

AutoEncoder & 異常検知

教師あり学習

代表例：画像分類

(問題点)

バランスよく学習データを
集める必要がある

10例

1000例

通常：異常例は少なく，正常例は多い

アンバランスが一般的

教師なし学習

正常なデータのみから特徴抽出
そのからの逸脱を定量化

逸脱の大きい場合を

異常として検知

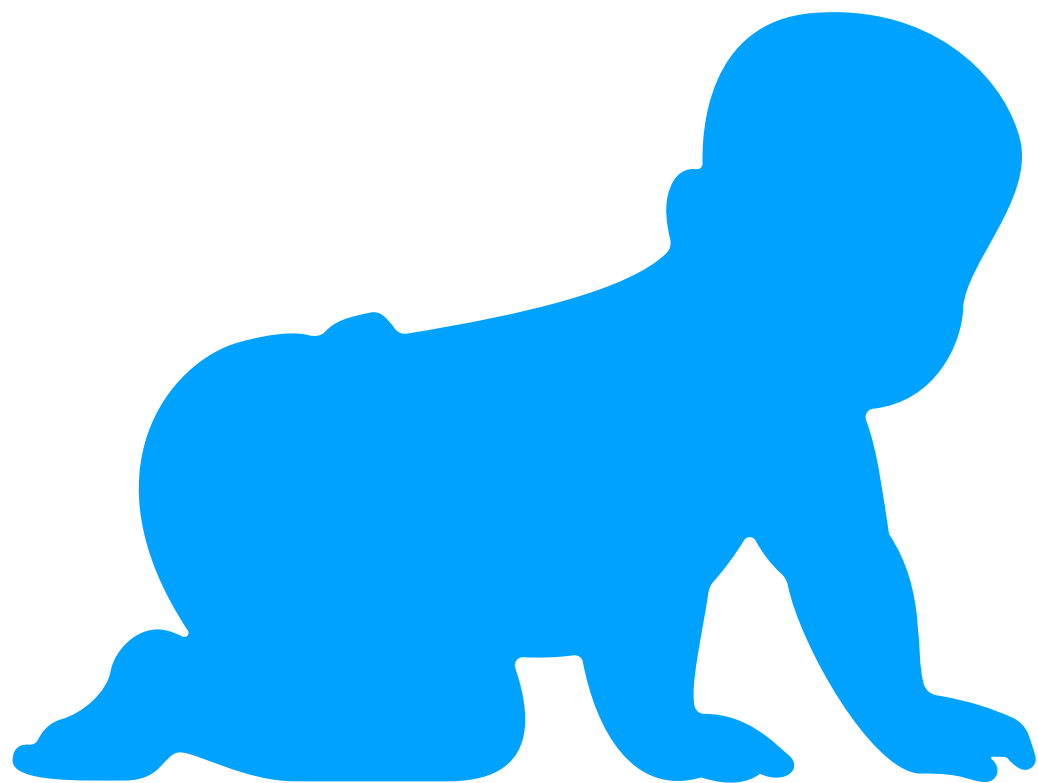
アンバランスなデータでも対応可能

教師なし学習

異常検知



いつも吉野家
たまに「すき家」



いつも吉野家
吉野家好き

吉野家が
入力

吉野家を
想像



たまねぎ

...

一致していると大満足！

吉野家が

$\rightarrow$
 a

吉野家を

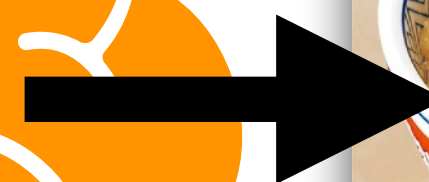
想像

x

入力

甘さ

$\hat{x}$



たまねぎ

$$E = \|x - \hat{x}\|$$

一致していると大満足！

Encode

画像をベクトルに変換して

画像よりも低い次元数

 $\vec{a}$ $=$ $f(x)$

画素数 = 次元数

画像をベクトルに符号化

Decode

変換したベクトルから
再び画像に戻す

ベクトルから復元した画像

 $\hat{x}$ $=$ $g(\vec{a})$

ベクトルを画像に復号化

その差がゼロに近ければ
ベクトルは画像を効率よく
表現している と言える

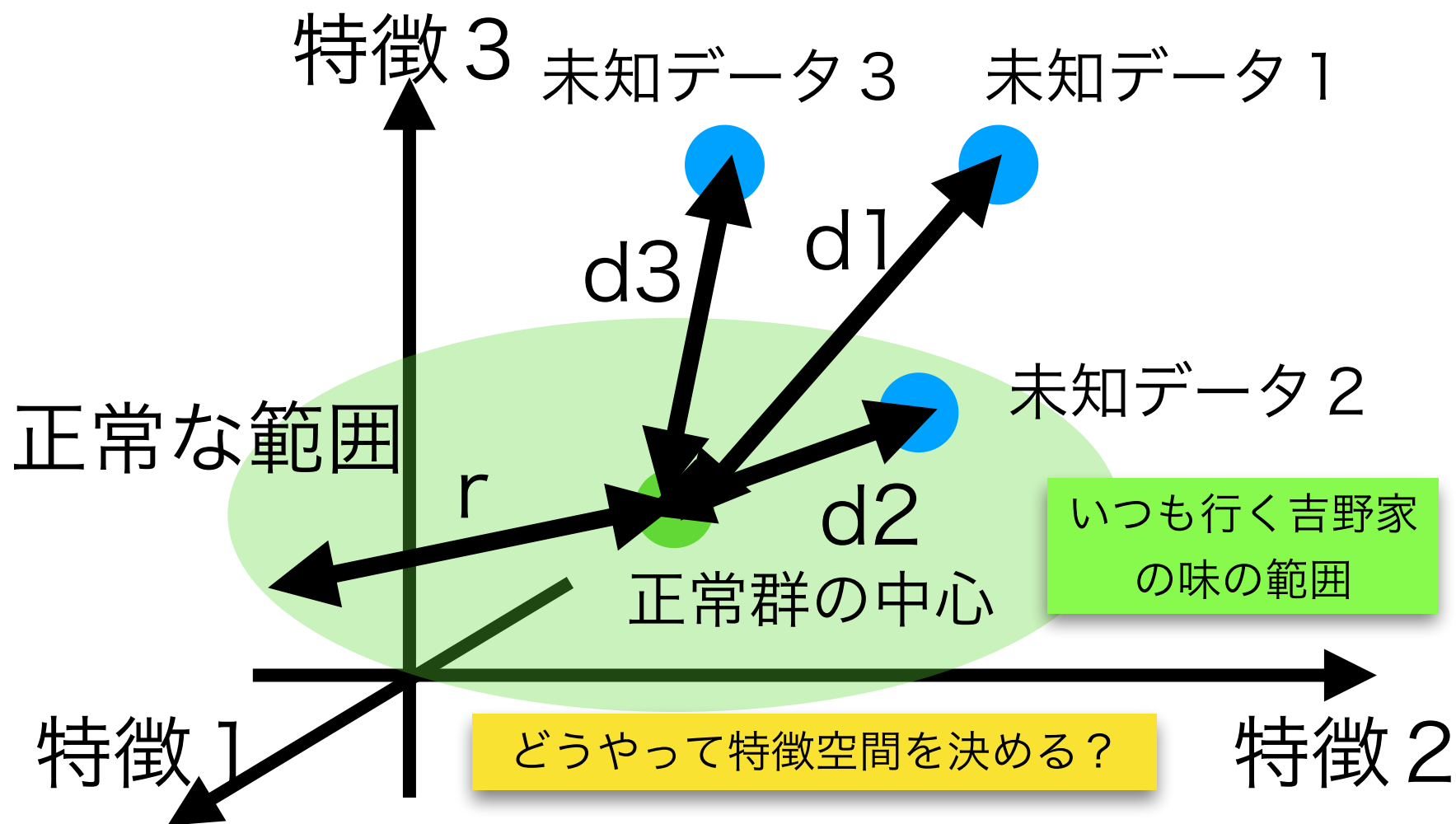
 E $=$ $\|x - \hat{x}\|$

元画像と復元画像の差

自己複製の自己 = Auto

異常検知の考え方

範囲が決まれば そこからの逸脱で判定



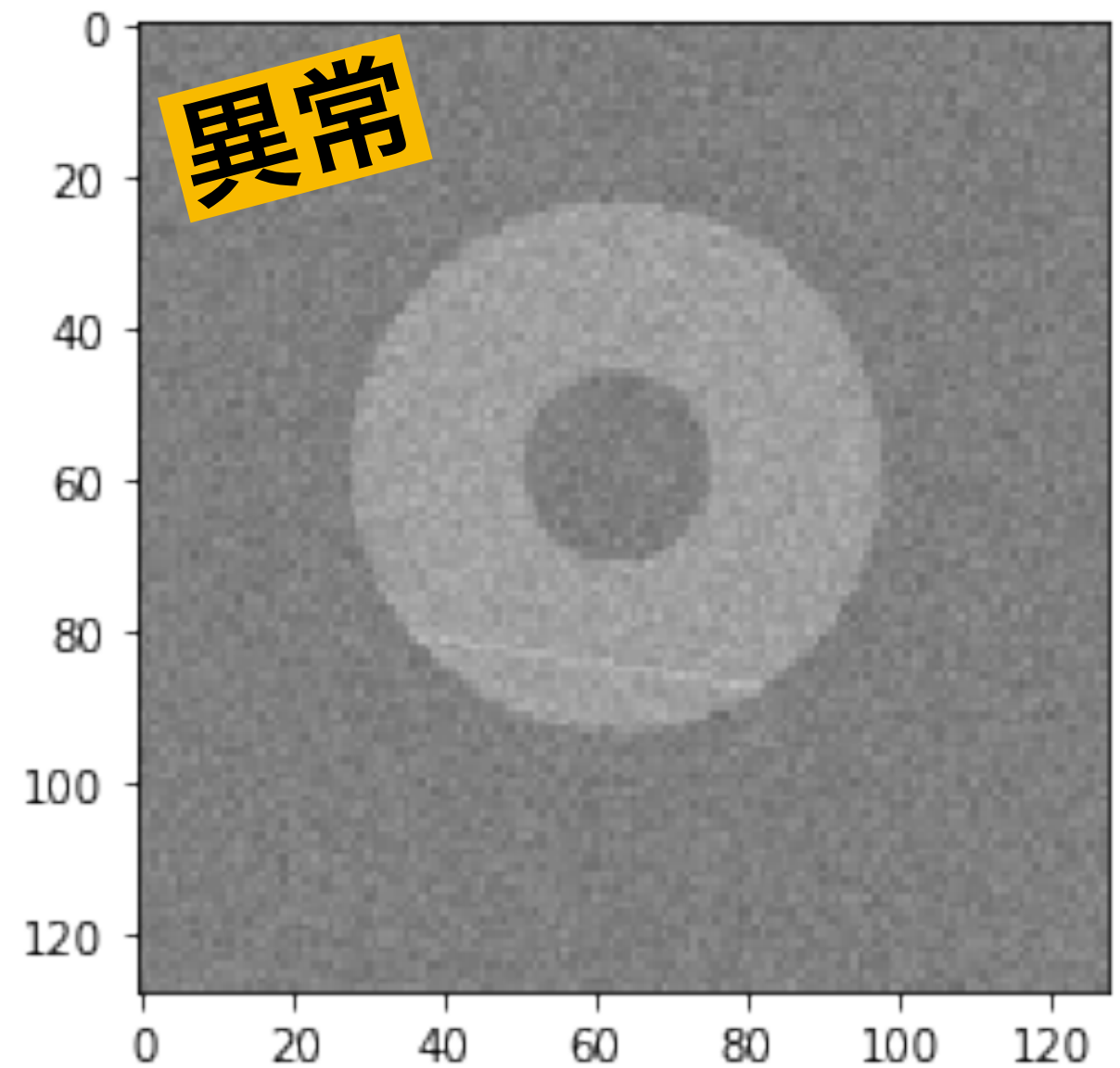
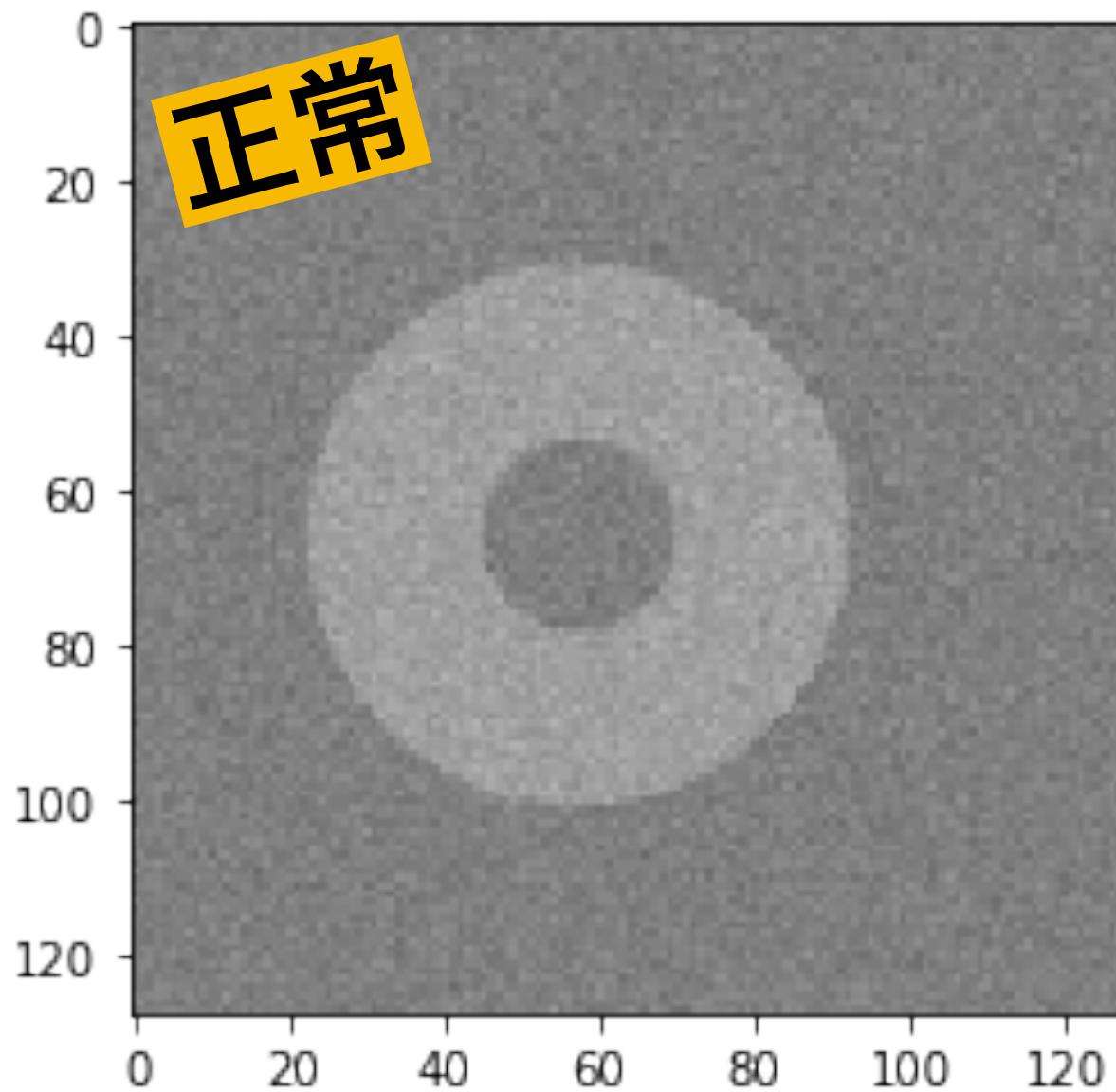
$r < d3 < d1$ なのでデータ 1 と 3 は異常である
 $d2 < r$ なのでデータ 2 は正常である

正常群から
中心と範囲を決める

未知データと
正常群の中心の
距離を計算する

その距離が正常な
範囲であるか
判定する

データだけ集めて手法開発



65536次元

画像を入力

256x256画素=65536

128x128x32

64x64x64

32x32x128

8x8x256

4096次元 4x256=4096

8x8x256

16x16x128

32x32x64

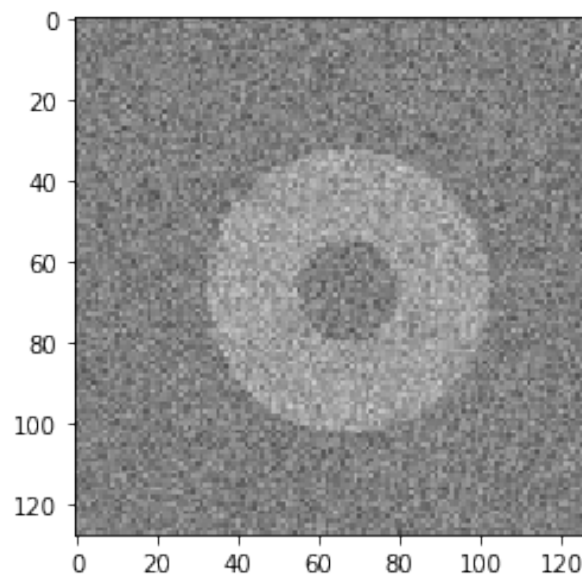
64x64x32

128x128x16

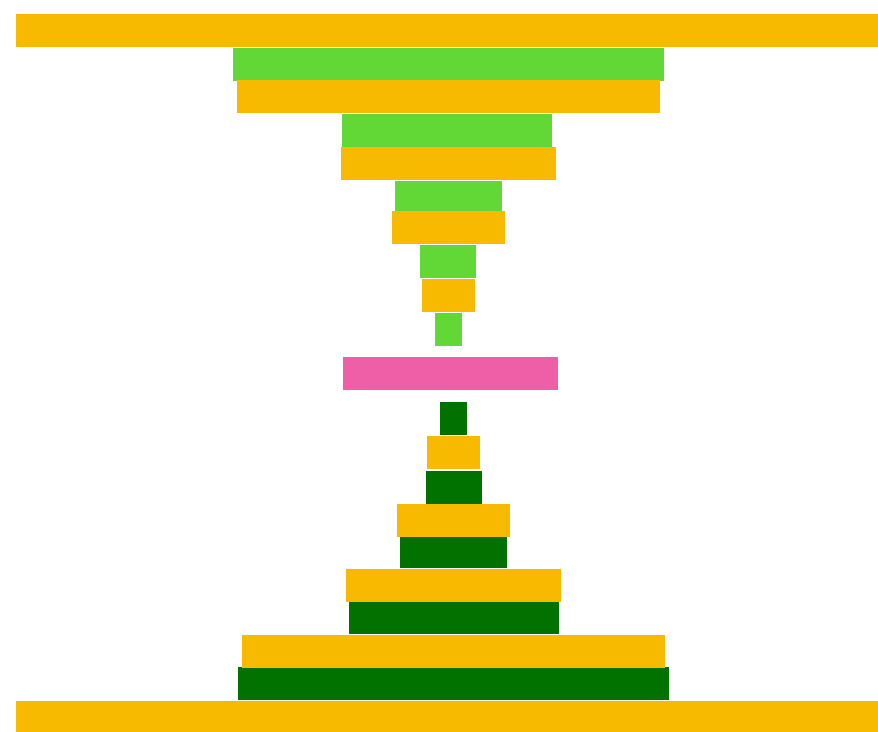
256x256x1

同じ画像を出力

(同じ=自己=Auto)



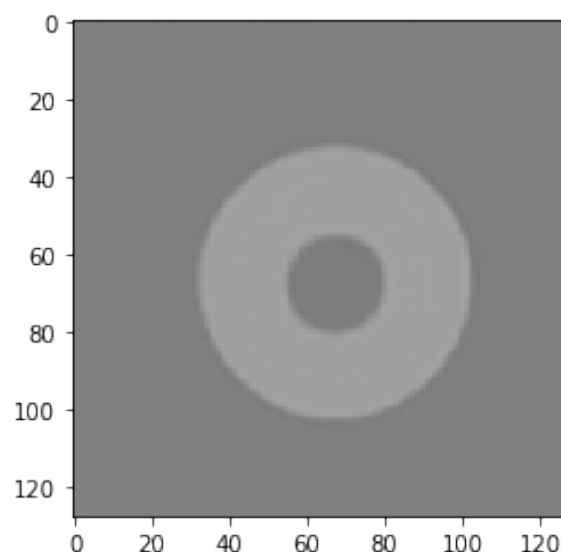
画素値



特徴



画素値



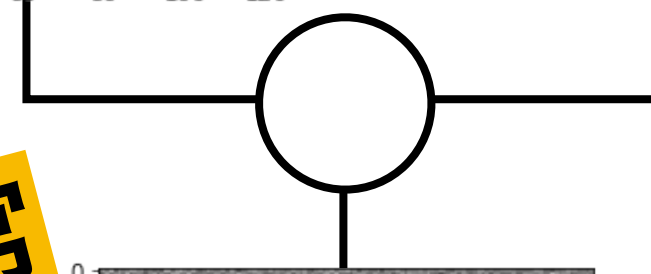
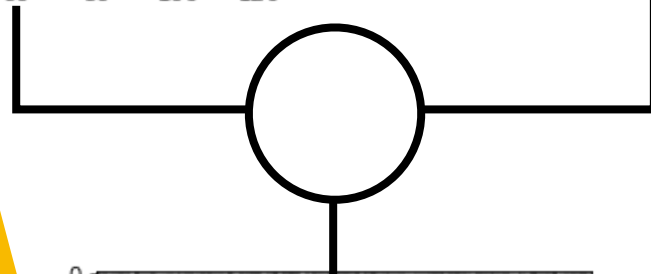
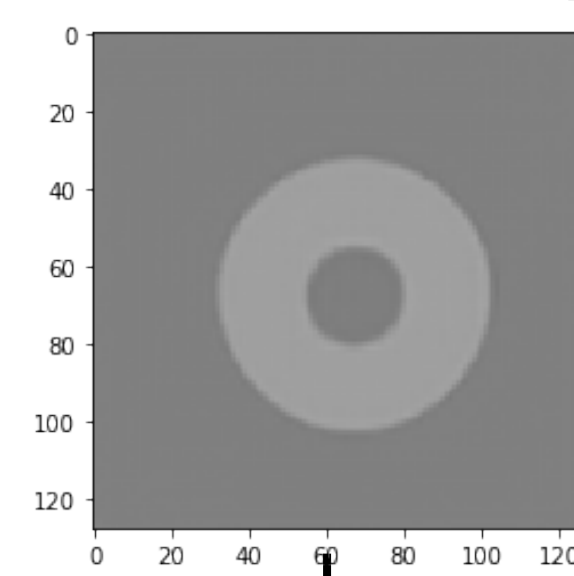
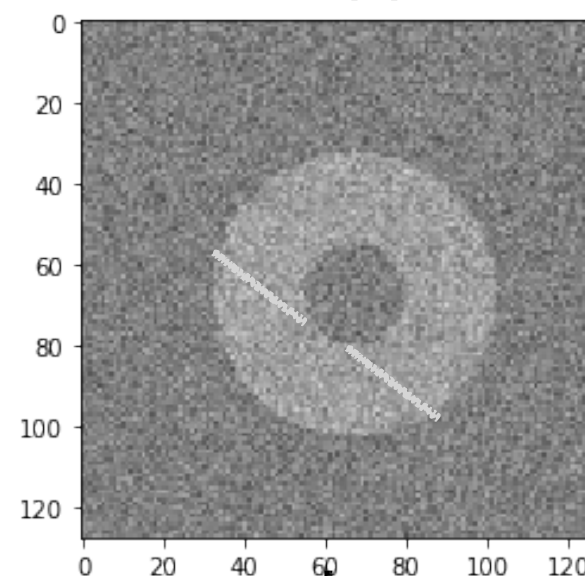
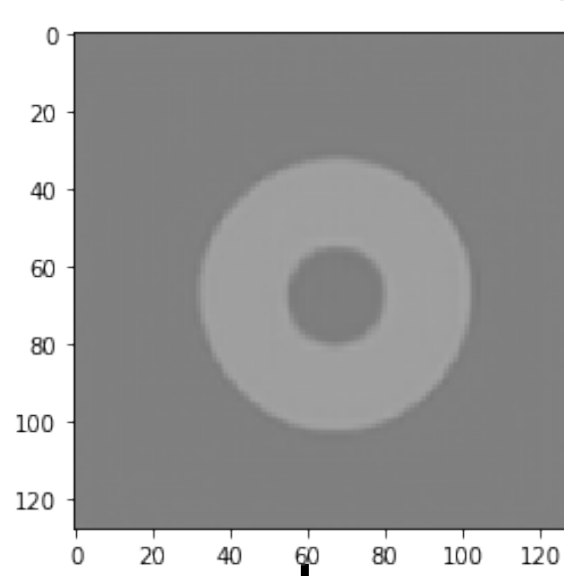
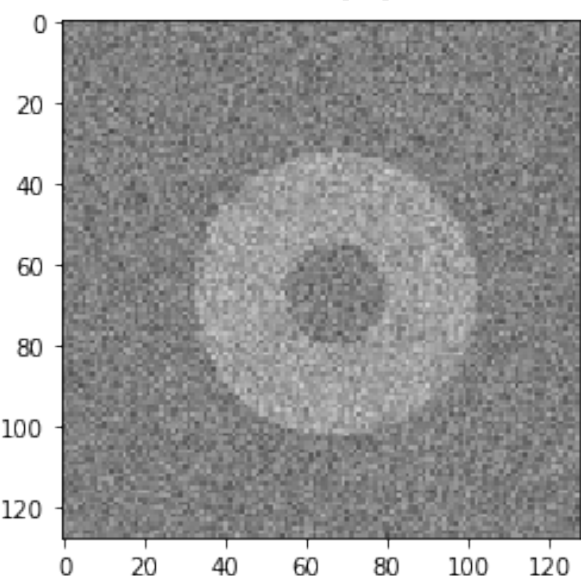
Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 256, 256, 16)	160
max_pooling2d_1 (MaxPooling2)	(None, 128, 128, 16)	0
conv2d_2 (Conv2D)	(None, 128, 128, 32)	4640
max_pooling2d_2 (MaxPooling2)	(None, 64, 64, 32)	0
conv2d_3 (Conv2D)	(None, 64, 64, 64)	18496
max_pooling2d_3 (MaxPooling2)	(None, 32, 32, 64)	0
conv2d_4 (Conv2D)	(None, 32, 32, 128)	73856
max_pooling2d_4 (MaxPooling2)	(None, 16, 16, 128)	0
conv2d_5 (Conv2D)	(None, 8, 8, 256)	295168
max_pooling2d_5 (MaxPooling2)	(None, 4, 4, 256)	0
flatten_1 (Flatten)	(None, 4096)	0
reshape_1 (Reshape)	(None, 4, 4, 256)	0
conv2d_6 (Conv2D)	(None, 4, 4, 256)	590080
up_sampling2d_1 (UpSampling2)	(None, 8, 8, 256)	0
conv2d_7 (Conv2D)	(None, 8, 8, 128)	295040
up_sampling2d_2 (UpSampling2)	(None, 16, 16, 128)	0
conv2d_8 (Conv2D)	(None, 16, 16, 64)	73792
up_sampling2d_3 (UpSampling2)	(None, 32, 32, 64)	0
conv2d_9 (Conv2D)	(None, 32, 32, 32)	18464
up_sampling2d_4 (UpSampling2)	(None, 64, 64, 32)	0
conv2d_10 (Conv2D)	(None, 64, 64, 16)	4624
up_sampling2d_5 (UpSampling2)	(None, 128, 128, 16)	0
up_sampling2d_6 (UpSampling2)	(None, 256, 256, 16)	0
conv2d_11 (Conv2D)	(None, 256, 256, 1)	145
Total params: 1,374,465		
Trainable params: 1,374,465		
Non-trainable params: 0		

入力画像

AutoEncoderの出力画像

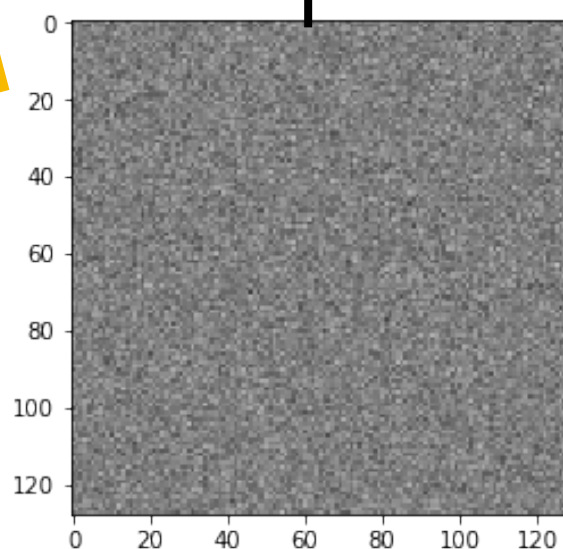
入力画像

AutoEncoderの出力画像



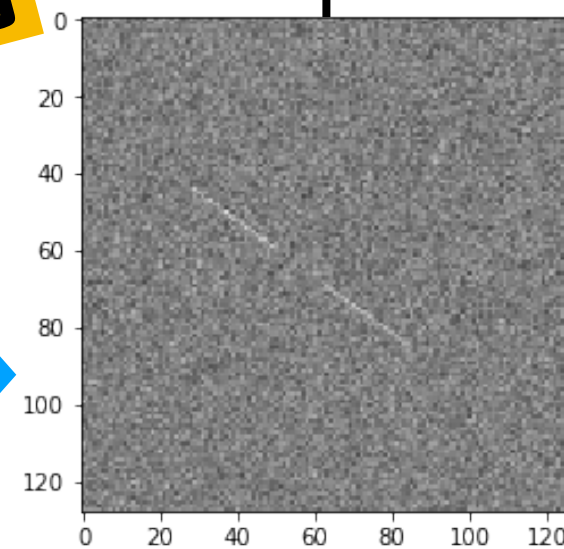
正常

異常



差分像

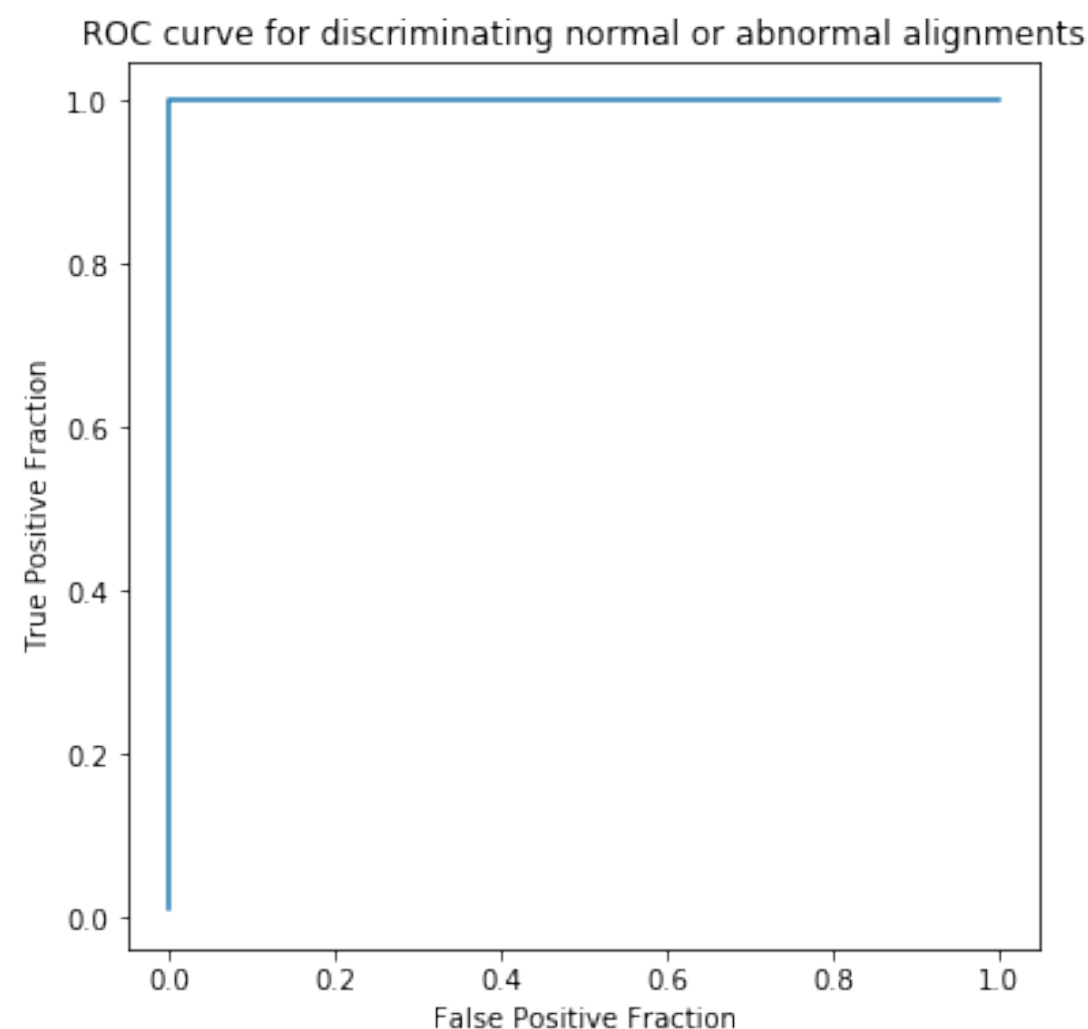
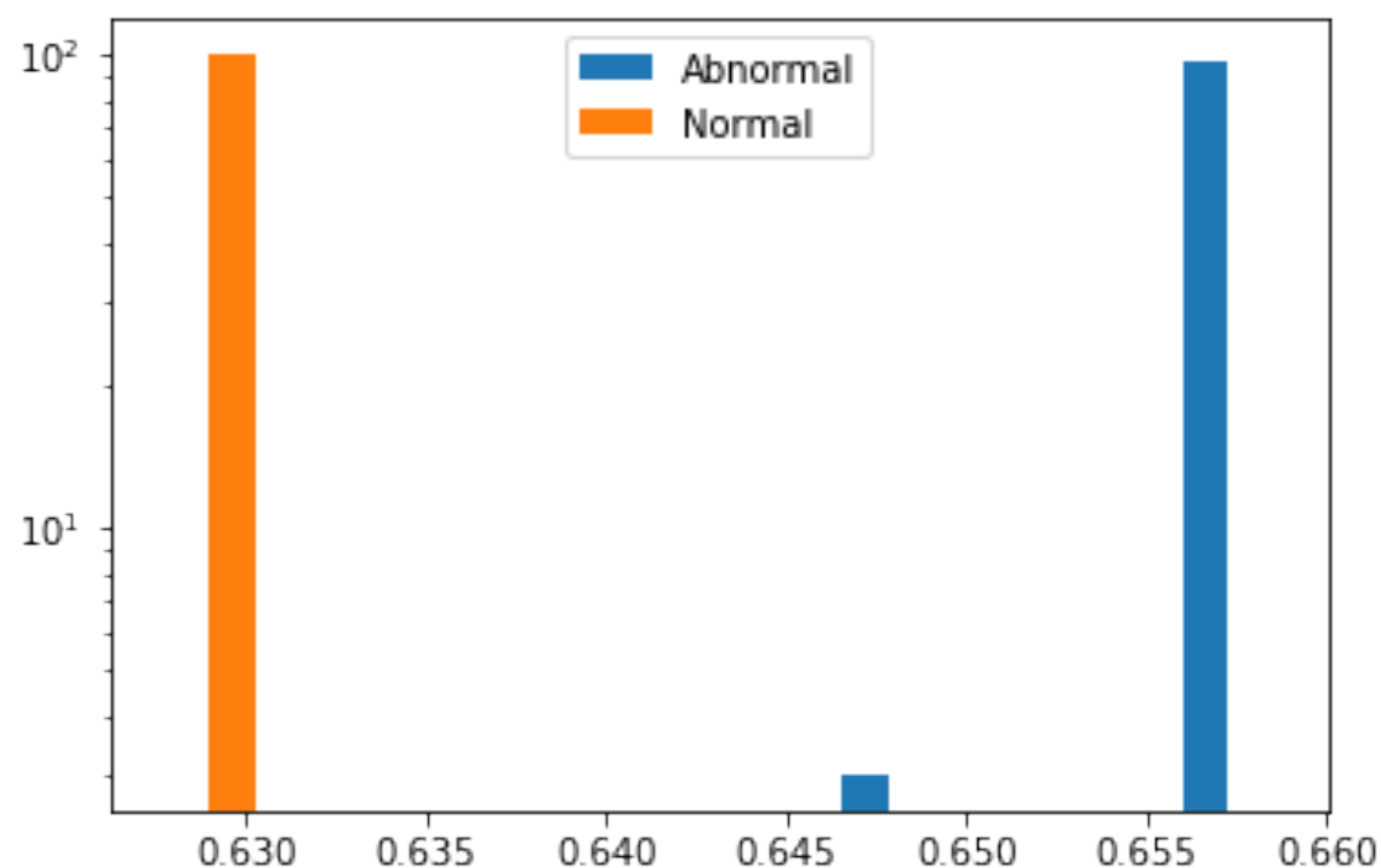
最大値に違いがあるかも？



差分像

極端な例ですが...

AutoEncoderを前処理としてうまく利用できれば
単純な特徴量（例えば最大値）でも分離可能



65536次元

画像を入力

256x256画素=65536

256x256x16

128x128x32

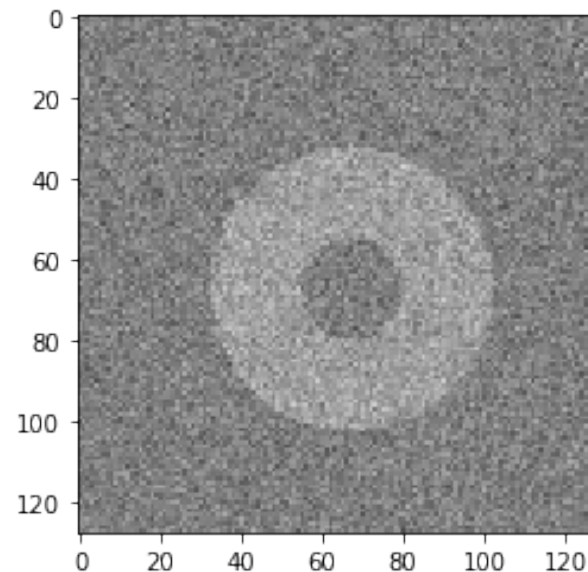
64x64x64

32x32x128

8x8x256

4096次元

4x4x256=4096



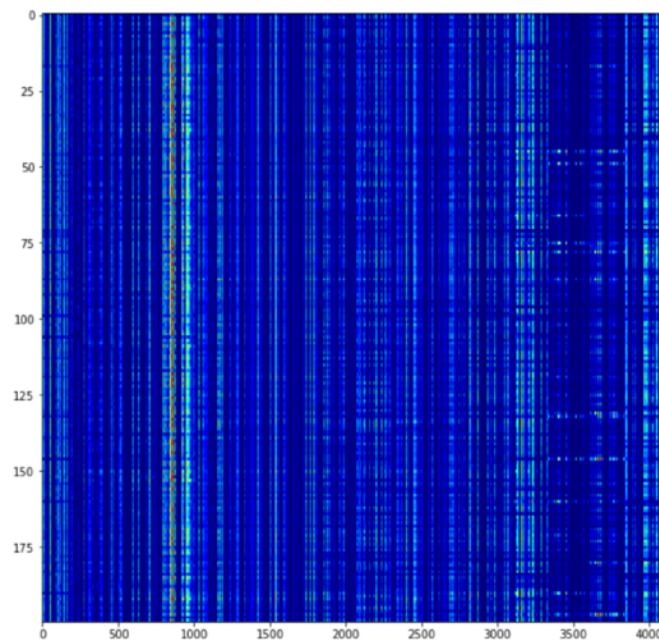
画素値



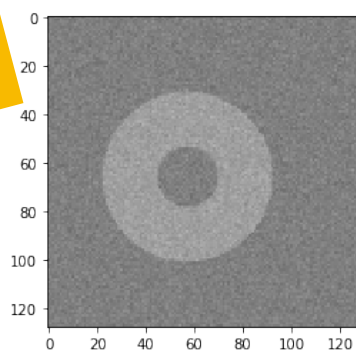
特徴

Layer (type)	Output Shape	Param #
conv2d_1_input (InputLayer)	(None, 256, 256, 1)	0
conv2d_1 (Conv2D)	(None, 256, 256, 16)	160
max_pooling2d_1 (MaxPooling2)	(None, 128, 128, 16)	0
conv2d_2 (Conv2D)	(None, 128, 128, 32)	4640
max_pooling2d_2 (MaxPooling2)	(None, 64, 64, 32)	0
conv2d_3 (Conv2D)	(None, 64, 64, 64)	18496
max_pooling2d_3 (MaxPooling2)	(None, 32, 32, 64)	0
conv2d_4 (Conv2D)	(None, 32, 32, 128)	73856
max_pooling2d_4 (MaxPooling2)	(None, 16, 16, 128)	0
conv2d_5 (Conv2D)	(None, 8, 8, 256)	295168
max_pooling2d_5 (MaxPooling2)	(None, 4, 4, 256)	0
flatten_1 (Flatten)	(None, 4096)	0
Total params: 392,320		
Trainable params: 392,320		
Non-trainable params: 0		

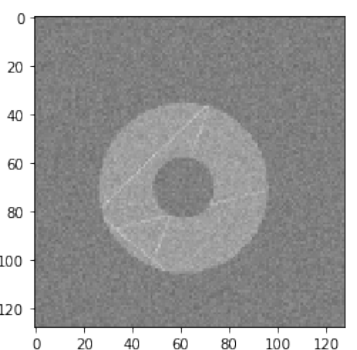
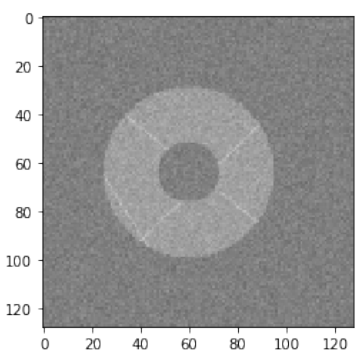
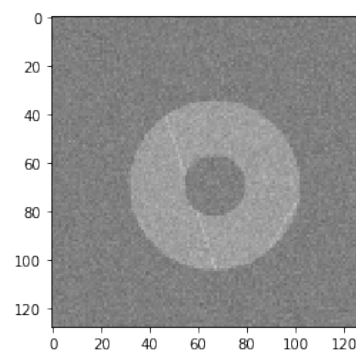
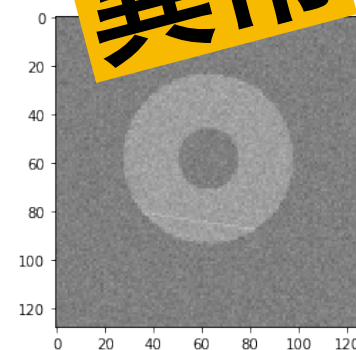
同じ画像を出力
(同じ=自己
=Auto)



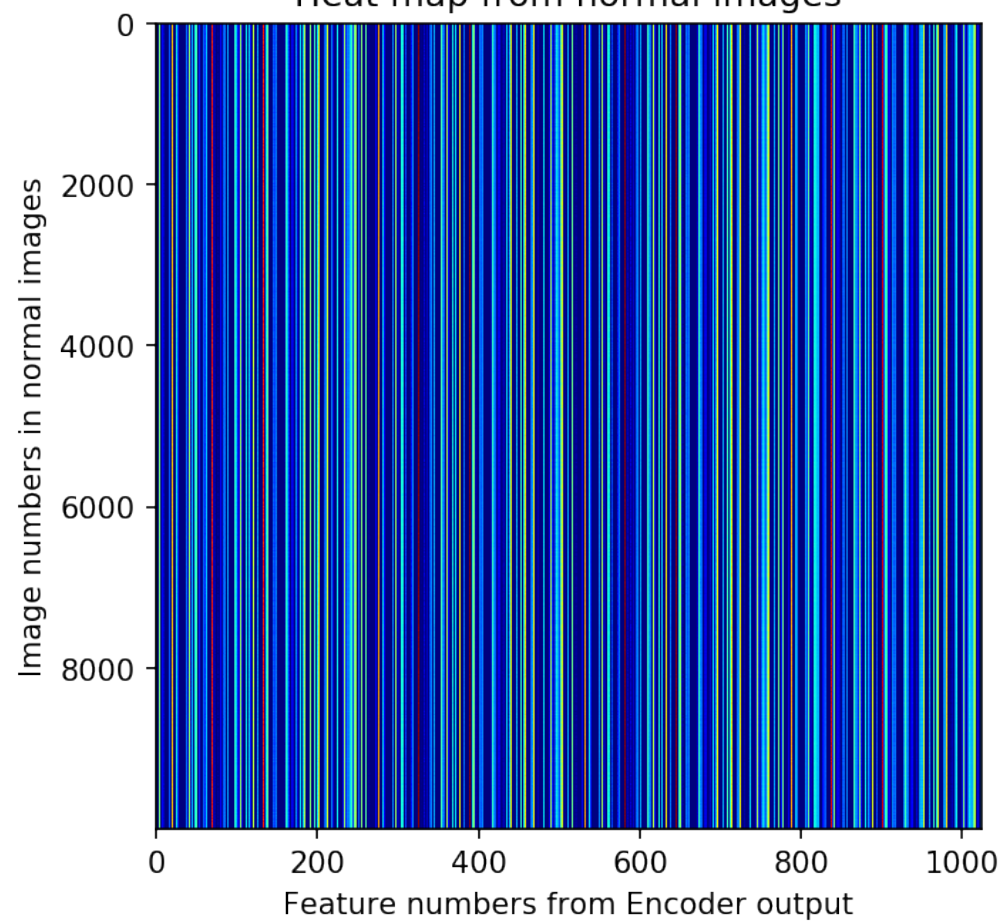
正常



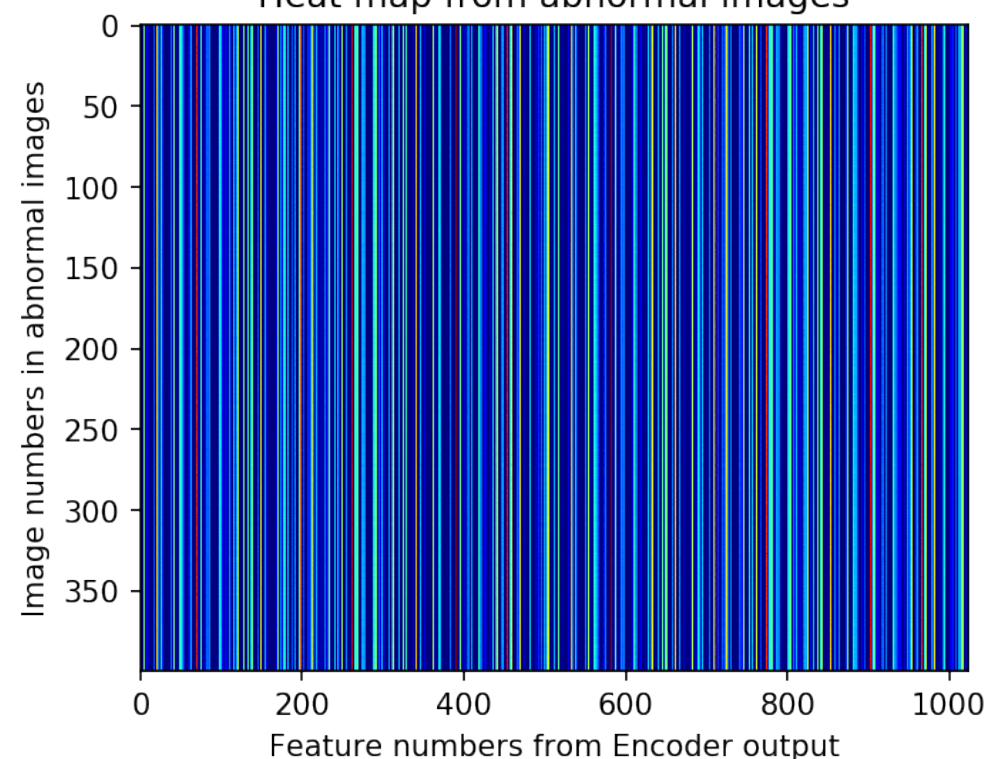
異常



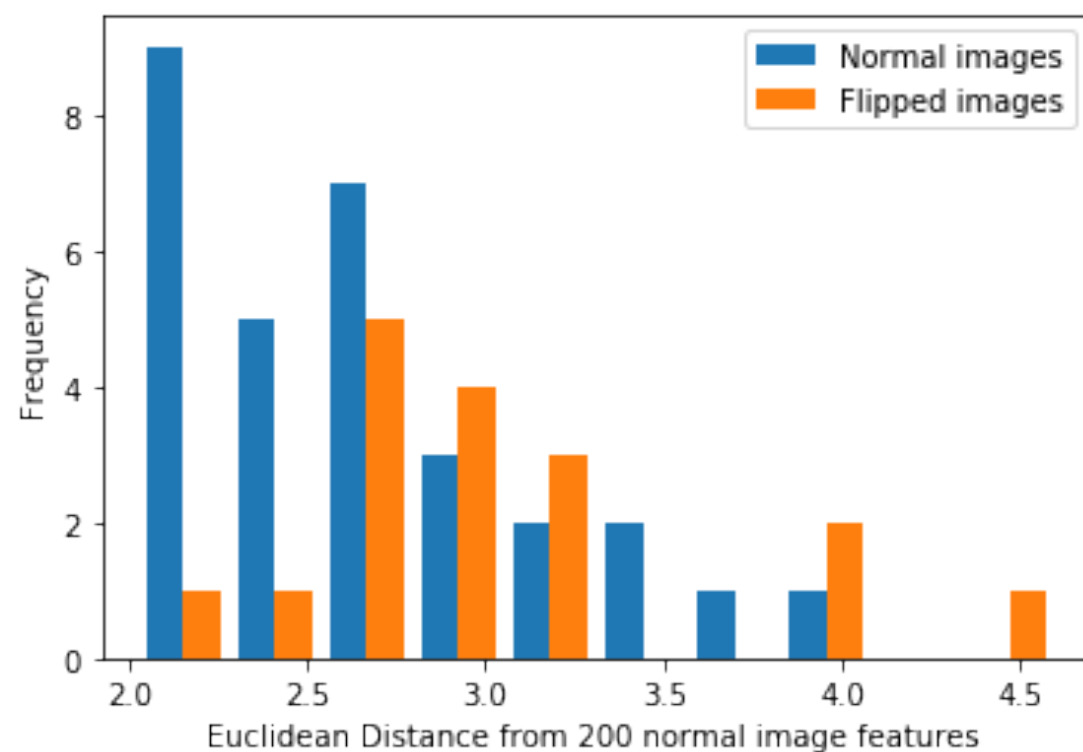
Heat map from normal images



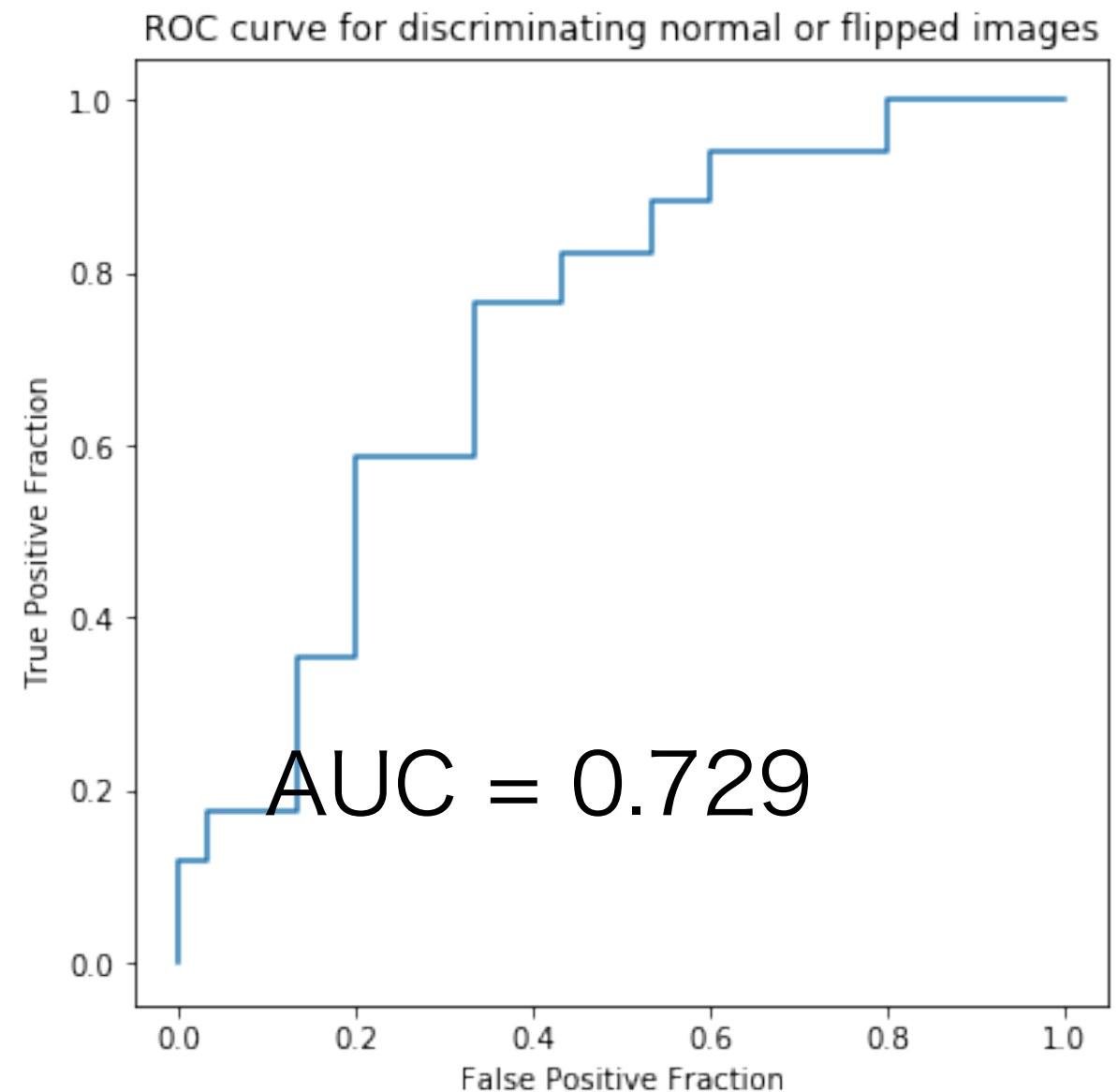
Heat map from abnormal images



データだけ集めて手法開発



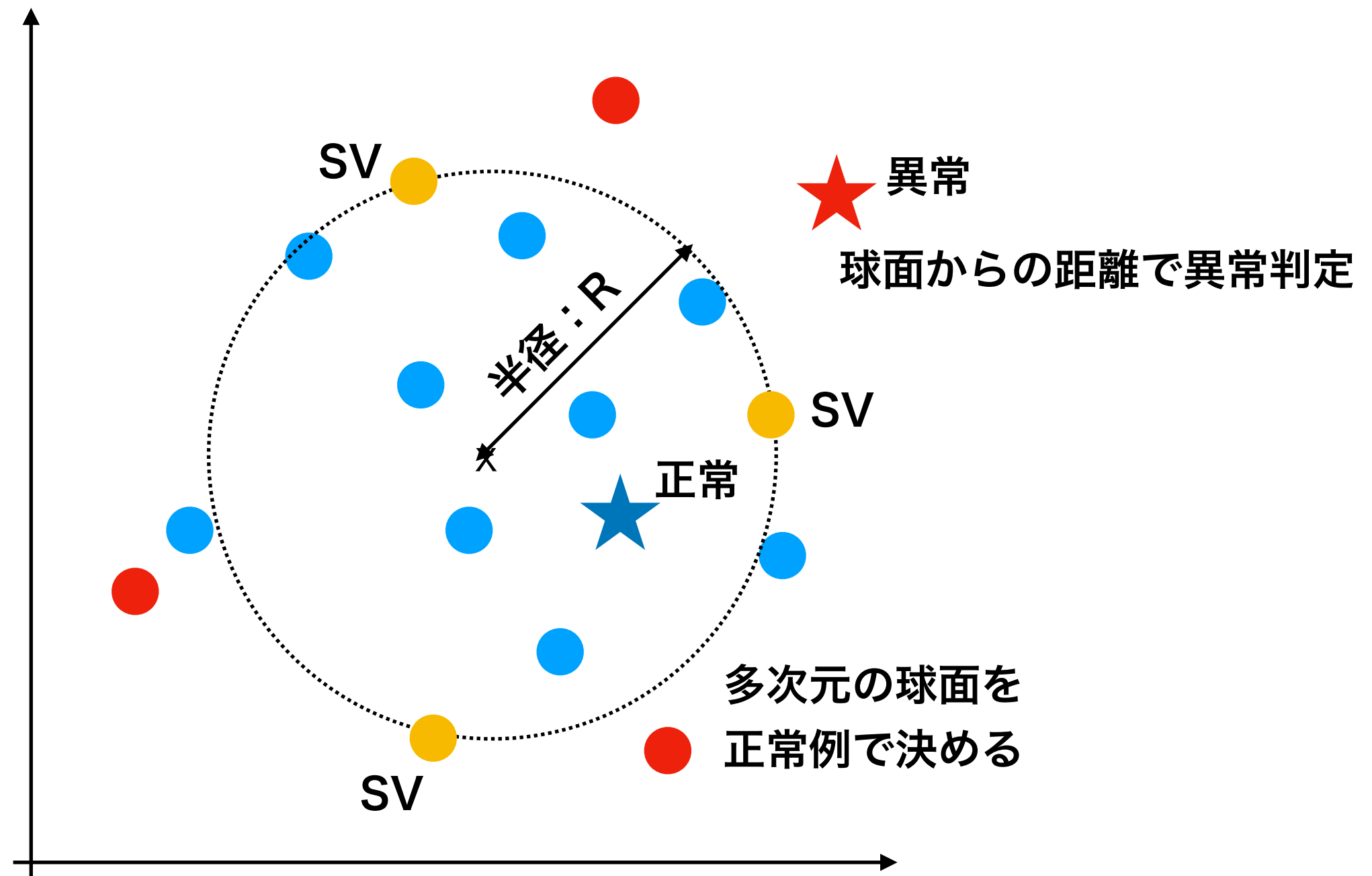
ユークリッド距離のヒストグラム



距離を確信度としたときのROC曲線

DeepSVDD

SVDD: support vector data description



DeepSAD

SAD: SEMI-SUPERVISED ANOMALY DETECTION

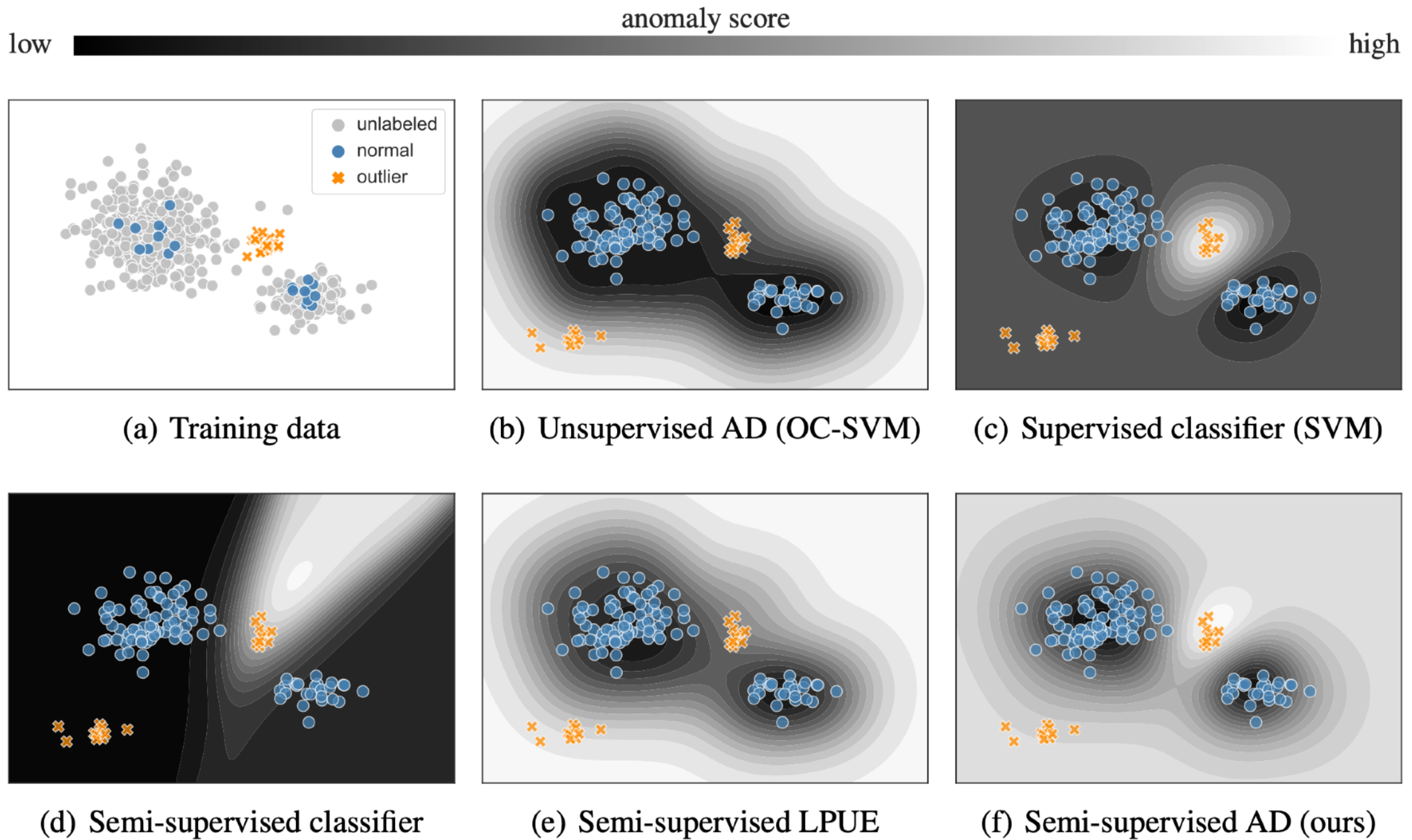
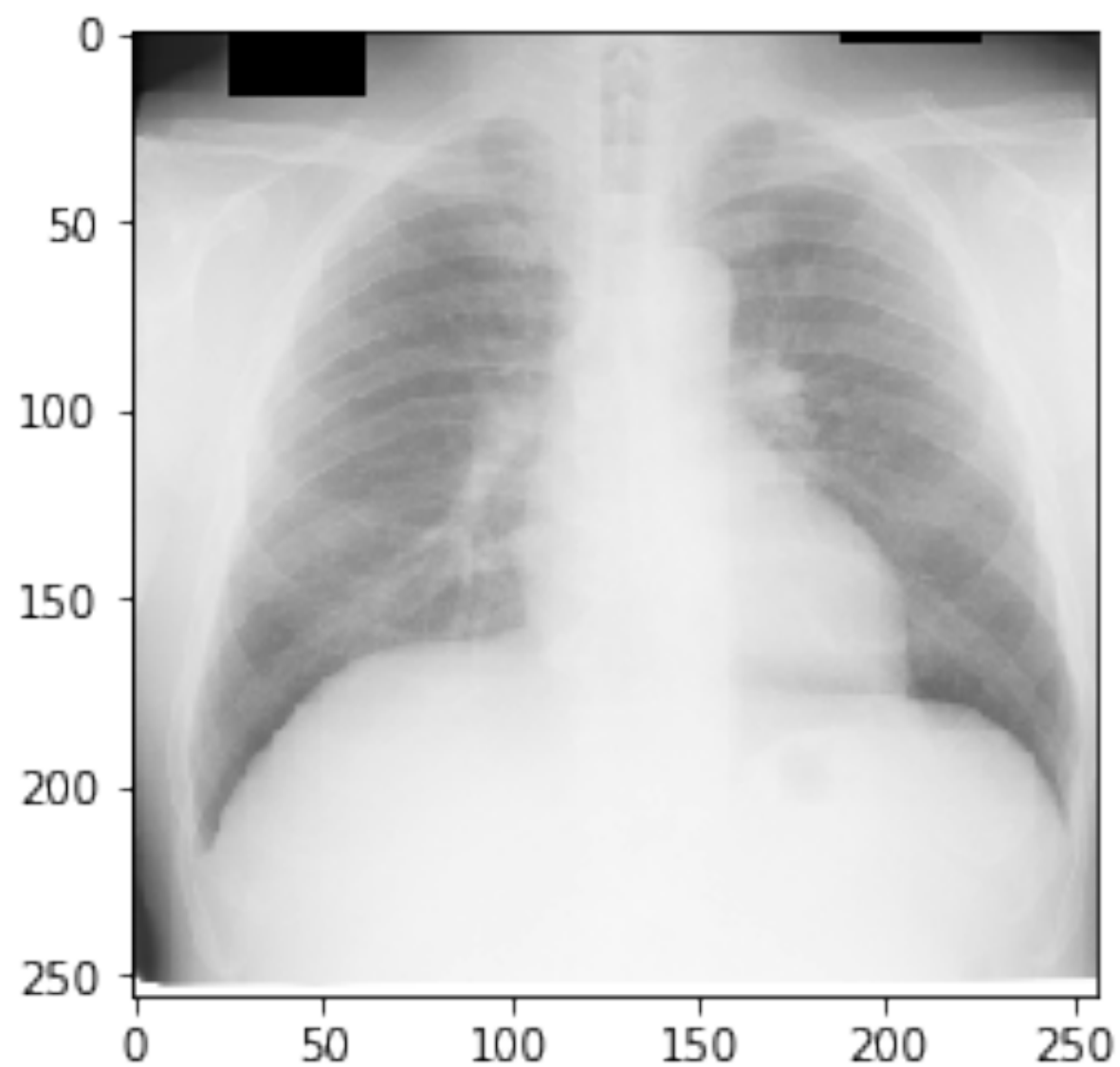
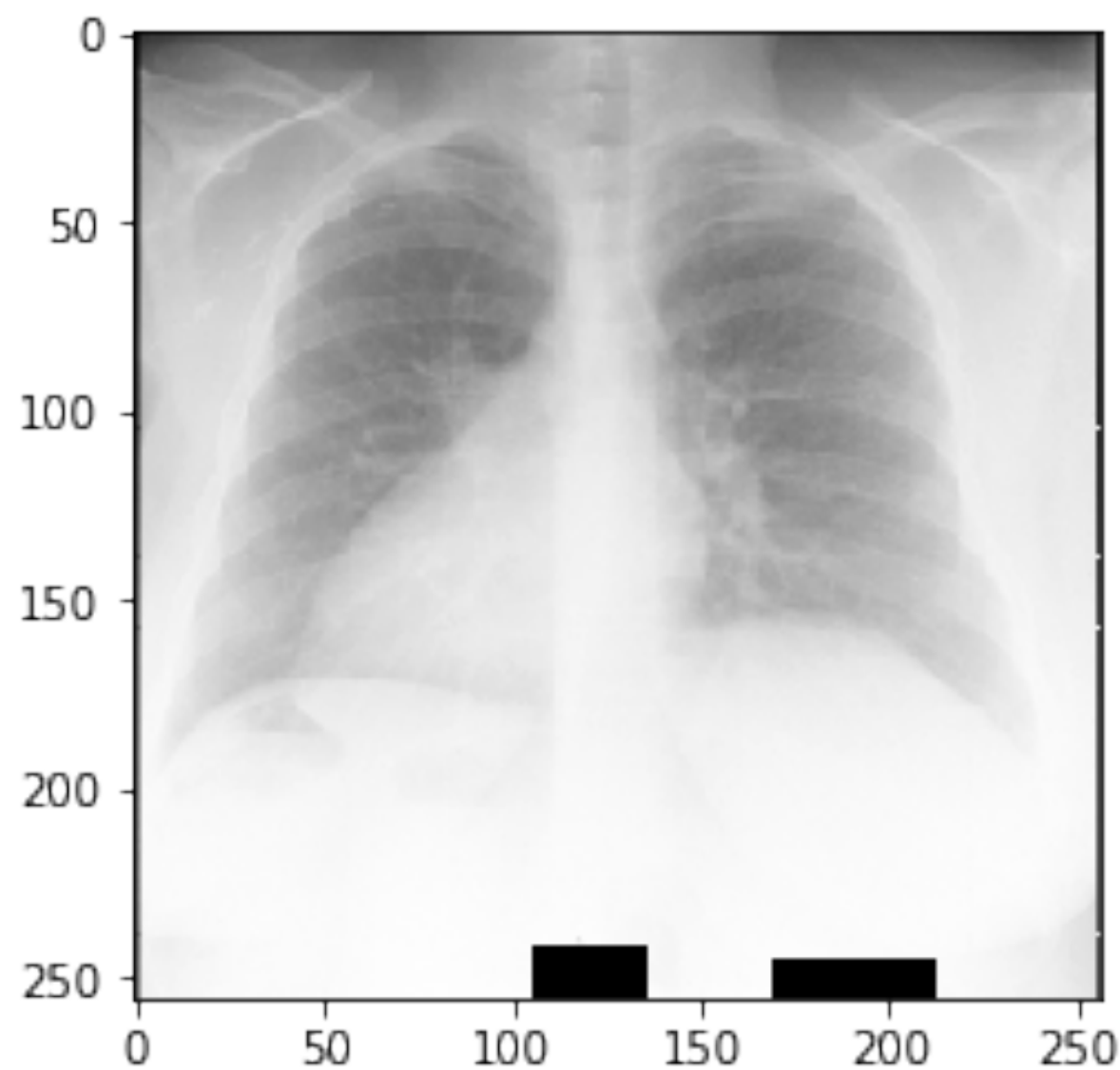


Figure 1: The need for semi-supervised anomaly detection: The training data (shown in (a)) consists of (mostly normal) unlabeled data (gray) as well as a few labeled normal samples (blue) and labeled anomalies (orange). Figures (b)–(f) show the decision boundaries of the various learning paradigms at testing time along with novel anomalies that occur (bottom left in each plot). Our semi-supervised AD approach takes advantage of all training data: unlabeled samples, labeled normal samples, as well as labeled anomalies. This strikes a balance between one-class learning and classification.

サンプルは胸部X線画像で作成済み：4種類のプログラムあり



正常像



反転像

異常部位の位置情報は？

PaDiM

PaDiM: a Patch Distribution Modeling Framework
for Anomaly Detection and Localization

Thomas Defard, Aleksandr Setkov, Angelique Loesch, Romaric Audigier

<https://arxiv.org/pdf/2011.08785>

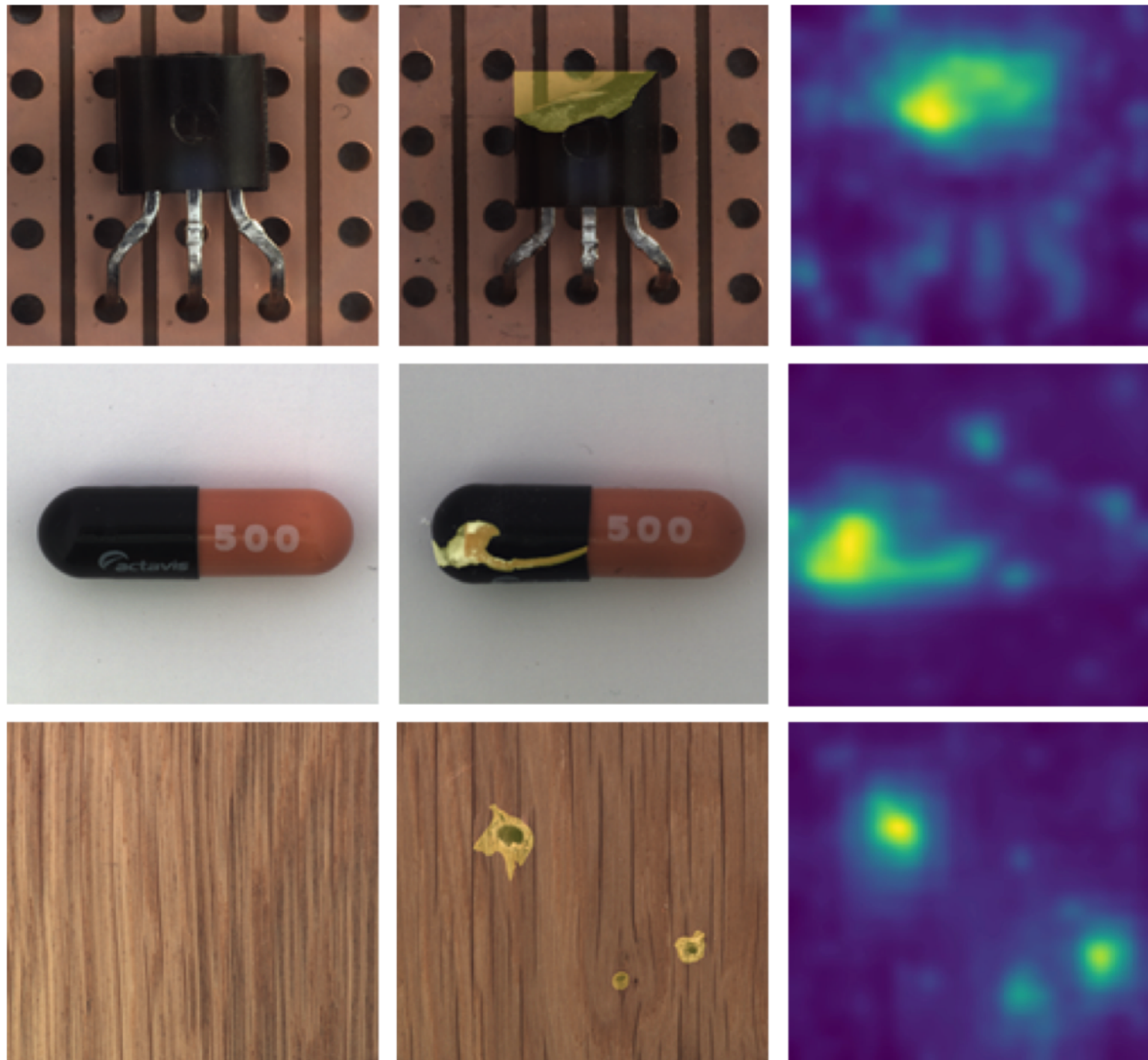


Fig. 1. Image samples from the MVTec AD [1]. *Left column*: normal images of Transistor, Capsule and Wood classes. *Middle column*: images of the same classes with the ground truth anomalies highlighted in yellow. *Right column*: anomaly heatmaps obtained by our PaDiM model. Yellow areas correspond to the detected anomalies, whereas the blue areas indicate the normality zones. Best viewed in color.