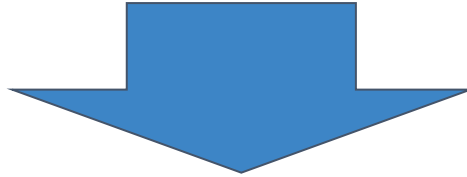


手法の概要

今回検出する欠陥の定義を次のようにした

1,白から灰色の画像内にある小さい黒色の塊

2,黒のノイズがかった画像内にある大きな黒色の塊

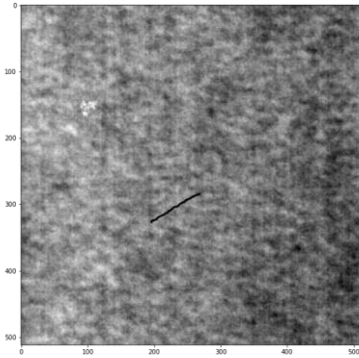


- そのため画像のテクスチャによって検出する欠陥の種類を分けている
- 画素が10以下であるピクセル数が20以下であるかどうか
- 1枚の画像内にはどちらかの欠陥しか存在していないとしている

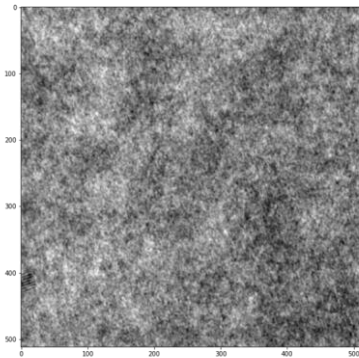
小さい欠陥検出

- 1.エッジ検出(Sobelフィルタ)
- 2.画像の値を0から1に正規化
- 3.0.4を閾値として二値化
- 4.それぞれの領域の面積を計算
- 5.いずれかの領域の面積が55以上のものがあれば欠陥

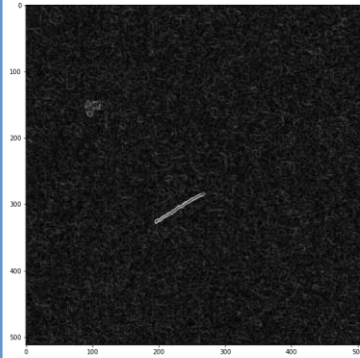
小さい欠陥検出



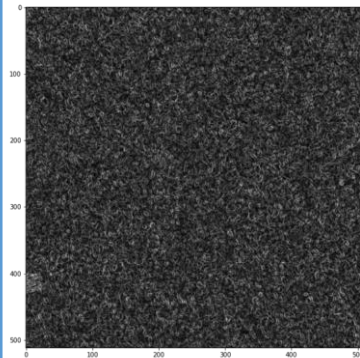
train_0001.png + 黒い直線



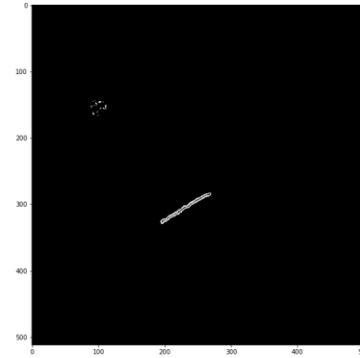
test_0000.png
入力画像



1 終了後



3 終了後



5 終了後

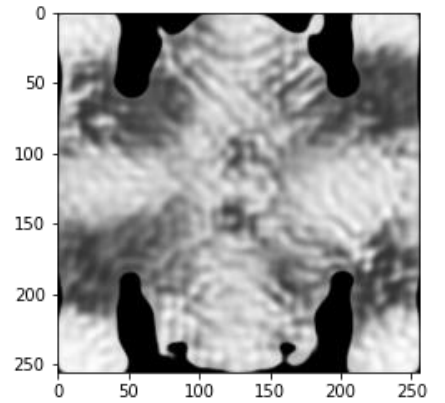
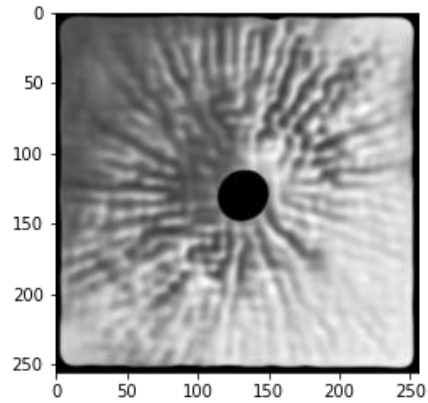
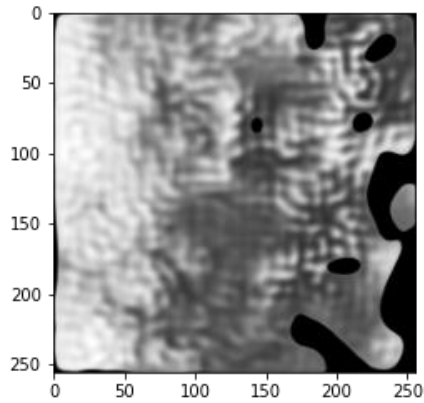
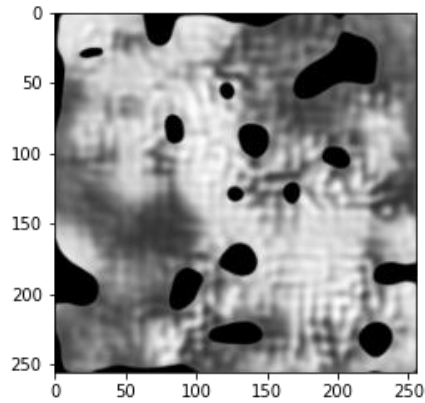
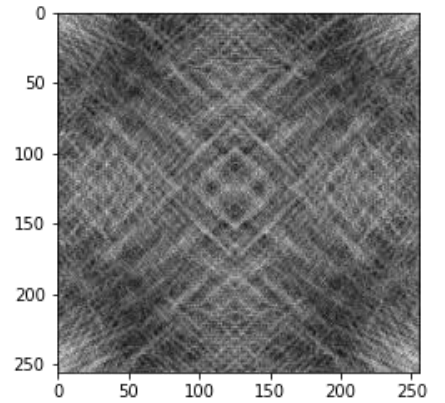
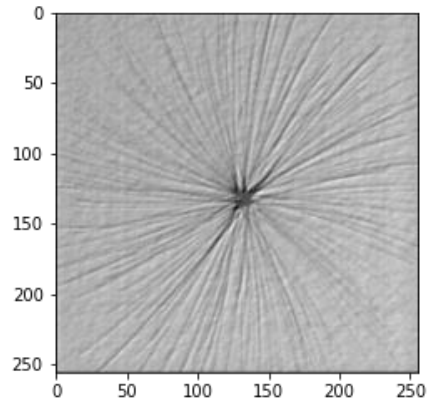
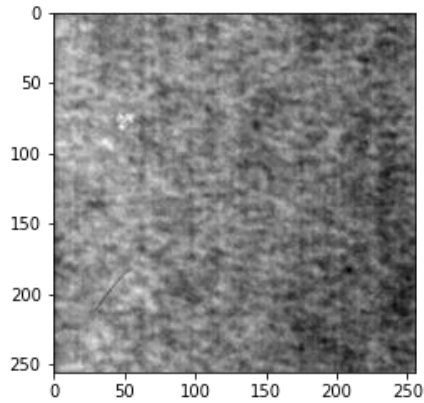
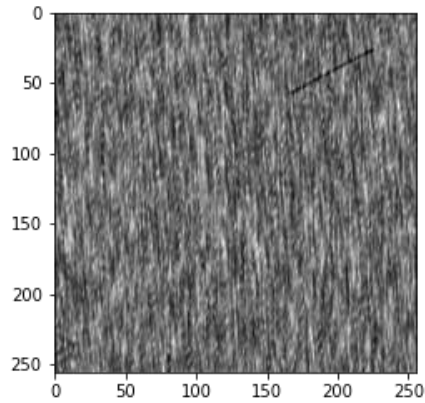
大きい欠陥検出

1. 画像のサイズを(256,256)へリサイズ
2. AutoEncoderによる画像の再現→再現画像
3. 入力画像を閾値10を基準として二値化(10以下を1)
4. 再現画像の閾値10を基準として二値化(10以下を1)
5. 入力画像と再現画像がともに1であるピクセル数の割合を計算
6. 割合が0.005以上であれば欠陥あり

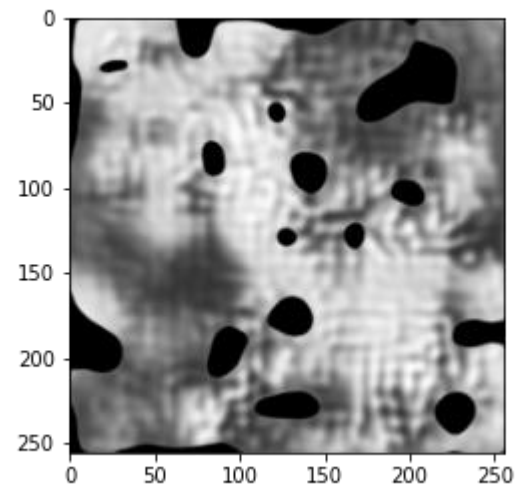
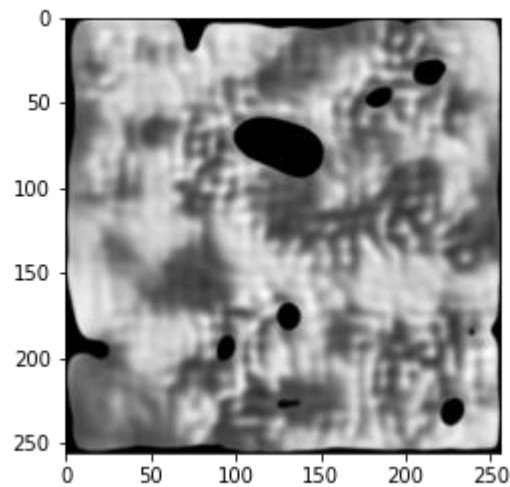
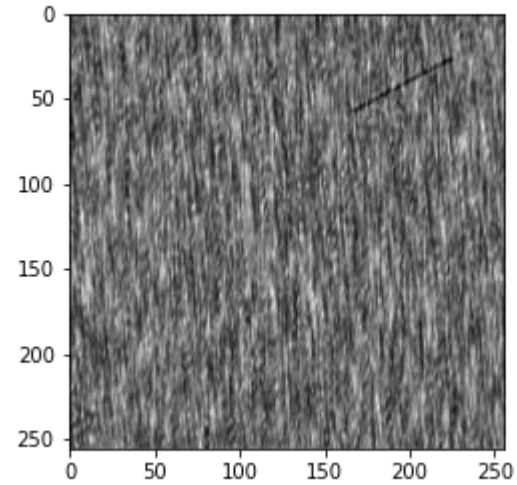
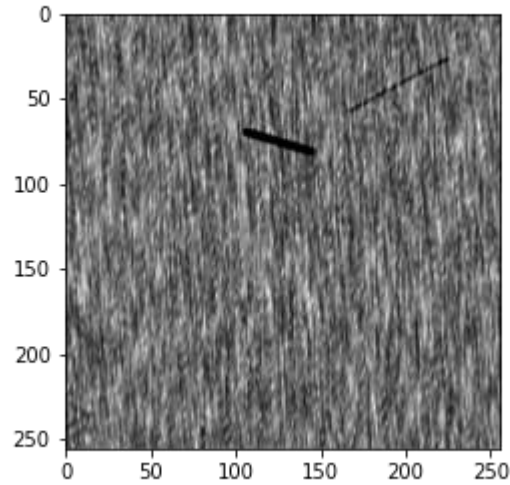
大きい傷検出用AutoEncoder概要

DeepLearningフレームワーク	PyTorch
学習率	0.005
最適化手法	adam
学習エポック	10
損失関数	ssim

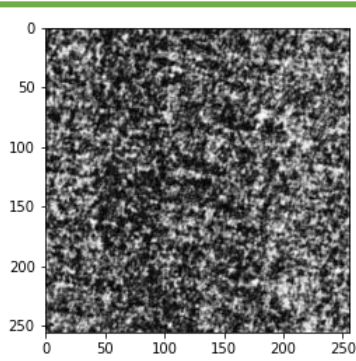
AutoEncoder學習結果



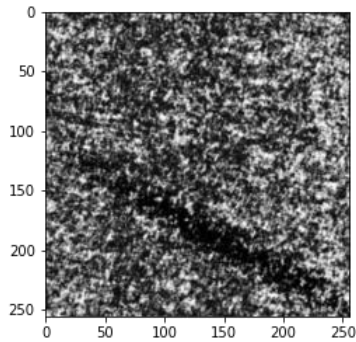
大きい傷検出用AutoEncoder概要



大きい欠陥検出

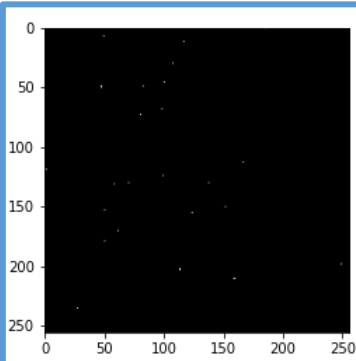


test_0036.png

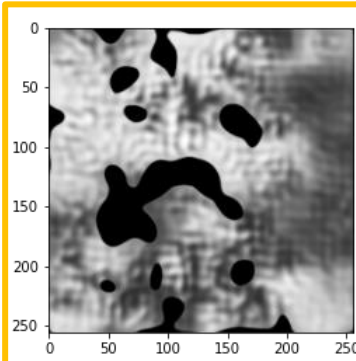


test_0035.png

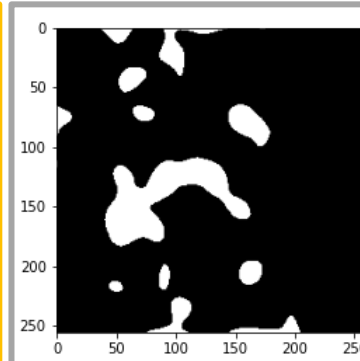
入力画像



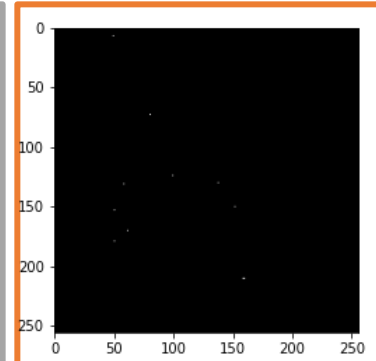
入力画像
二値化



再現画像



再現画像
二値化



入力画像、再現
画像がともに1
であるか

欠陥検出のフローチャート

