

## AI 実験テキスト 3 (教師あり学習によるライントレース)

### 教師あり学習によるライントレースの概要

今回は図1のようなコースを使用し、黒いラインに沿ってライン上のみを走行するように機械学習を行う。JetBot をライン上に置いて画像データを取得し、取得した画像の中で進むべき場所を JetBot に教える。その後、JetBot を手で押して少し前進させ、その位置で進むべき場所を JetBot に教えるという動作を繰り返す。JetBot は収集したデータを基に機械学習を行う。今回も衝突回避と同様に、データにラベル（正解）を与えて学習する方法の教師あり学習を行う。ただし、今回は直進すべきか曲がるべきかではなく、 $xy$  座標でラベル（正解）を付ける。上手くライントレースをするためには、データが 80 枚程度必要である。



図1 教師あり学習によるライントレースのコース

## 実行手順

- (1) AI 実習テキスト 1 の(1)準備を完了し、図 2 の画面にする。図 3 のように既にディレクトリ移動されている状態の場合は、赤丸で囲んだファイルのマークを押して図 2 の画面にする。

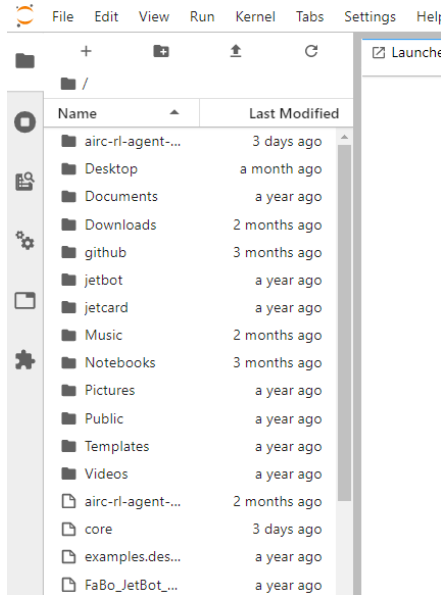


図 2 Jetbot に接続された JupyterLab の画面

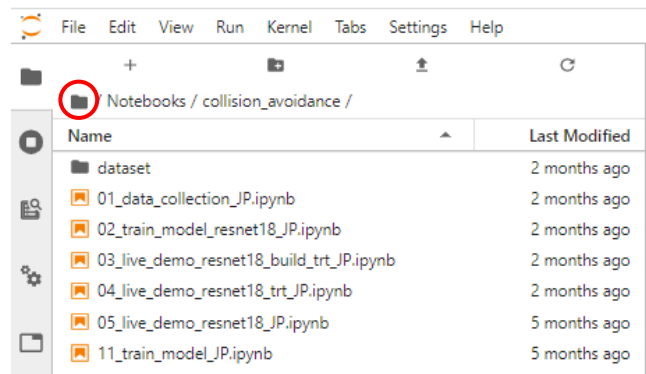


図 3 既にディレクトリ移動されている場合

- (2) 図 4 のように「Notebooks」を選択して開く。  
(3) 図 5 のように「road\_following」を選択して開く。

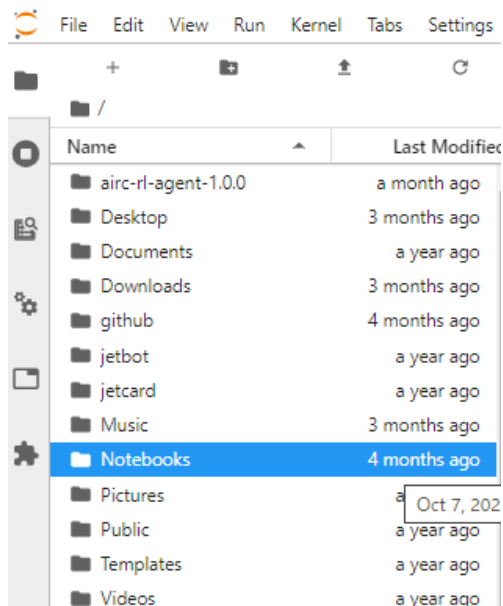


図 4 「Notebooks」選択画面

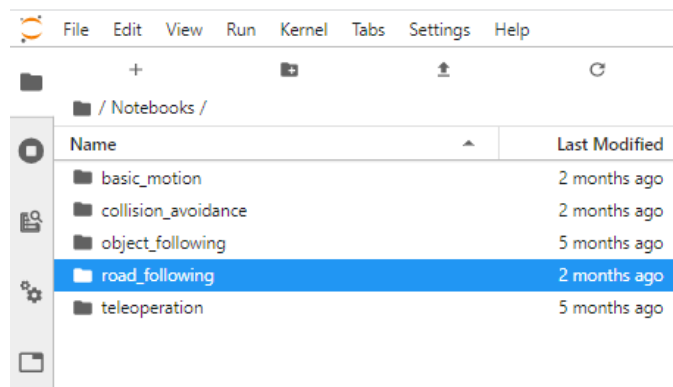


図 5 「road\_following」選択画面

- (4) 「road\_following」を開くと、図6のような画面になる。教師あり学習によるライントレースでは、このディレクトリの01から04までを順番に実行する。始めに「01\_data\_collection\_JP.ipynb」を選択して開く。

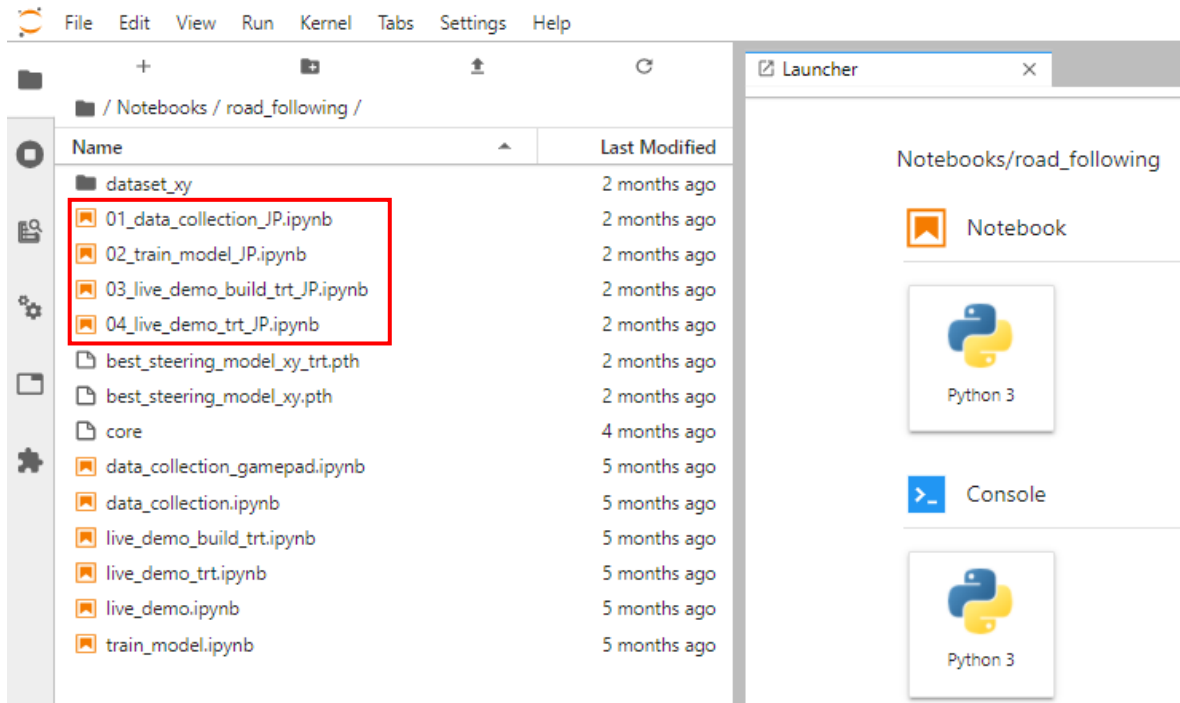


図6 「01\_data\_collection\_JP.ipynb」選択画面

- (5) 「01\_data\_collection\_JP.ipynb」を開くと図7のような画面になる。

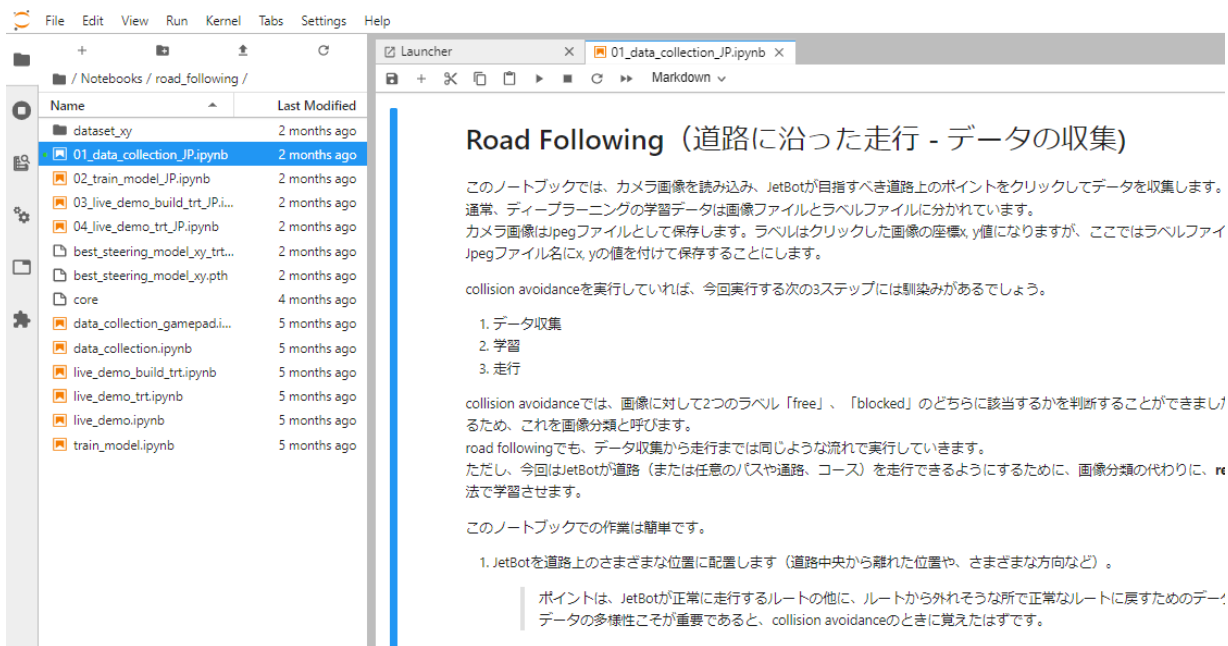


図7 01\_data\_collection\_JP.ipynb の画面

(6) これまでと同様に、説明を読み、コードを実行する。ただし、[1]のコードはデモ動画を再生するコードで、ライトレースに直接の関係はないため実行せずに実習を進める。

図8のように[2]のコードのセルを選択して、画面上部の▶マークを押し、セルのプログラムを実行する。実行中は図9のように[\*]となり、タブのJupyter labの表記の左隣のマークが砂時計になる。実行が終わると、実行が終わると自動的に次のセルが選択された状態になるため、次のセルに書かれている説明を読む。

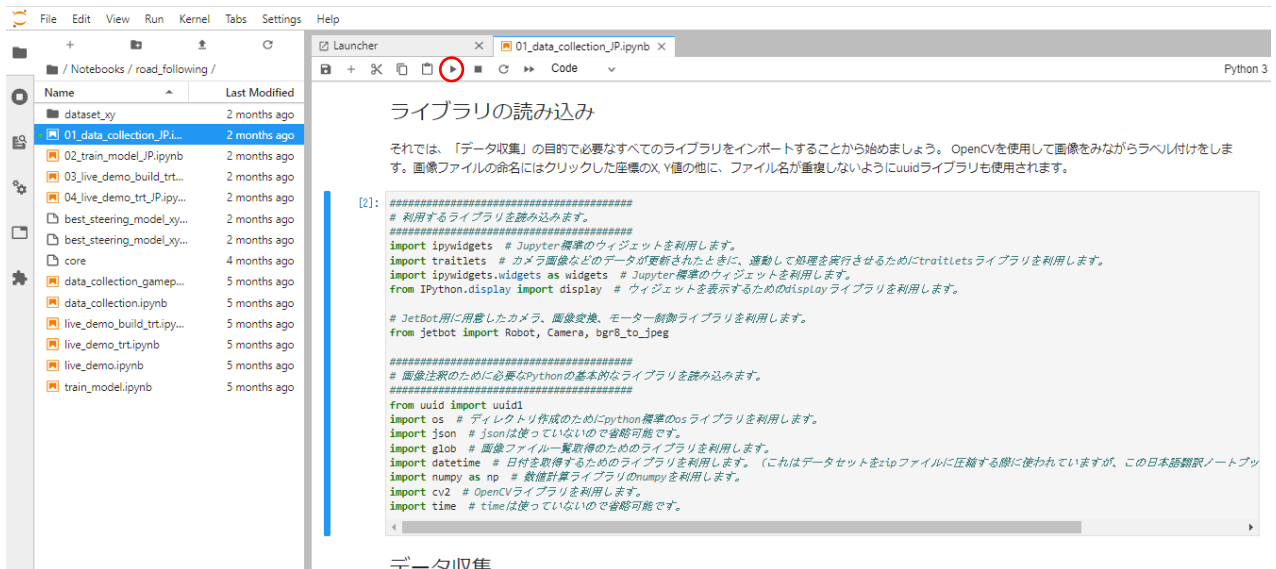


図8 [2]コードの実行画面

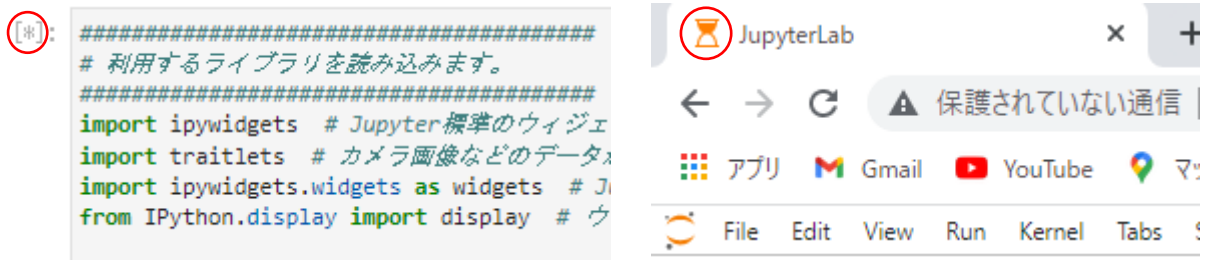


図9 [2]コードの実行中画面

(7) 説明を読み終えたら[3]のコードを実行する。このコードでは、データを保存するためのディレクトリの「dataset\_xy」ディレクトリと各ウィジェットの作成や、作成したウィジェットの表示などを行う。実行を終えると、図10のように、ウィジェットが表示されるため、実際にデータを収集していく。データ収集の手順は以下の通りである。

- ① JetBot をライン上に置く。
- ② 図10に示す左側の画面上で JetBot が進んでほしい場所をクリックする。
- ③ クリックすると図10に示す右側の画面が変わり、図10の画面が図11のようになり、右側の

画面上のクリックした位置に緑色のマークが表示される。

- ④ JetBot をラインに沿って少し手で移動させる。
- ⑤ ②から④を繰り返して 80 枚程度データを収集する。

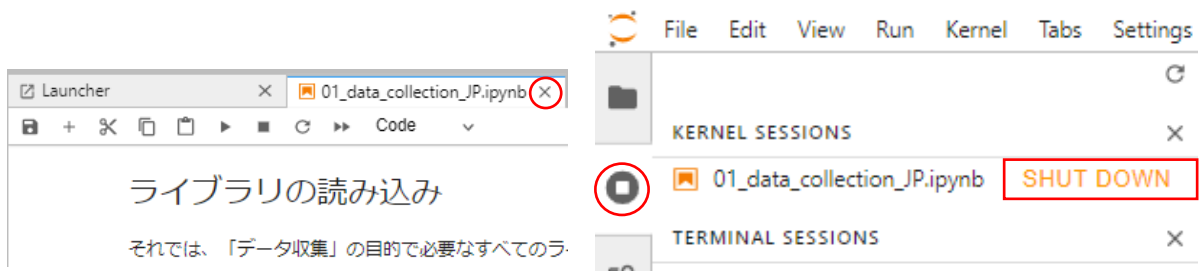


図 10 初期のデータ収集画面



図 11 データ収集

- (8) 十分なデータを収集したならば, [4]のコードを実行し, カメラの停止を行う。
- (9) 最後のセルの説明を読み終わったら, これまでと同様に, 図 12(a)のように Notebook を閉じる。このとき, Notebook の変更を保存するか聞かれるため, save のボタンを押して保存する。その後, 図 12(b)のように, 今開いている Notebook の Kernel をシャットダウンする。



(a) Notebook のシャットダウン

(b) Kernel のシャットダウン

図 12 Notebook の終了方法

(10) 「02\_train\_model\_JP.ipynb」を選択して開く。すると図 13 のような画面になる。

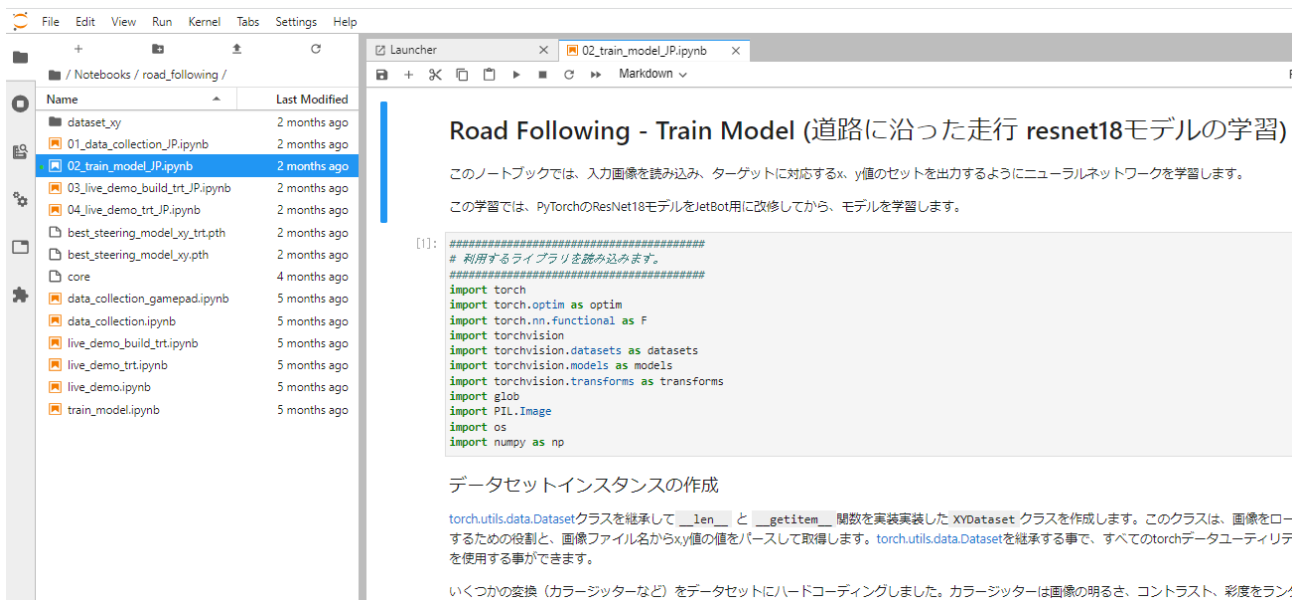


図 13 02\_train\_model\_JP.ipynb の画面

- (11) これまでと同様に、説明を読み、コードを実行する動作を繰り返して実習を進め、最後のセルの説明を読む。この Notebook では、衝突回避と同様に、転移学習を行う。ここでは、すでに画像分類ができるモデルを使って転移学習を行い、ライントレースをするためのモデルを作成する。
- (12) (9)と同様に Notebook を終了させる。Kernel をシャットダウンするのを忘れないよう注意する。
- (13) 「03\_live\_demo\_build\_trt\_JP.ipynb」を開く。すると図 14 のような画面になる。

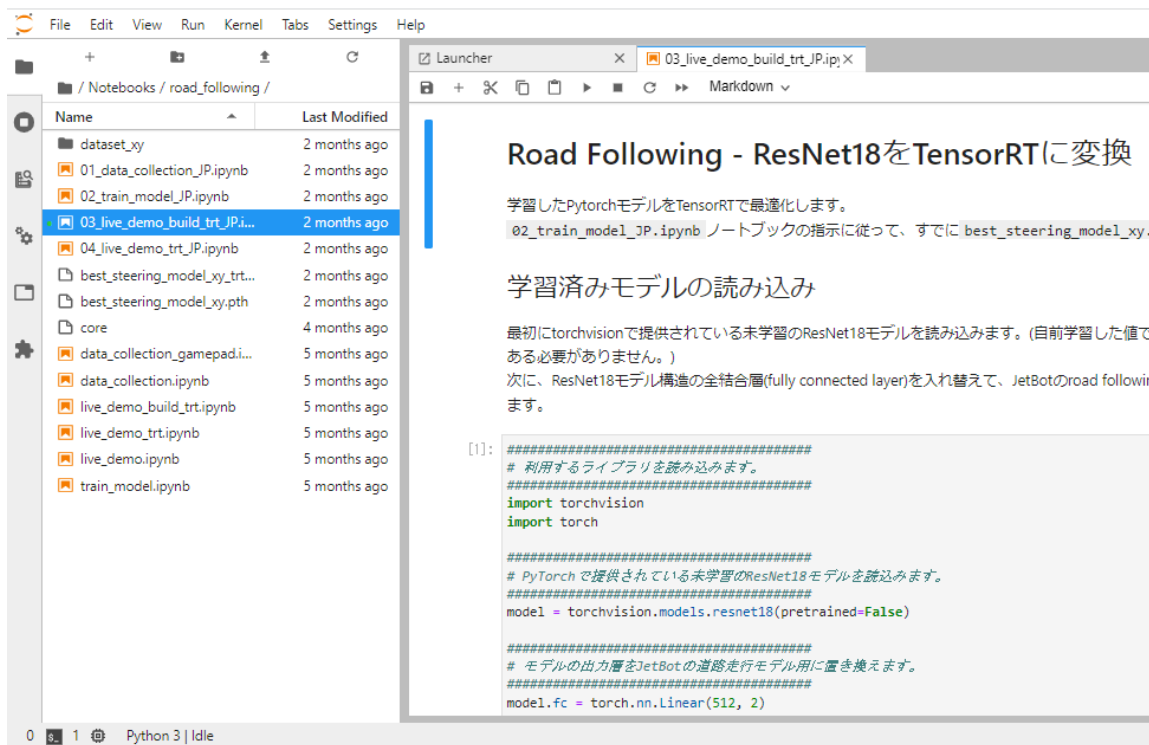


図 14 03\_live\_demo\_resnet18\_build\_trt\_JP.ipynb の画面

- (14) これまでと同様に、説明を読み、コードを実行する動作を繰り返して実習を進め、最後のセルの説明を読む。この Notebook では、モデルの読み込みと高速化および最適化のためのモデル変換を行う。すべて実行し終わったら(9)と同様に Notebook を終了させる。
- (15) 「04\_live\_demo\_trt\_JP.ipynb」を開く。すると図 15 のような画面になる。

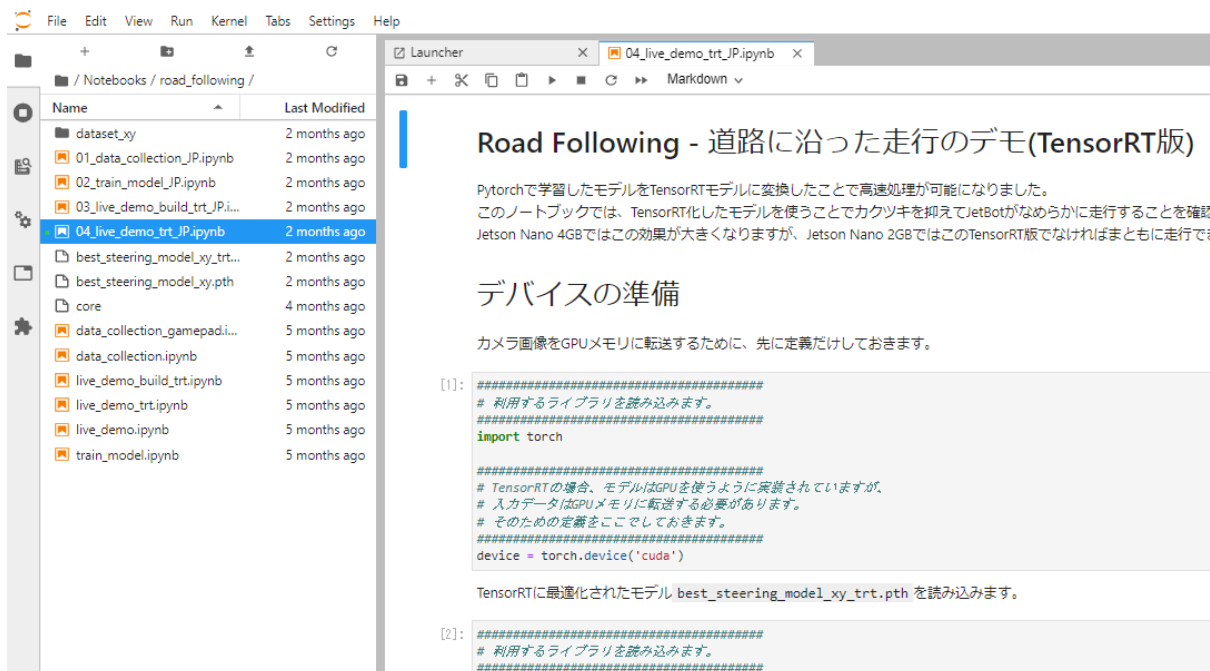


図 15 04\_live\_demo\_trt\_JP.ipynb の画面

(16) これまでと同様に、説明を読み、コードを実行する動作を繰り返して実習を進める。この Notebook では、実際に JetBot を走行させ、ライントレースを行う。自分で操作が必要になるスライダーウィジェットなどもあるため、各コードの説明と実行手順を併せて、少し細かく説明していく。

- ① 説明を読みながら、[1][2][3]のコードを実行する。ここでは、デバイスの準備とカメラ画像の前処理の関数を定義する。
- ② 説明を読み、[4][5]のコードを実行する。[5]のコードを実行した後、カメラのウィジェットが表示されるので、カメラの動作確認を行う。
- ③ 説明を読み、[6][7]のコードを実行する。[7]のコードを実行した後、図 16 のようなスライダーウィジェットが表示される。このスライダーは、JetBot の走り方を調整するものである(各パラメータの簡単な説明は p.9 で述べる)。JetBot が走行を始めた後、このスライダーを使ってよりスムーズに走行できるように調節する。
- ④ 説明を読み、[8]のコードを実行する。[8]のコードを実行した後、図 17 のようなスライダーウィジェットが表示される。このスライダーは、JetBot がどのように判断しているかを表示するもので、JetBot が走行を始めた後、x スライダーと y スライダーは、JetBot が考えた進むべき位置を表す。

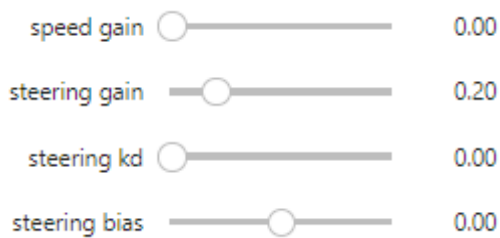


図 16 JetBot の制御スライダー

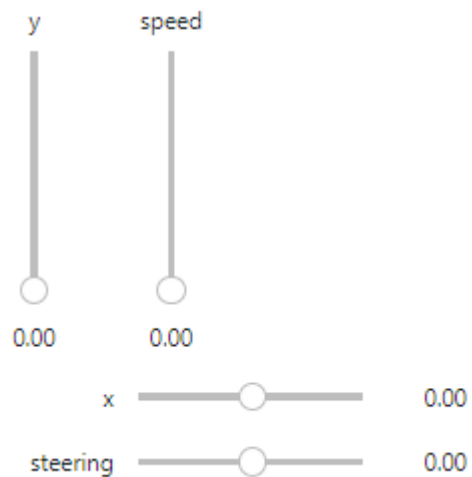


図 17 JetBot の判断を示すスライダー

- ⑤ 説明を読み、[9][10]のコードを実行する。[10]のコードを実行した後、図 18 のようなウィジェットが表示される。start jetbot ボタンを押すと走行開始、stop jetbot ボタンを押すと走行停止する(注意: ここではまだ走行を始めない)。steering と書かれているスライダーは、JetBot が推定したステアリング値を表している。



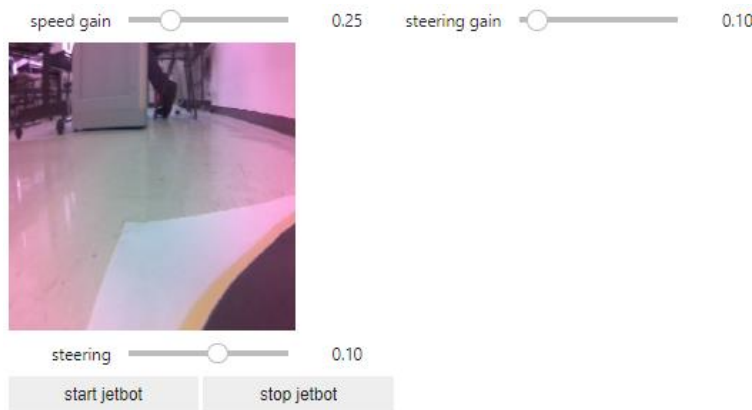


図 18 教師あり学習によるライトレースの走行画面

- ⑥ 実習をしやすくするために、今実行中の Notebook を複製し、2 画面で表示する。図 19 のように Notebook のタブを右クリックし、その後、「New View for Notebook」を選択する。

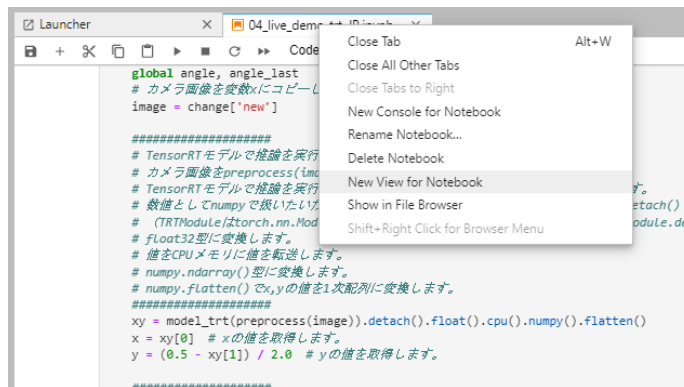


図 19 「New View for Notebook」選択画面

- ⑦ すると、04\_live\_demo\_trt\_JP.ipynb が複製され、2 画面表示になる。図 20 のように、片方の画面に図 18 に示した走行画面を、もう片方の画面に図 16 に示した JetBot の制御スライダーを表示する。
- ⑧ 図 18 の画面の start jetbot ボタンを押す。しばらく待つと走行を始める。speed gain の値が 0 のときは、走行を始めるまで値を少しずつ増やしていく。
- ⑨ この段階では、制御パラメータがデフォルトのままであり、自分の動作環境にあっていないのでラインに沿って走行しないことが多い。そのため、図 18 のスライダーウィジェットで調節する。以下にそれぞれのパラメータの簡単な説明を記す。詳しくは[7]のコードの説明を参照する。

speed gain : モータの出力調節をするパラメータ  
 steering gain : タイヤの角度調節をするパラメータ  
 steering kd : タイヤが元の角度に戻るまでの時間の調整をするパラメータ  
 steering bias : JetBot の走行位置を調整するパラメータ

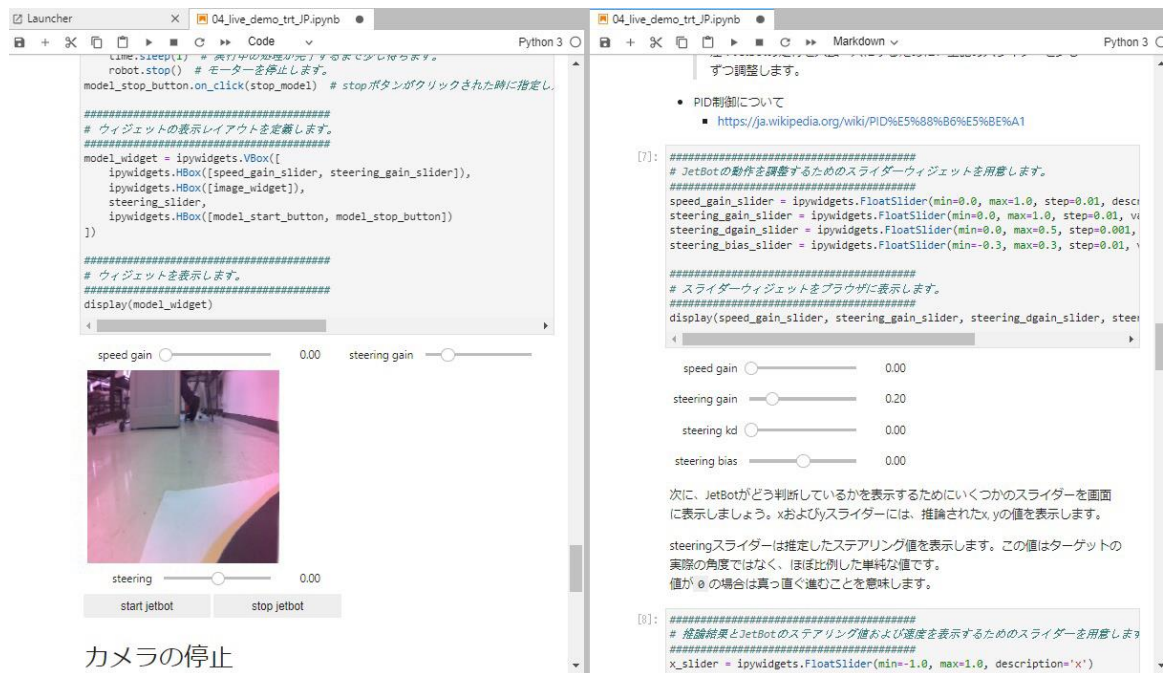


図 20 Notebook の 2 画面表示

- ⑩ スライダーウィジェットの各値を少しずつ変化させ、滑らかにラインに沿って走行するように調節する。調節しても上手くライン上を走行しない場合は、次の(17)に進む。
- (17) [11]のコードを実行してカメラを停止させ、結論を読む。読み終えたら(9)と同様に Notebook を終了させる。
- (18) ライントレースを上手くできなかったならば、画像数を増やして、もう一度学習と走行をするために、「01\_data\_collection\_JP.ipynb」から再度実行手順を実施する。上手くラインレースができたなら、指導教員に上手くできた旨を伝える。その後、画像数やエポック数を変えるとどのような変化が見られるかを考え、画像数やエポック数を変えて再度実行手順を実施し、実験してみる。なお、画像数とエポック数を増加するほど学習にかかる時間が長くなり、JetBot のメモリが不足しやすくなるため、注意が必要である。
- (19) ライントレースを終えたら、衝突回避と同様に、次に実習を受ける学生のために、データを初期の状態に戻す。図 21 のように、Launcher タブの Other から「Terminal」をダブルクリックし、Jupyter lab 上でターミナルを開く。

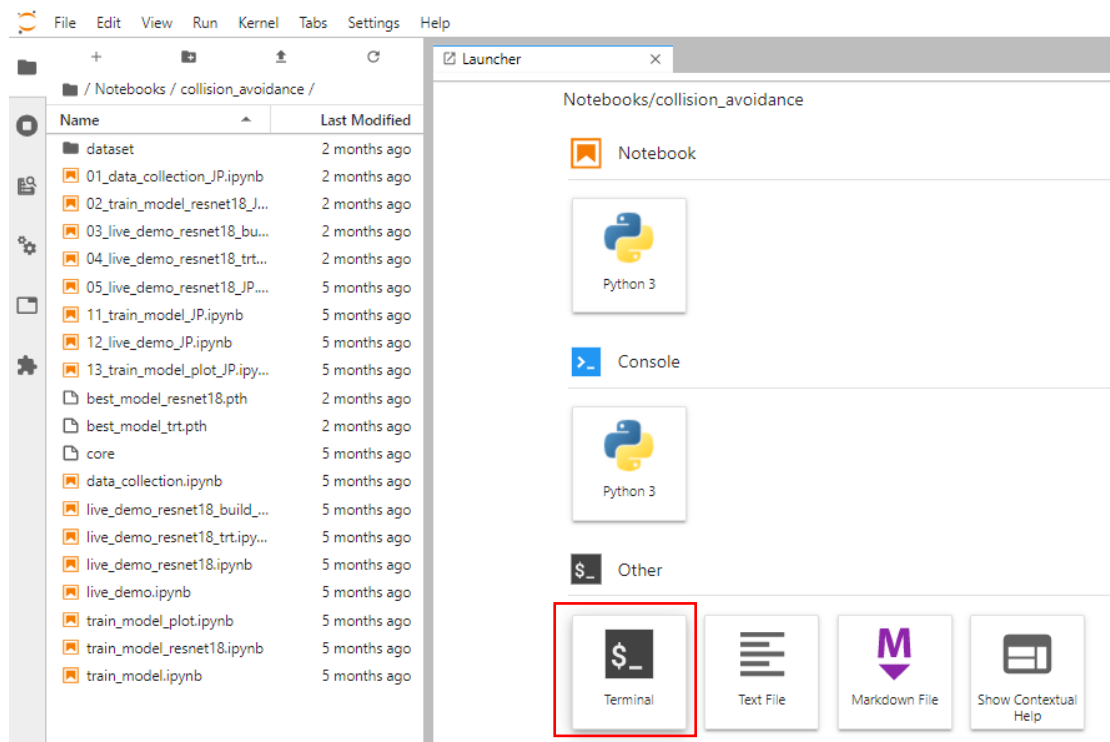


図 21 jupyter lab 上のターミナルの開き方

(20)jupyter lab 上のターミナルが開いたら、図 22 のように、以下の①のコマンドを入力し、enter キーを押して衝突回避のディレクトリに移動し、その後以下の②のコマンドを入力して、enter キーを押し、「dataset\_xy」ディレクトリを消去する。

①cd \_Notebooks/road\_following ( \_ は空白)

②rm \_-r \_dataset\_xy ( \_ は空白)

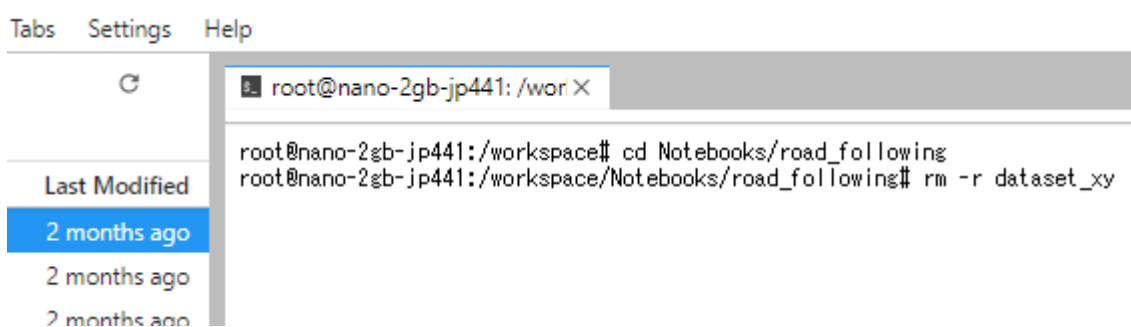


図 22 「dataset\_xy」ディレクトリの削除方法

(21) 「dataset\_xy」ディレクトリの削除が終わったら、(9)の Notebook の終了方法と同様に、jupyter lab のターミナルを閉じ、図 23 のように Kernel をシャットダウンする。

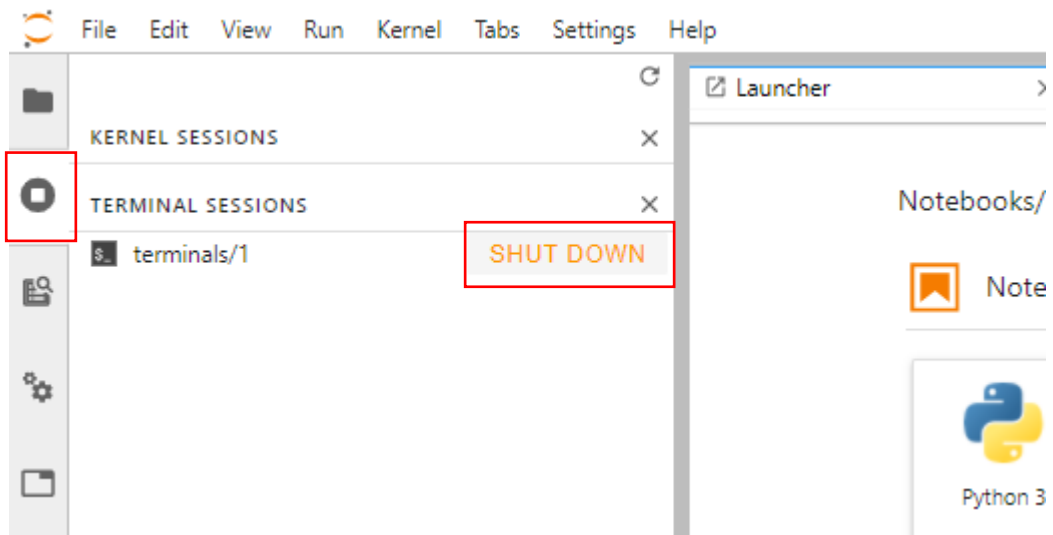


図 23 jupyter lab のターミナル Kernel のシャットダウン

(22)jupyter lab のターミナルのシャットダウンが終わったら、AI 実習テキスト 1 の第 3 章に従って JetBot をシャットダウンする。