# 知能情報実験 III（データマイニング班）
# CNN を用いた画像識別
# ～イーブイフレンズ識別器～

215745E 225710G 225739E 225726C

提出日：2024 年 8 月 1 日

# 目次

**概要**

本実験の最終目標はポケモンに登場するイーブイフレンズを機械学習を通じて正確に画像識別を行うことである。具体的には、各ポケモンのデータセットを作成し、データセットから学習をするプログラム、次に学習済みモデルにより画像識別を行うプログラムを作成する。その後正確さを向上 させるためのパラメータ調整やデータセット内の画像の圧縮、選別を行なった。本実験ではCNN (Convolutional Neural Network: 畳み込みニューラルネットワーク) を用いた画像識別を行なっているが、データセットを手動で作成しており、データ数が少ないため識別の質に不安があった。そこで今回はこれまでの実験の中で識別の質を高めるために行ったことを示す。

本実験では、学習モデルとして、自作モデル、AlexNet モデル、VGG16 モデル、VGG-16 モデル (fine trained) を用いた。VGG-16 モデル (fine trained) を除いた 3 つのモデルは、過学習の傾向などが見られ、期待していた精度には至らなかったが、VGG-16 モデル (fine trained) は転移学習により、大幅な精度向上が見られた。

この実験を通して、CNN における画像識別において、データセットの質と量が機械学習モデルの性能に大きく影響するということ、データの前処理とデータ拡張技術の重要だということが明らかとなった。

# 1 はじめに

## 1.1 実験の目的と達成目標

今回の実験の最終目標はポケモンに登場するイーブイフレンズを機械学習を通じて正確に画像識別を行うことである。

そのためにまずは各ポケモンのデータセットを作成 (1) し、データセットから学習をするプログラム、次に学習済みモデルにより画像識別を行うプログラムを作成する (2)。その後正確さを向上させるためのパラメータ調整 (3)、またポケモンには色違いも存在するためそのデータセット作成 (4) や識別 (5) までできると良い。

本実験ではグループで中間報告までに上記の (1)(2) までを行うことを目標に作業を行い、その後 (3) に取り組む。(4)(5) は作業の中で難しいと判断し、正確さに重きを置いて実験をするため取り組めるとベターほどに考えながら行った。

### 1.2　イーブイフレンズ識別器とは

本グループでは画像認識によって「ポケモン」の「イーブイフレンズ」を各種認識することを対象問題として設定した。「イーブイフレンズ」とは、イーブイとその進化形を全種をまとめたものを言う。「イーブイズ」、「ブイズ（非公式用語）」と呼ばれることもある [1]。今回画像認識を学ぶことで、機械学習を実践的に学び、他の技術に応用することに寄与することを目的とするため、今回の画像認識では、比較的分かりやすく、自分たちに身近な世界的コンテンツ「ポケモン」から「イーブイフレンズ」をピックアップして画像認識を行う。

## 2　実験方法

最終的にイーブイフレンズを識別するためには、大きく分けてデータセットの構築、転移学習によるモデルの作成、各種パラメータ調整の 3 つの手順で達成できる。ここでは、

1. 実験目的
2. 実験計画
3. データセット構築
4. モデルの選定
5. パラメータ調整

の順に細かい内容を示す。

### 2.1　実験目的

今回の実験で明らかにすることの最優先事項は、「イーブイフレンズは画像識別可能か」である。元々がイラストで輪郭がはっきりしていることもあり、写真などに比べて識別がしやすいと考えられるが、似ているポケモンたちを選択しているため実際に検証する必要がある。

また、追加で確認したい事項として、「データセット内の学習データの質による識別の正確さ」、「識別の正確さを上げる方法としてどれが有効か」、「公式の画像のほかに創作のイラストや色違いなども識別できるか」がある。

「データセット内の学習データの質による識別の正確さ」に関して、今回の実験のデータセットは手動で各ポケモンの画像を 100 枚集めているため、学習データの質によっても識別の正確さが変わると考えられる。よって実験の中でデータセットの中を適宜編集して正確さに変化があるか調べる。

「識別の正確さを上げる方法としてどれが有効か」に関して、上記もこれに相当し、「学習データを増やす」、「プログラムを見直す」などの方法が挙げられるが、どの方法が識別の正確さを上げる方法として有効かを調べる。

「公式の画像のほかに創作のイラストや色違いなども識別できるか」に関して、ポケモンは公式

の画像の他にポケモンファンが創作したイラストが多くある。それらをデータセットに入れて他のイラストも識別できるか調べる。また、イーブイフレンズの色違いはとても判別が難しい (例、イーブイの色違いは色が全体的に白っぽくなっているだけ、エーフィの色違いはリーフィアと同じ緑色など) ため、識別に挑戦をしたいと考えている。

## 2.2 データセット構築

今回の実験では、イーブイとその進化系であるブースター、シャワーズ、サンダース、エーフィ、ブラッキー、リーフィア、グレイシア、ニンフィアの計９匹のデータセットを作成する。今回使用したデータセットは手作業で画像保存を行い、約 100 枚の画像を９匹それぞれファイルに分けて作成した。ファイル名はそれぞれイーブイフレンズの英語名である “eevee”、“flareon”、“vaporeon”、“jolteon”、“espeon”、“umbreon”、“leafeon”、“glaceon”、“sylvenon” としている。

画像保存に関して、Google から各ポケモンを調べ、画像をワンクリックで保存できる Google の拡張機能を用いて目視でデータに適した画像を確認し保存を行った。検索欄１つにつき１ファイルが作られる仕組みなので、画像保存の時にはすでに各ポケモンのファイルができている。

実験の中でデータセットの質を改善する必要が出てきたため、後に重複している画像やデータとして適さない画像を削除、その後画像のサイズを統一するために圧縮、回転した画像を加えかさ増しをして 100 枚のデータにした。この過程はモデルの中に含まれている。

## 2.3 モデル選定

本実験のモデルについて、はじめに自作のモデルを作成し学習させた。今回の実験の目的は CNN について学ぶことを目的にしているため、CNN における有名な構造を元に自分たちでモデルを実装した [5]。このモデルでの学習と実験初期の段階のデータセットでは識別の質が悪く、データセットの改善によっても識別の質が改善しなかったため、既存のモデルを参照することにした。

ここで目についたのは VGG16 である。VGG16 というのは,「ImageNet」と呼ばれる大規模画像データセットで学習された 16 層からなる CNN モデルであり、Oxford 大学の研究グループが提案し 2014 年の ILSVR で好成績を収めてモデルである [2][6]。このモデルなら良い結果が得られそうだと考え、これを元に実験を再開。しかし結果を示すグラフがおかしな値を示していた。調べた結果おそらくデータの少なさによるものだと考えられる。

さらに他のモデルを探して、AlexNet を試してみることに。AlexNet は画像認識の世界的な大会である ISLVRC で登場し、AlexNet が登場する前年は優勝したモデルの画像認識の誤差が 26% でしたが、2012 年に登場した AlexNet は誤差 15.3% を叩き出し、前年の記録を大幅に更新したモデルである [3]。AlexNet を使用しても若干の改善は見られたが精度は悪いままだった。このことからデータセットの質と量の少なさに原因があると結論づけた。

ここで少ないデータでも高い精度で識別できる転移学習の存在を知り、利用することに。転移学習とは、別のタスクで学習された知識を別の領域の学習に適用させる技術である [4]。

この技術をもとに以前使用した VGG16 を転移学習で用いて実験を行った結果、大幅に識別の質が改善された。

最終的に今回の実験で用いたのは、自作モデル、VGG16、AlexNet、VGG16 を用いた転移学習の 4 つである。

## 2.4 パラメータ調整

### 2.4.1 自作モデル

- 畳み込み層フィルター数およびサイズ：
  - 畳み込み層 1：32, 3x3
  - 畳み込み層 2：64, 3x3
  - 畳み込み層 3：128, 3x3

  32、64、128、... というフィルター数の設定は、モデルの学習能力を最大化しながら、計算コストとメモリ使用量を抑えるためのバランスの取れたアプローチである。
  また、3x3 というフィルターサイズの設定により、小さすぎず、大きすぎないサイズで、計算量が抑えられ、モデルの学習が効率的に行える。
- 活性化関数：relu（畳み込み層および全結合層）、softmax（出力層）
  それぞれ、さまざまな実績のあるニューラルネットワークで採用されていたため、このように設定した。
- プーリング層プーリングサイズおよびストライド：
  - プーリング層 1〜3：プーリングサイズ 2x2, ストライド 2x2

  2x2 のプーリングサイズは、特徴マップのサイズを効果的に圧縮しながらも、重要な情報を保持するのに適しているサイズ。
  また、ストライドをプーリングサイズを同じ 2x2 にすることで、プーリング操作が重ならずに実行できる。
- 全結合層のユニット数：512
  多すぎず少なすぎず、適度な複雑さを持つ全結合層を構成し、過学習を防ぎながら十分な学習能力を発揮できる。
- 最適化アルゴリズム：adam
  活性化関数と同様にさまざまな実績のあるニューラルネットワークで採用されていたため、このように設定した。
- バッチサイズ：32
  32 に設定した理由は、32 のバッチサイズは一般的な GPU で効率的に扱えるサイズであり、バッチサイズを 32 にすることは、様々な問題設定において適切であることが今までの実績から証明されているためである。
- エポック数：30
  初めは比較的少ない数として、エポック数 10 で実験を行った。次に、エポック数 30 でも

実験を行ったが、結果に違いが見られず、また、これ以上エポック数を増やしても過学習になってしまうと考えたため、エポック数を 30 とした。

- 学習率：0.001
  0.001 は adam アルゴリズムのデフォルト学習率である。

### 2.4.2 VGG-16

VGG-16 の構造を真似したため、調整したパラメータは、エポック数、パッチサイズ、学習率のみ。

- 畳み込み層フィルター数およびサイズ：
  - 畳み込み層 1：32, 3x3
  - 畳み込み層 2：64, 3x3
  - 畳み込み層 3：128, 3x3
  - 畳み込み層 4：256, 3x3
  - 畳み込み層 5：512, 3x3
- 活性化関数：relu（畳み込み層および全結合層）、softmax（出力層）
- プーリング層プーリングサイズおよびストライド：
  - プーリング層 1〜5：プーリングサイズ 2x2, ストライド 2x2
- 全結合層のユニット数
  - 全結合層 1,2：4096
- 最適化アルゴリズム：adam
  Adam は学習率を自動的に調整することができ、学習率を手動で調整する必要が少なくすることができる。また、Adam は現代の深層学習モデルで広く使用されており、その安定性と効率性が実証されている。
- エポック数：50
  epoch を 50 にした理由は、epoch 数を 50 にした時、VGG-16 モデルのグラフの精度データが明らかにおかしい様子を見せていたが、その理由はがデータの数不足にあると思い、これ以上 epoch 数を調節しても意味がないと判断したためである。
- パッチサイズ：32
  32 に設定した理由は、32 のバッチサイズは一般的な GPU で効率的に扱えるサイズであり、バッチサイズを 32 にすることは、様々な問題設定において適切であることが今までの実績から証明されているためである。
- 学習率：0.001
  0.001 は adam アルゴリズムのデフォルト学習率である。

### 2.4.3 AlexNet

基本的な構造は AlexNet を真似て重要なパラメータを変えた。

- エポック数: 100
  VGG-16 を 30 の epoch で実行させた理由と同じく、epoch=100 の時、データ数の数が原因で epoch 数の調整はもはやモデルの精度を上げられないと判断したためである。
- パッチサイズ: 32
  32 に設定した理由は、32 のバッチサイズは一般的な GPU で効率的に扱えるサイズであり、バッチサイズを 32 にすることは、様々な問題設定において適切であることが今までの実績から証明されているためである。
- 最適化アルゴリズム：adam
  Adam は学習率を自動的に調整することができ、学習率を手動で調整する必要が少なくすることができる。また、Adam は現代の深層学習モデルで広く使用されており、その安定性と効率性が実証されている。
- 入力する画像のサイズ: 224x224 ピクセル
- 学習率: 0.001
  0.001 は adam アルゴリズムのデフォルト学習率である。

### 2.4.4 VGG-16 (fine-trained)

VGG-16 (fine-trained) 学習機では、VGG-16 から最後の全結合層を取り除いて次の層を付け加えた。

Flatten(): VGG-16 の出力を 1 次元にフラット化します。これにより、全結合層に接続する準備が整う。

Dense (512, activation='relu'): 512 個のニューロンを持つ全結合層を追加する。
活性化関数として ReLU が使用する。

Dropout (0.5): ドロップアウト層を追加する。これにより、学習時においてランダムに 50% のニューロンを無効化し、過学習を防ぐ。

Dense (9, activation='softmax'): 最終的な出力層です。9 個のクラスを分類するために、softmax 活性化関数を使用して確率分布を生成する。

設定したパラメータ

- エポック数：16
  epoch 数が 18 を超えたとき、学習の後半に accuracy が高くなり、val_accuracy が低くなる過学習の様子が見え、epoch はそれより低く設定するべきだと思った。
- パッチサイズ：32 32 に設定した理由は、32 のバッチサイズは一般的な GPU で効率的に扱

えるサイズであり、バッチサイズを 32 にすることは、様々な問題設定において適切であることが今までの実績から証明されているためである。

- 最適化アルゴリズム：adam
  Adam は学習率を自動的に調整することができ、学習率を手動で調整する必要が少なくすることができる。また、Adam は現代の深層学習モデルで広く使用されており、その安定性と効率性が実証されている。
- 学習率：0.001 0.001 は adam アルゴリズムのデフォルト学習率である。

## 3　実験結果

実験結果について、以下の通りである。

１～５の図は、自作の機械学習モデルの訓練過程における損失関数（Loss）と精度（Accuracy）の変化を示している。左側のグラフは訓練と検証の損失を、右側のグラフは訓練と検証の精度をそれぞれ表している。また、図 2、図 3 における縦軸上限は 1.0 ではないので図を見る際には注意が必要である。

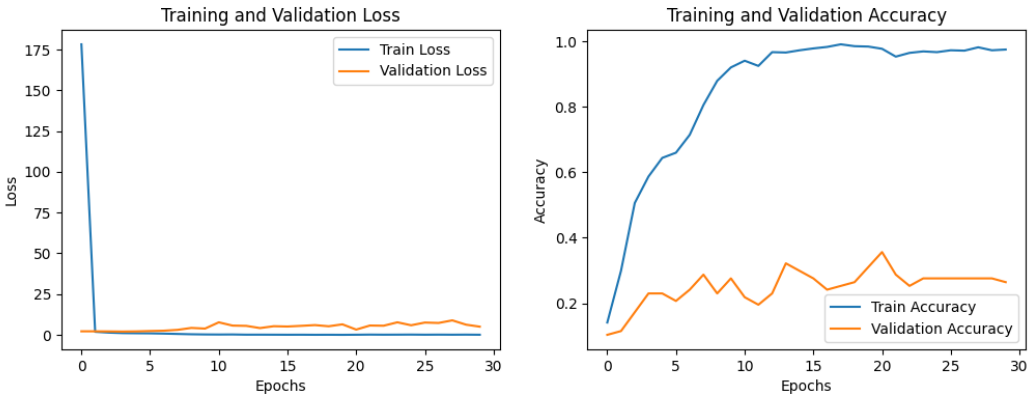


図 1　自作モデルの実行結果（エポック 30）

まず、図 1 は初期エポック（0 から 5 エポック）で訓練損失（Train Loss）が急激に減少しているのがわかる。これはモデルが急速に学習していることを示している。そして、検証損失（Validation Loss）は比較的一定で、特にエポック 15 以降、若干の増加傾向が見られる。これは過学習である可能性がある。

訓練精度（Train Accuracy）はエポック 5 まで急激に上昇し、その後ほぼ 0.9（90%）の高い精度で安定している。また、検証精度（Validation Accuracy）はエポック 5 まで上昇した後、約 0.4（40%）で変動しているが、大きく向上していない。これは、モデルが訓練データに対しては高い精度を達成している一方で、検証データに対しては過学習している可能性を示している。

全体的に、図 1 から、訓練データに対して高い精度と低い損失を達成している一方で、検証データに対しては精度が低く、損失がエポックの進行とともに増加しているため、過学習が発生してい

る可能性があることがわかる。

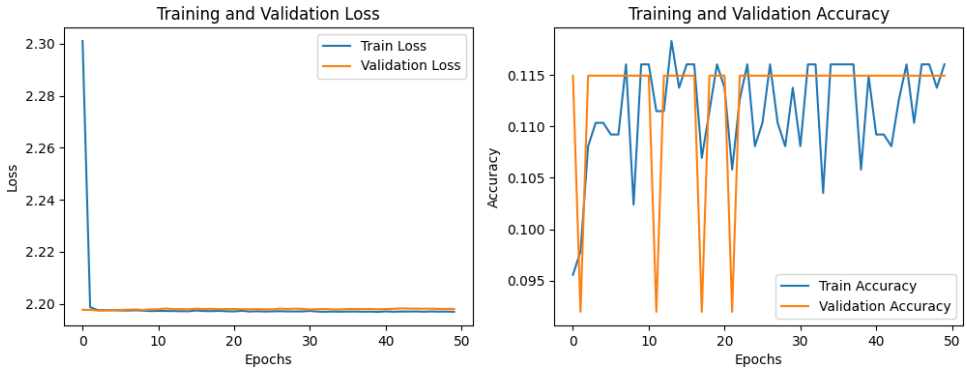


図 2　VGG-16 モデルの実行結果（エポック 50）

図 2 は、VGG16 モデルの訓練過程における損失関数（Loss）と精度（Accuracy）の変化を示している。初期エポック（0 から 5 エポック）で訓練損失（Train Loss）が急激に減少しているが、その後ほぼ一定の値で安定している。検証損失（Validation Loss）も同様に、初期エポックで急激に減少し、その後ほぼ一定の値で安定している。訓練損失と検証損失の値が非常に近いことから、過学習の兆候は見られないことがわかる。

また、訓練精度（Train Accuracy）と検証精度（Validation Accuracy）は非常に低く、エポックを通じて大きな変動が見られている。特に検証精度は頻繁に変動し、安定していないことがわかる。

訓練精度と検証精度が非常に低いことから、モデルがデータセットを十分に学習できていない可能性がある。これはモデルのアーキテクチャやハイパーパラメータ設定、あるいはデータセット自体に問題があると思われる。

図 2 から、VGG-16 モデルの訓練と検証の精度が低く、損失が安定しているにもかかわらず精度が向上しないことから、モデルがデータセットを適切に学習できていないことが示唆されている。


図 3　Alexnet モデルの実行結果（エポック 100）

図 3 は、AlexNet モデルの訓練過程における損失関数（Loss）と精度（Accuracy）の変化を示している。初期エポック（0 から 5 エポック）で訓練損失（Train Loss）が急激に減少しているが、その後ほぼ一定の値で安定している。検証損失（Validation Loss）も同様に、初期エポックで急激に減少し、その後ほぼ一定の値で安定している。訓練損失と検証損失の値が非常に近いことから、過学習の兆候は見られないことがわかる。

また、訓練精度（Train Accuracy）と検証精度（Validation Accuracy）は非常に低く、エポックを通じて大きな変動が見られている。特に検証精度は頻繁に変動し、安定していないことがわかる。

訓練精度と検証精度が非常に低いことから、モデルがデータセットを十分に学習できていない可能性がある。これはモデルのアーキテクチャやハイパーパラメータ設定、あるいはデータセット自体に問題があると思われる。

図 3 から、AlexNet モデルの訓練と検証の精度が低く、損失が安定しているにもかかわらず精度が向上しないことから、モデルがデータセットを適切に学習できていないことが示唆されている。

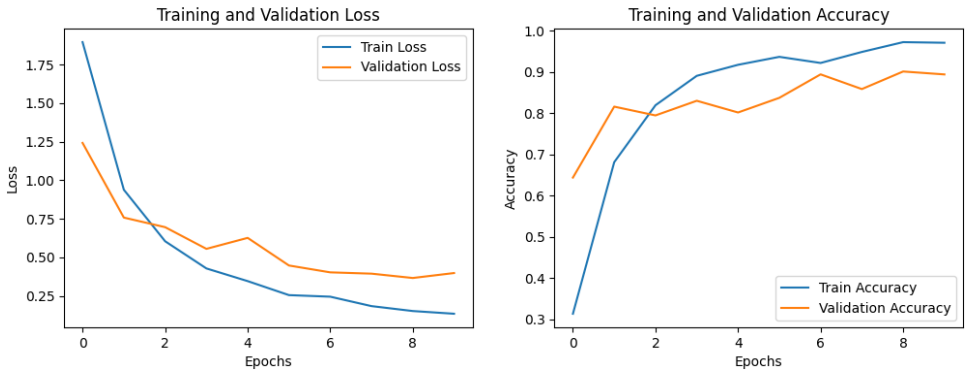


図 4　VGG-16 モデル (fine trained) の実行結果（エポック 10）

図 4 は VGG-16 モデルに転移学習をさせて作ったモデル（エポック 10）の訓練過程における損失関数（Loss）と精度（Accuracy）の変化を示している。この図から、初期エポック（0 から 5 エポック）で訓練損失（Train Loss）が急激に減少しているが、その後ほぼ一定の値で安定していることがわかる。検証損失（Validation Loss）も同様に、初期エポックで急激に減少し、その後ほぼ一定の値で安定している。訓練損失と検証損失の値が非常に近いことから、過学習の兆候は見られない。

また、訓練精度（Train Accuracy）と検証精度（Validation Accuracy）は初期エポックで急激に上昇し、その後はエポック 8 以降ほぼ安定しているが、検証精度に多少の変動が見られる。訓練精度と検証精度の間に大きな乖離はなく、モデルはデータセットを適切に学習していると考えられる。

このことから、図 4 から転移学習をした VGG-16 モデル（エポック 10）は、訓練と検証の精度が安定しており、損失も安定しているため、モデルがデータセットを適切に学習していることが示

唆される。


図 5　VGG-16 モデル (fine trained) の実行結果（エポック 16）

図 5 は VGG-16 モデルに転移学習をさせて作ったモデル（エポック 16）の訓練過程における損失関数（Loss）と精度（Accuracy）の変化を示している。

この図から、初期エポック（0 から 5 エポック）で訓練損失（Train Loss）が急激に減少しているが、その後ほぼ一定の値で安定していることがわかる。検証損失（Validation Loss）も同様に、初期エポックで急激に減少し、その後ほぼ一定の値で安定している。訓練損失と検証損失の値が非常に近いことから、過学習の兆候は見られない。

また、訓練精度（Train Accuracy）と検証精度（Validation Accuracy）は初期エポックで急激に上昇し、その後はエポック 8 以降ほぼ安定しているが、検証精度に多少の変動が見られる。訓練精度と検証精度の間に大きな乖離はなく、モデルはデータセットを適切に学習していると考えられる。

このことから、図 5 から転移学習をした VGG-16 モデル（エポック 16）は、訓練と検証の精度が安定しており、損失も安定しているため、モデルがデータセットを適切に学習していることが示唆される。

これらの結果は、各モデルの識別精度を示しており、転移学習を用いた VGG16 が最も高い精度を達成したことが分かる。

# 4　考察

実験を通じて最も重要だと感じたのは、データセットの収集の重要性だ。機械学習を行う際、モデルの構築と学習が最も重要だと考えていたが、実際のところ、開発プロセスの中で最も時間がかかり、苦労した部分はデータセットの収集だった。VGG-16 や AlexNet のモデルをうまく使えなかった理由もデータセットの数の不足にあった。このことから、機械学習で一番大事なのはデータセットの収集だとわかった。

また、データセットの質も非常に重要であることがわかった。最初に自作モデルを作る際、

Google でいろいろな写真を集めて学習を行った結果、モデルの精度が非常に低かった。その理由は、イーブイが写っている写真に変な背景が入っていたり、画像が横に長かったりしたためだ。データセットの数を減らしてでも質を向上させた結果、グラフが良い方向に変化した。そのため、データセットの量が多いことだけが良いモデルを作る要因ではなく、データセットの質も非常に重要であることがわかった。

さらに、CNN で画像を分類するとき、AlexNet や VGG-16 などのモデルを使って精度を上げる方法についても学んだ。具体的には、データの前処理やデータ拡張技術を活用することで、より良い結果を得ることができた。

失敗やつまづきも多く経験したが、それらを通じて多くの学びがあった。初期のデータ収集段階で、背景ノイズやアスペクト比の不一致が多く含まれたデータを使用していたため、モデルの精度が低下した。この問題を改善するためには、データ収集の段階でのフィルタリングや前処理が重要だと感じた。また、十分なデータがない状態で複雑なモデル（VGG-16 や AlexNet）を使用したため、モデルの性能を発揮できなかった。そのため、データセットの規模や質に応じた適切なモデル選定が必要だと思い、データセットの数が確保できない環境では、転移学習などを使うこともできるということを学んだ。

今後の展望としては、より多くの高品質なデータを収集し、データセットの質を向上させることを目指す。また、データの前処理や拡張技術を駆使して、モデルの性能を最大限に引き出す工夫を行う。データセットの規模や質に応じたモデルの選定とパラメータ調整を行い、より効率的な学習プロセスを確立する。さらに、定期的にモデルの性能を評価し、改善点を見つけて次の実験に活かすサイクルを確立する。このように、実験を通して得られた知見を活かし、今後のプロジェクトにおいてより高い成果を目指していく。

## 5　意図していた実験計画との違い

今回の実験では、データセットを自作で用意したためデータセットの作成時間を多く必要としたため、効率の良いモデル選定やプログラムの改善のための気づきを得る頻度などが減ってしまっていたようにに感じる。またデータセットの量に限りがあったため転移学習が効率が良い方法という結論に至ったが、事前調査をもっと深く行っていたら転移学習にたどりつくことはもっと早く、作業や実験、考察もより深く行うことができた。　自作モデルについて全体でのすり合わせが甘くデータ内の画像の質や学習結果の考察など、改善できることは多くあったがコミュニケーション不足により改善が遅れる、または改善できない部分が生まれてしまい、手がつけられず結局既存のモデルを利用する方法にシフトしてしまったため、全体でのコミュニケーションを多くとっていれば自作モデルで画像認識の質が上がるより理想的な結果にできた可能性がある。

# 6 まとめ

私たちのグループでは既存のデータセットがなく、データセットを１から構築する必要があった。必要な画像データを集める上でどうしても量が少なく、画像を正方形に切り取ったり、背景を削除したりしてデータの質の部分を上げた。しかし、質を上げても画像データの量が足りず期待した結果が得られなかった。そのため、画像を回転、反転、拡張、縮小などのデータの水増しを行い量の部分を補った。このことから、データの質と量はどちらも機械学習モデルの性能に直結することを強く実感した。質の高いデータを構築するだけでなく、十分な量のデータを確保することが、正確で信頼性の高いモデルを作成するためには不可欠であると学んだ。今後はデータセットを構築する機会がある際には、データの質と量を向上させることにもっと時間とリソースを割いてより高性能な機械学習モデルを実現していきたいと思う。

# 参考文献


[1] イーブイフレンズとは, https://ja.wikipedia.org/wiki/ポケモンの一覧_(102-151)#イーブイ, 2024/05/23

[2] VGG16 モデルを使用してオリジナル写真の画像認識を行ってみる https://newtechnologylifestyle.net/vgg16originalpicture/,2024/06/13

[3] 画像認識 CNN の AlexNet を分かりやすく解説 https://dx-consultant-fast-evolving.com/alexnet/,2024/06/20

[4] theme4) 転移学習とは？ https://aismiley.co.jp/ai_news/transfer-learning/#:~:text=転移学習は、機械学習,することになります%E3%80%82 ,2024/06/20

[5] CNN（畳み込みニューラルネットワーク）とは？わかりやすく解説 https://www.hitachi-solutions-create.co.jp/column/technology/cnn.html,2024/06/20

[6] ImageNet https://www.image-net.org,2024/06/20