A Project Report

On

CLASSIFICATION OF SURFACE DEFECTS ON A STEEL SHEET

submitted for partial fulfillment of the requirements of course
Machine learning Lab (CS5101)

By

ATHIRA S MENON (11114007)
MUHAMMED FAVAS K (142102007)
VIJAY KUMAR M (142102015)

under the guidance of

Dr. Sahely Bhadra

IIT PALAKKAD

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD
PALAKKAD - 678557, KERALA

Acknowledgment

We would like to thank **Dr. Sahely Bhadra** for her constant motivation and invaluable guidance and suggestions. During our course project '**Classification of surface defects on a steel sheet**', we were able to implement those concepts which we learned during the Machine Learning lab course. We are also grateful to all our TAs for their kind cooperation which helped us in the completion of this project.

Nov 17, 2021
IIT Palakkad

Group-1

Abstract

Steel is one of the most important building materials of modern times. Steel buildings are resistant to natural and man-made wear which has made the material ubiquitous around the world. To help the production of steel more efficiently, this project will help identify defects. In this project, we are classifying surface defects on a steel sheet based on Convolutional Neural Network (CNN) models. Binary classification is performed on the given dataset to filter no defect images and a multi-label classification to predict probabilities of images belonging to each defect class. Later image segmentation is performed for different classes to generate masks for each test image. Generated masks are converted to encoded pixels based on run length encoding (RLE) and filter them as per classification probabilities. The Efficiency of the implemented method is evaluated based on the dice coefficient. The Dice coefficient can be used to compare the pixel-wise agreement between a predicted segmentation and its corresponding ground truth.

Contents

Part.1

Introduction

1.1 Motivation

Steel is one of the most important building materials of modern times. Steel buildings are resistant to natural and man-made wear which has made the material ubiquitous around the world. Identifying defects will help the production of steel more efficiently. To help the production of steel more efficiently, this project will help identify defects. In this project, we try to classify surface defects on a steel sheet based on convolutional neural network models.

1.2 Data Description

- Training Image - 12568 images of size 256x1600 resolution is used for training the model
- Test images - 5506 images are used to test the model
- Train.csv - Contains the details of defected images such as type of defect (Class Id = [1, 2, 3, 4]) and their corresponding Encoded Pixels
- Sample submission.csv - A sample submission file in the correct format

1.3 Tool and Technologies

Programming Language: Python

Python Libraries:

- Pandas: Python library used to perform data manipulation and analysis
- NumPy: Provides support for large, multi-dimensional arrays and matrices along with a large collection of high-level mathematical functions to operate on these arrays.
- TensorFlow: Machine learning framework used for training and inference of deep neural network
- Keras: It acts as an interface for TensorFlow and enables fast experimentation with deep neural networks.
- Scikit-learn: It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python.
- Seaborn -Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis.

1.4 Image classification using CNN

Convolutional Neural Network (CNN) is a class of deep learning neural networks used to analyze visual imagery and is frequently used as a backbone for image classification. The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. Convolutional networks are a specialized type of neural network that uses convolution in place of general matrix multiplication in at least one of their layers. Image classification is the process of taking an input (like an image) and outputting a class (like “cat”) or a probability that the input is a particular class (i.e., a 90% probability that input is a cat”).

In binary classification problem user tries to classify an entity into one of the two possible categories. Multi-label classification is a generalization of multiclass classification, which is the single-label problem of categorizing instances into precisely one or more than two classes; in the multi-label problem, there is no constraint on how many of the classes the instance can be assigned to.

1.5 Image segmentation

Image segmentation is the process of partitioning a digital image into multiple segments i.e., sets of pixels, also known as image objects. The goal of segmentation is to simplify or change the representation of an image into something more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries in images.

Image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. With the help of image segmentation, we can partition the image into multiple segments. This will make it easy to learn from patterns in these multiple segments.

Part.2

Methodology

The objective of this project is to predict the location and type of defects found in steel manufacturing. Images are named with a unique Image Id. In this project, we try to segment and classify the defects in the test set. Each image may have no defects, a defect of a single class, or defects of multiple classes. For each image we segment defects of each class (Class Id = [1, 2, 3, 4]). The segment for each defect class will be encoded into a single row, even if there are several non-contiguous defect locations on an image.

2.1 Data preprocessing and analysis

Given dataset mentioned in section 1.2 is preprocessed by merging details from train.csv and training image folder and obtained preprocessed data is used in the training phase as input. Data cleaning and extraction are performed as part of data preprocessing. During data analysis, preprocessed data are visualized as bar charts to understand the distribution of images with defects and without defects. From the visualization of data, we can observe that data is imbalanced and more than half of the dataset belongs to the no defect category.

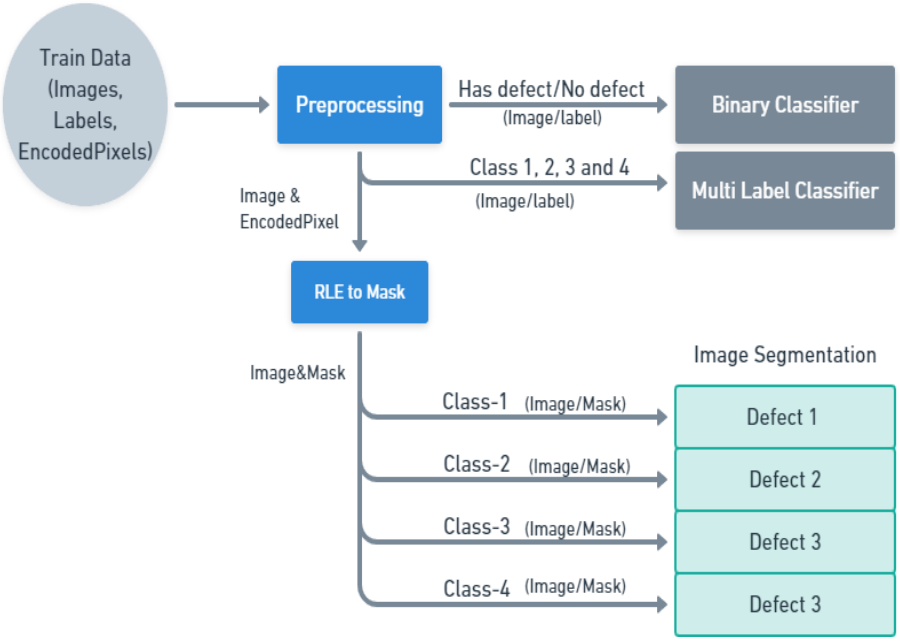

Figure 2.1: Model training pipeline

2.2 Training Phase

Binary classification is performed to filter images with defects and later multi-label classification is performed to classify images into defect classes [1,2,3,4]. Image segmentation is performed for different defect classes to generate masks for each test image. Generated masks are converted to encoded pixels and filtered as per classification probabilities. Figure 3.1 shows the pipeline for training the model

2.2.1 Binary Classification

Convolutional Neural Network (CNN) model with a base model as Xception from the Keras application is used for performing binary classification. Model is trained with entire train images to obtain good accuracy. This model is used to filter images with no defect.

- Xception [1] by Google, which stands for Extreme version of Inception, is reviewed. It is a deep convolutional neural network architecture that involves Depthwise Separable Convolutions.

2.2.2 Multi-Label Classification

In Multi-label classification images with defects are classified into one of the defect classes as 1,2,3 and 4. Some images may have more than one defect class. A CNN model similar to a binary classifier with the base model as Xception is used to perform multi-label classification.

2.2.3 Run Length Encoding (RLE)

Run-length encoding (RLE) is a form of lossless data compression in which runs of data (sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run.

2.2.4 Segmentation Models

Image segmentation is performed for each defect class to generate a mask for test images. In this project, we have used UNet architecture with the EfficientNet backbone and used pre-trained weights for faster convergence. Mask is generated based using encoded pixels by the Run Length Encoding technique.

- UNet: Unet is a popular semantic segmentation architecture, which uses a Fully Convolutional Neural Network model [2]. The network is based on the fully convolutional network and its architecture was modified and extended to work with fewer training images and to yield more precise segmentation. Segmentation of a 512 × 512 image takes less than a second on a modern GPU.

- EfficientNet: EfficientNet model was proposed by Mingxing Tan and Quoc V. Le of Google Research, Brain team [3]. They proposed a new scaling method that uniformly scales all dimensions of depth, width and resolution of the network. They used the neural

architecture search to design a new baseline network and scaled it up to obtain a family of deep learning models, called EfficientNets, which achieve much better accuracy and efficiency as compared to the previous Convolutional Neural Networks.

2.3 Testing and Validation Phase

The performance of the binary classifier and multilabel classifier are monitored using metrics such as accuracy, precision, recall and f1 score. The Loss function used in classification models is binary cross-entropy. Dice coefficient is used for evaluating the performance of the segmentation model.

- Dice coefficient is the ratio of two times the area overlapped by the ground truth mask and predicted mask to the total area of both masks. Here area referred to the number of pixels.

$$Dice\ coefient\ (X,Y) = \frac{2|X \cap Y|}{|X \cup Y|}$$

2.4 Prediction Phase

During the prediction phase given test data is used to predict the class using binary and multi label classification models and the segmentation model is used to predict encoded pixel value using the run length encoding technique. Figure 3.2 depicts the prediction pipeline.

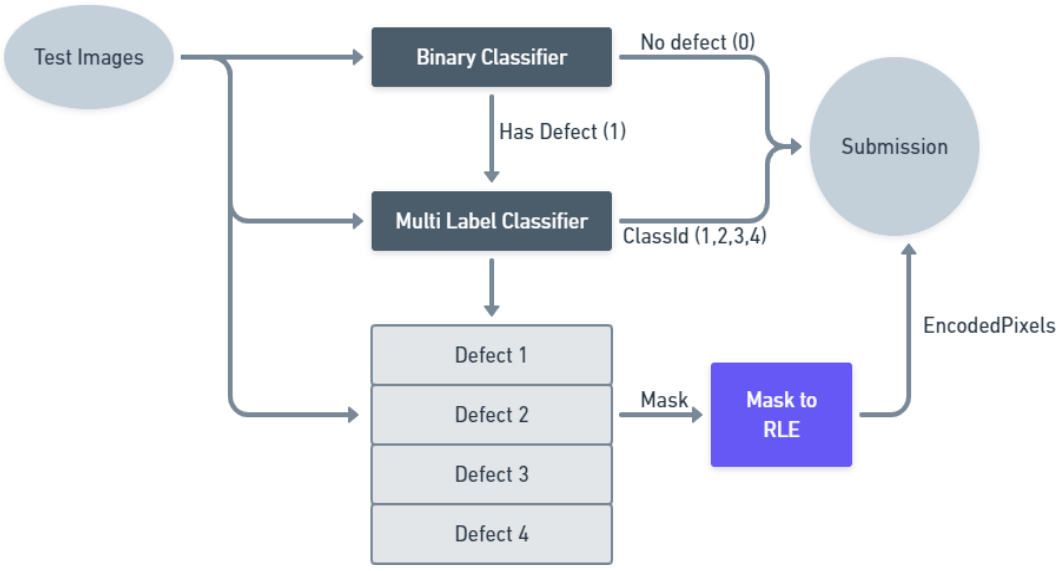

Figure 2.2 Prediction pipeline

2.5 Result

In this project, we have obtained an accuracy of 79.58 % in classification and a mean dice coefficient of 67.22 %. Individual performance of each model is shown in the tables below.

Metric	Binary Classifier			Multi-label Classifier		
	Train data	Validation Data	Test Data	Train data	Validation Data	Test Data
Precision	0.94	0.93	0.95	0.92	0.92	0.90
Recall	0.93	0.92	0.91	0.85	0.86	0.85
F1 Score	0.93	0.92	0.92	0.89	0.89	0.88
Accuracy	0.93	0.92	0.93	0.86	0.87	0.86

Table 2.1 Performance metrics of classification models

Model	Train data	Validation Data	Test Data
Defect -1	0.67	0.64	0.60
Defect -2	0.73	0.67	0.64
Defect -3	0.71	0.70	0.70
Defect -4	0.79	0.75	0.78

Table 2.2 Dice coefficients of segmentation models

Some of the true mask and predicted mask of segmentation models for test data are given in the figure 2.3.

Figure 2.3 True mask and predicted mask by segmentation models for test data

Part. 3

Conclusion

3.1 Conclusion

Images and its mask are used to train deep learning models to detect and classify defects in steel. From data analysis, we could observe that the given dataset is imbalanced and half of the given data set doesn't have the defects. Hence six model architecture is used to train and test on this dataset. One binary classifier to filter images with defects and a multi-label classifier to classify into defect classes 1,2,3,4. Image segmentation is used to predict encoded pixel values using the run length encoding technique. Using this architecture an accuracy of approximately 79 % and a dice coefficient of 67 % was achieved.

3.2 Future-work

A Model can be trained for a higher number of epochs to obtain better performance. Area threshold can be considered while predicting to overcome the issue of overlapping defects which can improve the efficiency of the system.

References

[1] Francois Chollet , Xception: Deep Learning with Depthwise Separable Convolutions , *arXiv:1610.02357v3 [cs.CV] 4 Apr 2017*

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox , U-Net: Convolutional Networks for Biomedical Image Segmentation , *arXiv:1505.04597v1 [cs.CV] 18 May 2015*

[3] Mingxing Tan , Quoc V. Le , EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, arXiv:1905.11946v5 [cs.LG] 11 Sep 2020

[4] https://segmentation-models.readthedocs.io/en/latest/tutorial.html

[5] https://keras.io/api/applications/

Part. A

Appendix

Abbreviation

- CNN – Convolutional Neural Network
- RLE – Run Length Encoding