

リアルタイム 30fps! AI計測制御に挑戦

鎌田 智也

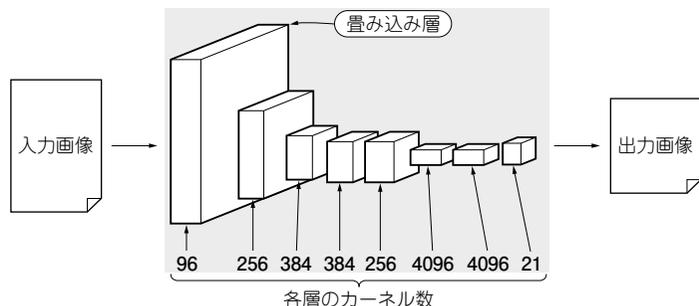


図1 ロボット・アームの部位推定に使用したセグメンテーション用ニューラル・ネットワークFCN-Alexnet

ここまでの実験では、ロボット・アームの位置推定のためのセマンティック・セグメンテーションに、DIGITSに付属するサンプル・モデルFCN-Alexnetをそのまま使いました。

しかし、Jetson Nanoに実際に実装して動作させてみたところ、処理速度が3~4fps(フレーム/秒)程度しかありませんでした(mode=1)。モータ駆動するアームの動きをリアルタイムに認識させるためには、モデルの推論処理のスピードアップを実現していかななくてはなりません。

本章では、モデルのネットワーク構造をカスタマイズして、Jetson Nano上でのパフォーマンス改善と測定精度の向上にチャレンジしていきます。

今回のニューラル・ネットワーク FCN-Alexnetの特徴

● ネットワーク構造記述ファイルdeploy.prototxt

カスタマイズによるリアルタイム化に挑戦する前に、ここまで実験で使ってきたFCN-Alexnetのネットワーク構造についておさらいします。

ネットワークの構造は、deploy.prototxtというファイルに記述されています。PC上のDIGITSはこのファイルを内部でCaffeに渡して、モデルの学習と推論を実行しています。エッジJetson Nano側に

組み込んだ際にも、jetson-inferenceライブラリがこのファイルを解釈して、ネットワーク構造を組み立てています。

● ネットワーク構造&処理

図1は、FCN-Alexnetのネットワーク構造です。合計7レイヤ(層)の畳み込み層で構成されています。このようなネットワーク・モデルは、全層畳み込みネットワーク(FCN = full convolutional network)と呼ばれ、これがモデル名についているFCNの由来です。

● 各層の画像サイズ

図2(a)は、640×480ピクセルの画像を入力した場合の各レイヤの画像サイズです。最終的に21chの横21ピクセル、縦16ピクセルのセグメント画像を出力します。各層における画像サイズは、各層のカーネル・サイズやstrideなどのパラメータによって決まります。

図2(a)で示した各層の画像サイズを、層ごとのパラメータとともにまとめたのが図2(b)です。例えばpool1は、MAXプーリングという処理を行うレイヤで、隣接する横3×縦3ピクセルの中からMAX値を出力していきます。3×3のウィンドウを1ピクセルずつ移動させる指定がstride=1ですが、このpool1ではstride=2なので、ウィンドウを2ピクセルずつ処理