

## 進捗報告

### 1 今週やったこと

- CNN を使った分類実験
- Triplet Loss の導入検討

### 2 今週の収穫

#### 2.1 CNN を使った分類実験

今回は, VGG16, VGG19, ResNet50 の 3 つの CNN をファインチューニングして分類実験をしてみた. 表 1 に実験パラメータを示す. なおいずれのネットワークにも early stopping を使用している.

表 1: CNN の実験パラメータ

最適化関数	SGD
学習率	$1.0 \times 10^{-4}$
損失関数	categorical cross entropy
エポック数	100
バッチサイズ	16

図 1, 図 2 に VGG16 を使った際の訓練時と検証時の accuracy と loss の推移を示す. early stopping の結果 8 epoch 目で終了した. accuracy は増加しており, loss は減少していることから学習は進んでいるようであった.

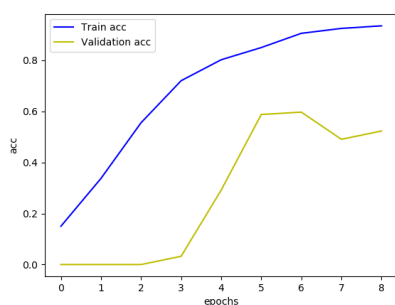


図 1: VGG16 使用時の accuracy の推移

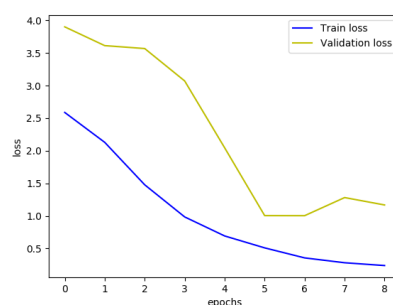


図 2: VGG16 使用時の loss の推移

表 2 に識別結果を示す. 識別率の結果自体は CAE とそれほど変わらなかったが, CAE では識別できていなかったふたご座やオリオン座がしっかり識別できている.

次に, VGG19 での結果を述べる. 図 3, 4 に VGG19 を使った際の訓練時と検証時の accuracy と loss の推移を示す. early stopping の結果 9 epoch 目で終了した. accuracy は増加しており, loss は減少していることから学習は進んでいるようであった.

表 3 に識別結果を示す. 識別結果は CAE, VGG16 より少し良い結果となった.

表 2: VGG16 の識別結果

		星図														
		はくちょう	ふたご	いて	カシオペア	こぐま	こと	おおぐま	おおいぬ	オリオン	おとめ	おうし	ペガサス	さそり	しし	わし
写真	はくちょう	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0
	ふたご	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
	いて	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0
	カシオペア	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0
	こぐま	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0
	こと	0	0	0	0	0	0	0	2	0	1	0	0	0	0	0
	おおぐま	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0
	おおいぬ	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0
	オリオン	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
	おとめ	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0
	おうし	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0
	ペガサス	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0
	さそり	0	0	0	0	0	0	0	0	0	0	2	0	0	1	0
	しし	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0
	わし	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0
正答率		0.200														
ベースライン		0.067														

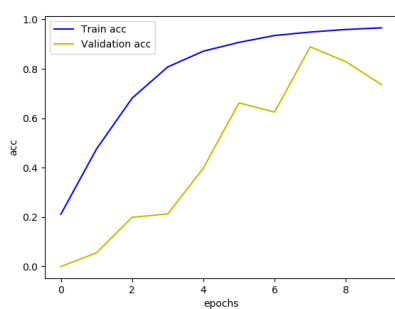


図 3: VGG19 使用時の accuracy の推移

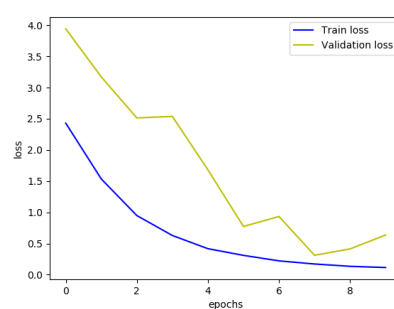


図 4: VGG19 使用時の loss の推移

表 3: VGG19 の識別結果

		星図														
		はくちょう	ふたご	いて	カシオペア	こぐま	こと	おおぐま	おおいぬ	オリオン	おとめ	おうし	ペガサス	さそり	しし	わし
写真	はくちょう	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0
	ふたご	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
	いて	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0
	カシオペア	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0
	こぐま	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
	こと	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0
	おおぐま	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0
	おおいぬ	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0
	オリオン	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
	おとめ	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0
	おうし	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0
	ペガサス	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0
	さそり	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0
	しし	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0
	わし	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0
正答率		0.289														
ベースライン		0.067														

最後に, ResNet50 での結果を述べる. 図 5, 6 に ResNet50 を使った際の訓練時と検証時の accuracy と loss の推移を示す. early stopping の結果 14 epoch 目で終了した. accuracy は増加しており, loss は減少していることから学習は進んでいるようであった.

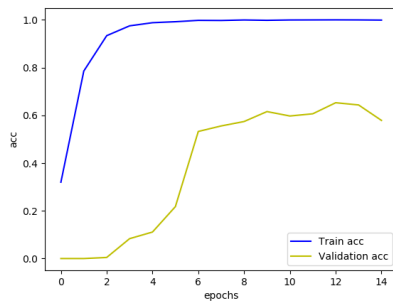


図 5: ResNet50 使用時の accuracy の推移

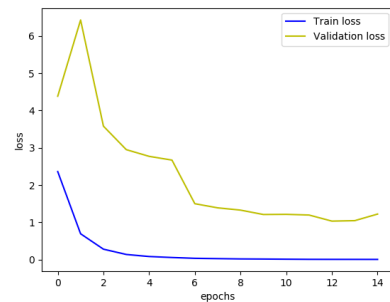


図 6: ResNet50 使用時の loss の推移

表 4 に識別結果を示す. 識別結果は CAE, VGG16 とそれほど変わらなかった.

表 4: ResNet50 の識別結果

		星図														
		はくちょう	ふたご	いて	カシオペア	こぐま	こと	おおぐま	おおいて	オリオン	おとめ	おうし	ペガサス	さそり	しし	わし
写真	はくちょう	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0
	ふたご	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0
	いて	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0
	カシオペア	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0
	こぐま	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0
	こと	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1
	おおぐま	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0
	おおいて	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
	オリオン	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1
	おとめ	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0
	おうし	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
	ペガサス	0	0	0	0	0	0	0	0	0	1	0	2	0	1	0
	さそり	0	1	0	0	1	0	0	0	0	1	0	1	0	0	0
	しし	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
	わし	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0
正答率		0.178														
ベースライン		0.067														

結果的にどの CNN でも CAE と同じくらい (またはそれ以上) の精度が出ていたので, このまま Triplet Loss の導入を検討していきたいと考える.

## 2.2 Triplet Loss の導入検討

github 上にあった, MNIST を使った Triplet Loss のコードは動かしてみた. 自分のデータを使った実験もしていきたいと思う.

## 3 今後の方針

Triplet Loss の実装と結果の検証