

# Yoloアルゴリズムを使用した自動運転での車の検出

これはオンラインコースCoursera (deep learning.ai-ConvolutionalNeuralNetworks-Week3)のプロジェクトの1つです。

## 1. 序説

- このプロジェクトはRedmon et al., 2016 と Redmon and Farhadi の2つの論文で説明したYOLOアルゴリズムを実装します。
- このアルゴリズムは、高い精度を実現し、リアルタイムで実行できるため、非常に人気があります。
- 目的は、車検出データセットでオブジェクト検出を使用し、境界ボックスを処理することです。

## 2. 問題文

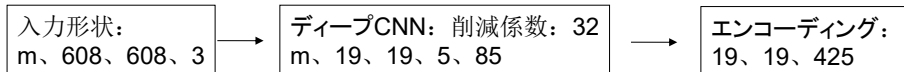
- このプロジェクトは自動運転車に関連しているため、データセットはボンネット上のカメラからのものです。
- データをトレーニングする前に、データを同じフォルダーに収集し、データセット内のすべての車の周りに境界ボックスを描画してラベルを付ける必要があります。

## 3. 詳細モデル

- 簡単に言えば、YOLOアルゴリズムは、インプット、ディープコンボリューションニューラルネットワーク、エンコード/アウトプットで構成しています。
- このプロジェクトでは、①インプットの形状→(m、608、608、3)、②darknet (convolution neural network layers)、yolo (アンカー処理および評価)、③アウトプット (m、19、19、5、85)。
- m→データ例の数、608x608→画像サイズ、3→チャンネル数、85→80クラス+ (pc、bx、by、bh、bw)、5→アンカーボックスの数。

## 4. 詳細モデル

- 概要



各ボックスのクラスに基づいてスコアが計算します。  
スコア =  $pc * (c1, c2, \dots, c80)^T = (0.12, 0.13, 0.44\dots, 0.09)$   
score: 0.44  
box: bx, by, bh, bw  
class: 3 → car

## 5. クラススコアのしきい値を使用したフィルタリング

- クラスのスコアを選択するには、しきい値を使用してスコアをフィルタリングする必要があります。
- しきい値処理中に、エンコードを次の変数に再配置すると便利です。
  - ① boxes\_confidence: pcを含む形状のテンソル（19x19、5、1）。
  - ② boxes: 5ボックスごとに（bx、by、bh、bw）を含む形状のテンソル（19x19、5、4）。
  - ③ box\_class\_probability: 検出（c1、c2、...、c80）を含む形状のテンソル（19x19、5、80）。

## 6. Non-Max Suppression

- 重複を回避するには、Non-Max Suppression (NMS) というフィルターが必要です。
- NMSで使用する関数は、Intersection over Union function (IoU)です。

## 7. 画像の事前学習済みモデルのテスト

- クラス、アンカー、および画像形状の定義。
- 事前学習済みモデルの読み込み。Yolo1はトレーニングに非常に長い時間がかかる場合があります。そのため、転送学習を使用して、「yolo.h5」に保存されている事前学習済みのKeras YOLOモデルを読み込みます。
- モデルの出力outputを使用可能な境界ボックステンソルに変換します。
- フィルタリングボックス
- 画像でグラフを実行する