

## 4CH 出力 CNN を用いた超解像処理のハードウェア化 A 4CH CNN Hardware Architecture for Image Super-Resolution

鈴木 洸陽<sup>†</sup> 澤井 剛史<sup>†</sup> 廣瀬 哲也<sup>†</sup> 黒木 修隆<sup>†</sup> 沼 昌宏<sup>†</sup>  
Koyo Suzuki Takeshi Sawai Tetsuya Hirose Nobutaka Kuroki Masahiro Numa

### 1. はじめに

本稿では、画像の高品質な 2 倍拡大を実現する超解像を少ないリソースで高速動作可能なハードウェアで実現することを目的として、4CH 出力 CNN [1] を用いた超解像処理ハードウェア・アーキテクチャを提案する。

図 1 に 4CH 出力 CNN の構造を示す。4CH 出力 CNN は、3 層の畳み込み層で構成され、 $90^\circ$  ずつ回転させた入力画像に対応する 4 つのチャンネル (4CH) を出力する。超解像処理後の出力画素は、これら 4CH から生成される。

また、4CH 出力 CNN のように複雑な処理を FPGA 上のハードウェアで実現しようとする、その回路規模拡大が避けられないため、処理の高速動作を保ちつつリソース削減を図る必要がある。そこで、それらを満たす二つのアーキテクチャを提案・評価する。

### 2. 4CH 出力 CNN 回路に関する 2 種類の提案構成

超解像処理の高速化とリソース削減を目的として、4CH 出力 CNN 回路に関して、2 種類のハードウェア・アーキテクチャを提案する。

#### 2.1 提案構成 1

図 2 に、提案構成 1 の全体アーキテクチャを示す。提案構成 1 では、高速化のために第 2 層目では 8 台のフィルタサイズ  $5 \times 5$  の畳み込み回路による並列処理を、第 3 層目では 16 台のフィルタサイズ  $3 \times 3$  の畳み込み回路による並列処理を行う。処理の順序としては、第 1 層目で入力画像から畳み込み処理によって 1 枚の第 1 層目特徴マップ  $n_{11}$  を得る。次に、第 2 層目でそれを入力とした畳み込み処理を行うことによって、 $n_{11}$  に対応した第 2 層目特徴マップが得られ、それを FPGA 内蔵 SRAM である BRAM (Block RAM) に保持する。その後、第 1 層目で第 1 層目特徴マップ  $n_{12}$  を

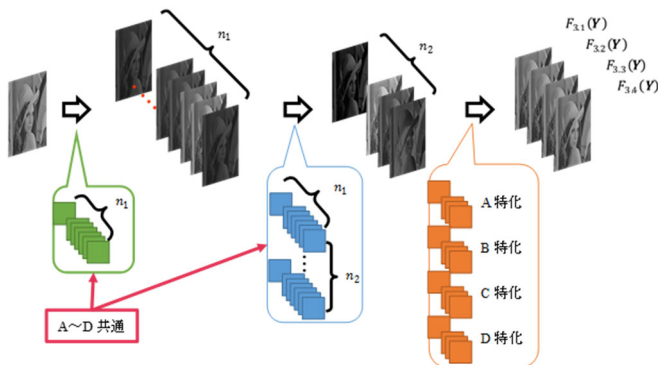


図 1 4CH 出力 CNN の構造

得る。また第 2 層目での畳み込み処理によって  $n_{12}$  に対応した第 2 層目特徴マップが得られ、それを以前に BRAM で保持した第 2 層目特徴マップと足し合わせて BRAM に保持する。同様の処理を繰り返すことによって、第 2 層目特徴マップが完成する。この第 2 層目特徴マップに基づき、第 3 層目における畳み込み処理によって、A 特化、B 特化、C 特化、D 特化の順に、第 3 層目特徴マップを得る。

#### 2.2 提案構成 2

図 3 に、回路の一部を再利用したリソース削減の実現を特徴とする提案構成 2 の全体アーキテクチャを示す。各層の特徴マップを得る処理内容は、提案構成 1 の場合と同様である。提案構成 1 では、第 3 層目の畳み込み処理が行

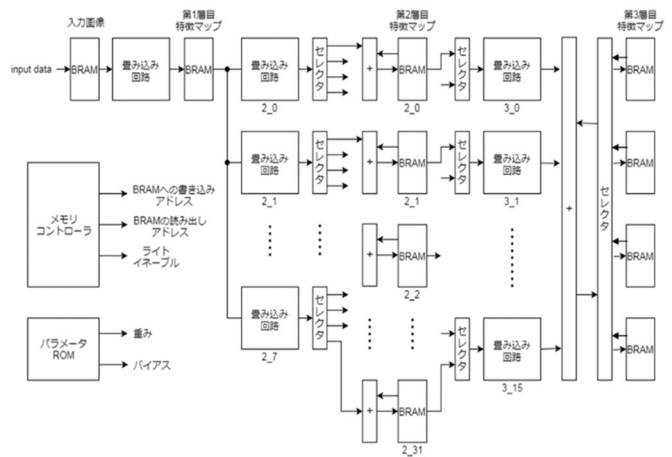


図 2 提案構成 1 の全体アーキテクチャ

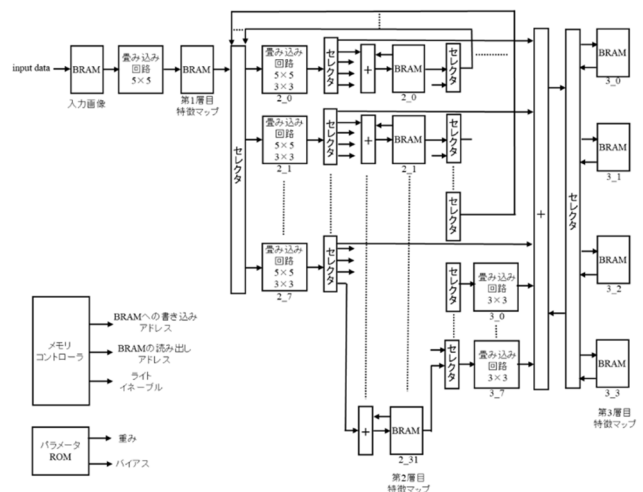


図 3 提案構成 2 の全体アーキテクチャ

われている間は、第 2 層目の畳み込み回路が動作しない点が課題となる。そこで提案構成 2 では、第 3 層目の処理の際に第 2 層目で用いたフィルタサイズ可変の畳み込み回路に特徴マップを入力して、畳み込み回路を再利用することによって、必要となるリソースを削減する。

次に、フィルタサイズ可変の畳み込み回路について述べる。図 4 に、フィルタサイズ  $3 \times 3$  と  $5 \times 5$  とで可変な畳み込み回路におけるシリアル・パラレル変換部を示す。din よりストリームデータを入力することにより、畳み込み処理のうち積和演算部に必要なデータが出力される。その出力を図 5 に示す積和演算部に入力することにより、ストリームデータとして演算結果を dout から出力する。シリアル・パラレル変換部と積和演算部の回路中のセレクタは、メモリ・コントローラからの出力信号によって制御される。

### 3. 提案構成の評価

提案構成 1, 2 に基づく回路について、それぞれ FPGA へマッピングして評価を行った。論理合成に Vivado 2016.4, FPGA に Vertex7 (XC7VX485T-2FFG1767C) を用いた。

表 1 に、提案構成 1 の回路全体に関するマッピング結果を示す。この表から、提案構成 1 は FPGA のリソース数の範囲内で実現可能であることを確認した。

表 2 に、提案構成 2 の回路全体に関するマッピング結果を示す。FPGA のリソース数の範囲内で実現可能であることと、提案構成 2 は提案構成 1 と比べて、LUT や DSP のリソースを約 10% 削減可能であることを確認した。

さらに、4CH 出力 CNN を用いた超解像の処理時間について、ソフトウェア実行と、提案構成 1, 提案構成 2 それぞれを適用した場合のシミュレーション結果 (クロック周波数 200 MHz) との比較を行った。ソフトウェアについて、CPU: Intel Core i7-7700K 4.20 GHz クロック、メモリ: 16 GB の環境で実行した結果、処理時間は約 140 ms となった。一方、提案構成 1 を用いた場合の処理時間は 25.61 ms, 提案構成 2 を用いた場合は 25.62 ms と評価された。よって、提案構成 1 と提案構成 2 とでは処理時間はほぼ同等であり、

表 1 提案構成 1 全体のマッピング結果

| リソース    | 搭載数                | 利用数                | 利用率 (%) |
|---------|--------------------|--------------------|---------|
| LUT     | $3.04 \times 10^5$ | $2.22 \times 10^5$ | 73.02   |
| LUT RAM | $1.31 \times 10^5$ | $1.82 \times 10^4$ | 13.89   |
| FF      | $6.07 \times 10^5$ | $3.77 \times 10^5$ | 62.11   |
| DSP     | $2.80 \times 10^3$ | $1.16 \times 10^3$ | 41.43   |
| BRAM    | $1.03 \times 10^3$ | 580                | 56.31   |

表 2 提案構成 2 全体のマッピング結果

| リソース    | 搭載数                | 利用数                | 利用率 (%) |
|---------|--------------------|--------------------|---------|
| LUT     | $3.04 \times 10^5$ | $1.93 \times 10^5$ | 63.49   |
| LUT RAM | $1.31 \times 10^5$ | $1.44 \times 10^4$ | 10.99   |
| FF      | $6.07 \times 10^5$ | $3.28 \times 10^5$ | 54.04   |
| DSP     | $2.80 \times 10^3$ | $8.72 \times 10^2$ | 31.14   |
| BRAM    | $1.03 \times 10^3$ | 580                | 56.31   |

ハードウェア化によって、ソフトウェア実行と比べて約 5.5 倍の高速化が達成されることを確認した。

### 4. まとめ

本稿では、4CH 出力 CNN を用いた超解像処理について、処理の高速化とリソースの削減を目的とする 2 種類のハードウェア・アーキテクチャを提案した。提案構成 2 では提案構成 1 と比べて LUT や DSP リソースを約 10% 削減するとともに、いずれの提案構成もソフトウェアと比較して約 5.5 倍の高速動作が可能であることを確認した。

#### 参考文献

- [1] 大谷真也, 加藤 裕, 黒木修隆, 廣瀬哲也, 沼 昌宏, “4 並列の畳み込みニューラルネットワークを用いた超解像”, 電子情報通信学会論文誌, vol. J99-D, no. 5, pp. 588-593, May 2016.

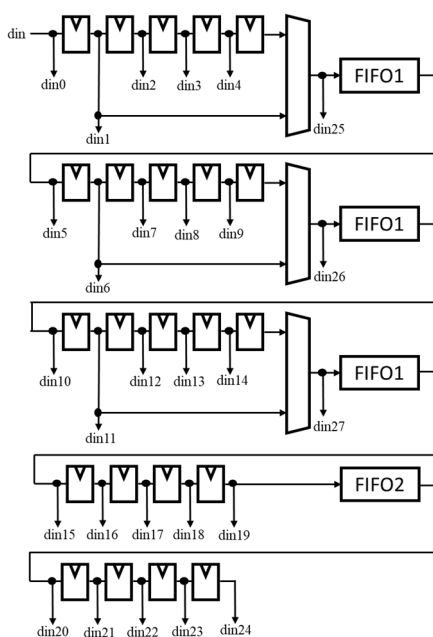


図 4 シリアル・パラレル変換部

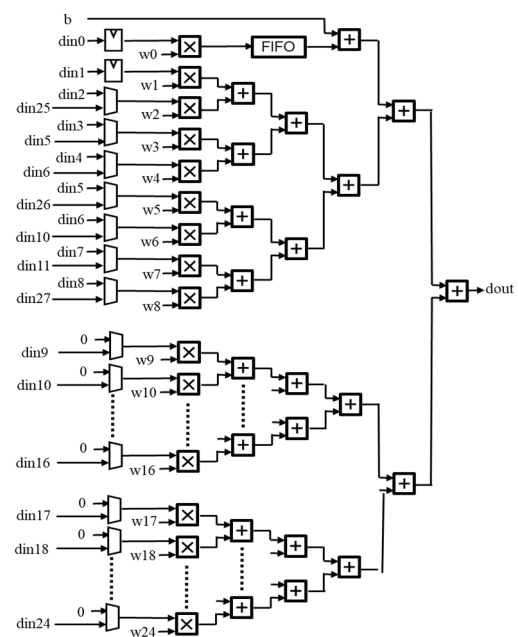


図 5 積和演算部