MoveNetによる姿勢推定を簡便に実行するための Pythonモジュールの開発

八木徹 *,**　　山口敏和 *,**
Toru Yagi　　Toshikazu Yamaguchi

* 江戸川大学　　** 江戸川大学情報教育研究所
*Edogawa University　**Edogawa Institute of Information Education

機械学習のフレームワークを利用しやすくし，AI利用プログラム作成の敷居を下げることは，多くの人にAI活用を促すためにも重要である．そのため，代表的なフレームワークであるTensorFlowの機能や，TensorFlow Hubにある学習済みモデルを活用しやすくするためのPythonライブラリを作成することは，特に意義深いと言える．そこで本研究では，TensorFlow HubのMoveNetモデルを利用して，姿勢推定を行うPythonモジュールを開発した．さらに，完成したモジュールを用いて，実際に姿勢推定を行うためのサンプルプログラムを作成した．既存の手法であるStretch3を用いたScratchプログラムや，ml5.jsを用いたJavaScriptプログラムとの比較を行い，姿勢推定を行うプログラムが容易に作成可能であることを確認した．

キーワード：TensorFlow Hub, MoveNet, 姿勢推定, Stretch3, ml5.js

1. はじめに

現在AI関連技術は発達を続けており，これを適切に理解して活用する力の重要性が増している．日本のAI戦略2019やAI戦略2022では，「文理を問わず，全ての大学・高校生が，過程において初級レベルの数理・データサイエンス・AIを習得」することを目標に掲げている[1，2]．また，数理・データサイエンス・AI教育プログラム認定制度においては，リテラシーレベルとして「学生の数理・データサイエンス・AIへの関心を高め，適切に理解し活用する基礎的な能力を育成」すること，応用基礎レベルとして「数理・データサイエンス・AIを活用して課題を解決するための実践的な能力を育成」することが求められている[3，4]．コンピューターの利用が，現在では誰にも求められる情報リテラシーの一つになっているのと同じように，今後の社会ではAIを各人の問題解決に活用する力も求められるようになると考えられる．

AIそのものを研究開発する立場ではなく，AIを利用する立場の人にとって，AI研究の過程で得られた技術に対して関心を高め，これを活用できるようにするためには，どのような学修が必要となるであろうか．様々な方法が考えられるが，AI技術を利用したプログラムを実際に作成して動作させる，という方法は一つの重要なアプローチになり得る．

AI技術を活用したプログラムの作成には，TensorFlow[5]などの機械学習用フレームワークが用いられる．TensorFlowを利用するとPythonやJavaScriptでAI活用アプリケーションを開発できるが，特にJavaScriptにおいては，TensorFlowを使いやすくするml5.jsライブラリ[6]があり，開発しやすい環境を提供している．

また，初学者向けのビジュアルプログラミング言語としてScratchがあるが，このScratchでAI機能を利用できるようにした拡張ブロックも開発されている(Stretch3)[7，8]．Stretch3では，拡張ブロックを用いて，気軽にAI利用プログラムを作成することができる．

では，Stretch3を用いてAI活用プログラミングを体験した学修者がさらなる興味を持った場合，どのようにステップアップすれば良いだろうか．例えば，p5.jsのようなJavaScriptライブラリとml5.jsを利用したプログラム作成も良い方法となる．実際，ml5.jsのサイトでは，p5.jsを用いたプログラム例が多数示されている[6]．

一方，Pythonはデータサイエンスや機械学習で広く活用されているため，Scratchプログラミングの次のステップとしてPythonを用いた学修も良い選択肢となる．ところがPythonでは，JavaScriptにおけるml5.jsのように，TensorFlowを利用しやすくするためのライブラリがない．例えば，TensorFlow Hubには，学習済みのモデルやその

2023年1月31日受付　2023年2月22日受理

チュートリアルが公開されており，これを使うことでAI機能を活用するプログラムを作成できる[9, 10]．しかし，TensorFlowを直接利用するプログラム作成は，初学者にとっては敷居が高い面もある．

このため，TensorFlowの機能やTensorFlow Hubの学習済みモデルを簡便に活用するためのPythonライブラリを開発することは，初学者のAI学修教材を開発する上でも非常に意義深い．そこで，本研究は，PythonでのAIプログラミングを簡易化するライブラリの開発を目指す．その具体的な一歩として，人物が映っている入力画像に対して姿勢推定を行うためのPythonモジュールを開発することとした．今回利用する姿勢推定モデルには，TensorFlow Hubに格納されているMoveNet[11]を用いた．また，Scratchプログラマが，Stretch3を使ったAIプログラミングを経験した後に，なるべくStretch3に近い感覚で，Pythonプログラムの作成に移行できるようなモジュールの作成を目指した．これにより，初学者のAI利用プログラムのハードルを下げ，今後のAI学修教材開発の基礎とする．

2. 設計と開発

本研究で開発するモジュールは，Pygame Zeroとともに利用することを想定して設計する．Pygame Zeroは，PythonでのGUIアプリ開発を簡易化するライブラリである[12]．また，Pygame ZeroではScratchからの移行も想定されており，そのドキュメントには，Scratchコードと比較しながらPythonでのゲーム作成に取り組めるよう解説がなされている．このように，Pygame Zeroは，Scratchでプログラミングを学んだ学修者が次のステップに進み，より自由度の高いプログラムを作成しやすくする環境を提供している．今回の研究でも，Stretch3でAIプログラムを作成した学修者が，PythonでのAI利用プログラム作成に移行することを想定するため，Pygame Zeroとともに利用することが可能なPythonモジュールを構築することとした．

本研究で作成するシステムのクラス図をFig. 1に示す．今回のシステムは２つのクラスからなるシンプルな構造となっている．EtoolPoseクラスには，姿勢推定を行う機能を持たせ，EtoolCameraはWebカメラを利用して画像を取得するクラスとし，これら2つのクラスによって，リアルタイムの画像を用いた姿勢推定ができるようにする．

EtoolCameraは，OpenCVのVideoCaptureを利用してWebカメラからの画像を取得するクラスである．データ属性のcameraは，VideoCaptureのオブジェクトを保持する．また，画像を取得するメソッドとして，read( )とget_surface( )を用意している．read( )メソッドは，呼び出された時点での画像フレームを取得してNumPyのndarrayとして返す．その際の色情報はRGBの順としている．また，get_surface( )メソッドは，画像フレームをPyGameのSurfaceオブジェクトとして返す．このメソッドで得られた画像フレームは，そのままのデータでPygame Zeroの画面にblit( )関数を用いて貼り付けることができる．

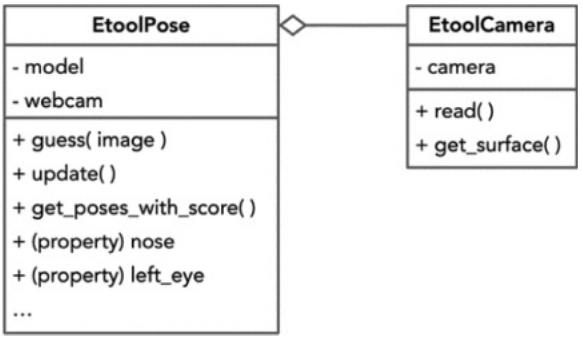

Fig. 1　開発するモジュールのクラス図

EtoolPoseは，姿勢推定モデルMoveNetを利用して，画像データから姿勢を推定するクラスである．具体的には，鼻や目，肩など体の17箇所の点の座標を取得することができる．データ属性のmodelはMoveNetモデルのオブジェクトを保持する．もう一つのデータ属性であるwebcamは，EtoolCameraのオブジェクトを保持し，Webカメラからの画像を取得する．これによりWebカメラの画像情報を都度取得しながら解析することができるようにしている．また，EtoolPoseクラスは，コンストラクタにおいてMoveNetの学習済みモデルをTensorFlow Hubからダウンロードし，セットアップを行っているため，オブジェクトの生成後，すぐに姿勢推定を実行できる．

EtoolPoseを利用して，姿勢の情報を取得するためのメソッドをTable 1に示す．get_poses_with_score( )メソッドは，呼び出された時点でのWebカメラの画像フレームを用いて姿勢推定を行い，結果をまとめて返す．また，num_peopleやnose，left_eyeなど体の部位を表すメソッドは@propertyデコレーターを使って定義されており，名称を指定するだけで対応する値を取得することができる．

Table 1で示した他にも，guess(image)メソッド(指定した画像で姿勢推定を行う)，get_keypoint_by_name(index，point_name，threshold)メソッド(姿勢推定結果から対応する人物の体の部位の座標を取得する)などを定義しているが，ここでは割愛する．

Table 1　EtoolPoseのメソッド

メソッド	内容

get_poses_with_score( )	戻り値：
	[{
	‘キーポイント名’,［x座標，y座標，確度］,
	‘キーポイント名’,［x座標，y座標，確度］,
	…
	},
	{
	‘キーポイント名’,［x座標，y座標，確度］,
	‘キーポイント名’,［x座標，y座標，確度］,
	…
	},
	…
	]
	呼び出し時にWebカメラから画像フレームを切り出し，その画像をもとに姿勢推定を実行する．
	戻り値として，体の各点の座標情報とその点を推測した際の確度を返す．戻り値の構造は，検出された人物ごとに，体の部位の名称(キーポイント)をキーとし，その地点の［x座標，y座標，推定の確度］を値とした辞書データを格納したリストである．

(property) num_people	戻り値 (int) 人数 検出された人の数を返す．
(property) nose	戻り値 [(float) x座標，(float) y座標] 検出された鼻の座標をリストとして返す．
(property) left_eye (property) right_eye (property) left_ear (property) right_ear (property) left_shoulder (property) right_shoulder (property) left_elbow (property) right_elbow (property) left_wrist (property) right_wrist (property) left_hip (property) right_hip (property) left_knee (property) right_knee (property) left_ankle (property) right_ankle	noseと同様に，対応する体の部位の座標をリストとして返す．

3. 結果

本研究で開発したPythonモジュール，EtoolCameraとEtoolPoseを利用したサンプルプログラムをFig. 2から4に示す．各プログラムで重複する箇所は適宜省略している．

Fig. 2のサンプルプログラム1では，8，9行目でそれぞれEtoolCameraとEtoolPoseのオブジェクトを生成している．EtoolCameraオブジェクトの生成時には画面サイズを指定している．一方で，EtoolPoseオブジェクトの生成時には，事前に生成したEtoolCameraオブジェクトを引数に渡し，EtoolPoseオブジェクト内で，必要に応じて画像デー

タを取得できるようにしている．

一度EtoolPoseオブジェクトを生成すれば，その後は，任意のメソッドを使うことで，目的の情報(体の特定部位の座標)を取得できる．例えば，17行目に示したようにleft_wristを用いることで，左手首のx，y座標を取得できる．ここでは@propertyデコレーターを指定しているため，データ属性から値を取得しているかのように扱うことができるが，実際の内部処理では，呼び出された瞬間の画像を取得し，姿勢推定を実行してから対応する部位の座標を返している．サンプルでは取得した座標の位置に円を描いている(19行目)．

ここで，Fig. 2のサンプルプログラム1と，他の言語のプログラムの比較を行う．サンプルプログラム1と同等内容のプログラムをStretch3やp5.js + ml5.jsを用いて記述した例をそれぞれFig. 5およびFig. 6に示す．Fig. 5は，Stretch3でPosenet2Scratch拡張ブロックを呼び出し，左手首の座

```
 1  import pgzrun             # Pygame Zero のインポート
 2  import etoolcam as ec     # Web カメラ読み込みツール
 3  import etoolpose as ep    # 姿勢推定ツール
 4
 5  WIDTH = 640       # 生成する画面の幅 (pixel)
 6  HEIGHT = 380      # 生成する画面の高さ (pixel)
 7
 8  webcam = ec.EtoolCamera(0, WIDTH, HEIGHT)  # EtoolCamera オブジェクト生成
 9  poseEstimate = ep.EtoolPose(webcam)        # EtoolPose オブジェクト生成
10
11  def draw():
12      # 画面更新
13      img = webcam.get_surface()  # ビデオの 1 フレームを取得(Surface オブジェクト)
14      screen.blit(img, (0,0))     # 画像を貼り付ける
15
16      # 指定位置に図形を描画
17      x, y = poseEstimate.left_wrist         # 左手首の座標を取得
18      if x is not None and y is not None:    # 値が None ではない場合
19          screen.draw.filled_circle((x,y), 40, (255,255,255))    # 円を描画
20
21  def update():
22      pass        # このサンプルでは update()関数は空としている
23
24  pgzrun.go()     # Pygame Zero の実行
```

Fig. 2　サンプルプログラム1

```
 1  #  (略) sample1 の 1 から 9 行と同じ
 2
 3  act1 = Actor("circle", center=(0,0))        # Actor のオブジェクト生成
 4
 5  def draw():
 6      # 画面更新
 7      img = webcam.get_surface()  # ビデオの 1 フレームを取得(Surface オブジェクト)
 8      screen.blit(img, (0,0))     # 画像を貼り付ける
 9
10      # Actor の描画
11      act1.draw()
12
13  def update():
14      # Actor の座標を更新
15      x, y = poseEstimate.left_wrist        # 左手首の座標を取得
16      if x is not None and y is not None:  # 値が None ではない場合
17          act1.x = x   # Actor の x 座標を更新
18          act1.y = y   # Actor の y 座標を更新
19
20  pgzrun.go()     # Pygame Zero の実行
```

Fig.3　サンプルプログラム2

```
1  # （略）sample1 の 1 から 9 行と同じ
2
3  class Circle(Actor):   # Actor を定義（Actor クラスを継承）
4      def set_center(self, x, y):  # 独自メソッドの追加
5          if x is not None and y is not None:
6              self.x = x   # Actor の x 座標を更新
7              self.y = y   # Actor の y 座標を更新
8
9  act1 = Circle("circle", center=(0,0))       # Actor のオブジェクト生成
10
11 def draw():
12     # （略）sample2 の 6 から 11 行と同じ
13
14 def update():
15     # Actor の座標を更新
16     x, y = poseEstimate.left_wrist        # 左手首の座標を取得
17     act1.set_center(x,y)                  # 作成した Actor のメソッドを利用
18
19 pgzrun.go()
```

Fig. 4　サンプルプログラム3

Fig. 5　Stretch3を用いたプログラム

```
1   function draw() {
2     // 画面更新
3     image(video, 0, 0, width, height);
4
5     if(poses.length > 0){
6       // 左手首の座標取得
7       let x = poses[0].pose.leftWrist.x;
8       let y = poses[0].pose.leftWrist.y;
9
10      // 指定位置に図形を描く
11      fill(255,255,0);
12      circle(x,y,80);
13    }
14  }
```

Fig. 6　p5.jsとml5.jsを用いたプログラム（一部）

標にスプライトを移動するプログラムである．Fig. 6は，ml5.jsで紹介されているサンプルプログラム[13]のdraw( )関数のみを修正して，左手首の座標に円を描くようにしたプログラムである．いずれも，サンプルプログラム1と同じように，姿勢推定で求めた左手首の座標に何らかの表示を行うプログラムとなっている．

Stretch3では，Fig. 5で示したように，Scratchの変数として「左手首のx座標」「左手首のy座標」などが用意されている．体の部位17箇所のxとy座標に対応し，座標に関してだけで34個の変数が用意されている．プログラムを作る際には，目的の変数のブロックを取り出して，必要な加工をすれば良いことになる．

p5.js + ml5.jsのサンプル(Fig. 6)では，poses[idx]に，識別したidx番目の人物のデータが格納されている．poses[0].pose.leftWrist.xとposes[0].pose.leftWrist.yで，それぞれ0番目の人物の左手首のx座標，y座標が取得できる．Fig. 6の12行目では，得られた座標を中心とした円を描いている．

これに対して，今回開発したPythonモジュールでは，(1)サンプルプログラム1の8，9行目に相当するオブジェクト生成の準備が必要であること(Stretch3と比べた場合)と，(2)体の部位の座標をリストで返していること(Stretch3及びp5.js + ml5.jsと比べた場合)，の2点が異なっている．(1)については，Pythonの仕様上，オブジェクト生成の記述が欠かせないことから，これ以上の簡略化は難しい．Pythonモジュールを読み込むimport文と，オブジェクトの生成をする文を合わせて，Scratchで「拡張機能を追加」する作業に対応すると考えるのが妥当と思われる．(2)については，姿勢推定を行なった結果として，leftWrist.xやleftWrist.yのような変数(@propertyデコレーターを用いたプロパティの取得)を用意することで，Stretch3やml5.jsと同じような扱いにすることもできる．しかし，今回は，x，y座標をペアで取得する方がデータとしてのまとまりが良いこと

と，その部位を推測した確度を合わせて返すなどの拡張を考えた際に「一つの部位」の情報としてまとめて取得した方がわかりやすいとの判断からリストにまとめることとした．

今回作成したEtoolPoseクラスは，ここで示した(1)(2)のポイント以外は，Stretch3やml5.jsと類似しており，体の部位に対応する属性名を記述するだけで対応する点の座標を取得することが可能となっている．

Fig. 3のサンプルプログラム2は，画面中に登場する物体をPygame ZeroのActorクラスのオブジェクトを用いて記述している．Actorクラスのdraw( )メソッドを用いることで，簡単にオブジェクトの画像を画面に描画することができる(11行目)．オブジェクトの座標はupdate( )関数の中(17，18行目)で更新している．Fig. 4のサンプルプログラム3では，Actorクラスを継承したCircleクラスを定義している(3から7行目)．このようにすることで，エラーチェックなどを適切に処理するメソッドを定義することも可能である．

サンプルプログラム2，3のようにPygame ZeroのActorクラスを用いると，Scratchの「スプライト」を意識したプログラム作成が可能になる．Scratchで各スプライトに記述していたプログラムのうち，状態の変化(座標，見た目(コスチューム)など)をPygame ZeroのActorを継承したクラス内部で記述することで，Scratchとの類似性を高めることができる．このような類似性は，Scratchからの移行のしやすさに繋がる．

4. まとめ

本研究では，ScratchにおけるStretch3や，JavaScriptにおけるp5.jsとml5.jsなどのAIプログラム環境と同じように，PythonでMoveNetを用いた姿勢推定を簡単に利用できるモジュールを開発した．特に，Stretch3利用者がPythonに移行しやすくするために，Pygame Zeroでの利用が可能なモジュールとした．完成したPythonモジュールを用いて，Webカメラから取得した画像の姿勢推定を行い，体の部位の座標を取得するプログラムを容易に作成できることを確認した．

今後は，今回作成したPythonモジュールを活用した学修教材の作成と，MoveNet以外のモデルを同じ操作感で利用できるライブラリを開発していくこととする．

参考文献

[1]統合イノベーション戦略推進会議，“AI戦略2019”，内閣府(2019)
[2]統合イノベーション戦略推進会議，“AI戦略2022”，内閣府(2022)
[3]文部科学省高等教育局，“数理・データサイエンス・AI教育認定制度(リテラシーレベル)実施要項細目(改訂)”，(2022)
[4]文部科学省高等教育局，“数理・データサイエンス・AI教育認定制度(応用基礎レベル)実施要項細目”，(2022)
[5]Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng., “TensorFlow: Large-scale machine learning on heterogeneous systems”, Software available from tensorflow.org. (2015)
[6]ml5.js, https://ml5js.org/, (2023/2/12)
[7]石原純也, 倉本大資，“Scratchではじめる機械学習”，オライリー・ジャパン(2020)
[8]Junya Ishihara (2020), “Stretch3”, https://github.com/stretch3/stretch3.github.io (2023/2/12)
[9]TensorFlow Hub, “TensorFlow Hub”, https://www.tensorflow.org/hub (2023/2/12)
[10]TensorFlow, “Tutorials (TensorFlow Hub)”, https://www.tensorflow.org/hub/tutorials (2023/2/12)
[11]Ronny Votel and Na Li (2021), “Next-Generation Pose Detection with MoveNet and TensorFlow.js”, https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html (2023/2/13)
[12]Daniel Pope and Richard Jones (2015), Pygame Zero, https://github.com/lordmauve/pgzero (2023/2/13)
[13]Bomani Oseni McClendon (2020), PoseNet_webcam, https://github.com/ml5js/ml5-library/tree/main/examples/p5js/PoseNet/PoseNet_webcam (2023/2/12)