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

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。

3　使用するデータの前処理

以下のようにして、機械学習モデルの正確性を上げた。

3.1　データセットにおける前処理

毎実験ごとに画像データを一枚一枚目でチェックし、クレンジングを行った。具体的には、他の動物が写っていたり、体の一部が欠けている蛇、アルビノなどノイズになりそうな画像などを各々分担しながら消去し、実験に用いた。

3.2　データの正規化

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。具体的には、0~255 の数字の集合である画像データを 0~1 の範囲に縮小することで、学習コストの削減を試みた。

3.3　データ分割

ホールドアウト法を用いて、以下のような割合でデータを分割した。

Train :70% Val:20% Test:10%

4 実験 1

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。

4.2 モデル選定

本実験では画像から毒あり毒無し蛇の特徴量を抽出してもらった上で予測を行うため、畳み込みやプーリングを行うことができる CNN をアルゴリズムとして採用した。最後の出力としては 0 か１を出力させるのが目的なので、sigmoid 関数を使う。

学習モデルの作成は、ImageClassification[3] を参考に行なった。

4.3 使用するデータセット

Kaggle で取得したデータセットの classification snake species を利用する。

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

4.4 パラメータ調整

実験 1 ではパラメータを以下のように設定した。

Listing 1 パラメータ (実験 1)

```
model = Sequential()
model.add(Conv2D(16, (3,3), 1, activation='relu', input_shape=(256,256,3)))
#first layer needs to have input
model.add(MaxPooling2D())#take maximum value from 2x2 filter,only condence

model.add(Conv2D(32, (3,3), 1, activation='relu'))
model.add(MaxPooling2D())

model.add(Conv2D(16, (3,3), 1, activation='relu'))
model.add(MaxPooling2D())

model.add(Flatten())#make it to 1D

```

```
#fully connected layers
#256neurons
model.add(Dense(256, activation='relu'))#going to have 256 values as our
    output
model.add(Dense(1, activation='sigmoid'))
#single output & map to 0 or1
```

以下は summary の結果である。

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 254, 254, 16)      448

 max_pooling2d (MaxPooling2  (None, 127, 127, 16)      0
 D)

 conv2d_1 (Conv2D)           (None, 125, 125, 32)      4640

 max_pooling2d_1 (MaxPoolin  (None, 62, 62, 32)        0
 g2D)

 conv2d_2 (Conv2D)           (None, 60, 60, 16)        4624

 max_pooling2d_2 (MaxPoolin  (None, 30, 30, 16)        0
 g2D)

 flatten (Flatten)           (None, 14400)             0

 dense (Dense)               (None, 256)               3686656

 dense_1 (Dense)             (None, 1)                 257

=================================================================
Total params: 3696625 (14.10 MB)
Trainable params: 3696625 (14.10 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

4.5 実験結果

4.5.1 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)

4.6 考察

うまくいかなかった原因として、データセットと学習モデル両方に問題があったと考えられる。データセットは、kaggle で取得したデータセットを利用したが、体の一部が無い絶命状態の蛇や、

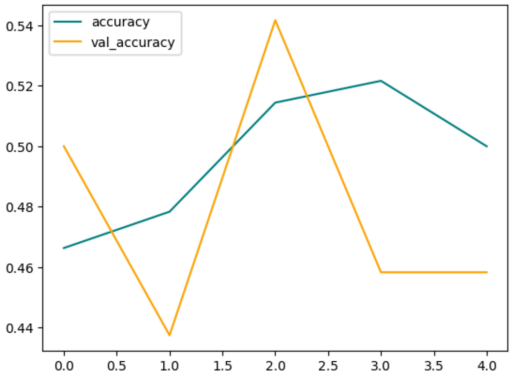

図 1　実験 1 Accuracy

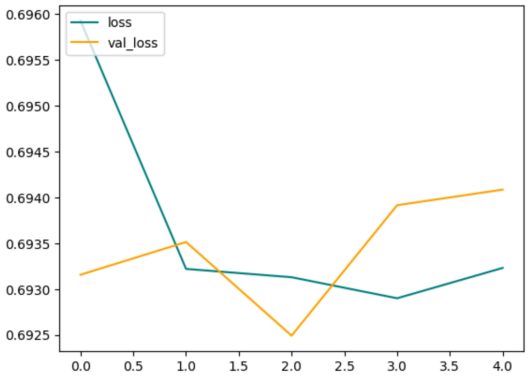

図 2　実験 1 Loss

人間の写真など投稿者のプライベート画像が大量に混入しており、信頼性のないものであった。学習モデルは、学習反復回数が 5 回のみで非常に少なかったことが挙げられる。また、過学習対策を行わなかったことも原因の一つとして挙げられる。他にも、蛇の画像を識別するのに、人間の表情を見分ける学習モデルを用いたことも原因として挙げられる。

実験 1 から学んだ改善点として、データセットに関しては、より適切なデータセットを探した上で、データのクレンジングを行うことが挙げられる。また、毒蛇と無毒蛇を種類ごとに調べることで、データセットの正確性を高められると考えた。学習に関しては、学習回数が足りてないように感じたため、学習反復回数を 5 回から 20 回に増やすことが挙げられた。また、VGG16 のような汎用的な学習済みモデルにファインチューニングを施すことや、過学習による汎化性能の低下の対策として、NN モデルに dropout layer を追加することが挙げられた。

5　実験 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。

5.2　モデル選定

実験 1 と同様に CNN を用いるが、すでに事前学習されている VGG16 の全結合層以外のレイヤーにオリジナルで layer を追加し、追加分レイヤーだけをファインチューニングして利用する。

オリジナルで追加した layer の特徴としてはロバストネスのための Dropout layer を用いている。

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。

評価用データセット

- Indian-Snakes-Dataset(以下 indian datasets と呼ぶ)…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`)

5.4 パラメータ調整

実験 2 ではパラメータを以下のように設定した。

Listing 2　パラメータ (実験 2)

```
model = Sequential()

model.add(base_model)

model.add(Conv2D(64, kernel_size=3, padding="same", activation="relu"))
model.add(MaxPooling2D())
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(128, activation="relu"))
model.add(Dense(64, activation="relu"))
model.add(Dropout(0.5))

model.add(Dense(1, activation='sigmoid'))
```

以下は summary の結果である。

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 8, 8, 512)         14714688

 conv2d_1 (Conv2D)           (None, 8, 8, 64)          294976
```

```
 max_pooling2d_1 (MaxPoolin  (None, 4, 4, 64)          0
 g2D)
 dropout_3 (Dropout)         (None, 4, 4, 64)          0
 flatten_1 (Flatten)         (None, 1024)              0
 dense_4 (Dense)             (None, 256)               262400
 dropout_4 (Dropout)         (None, 256)               0
 dense_5 (Dense)             (None, 128)               32896
 dense_6 (Dense)             (None, 64)                8256
 dropout_5 (Dropout)         (None, 64)                0
 dense_7 (Dense)             (None, 1)                 65
=================================================================
Total params: 15313281 (58.42 MB)
Trainable params: 598593 (2.28 MB)
Non-trainable params: 14714688 (56.13 MB)
_________________________________________________________________
```

実験の際には、VGG16 の全結合層以外のレイヤーを用い、その下に layer を追加した。これらの重みに関しては、VGG16 モデルの重みを ImageNet データセットから事前に学習されたものに設定された状態のままにしておくため、これの重みは更新しないこととする。オリジナルで追加した layer にのみ重みを更新することとする。

5.5 実験結果

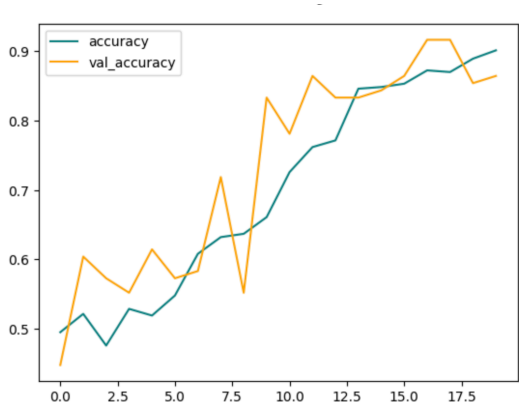

図 3　実験 2 Accuracy

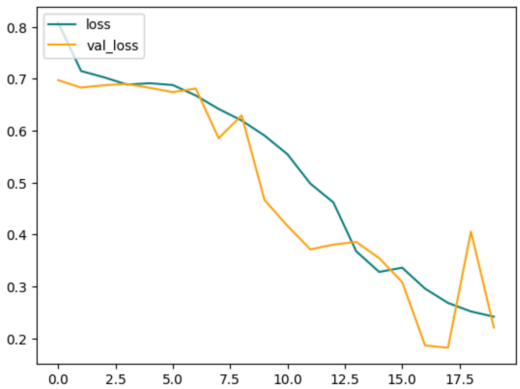

図 4　実験 2 Loss

5.5.1 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)

5.5.2 indian datasets 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)

5.6 考察

実験 2 で用いた訓練用データセットの蛇の種類は、主に日本の蛇が中心だったため、indian datasets の特徴量と合致していなかったと考えられる。元々、毒蛇共通の特徴があると仮定して実験を行なっていたが、どちらにせよ外国の蛇のデータは必要だと考えた。

また、今回用いたデータセットに世界各国の毒蛇と無毒蛇の画像をそれぞれ追加することが必要だと考えた。膨大な種類の画像が必要になるため、今後の開発ではデータセットを作成する人員を増やして実験を進めていくべきだと考えられる。

6 実験 3

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 datasets を用いて精査を行う。

6.2 モデル選定

実験 2 で用いた学習モデルをそのまま使用する。

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。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 datasets …実験 2 同様のデータセット。

6.4 パラメータ調整

実験 2 と同様に行う。

6.5 実験結果

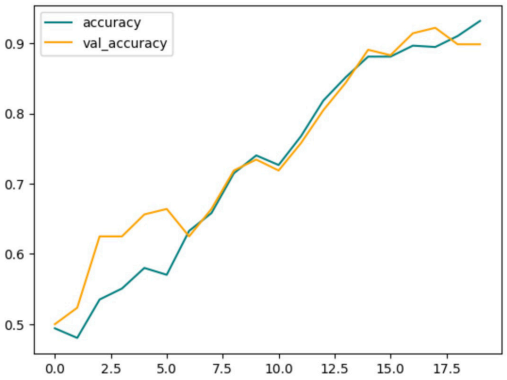

図 5　実験 3 Accuracy

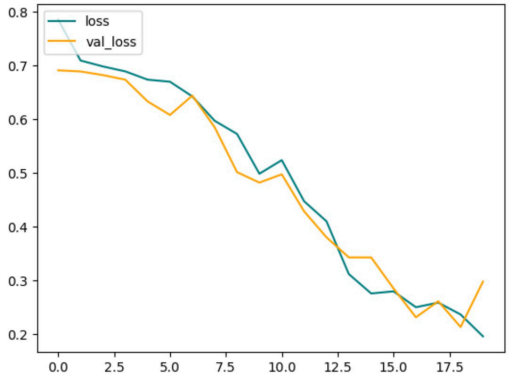

図 6　実験 3 Loss

6.5.1 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)

6.5.2 indian datasets 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)

6.6 考察

画像を追加したデータセットの内容は一種ごとで 10 倍の差があったため、外国の蛇に対する特徴量は導きにくいと予想した。一方、精査結果のスコアが向上していたため、さらにデータセットを充実させれば目標に達する可能性があると考えられる。

そこで、データセットに追加する分の画像を蛇一種あたり 30 枚ほどに増加させることで、正解

率向上を目指せるのではないかと考えた。なお、追加する蛇の種類はすでに確定しており、実験 2 での画像追加より作業量は多くならないと見込まれるため、データセットを作成する人員を減らして進める。

7　実験 4

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 datasets を用いて精査を行う。

7.2　モデル選定

実験 2 で用いた学習モデルをそのまま使用する。

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。

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

評価用データセット

- indian datasets …実験 2 同様のデータセット。

7.4　パラメータ調整

実験 2 と同様に行う。

7.5　実験結果

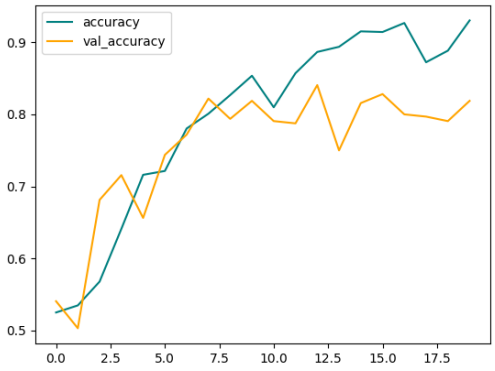

図 7　実験 4 Accuracy

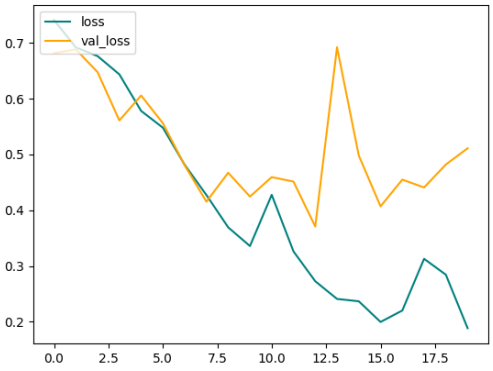

図 8　実験 4 Loss

7.5.1　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

7.5.2 indian datasets 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

7.6 考察

うまくいかなかった原因として、Train,Val,Test データにそれぞれデータの偏りがあったことが挙げられる。また、データセットの画像が多すぎたことから、ノイズを完全に取り除けていなかった可能性があった。

改善点として、交差検証を行うことが挙げられる。本実験は、ホールドアウト法で行なっていたため、データの偏りを考慮していなかった。また、説明変数を減らして汎化性能を高めるために、正則化の実装が挙げられる。[4]

8　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。

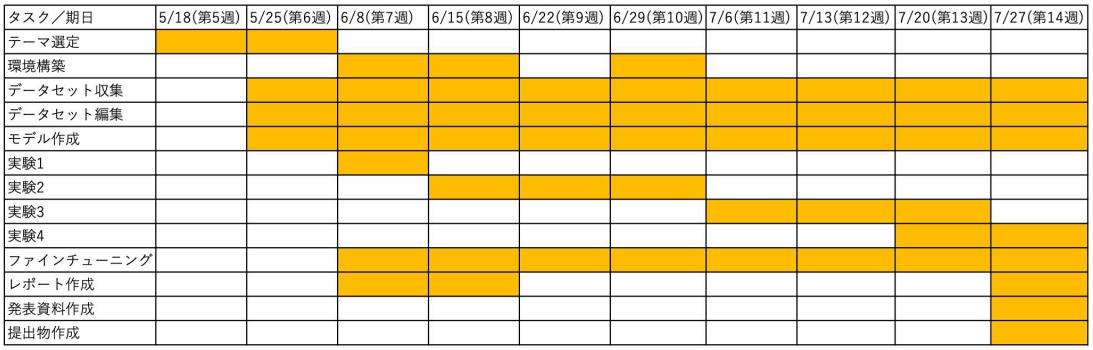

図 9　実験の流れ

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。

9　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.

[3] ImageClassification, https://github.com/nicknochnack/ImageClassification, 2023/06/08

[4] NRI 野村総合研究所 用語解説 過学習 https://www.nri.com/jp/knowledge/glossary/lst/ka/overfitting,2023/08/03