

自作の画像データを用いたCNNによるペットボトルごみの識別

逗子開成高等学校 高校1年 白井千裕 鈴木翔天



研究背景・目標

深層学習を学んでいるとき、近年ゴミの分別の問題を知る。

近年ペットボトルごみの捨て方が注目されている。実際にサントリーの会社の取り組みではリサイクルするためにペットボトルの分別を推進している。

深層学習による海洋ごみの分別の研究はあった。様々なものではなく、特定のゴミに関する画像認識の研究はなかった。



サントリーさんの取り組み

最終的な目標

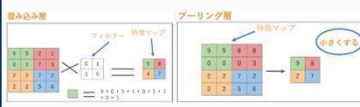
ペットボトルごみにラベル、キャップがついているかをカメラで判別する画像認識モデルを作成

人々に分別を促す



CNNとは

CNN (Convolution Neural Network)は「畳み込みニューラルネットワーク」で主に画像認識の分野で力を発揮し、「畳み込み層」、「プリーング層」、「全結合層」の3つの層で構成されている。



「畳み込み層」は入力データをフィルターを使い、画像から特徴の抽出を行う。抽出したものを特徴マップという。

「プリーング層」は抽出された特徴の空間サイズを小さくする。これにより特徴マップが平行移動しても影響を受けることがないようにしたり、データ量を減らすことができる。

「全結合層」は特徴マップを結合し一次元の数値にすることで元のデータがどういう画像なのかを確率で示す。

環境開発

Dockerを用いてtensorflow-gpuをインストールしたjupyterlab環境で行う。

Pythonを使用し理由

豊富なライブラリ「科学計算用」 + 文献が多い「エラーを対処しやすい」

NVIDIA製のGPUを用いて学習できるようにNVIDIAのCUDAやcuDNNもインストールする。GPUを用いることでCPUを用いるより格段に高速に学習することができる。左下の写真は実際に用いたGPUである。

他にもNumpy, matplotlib, pillowといったpythonのライブラリをインストールする。

- Docker...コンテナ型の仮想環境を構成、管理するソフトウェア。
- Jupyterlab...pythonなどで機械学習をする際によく用いられる開発環境。
- Tensorflow...Googleが開発した代表的な深層学習用のライブラリ。
- CUDA, cuDNN...NVIDIAが提供するGPU向けの統合開発環境及び深層学習用のライブラリ。
- Numpy...ベクトル・行列計算に特化。 Matplotlib...グラフを表示。 Pillow...画像処理に特化。

スペック... (CPU: intel Core i5 12400, GPU: GeForce RTX 4060 8GB, メモリ: 48GB)

データ拡張

<引数一覧>

引数	説明
rotation_range	ランダムに回転する回転範囲
width_shift_range	ランダムに水平移動する範囲
height_shift_range	ランダムに垂直移動する範囲
shear_range	せん断の割合
zoom_range	ランダムに拡大縮小する範囲
channel_shift_range	ランダムにRGBを足し引きする範囲
horizontal_flip	水平方向に入力をランダムに反転
vertical_flip	垂直方向に入力をランダムに反転

画像認識の機械学習には大量の画像データが必要になるが、大量の画像を集めることは難しい。そのような場合、データを増やす方法として「データ拡張」がある。これは画像の水増しであるが、単にデータをコピーするのではなく、画像を反転させたり、ずらしたりして画像のデータを増やすのである。

データ拡張の際、TensorFlowに含まれるImageDataGeneratorという機能を使う。この機能で引数を指定することで様々な方法で画像を増やすことができる。(左の表)

本研究ではテスト用データを拡張した上で、トレーニング用データを拡張したもの、していないものを、この後の「学習及び評価」のところで学習し、比較する。

一枚の写真データを拡張した様子



(左の表の濃い緑の部分は実際に使用した引数)

データセットの作成

画像認識モデルには学習するためのデータセットが必要となる。一般的にデータはネットなどから収集したり、そもそも最初からデータセットが提供されていることがある。しかし、ペットボトルごみ(ラベルやキャップがない状態)の画像はあまりないため、自分たちで画像を収集し、データセットを作成する必要がある。

まずはデータセットを作るために必要なペットボトルの画像を収集する。逗子開成高等学校のごみ捨て場で収集した。清掃員の手に頼みペットボトルを集めた本数は合計42本である。

ゴミの収集、洗浄

ラベル、キャップ分け

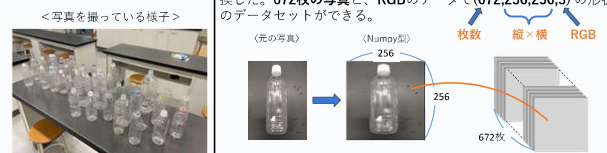
4通り

- キャップとラベルの両方ともついているもの
- キャップのみついているもの
- ラベルのみついているもの
- 全てついていないもの

一つの通りで4枚の写真を撮る。それを42本分るので合計42×4×4=672枚の写真が集まる。

Numpy型への変換

<写真を撮っている様子>

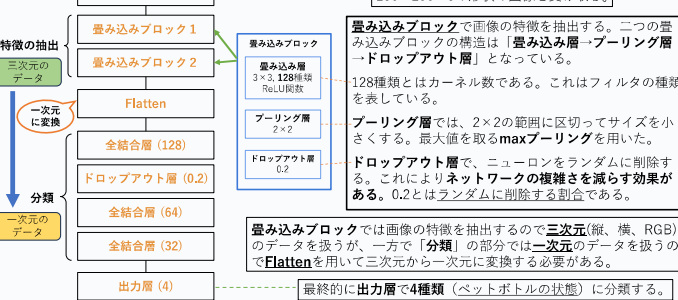


最後に672の画像データをトレーニング用とテスト用に分ける。トレーニング用はモデルに学習させるデータ、テスト用は学習したモデルをテストするためのデータである。学習したモデルによってテスト用のデータは未知のデータである。今回は604枚の画像をトレーニング用、68枚をテスト用にした。

モデルの作成

CNNを実装するモデルのネットワーク構造は左の図の通りである。

256×256×3の形状の画像を受け取る。



特徴抽出

畳み込みブロック

畳み込み層 3×3, 128核, ReLU関数

プリーング層 2×2

ドロップアウト層 0.2

全結合層 (128)

ドロップアウト層 (0.2)

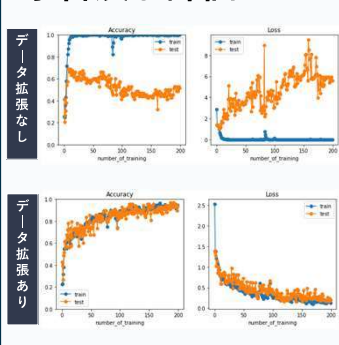
全結合層 (64)

全結合層 (32)

出力層 (4)

最終的に出力層で4種類(ペットボトルの状態)に分類する。

学習及び評価



学習

左のグラフの「Accuracy」は正解率、「Loss」は誤差

[number of training]は学習回数(エポック数)を表す。

正解率は上がるほどよく、逆に誤差は下がるほどよい。

	拡張なし	拡張あり
Accuracy	トレーニング用は最初から急上昇しその後は一定。テスト用は上昇しきれず。	共に徐々に上昇していった。
Loss	トレーニング用は収束。テスト用は収束せず。	共に徐々に収束していった。

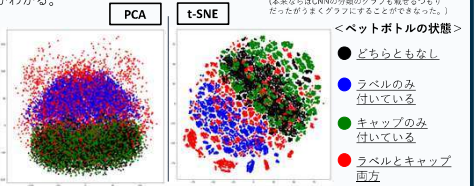
評価

評価はモデルの実際の性能を計る。ここで用いるデータは拡張されたものである。計測すると、拡張なしは45%と低いが、一方で拡張ありは97%と高い認識率である。黄色の線で示した数値が正解率を表している。

<データ拡張なし>	<データ拡張あり>
Accuracy: 0.45	Accuracy: 0.97
Loss: 0.55	Loss: 0.03

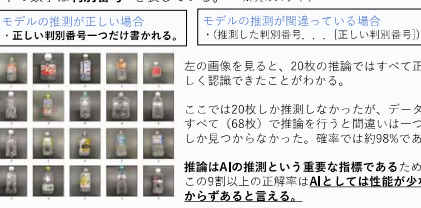
PCA, t-SNEによる分類

本研究ではCNNによる画像の識別を行ったが、ここではPCA及びt-SNEを用いて分類し、CNNと比較する。2つのグラフでは4種類の間まじりはある程度できているが、かなり重なっているため、PCA及びt-SNEにはCNNのような97%の精度の識別は難しいと言える。また緑の黒のデータは数値的に近いことがわかる。



推論

推論とはトレーニング用のデータで学習したモデルが新たなデータを推測することである。左の画像は20枚を推論した結果である。画像の下の数字は判別番号を表している。※次のスライド



モデルの推論が間違っている場合(推測した判別番号... [正しい判別番号])

正しい判別番号一つだけ書かれる。

左の画像を見ると、20枚の推論ですべて正しく認識できたことがわかる。

ここでは20枚しか推論しなかったが、データすべて(68枚)で推論を行うと間違いは一つしか見つからなかった。精度は約98%である。

推論はAIの推論という重要な指標であるため、この9割以上の正解率はAIとしては性能が少なからずあると言える。

考察

学習において、「学習及び評価」のグラフから下の表にまとめたことが言える。本研究のCNNのモデルはPCAやt-SNEによる分類よりも正確に判別することができる。データ拡張により影や少し傾いた写真にも対応できるようになった。「PCA, t-SNEによる分類」や「カメラ環境への応用」からキャップが認識しづらい事がわかる。

	拡張なし	拡張あり
トレーニング	単独なデータのみで学習するので正解率が急上昇する。	様々なデータで学習するので徐々に正解率が上昇する。
テスト	未知のテスト用データなどで対応できず、多く間違える。	拡張したデータで学習したので未知の複雑なデータでも対応できる。
例えるなら	簡単な問題集のみ	応用問題集

カメラ環境への応用

カメラを用いてペットボトルをリアルタイムで判別するモデルを作成する。カメラの画像をpythonで扱うにはcv2というライブラリを用いる。

cv2でカメラから読み込んだ画像を作成したモデルに判別させ、推測したペットボトルの状態を判別番号で表示させる。	ペットボトルの状態
0	どちらともなし
1	ラベルのみ付いている
2	キャップのみ付いている
3	ラベルとキャップ両方



<カメラでペットボトルを判別している様子>

左の画像の場合、黄色の線を見ても正しい「0」を高い確率で認識している。一方で「2」もある程度の確率で推論している。これはキャップを判別しづらいことを表している。

拡大

状態	0	1	2	3
確率(%)	97.69	0.0118	12.29	0.00442

今後の展望

- より様々な(大きさや色)データを集める。
- CNNなど人工知能のアルゴリズムをさらに理解、研究し、最適なパラメータを模索する。
- カメラで社会に応用する際に影や光が当たるといった、実際の環境に適切できるように学習させる。

今回はDockerやGPU、データ拡張など様々な機能を用いたゴミの画像認識の研究を試みた。初めて作るには正解率は良かったが社会で実際に使われるにはほどよい。

このようなことの重要性をこの研究で認識することができた。また、今回はAIを一から作ることができたので、今後は既存のものに依存せず、自ら考えオリジナルのAIを作成していきたい。

参考文献 自作のデータで画像認識モデルをつくって遊ぼう! https://qiita.com/14822322574936cd7fca25232b0c5e3f3813a15t4e580907a75e58b8b9535338... ImageDataGeneratorを使ってみた https://qiita.com/okunemi/items/14822322574936cd7fca25232b0c5e3f3813a15t4e580907a75e58b8b9535338... PCA, Kernel PCA, t-SNE, CNNによる可視化のための次元削減の比較 https://qiita.com/reshin2/items/52195988c94719315e... nvidiaGPUをDockerのTensorflowで動かす https://qiita.com/okunemi/items/52195988c94719315e... pythonで動かす! あららしい深層学習の教科書 機械学習の基礎から深層学習まで。 [AlphaZero 深層学習、強化学習、深層、人工知能プログラミング実践入門] 石川 龍彦 翔海社 2018 市川 真一 ホンテック 2019 [python3をディープラーニングのしくみ 体験してわかる会話でまなぶ] 森 均明 翔海社 2023 [TensorFlowによる深層学習入門]OpenAI Gym PyBulletによるシミュレーション 飯野 謙二 西陽館 2021