# Implementation

1. Pre-trained Resnet model extracts features
2. Trained autoencoder reconstructs original image using Resnet extracted features

**Only good samples are required to train the model!**

Industry-standard approach

Watch this for more details: Deep Learning for Computer Vision with Pytorch: Complete Project for Beginners - YouTube

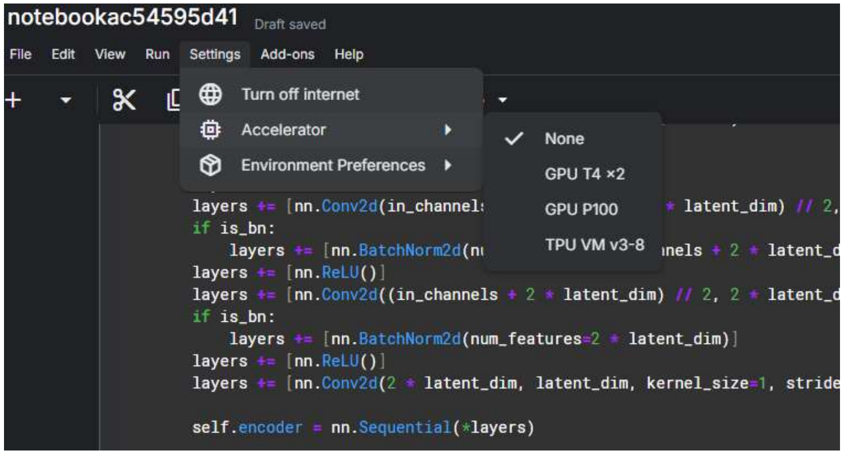
Image capture and inference on HP 840 G7 laptop.
-No GPU
-1MP webcam (224x224 cropped region of interest)

# Training

Augmented limited sample size of good caps with random rotation angles, lighting variations, positional variation.

Trained model on ~177 images using cloud-based Nvidia T4 GPU on Kaggle.com

~2mins training time



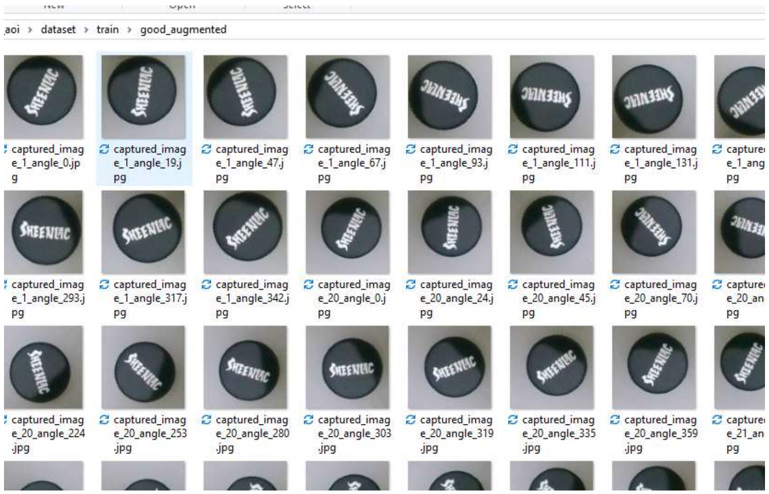
notebookac54595d41 Draft saved

Settings › Accelerator: ✓ None

```
layers += [nn.Conv2d(in_channel                    * latent_dim) // 2,
if is_bn:
    layers += [nn.BatchNorm2d(nu                    nels + 2 * latent_d
layers += [nn.ReLU()]
layers += [nn.Conv2d((in_channels + 2 * latent_dim) // 2, 2 * latent_d
if is_bn:
    layers += [nn.BatchNorm2d(num_features=2 * latent_dim)]
layers += [nn.ReLU()]
layers += [nn.Conv2d(2 * latent_dim, latent_dim, kernel_size=1, stride

self.encoder = nn.Sequential(*layers)
```

File
Edit
View
Run
Settings
Add-ons
Help
Turn off internet
Accelerator
Environment Preferences
None
GPU T4 ×2
GPU P100
TPU VM v3-8



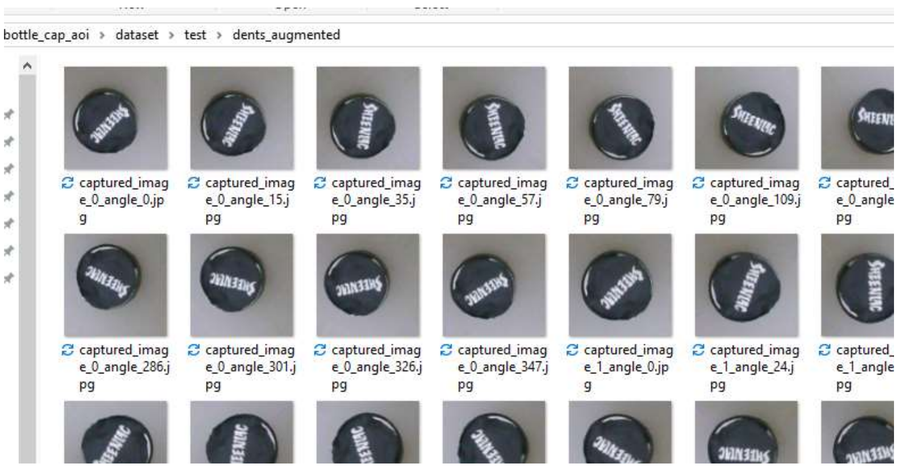
aoi › dataset › train › good_augmented

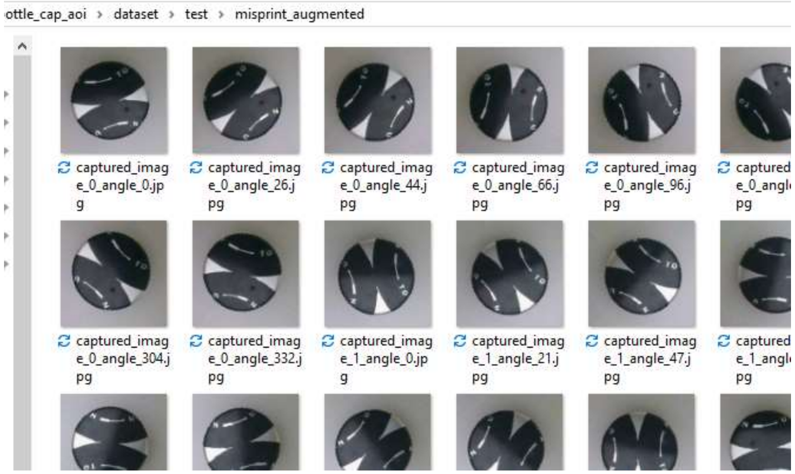
captured_image_1_angle_0.jpg
captured_image_1_angle_19.jpg
captured_image_1_angle_47.jpg
captured_image_1_angle_67.jpg
captured_image_1_angle_93.jpg
captured_image_1_angle_111.jpg
captured_image_1_angle_131.jpg
capture e_1_ang pg
captured_image_1_angle_293.jpg
captured_image_1_angle_317.jpg
captured_image_1_angle_342.jpg
captured_image_20_angle_0.jpg
captured_image_20_angle_24.jpg
captured_image_20_angle_45.jpg
captured_image_20_angle_70.jpg
capture e_20_an pg
captured_image_20_angle_224.jpg
captured_image_20_angle_253.jpg
captured_image_20_angle_280.jpg
captured_image_20_angle_303.jpg
captured_image_20_angle_319.jpg
captured_image_20_angle_335.jpg
captured_image_20_angle_359.jpg
capture e_21_an pg

# Testing

Augmented the limited sample set of caps with random rotations, lighting variations, position/zoom within the frame.

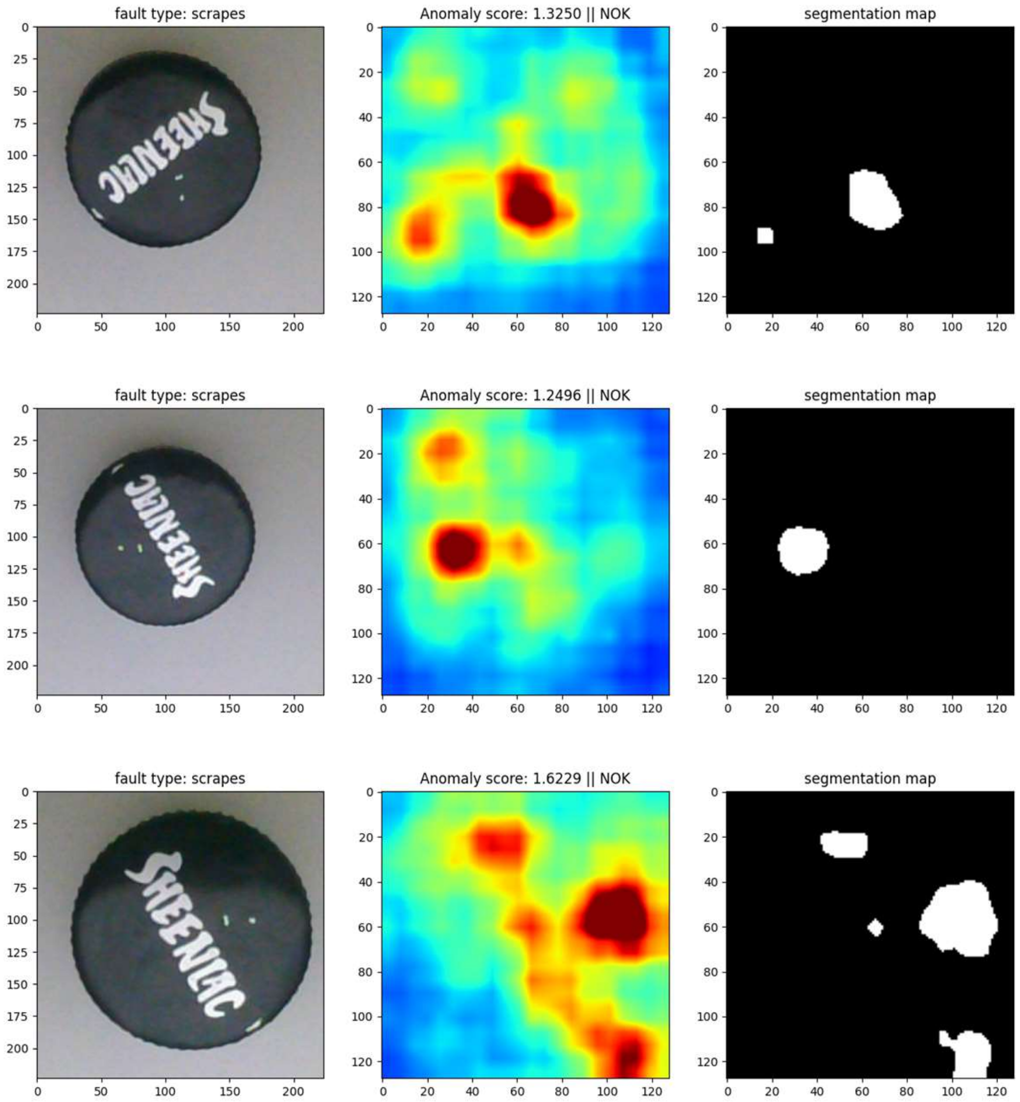
Ran model inference on both original and augmented images to determine pass/fail thresholds and performance


bottle_cap_aoi › dataset › test › dents_augmented
captured_image_0_angle_0.jpg
captured_image_0_angle_15.jpg
captured_image_0_angle_35.jpg
captured_image_0_angle_57.jpg
captured_image_0_angle_79.jpg
captured_image_0_angle_109.jpg
captured_e_0_angle pg
captured_image_0_angle_286.jpg
captured_image_0_angle_301.jpg
captured_image_0_angle_326.jpg
captured_image_0_angle_347.jpg
captured_image_1_angle_0.jpg
captured_image_1_angle_24.jpg
captured_e_1_angle pg


ottle_cap_aoi › dataset › test › misprint_augmented
captured_image_0_angle_0.jpg
captured_image_0_angle_26.jpg
captured_image_0_angle_44.jpg
captured_image_0_angle_66.jpg
captured_image_0_angle_96.jpg
captured e_0_angl pg
captured_image_0_angle_304.jpg
captured_image_0_angle_332.jpg
captured_image_1_angle_0.jpg
captured_image_1_angle_21.jpg
captured_image_1_angle_47.jpg
captured e_1_angl pg

# Scrapes

Excellent model performance under all rotation, zoom, and shadow/reflection conditions



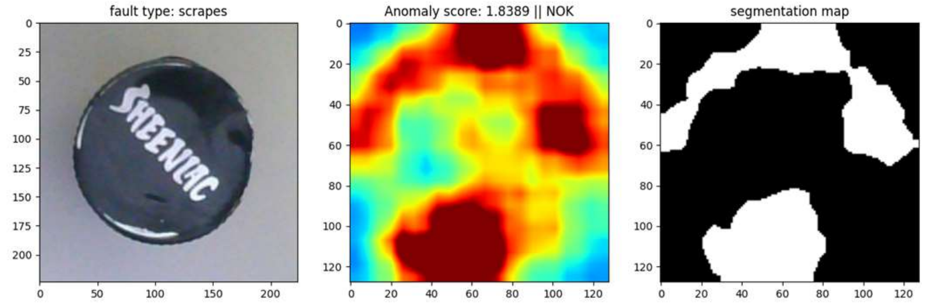
# Dents

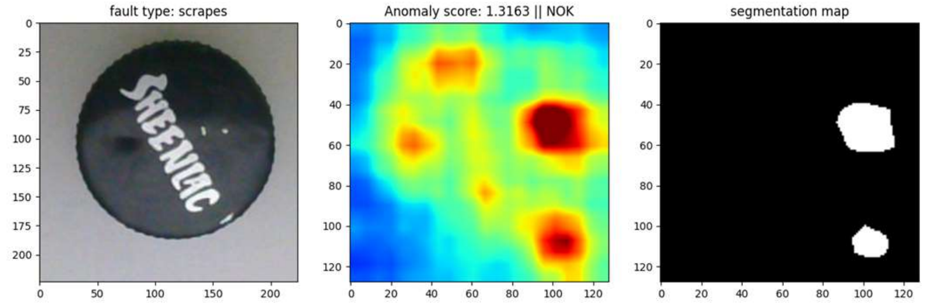
Major dents easily flagged as fail

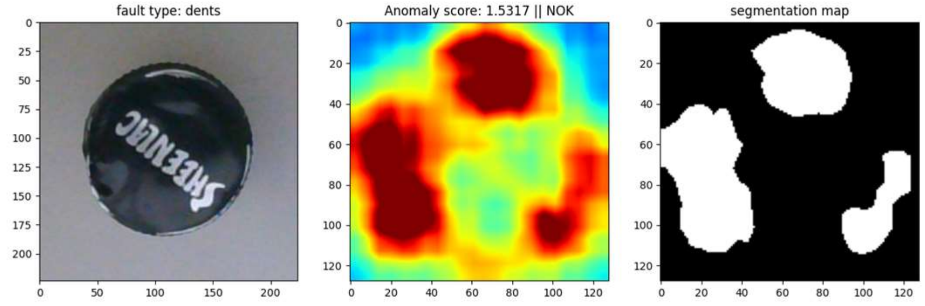
Minor dents are visible on heatmap but do not exceed the fail threshold (not visible on segmentation map)

Human visual identification of minor dents in caps is also difficult without ideal lighting

Experiment further with higher resolution camera if needed

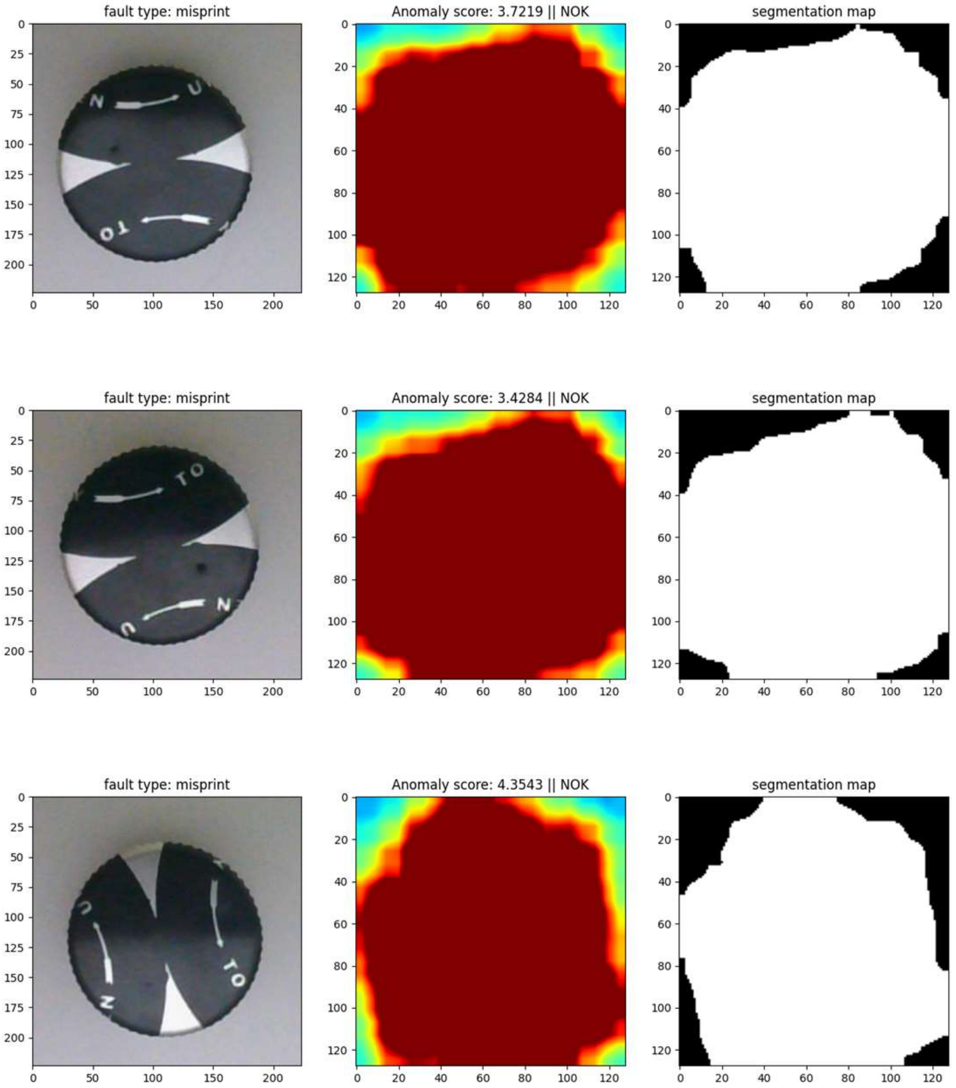






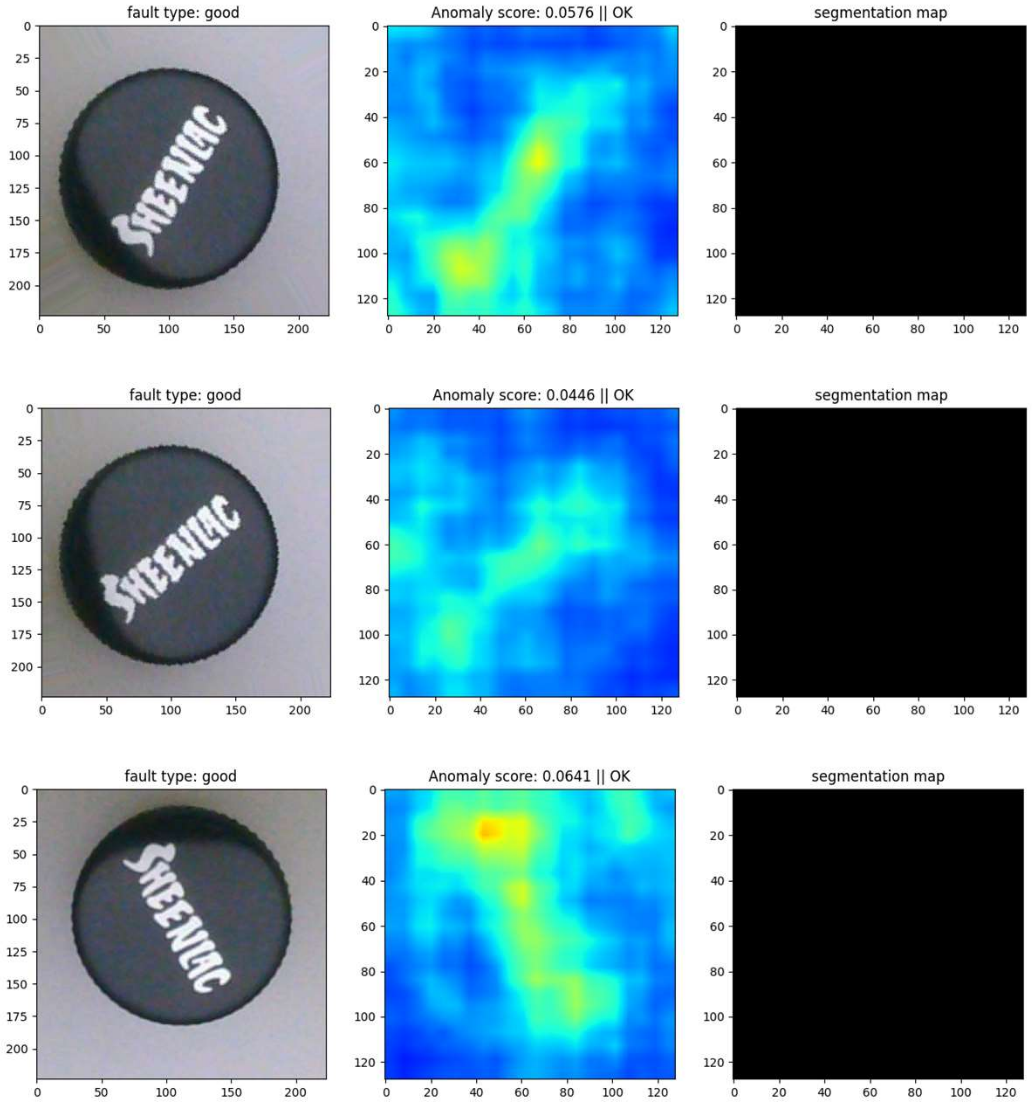
# Misprints

Trivial to identify.



# No Defect

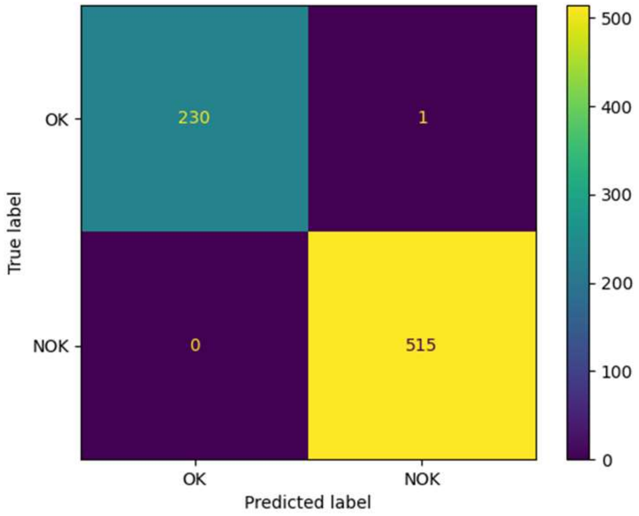
Good performance across many rotations and shading.

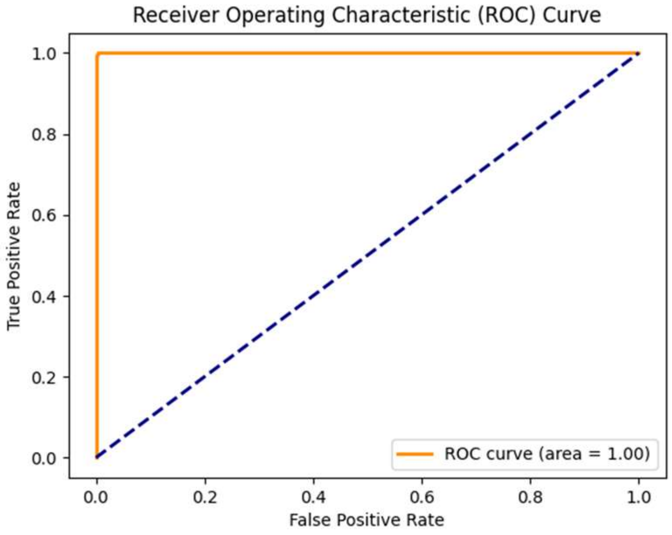


# Overall Performance

Excellent performance over ~700 augmented sample images with varying zoom, lighting, position, and rotation (1 incorrectly identified sample – ~0.13% failure rate)

OK = Pass
NOK = Fail


OK
NOK
True label
OK
NOK
Predicted label


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)