

進捗報告

1 今週やったこと

Cutout による訓練データの拡張とその考察, 以前実装した attention map の紹介と対策

2 Cutout

元画像を 9 分割した画像を前回の識別器に入力した。分割前が多義図形で分割後も多義図形と識別された画像の中で、自分の主観で明らかに多義図形である画像は 59 枚であった。また、分割前が多義図形ではなく分割後に多義図形と識別された画像の中で、同様に明らかに多義図形である画像は 5 枚であった。これらの計 64 枚の画像を前回有効とわかったグレースケール化することで $64 \times 2 = 128$ 枚を訓練データに加えることを行った。

表 1 に縦軸を真値、横軸を予測値とした混同行列結果を示す。

表 1: 元画像, グレースケール画像, 分割画像を含めた再学習

真値	多義図形	64	3	5
	風景画	0	72	0
	肖像画	1	1	70
		多義図形	風景画	肖像画
		ViT による予測値		

表 1 より識別率は 95.4% であり、これは前回の元画像 + グレースケール画像を訓練データとした識別率と比べると 0.5% 高くなっている。多義図形の識別率に関しては変わらないという結果となった。Cutout による訓練データの拡張はあまり感じられないことがわかった。今後は TDGA AA を用いて変換の自動化を図りたい。

追記：訓練データを 9 分割してテストデータとして識別することで、明らかに多義図形である分割画像を再び訓練データとすることで Cutout による拡張とするだったが、間違えてテストデータを 9 分割してそれを訓練データに入れることで拡張してしまっていた。今朝、増やしたデータ数が少ないことに違和感を感じて気づいた。次回、訂正版もかねてまとめたい。

3 attention map

以前簡易に実装した (attention のかかり方が微妙なため実装できているのかは不明) attention map の例を示す。図 1 にテスト画像と attention map を適用した画像を示す。一番下の画像のみ風景画で、他全ては多義図形である。attention がまだ見やすい画像を選んではいくものの明瞭に attention が見えるものはかなり少ない。現在取り組んでいる attention map は pytorch で実装をしようとしているが、実装に手間取っている。図 2 に対処しかねているエラーコードを示す。

4 今後の方針

attention map の実装, 多義図形画像を探す, 別の DA の実装

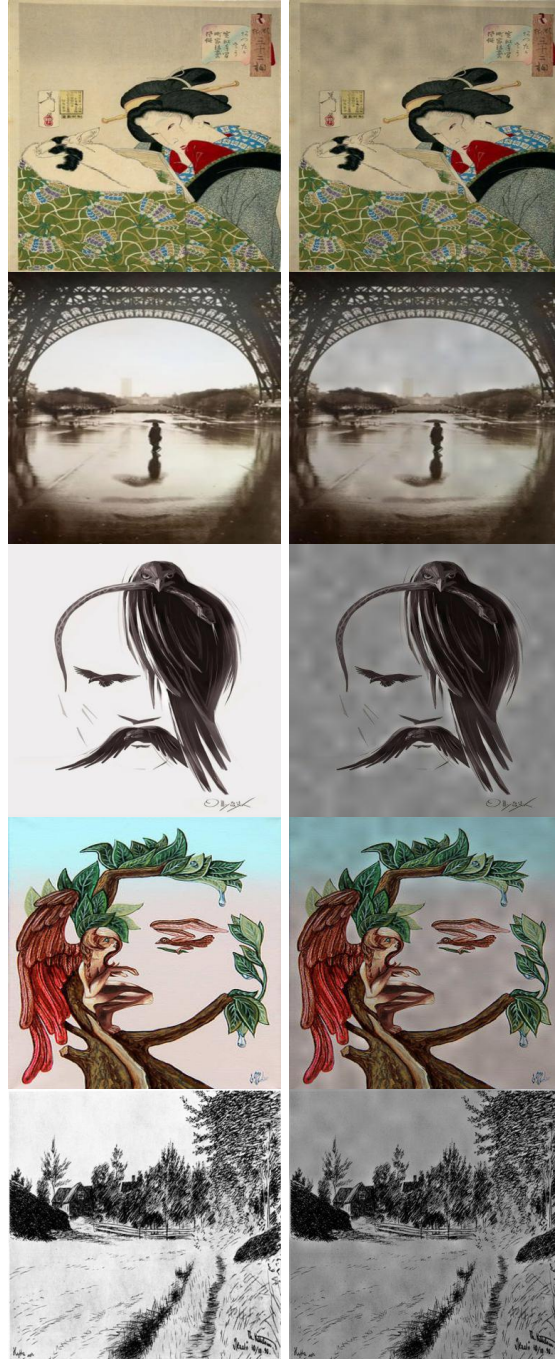


図 1: 左 : テストデータ 右 : attention map を適用した画像

```

Traceback (most recent call last):
  File "vit_attention_map2.py", line 106, in <module>
    result1=get_attention_map(img1)
  File "vit_attention_map2.py", line 64, in get_attention_map
    att_mat = net(x.unsqueeze(0))
  File "/home/setting/.pyenv/versions/anaconda3-2019.03/lib/python3.7/site-packages/torch/nn/modules/module.py", line 727, in _call_impl
    result = self.forward(*input, **kwargs)
  File "/home/ito/Atn/vision_transformer_pytorch/model.py", line 309, in forward
    feat, attn_weights = self.extract_features(x)
  File "/home/ito/Atn/vision_transformer_pytorch/model.py", line 297, in extract_features
    feat = self.transformer(emb)
  File "/home/setting/.pyenv/versions/anaconda3-2019.03/lib/python3.7/site-packages/torch/nn/modules/module.py", line 727, in _call_impl
    result = self.forward(*input, **kwargs)
  File "/home/ito/Atn/vision_transformer_pytorch/model.py", line 209, in forward
    out, weights = layer(out)
  File "/home/setting/.pyenv/versions/anaconda3-2019.03/lib/python3.7/site-packages/torch/nn/modules/module.py", line 727, in _call_impl
    result = self.forward(*input, **kwargs)
  File "/home/ito/Atn/vision_transformer_pytorch/model.py", line 160, in forward
    out = self.dropout(out)
  File "/home/setting/.pyenv/versions/anaconda3-2019.03/lib/python3.7/site-packages/torch/nn/modules/module.py", line 727, in _call_impl
    result = self.forward(*input, **kwargs)
  File "/home/setting/.pyenv/versions/anaconda3-2019.03/lib/python3.7/site-packages/torch/nn/modules/dropout.py", line 58, in forward
    return F.dropout(input, self.p, self.training, self.inplace)
  File "/home/setting/.pyenv/versions/anaconda3-2019.03/lib/python3.7/site-packages/torch/nn/functional.py", line 983, in dropout
    else _VF.dropout(input, p, training))
TypeError: dropout(): argument 'input' (position 1) must be Tensor, not tuple

```

図 2: エラーコード