

進捗報告

1 今週やったこと

分割によるタッチ識別の考察, 画像にグレースケールを施した, ViT の pytorch 実装による識別率の向上

2 タッチ識別の考察

タッチ識別か否かを確認するために元画像を 9 分割し, resize したものを訓練データとして学習させた. 同様にテストデータも 9 分割し, 入力画像は全 $216 \times 9 = 1944$ 枚とした. また, ラベルは分割する前の画像と同じものを付与した. 表 2 に縦軸を真値, 横軸を ViT による予測値とした混同行列を示す. また, 表 1 に元の 216 枚を ViT を用いて識別した混同行列結果を示す. 表 2 より表 1 は単に各値を 9 倍したものにはなっていないかった. 全体の識別率は元画像のほうが 93.1% であるのに対して, 9 分割した画像では 76.8% となった. 識別率が大きく下がっていることから全てをタッチ識別しているわけではないことがわかる. しかし肖像画に関しては 9 分割しているにもかかわらず正しく肖像画と判別しているのは 563 枚で 86.9% を占めている. このことから肖像画は大部分がタッチ識別されているとわかる. また, 多義図形に関しては正しく多義図形と判別しているのは 384 枚で 59.2% と全体の半分程度しか判別できておらず, 9 分割画像による識別がうまくいっていない. このことは多義図形性を学習できていないことを示しており, タッチ識別によって判別できていないことがわかる. 以上より, 多義図形に関しては多義図形性を認識することが重要であり, タッチ識別によって十分な識別を行うことはできないと考えられる.

表 1: 元画像の混同行列

真値	多義図形	61	5	6
	風景画	0	72	0
	肖像画	1	3	68
		多義図形	風景画	肖像画
		ViT による予測値		

表 2: 9 分割画像の混同行列

真値	多義図形	384	104	160
	風景画	1	545	102
	肖像画	9	76	563
		多義図形	風景画	肖像画
		ViT による予測値		

3 グレースケールについて

テストデータにグレースケールを行った. グレースケール化した画像を識別器に入れてみて識別率に変化がなければ多義図形性を保持した加工といえる. 多義図形性を保持した加工を探索したら訓練データに DA することで識別率の向上を図る. 表 3 にグレースケール化した画像を入力画像とした混同行列結果を示す. 表 3 より明らかに多義図形性を失っていることがわかる. この原因として考えられることは, 多義図形をタッチ識別で判断している可能性, 多義図形は色のグラデーションを重要視し, それが加工によって失われている可能性である. 画像分割実験よりも明らかに多義図形を識別できていないことから, タッチ識別である可能性は否めない. 他の加工とも比較して明らかにしていきたい.

表 3: グレースケール画像の混同行列

真値	多義図形	24	23	25
	風景画	0	70	2
	肖像画	1	5	66
		多義図形	風景画	肖像画
		ViT による予測値		

4 識別率の向上

Vision Transformer の実装において EfficientNet と VisionTransformer を合わせた VisionTransformer の Pytorch 実装を用いた. 今回実装した ViT も同様に元の 216 枚を用いて識別した. 表 4 に縦軸を真値, 横軸を ViT (pytorch) による予測値とした混同行列を示す. 以前は識別率 93% であったのに対して, 識別率が 94% に上がった. どちらの識別器も Imagenet21k で転移学習したもの

であるが, EfficientNet と VisionTransformer を合わせたもののほうが識別率が上がったことからより良い識別器といえる.

表 4: 元画像の混同行列

真値	多義図形	63	4	5
	風景画	0	72	0
	肖像画	2	2	68
		多義図形	風景画	肖像画
		ViT (pytorch) による予測値		

5 今後の方針

attention map のより明瞭な表現の実装, TransGAN
の実装, DA の実装