Severstal defect segmentation
IACV 2019-2020 Project

Andrea Bionda
Politecnico di Milano
andrea.bionda@mail.polimi.it

Abstract

Steel is one of the most important building materials of modern times. It was observed that the future of metallurgy requires the development of even more technological tools to face contemporary economic, ecological, and social challenges. In order to increase efficiency in production, Severstal collected gigabytes of data looking for a deep learning approach that helps them in recognizing and classifying automatically surface defects on steel sheets. The goal of this challenge is not only to provide a good segmentation and classification of defects, but also an accurate classification between defective and non-defective images, in order to reduce false positives, and so waste of money and resources. In this paper, I will try to give a solution to the above problem, using a combination of two famous deep learning tools, named classification and segmentation respectively. The evaluation metrics used in this competition is the so called Dice coefficient, which is not only an accuracy index, but it also penalizes the false positives that the method finds, more similar to precision. (172)

The abstract should be self contained and explain what the paper is about. Usually abstracts are no longer than 300 words. You should state what are the main contributions of your work and tempt the reader to continue to read your paper. A good abstract briefly describes your problem, approach, and key results. This document is based on the CVPR submission template, and it has been adapted to submit a technical report of a project of the Deep Learning and Image Classification course.

1. Introduction

The main contribution of this paper is to present a method to solve the above problem, that provides a good tradeoff between accuracy on defect segmentation and precision in defective-image identification.
The proposed solution takes in input images of steel sheet that come from high-frequency cameras and output the same-size segmented images, which contain the area of the defect(s) identified (if any) and the corresponding type.
The first explored approach is to use a single Fully Convolutional Neural Network that performs semantic segmentation over the four defects; in particular I have used U-Net architecture. This approach gives a Dice coefficient of 0.48 in the test set. In order to improve the precision, and so reducing false positives, the second and final approach is a pipeline of two different techniques: in the first stage is present a Binary Classifier that is trained to filters out defect-free images; in the second stage, it is placed the segmentation model presented in the first approach, that will segment only images that are labelled defective by the Classifier. This second approach gives a Dice coefficient of 0.71 in the test set.

This is the first section of your paper and you should contextualize the problem you are working on, why it is important, and give an overview of your results. It is useful to list all the contributions of your work very clearly, so that the reviewer can easily understand the value of your paper. This is helpful to guide the reader through the paper. If your project consists in replicating and/or extending another paper, then you should be very clear about it explaining what you did and how you proceeded. For instance, the main contribution of this paper are:

- A brief tutorial on how to write a paper/technical report.
- Specific good/bad examples of writing.
- An example of structure that is suggested to follow for a technical report and can be reorganized as you need.

Each researcher has its own style and preferences when it comes to write a paper, so feel free to modify the structure as it pleases you. This tutorial is especially directed to master students who might need more help to organize their reports.

1.1. Language

All manuscripts must be in English. Do not use contraction forms, *e.g. doesn't → does not*, *isn't → is not*.

1.2. Paper length

Papers, excluding the references section, must be between **4-6** pages, bust must be no longer than **six pages**. The maximum number of pages is strict. Overlength papers will simply not be graded or graded negatively. Note that this LaTeX guide already sets figure captions and references in a smaller font. The references section will not be included in the page count, and there is no limit on the length of the references section. For example, a paper of six pages with two pages of references would have a total length of 8 pages, so it is fine.

2. Related work

2.1. Semantic Segmentation

spiegone

2.2. U-Net

spiegone

2.3. FPN (se non ho alternative)

spiegone

2.4. Image classification (boh)

spiegone

In this section you should discuss published work that relates to your project. This is expected to be full of references, meaning that you have read the existing literature and you know what you are working on very well. This is not just a list or works, but you are not supposed to cluster papers that use similar approaches and compare them each other using very short sentences. I strongly suggest to check [4, 5] blog posts for some good practices in writing a paper. A quote that I like very much is *"Research is spending 6 hours reading 35 papers, so you can write one sentence containing 2 references"* [3]. Keep it in mind while you are writing!

3. Dataset analysis

The first, and one of the most important phase of every Machine Learning task, is the analysis of input data. In this problem we have 12568 greyscale images of size 1600x256x3 that comes from real camera shot of steel sheets and a data file (.csv) that contains all the information in order to generate each defective mask; this dataset is obtained from [1].The types of defects can be classified in

Figure 1. Pitting

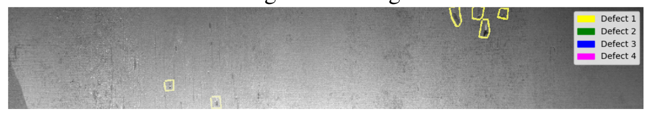


Figure 2. Inclusion

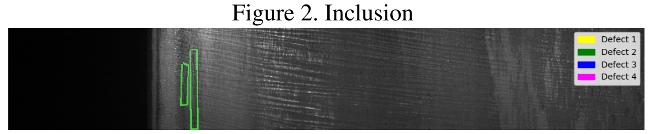


Figure 3. Scratches

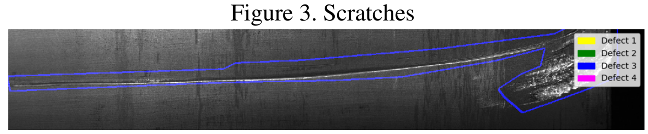
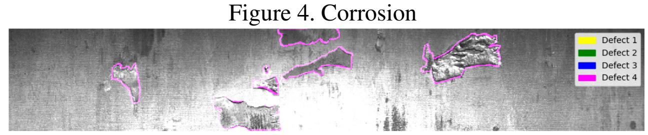


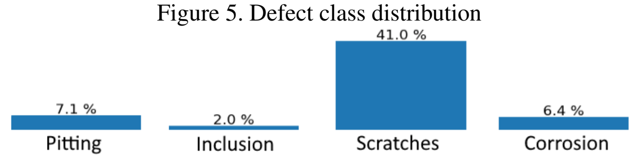
Figure 4. Corrosion



Pitting (Figure 1), Inclusion (Figure 2), Scratches (Figure 3) and Corrosion (Figure 4).

The dataset exhibit a fairly even distribution between defective (56%) and non-defective images (44%), but among defective ones we can notica a strong imbalance between defect classes (Figure 5). This distribution can lead to recognize only one type of defect (the third one), since it represents the largest part of the defective images. During the discussion of the proposed approach I will explain a solution to this problem.

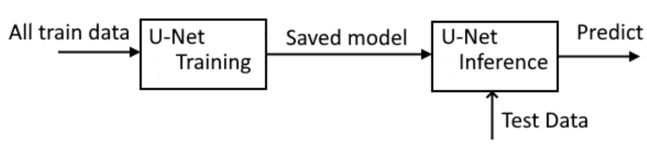
Figure 5. Defect class distribution



4. Proposed approach

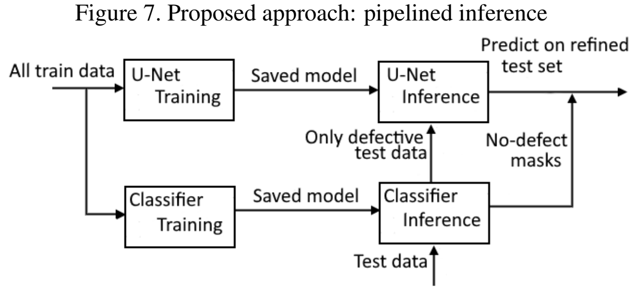
The first and straightforward solution that I have applied to the problem is build a U-Net architecture for semantic segmentation in such a way that outputs same-size images with 4 channels, one for each defect class (Figure 6). Each channel(layer) of the output represents the heatmap (image where for each pixel is assigned a probability that it belongs to a defect in the original image) of the corresponding defect class.

Figure 6. First approach: segmentation inference only



This method has a very good capibility in defect recognition, but at the same time it hardly distinguishes real defects from false defects due to their similar pattern (an example on figure xxx). This problem is higly penalized in the case in which no defects is present with Dice metric (link to the formulae), that could be reasonable in high capacity foundries where the goal is also reduce at minimum the false scrap and so wasting resources (if the machine labels part of the steel sheet as defective it will be throwed away).
For this reason the second, and real proposed approach, is to train a Binary classifier with the goal of distinguish defective from non-defective images. The idea is to use the classifier to decide, during inference time, weather an image could contain a defect or not and if the answer is true pass it to the U-Net to segment the defect, otherwise the segmentation phase is skipped and whole-zero masks are generated as heatmaps (Figure 7).
This pipeline will reduce the false scrap and increase Dice coefficient as you can see in the result section (result section).

Figure 7. Proposed approach: pipelined inference



This is the core of your paper, where you describe the details of the proposed method for solving the problem that you set up in the introduction. This is the most important section. It has to be clear why the chosen approach is the right thing to do with respect to the possible alternatives. The explanation of the method has to be readable and understandable and it should not raise obvious questions from the reader. You can divide the section in paragraphs or subsections that can be useful for the presentation of your method. Usually at this point you may want to place equations, figures and tables to clarify what you are explaining.

4.1. Mathematics

Please number all of your sections and displayed equations. It is important for readers to be able to refer to any particular equation. Just because you didn't refer to it in the text does not mean some future reader might not need to refer to it. It is cumbersome to have to use circumlocutions like "the equation second from the top of page 3 column 1".

4.2. Footnotes

Please use footnotes[1] sparingly. Indeed, try to avoid footnotes altogether and include necessary peripheral observations in the text (within parentheses, if you prefer, as in this sentence). If you wish to use a footnote, place it at the bottom of the column on the page on which it is referenced. Use Times 8-point type, single-spaced.

4.3. References

List and number all bibliographical references in 9-point Times, single-spaced, at the end of your paper. When referenced in the text, enclose the citation number in square brackets, for example [2]. Where appropriate, include the name(s) of editors of referenced books.

4.4. Illustrations, graphs, and photographs

All graphics should be centered. Please ensure that any point you wish to make is resolvable in a printed copy of the paper. Resize fonts in figures to match the font in the body text, and choose line widths which render effectively in print. Many readers (and reviewers), even of an electronic copy, will choose to print your paper in order to read it. You cannot insist that they do otherwise, and therefore must not assume that they can zoom in to see tiny details on a graphic.

5. Experiments

In this section you validate your method showing the experiments that you performed. The experiments will vary depending on the project, but you might compare with previously published methods, perform an ablation study to determine the impact of various components of your system, experiment with different hyperparameters or architectural choices, use visualization techniques to gain insight into how your model works, discuss common failure modes of your model, etc. You should include graphs, tables, or other figures to illustrate your experimental results. Divide in subsections or paragraphs to help the reader navigate in your paper.

Datasets. Describe the data you are working with for your project. Usually you need to explain what type of data is it, how much data are you working with and if you applied any pre-processing, filtering, or other special treatment to use it. Remember that you have to cite each dataset you used in your project if it has been published from someone else. Instead, if you collected it by yourself you have to describe accurately how you gathered (and labeled) your data.

[1] This is what a footnote looks like. It often distracts the reader from the main flow of the argument.

Figure 8. Example of caption. It is set in Roman so that mathematics (always set in Roman: $B \sin A = A \sin B$) may be included without an ugly clash.

Experiments setup. Here you describe all the architectural choices of your model, the hyper-parameters of your model, *e.g*. optimizer, learning rate, momentum, batch size and if you cross-validate on them.

Results and discussion. Discuss your results and compare with other methods. You can also perform an *ablation study* on your model switching on and off some components to understand their contributions.

6. Conclusion

Summarize your key results. What have you learned from the project and suggest future extensions or new applications of your ideas.

Acknowledgements

Here you can thank all the people that have been helpful during the project, but are not coauthors. It is important to add this section only in the final version that you send to the professors and **NOT in the review version** in order to not compromise the double blind review process. We thank Andrea Romanoni for the insightful comments during the review of the manuscript, and all the old and new Gods who stand beside us when the deadline is dark and full of terrors.

A. Supplementary Material

Supplementary material is not counted toward your 4-6 page limit and should be submitted as a separate zip file. Your supplementary material might include:

- Source code (if your project proposed an algorithm, or code that is relevant and important for your project.).
- Cool videos, interactive visualizations, demos, etc.

Method	Frobnability
Theirs	Frumpy
Yours	Frobbly
Ours	Makes one's heart Frob

Table 1. Results. Ours is better.

Examples of things to NOT put in your supplementary material

- The entire PyTorch/TensorFlow Github source code.
- Any code that is larger than 10 MB.
- Model checkpoints.
- A computer virus.

References

[1] Severstal: steel defect detection. 2

[2] Authors. The frobnicatable foo filter, 2014. Face and Gesture submission ID 324. Supplied as additional material `fg324.pdf`. 3

[3] Bryan Gaensler. Tweet, 2018. https://twitter.com/SciBry/status/989623027393531905. 2

[4] Jacob Steinhardt. Advice for Authors, 2018. https://jsteinhardt.wordpress.com/2017/02/28/advice-for-authors/. 2

[5] Zachary C. Lipton. Heuristics for Scientific Writing (a Machine Learning Perspective), 2018. http://approximatelycorrect.com/2018/01/29/heuristics-technical-scientific-writing-machine-learning-perspective/. 2

Figure 9. Example of a short caption, which should be centered.