

知能情報実験 III（データマイニング班）
毒のある蛇かそうでないかを画像判別

215732C 佐久本元気

215736F 西野大河

215742A 米須悠

215746C 新垣樹

提出日：2023 年 8 月 3 日

目次

1	テーマ「毒のある蛇かそうでないかを画像判別」とは	2
2	実験方法	2
2.1	実験目的・目標	2
2.2	データセット構築	2
2.3	前処理	2
2.4	実験 1	2
2.5	実験 2	3
2.6	実験 3	3
2.7	実験 4	4
3	実験結果	5
3.1	実験 1	5
3.2	実験 2	6
3.3	実験 3	7
3.4	実験 4	8
4	考察	8
5	意図していた実験計画との違い	9
6	まとめ	9

1 テーマ「毒のある蛇かそうでないかを画像判別」とは

本グループでは画像で提示された蛇の毒の有無を予測することを対象問題として設定した。具体的な問題解決の方法としては、機械学習における画像認識・分類の技術を活用し、主に CNN（畳み込みニューラルネットワーク）を用いて行う。CNN は、[1] によると「CNN はいくつもの深い層を持ったニューラルネットワークであり、主に画像認識の分野で価値を生んでいるネットワーク」で、「一般物体認識と呼ばれる画像認識のタスクで価値を発揮し、優れた性能を備えるアルゴリズム」とある。また、[2] によると、「CNN の出力層はデータを解釈し、画像の予測や分類を行う」とある。まとめると、CNN を用いることで蛇の画像の認識を行うことができ、蛇の毒の有無を予測することが可能であると考えられる。このテーマの実験を行うことの意義は、機械学習を用いた画像認識の仕方を学び、それらを様々なものに対する予測や分類に適用することができる点にあると考える。

2 実験方法

2.1 実験目的・目標

画像に写っている蛇の毒の有無を予測し、正しいか確認することが目的である。また、予測の正答率をできる限り高めることも目的の一つである。具体的な数値としては、正解率 80% を目標とする。

2.2 データセット構築

本実験では 4 種類のデータセットを構築して実験を行った。

なお、データセット 2~4 の画像収集には Google Chrome の拡張機能である「Image downloader - Imageye」を利用した。

2.3 前処理

OpenCV で画像の前処理を行い、予測を適切に行えるようにした。

2.4 実験 1

2.4.1 実験内容

データセットを train, val, test で分割し、人が happy か sad を識別する学習モデルを用いて、学習モデルの評価を行う。また、学習用として Kaggle で取得したデータセットの classification snake species を利用する。

2.4.2 使用する学習モデル

ImageClassification (<https://github.com/nicknochnack/ImageClassification>)

2.4.3 使用するデータセット

classification snake species (<https://www.kaggle.com/datasets/nikhilshingadiya/sample-0>)

2.5 実験 2

2.5.1 実験内容

学習済みモデルの作成および学習モデルの評価を行う。新たなデータセット dataset1 を用いて学習し、上述のデータセットとは異なる indian datasets を用いて再テストを行う。

2.5.2 使用する学習モデル

学習モデルは、VGG16 (TensorFlow 上にある学習済みモデル) に drop out layer の追加などのファインチューニングを行い作成したものを使用する。

2.5.3 使用するデータセット

訓練用データセット

- dataset1…無作為に選んだ毒蛇・毒なし蛇の画像 (ハブやアカマタ等 16 種) を合計 600 枚程度集めたデータセット。用いる画像の条件は、「人や関係のない動物が写っていない」「文字がなるべく入っていない」とした。

評価用データセット

- Indian-Snakes-Dataset…インドの毒蛇と毒なし蛇が含まれている信頼性の高いデータセット。GitHub 上から、ダウンロードし利用する。

Indian-Snakes-Dataset (<https://github.com/arjun921/Indian-Snakes-Dataset>)

2.6 実験 3

2.6.1 実験内容

実験 2 で作成した学習済みモデルを利用し、同様に学習モデルの評価を行う。新たなデータセット dataset2 を用いて学習し、Indian-Snakes-Dataset を用いて精査を行う。

2.6.2 使用する学習モデル

実験 2 と同様の学習モデルを使用する。

2.6.3 使用するデータセット

訓練用データセット

- dataset2…実験 2 で訓練用データセットとして用いた dataset1 に世界各地の毒蛇、毒なし蛇の画像を追加したデータセット。まず、地域を北アメリカ州、南アメリカ州、アフリカ州、ヨーロッパ州、アジア州（さらに東南アジア、南アジア、中央アジア、西アジアに分ける）、オセアニア州に分け、各地域ごとに毒蛇、毒なし蛇を各 2~5 種類選定する。その中から地域ごとの蛇の画像が合計 5 枚になるようにデータセットに追加した。

評価用データセット

- Indian-Snakes-Dataset …実験 2 同様のデータセット。

2.7 実験 4

2.7.1 実験内容

実験 2 で作成した学習済みモデルを利用し、同様に学習モデルの評価を行う。新たなデータセット dataset3 を用いて学習し、Indian-Snakes-Dataset を用いて精査を行う。

2.7.2 使用する学習モデル

実験 2 と同様の学習モデルを使用する。

2.7.3 使用するデータセット

訓練用データセット

- dataset3…実験 3 で訓練用データセットとして用いた dataset2 に各地域ごとの蛇の画像をさらに追加したデータセット。追加した枚数の内訳は以下の通りである。

評価用データセット

- Indian-Snakes-Dataset …実験 2 同様のデータセット。

表 1 データセット 3 に追加した枚数の内訳

地域	毒あり	毒なし
北アメリカ州	96	120
南アメリカ州	135	60
アフリカ州	143	80
ヨーロッパ州	148	100
東南アジア	89	60
南アジア	143	180
中央アジア	54	30
西アジア	61	90
オセアニア州	91	30
合計	960	750

3 実験結果

各実験における学習モデルの評価や精査の結果は以下の通りになった。ここで実験結果として示されている数値として、性能評価指標（適合率・再現率・正解率）がある。適合率は、正確に予想された陽性サンプル数/予測された陽性サンプル数で表される。再現率は、正確に予測された陽性サンプル数/実際の陽性サンプル数で表される。正解率は、(正確に予測された陽性サンプル数 + 正確に予測された陰性サンプル数)/全サンプル数で表される。

3.1 実験 1

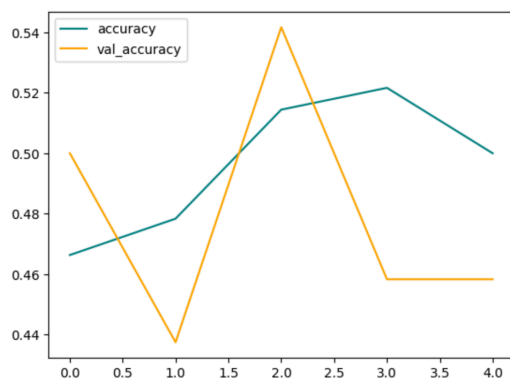


図 1 実験 1 Accuracy

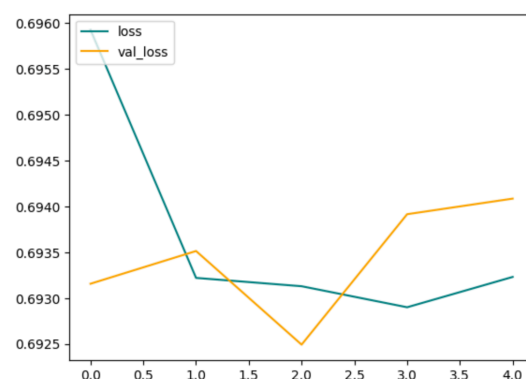


図 2 実験 1 Loss

3.1.1 学習モデルの評価

- 適合率：tf.Tensor(0.453125, shape=(), dtype=float32)
- 再現率：tf.Tensor(1.0, shape=(), dtype=float32)
- 全体の正解率：tf.Tensor(0.453125, shape=(), dtype=float32)

3.2 実験 2

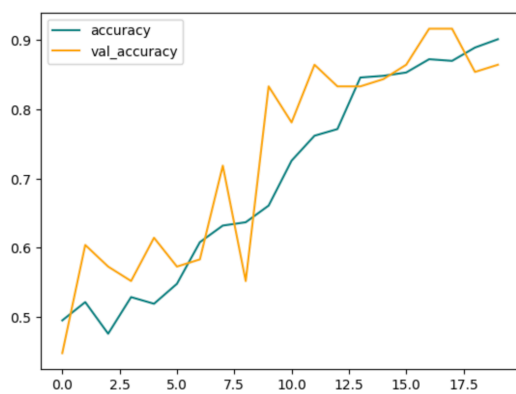


図 3 実験 2 Accuracy

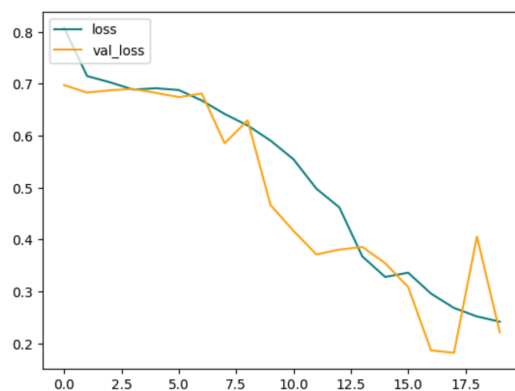


図 4 実験 2 Loss

3.2.1 学習モデルの評価

- 適合率：tf.Tensor(0.96638656, shape=(), dtype=float32)
- 再現率：tf.Tensor(0.7615894, shape=(), dtype=float32)
- 正解率：tf.Tensor(0.8566308, shape=(), dtype=float32)

3.2.2 Indian-Snakes-Dataset による精査

- 適合率：tf.Tensor(0.6117647, shape=(), dtype=float32)
- 再現率：tf.Tensor(0.3939394, shape=(), dtype=float32)
- 正解率：tf.Tensor(0.44607842, shape=(), dtype=float32)

3.3 実験 3

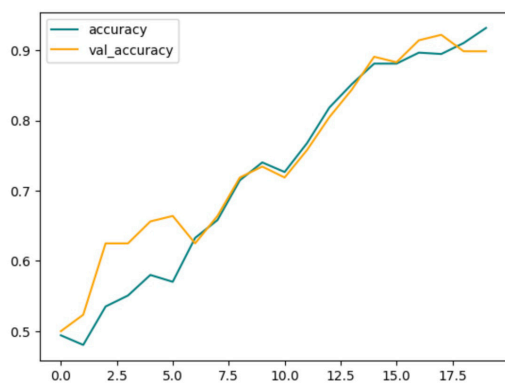


図 5 実験 3 Accuracy

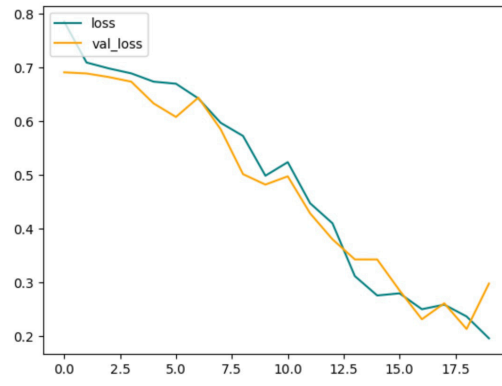


図 6 実験 3 Loss

3.3.1 学習モデルの評価

- 適合率：tf.Tensor(0.972973, shape=(), dtype=float32)
- 再現率：tf.Tensor(0.8181818, shape=(), dtype=float32)
- 正解率：tf.Tensor(0.9052632, shape=(), dtype=float32)

3.3.2 Indian-Snakes-Dataset による精査

- 適合率：tf.Tensor(0.7126437, shape=(), dtype=float32)
- 再現率：tf.Tensor(0.5344828, shape=(), dtype=float32)
- 正解率：tf.Tensor(0.5611111, shape=(), dtype=float32)

3.4 実験 4

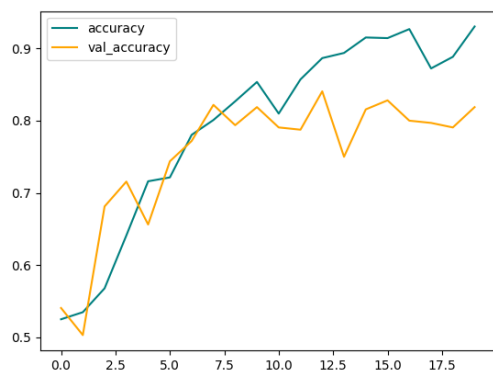


図 7 実験 4 Accuracy

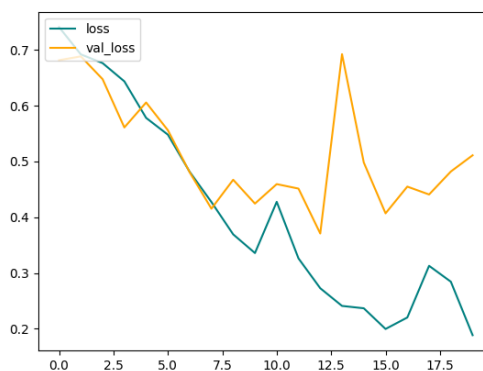


図 8 実験 4 Loss

3.4.1 学習モデルの評価

- 適合率：Precision result: 0.8222222328186035
- 再現率：Recall result: 0.9367088675498962
- 正解率：Binary Accuracy result: 0.8432835936546326

3.4.2 Indian-Snakes-Dataset による精査

- 適合率：Precision result: 0.699999988079071
- 再現率：Recall result: 0.6968609690666199
- 正解率：Binary Accuracy result: 0.6092020869255066

4 考察

5 意図していた実験計画との違い

行った実験の流れをガントチャートを用いて、表すと以下の通りになった。

タスク／期日	5/18(第5週)	5/25(第6週)	6/8(第7週)	6/15(第8週)	6/22(第9週)	6/29(第10週)	7/6(第11週)	7/13(第12週)	7/20(第13週)	7/27(第14週)
テーマ選定										
環境構築										
データセット収集										
データセット編集										
モデル作成										
実験1										
実験2										
実験3										
実験4										
ファインチューニング										
レポート作成										
発表資料作成										
提出物作成										

図9 実験の流れ

想定していた以上に環境構築で躓いてしまい、本来1週で構築する予定だったが3週ほどかかってしまった。データセットに関しては、開発期間全体を通して収集・編集をおこない、正答率向上を目指した。

6 まとめ

今回の実験では、画像に写っている蛇の毒の有無を予測し、正しいか確認することを目的として、正解率80%を目指して実施した。残念ながら、最終的な正解率は約60%と目標を達成することができなかった。しかし、実験の中で気づいた点として、「データセットの充実性」についての重要性が挙げられる。データセットをテスト用のデータに対応させるために工夫するたび、正解率が上がっていったことから、機械学習におけるデータセット内容の重要性について再認識することができた。また、実験を進めていくにつれ、機械学習の手法や開発に対する取り組み方についての理解を深めることができた。他にもGitHubを利用したバージョン管理や学科サーバーを利用した実験なども行うことができ、チーム開発として良い点だったと考える。

参考文献

- [1] 画像認識でよく聞く「CNN」とは？仕組みや特徴を1から解説, https://aismiley.co.jp/ai_news/cnn/, 2023/06/14.

[2] 画像認識の分野では欠かせない「CNN（畳み込みネットワーク）とは」, <https://www.paloaltoinsight.com/2022/12/09/cnn/>, 2023/06/14.