Hypothesis supposed

- An anomaly is always bounded within the size of a predefined patch (64 x 64 x 3)
- Each image contains at most ONE anomaly
- All the images come from the same domain, which for this project corresponds to the domain of images depicting bark from different species of trees (for the first approach)

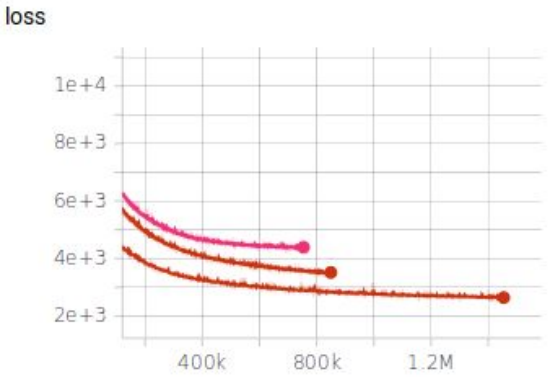
Annotations

- All the implementations are developed with TensorFlow v1.14.0
- If training in different days then each day the random seed changes: 200, 300, 400, etc
- Scikit-Image, MatPlotLib and Numpy are used for the processing, visualization, and manipulation of images
- The bark dataset used is called Bark-101 and can be found in: http://eidolon.univ-lyon2.fr/~remi1/Bark-101/
- Bark images are normalized to 256 x 256 x 3
- Patch size must be a multiple of 2 lower than 256
- **https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/r2/guide/distribute_strategy.ipynb#scrollTo=tuOe1ymfHZPu**
- Estimations done: when plotting ROC curves, then the sliding window stride of 16 has a random accuracy of 36 / 81 (40º slope), for a stride of 32 the random accuracy is 9 / 25 (32,4º slope) and for a stride of 64 the random accuracy is 4 / 9 (44,44º slope). Therefore the mean of the slopes is computed = 38,95º approximated to 39º which means a mean accuracy of 0,4163

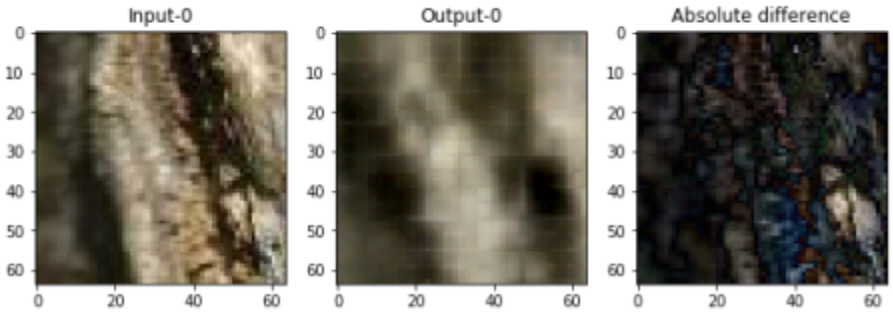
Random accuracies: 128_64=38,40% 128_32=42,64% 64_32=8,14% 64_16=9,68%
32_16=1,86% 32_8=1,90%

First approach: Convolutional Autoencoder

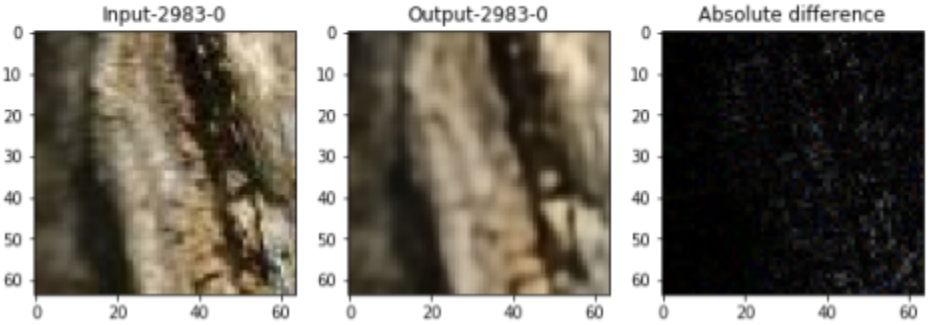
Patch size: 64
Patches per image: 16
Dataset size: 59.664 (3.729 images)
Train size: 47.728 (~80%) (2.983 images)
Test size: 11.936 (~20%) (746 images)



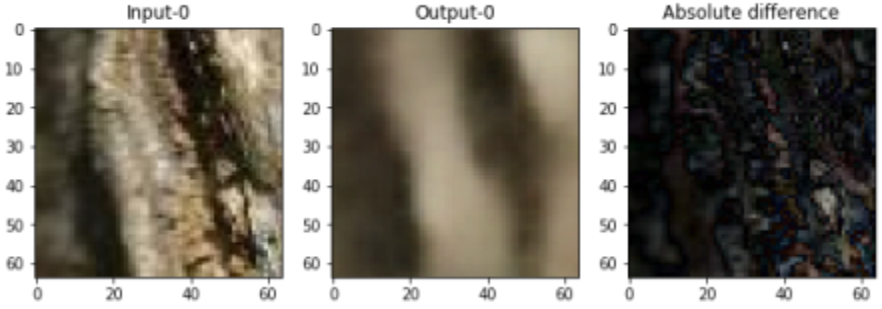
Embedding size	Training time	Test loss	Data Augmentation
16	10h	17.527	soft
32	4h	14.135	soft
64	4h	12.414	None
1024	26h	4.395 (3.470?)	soft
2048	26h	3.498	soft
4096	60h	2.643	hard



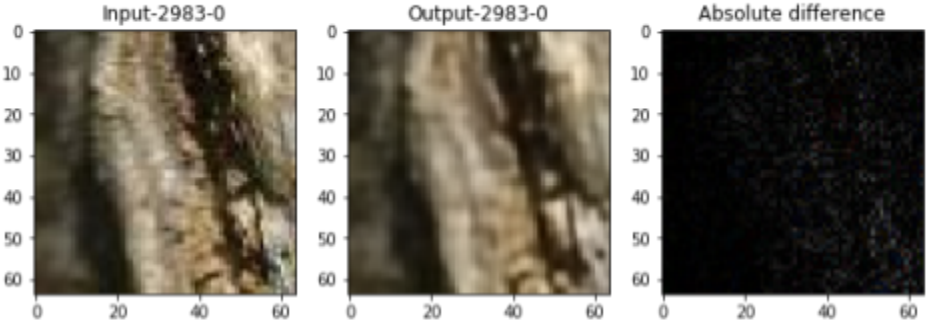
16 embedding size Test



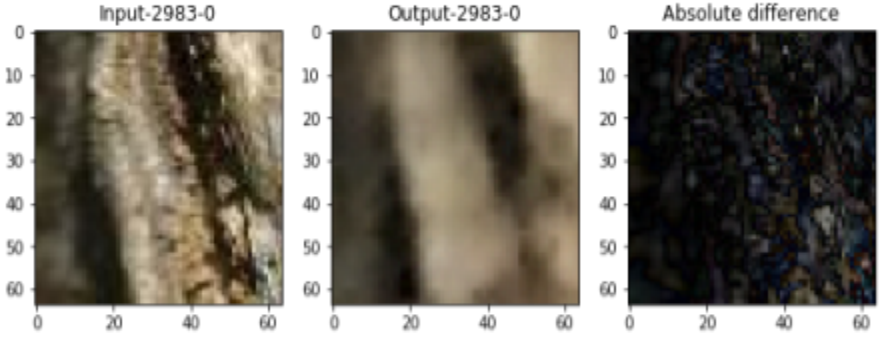
1024 embedding size Test



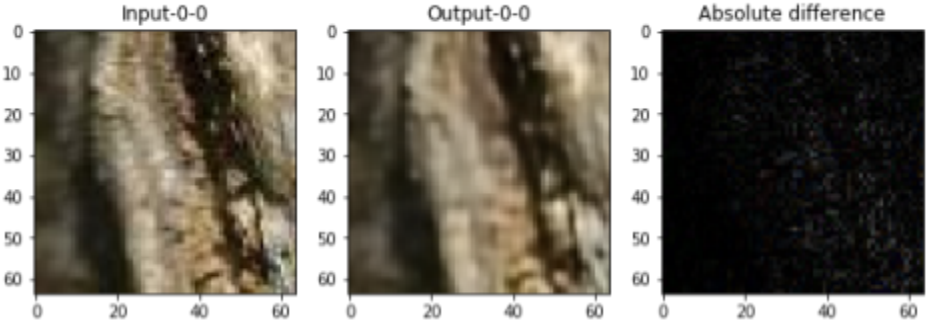
32 embedding size Test



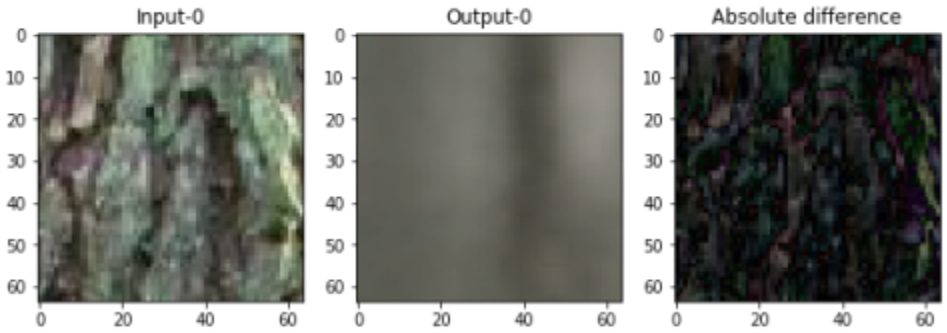
2048 embedding size Test



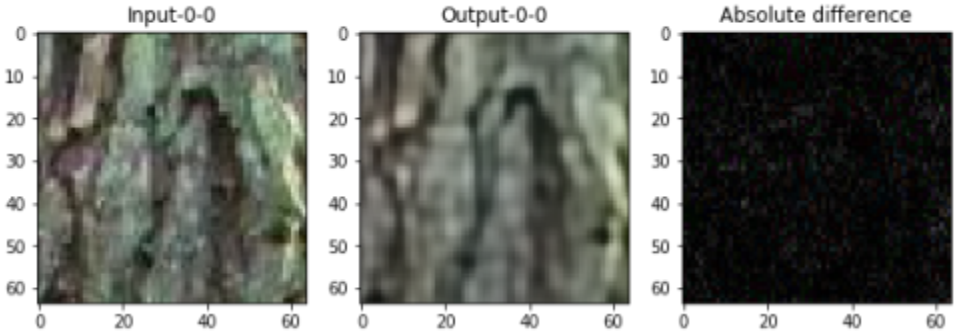
64 embedding size Test



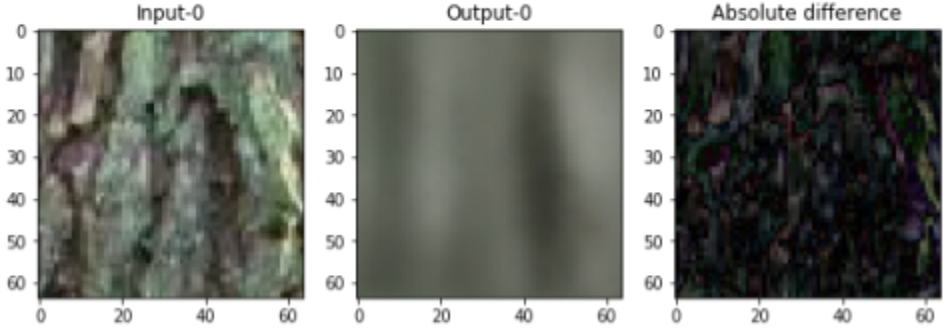
4096 embedding size Test



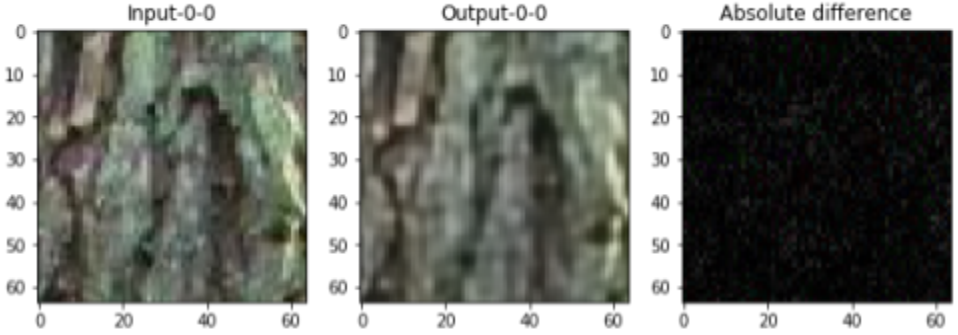
16 embedding size Train



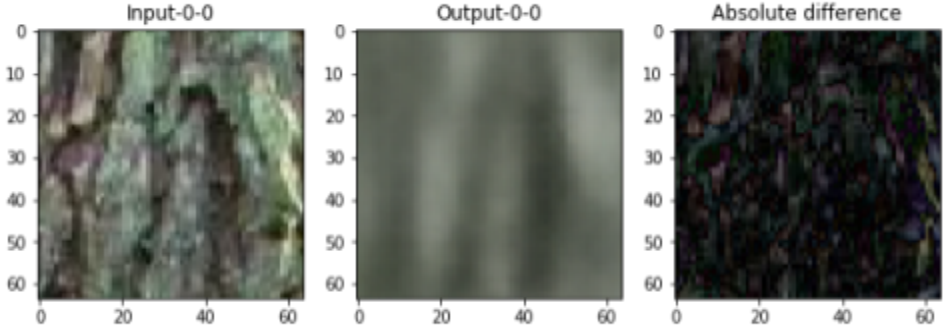
1024 embedding size Train



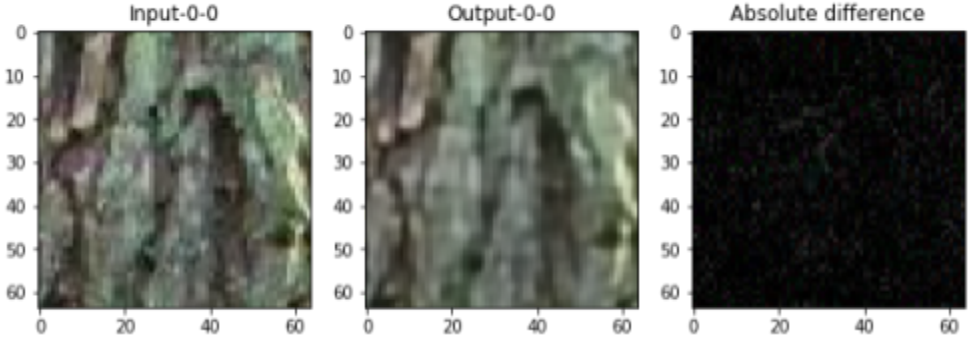
32 embedding size Train



2048 embedding size Train



64 embedding size Train



4096 embedding size Train

Bark-Colours-Quantitative		
Embedding size - Window stride	**Top 1 score All**	**Top 50 score All**
16 - 8	32,40%	68,40%
32 - 8	40,80%	74,40%
64 - 8	46,80%	76,20%
1024 - 8	49,40%	78,00%
2048 - 8	54,00%	79,80%
4096 - 8	57,40%	81,20%

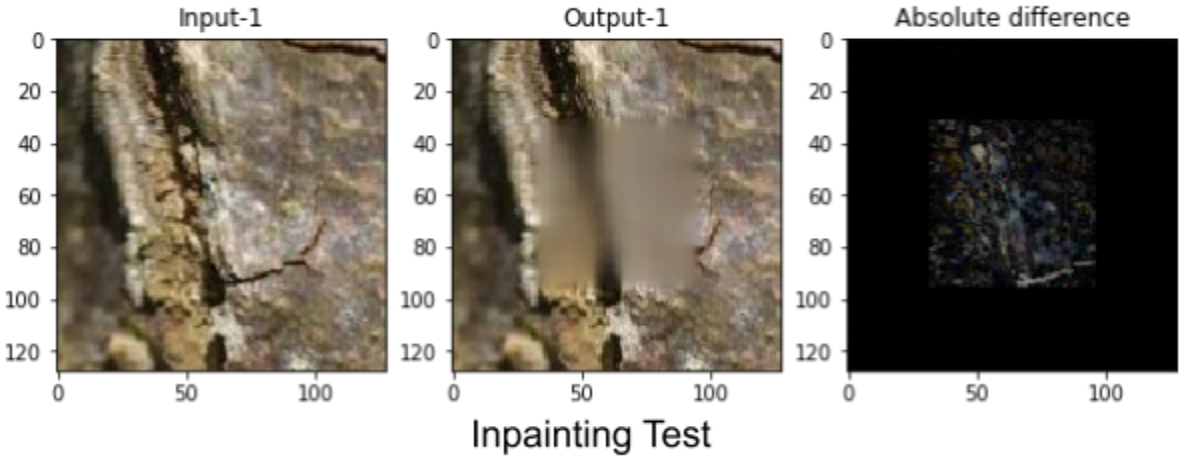
Bark-External-Quantitative		
Embedding size - Window stride	**Top 1 score All**	**Top 50 score All**
16 - 8	21,80%	56,60%
32 - 8	20,80%	59,40%
64 - 8	19,40%	55,40%
1024 - 8	29,40%	64,0%
2048 - 8	33,0%	65,40%
4096 - 8	32,40%	67,60%

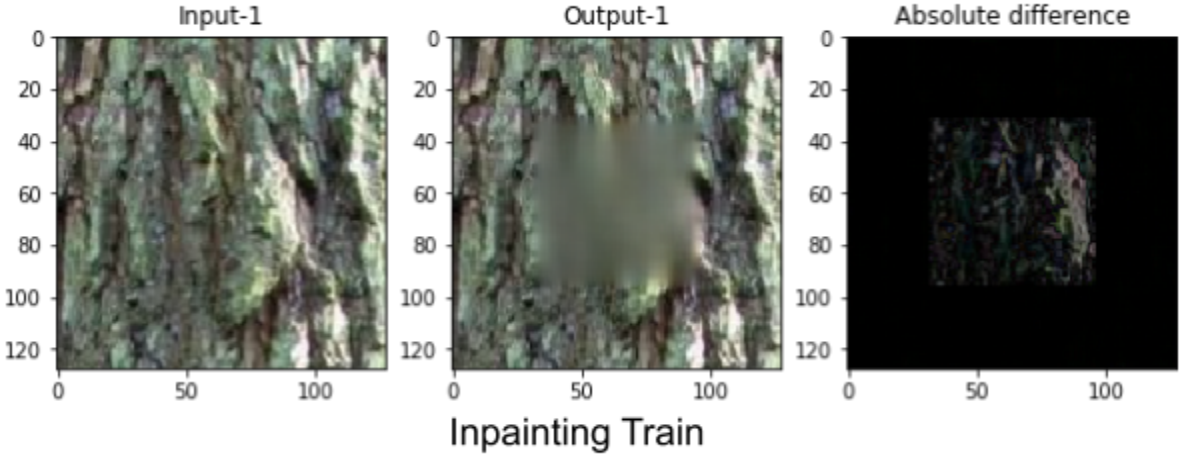
Second approach: Inpainting

Dataset size: 3.729 images
Train size: 2.983 images (~80%)
Test size: 746 images (~20%)
Dataset image shape = 256 x 256 x 3
CNN input image shape = 128 x 128 x 3

For training, each dataset image is cropped to 128x128x3 images using sliding window with a stride of ‘crop size’ (or a division of it)

Crop size	**Training time**	**Test loss**	**Data Augmentation**
32	60h	2.091	soft
64	21h	3.518	soft


Input-1
Output-1
Absolute difference
Inpainting Test


Input-1
Output-1
Absolute difference
Inpainting Train