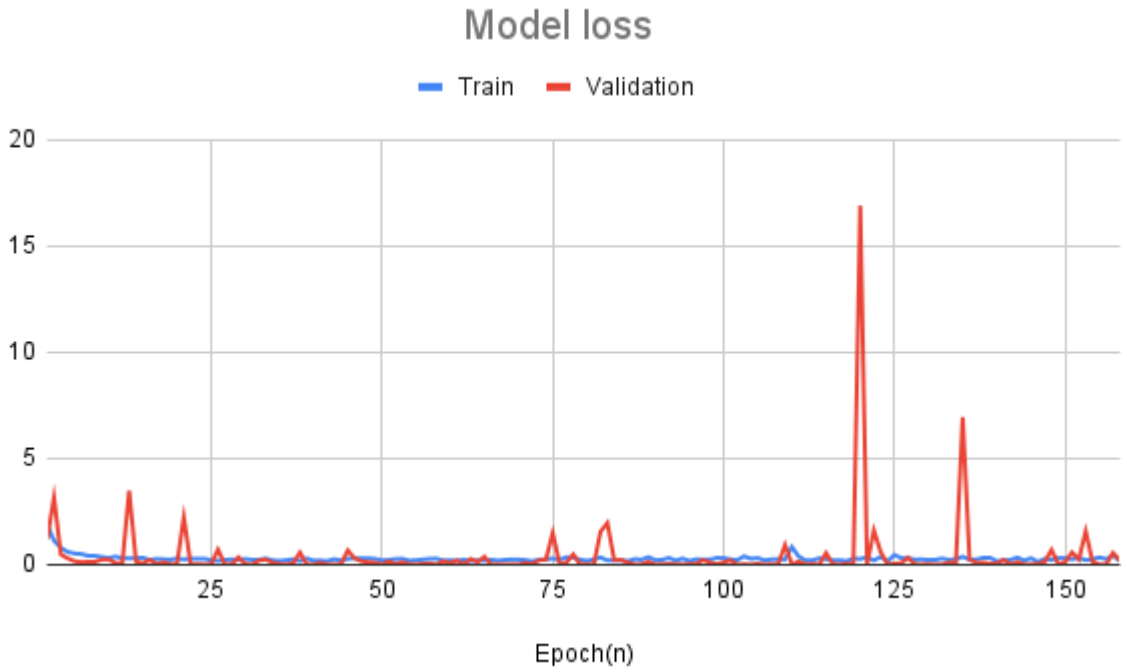


Figure 5. *Loss of the model plotted against the number of epochs. The blue curve being the training loss and the red curve being the validation loss.*

3.2 Sample

Once done training the model the code choses random pictures from the test data set which the model hasn't seen and displays a sample as seen in ***Figure 6***. Here the correctly guessed defects are marked in green and the not correct defects are marked in red with the correct answers in parenthesis. The sample picture is given in a 5 by 5 format which gives us a sample of 25 metal defects from the dataset. In the sample 22/25 of the defects are correctly guessed which is an 88% correct estimation of the defects.

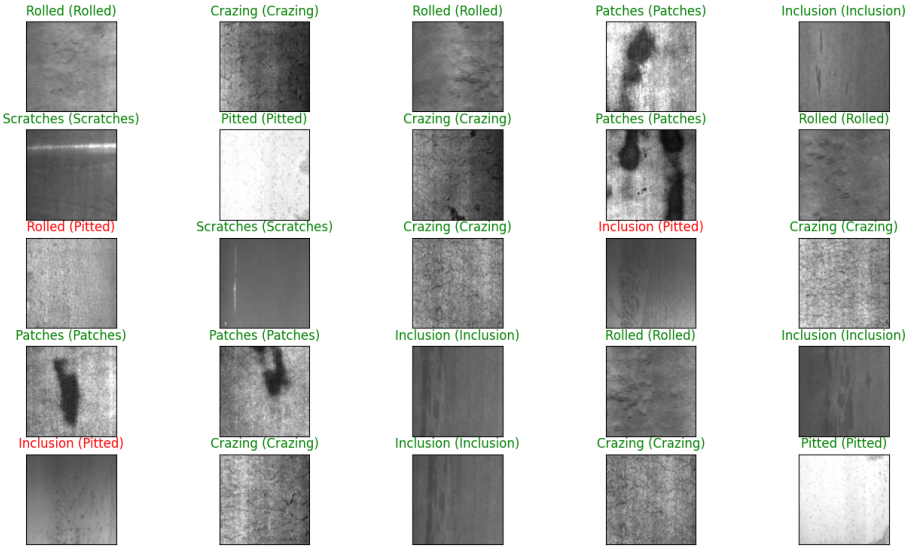


Figure 6. *Pictures from the test dataset with the prediction from the model and the correct answer in the parenthesis next to the predication.*

3.3 Time estimation

In ***Figure 7*** the time in seconds it took per epoch is shown. The average time per epoch was approximately 13.36 seconds, with a total training time of around 35 minutes for 158 epochs.

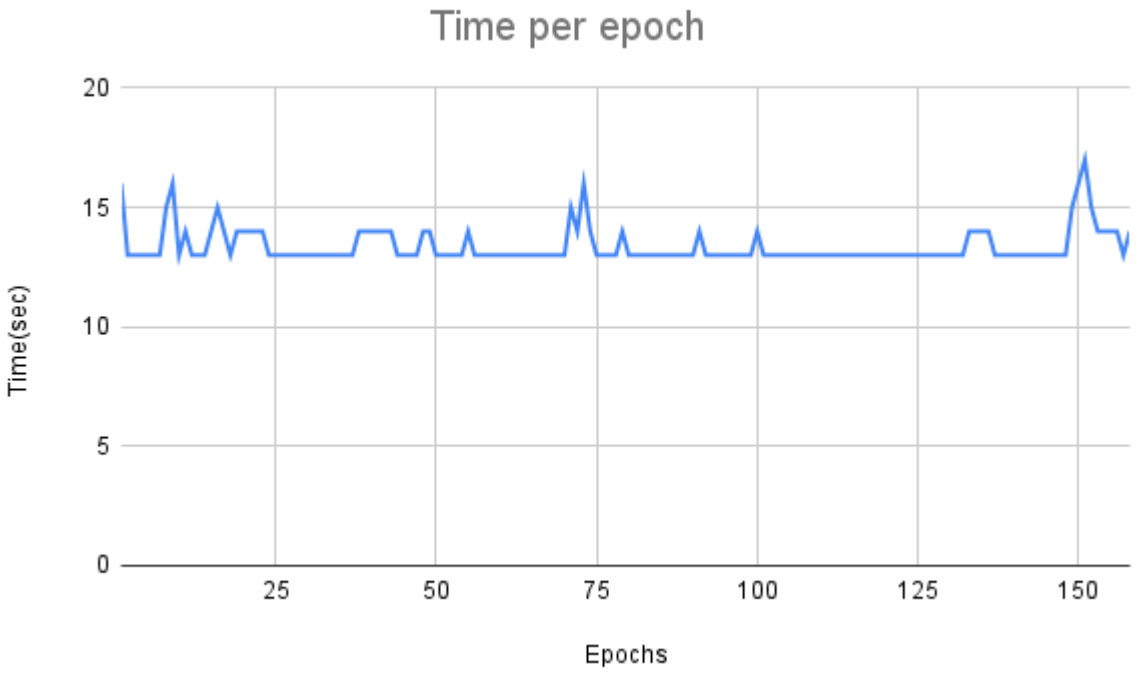


Figure 7. *Time in seconds plotted against the number of epochs.*

3.4 Energy consumption

The model trained on a system using a 800W power supply. Assuming all the energy in the system was used for the training at 100% efficiency, running the model for 35 minutes takes 1.68 MJ of energy.

4 Discussion

In the way that sequential training uses the new data and updates itself to improve every decision making it can be interpreted as a “Bottom up” approach. That means every time the algorithm processes an epoch the accuracy level of the result increases and is therefore mimicking the evolution and self learning process which was previously mentioned.

4.1 Code efficiency

The quality of data which is used in the program is very crucial to raise the accuracy of the algorithm. The lack of the surface data of the hot rolled metal sheets is the main problem, in addition this can cause imbalance in the number of samples which are used to distinguish and classify different types of defects. Deeper network levels are not very helpful when the amount of data is not broad enough. The lack of data can further decrease accuracy, if there is not enough data for a specific defect classification, such as oxides for example, it can be dealt with by regulating the sample size. The other way is to increase the data manually and combine it with deep learning methods, which may improve the accuracy when data is not sufficient.

4.1.1 Is our method fit for the industry?

Considering the economical aspects of having manual labor to do quality inspections of large quantities of metal sheets versus the algorithm, requiring 1.68 MJ of power for training, and some optical equipment it can be said this type of algorithm can be beneficial. Achieving higher accuracies will be of further benefit, but only up to a certain point, since a 100% accuracy for arbitrary input is never achievable, where there exists an optimal train and profit margin for a specific timespan.

However, the energy aspect for our case cannot be compared to all types of models. Certain projects simply require one large training session, like the ones that need megaflops of processing power and require significantly more energy. If the energy cost of this process is larger than the profit of the discovery it raises the question if it is really worth doing in the first place. This can be comparable to the crypto mining dilemma where the energy cost of mining must be lower than the profits of the crypto mined. Which puts a requirement for such projects to make a cost versus reward prediction.

4.2 Result reliability

Factoring in our results and the common pitfalls of training a model, exactly how reliable is the result we have reached?

4.2.1 Overfitted or underfitted

Is the model we trained overfitted or underfitted? Looking at our results and comparing the average accuracy of the training data and the average accuracy of the validation data it’s easy

to see that the two doesn’t stray that far away from each other which tells us that our model is quite alright and neither overfitted nor underfitted.

4.2.2 Epochs and sample

Why didn’t we keep training the model? Why stop at epoch 158 which gave us 98% accuracy? Say instead of setting the condition to stop training at 98% we set a fixed number of epochs for example 600, wouldn’t it yield a better accuracy? Well not necessarily,this comes back to the problem with overfitting the model where if it would've had trained over 600 epochs it most definitely would skewed our model and it would’ve not been able to accurately predict the defects.

The sample result of 88% or 22/25 correct predictions maybe isn’t the best when our model has 98% accuracy, but we have to factor in that it is only a sample of 25 images and if we would’ve done a wider sampling the result most likely would’ve been a lot better.

4.2.3 Different approach

Was the model we used to predict these defects the best available to us? The sequential model is rather basic when it comes to visual imagery, it does a good job but the time spent running the code and training the model is quite substantial when compared to a CNN model such as the YOLO model which is more or less created for predicting visual imagery. Then why didn’t we use such a model? Well, the time spent coding would’ve drastically gone up and considering we only have a basic understanding of programming there might’ve been a risk that we would’ve failed and not gotten any result at all.

4.3 Material science use for machine learning

Identifying defects in hot rolled steel sheets would consume a lot of time if it would be done manually. To do it manually a person would have to look at every picture individually to identify it by hand. It would consume a lot of time and probably wouldn't be as accurate as the program used. Just by comparing times for something that would probably take hours of work for a human being is just taking approximately 35 minutes for the computer..

It's also crucial to actually know what the defects on the surface are caused by. If there is a majority of any kind of defect there would probably be some kind of manufacturing problem and it is crucial to identify those problems for future production.

There are also a lot of other areas in material science that Ai is relevant to as discussed above and it just keeps developing.

4.4 Social and ethical aspects

AI and machine learning are tools which can be used for different objectives. In this report the sequential method which is a type of machine learning is used for finding and classifying

defects on hot rolled metal sheets. This is an example of ethically good application of machine learning, on the other hand similar methods in machine learning and AI can be used to create murder robots and war machines.

5 Conclusions

- Our primary goal of reaching 98% accuracy was reached. Thanks to the program running on the GPU instead of the CPU the learning phase was decently fast for our limited hardware.

- The accuracy of 98% could in theory work for the industry. But a more trained algorithm will always be better.

- There exists other solutions to this problem, for example using the YOLO method. But due to restricted time and knowledge in this sector it was ignored.

- If an extensive machine learning project requires a great amount of processing power over a longer period of time, predictions must be made to evaluate if the calculation results outweigh the energy cost of the calculation.

- Machine learning is something relevant for material science. It can be used in plenty of different departments, in our case for image classification of metal defects.

5.1 Future work

For future work there are plenty of things which can be either improved or further developed.

The implementation to import any picture of hot rolled metal sheet surface to detect any defects with the help of the existing code is something that would improve the program alot for it to be useful in plenty of industries.

Also there is a lot of room to improve the actual code to make it more efficient eventually even creating a new dataset from scratch to have for example more types of defects and a higher accuracy.

6 Acknowledgements

We send our dearest regards and appreciation to Anders Eliasson and Mikael Ersson, examinator and supervisor, at the Royal Institute of Technology for all their guidance and support during this project.

7 References

[1] M. Kaku, *Physics of the Impossible: A Scientific Exploration Into the World of Phasers, Force Fields, Teleportation, and Time Travel*, New York City: Anchor Books, 2009.

[2] E. García-Martín, C. Faviola Rodrigues, G. Riley and H. Grahn, *Estimation of energy consumption in machine learning*, Journal of Parallel and Distributed Computing, vol. 134, pp. 75-88, 2019. Available: https://doi.org/10.1016/j.jpdc.2019.07.007 [Accessed 1 March 2022].

[3] F. Islam. Available: https://www.kaggle.com/fantacher/neu-metal-surface-defects-data [Accessed 1 March 2022].

[4] S. Afaq and Dr. S. Rao, *Significance Of Epochs On Training A Neural Network*, International Journal of Scientific & Technology Research, vol. 9, no. 6, pp. 485-488, 2020. Available: https://www.ijstr.org/final-print/jun2020/Significance-Of-Epochs-On-Training-A-Neural-Network.pdf [Accessed 1 March 2022].

[5] Precision Machine Products Association, *Scratches On Rolled Steel Products*. Available: https://pmpaspeakingofprecision.com/2012/06/12/scratches-on-rolled-steel-products/ [Accessed 3 March 2022].

[6] Rolled Alloys, *Inclusions and Laminations*. Available: https://www.rolledalloys.com/technical-resources/blog/inclusions-and-laminations [Accessed 3 March 2022].

[7] Metallic Steel, *Analysis of common surface defects of hot rolled steel sheet*, 2020. Available: https://www.metallicsteel.com/analysis-of-common-surface-defects-of-hot-rolled-steel-sheet.html#Pitting [Accessed 3 March 2022].

[8] W. Sha, Y. Guo, Q. Yuan, S. Tang, X. Zhang, S. Lu, X. Guo, Y-C. Cao and S. Cheng, *Artificial Intelligence to Power the Future of Materials Science and Engineering*, Advanced Intelligent Systems, vol. 2, no. 4, pp. 1-12, 2020. Available: https://doi.org/10.1002/aisy.201900143 [Accessed 1 March 2022].

[9] X. Feng, X. Gao and L. Luo, *X-SDD: A New Benchmark for Hot Rolled Steel Strip Surface Defects Detection*, Symmetry, vol. 13, no. 4, pp. 1-16, 2021. Available: https://doi.org/10.3390/sym13040706 [Accessed 1 March 2022].

[10] N. Heess, J. Merel and Z. Wang, *Producing flexible behaviours in simulated environments*, 2017. Available: https://deepmind.com/blog/article/producing-flexible-behaviours-simulated-environments [Accessed 1 March 2022].

[11] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, *You Only Look Once: Unified, Real-Time Object Detection*, 2016. Available: https://arxiv.org/pdf/1506.02640v5.pdf [Accessed 1 March 2022].

[12] K. Hoffman, *Machine Learning: How to Prevent Overfitting*, 2021. Available: https://medium.com/swlh/machine-learning-how-to-prevent-overfitting-fdf759cc00a9 [Accessed 1 March 2022].

[13] S. Saha, *A comprehensive Guide to Convolutional Neural Networks - the ELI5 way*, 2018. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 [Accessed 1 March 2022].

[14] TensorFlow. Available: https://www.tensorflow.org/ [Accessed 3 March 2022].

[15] Keras. Available: https://keras.io/ [Accessed 3 March 2022].

[16] F. Islam. Available: https://www.kaggle.com/fantacher/metal-surface-defects-inspection [Accessed 3 March 2022].

[17] TensorFlow, *tf.keras.Sequential*, 2022. Available: https://www.tensorflow.org/api_docs/python/tf/keras/Sequential [Accessed 1 March 2022].

[18] Google Developers, *Classification: Accuracy*, 2020. Available: https://developers.google.com/machine-learning/crash-course/classification/accuracy [Accessed 3 March 2022].

8 Appendix

https://github.com/ackemag/MH1025-Project