KaiRA

京都大学人工知能研究会

KaiRA

Kyoto univ. AI Research Association

KaiRA Journal

Kyoto univ. AI Research Association

2024

KYOTO UNIV. NOVENVER FESTIVAL

はじめに

まえがき
まえがき
まえがき
まえがき
まえがき
まえがき
まえがき
まえがき

京都大学人工知能研究会 KaiRA 会長
松田拓巳

目次

第 1 章　カメラ入力を用いた強化学習によるライントレーサの実現　6
1.1　はじめに 6
1.2　全体像 6
1.3　シミュレーション環境の作成 7
1.3.1　ランダムコース生成 7
1.3.2　アクションの適用 8
1.3.3　観測の作成 8
1.3.4　報酬関数 9
1.4　エージェントについて 9
1.4.1　活性化関数と最適化手法の変更 9
1.4.2　画像とスカラーの同時入力への対応 9
1.4.3　パラメータ数の削減 10
1.5　Sim2Real 10
1.5.1　モータの制御 10
1.5.2　観測の作成 11
1.6　まとめ 11

第 2 章　目線で操るマウスカーソル　12
2.1　視線追跡モデルを開発した理由 12
2.2　全体像 12
2.3　目の位置の推定 13
2.4　視線の角度推定 14
2.4.1　視線の角度推定（手法 1） 14
2.4.2　視線の角度推定（手法 2） 15
2.5　最終結果 16
2.6　最後に 18

第 3 章　KaiRA くんを動かそう　19
3.1　はじめに 19
3.2　モーションの生成 19
3.3　アニメーションの生成 20
3.4　おわりに 21

第 4 章　最強じゃんけん AI　22

4.1 概要 . . . 22
4.2 骨格推定モデル . . . 22
4.3 実験 . . . 23
4.3.1 データ . . . 23
4.3.2 時系列予測 . . . 23
4.3.3 推論速度 . . . 24
4.4 結論 . . . 24

第 5 章 京大シラバス検索 RAG システム - システム概要編 25
5.1 京大シラバス検索 RAG システムの基本情報 . . . 25
5.1.1 シラバスと RAG システム . . . 25
5.1.2 実装の優先事項と概要 . . . 26
5.2 絞り込み . . . 26
5.2.1 KULASIS の絞り込みのための検索項目 . . . 26
5.2.2 追加した絞り込みのための検索項目 . . . 27
5.2.3 絞り込みの方法 1 metadatas . . . 28
5.2.4 絞り込みの方法 2 delete . . . 29
5.3 前処理 . . . 29
5.3.1 前処理の目的 . . . 29
5.3.2 URL の取得 . . . 30
5.3.3 絞り込みのためのデータを抽出・生成 . . . 30
5.3.4 類似度検索のためにデータ . . . 31
5.3.5 その他の前処理 . . . 31
5.4 改善案 . . . 31

第 6 章 京大シラバス検索 RAG システム - 検索手法編 33
6.1 探索方法の基本情報と目的 . . . 33
6.1.1 埋め込みベクトルによる方法 . . . 33
6.1.2 単語出現頻度に基づいた検索手法 . . . 33
6.2 埋め込みベクトル探索方法の実験 . . . 33
6.2.1 手法 . . . 33
6.2.2 コサイン類似度 (similarity) . . . 34
6.2.3 MMR(周辺関連性最大化:Maximal marginal relevance) . . . 34
6.2.4 結果のまとめ . . . 34
6.3 単語出現頻度に基づく検索手法の実験 . . . 34
6.3.1 BM25 （Best Matching 25）とは . . . 34
6.3.2 実験結果・考察 . . . 34

第 7 章 お絵描き予測 AI 39
7.1 概要 . . . 39
7.2 点群 . . . 39
7.2.1 実験 . . . 39
7.2.2 失敗した要因 . . . 40

7.3 時系列データ 40
7.4 終わりに 41

参考文献 42

第 1 章

カメラ入力を用いた強化学習によるライントレーサの実現

1.1 はじめに

DQN（Deep Q-Network[1]）の登場以降、強化学習は大きく発展し、これまでに様々な手法が提案されてきました。例えば、DreamerV3[2] がマインクラフトにおいてダイヤモンドの採掘に成功するなど、ゲーム・シミュレーション分野では強化学習は一定の成功を収めています。しかしその一方で、実世界のロボットへの応用は未だに限定的と言えます。その理由はいくつかありますが、大まかに言えばシミュレーションと現実のギャップ、データ効率の問題、リアルタイム性の問題があります。それらの課題に対処するための研究も活発に行われており、模倣学習やモデルベース強化学習、オフライン強化学習や Sim2Real など、現在でも話題には事欠かない状態です。

この章では、実際に私が製作した強化学習で動作するライントレーサについて、その実装から結果までを説明します。

1.2 全体像

まず初めに、ライントレーサの全体像について述べます。このライントレーサのコンセプトは強化学習と全方向移動です。これら 2 つのコンセプトは互いに独立しているわけではなく、全方向移動が可能であることは強化学習にとってメリットとなっています。例えば非ホロノミックな対向二輪ロボットをベースに強化学習を行う場合、横方向には移動できないため、機体が向いている方向と、行動の関係を強化学習モデルに学習させる必要があります。一方で全方向移動が可能であれば、機体の方向に関わらず任意の方向に進めるため、学習難易度は低下します。

ライントレーサの外観は図 1.1 のようになっています。透明な球殻の内部にオムニホイールを備えた本体が格納されており、球の重心の移動によって転がりながら動くような仕組みになっています。本体の底部中央には Web カメラが設置してあり、それにより床面の様子を観測できます。

そして、回路系のシステムは図 1.2 のようになっています。ルーターやバッテリーを内蔵することで、一台で完結するような構成となっています。

また、ソフトウェアについても簡単に説明します。Web カメラから取得した画像は圧縮されたのち、二値化処理を行い強化学習エージェントに観測として与えられます。そして、エージェントは進行方向を表すスカラーを出力し、その値を用いて 2 つのモータの出力が制御されます。

今回、エージェントの学習は自作した単純なライントレーサのシミュレータ上で行い、現実での追

加学習は行いませんでした。

以下の部分では、ソフトウェアの実装についてシミュレーション環境、エージェント、Sim2Real の 3 つに分けて説明し、最後に実際にライントレースさせた結果を示します。

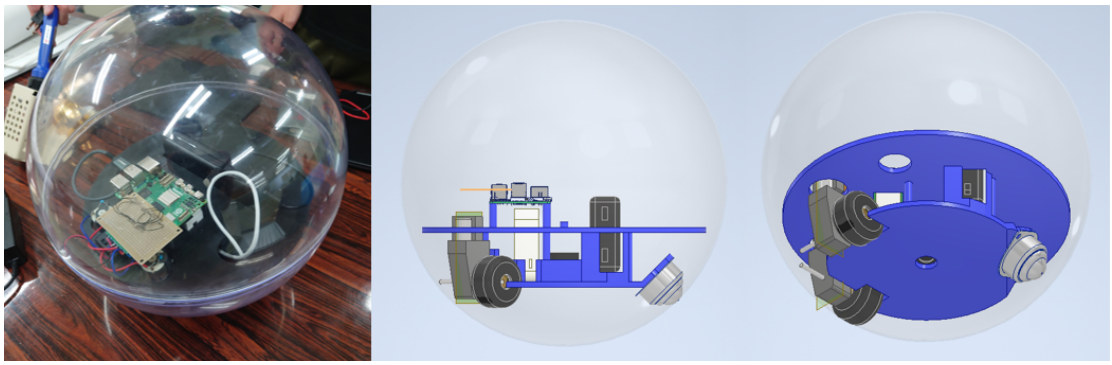


図 1.1: ライントレーサの外観

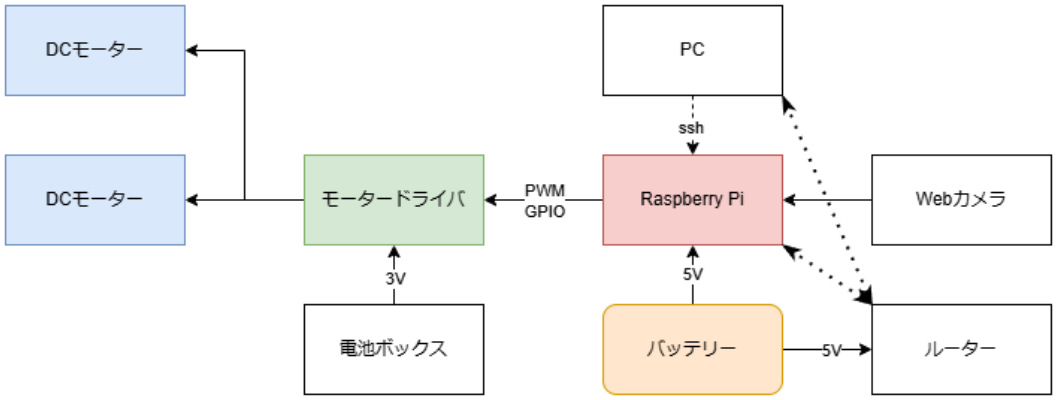


図 1.2: 回路系システム

1.3　シミュレーション環境の作成

1.3.1　ランダムコース生成

ライントレース用のランダムなコース生成は、エージェントの汎化性能を高めるために重要です。単一のコースで学習を行うとエージェントはそのコースに特化してしまい、たとえ同じコースだとしても実世界とシミュレーションのギャップに耐えられなくなってしまいます。そこで、事前に複数のランダムなコースで学習することで、様々な状況に対応できるエージェントを育成します。

コースの生成手順としては、まず領域を 4 × 4 のグリッドに分割し、幅優先探索アルゴリズムでスタート地点からゴール地点までの経路を生成します。その後、マスの境界のランダムな位置に線を通すポイントを定め、OpenCV の関数を用いて黒線を描画することでコースを完成させます。その様子を図 1.3 に示しています。

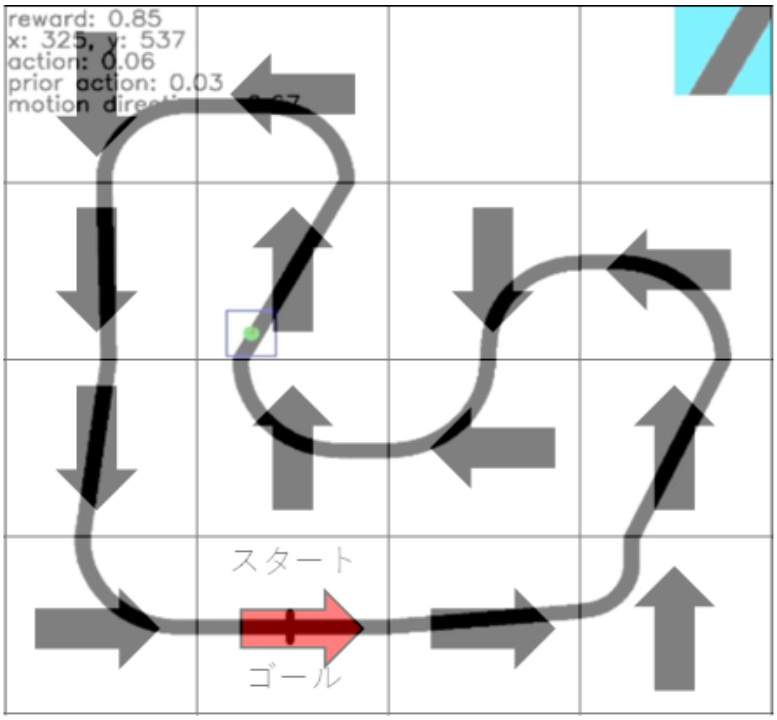


図 1.3: コース生成のイメージ

1.3.2　アクションの適用

エージェントから出力されたアクションは、仮想的なライントレーサの移動方向に変換されます。初期モデルでは、アクションを 2 次元ベクトルで表現していましたが、コースの途中で停止したり、逆走する等の問題があり、最終的には速度を固定し自由度を削減することで、1 次元のスカラーで移動方向を表現する方法を採用しました。

具体的には、図 1.4 のように、アクション *action* と概念的な進行方向 *action_average* を用いて、移動方向 θ を計算します。なお、*action_limit* は進行方向に対して、移動方向がどれだけ逸脱できるかを制御するパラメータです。

$$\theta = (\text{action\_average} + \text{action} \times \text{action\_limit}) \times \pi$$

これにより、基準方向と移動方向が分離され、エージェントの学習効率が向上するとともに、前述の問題を解決することができました。

1.3.3　観測の作成

本シミュレータでは、エージェントが観測する情報として、ライントレーサ周辺のコースを切り取った 64 × 64 のグレースケール画像、進行方向を示すスカラー値、前回のアクションを示すスカラー値の 3 種類を使用しています。

ただし、Gym の仕様上スカラー値も一度画像に変換し 3 チャンネルの画像としたうえで、エージェント側に渡しています。

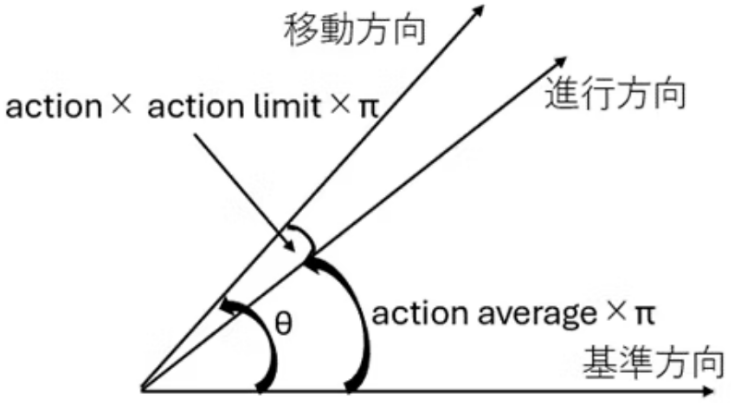


図 1.4: 移動方向の計算

1.3.4 報酬関数

報酬関数は、エージェントの行動を適切に誘導するために設計されます。今回の報酬は 2 つの要素から構成されています。1 つ目は、前回のアクション *previous_action* との L1 誤差に基づく報酬で、アクションの振動を抑制するために設けられています。2 つ目は、観測画像の中心 *image_center* と、黒ピクセル（ライン）の平均座標 *black_center* との距離に基づく報酬で、ロボットをライン上に維持することを目的としています。

$$\text{reward} = 1 - \frac{|\text{previous\_action} - \text{action}|}{2} - \frac{\|\text{black\_center} - \text{image\_center}\|}{\|\text{image\_center}\|}$$

これにより、エージェントはライン上をスムーズに移動し続けるように学習します。

1.4 エージェントについて

1.4.1 活性化関数と最適化手法の変更

今回エージェントとして採用した DrQ-v2[3] は、Meta が開発した Q 学習ベースの強化学習アルゴリズムであり、画像を観測として連続値制御が可能な点や処理が軽量である点が特徴です。しかし、2021 年の発表以降の技術進歩を踏まえ、今回の実装では活性化関数と最適化手法を変更しました。

まず、DrQ-v2 では全ての活性化関数が ReLU でしたが、現在は GELU や SiLU が主流です。そこで、画像処理に関わるエンコーダ部分の活性化関数を SiLU に、その他の部分は GELU に変更しました。また、最適化手法も Adam から Weight Decay を導入した AdamW に変更し、パフォーマンスと安定性の向上を図りました。

1.4.2 画像とスカラーの同時入力への対応

DrQ-v2 は画像入力を前提としていますが、今回のタスクでは環境から取得した 3 チャンネルの画像のうち 2 チャンネルはスカラー情報に過ぎません。これをそのまま画像として処理するのは非効率であるため、エンコーダ部分を改良し、スカラー情報を適切に処理できるようにしました。

具体的には、画像データの内スカラーの情報しか持たない 2 チャンネルはスカラー値として、CNN には通さずそのまま画像の潜在表現に結合しました。この変更によって、より効率的な学習が可能となりました。

1.4.3　パラメータ数の削減

DrQ-v2 の元々のパラメータ設定は複雑なタスクを想定していたため、今回のライントレースタスクに対しては過剰でした。そのため、エンコーダやアクタークリティックのパラメータを削減し、重みのファイルサイズを 75158kB から 3278kB へと削減しました。

結果として、図 1.5 に示すように、若干の劣化が見られるもののモデルは十分な性能を維持しており、Raspberry Pi 5 上での制御周期を 1Hz から 3Hz まで向上させることができました。

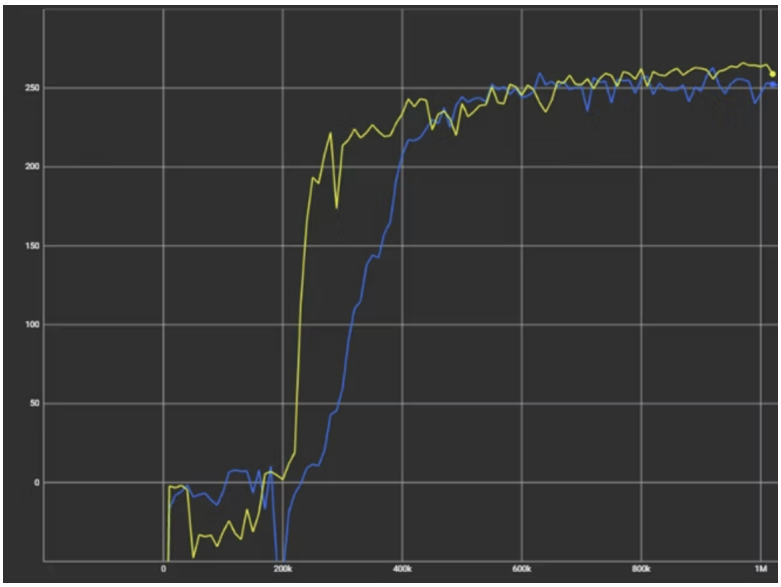


図 1.5: 報酬の推移（黄色: 変更前のモデル, 青: 変更後のモデル）

1.5　Sim2Real

1.5.1　モータの制御

シミュレータ上でトレーニングしたモデルを現実世界に持ってくるにあたっては、その差異をどのように減らすのかであったり、埋め合わせるのかということが重要になります。そのため今回のライントレーサは、できるだけシミュレータ上でエージェントの汎化性能を高めたうえで、現実世界の観測をシミュレータの観測に近づけるという方針で実装しました。

強化学習モデルから出力されたアクションは、前述の方法で移動方向 θ に変換されます。この移動方向を基に、2 つのモータの回転方向と duty（出力比率）を計算し、モータードライバの制御テーブルに従って GPIO を制御します。GPIO の制御には Raspberry Pi 用のライブラリである `gpiozero` を使用しています。

1.5.2 観測の作成

次に観測データの作成について述べます。処理の流れとしては、まず Web カメラから取得した 1920 × 1080 × 3 の RGB 画像を正方形にトリミングして 1080 × 1080 × 3 にした後、64 × 64 × 3 に圧縮しています。

次に、圧縮した RGB 画像を 64 × 64 × 1 のグレースケール画像に変換し、その画像のピクセルの値の平均 μ を求めています。その後、求めた平均値から動的に閾値を計算して、グレースケール画像を二値化し観測としています。

$$\text{threshold} = \mu \times \text{thresh}$$

$$I_{\text{binary}}(x, y) = \begin{cases} 255 & \text{if } I(x, y) \geq \text{threshold} \\ 0 & \text{if } I(x, y) < \text{threshold} \end{cases}$$

こうすることで、周辺環境が多少変化したとしても自動的に閾値を調整して、安定的にラインを検出できるようになります。

1.6 まとめ

以下の URL から実機を動作させた時の映像を見る事ができます。

`https://youtu.be/tTh6BYUjfMs?si=XSdIw1bt-LwlgnkB`

この章では強化学習によって動作するライントレーサについて、全体的なシステム構成や実装方法を説明してきました。

今回、私がライントレーサを製作した目的の一つは、強化学習を実世界のロボットに応用することは可能なのか、また、その過程でどのような困難があるのかを検証するためでした。その結果、シミュレーションと現実のギャップや、処理速度の問題など、普段は意識しないような難しい課題が存在することが分かりました。一方で、それらの課題を解決することができれば、現実世界でも強化学習エージェントを動作させることができました。

現在、ロボティクス領域において、今回紹介した強化学習をはじめとして様々な機械学習技術の応用が行われています。今後も、機械学習とロボティクスという二つの分野が相互に影響を及ぼし、発展していく事を願っています。

ソースコードは `https://github.com/Azuma413/rl_linetrace` にて公開しています。

第 2 章

目線で操るマウスカーソル

2.1 視線追跡モデルを開発した理由

近年、AI 技術が急速に進化し、画像から様々な情報を取得することが可能になりました。たとえば、物体検出の分野では、YOLO などの様々なモデルによって、カメラ画像から人や物体の位置をすばやく検出できるようになっています。こうした技術を利用し、視線の位置を特定することができれば、視線を利用した新しい操作インターフェースを提供できるのではないかと考えました。

本プロジェクトの目的は、カメラで取得した視線情報をもとに画面上のカーソルを操作するシステムを開発し、ユーザーが手を使わずに目線のみで操作できる新しいインターフェース体験を提供することです。

2.2 全体像

まず初めに、今回開発した「視線で操作するマウスカーソル」のアプリの全体像について説明していきます。

このアプリでは以下の流れによってユーザーの見ている場所にマウスカーソルを動かします。

Algorithm 1 マウスカーソルを視線位置に動かす流れ

Require: PC 内蔵カメラ、PC 画面サイズ (W, H)

1: **初期化**: ユーザーに PC 画面の 4 隅（右上、右下、左下、左上）を順に見てもらい、対応する顔画像 $T_i, (i =$ 右上, 右下, 左下, 左上$)$ を取得
2: 顔画像 T_i をもとに、モデルが PC 画面上の視線座標 (x_i, y_i) を推定
3: 視線座標 (x_i, y_i) と PC 画面サイズ (W, H) を用いて、射影変換行列 W を作成
4: **while** アプリケーションが動作中 **do**
5: 　　カメラから取得した現在の顔画像をもとに視線座標を推定
6: 　　射影変換行列 W を適用して、PC 画面上でのユーザーの視線座標を計算
7: 　　計算した座標にマウスカーソルを移動
8: **end while**

射影変換行列 W は、モデルが推定した PC 画面の 4 隅の座標に基づく「推定された PC 画面の概形」を「実際の PC 画面の概形」に変換するための行列です。

図 2.1 に、画面の 4 隅を見ているときの視線推定位置と変換後（画面サイズ）の関係を示します。青い点がモデルによって推定された視線位置を表し、赤い正方形が PC 画面の領域を表しています。

このとき、青い点（推定された視線位置）を赤い正方形（PC 画面の座標系）に変換するためには、射影変換行列 W が必要となります。4 隅分の視線位置を結ぶ四角形は、必ずしも正方形になるとは限りません。そのため、推定された座標系を PC 画面の座標系に正確に対応させるために射影変換が必要です。

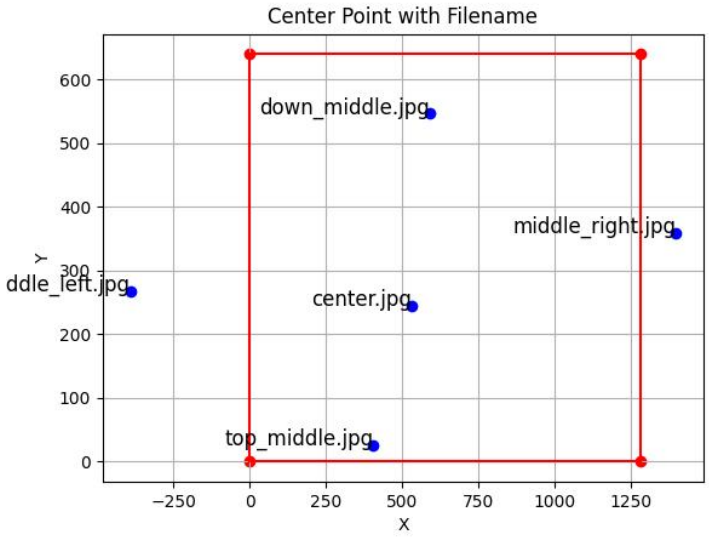


図 2.1: 視線推定位置（青い点）と PC 画面（赤い正方形）の関係

また、本モデルによる顔画像からの PC 上で見ている座標の推定の流れは以下の通りです。

■モデルの推論の流れ

1. 目の位置を推定
2. 視線の角度を推定
3. 1. と 2. で取得した「視線の角度・目の位置」をもとに、PC 上で見ている箇所を計算

以降では、この「目の位置の推定」と「視線の角度の推定」を行う具体的な方法、そして最終的なアプリによる視線の推定結果について説明していきたいと思います。

2.3 目の位置の推定

このモデルは、顔画像から目の位置を推定する際に、PnP（Perspective-n-Point）問題として扱います。PnP 問題では、以下の情報をもとにカメラの位置と回転方向を推定します:

- カメラの内部パラメータ（焦点距離や歪み係数など）
- 画像上の物体の位置と、物体が存在する座標系 D_{world}（カメラ基準の座標系 D_{camera} とは異なる基準座標系）における対応関係

本モデルでは、これを活用して、顔画像から目の 3 次元位置を推定します。
また、顔画像から目の位置を求める流れは以下の通りで、[4] をもとに作成しました。

■目の位置を推定する流れ

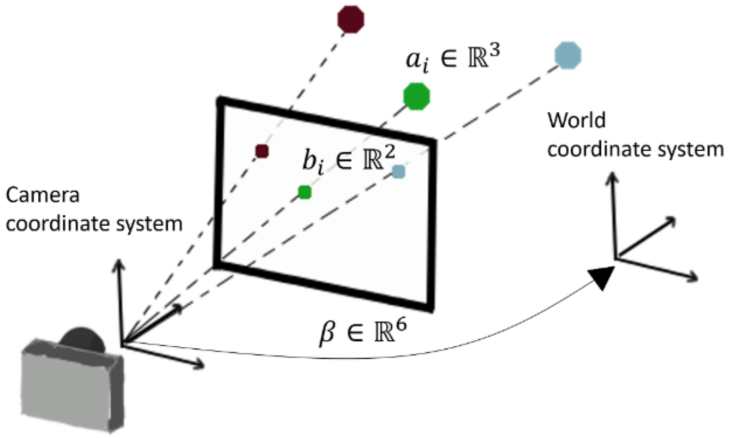


図 2.2: Camera coordinate system(D_{camera}) と World coordinate system(D_{world}) の関係性

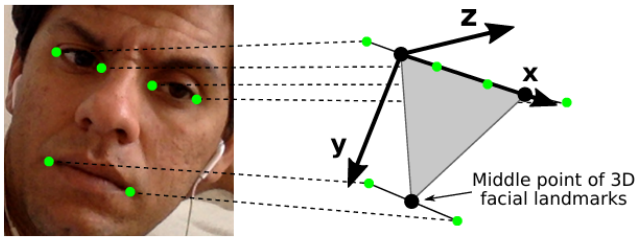


図 2.3: World coordinate system(D_{world}) 上における顔の特徴点

1. dlib[*1]を使用して、画像上の顔の特徴的な部分（目や鼻など）を検出
2.「カメラの情報（実際のカメラの位置やカメラの焦点距離・歪み係数)」と「1. で検出した特徴点と D_{world} 上での一般的な顔における関係性」をもとに、`cv2.solvepnp` を使用してカメラの D_{world} 上での位置と向き取得
3.「1. で求めた画像上での目の場所」と「2. で求めた D_{world} 上でのカメラの位置と向きから考えられる D_{camera} 上での頭の位置と向き」を元に、D_{camera} 上での目の座標を取得

2.4　視線の角度推定

次に、視線の角度を推定するモデルの開発を行いました。このモデル開発においては複数の手法を試しましたので、それらを 1 つずつ説明していきます。

2.4.1　視線の角度推定（手法 1）

手法 1 においては、[4] にて紹介された視線の角度推定モデルを用い、応用しました。推定の流れは以下の通りです。

[*1] dlib は顔のランドマーク（鼻の位置や目の位置など）を検出するためのライブラリで、今回は ERT ベースのモデル [5] を使用しています。

■視線の角度を推定する流れ

1. **■ 目の位置を推定する流れ** (2.3 節) の 2. から、顔の向き R とカメラの位置を取得
2. `cv2.warpPerspective` で R とカメラの位置に基づき、D_{world} 上での目の画像 (60 × 36) を生成
3. 生成した画像をモデルに入力し、視線角度 S を推定
4. S と R をもとに、実際のカメラから見た視線情報を算出

また、上記の 3. における視線角度推定モデルの構造と学習データセットは以下の通りです。

■モデルの構造

- 1 層目：EfficientNet(b0)[*2]
- 2 層目：Conv2d
- 3 層目：Linear+ReLU+Dropout
- 4 層目：出力値に R を結合
- 5 層目：Linear+ReLU+Dropout+Linear

■学習用データ

- MpiiGaze（片目の画像（60 × 36）と視線情報・顔の向き情報を含むデータセット）

論文 [4] では、EfficientNet ではなく VGG-16 を用いておりました。しかし、VGG-16 よりも EfficientNet の方が計算量が小さく、一般的な画像認識タスクにおける精度も上回っていることから、EfficientNet を採用しました。このモデルによって、顔の向き（yaw,pitch）を用いて視線方向を実用可能な時間 (0.04 秒/枚) で推論できました。yaw と pitch の定義は図 2.4 に示されているとおりです。

最適化手法として Adam を使用し、学習率スケジューラーは StepLR を採用しました。

しかし、結果は失敗に終わってしまいました。考えられる原因とその根拠は以下の通りです。

■考えられる原因と根拠

- 使用したデータの視線角度範囲が狭く、±20° の範囲に限定されていたため
- 訓練時とテスト時の損失値がともに低かったことから、モデルは訓練データの範囲内で良好に動作していることが確認された。しかし、データの視線角度範囲が狭いため、モデルがそれを超える視線角度に対する一般化能力を持てなかった可能性がある。

2.4.2 視線の角度推定（手法 2）

手法 2 においては、画像から直接視線の角度を推定する外部モデル (AxGazeEstimation) を活用しました。モデルの推論の流れや実験結果などは以下の通りです。

■視線推定モデル (AxGazeEstimation) による推論の流れ

[*2] 最初の層は conv2d(出力チャンネル数：32・カーネルサイズ：3・ストライド：2) に変更しました。

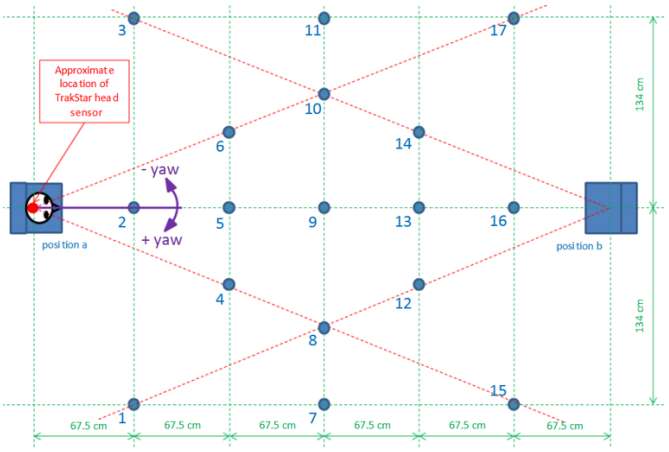


Figure 2. Overhead view of experimental set-up. The blue disks denote object locations on the ground floor. The locations consist of an equally spaced grid in terms of distance (locations 1, 2, 3, 7, 9, 11, 15, 16, 17) and angle (locations 4, 5, 6, 8, 9, 10, 15, 16, 17 from vantage point a, and locations 12, 13, 14, 8, 9, 10, 1, 2, 3 from vantage point b).

In this paper, we investigated the relationship between the yaw and pitch of a human's gaze and the yaw and pitch of the human's head pose. In an experiment we measured head orientations when participants looked at known object locations from two vantage points. The relative position of objects was chosen such that the viewer's gaze elevation, angle, and azimuth were systematically varied. With a linear model, which turns out to be sufficient, we relate the measured head pose to the ground truth of the gaze direction. From this relation the actual gaze direction can be inferred from the measured head pose.

2 Method

2.1 Design

The experiment was set up in a living room environment as illustrated in Figures 2 and 3.

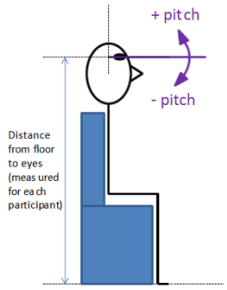


図 2.4: yaw と pitch の定義

1. BlazeFace[6] によって顔とその特徴点（目や鼻など）を検出
2. 1. にて検出した顔とその特徴点を用いて、ResNet の縮小版 (stage3 モデル) によって視線角度を推定

■結果

- 視線角度の推定に成功しました。(図 2.5)

この結果から、視線の角度推定では手法 2 を採用することにいたしました。

2.5 最終結果

図 2.6 に、視線角度推定の最終結果を示します。図中の `middle_left.jpg`・`down_middle.jpg`・`middle_right.jpg`・`top_middle.jpg` は、それぞれ画面の左中央・下中央・右中央・上中央[*3]を見ている状態に対応しており、概ね正確に推定できていることが確認できます。

[*3] 画面上の x 座標・y 座標における PC との関係は図 2.7 に示されている通りです。

図 2.5: 視線角度推定の結果

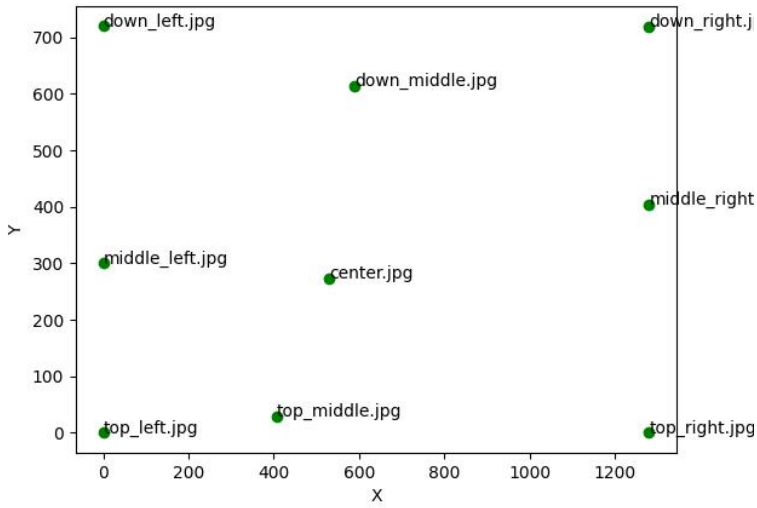


図 2.6: 視線角度推定の最終結果

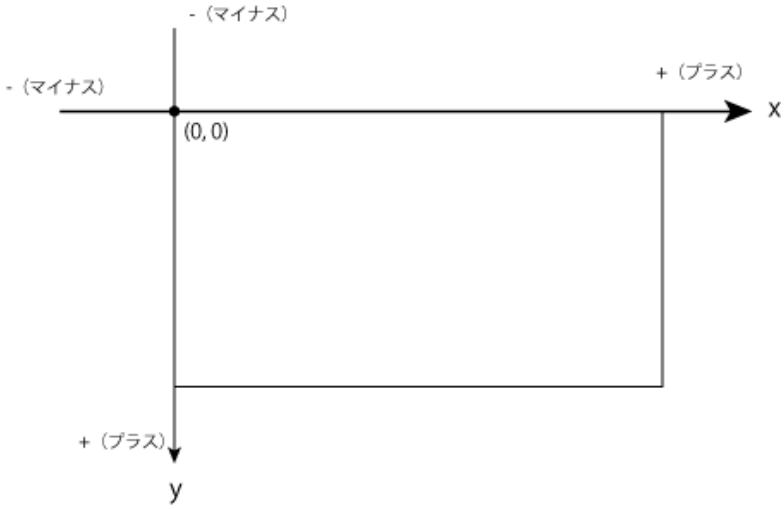


図 2.7: PC 上のスクリーン座標系 (x: 0〜1280, y: 0〜720)

2.6　最後に

最終的にモデルを作成できましたが、2.4.1 節で述べた手法では十分な性能を達成できませんでした。この原因として、学習データに問題がある可能性を考えています。この結果から、モデル構築だけでなく、学習に用いるデータの選定がモデル性能向上において極めて重要であることが分かりました。

他にも、yaw と pitch を推定するための様々なモデルや、値の範囲が広い学習データもありましたが、いずれもデータサイズが大きく、ダウンロードしにくい状況でした。そのため、今後はモデルの改良だけでなく、適切なデータ選定についても学ぶ必要があることが分かりました。

また、今後はデータの yaw や pitch の範囲が $-20° \sim 20°$ と狭いことが影響している可能性があるため、yaw や pitch が $\pm 20°$ に近い角度のデータの比重を高め、モデルが広範囲の視線角度に対しても正確に推定できるようにして実験したいです。

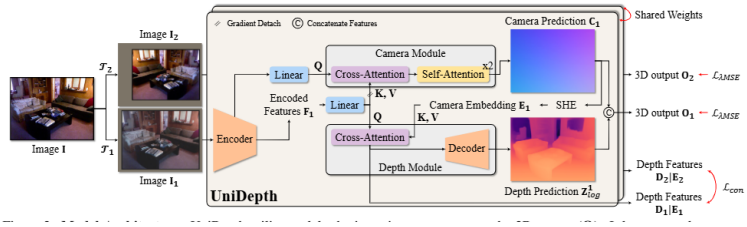


図 2.8: UniDepth の構造 [7]

また、深度推定においては、UniDepth(図 2.8, [7]) などの様々な深度推定モデルが存在します。こういったモデルを活用することで、目とカメラの距離を推定し、それをもとに目とカメラの正確な位置関係を求めるようにしていきたいです。

第 3 章

KaiRA くんを動かそう

3.1 はじめに

当サークルのマスコットキャラクターである KaiRA 君を動かして愛でてみたい、というのがこのプロジェクトを製作した動機です。当プロジェクトでは、ユーザーが自然言語による命令を入力すると、KaiRA 君がそれに従って動いてくれます。命令と動きによる応答の関係を通じて、時には命令にうまく従ってくれないことも含め、KaiRA 君との不思議な交流が楽しめます。

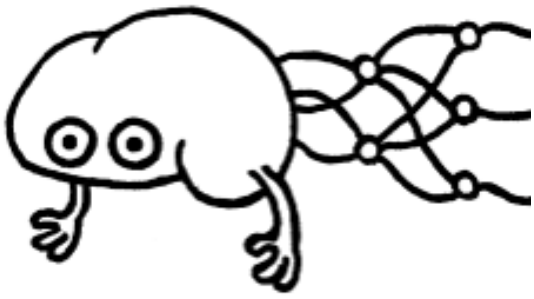


図 3.1: 当プロジェクトで使用した KaiRA 君のイラスト

3.2 モーションの生成

KaiRA 君のモーションを自然言語から生成するために、当プロジェクトでは Human Motion Diffusion Model[8] を用いています。テキストからモーションを生成するモデルはこれ以外にも、MotionGPT[9] 等が挙げられます。しかし、生成されるモーションの質に大きな差が見られなかったので、生成速度の観点からこの手法を選択しました。Human Motion Diffusion Model は、ノイズを 50 ステップという少ないステップ数で取り除いても十分な質のモーションを生成できます。

Human Motion Diffusion Model はその名の通り、基本的な生成の仕組みは拡散モデルに基づい

ています。Transformer のエンコーダ部分を用いて、各時刻における関節の位置や回転が予測されています。ただし、学習に用いられる損失関数には工夫が施されていて、通常の拡散モデルによる手法のものとは異なります。通常はある時刻で与えたノイズと、それを予測したノイズとの間で損失を取りますが、この手法ではノイズが加えられる前のものと、ある時刻でノイズが除去された後のものとの間で損失を取っています。これに、各関節の位置や速度の予測と正解の差を考慮した損失を組み合わせたものを用いて、学習を行っています。

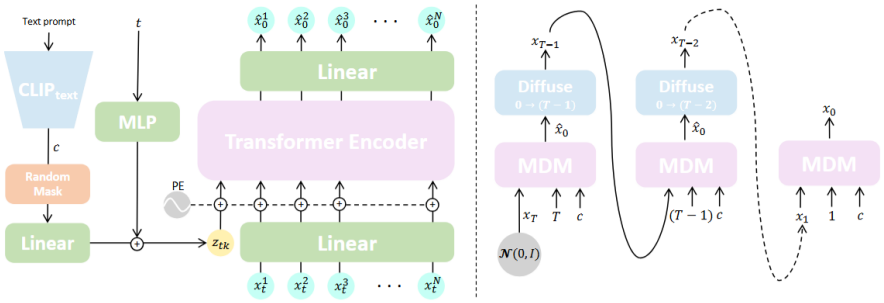


図 3.2: Human Motion Diffusion Model の全体像

Human Motion Diffusion Model で生成されたモーションは、NumPy 配列として出力されます。当プロジェクトでは、出力された NumPy 配列を BVH ファイルに変換したものを Animated Drawings に入力しています。

3.3　アニメーションの生成

生成されたモーションに従って KaiRA 君を動かすために、当プロジェクトでは Animated Drawings[10] を用いています。実のところ、当初は 2D のアニメーションではなく、KaiRA 君の画像から生成した 3D モデルを用いるつもりでした。しかし、LRM[11] や DreamGaussian[12] 等の手法を試したものの、平面的な 1 枚のイラストを 3D にうまく変換することはできませんでした。そのような経緯で、3D モデルを生成することは一旦諦め、KaiRA 君を 2D のイラストのまま動かすことにしました。

Animated Drawings は、キャラクターの関節の位置を 1 枚の画像から推定し、それを基づいてアニメーションを生成することが可能です。しかし、学習に用いられたイラストは人の形をしたものが多いため、KaiRA 君に対して自動で関節を割り当ててもうまく機能しません。また、KaiRA 君は人間のような下半身を持たないため、生成された人間のモーションをそのまま適用すると、胴体がねじれる等の不都合が生じます。そこで当プロジェクトでは下図のように、腕の動きを反映させることを重視し、下半身の関節については画像の下端に配置することで、その影響をなるべく排除しています。

Animated Drawings は BVH ファイルにより、アニメーションの動きを指定することができます。このとき入力されるモーションは 3D ですが、これを平面に投影することでキャラクターの関節と対応させています。ただし、BVH ファイルの関節と、キャラクターの関節との対応関係は手動で設定する必要があります。

Animated Drawings の手法の詳細には立ち入りませんが、これを Human Motion Diffusion Model と組み合わせることで、自然言語の命令により KaiRA 君をアニメーションで動かすことが可能になります。

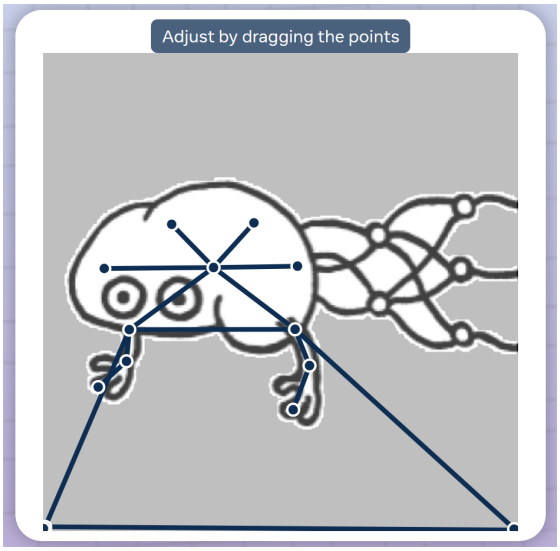


図 3.3: KaiRA 君の関節の配置の設定

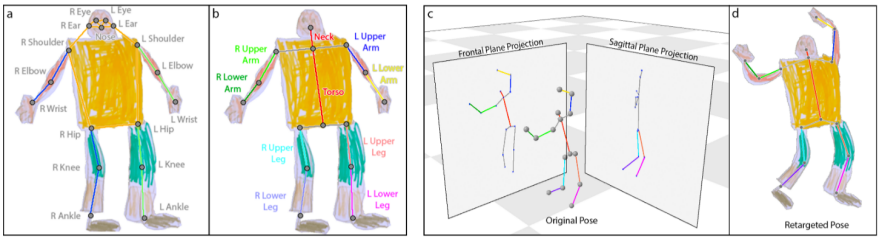


図 3.4: Animated Drawings の全体像

3.4　おわりに

現在の AI はすでに高度な言語処理能力を持つようになりましたが、現実の人間による言語活動は純粋な言語でのみ構成されるものではなく、身体的な要素が多大に含まれています。その意味では、言語と身体を繋ぐ道具として、テキストからモーションを生成する手法は想像以上に重要かもしれません。当プロジェクトでは人間が入力したテキストを用いてモーションを生成していますが、AI が自ら適切なモーションを選択し生成できるようになれば、人間と AI のコミュニケーションはより豊かなものになり得ると思います。

第 4 章

最強じゃんけん AI

4.1　概要

この「じゃんけん AI」では、リアルタイムの映像から手の形を推測し、各時刻ごとに手の動きを予測しています。じゃんけんの大規模な動画データを用意するのが難しく、独自の小規模なデータで学習を行うために、学習済みの骨格推定モデルを用いて映像から手の位置や形を骨格データとして抽出し、骨格データからじゃんけんの手を推測する方法を取りました。この骨格の時系列データをもとに、次の瞬間の手の形状を予測するためにリカレントニューラルネットワーク（RNN）を活用しました。RNN の時系列予測の強みを活かし、じゃんけんの手の変化に応用することで、手を出し終わる前に予測を行い、後出しではなく自然な流れでじゃんけんができるよう工夫しています。

4.2　骨格推定モデル

骨格推定とは、人間の関節や目、鼻などの特徴点（ランドマーク）の位置を推定する技術であり、深度センサーや慣性センサーなどの高度な機器を用いた高精度な 3 次元推定から、通常のカメラを用いた 2 次元画像からの推定まで、幅広い手法が存在します。代表的な骨格推定モデルには、カーネギーメロン大学の「OpenPose[13]」や Google の「MediaPipe[14]」、および「PoseNet[15]」などがあります。今回のじゃんけん AI では、リアルタイムに手の動きを認識する必要があるため、「MediaPipe」を採用しました。MediaPipe は、他のライブラリと比較して高速なリアルタイム処理に優れており、手のみの骨格推定を行うことが出来ます。手の骨格推定では、手の写った画像から 21 個の関節位置を検出し、深度推定を用いて 3 次元座標に推論することができます。

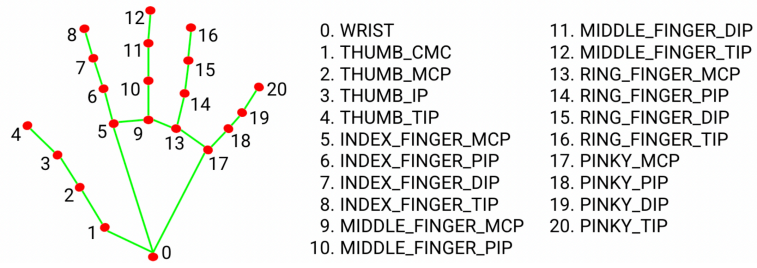


図 4.1: mediapipe が検出する手の骨格 [14]

4.3 実験

4.3.1 データ

学習データとして、独自に用意した 246 本のじゃんけん動画を使用しました。データ作成時には、多様性を確保するために、さまざまな手の握り方や角度を含めるよう意識しました。データ数が少ないため、手の平行移動を利用したデータ拡張を行ったところ、精度が大幅に向上しました。また、右手のデータのみを用いていたため、リアルタイム推論時に左手の認識がやや不安定になるという問題がありましたが、左右反転のデータ拡張を加えることで、左手も右手と同等の精度を実現しました。

4.3.2 時系列予測

骨格の時系列データの処理には、RNN・LSTM・GRU を用いて精度と推論時間の比較を行いました。時系列データを扱う強力なモデルとして Transformer も考慮しましたが、今回は小規模なデータセットによる学習であることから使用しませんでした。

最初は、手を出す 2 秒前 (60 フレーム分) のデータで学習を行い、精度は表 4.1a のようになりました。手を出す直前での精度に違いは見られませんでしたが、タイミングが離れるほど RNN の精度が低下し、LSTM と GRU に関しては、1 秒前まではほぼ同等の精度となりますが、2 秒前になると LSTM の方が少し精度が良くなります。しかし、学習時の 2 秒前の精度の高さに反し、リアルタイムで推論を行う際には、手を出す前はほとんどが「グー」となり、さらに推論が不安定になってしまいました。

手の形が変化しだすのが、手を出す約 0.15 秒前であることから、学習データとして手を出す瞬間の前後約 0.2 秒の 15 フレームを学習させることにしました。この 15 フレームでの学習では、表 4.1b のように RNN,LSTM,GRU 全てで全時刻精度 1 となり、リアルタイム推論でも安定して精度よく推論できるようになりました。

	RNN	LSTM	GRU
2 秒前	0.596	0.914	0.876
	0.617	0.965	0.943
⋮	⋮	⋮	⋮
	0.684	0.979	0.980
1 秒前	0.676	0.979	0.980
	0.664	0.979	0.980
⋮	⋮	⋮	⋮
	0.984	0.985	0.985
ポン	0.984	0.985	0.985

(a) 60 フレーム学習の精度比較

	RNN	LSTM	GRU
−7 フレーム	1.00	1.00	1.00
	1.00	1.00	1.00
⋮	⋮	⋮	⋮
	1.00	1.00	1.00
ポン	1.00	1.00	1.00
	1.00	1.00	1.00
⋮	⋮	⋮	⋮
	1.00	1.00	1.00
+7 フレーム	1.00	1.00	1.00

(b) 15 フレーム学習の精度比較

4.3.3　推論速度

推論速度としては、全データを推論するのにかかる時間は表 4.2 のようになり、全モデルで GPU よりも CPU での推論が早く、CPU での推論時間は RNN>GRU>LSTM となりました。15 フレームの学習では精度が変わらないので、CPU での推論速度が一番速い RNN を採用しました。

	RNN	LSTM	GRU
GPU	241	202	243
CPU	74	135	114

表 4.2: GRU

4.4　結論

今回のじゃんけん AI では、時系列予測を活用して後出しではない自然な動作を目指し、最終的に手を出し終わる前に正しい手を予測するモデルを開発することが出来ました。しかし、実際にじゃんけんをする際には、画面とのタイミングを人間側で合わせるのが難しく、モデル以外の部分に課題が残ります。今後は、人と AI のタイミングをさらに自然にする工夫を加え、よりスムーズなじゃんけん AI を目指していきたいと思います。

第 5 章

京大シラバス検索 RAG システム - システム概要編

5.1 京大シラバス検索 RAG システムの基本情報

5.1.1 シラバスと RAG システム

シラバスとは、京都大学情報教務システム KULASIS が公開している京都大学の科目の情報が載っているページである [16]。これは京都大学の学生・職員でなくとも閲覧することができる。シラバス検索というページから、知りたい科目の条件（学部、学科、曜時限等）で検索をして、各科目のページに飛ぶことができる。各科目のページには以下のような項目がある。括弧の中の項目は一部の科目にある。

- 科目ナンバリング、科目名、英訳
- 所属部局・職名・氏名
- 使用言語、単位数、授業形態、開講年度・開講期、配当学年、対象学生、曜時限、（時間数）、（キーワード）
- 授業の概要・目的、到達目標、授業計画と内容、（題目）
- 履修要件、成績評価の方法・観点、教科書、参考書等、授業外学修、（関連 URL）、（実務経験のある教員による授業）

RAG（Retrieval-Augmented Generation）とは、LLM（大規模言語モデル）の文章生成に外部データの検索機能を組み合わせる手法である。似た手法にファインチューニングがあるが、これは事前学習された LLM をさらに外部データで学習させる手法である。今回は外部データを京大シラバスとして、RAG システムを実装した。また、1 つの検索対象のテキストの集まりを**チャンク**と呼ぶ。

今回の実装では、**データの埋め込み**（文章をベクトル化すること）の計算が多く、前処理を変更するたびに埋め込みをすると時間がかかるので、大学院の専門科目（約 10000 科目）を除いた全学共通科目と学部の専門科目（約 8000 科目）だけを利用した。さらに、この中で医学部医学科だけがシラバスが PDF で個別の処理が必要なので、医学部医学科の科目も除外した。最終的に外部データとしたのは、全学共通科目と医学部医学科以外の学部の専門科目となった。

5.1.2　実装の優先事項と概要

今回の実装をするに当たって、自分が最も重要視したことが実用性である。RAG システムの基本的な流れは、ChatGPT などと同様に質問を入力して、質問のテキストと類似度が高い外部データのテキストを抽出し、質問と抽出したデータを入力として LLM で回答を生成する。今回の実装した RAG システムは、京都大学の学生の要求に対応した科目の検索ができるようになることを想定している。従って、質問と類似度が高い外部データのテキストを抽出する前に、外部データの**絞り込み**が必要だと考えた。つまり、シラバス検索ページのような科目の条件（学部、学科、曜時限等）で絞り込みをする機能を RAG システムに組み込む形で実装をすることに決めた。

実装で使用したプログラミング言語は Python で、主に使用したライブラリは、LLM の機能拡張をするための Langchain[17]、外部データの埋め込みを扱える FAISS[18] と、前処理をする際に HTML からテキストを抽出するための Beautiful Soup[19] と、Python で簡単に Web アプリを実装できる Streamlit[20] である。LLM のモデルは Google が開発した Gemini[21] を使用しており、無料で API を利用できる。埋め込みのモデルは HuggingFaceEmbeddings の多言語に対応している `intfloat/multilingual-e5-base`[22] を使用した。

5.2　絞り込み

5.2.1　KULASIS の絞り込みのための検索項目

科目の条件で絞り込みを行うときに参考にするのは、もちろん KULASIS のシラバス検索だが、これには京都大学の学生・職員でなくとも利用できる**全科目のあるシラバス検索**と、京都大学の学生・職員がログインして利用できる**全学共通科目だけのシラバス検索**と、スマートフォンでの **KULASIS アプリで利用できるシラバス検索**が存在し、少し検索項目に違いがあることが分かった。それぞれの検索項目を表にすると表 5.1 のようになった。

説明が必要だと感じた検索項目は以下のようになる。

- 課程 — 学部、大学院
- 旧群 — A 群、B 群、C 群、D 群 (平成 25 から 27 年の入学者向け)
- 開講期 — 前期、後期、前期集中、後期集中など
- 授業形態 — 講義、演習、実験など
- E 科目 — E1、E2、E3
- 対象学生 — 全学向、文系向、理系向、留学生
- レベル — 導入的な内容、基礎的な内容、発展的な内容、卒業論文・卒業研究関連など
- 学問分野 — 情報学基礎、地球環境学、哲学、言語学など
- 科目名 — 科目名に含まれる文字を入力する
- キーワード — シラバスに載っている言葉・文字を入力する
- 教員名 — 担当教員に含まれる文字を入力する
- 実務経験科目 — 実務経験科目の主要な 4 つ形式の科目とその他の科目

シラバス検索から絞り込みに使う検索項目について以下のように整理して考えた。

- 各科目のシラバスのページ以外にシラバス一覧のページから細かい科目の区分の情報が得ら

表 5.1: シラバス検索の検索項目

	全科目のシラバス	全学共通科目のシラバス	アプリのシラバス
学部／大学院	○	-	○
課程	×	○	○
学科	○	-	○
群	○	○	○
旧群	×	×	○
開講期	○	○	○
曜時限	○	○	○
授業形態	○	○	○
E 科目	×	○	○
使用言語	○	○	○
対象学生	×	○	○
レベル	○	○	○
学問分野	○	○	○
科目名	×	○	○
キーワード	○	○	○
教員名	○	○	○
実務経験科目	×	○	○

れる。

  - 全学共通科目の群の中でも、例えば人文社会科目の中に哲学・思想、歴史・文明、地域・文化などのさらなる区分が存在する。これらの区分を**分野**とする。
  - 学部の中には、工学部のように地球工学科や物理工学科などの学科の区分や、総合人間学部のように人間科学系や国際文明学系などの学科以外の区分が存在する。これらの区分を**学科など**とする。
- **レベル**と**学問分野**は各科目のシラバスのページに情報が載っていないので、前処理でこの情報を取り出せず、絞り込みに利用するのは困難である。
- **課程**はそもそも今回の実装では大学院の専門科目を除外しているので、絞り込みに利用する価値が低い。
- **旧群**は現在適用される人がいないので絞り込みに必要ない。
- **科目名**は**キーワード**に包含されることが可能である。
- **実務経験科目**は選択肢の文章が長く見栄えが悪い上に、そもそも実務経験科目が少なく絞り込みに利用する価値が低い。

以上のことから KULASIS のシラバス検索から絞り込みに使う検索項目は、**学部、学科など、群、分野、開講期、曜時限、授業形態、E 科目、使用言語、対象学生、キーワード、教員名**とした。

5.2.2 追加した絞り込みのための検索項目

今までは、検索項目はシラバスのページから容易に抽出できることが前提だった。しかし、LLM を利用すれば、各科目のシラバスの内容から新たに検索項目を生成できる。どのように検索項目を生

成するのかを詳しく説明すると、入力に対する出力が必ず JSON というデータ形式になるモードである **JSON モード**を使用する。検索項目として生成するのは、KULASIS の検索項目で抽出が困難だとして断念した**レベル**や**学問**の他に、シラバスの履修要件にある履修していることが望ましいとされた科目や、シラバスの成績評価の方法・観点にある定期試験や平常点などの評価指標の占める成績評価の割合などが考えられる。

しかし、検索項目を生成するには API へのリクエストが必要であり、Gemini API の無料枠の場合、1 分間に 15 リクエスト、1 日に約 1500 リクエストという上限がさだめられている。これでは、約 8000 個のシラバスのデータから検索項目を生成するには、多くの実行時間や日数が必要となる。これらを削減する手段として以下のようなものがある。

1. 複数の API キーを使用する
2. 同時に複数の科目のテキストを入力として検索項目を生成する
3. 同時に複数の種類の検索項目を生成する

今回の実装では、まずは検索項目は有用性の観点から成績の評価指標の割合を追加することにしたが、手法 2 を試すと複数の科目が、例えば哲学 1 とその他の哲学 1 のように科目の内容が被ったときに、出力が 1 つになって上手く行かなかった。時間の兼ね合いもあり、検索項目は有用性の観点から**成績指標の割合**だけを追加した。

5.2.3　絞り込みの方法 1 metadatas

絞り込みの検索候補は決定した。次に絞り込みの方法を説明する。LangChain の FAISS には、埋め込み時に辞書形式の `metadata` をチャンクごとに登録できる「`metadatas`」という機能がある。この機能には主に以下の 2 つの役割がある:

1. **フィルタリング**: 指定した `metadata` のキーと値を持つチャンクに限定して類似度検索を実行する機能。
2. **情報の取得**: 検索結果から特定の `metadata` のキーを指定し、対応する値を引き出す機能。

例えば、質問文を `query`（`int` 型）、外部データを `texts`（`list` 型）として、`metadatas` を活用した RAG を実行する場合、以下のような流れになる。

コード 5.1: metadatas の例

```
metadatas = [{"name":"philosophy",classtype":"lecture","timetable":"Tu2"},
             {"name":"thermodynamics","classtype":"lecture","timetable":"Fr2"},
             {"name":"physics experiment","classtype":"experiment","timetable":"
                 Mo3,Mo4,Tu3,Tu4"}]
store = FAISS.from_texts(texts,embedding,metadatas)
a = store.similarity_search(query,filter={classtype":"lecture","timetable":"Tu2
    "})
for i in range(len(a)):
    print(a[i].metadata["name"])
```

5.2.4 絞り込みの方法 2 delete

metadatas の機能は便利だが、metadatas による絞り込みでは複数の key を指定すると AND 検索になるので、OR 検索ができないという欠点がある。これを解決するために次のような手法を考えた。Langchain ではデータの埋め込みに時間がかかるので、埋め込み後にデータを絞り込む必要がある。ここで埋め込み後のデータを **FAISS ファイル**と呼ぶ。FAISS ファイルを扱えるのは LangChain の関数だけなので、その中から OR 検索のために delete 関数を利用する。delete 関数は、FAISS ファイルから削除したい ID をリストで指定して削除する関数である。削除したい ID を deleteID とすると、delete 関数を用いた RAG の実行は以下のようになる。

コード 5.2: delete の例

```
metadatas = [{"name":"philosophy"},{"name":"thermodynamics"},
             {"name":"physics experiment"}]
ids = [str(i) for i in range(len(texts))]
store = FAISS.from_texts(texts,embedding,metadatas,ids = ids)
deleteID = ["1","2"]
store.delete(deleteID)
a = store.similarity_search(query)
for i in range(len(a)):
    print(a[i].metadata["name"])
```

このように delete 関数を用いることで、AND 検索と OR 検索を自由に行うことができる。

- **AND 検索** — 複数の条件に当てはまる ID の集合の**和集合**を削除する
- **OR 検索** — 複数の条件に当てはまる ID の集合の**積集合**を削除する

詳しく検索手法を知りたい場合は、GitHub の実装を確認してほしい。

5.3 前処理

5.3.1 前処理の目的

前処理の目的は 2 つある。1 つ目は、質問と外部データの類似度を計算するときに、外部データに科目に特有の情報だけがあることが望ましいので、外部データの共通したテキストや絞り込みで利用したテキストを除去することだ。2 つ目は、説明した絞り込みの項目と方法に適したデータを外部データから抽出することだ。

大まかな前処理の流れは以下のようになる。

1. 全科目の URL を取得する
2. URL から HTML を取得する
3. 絞り込みのためのデータを抽出・生成する
4. 類似度検索のためにデータをまとめる

5.3.2　URL の取得

まず、URL を取得するために、KULASIS のシラバス一覧のページから全科目の URL を Beautifulsoup で取得する。医学部医学科の科目の URL は共通しているので除去する。人間総合学部の英米文学入門が学科の専門科目の最後なのでこれを取得したら実行を終了する。プログラムは以下のようになる。

コード 5.3: URL の取得

```
url1 = "https://www.k.kyoto-u.ac.jp/external/open_syllabus/all"
response = requests.get(url1)
soup = BeautifulSoup(response.content, "html.parser")
urls = []
for i in range(3,len(soup.find_all("a"))):
    url = "https://www.k.kyoto-u.ac.jp/external/open_syllabus/"+str(soup.
        find_all("a")[i].attrs["href"])
    if url != "https://www.k.kyoto-u.ac.jp/external/open_syllabus/https://www.
        med.kyoto-u.ac.jp/for_students/affairs_m/class/":
        urls.append(url)
    if url == "https://www.k.kyoto-u.ac.jp/external/open_syllabus/
        department_syllabus?lectureNo=10192&departmentNo=61":
        break
```

この後、取得した URL ですべての科目の html をリクエストする。これに 3 時間ほどかかった。科目の取得する順番も ID として後で利用するので、ID を保存しながら非同期処理をするともっと早く実行できるかもしれない。

5.3.3　絞り込みのためのデータを抽出・生成

絞り込みのためのデータの形は以下の 4 パターンがある。

- ラベルデータ — 科目のラベルが `int` で保存されている。
- 複数のラベルデータ — 科目の複数のラベルが `list` の中に `int` で保存されている。
- テキストデータ — 科目に関するテキストが `int` で保存されている。空白で AND 検索できるようにする。
- 辞書データ — 科目の情報が辞書形式で保存されている。

以下が全ての絞り込みのためのデータである。

- **学部、群、学科など、分野**
  各科目の html を取得した時の ID と URL の ID から、科目の**学部、群、学科など、分野** を振り分けた。**学部、群、分野**はラベルデータ、**学科など**は複数のラベルデータとした。
- **曜時限、授業形態、使用言語、開講年度・開講期、対象学生**
  各科目の html からそのまま取得した。**曜時限**は月 1 から金 5 と集中、**授業形態**は「講義」「演習」「実習」「実験」「特殊講義」「語学」「講読」「卒業研究」「ゼミナール」の単語を含むか、**使用言語**は「日本語」「英語」「日本語及び英語」「その他」で振り分けを行った。**曜時限、授業形態**は複数のラベルデータ、**使用言語、開講年度・開講期、対象学生**はラベルデータとした。

- **E 科目**
  各科目の科目名に E1、E2、E3 のいずれかを含むかで振り分けた。**E 科目**はラベルデータとした。
- **キーワード、教授・教員**
  キーワードはシラバスの科目名、授業の概要・目的、到達目標、授業計画と内容、題目、キーワード、履修要件、成績評価の方法・観点、教科書、参考書等のテキストから成る。**教授・教員**はシラバスの所属部局、職 名、氏 名から成る。いずれもテキストデータとした。
- **成績評価**
  5.2.2 節で紹介した追加した絞り込みのための検索項目であり、JSON モードでシラバスの成績評価の方法・観点のテキストを gradetext として、以下のようなプロンプトで辞書データとして生成した。

  f"次の JSON スキーマを使用して、{gradetext}の成績評価の方法・観点について、平常点、課題、発表、討論、小テスト、小レポート、期末レポート、期末試験のいずれか占める割合をリストアップしてください。seiseki = {'平常点': int, '課題': int, '発表': int, '討論': int, '小テスト': int, '小レポート': int, '期末レポート': int, '期末試験': int} Return: seiseki"

5.3.4 類似度検索のためにデータ

類似度検索のためにデータは、科目名、授業の概要・目的、到達目標、授業計画と内容で構成することにした。RAG での精度向上のためにチャンクをどのように分割するかが議論になるが、京大シラバス RAG システムにおいては 1 科目を複数のチャンクに分割すると情報が局所的となりテキストの全体から類似度を計算した方が良いと感じたので、科目とチャンクが一対一対応となるようにした。

5.3.5 その他の前処理

その他の前処理として以下のような処理をした。

- metadata の登録
  絞り込みは delete 関数で行うが、要約を生成するときに metadata があると便利なので、「科目名」「URL」「ID」を登録した。
- 要約生成のためのデータ
  要約生成のためのデータは、シラバスの情報をほぼ全てテキストにした。要約生成のプロンプトもそこまで工夫せずに「以下の文章を日本語で箇条書きで要約を生成してください。」とした。

5.4 改善案

今回の実装でできなかった改善案は以下のようになる。

- さらに絞り込みのための検索項目を追加する。例えば、履修していることが望ましいとされた

科目や科目のレベルなど。

- チャンク分割をもっと工夫する。
- 要約生成が安定しないときがあるので、要約のためのデータや要約生成のプロントを工夫する。
- 教科書、参考書の情報も検索に組み込む。
- シラバスの要約を外部データとして埋め込む。

第 6 章

京大シラバス検索 RAG システム - 検索手法編

6.1 探索方法の基本情報と目的

今回は埋め込みベクトル用いる方法と、単語出現頻度を用いる手法 の 2 つの探索手法を実験した。

6.1.1 埋め込みベクトルによる方法

この手法は、文書を多次元のベクトル空間上のベクトルとして数値表現したデータを用いて、ベクトル間の距離や類似度によって文書を検索する。

今回は `HuggingFaceEmbeddings` を用い埋め込みを行っている。埋め込みの前処理と計算はすでにしていただいたデータを用いて、ここではその後の検索手法について実験している。具体的には、`langchain` の `vectorstore` の `as_retriever` メソッドに備え付けられているコサイン類似度 (similarity) による検索と、Maximal Marginal Relevance (MMR) による検索を実験した。

6.1.2 単語出現頻度に基づいた検索手法

文書を単語分割したデータを用いて、単語の出現頻度に基づいた `BM25` という手法を実験した。

文書の単語分割には、Python のライブラリ `rank_bm25` に標準で使用されている `split()` 関数が日本語に対応していないため、Python の日本語形態素解析エンジンである `janome` を用いている。

6.2 埋め込みベクトル探索方法の実験

6.2.1 手法

チャンク分けの仕方が異なるデータセット 3 つに対して、コサイン類似度と mmr の探索手法をいくつかのクエリに対して試す

- data1
- data2
- data3

6.2.2　コサイン類似度 (similarity)

コサイン類似度とは、２つのベクトルがどの程度似ているか表す尺度である。２つのベクトルの内積を２つのベクトルの大きさで割ることで得られる。

6.2.3　MMR(周辺関連性最大化:Maximal marginal relevance)

mmr とは、選択される文書の多様性を広げる検索手法である。クエリーに関連性を持ち、かつ、以前に選択された文書との類似が最小である場合に周辺関連性 (marginal relevance) は高くなる。

ここでいう周辺関連性とは、関連性と新規性を独立して計測し、線形結合した関連新規性（relevance novelty）という新たな視点を反映した尺度である。潜在的に優れた文書であるが、ユーザーとクエリの関連性という基準だけでは埋もれてしまう文書を見つけるためにある。

6.2.4　結果のまとめ

- 出力の厳密性は、similarity がよく反映する。キーワードを外さず同じような授業を出すという点で date3 の similarity は厳密性が一番高いように思われる。
- 多様性を求め、キーワードでは見つからないような授業を探すという基準では、やはり mmr がうまくいく。data1-3 の mmr の結果は特筆して変わることはないように思われる。

6.3　単語出現頻度に基づく検索手法の実験

6.3.1　BM25 （Best Matching 25）とは

BM25 は、情報検索において文書の関連性を評価する上で広く用いられる手法である。文書内の単語の出現頻度（tf）と、その単語がコーパス内のどのくらいの文書に含まれているかを示す逆文書頻度（idf）を組み合わせることで、各単語の重要度を数値化する。この計算式には、k1 と b という調整可能なパラメータが含まれており、特に k1 は、tf の重み付けを調整する役割を持つ。k1 の値が大きいほど、tf の影響が大きくなり、単語の出現頻度が高い文書ほど高いスコアが得られやすくなる。今回はそのパラメータを調節しどの程度が良いか結論づける。

6.3.2　実験結果・考察

具体的である 1 つ目の質問、「電気回路を学べる科目はなんですか」を参考にすると、$k = 5$ で全く関係のない科目が消えるため、$k = 5$ がこのパラメータとしては適切なのではないかと思われる。BM25 はキーワードが合致していないとなかなか適合せず、抽象的な質問には答えることができないようで、シラバス RAG の検索手法でそのまま BM25 を用いるのは、実用としては難しいと思った。そのため、キーワードを抜き出し検索したり、生成 ai でシラバスに含まれそうなキーワードを増幅したりしながら、抽象的なクエリに対して検索をもっと工夫する必要があると思った。

質問/応答	similarity	mmr
電気回路を学べる科目はなんですか？	電気・電子工学電気電子回路電気電子回路演習電気電子回路入門	電気・電子工学電磁気学 B 電気電子回真空電子工学
機械学習を学べる授業は何ですか？	機械システム学セミナー（機）パターン認識と機械学習機械学習学術連携共同：数理科学の研究フロンティア	機械システム学セミナー（機）学術連携同：数理科学の研究フロンティアパタン認識と機械学習 ILAS セミナー ボットとの未来を考える
脳神経について学べる科目**を教えてください**	神経科学の基礎記憶機能論神経心理学 I（神経・生理心理学）神経心理学 I	神経科学の基礎生物科学課題研究 19経生理学基礎演習：神経心理学
脳神経について学べる科目**は何ですか？**	神経科学の基礎物科学課題研究 19。記憶神経科学ゼミ B記憶神経科学ゼミ A	神経科学の基礎生物科学課題研究 ILAS セミナー ：霊長類脳神経科学レーニングコース。神経生理学
神経科学について学べる科目はなんですか？	生物科学課題研究 19 神経科学の基礎神経生理学 I 神経生理学	生物科学課題研究 19 神経科学の基礎経生理学生物科学特別講義 2
日本文学に関する授業は何がありますか？	日本語学・日本文学演習 IIB 日本語学・日本文学演習 IIA 日本語学・日本文学演習 IV 国語国文学 II	日本語学・日本文学演習 IIB 日本語日本文学演習 IV 英米文芸表象論演習日本史学 (特殊講義
日本文学を学べる授業は何がありますか？	日本語学・日本文学演習 IIB 日本語学・日本文学演習 IIA 国語学国文学 (演習) 日本語学・日本文学演習 IIIB	日本語学・日本文学演習 IIB 日本語日本文学演習 IV 英米文芸表象論演習日本の歴史と文化
心理学**の入門科目**は何がありますか？	心理学（実習 IA）(心理学実験) 心理学概論心理学概論心理学概論	心理学（実習 IA）(心理学実験) 心理学論言語科学入門（認知情報学系入門科社会心理学（社会・集団・家族心理学
ビジネス**関連の授業**は何がありますか？	企業分析商法（総則・商行為）ビジネスエシックス商法（会社	企業分析商法（総則・商行為） ビジネエシックスアントレプレナーシップ特
生物学の**実験を含む**授業は何ですか？	生物学実習 I ［基礎コース］生物学実習 I ［基礎コース］ 生物学実習 I ［基礎コース 生物学実習 I ［基礎コース］	生物学実習 I ［基礎コース］生物科学題研究 19。分子細胞生物学演習生物科課題研究 22
環境問題について学べる授業**を探しています。**	環境学国際環境政治学 基礎地球科学 B（地球システムと環境）演習 (4 回生) テーマ：エネルギー	環境学 演習 (4 回生) 地球生存リスク特環境法
環境問題について学べる授業は何ですか？	環境学環境経済論演習 (3 回生) 環境問題に関する経済学的研究演習 (4 回生) テーマ: 環境とエネルギーの経済学	環境学 演習 (3 回生演習 (4 回生) 大気地球環境工学

表 6.1: data1 の質問応答表

data2 の質問応答表	similarity	mmr
電気回路を学べる科目はなんですか？	電気・電子工学電気電子回路電気電子回路入門電気回路	電気・電子工学電気電子回路電気回路基礎論
機械学習を学べる授業は何ですか？	機械学習パターン認識と機械学習データ分析演習 I 人工知能	機械学習機械製作実習（機）データ分析演習 I 機械学習
脳神経について学べる科目を教えてください	神経科学の基礎心理学 (特殊講義 A)（神経・生理心理学）神経心理学 I （神経・生理心理学）神経心理学 I	神経科学の基礎 ILAS セミナー ：神経心理学神経生理学の基礎 －生体情報論－霊長類学入門 I
脳神経について学べる科目は何ですか？	神経科学の基礎神経生理学の基礎－生体情報論－神経科学の基礎記憶神経科学ゼミ B	神経科学の基礎神経生物学記憶神経科学ゼミ A 神経生理学の基礎－生体情報論－
神経科学について学べる科目はなんですか？	神経科学の基礎 神経科学の基礎神経生物学神経生理学の基礎 －生体情報論	神経科学の基礎神経生理学神経生物学神経生物学の
日本文学に関する授業は何がありますか？	日本の歴史と文化 日本語学・日本文学演習 IV A 日本語学・日本文学演習 IV B メディア文化学 (特殊講義)	日本の歴史と文化言学 I 基礎演習: 日本近代文学日本語学・日本文学 IIIA
日本文学を学べる授業は何がありますか？	基礎演習：日本近代文学基礎演習：日本近代文学日本近代文学 II 日本近代文学 II	基礎演習：日本近代文学日本語学・日本文学演習 IIIB 日本語学・日本文学演習 IV A 日本語学・日本文学 IIIA
心理学の入門科目は何がありますか？	心理学概論心理学概論系共通科目 (心理学)(講義 I) 基礎演習：社会心理学	心理学概論 ILAS セミナー ：社会心理学心理学 (演習) (心理演習) 基礎演習：社会心理学
ビジネス関連の授業は何がありますか？	商法（総則・商行為）商法（会社）商法（会社）【旧商法第二部】	商法（総則・商行為）ビジネスのための情報システム AI 技術利活用実践 ビジネスエシックス
生物学の実験を含む授業は何ですか？	細胞と分子の基礎生物学実験実験動物学生物学実習 I ［基礎コース生物学実習 I ［基礎コース］	細胞と分子の基礎生物学実験生物・生命科学入門 実験動物学生物学実習 B
環境問題について学べる授業を探しています。	環境学環境学統合科学 ：持続可能な地球社会をめざして環境動態学	環境学演習 (3 回生環境と法自然と環境の化学
環境問題について学べる授業は何ですか？	環境学 環境学 環境と法 環境と法	環境法 森林環境学

表 6.2: data2 の質問応答表

data3 の質問応答表	similarity	mmr
電気回路を学べる科目はなんですか？	電気回路基礎論電気回路電気・電子工学電気電子回路入門	電気回路基礎論電気電子回路電気・電子工学電気回路
機械学習を学べる授業は何ですか？	機械学習機械学習パターン認識と機械学習パターン認識と機械学習	機械学習機械製作実習（機）パターン認識と機械学習機械システム学セミナー（機）
脳神経について学べる科目を教えてください	神経科学の基礎神経科学の基礎心理学 (特殊講義 B)（神経・生理心理学）神経生理学の基礎 －生体情報論－	神経科学の基礎心理学 (特殊講義 B)（神経・生理心理学）神経生理学の基礎 －生体情報論－系共通科目 (心理学)(講義 Kc)（知覚・認知心理学）
脳神経について学べる科目は何ですか？	神経科学の基礎神経科学の基礎神経科学の基礎 神経生理学の基礎 －生体情報論－	神経科学の基礎神経生物学記憶神経科学ゼミ A 神経生理学の基礎 －生体情報論－
神経科学について学べる科目はなんですか？	神経科学の基礎神経科学の基礎神経科学の基礎 神経科学の基礎	神経科学の基礎神経生物学神経生理学 I 記憶神経科学ゼミ A
日本文学に関する授業は何がありますか？	日本語学・日本文学演習 IIB 日本語学・日本文学演習 IV B 日本語学・日本文学演習 IIB 日本語学・日本文学演習 IV A	日本語学・日本文学演習 IIB 日本語学・日本文学演習 IV B 日本の歴史と文化 日本語学・日本文学演習 IV B
日本文学を学べる授業は何がありますか？	日本語学・日本文学演習 IIB 日本語学・日本文学演習 IIB 日本語学・日本文学演習 IIA 日本語学・日本文学演習 IV B	日本語学・日本文学演習 IIB 本語学・日本文学演習 IV B 国語学国文学 (演習) 日本の歴史と文化
心理学の入門科目は何がありますか？	基礎演習：社会心理学心理学概論系共通科目 (心理学)(講義 Kc)（知覚・認知心理学）心理学概論	基礎演習：社会心理学心理学概論系共通科目 (心理学)(講義 I) 心理学概論
ビジネス関連の授業は何がありますか？	商法（総則・商行為）ビジネスエシックス起業と事業創造 商法（総則・商行為）	商法（総則・商行為）演習 (4 回生 医療ビジネス・イノベーション概論 Business English-E3
生物学の実験を含む授業は何ですか？	生物学実習 I ［基礎コース］細胞と分子の基礎生物学実験個体の基礎生物学実験生物先端科学実験及び実験法 II	生物学実習 I ［基礎コース］個体の基礎生物学実験生物物理学分子生物学実験及び実験法
環境問題について学べる授業を探しています。	環境学国際環境政治学環境学環境法	環 境 学 Human-environmental Interactions-E2 環境法基礎地球科学 B （地球システムと環境）
環境問題について学べる授業は何ですか？	環境学環境学環境学 環境と法	環境学環境法大気・地球環境工学演習 (4 回生)

表 6.3: data3 の質問応答表

質問/応答	$k = 1$	$k = 3$	$k = 5$	$k =$
稀な文書の重みが強すぎそうである	フランス語の順位がだんだん下がってきた	一個目の質問でフランス語が消えた		
電気回路を学べる科目はなんですか？	電気電子回路入門 フランス語 II B F2155 応用生命科学入門 I 電子回路 ビルマ（ミャンマー）語 I（初級）	電気電子回路入門 電気・電子工学 電気電子工学基礎実験 フランス語 II B F2155 電気機器基礎論	電気電子回路入門 電気・電子工学 電気電子工学基礎実験 電気回路基礎論 電気電子回路	電気 電気 電気 電気 電気
機械学習を学べる授業は何ですか？	学術連携共同：数理科学の研究フロンティア 英語リーディング ER27(技能領域) アカデミックリーディング 英語リーディング ER26(技能領域) アカデミックリーディング ヒューマンインタフェースの心理と生理 フランス語 II B	フランス語 II B F2155 英語リーディング ER27(技能領域) アカデミックリーディング 行動生態学入門 ヒューマンインタフェースの心理と生理 ILAS セミナー ：障害とは何か	フランス語 II B F2155 行動生態学入門 英語リーディング ER27(技能領域) 地震学 ヒューマンインタフェースの心理と生理	行動 フラ 英語 地震 ILAS

表 6.4: 質問応答表の結果

第 7 章

お絵描き予測 AI

7.1 概要

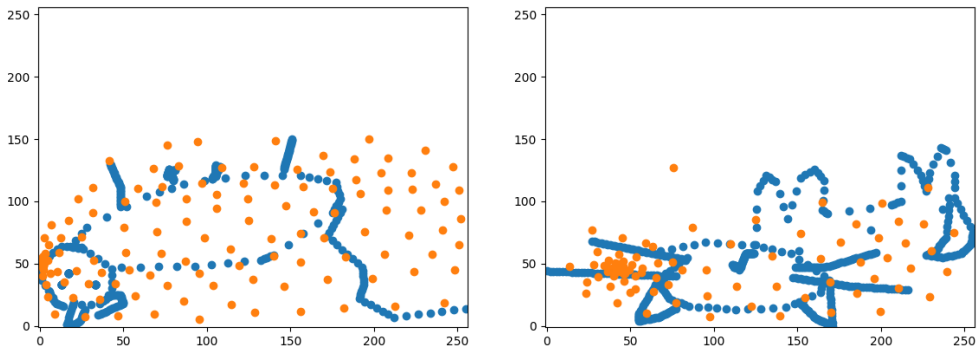
この「お絵描き AI」では、人間が途中まで書いた絵をもとに、完成形を予測して残りの部分を描き足すモデルの作成を目指しました。学習データとして、Google が提供する Quick, Draw！ [23] のデータセットを使用しました。このデータセットは、ユーザーが指定されたお題に対して 20 秒以内に絵を描き、それを AI が判定するというウェブアプリから収集されたものです。全 345 クラス、合計 5000 万件以上のスケッチデータが含まれており、それぞれが連続する点の座標データとして記録されています。これを基に、点群、画像、系列データの各アプローチからモデル構築を試みました。

7.2 点群

7.2.1 実験

Quick Draw！ [23] のデータ形式が連続する点による表現であるため、絵の描き順に依らない点群としてのアプローチが有効だと考え、エンコーダとして PointNet[24] を用いたモデルを構築しました。PointNet[24] とは、3 次元点群データを処理するために提案されたモデルであり、各点を個別に処理した後に、Max Pooling を用いることで点群の持つ順序に依存しない特徴量を抽出するモデルです。デコーダとして、線形層による固定長の点群生成や、LSTM・Transformer による逐次生成などを試しました。しかし、どの手法も生成された点群はまばらで、図 7.1a のように絵に見えるような点群を生成することはできませんでした。生成モデルとして GAN や Diffusion モデルも試しましたが、期待する結果を得ることはできませんでした。特に GAN では、Descriminater と比較して Generater の性能がとても低く、どれだけ Descriminater の性能を落としても Descriminater が勝ってしまい学習が進みませんでした。

Quick Draw！ [23] のデータセットは、世界中のユーザーから集めたデータであるため質のばらつきが大きく、人間でも判別がつかないようなデータも多く含まれていました。このため、まずデータの選別を行う必要があると考え、PointNet[24] を用いたクラス分類モデルを構築し、Perplexity を用いて低品質なデータを除外しました。データセットを半分に削減した結果、図 7.1b のように生成された点群に多少のまとまりが見られるようになりましたが、それでも絵としての復元には至りませんでした。



(a) 全データ　　(b) Perplexity により質を改善したデータ

図 7.1: 青が完成形・オレンジがモデルの出力

7.2.2　失敗した要因

クラスの分類タスクでは、PointNet[24] を用いたモデルが 80% 以上の精度を達成したことから、点群によるアプローチが上手くいかなかった原因はエンコーダではなくデコーダにあると考えられます。固定長を一度に出力する手法では点群の順序に依存しない特性、逐次生成ではデータ内のの描き順のばらつきによる一貫性の欠如が、上手く生成できなかった原因であると思います。

7.3　時系列データ

Quick, Draw![23] のデータは、点の座標の時系列データとして表現されているため、まず単純な Transformer を用いて次の点を予測する学習を試みましたが、意味のある出力を得ることはできませんでした。

Quick Draw![23] のデータセットを用いた線画の生成に関する論文として、sketchRNN[25] というモデルがあり、Google が開発した初めのモデルとなります。sketchRNN[25] は、エンコーダに双方向性 LSTM を採用し、入力データを潜在空間に埋め込んだ後、デコーダとして LSTM を用いて復元する、VAE に基づいた生成モデルです。データの前処理として、直接的な座標ではなく前の点からの差分の座標を使用し、「ペンを下ろしている」・「ストロークの終わり」・「絵の終わり」の 3 つの状態を表すワンホットベクトルを合わせた 5 次元の系列データで学習を行います。出力については、座標の差分の分布に混合正規分布を仮定し、サンプリングによって次点の予測を行います。混合正規分布を用いることで、複数の書き順が存在することによって学習が不安定化する問題点を解決しています。ただし、複数のクラスを 1 つのモデルで同時に学習させることは難しく、1 クラス毎に個別のモデルを学習させる必要があります。

また、LSTM ではなく Transformer を用いた sketchformer[26] というモデルもあり、こちらのモデルでは sketchRNN[25] とは異なり、単一モデルで複数クラス全ての学習を同時に行っても精度よく生成することが可能となります。しかし、論文の内容を再現してみても学習が上手く進まず、時間の制約から本プロジェクトでは使用することができませんでした。

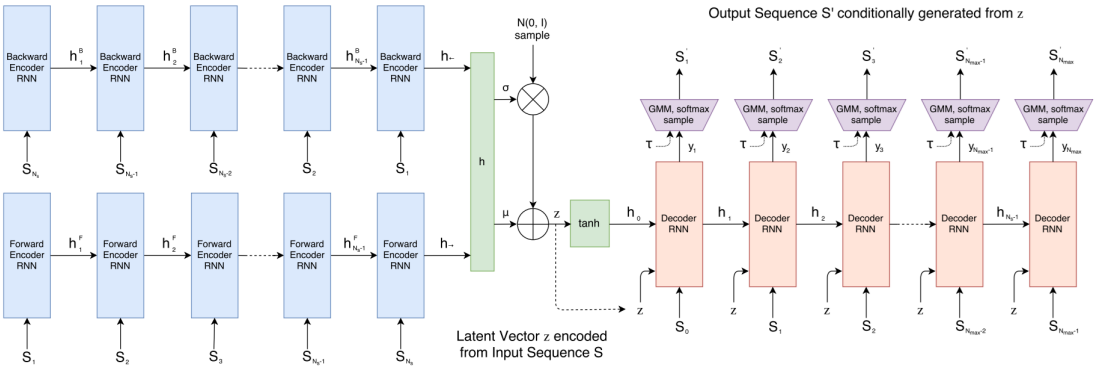


図 7.2: SketchRNN

7.4　終わりに

今回は、途中まで描かれた絵から完成形を描く AI の作成に挑戦しましたが、時間の制約と実力不足により、最終的には初期の線画生成モデルである SketchRNN[25] を採用する形となり、Transformer を用いた sketchformer[26] などの新しいモデルや、独自の工夫を加えた実験を十分に行うことができずに終わってしまいました。今後は、sketchformer[26] の実装や、Diffusion モデルなどの様々な生成手法を取り入れた実験を行い、より完成度の高いモデルを作成していきたいと思います。

参考文献

[1] Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[2] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024.

[3] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning, 2021.

[4] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 41, No. 1, pp. 162–175, 2017.

[5] Roberto Valle, Jose M Buenaposada, Antonio Valdes, and Luis Baumela. A deeply-initialized coarse-to-fine ensemble of regression trees for face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 585–601, 2018.

[6] Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, and Matthias Grundmann. Blazeface: Sub-millisecond neural face detection on mobile gpus. *arXiv preprint arXiv:1907.05047*, 2019.

[7] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation, 2024.

[8] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human motion diffusion model, 2022.

[9] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language, 2023.

[10] Harrison Jesse Smith, Qingyuan Zheng, Yifei Li, Somya Jain, and Jessica K. Hodgins. A method for animating children's drawings of the human figure, 2023.

[11] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d, 2024.

[12] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation, 2024.

[13] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields, 2019.

[14] Google. Hand landmarks detection guide. `https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker`.

[15] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network

for real-time 6-dof camera relocalization, 2016.

[16] Kulasis. https://www.k.kyoto-u.ac.jp/external/open_syllabus/top.

[17] LangChain. *LangChain*. https://langchain.com/.

[18] langchain_community.vectorstores.faiss.faiss. https://api.python.langchain.com/en/latest/vectorstores/langchain_community.vectorstores.faiss.FAISS.html.

[19] Leonard Richardson. *Beautiful Soup Documentation*, 2024. Accessed: 2024-11-16.

[20] Inc. Streamlit. *Streamlit: The fastest way to build and share data apps*, 2024. Accessed: 2024-11-16.

[21] Google. *Gemini*. Accessed: 2024-11-16.

[22] intfloat. intfloat/multilingual-e5-base, 2024. Accessed: 2024-11-16.

[23] Google. Quick draw! https://quickdraw.withgoogle.com/.

[24] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017.

[25] David Ha and Douglas Eck. A neural representation of sketch drawings, 2017.

[26] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. Sketchformer: Transformer-based representation for sketched structure, 2020.

[27] 新納浩幸. LLM のファインチューニングと RAG-チャットボット開発による実践. オーム社, 2024.

本会誌について

執筆者

第 1 章：カメラ入力を用いた強化学習によるライントレーサの実現　平塚謙良
第 2 章：目線で操るマウスカーソル　稲葉陽孔
第 3 章：KaiRA くんを動かそう　岡本和優
第 4 章：最強じゃんけん AI　千葉一世
第 5 章：京大シラバス検索 RAG システム - システム概要編　宮前明生
第 6 章：京大シラバス検索 RAG システム - 検索手法編　神原みちる
第 7 章：お絵描き予測 AI　千葉一世

京都大学人工知能研究会 KaiRA

代　　表　松田拓巳
活動日時　毎週木曜日 18:30〜
活動場所　文学部教室・株式会社 Deepcraft 京都オフィス（百万遍ビル 3F）など
入会資格　大学、学部、回生問わず、AI に興味がある方
会　　費　無料
活動内容　機械学習に関する本の輪読会、コード読み・実装会、kaggle への参加など

会誌 vol.8

発行日	2024 年 11 月吉日　初版発行
発行	京都大学人工知能研究会 KaiRA
公式サイト	https://kyoto-kaira.github.io/
メールアドレス	kyoto.kaira@gmail.com
X(旧 Twitter)	https://x.com/kyoto_kaira