07　ジェスチャーで絵を描くシステムの作成

内田陽大，古川純至，吉田朋弥

指導教員　菅野　研一

1. はじめに

学園祭で使えるような知識なしで,感覚で操作できるゲームを開発しようとした

子供が楽しめるように考えた結果,絵を使うのが最適なのではないかと思い,お絵描きゲームの開発に決定した

2. 研究概要

① 手の動きをカメラから読み込み座標を取得する

② 座標を用いて線を表示し絵を描く

③ 描いた絵を保存する

④ 戦わせるための戦闘力を AI で算出する

⑤ 描いた絵を戦わせる

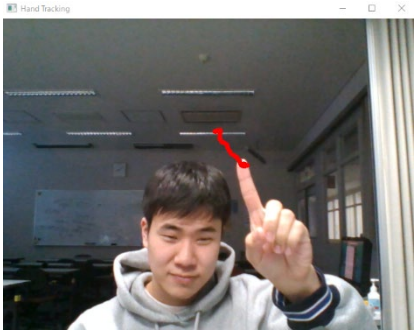

図 1

3. 開発環境

表 1.実行環境

OS	Windows10
言語	Python
ライブラリ	OpenCV
	MediaPipe
カメラ	Web カメラ
開発環境	Unity

表 2.AI 開発環境

OS	Ubuntu
	docker
言語	Python
使用機器	Deep Learning 専用 PC
GPU	GeForce RTX3090
CPU	Intel Xeon W-2223@3.6GHz
RAM	32GB
SSD	2×1TB

4. 仕様について

4.1 手の座標の取得

mediapipe を Python で使用し人差し指先端の座標の位置に線を描けるようにした

線を保存している配列の中身をすべて削除するキーを設定し絵を最初から書き直すために必要な一括で全ての線を削除する機能と

今描いている線の座標を保存している配列を削除もしくは直前に描いた線を削除する機能を割り当てた.

4.2 保存について

画面上に描画された軌跡から BGR カラーコードに基づき,R（赤）を抽出して保存する.ほかの赤色が映ったに画面上に赤色が出てこないように(0，0，254)～(0，0，255）の範囲で保存ができるようにしている.

4.3 AI 学習について

畳み込みニューラルネットワーク(CNN)を使い画像を入力とし,攻撃力,防御力などの戦闘にかかわるデータを予測する AI の開発を行った.今回の学習ではインプット層に画像一枚,アウトプット層に先頭にかかわるデータ 5 個を同時に出力させる形で学習を行った.

今回の CNN 学習を行うモデルは

[1]Conv2D 層:フィルタ数:32,入力形状:(300, 300, 4) 活性化関数: ReLU

[2]MaxPooling2D 層: プールサイズ: (2, 2)
[3]Conv2D 層: フィルタ数: 64,
活性化関数: ReLU
[4]MaxPooling2D 層: プールサイズ: (2, 2)
[5]Flatten 層: データを平滑化するための層
[6]Dense 層: ユニット数: 64, 活性化関数:
ReLU
[7]Dense 層: ユニット数: 5 (出力層)
合計 7 層で学習をかけていて学習結果はグラフに表す

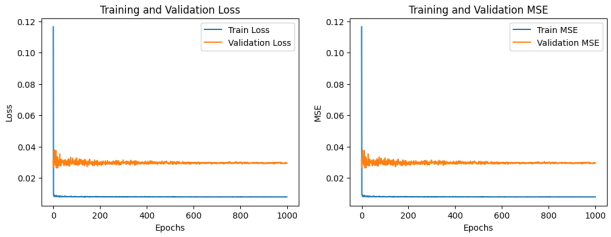

図 2

左はトレーニングと検証段階での損失と精度を表している.
右はトレーニング精度と検証精度を表している.
この図 2 のグラフから学習段階の精度は高いが実際にテストをすると精度が低い.そのため、精度を向上させる工夫を行う必要がある。入力として使用していた CSV ファイルを正規化させることで誤差関数値を 0.03 まで減少させることに成功した.モデルの学習層を増やして実験したが思ったよりも改善しなく,計算コストの増大を考えると現状の層で満足できるため完成とした.

5. Unity を用いたゲーム開発

ジェスチャーで描いたイラストを使って対戦をできるゲームを作成した.

ジェスチャーで描いたイラストを戦闘力算出 AI に渡し,5 つのパラメータを取得し,今まで描いたイラストから対戦相手を選び対戦することができる.
※流れを以下の図 3 に記載

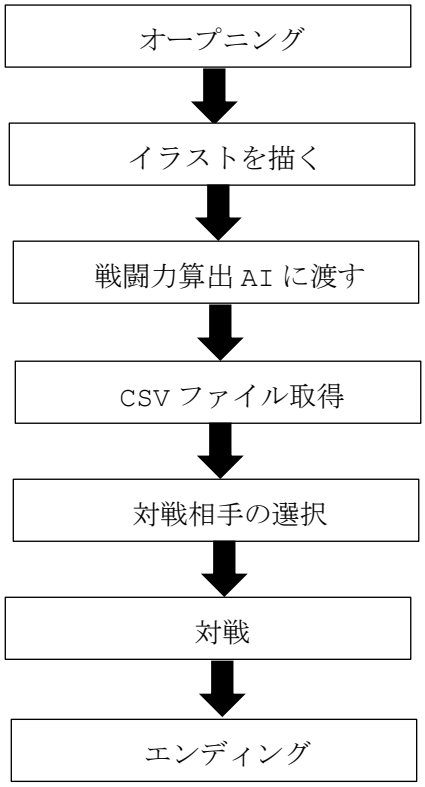

図 3 ゲームの流れ

6. おわりに

今回の研究では Mediapipe を活用して,子供たちや IT 技術未経験者でも楽しめるゲームを開発した.このゲームは,ユーザーが自分で描いた絵を戦わせることができ,子供たちが IT 技術を体験しながら遊ぶことができる.子供たちや初心者にとって IT に親しむきっかけとなれればと思う.

参考文献

[1] 畳み込みニューラルネットワーク (Convolutional Neural Networks)
https://www.tensorflow.org/tutorials/images/intro_to_cnns?hl=ja

[2] Mediapipe で手の形状検出を試してみた (Python)
https://qiita.com/bianca26neve/items/116814135739929759a0

[3] 回帰：燃費を予測する
https://www.tensorflow.org/tutorials/keras/regression?hl=ja