

# U-Net PyTorch実装

# 何をするか

- U-NetをPyTorchで実装し、Dice, ASSDをもとにモデルの評価をする。
- 各指標の最小値・中央値・最大値に対応する画像・予測・正解を可視化する。

# Image Segmentationとは

- Image Classification
    - 入力画像に対してクラスを割り当てる。
  - Image Segmentation
    - 画像の各ピクセルにクラスを割り当てる。
- <何が可能になるか>
- オブジェクトが画像のどこにあって
  - どんな形をしているのか知ることができる。

# チュートリアルで使用するデータセット

- Oxford-IIIT Pet Dataset
  - ペットの画像
  - 犬と猫
  - それぞれ37品種
  - 約200枚ずつ（184枚・198枚・200枚など）
- 各クラスの説明
  - class1：ペットに属するピクセル
  - class2：ペットを縁取るピクセル
  - class3：上記以外の周りのピクセル



# 使用するモデル

- U-Net

- 画像セグメンテーションに用いられるモデル

- エンコーダの特徴マップをデコーダに流すスキップ結合

# 指標

- 最適化手法

Adam

- 損失関数

CrossEntropyLoss

- 評価指標

Dice, ASSD

# 設定

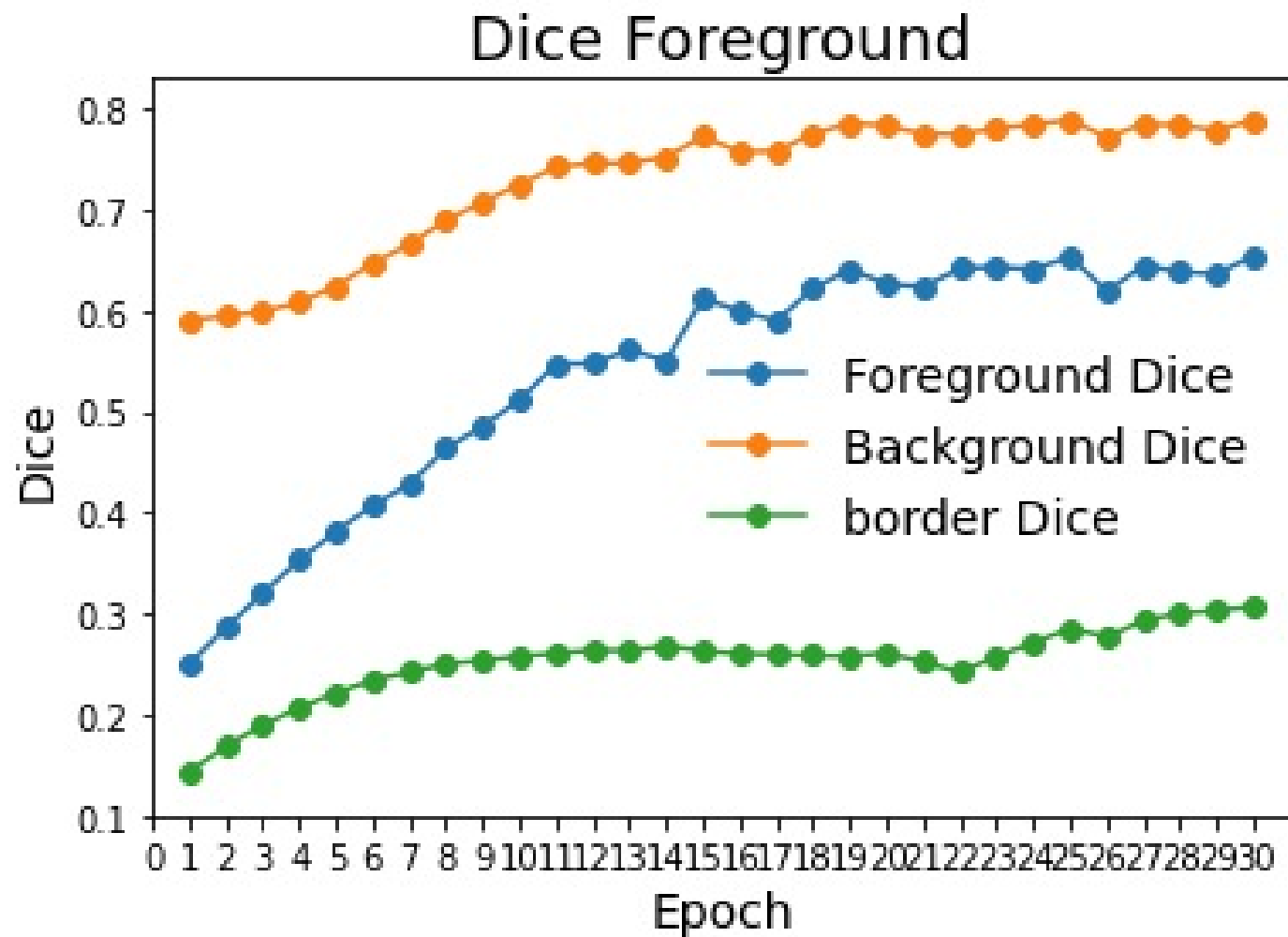
- 今回は全データ7349枚を使用
  - そのうちの70%(5144枚)を訓練用、残りの30%(2205枚)をテスト用として使う
  - 分け方はランダムで、sklearnのtrain\_test\_split関数を使う
- 
- 画像サイズ：128x128
  - バッチサイズ：64
  - エポック数：30
  - 学習率：0.0001
  - データ拡張：horizontalflip

# 結果

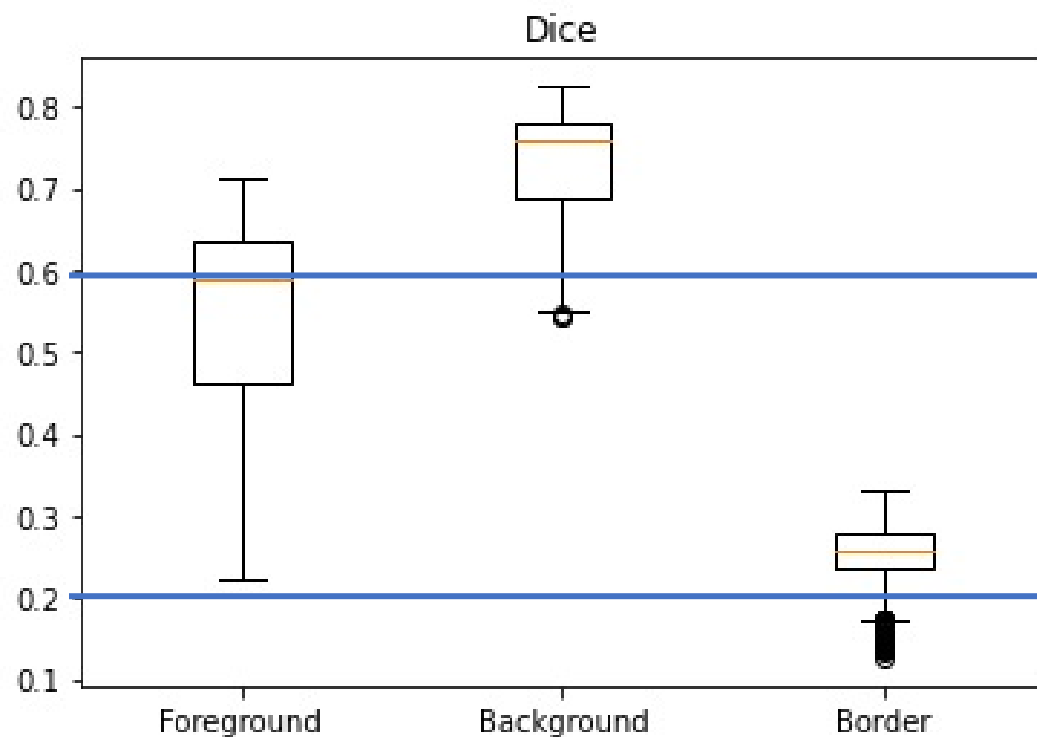
- [Foreground] Best val Dice:0.684106
- [Foreground] Median val Dice:0.464335
- [Foreground] Worst val Dice:0.244564
  
- [Background] Best val Dice:0.799941
- [Background] Median val Dice:0.692820
- [Background] Worst val Dice:0.585699
- 
- [Border] Best val Dice:0.321605
- [Border] Median val Dice:0.234757
- [Border] Worst val Dice:0.147909



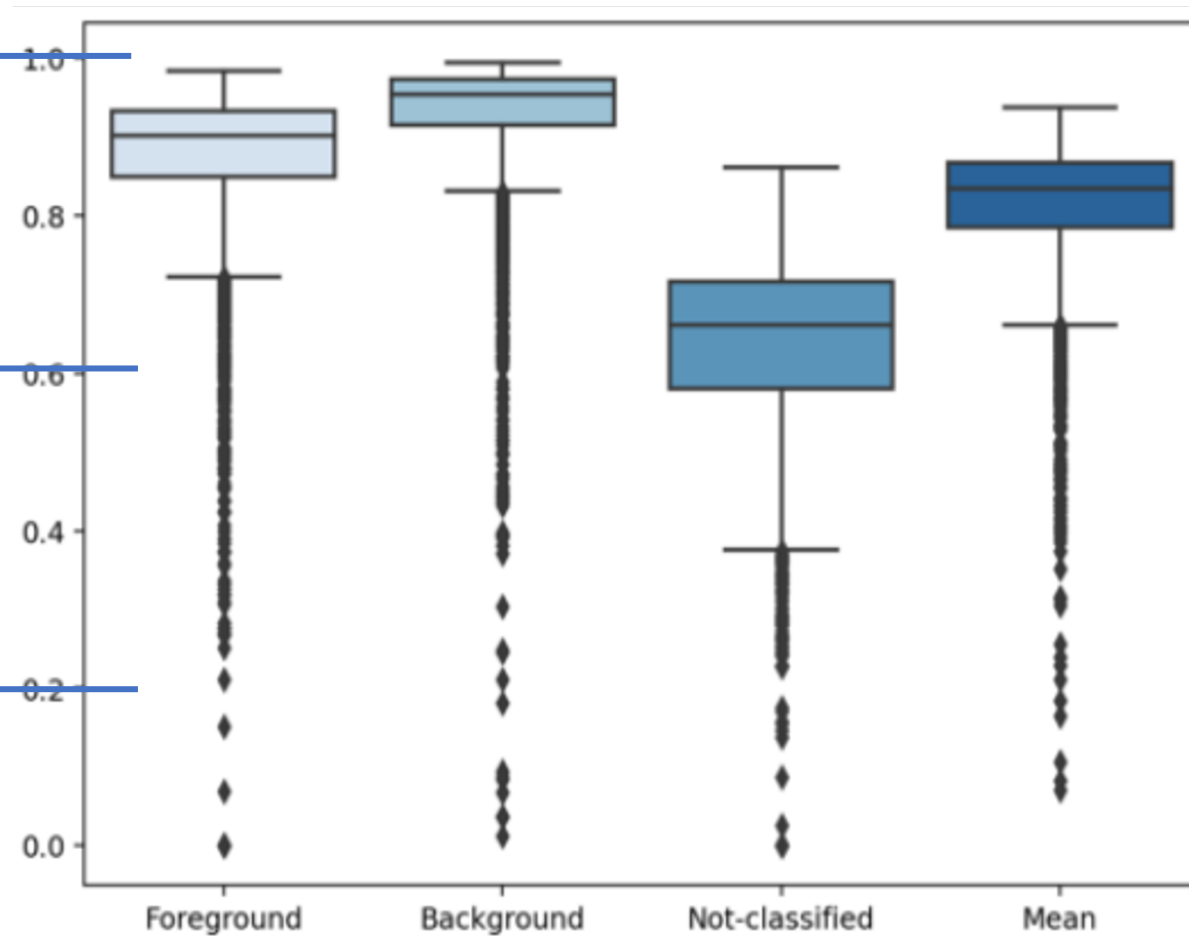
# Diceの変化



# Dice

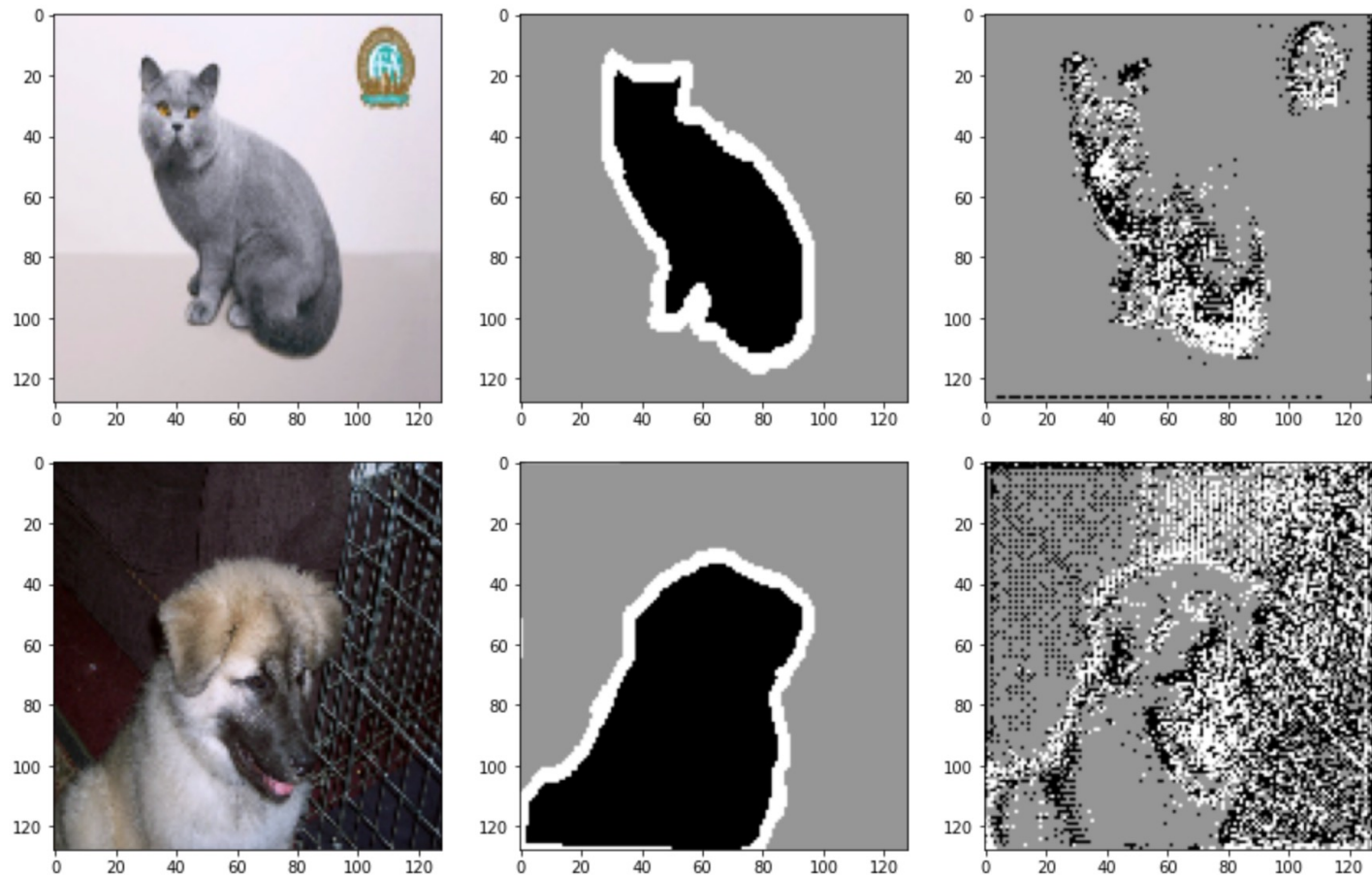


梶田の結果

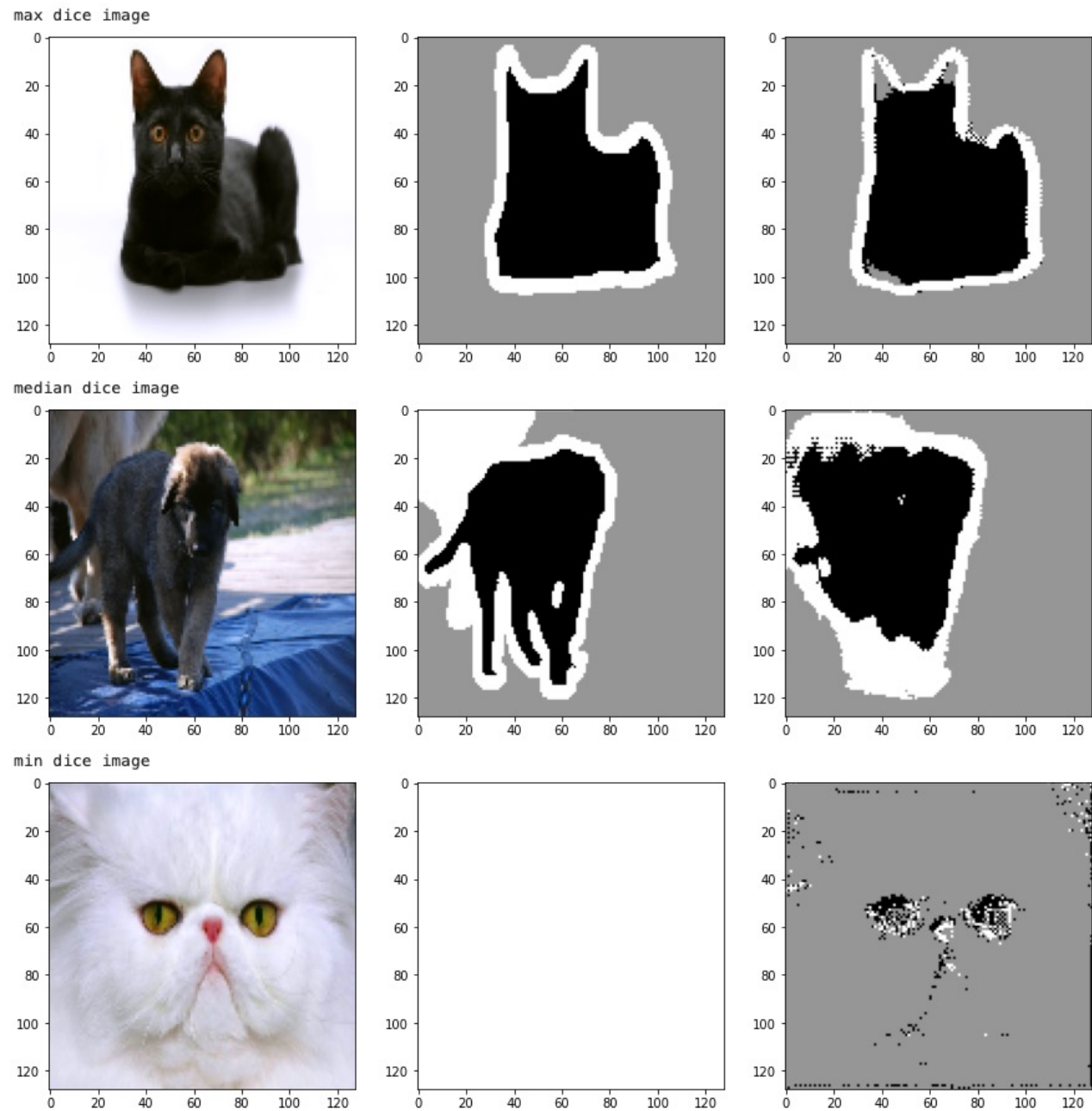


さんの結果

# 1 epoch目の予測結果



# Dice



Best Sample

Median Sample

Worst Sample

# 考察

- 差が出ている原因として考えられるもの

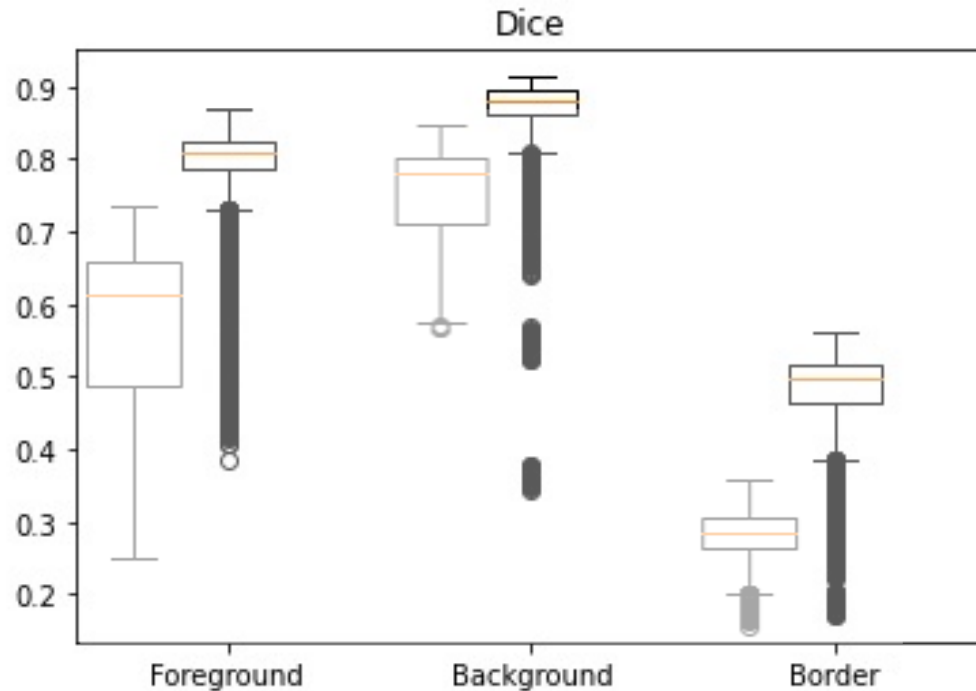
		■さん	梶田
エンコーダ		Torchvision Mobilenetv3 (pretrained=True)	U-Net official implementation Pretrained=False
訓練時	Epoch数	200	30
訓練時	Batch size	400	64*1
学習率	Adam	1e-3	1e-4

\*1 kaggle GPUのメモリが16GBでこれ以上厳しい

# 考察

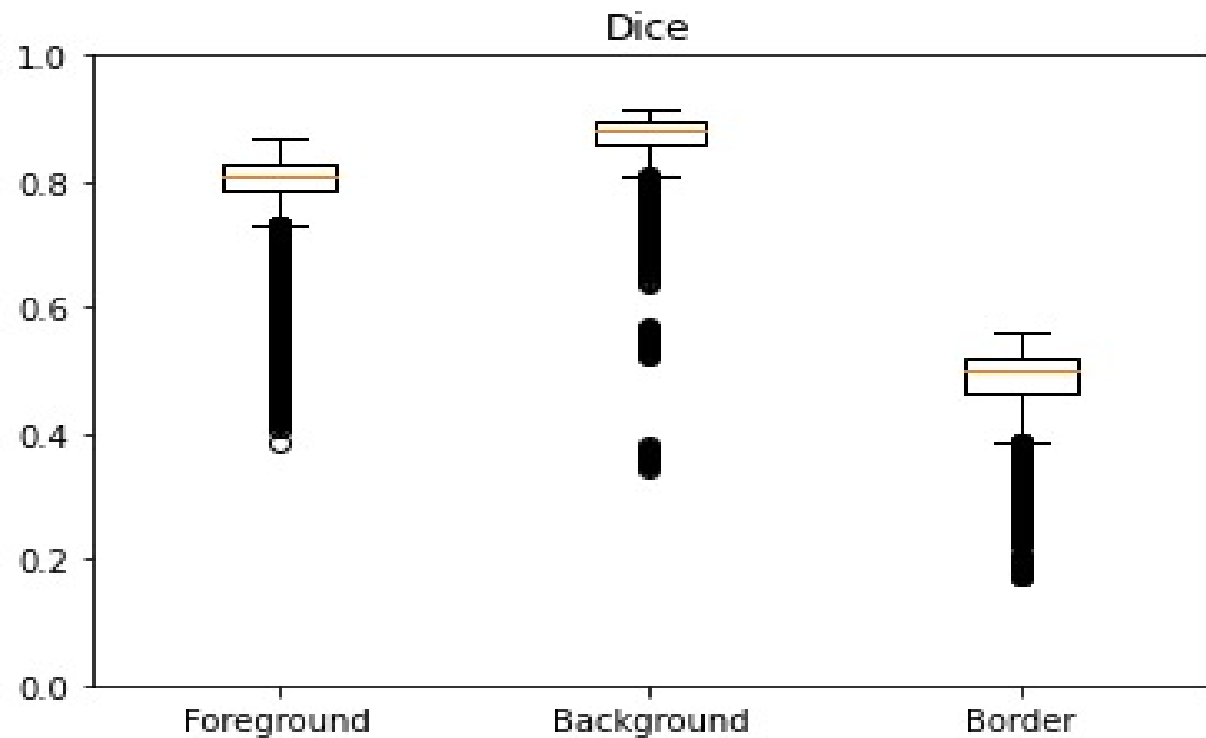
- Encoder → MobileNetV2 (pretrained=True)
- Epoch 30 → Epoch 100
- Lr 1e-4 → Lr 1e-3

各クラス右の箱ひげ図が  
変更後の結果

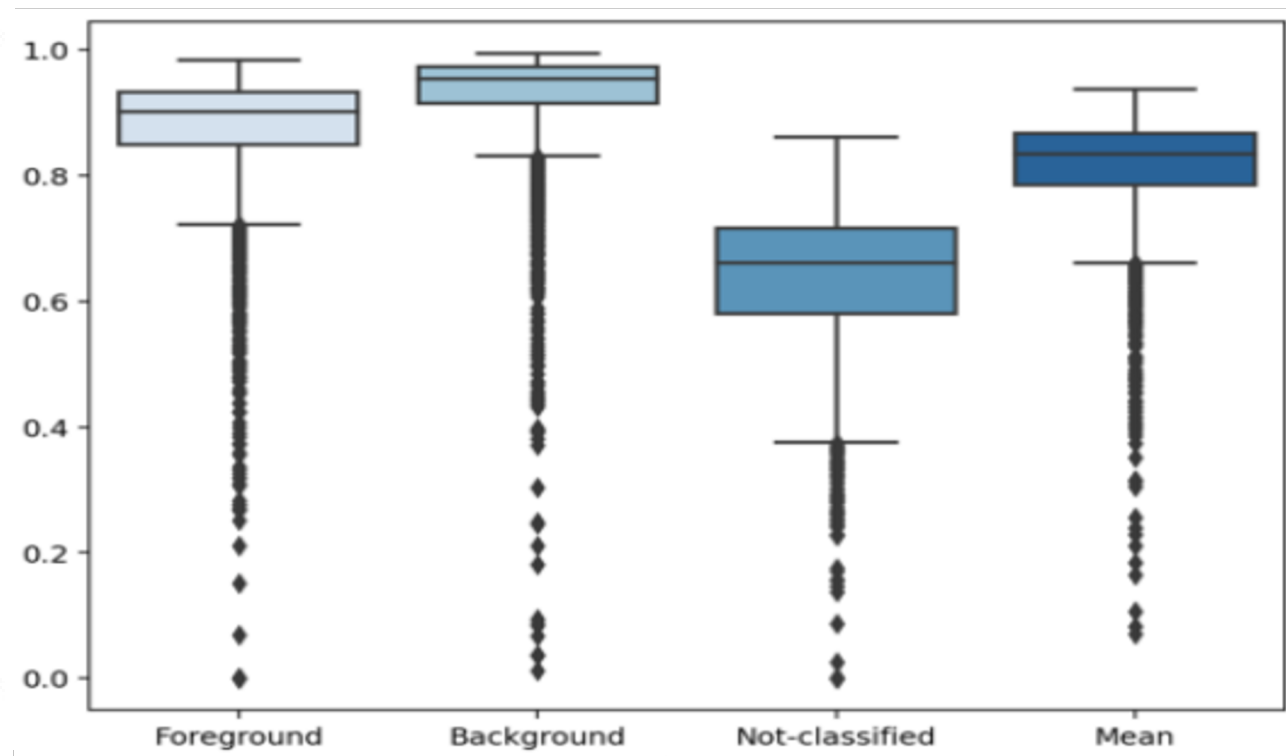


3つとも最大値・中央値が  
大きく変化

# Dice

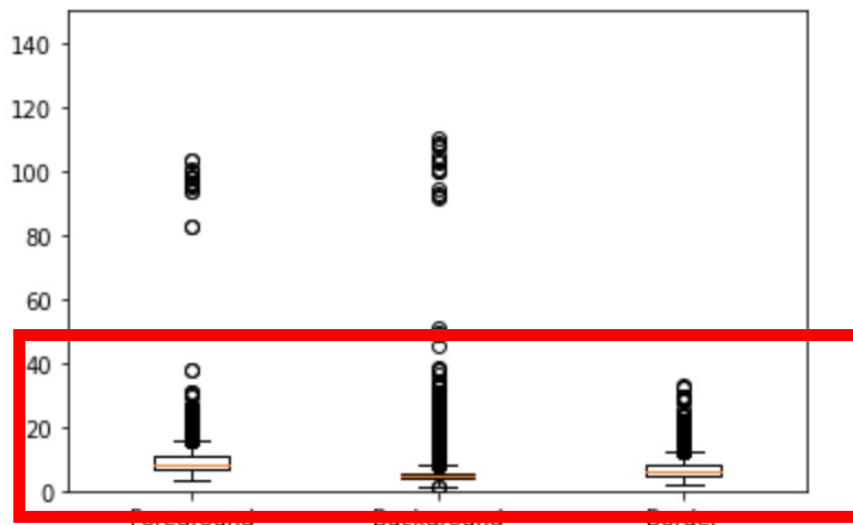


梶田の結果

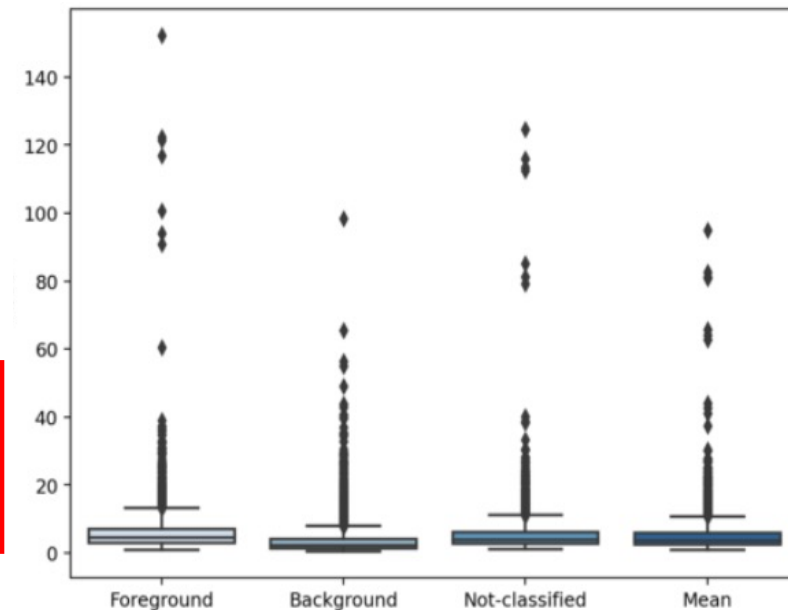


さんの結果

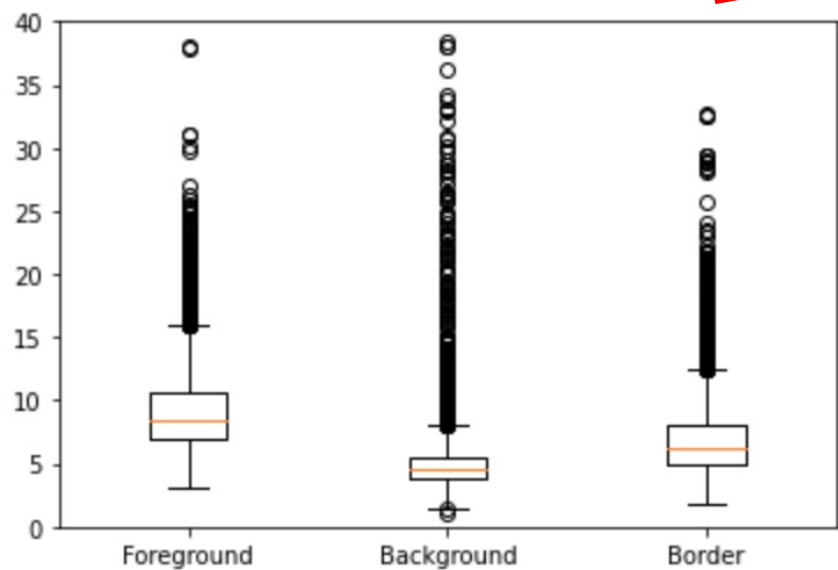
# ASSD



梶田の結果

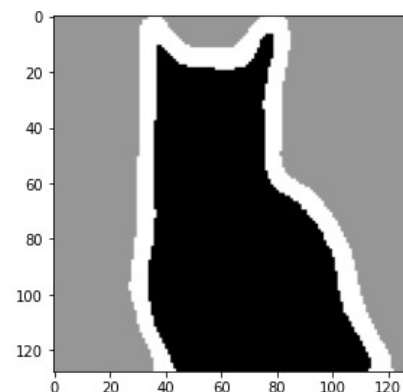
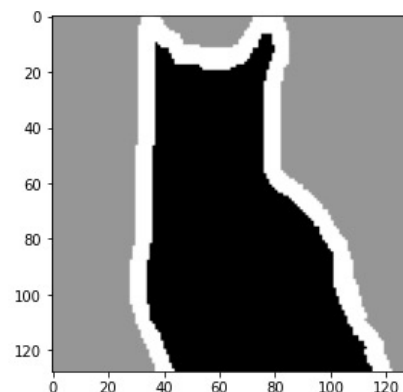
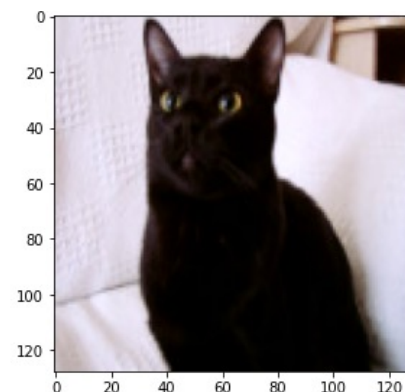


さんの結果

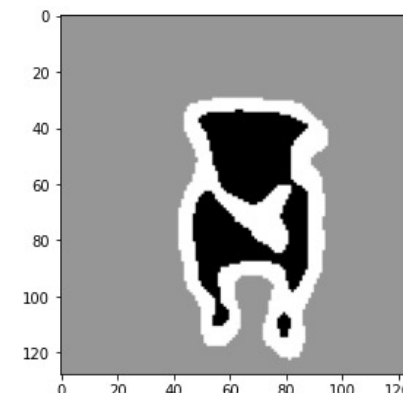
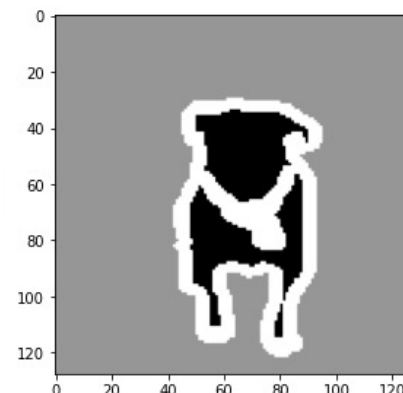
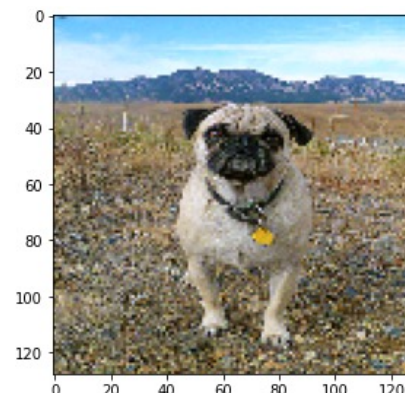




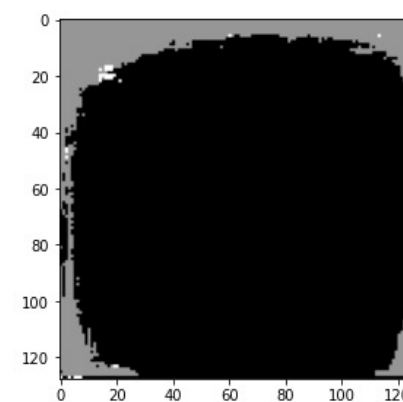
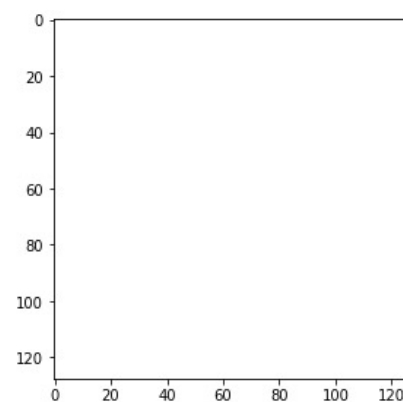
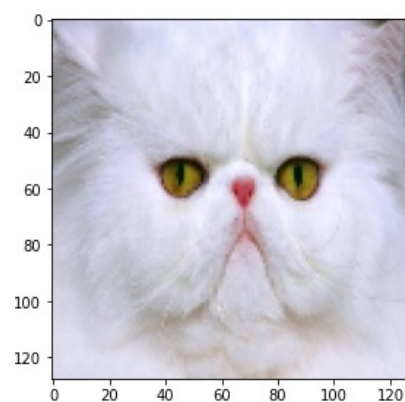
# Dice



Best Sample



Median Sample



Worst Sample