

SoC FPGAで画像処理をより低消費により高速に

研究概要

高度な機能を安価なシステムで実現するための
アルゴリズム及びその実装に関する検討
最先端の設計技術 (SoC-FPGA)を駆使したシステム実装

IoTの活用末端端末へのAIなどの高度な機能の要求

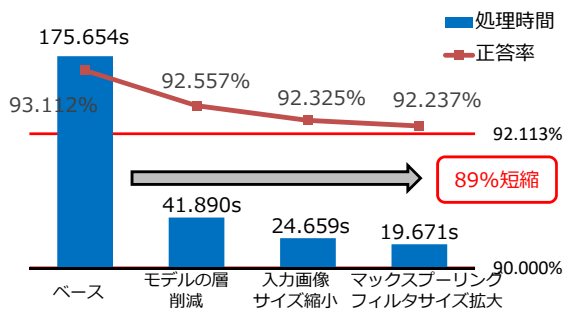
人工知能 や 画像処理は処理量が膨大

ハードウェアとソフトウェアを組み合わせた
 システム全体の最適化が必要不可欠

畳み込みニューラルネットワーク(CNN)の学習高速化

4カテゴリ分類(赤りんご、青りんご、さくらんぼ、トマト)で、**精度を保ったまま演算量を減らす**手法を提案

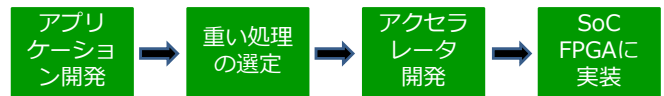
- モデルの層の削減 17層→9層
- 入力画像サイズの縮小 32x32→24x24
- マックスプーリング層のフィルタサイズ拡大2x2→4x6



FPGAでもCPUでもない! SoC FPGAとは

CPU (ARM) と FPGA を1チップにしたデバイス
FPGAでも、プロセッサでもない、両者の『いいとこ取り』
 をして生まれたプラットフォーム

開発フロー



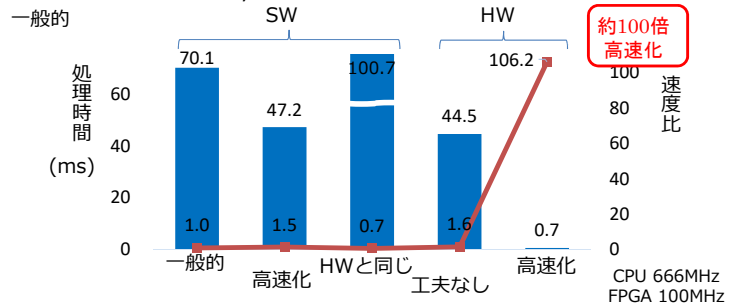
素早い商品化、低消費電力、低コスト、低設計リスク、および基盤の小型化を実現!

ソフトウェア・ハードウェアの協調設計 SoC FPGAを用いたフィルタ処理の低消費・高速化

画像のフィルタ処理のみをSoC FPGAで実現する手法の検討

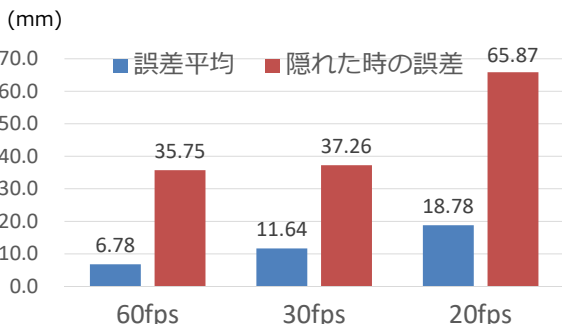
特+徴: VivadoHLSを用いて**手間をかけずにハード化**
 FPGA: ZynqBerry (Zynq搭載) カメラ→処理→HDMI出力
 比較: ソフト最適化のみ、ハードで加速、実行時間比較
 工夫点:

- ソフト (SW): 分岐の削減 ループの削減
- ハード (HW): **バッファの挿入 演算の整数化**



複数物体追跡の演算量の低減への取り組み

各個体を**時分割(低フレームレート)**して追跡
 色情報を利用した**カルマンフィルタ**を使用
 ・予測値と実測値の誤差比較
 ・**追跡物体が隠れた(22/60秒)場合**の誤差の比較
 ・演算量減少率: 2個追跡:39.6% 3個追跡:45.6%



研究課題

- DeepLearningをSoCで実現
 BNN-PYNQ*プロジェクトを利用した分類、認証、学習
 (*) Binarized Neural Network (BNN)
- 超解像(拡大縮小)をリアルタイムで処理
 機械学習を利用した超解像
 Lanczos (ランチョス法) による拡大縮小
 補完処理のFPGA化
- 物体追跡 **狙った獲物は見逃さない**
 追跡物体が隠れた場合の追加処理を検討
 誤差の削減
 行列演算部をHW化

