

Google Colaboratoryを用いたAIエッジデバイス 学習モデル構築に関する研究

A Study on Building AI Edge Device Learning Model Using Google Collaboratory

宮崎 英一¹ ・ 坂井 聡¹

Miyazaki Eiichi, Sakai Satoshi

要旨

本研究室では、以前からAIを用いたプログラミング教育としてエッジデバイスを用いた教材システムを開発してきた。しかし、以前に発表したRaspberry Pi 3とNeural Compute Stickを組み合わせたエッジデバイスは所定の精度で画像認識を行えたが、教育現場で実装するには予算の面でもシステム構成の手間の面でも負担が大きかった。そこで本研究では、簡便で安価なAIカメラを用いて画像認識を行ったが、クラウドで計算する標準的な学習モデルは専門的な知識が無くても利用できる反面、アクセス制限や画像認識精度に問題がある事が分かった。そこで、Google Colaboratoryを用いて自前で画像認識モデルを学習させる事でアクセス問題を回避し、認識精度を含めてAIカメラの教育現場での運用可能性を探るものである。

キーワード：AI、エッジデバイス、画像認識、プログラミング教育、Google Colaboratory

1. はじめに

現在、従来とは比べ物にならないくらい急速なスピードで技術革新が進んでいる。文部科学省の発表ⁱでも、「2030年：人体とコンピュータの融合」、「2040年：AIが人の代役となる」、「2050年：AIが人を超える」と進み、やがてSingularity（AIが人類の知能を超える技術的特異点）の世界が訪れると予想している。更に、文部科学省はSociety5.0に向けた「超スマート社会の実現に向けたデータサイエンティスト育成事業」を提案ⁱⁱし、これが現在の数理・データサイエンス・AI教育と繋がっている。これらの情報関連の事業は文部科学省だけでなく、経済産業省、総務省、金融庁と日本全体での取り組みとなっており、ICTに強い人材を育てるという事が今後の日本を左右する大きな問題である。

これと呼応するように学校教育現場においても「小学校プログラミング教育の必修化」、「中学校技術教育での双方向プログラミング」、「高校情報の必修化」、「大学入試における情報科目の新設」と大きな改革が進んでいる。

本研究室では、以前からAIの活用例として「ディープ

ラーニングを用いたモーションヒストリー解析の応用ⁱⁱⁱ」、「エッジデバイスを用いた画像認識システムの試作^{iv}」、「視覚障がい者に向けたディープラーニングを用いた物体提示システムの試作^v」等を作成してきた。しかしここで問題になったのが、エッジデバイスである。エッジデバイスは「Raspberry Pi 3」のようなシングルボードコンピューターとWEBカメラ、画像推論を行う推論エンジン（Neural Compute Stick）から構成されている。これらは所定の推論性能を有しているが、システム構成に手間がかかるという問題があった。一方、以前からこれらの機能をオールインワン化した画像認識カメラもあったが、画像推論においては、個人的な感想ではあるが今一つという感じであった。

しかし、最近の技術の進歩により高性能なニューラルネットワークプロセッサ（Kendryte K210）を実装したデバイス（M5StickV：以下AIカメラと称する）が市販されるようになってきた。この新しいデバイスで上記の「Raspberry Pi 3+WEBカメラ+ Neural Compute Stick」から構成されるエッジデバイスが置き換え可能になれば、安価で簡便な画像認識システムが実現出来ると考えた。

1 香川大学教育学部

しかし実際にテストしてみると、当方の環境では、このAIカメラにおいて標準的なタイプであるクラウドでの学習モデルの構築にはアクセス制限や画像認識精度等、いくつかの問題がある事が分かった。本研究ではこの問題点を解決するために、ローカル（自分自身のコンピュータで）学習モデルの計算を行い、上記の問題点を解決したので、これを報告するものである。

2. AIカメラ

上記で述べたAIカメラを図1に示す。AIカメラ本体の大きさは、親指程度のサイズであり、極めてコンパクトである。このデバイス単体で、認識対象モデルの撮影から認識までが可能である。同図で実行されているプログラムはBrownie^{vi}というソフトで、AIカメラを使って物体認識したものを音声で教えてくれるものである。勿論、インストールするプログラムを変更する事で、これらの機能はユーザー側で自由に変更可能である。

このAIカメラの代表的な特徴^{vii}を以下に示す。

- ・デュアルコア 64 bit RISC-V、V64HMAFDC (RV64GC) CPU / 400 MHz (ノーマル)



図1 AIカメラ (M5StickV)

- ・Dual Independent Double Precision FPU8 MiB 64 bit幅 On-Chip SRAM、ニューラルネットワークプロセッサ (KPU) / 0.8Tops (Trillion Operations per Second) である。

ここでは、演算性能が0.8Topsと表記されているが、以前に用いたNeural Compute Stickが1 Tops (ニューラルネットワーク演算専用値)、AppleのiPhoneで用いられているA11チップでは0.6 Topsなので、極端にAIカメラの演算性能が遅いというものでも無い。更に価格面ではAIカメラはカメラもついて5000円程度で購入可能であるが、Neural Compute Stickがカメラ無しで1万円と価格では2倍程度の差がある。

2.1 Vトレーニング

AIカメラで画像認識を行う手順を図2に示す。これはAIカメラの開発元であるM5STACK社の公式ドキュメント^{viii}であり、通常は同図で示す手順で開発を行う。同社では、この開発手順をVトレーニングと称しているが、以下にこのVトレーニングの詳細を説明する。

1) Burner Firmware

最初にAIカメラのOSともいべきファームウェアを書き込む必要がある。本研究では、このファームウェアはメーカーが提供する最新版 (M5StickV_Firmware_v5.1.2) を用いた。

2) Insert Card

AIカメラにマイクロSDカードを装着する。このマイクロSDカードには、AIカメラを用いて画像認識に用いる撮影画像や認識プログラムを保存する。注意点としては、AIカメラがアクセス出来るマイクロSDカードの制限があるので、動作が確認されているマイクロSDカードを使用する事が望ましい。そうしないと頻繁にアクセスエラーが発生する可能性がある。

3) Shoot

ユーザーがこのAIカメラで画像認識したい物体を撮影する。1物体につき最低で35枚以上と表記されているが、最低でも50枚程度撮影した方が精度の点から望ましい。

なお撮影においては対象物体のみを撮影するように注意が必要である。他の物体が同時に撮影フレーム内に収まっていたりすると認識精度が低下する。また認識精度を向上させるためには、物体の背景が白一色にする等の



図2 Vトレーニングの流れ

工夫が必要である。本研究では3つの物体（人形³⁶、ハサミ、ボードコンピューター）を画像認識対象とした。この中で実際にAIカメラで撮影した「ハサミ」の写真を図3に示す。この写真を納めたフォルダ以外に他の2つの物体の写真を納めた合計3つのフォルダが画像認識に用いられるデータとなる。

4) ZIP & Upload

上記で撮影した画像をマイクロSDカードからコンピュータにコピーする。コピーされた画像をコンピュータで1つのファイルに圧縮し、Vトレーニングのサーバにアップロードする。

アップロードが完了すれば、サーバ側にタスクが登録されてサーバ側で学習モデルの構築が行われる。学習モデルの構築が終われば、メールで学習モデル計算完了のメールが届く。メールには学習モデルのURLが記載されているので、このURLを介して学習モデルをコンピュータにダウンロードする。

ダウンロードされたファイルには学習モデル以外に、pythonで記載された画像判別プログラムが同封されている。このプログラムとダウンロードされた学習モデルを利用して画像認識を行う事が出来る。この学習モデルとプログラム（具体的には、判定プログラム：boot.py、判定タグを表記する：labels.txt、学習モデル：model.kmodel）をマイクロSDカードに保存する。

5) Reset & Run

プログラム等を保存したマイクロSDカードをAIカメラに装着した後、リセットをかける。そうすると画像認識プログラムが自動的に実行され、認識されたタグと精度が表示される。

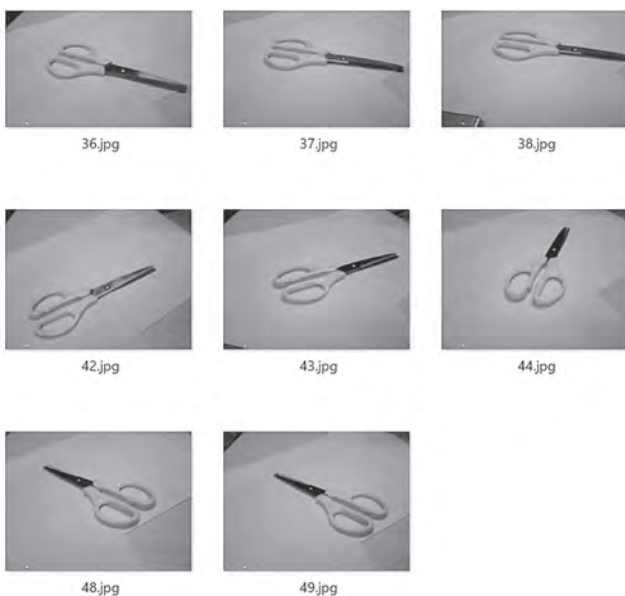


図3 撮影された画像モデル

このような手順でVトレーニングを行った後に、画像認識が可能になる。ここでの学習モデルの計算はローカルのコンピュータでは行われず、クラウド側のコンピュータで実行されるので、非力なAIカメラでも問題なく画像認識が実装できている。

しかしその反面、何らかの問題でクラウド側のサーバにアクセス出来なければ、画像認識モデルを計算する事が出来ない。実際、大学のネットワークを経由してはセキュリティ面からVトレーニングのサーバにアクセスが出来なかった。そこで本研究では、このVトレーニングのサーバ側で行われる学習モデルの計算をGoogle Colaboratoryを用いて、ユーザ側で行い、上記の問題点を解決するだけでなく、計算のカスタマイズを行い、学習モデルの特性を評価するものである。

3. Google Colaboratoryを用いた計算

実際、Vトレーニングに関する問題（アクセス不可、タスクが途中で止まる）についてはネットワーク上で散見された。そのため、いくつかのサイトで、独自に計算を行うプログラムが掲載^{37, 38}されている。現在、最新と思われるサイトを参考にして、計算を行ったが、元のソースコードのままでは、計算途中にいくつかのエラーが発生して、学習モデルの構築まで完了出来なかった。これはプログラム環境におけるバージョン問題が原因であり、機械学習を行う際にはよく起こるトラブルである。例えば機械学習によく使われるTensorFlow³⁹（Googleが開発しオープンソースで公開している、機械学習用のソフトウェアライブラリ）はバージョンアップが激しく、新しいバージョンを用いると他のプログラムのバージョンによっては、これらと整合性が取れず、敢えて古いバージョンを用いる事は通常である。よって、本研究ではこの整合性を中心にソースコードを見直し、Google Colaboratory⁴⁰を用いて計算を行った。Google Colaboratoryは、ブラウザからPythonを実行できるサービスであり、過去の論文⁴¹でも使用している。計算の規模にもよるが、基本的には無料（無料版ではCPUタイム等の制限はある）で行え、GPUも使用出来るので、高速な演算が可能である。

3.1 Google Colaboratoryコード

本研究では、参考文献に記載されているソースコード（https://colab.research.google.com/drive/1UdQV5nQMqYlckJP0qdS3kS7sW6sEQyzs#scrollTo=prOt_VofnYCo）を利用して頂いた。ただし、筆者の環境では計算中にいくつかのエラーが出たので、図4に示す修正等を行い、ライブラリの環境を再構築した。

これらの修正を行った後、計算を行った結果を図5に示す。同図_1は、学習回数（エポック）を横軸とした正確さのグラフであり、正確であればその値は1に近づく。同図_1では、ニューラルネットワークのハイパーパラメー

```
# See https://github.com/raghakot/keras-vis/issues/182
#!pip install -I scipy==1.2.*
#!pip install tensorflow==2.3.0

!pip3 install h5py==2.10.0

#OK
!pip install -I scipy==1.2.*
!pip install tensorflow==2.1.1
!pip install keras==2.3.1

#test
#!pip install -I scipy==1.2.*
#!pip install tensorflow==1.14.0
#!pip install keras==2.3.2

!pip install segmentation-models

Collecting h5py==2.10.0
```

図4 コード修正部分

タの良し悪しを確かめるvalidation accuracyはそうでもないが、ニューラルネットワークの重みを更新するtraining accuracyは学習回数が増えるにつれて綺麗に1に近づいている。同様に同図_2の誤差についても同様の結果が示されており、学習が問題無く行われている事が示された。

同図_3は、学習モデルにおける混同行列 (confusion matrix) を示しており、アルゴリズムの性能を可視化するための特有の表配置である。同図マトリックス (Danboo - Danboo : 1) (Danboo - Microbit : 0) (Danboo - scissor : 0) において、Danboを認識したデータ数1個、DanbooをMicrobitで認識したデータ数0、Danboをscissorとして認識したデータ数0個と判別が行われている事が示された。

この結果 (学習済みモデル) を用いて予測した結果を図6に示す。同図はGrad-CAM (radient-weighted Class Activation Mapping) を用いた画像予測の根拠結果を示したものである。各物体の上に表示されたキャプションはDanboo (0.98, Danboo) は、この物体を98%の確率でDanbooと認識したという事を示している。Grad-CAMとは予測値に対する勾配を重み付けすることで、重要なピクセルをヒートマップを用いて可視化する技術である。この図を見る事で、画像認識においてどの部分に着目して認識しているかを可視化出来る。具体的には、図面の赤色の部分が判断根拠になっている領域を示している。

判断根拠領域に着目すると、「はさみ」においては、「はさみ全体」を画像認識に使用していることが示されているが、逆に「Microbit」は本体でなく、本体付近の影で認識している事が分かる。この結果から考察すると、「はさみ」においては、どの方向から撮影しても「はさみ」として認識されるが、「Microbit」においては、影の状態が違う方向から撮影すると判別が難しいという事が導かれる。

この学習モデルを用いて実際に画像認識を行った結果を図7に示す。ここでは、「はさみ (scissor)」が100%の精

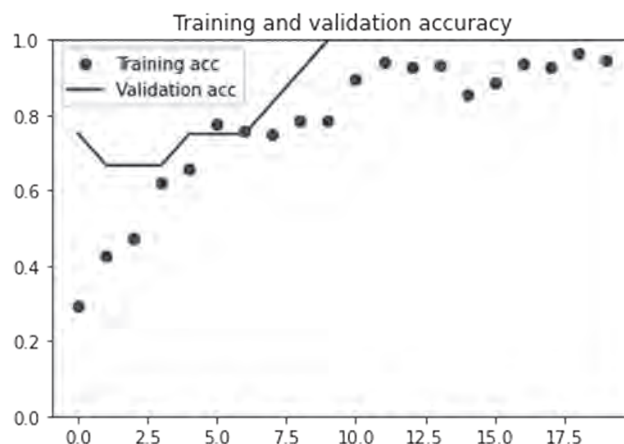


図5_1 エポックごとの精度

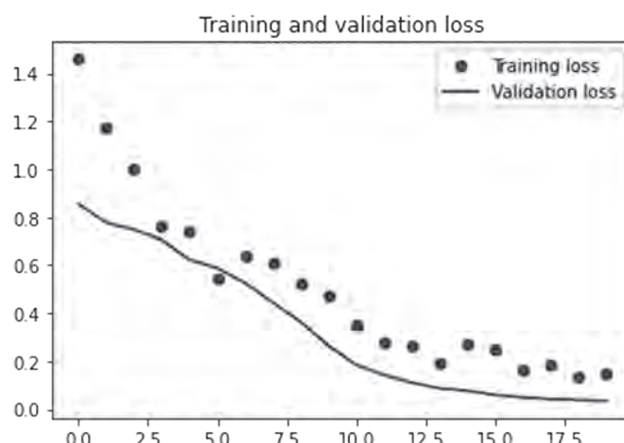


図5_2 エポックごとの損失

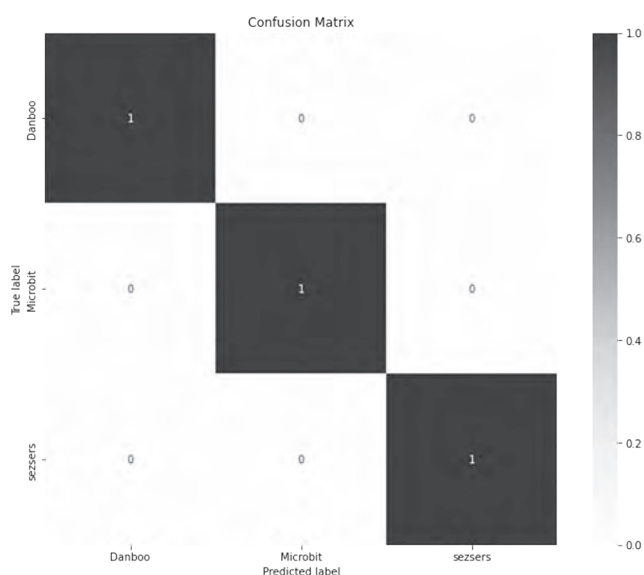


図5_3 混同行列

Grad-CAM (green: correct, red: incorrect)



図6 Grad-CAMを用いた画像予測の根拠結果



図7 リアルタイムで画像判別中

度で認識されている事が分かる。一方、他の2つの物体を認識させた場合には、実はあまり精度よく認識出来なかった。今回は動画で物体認識を行っているので、1秒間に数回程度画像認識が実行されているが、その認識結果が一定せず、ふらつきが多かった。この判別結果をコンピュータに取り込んで解析を行えば、時系列の平均等、定量的な評価が可能になるが、今回は図7に示すように動画画面を目視するだけだったので、定量的な分析は行えなかった。

定性的な評価としては、クラウドで計算された学習モデルより、Google Colaboratoryを用いて計算したモデルの方が、一般的に認識率が高かったが、それでも「はさみ」は実用に耐えるが、他の2つの物体認識は難しいような状態であった。

おわりに

本研究では、AIカメラを用いて画像認識を行った。AIカメラ単独では学習モデルの計算が出来ないので、クラウドもしくはローカルのコンピュータで画像認識モデルの計算を行う必要がある。クラウドタイプで画像認識モデルを

計算する標準的な学習モデル構築手法は専門的な知識が無くても利用できるが、その反面、アクセス制限や画像認識精度に問題がある事が分かった。

そこで、Google Colaboratoryを用いて自前で画像認識モデルを学習させ、AIカメラの教育現場での運用可能性を探ろうとしたが、それでも実用化にはまだ問題がある事が分かった。今後は、コンピュータとAIカメラを接続して定量的な評価が行えるシステムを作成すると同時に、学習モデルを生成する画像の枚数や、撮影状態（光源の位置）を配慮して、より実用度の高い学習モデルの構築を目指すものである。

4. 謝辞

本研究は、令和3年度「重度重複障害児のコミュニケーション獲得性に向けた支援者支援ディープラーニングシステムの試作（課題番号：19K11417）」の一部として行われたことを記して謝意を示す。

5. 参考文献

- i 小学校プログラミング教育の趣旨と計画的な準備の必要性について (1)、文部科学省初等中等教育局情報教育・外国語課情報教育振興室、https://www.mext.go.jp/content/20200210-mxt_jogai01-100013292_01.pdf
- ii 超スマート社会の実現に向けたデータサイエンティスト育成事業、https://www.mext.go.jp/a_menu/koutou/kaikaku/mirai-kachisouzou/1403520.htm
- iii ディープラーニングを用いたモーションヒストリー解析の応用、宮崎英一、坂井聡、谷口公彦、佐野将大、近藤創、香川大学教育学部研究報告 (2) 119-123 2020年3月
- iv エッジデバイスを用いた簡易顔認識システムの試作、宮崎英一、坂井聡、谷口公彦、佐野将大、近藤創香川大学教育学部研究報告 (3) 63-67 2020年9月
- v 視覚障がい者に向けたディープラーニングを用いた物体

提示システムの試作、宮崎英一、坂井聡、谷口公彦、佐野将大、近藤創、香川大学教育学部研究報告(2) 125-129
2020年3月

- vi Brownie、<https://github.com/ksasao/brownie/blob/master/README.ja.md>
- vii M5StickV (マイク搭載版)、<https://www.switch-science.com/catalog/6651/>
- viii Vトレーニング、https://docs.m5stack.com/en/related_documents/v-training
- ix よつばと! <https://dengekidaioh.jp/product/yotsubato/>
- x M5StickVで超お手軽エッジAI画像認識、<https://karaage.hatenadiary.jp/>
- xi Google Colaboratoryの機械学習でM5UnitVに動物フィギュアを認識させる、<https://qiita.com/Nabeshin/items/d526a94b8f3be85bbb05>
- xii TensorFlow、エンドツーエンドのオープンソース機械学習プラットフォーム <https://www.tensorflow.org/?hl=ja>
- xiii Colaboratoryへようこそ、<https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja>
- xiv Google Colaboratoryを用いたディープラーニング 画像認識学習システムの試作、宮崎英一、坂井聡、谷口公彦、佐野将大、近藤創、香川大学教育学部研究報告(3) 57-61
2020年9月