様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　３ 年　 ６ 月 ２１ 日現在

機関番号：１３３０１
研究種目：基盤研究(C)（一般）
研究期間：2018～2020
課題番号：１８Ｋ１１３５０

研究課題名（和文）SLAM用意味分割・距離推定・動き推定の実時間同時処理1チップCNN回路開発

研究課題名（英文）Development of 1-chip CNN real-time processor for semantic segmentation, distance estimation, and motion estimation

研究代表者

深山　正幸（Miyama, Masayuki）

金沢大学・電子情報通信学系・准教授

研究者番号：３０３２４１０６

交付決定額（研究期間全体）：（直接経費）　　3,400,000円

研究成果の概要（和文）：画像から周囲環境を認識するために物体輪郭推定と距離（視差）推定を同時に行う畳み込みニューラルネット(CNN)を考案し、FPGA(Field Programmable Gate Array; 手元で書き換え可能な集積回路)に実装した。輪郭検出を、従来の輪郭・非輪郭の二値分類ではなく、物体境界からの距離を推定する回帰問題へ帰着させ、視差推定とのマルチタスク学習を可能とした。そして二つの推定を同時に行う全重み共有のCNNを考案した。これを3ビットまで量子化したが、精度低下は僅かであった。FPGA実装の結果、250 MHzで動作し、480×320画素の画像に対するスループットは134 fpsであった。

研究成果の学術的意義や社会的意義
ステレオ画像を用いた一般物体輪郭検出と視差推定の同時学習とマルチタスクCNNはこれまでに報告されていない。これを3ビットまで量子化したCNNの回路設計やFPGA実装も行われていない。このCNN回路は畳み込みの重みを変更すれば画像を画素単位でカテゴリ分類する意味的分割も実行できる。開発したFPGAは解像度480×320画素の画像に対してスループット134 fpsで動作し、自律ロボットのSLAM（自己位置推定と地図作成の同時実行）や周囲環境認識に応用できる。

研究成果の概要（英文）：We devised a convolutional neural network (CNN) that simultaneously estimates the contour of an object and the distance (parallax) in order to recognize the surrounding environment from an image, and implemented it on an FPGA (Field Programmable Gate Array). Contour detection is transformed to a regression problem that estimates the distance from the object boundary instead of the conventional binary classification of contour and non-contour, enabling multitask learning with parallax estimation. Then, we devised a CNN with full weight sharing that performs two estimations at the same time. This was quantized to 3 bits, but the decrease in accuracy was slight. As a result of FPGA implementation, it operates at 250 MHz and has a throughput of 134 fps for an image of 480 x 320 pixels.

研究分野：VLSI画像処理

キーワード：CNN　意味分割　フロー推定　SLAM　FPGA

１．研究開始当初の背景

画像認識は、画像の解析によって画像上に投影されたエンティティの特性を推測するプロセスであり、パターン認識技術の一つです。物体検出は、画像内の物体を検出するタスクであり、画像認識の主要なタスクです。物体の輪郭検出は、画像内の物体間の境界を抽出するタスクです。これは、それ自体有用であり、物体検出の前処理としても利用されます。

ステレオビジョンによる視差推定は、平行に設置したカメラから同時に撮影された 2 つの左右画像に表れた画素の対応から視差を算出します。そして、カメラから物体への距離は三角法の原理によって決定されます。この方法は物体検出に使用され、物体の輪郭検出にも使用されます。ディープニューラルネットワーク(DNN)は、3 層以上のニューラルネットワークであり、近年画像認識への応用が進められ、急速に発展しています。カテゴリ分類は、画像に写った物体の種類を識別するタスクです。カテゴリ分類への DNN の適用は精度を劇的に向上させました。DNN の応用は、物体検出、画素単位でカテゴリ分類を行う意味的分割、2 つの画像間の対応する画素を見つける動き/視差推定、および物体輪郭検出に進化しています。

DNN の計算量は膨大であり、学習と推論には多大な電力が必要です。DNN 計算は通常浮動小数点数で行われますが、消費電力削減のために低ビット量子化と専用ハードウェアが開発されています。数値を 1 ビットに量子化すれば、畳み込みの積を XNOR で行い、合計をポップカウント(1 を数える)で行えるので、高い電力効率のハードウェアを実現できます。DNN の低ビット量子化とハードウェア実装の研究の大部分は、カテゴリ分類を目的としており、回帰アプリケーションはほとんどありません。

２．研究の目的

本研究では、ステレオビジョンを用いた一般的な物体輪郭検出用の畳み込みネットワークを提案します。提案された方法では、物体境界までの距離を画素値として輪郭画像が表されます。学習において本輪郭画像を推定すると、境界からの距離に応じて内側の偽陽性エッジのペナルティが増加し、偽陽性が減少します。ステレオビジョンの導入は物体内の非境界の誤検出をさらに減らします。

そして、完全に共有された重みを持つ輪郭検出と視差推定のためのマルチタスク CNN を提案します。視差推定の推定対象は、二つの画素間の距離であり、提案された方法による輪郭推定と同じ目標です。そこで全重みを共有したネットワークで両方の推定を同時実行します。このネットワークは、トレーニングと推論の計算量を半分に減らし、良好な精度を達成します。そして、提案されたネットワークの重みと活性化の両方を高精度で 3 ビットに量子化します。

さらに、内部メモリのみを使用した順伝播計算を特徴とする 3 ビット量子化ネットワーク専用のハードウェアを考案します。その結果、大量の電力を必要とするチップの外部でのデータ転送が不要となります。この回路は、8 行、16 入力チャンネル、16 出力チャンネル、3×3 画素の畳み込みを並列に計算します。動作周波数 250MHz の畳み込み計算スループットは 9 TOP/s です。3 ビットに量子化された輪郭検出と視差推定のためのマルチタスク CNN のハードウェアインプリメンテーションに関するレポートは見つかりませんでした。

３．研究の方法

（1）物体輪郭表現

一般に、画像中の境界の画素数は、非境界の画素数よりはるかに少なくなります。従来法では、この不均衡を解消するために、学習において境界を非境界と判断した場合、反対の場合と比較してペナルティを増やします。このとき、誤判定のペナルティは、境界からの距離とは無関係です。その結果、物体内のパターンのエッジが境界と見なされ、誤検出が増加し、精度が低下します。

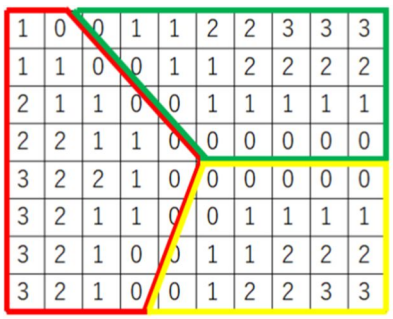

図１　物体輪郭表現

この問題を解決するために、物体輪郭の新しい表現法を提案します。この表現を図１に示します。図１は、3 つの領域の例です。領域境界の画素値は 0 です。非境界値は、常に 1 である従来の方法とは異なり、境界までの距離です。学習時に境界までの距離に応じたペナルティを課すことで、物体内のエッジに対する誤検出が減少します。

これは、コンピュータグラフィックスの衝突検出で使用される「ディスタンスフィールド」と同じ考え方です。物体輪郭検出にこのフィールドを使用する文献を見つけることができませんでした。本研究では、物体境界までの 8 近傍距離を画素値とする輪郭画像を入力画像から推定します。推定画像を閾値処理した後、物体の輪郭が検出されます。この表現は、輪郭検出を輪郭推定と呼ばれる回帰の問題に帰着させ、視差推定とのマルチタスクを容易にします。

（2）ネットワーク構造

マルチタスクのニューラルネットワークとして UNET を採用しました。UNET はエンコーダ・デ

コーダアーキテクチャで、同じ階層のエンコーダとデコーダをスキップ接続します。ネットワーク構成を図 2 に示します。レイヤー名'Enc'はエンコーダを表し、'Dec' はデコーダを表します。'cbr'は 3×3 画素の畳み込み、バッチ正規化、および ReLU を順番に実行します。'down'は水平および垂直方向に 1/2 の最大プーリングを行います。'up'はバイリニア補間で垂直および水平方向に 2 倍のアップサンプリングを行います。'cat' は下層からのアップサンプリング出力とデコーダと同じ階層のエンコーダ出力を連結します。'conv'は 3×3 画素の畳み込みを表します。レイヤー名に追加された数は、レイヤーの階層(解像度) レベルを示します。輪郭と視差の同時推定の場合、出力チャンネル数は 2 で、単独の場合、1 です。

No.	Name	Input channels	Output channels
1	Enc_cbr0	6	64
2	Enc_down1	64	64
3	Enc_cbr1	64	64
4	Enc_cbr1_1	64	128
5	Enc_down2	128	128
6	Enc_cbr2	128	128
7	Enc_cbr2_1	126	256
8	Enc_down3	256	256
9	Enc_cbr3	256	256
10	Enc_cbr3_1	256	256
11	Enc_down4	256	256
12	Enc_cbr4	256	256
13	Enc_cbr4_1	256	512
14	Enc_down5	512	512
15	Enc_cbr5	512	512

No.	Name	Input channels	Output channels
30	Dec_up0	1 or 2	1 or 2
29	Predict	256	1 or 2
28	Dec_up1	256	256
27	Dec_cbr2_1	384	256
26	Dec_cat2	256(Dec_cbr2),128(Enc_down2)	
25	Dec_cbr2	256	256
24	Dec_up2	256	256
23	Dec_cbr3_1	512	256
22	Dec_cat3	256(Dec_cbr3),256(Enc_down3)	
21	Dec_cbr3	256	256
20	Dec_up3	256	256
19	Dec_cbr4_1	768	256
18	Dec_cat4	512(Dec_cbr4),256(Enc_down4)	
17	Dec_cbr4	512	512
16	Dec_up4	512	512

図 2　ネットワーク構造

FPGA 実装を考慮して、元の UNET は次のように変更されます。各層の畳み込みの数を元の 3 から 2 に減らすことによって計算量が削減されます。プーリングの前ではなく、プーリングした後、同じ階層のデコーダに特徴量マップを接続することで、メモリ使用量が削減されます。活性化の量子化を容易にするために、エンコーダだけでなく、デコーダも畳み込み後にバッチ正規化を行います。

従来の動き推定ニューラルネット FlowNet はデコードで転置畳み込みを採用しています。転置畳み込みにはマスク処理が必要です。これは処理に必要な入力画素の位置はターゲット画素の位置によって異なるからです。この研究は転置された畳み込みをバイリニア補間と畳み込みの組み合わせに置き換えます。バイリニア内挿と畳み込みの組み合わせは簡単で、畳み込み回路を変更する必要はありません。

（3）量子化法

重みの量子化について説明します。最初のトレーニングは、重みの上限と下限をクリッピングしないで行います。2 回目のトレーニングは、最初のトレーニング結果を初期値とし、重み分布の標準偏差の 2 倍に上限と下限をクリッピングして行います。2 番目の結果の精度は、最初の結果の精度と変わりません。そこで、標準偏差の 2 倍を重み量子化の範囲に設定します。また、すべての畳み込みのバイアスをゼロとします。

アクティベーションの量子化について説明します。活性化関数の出力、つまり次の畳み込みの入力はアクティベーションと呼ばれます。このネットワークは、畳み込み後に常にバッチ正規化を実行し、ReLU による活性化を行います。バッチ正規化後のアクティベーションの分布は、平均値が 0 で、分散は 1 です。その後、ReLU が実行されるので、アクティベーションは 0 以上になります。したがって、全ての層に対してアクティベーションの量子化範囲を 0 以上 X 以下に設定します。ここで X は定数であり、全層同一です。

従来の量子化手順では、最初に重みとアクティベーションの両方が浮動小数点数であるネットワークをトレーニングします。次に、このトレーニング結果を初期値として、8 ビットネットワークをトレーニングします。その後、重みとアクティベーションを 1 ビットずつ減らしながら量子化ネットワークを訓練します。従来の手順を、すべての量子化されたネットワークの初期重みとして浮動小数点ネットワークのトレーニング結果を使用する手順と比較しました。予備的な実験の結果は、後者の手順がうまく機能することを示していました。

採用した従来の量子化ネットワークのトレーニング方法では、順伝播計算で量子化された重みと活性化を使用しますが、重みは浮動小数点数として格納され、逆伝播計算の勾配は浮動小数点数によって計算され、保存された重みは勾配によって更新されます。

4．研究成果

（1）アルゴリズム

我々は以前、従来法に対する提案された輪郭表現の利点を説明する実験結果を報告しています。今回は、各専用ネットワークと比較したマルチタスクネットワークの精度に関する実験結果を示します。量子化されたネットワークの実験結果についても説明します。

①実験方法　比較対象は次の 4 つのネットワークです。

- DN：視差推定専用の UNET
- CN：輪郭推定専用の UNET
- DC：視差と輪郭のマルチタスク UNET
- FN：視差と輪郭のマルチタスク FlowNet

DN と CN は、1 つの出力チャネルを持ち、構造は同じです。DC は DN と CN と同じ構造ですが、出力は 2 チャネルです。2 つの出力チャネルを持つ FN の構造は、2.3 節で説明されています。

我々はデータセットに FlyingThings3D と Driving を使用しました。これらは、動き推定と視差推定の研究のために開発されたデータセットです。正解輪郭画像は、データセットに含まれる物体インデックスイメージ（各物体に一意に割り当てられた数値を画素値として持つイメージ)から作成されました。

FlyingThings3D(FT) は、3D コンピュータグラフィックスによって描かれた非現実的なステレオ画像のデータセットです。画像では、机や椅子などの物体が立体空間を飛んでいます。解像度は 960×540 画素です。使用するシーンの数は 1,090、画像の数は 10,340 です。うち 90%がトレーニングに、10%が推論に使用されました。

Driving(DR)は、現実的なステレオ画像のデータセットです。車のドライバーの視点から前方を撮影した画像は、3D コンピュータグラフィックスで描かれています。表示される物体は、車、街灯、木、建物、道路です。私たちは、推論のために”15mm_focallength / scene_forward / slow”と”15mm_focallength / scene_backward / slow”を使用しました。各画像は 800 枚の連続ステレオ画像で、解像度は 960×540 画素です。私たちはすべてのを左上の原点を(120,0)として幅 720、高さ 360 でトリミングしました。推論にはシーケンスの最初の 700 枚が使用されました。

学習中の損失関数として L1 損失を用いました。L1 損失は、推定画像と正解画像の同じ位置の画素間で、差分の絶対値を求め、画像全体で平均化します。最適化方法は Adam、学習率は 1e-4、ミニバッチサイズは 8 でした。ランダムトリミングのサイズは、FT は 384×384、DR は 512×256 でした。FT のトレーニングでは、ランダムな初期化が行われ、50 エポックが実行されました。DR のトレーニングでは、FT のベストモデルからスタートし、25 エポックが実行されました。

次に量子化されたネットワークを実験しました。すべての畳み込みの重みの量子化範囲は-0.0625 から 0.0625 に設定されました。すべてのアクティベーションの量子化範囲は 0 ~ 3 に設定されました。これらの範囲は、予備実験によって決定されました。トレーニング手順は 3.3 節で説明しています。Adam の学習率は、4 ビット量子化まで 1e-6 でした。3 ビット量子化から、FT の場合は 1e-5、DR の場合は 1e-4 に設定しました。各量子化されたネットワークは、25 エポックの間、訓練されました。

実行環境に関しては、コンピュータの CPU はインテル Core i9-7900X 3.30GHz、メモリは 64 GB、GPGPU は NVIDIA GeForce GTX 1080 ti でした。ディープラーニングプラットフォームは PyTorch 0.4.1[37]を使用しました。PyTorch によって実装された FlowNet2 ソースコードを使用しました。

②実験結果　表 1 は 4 つのネットワークが FT および DR を推論したときの L1 損失を示します。各行は、輪郭推定(C)と視差推定(D)の L1 損失を表します。マルチタスクネットワーク DC の精度は、各専用ネットワークの精度に比べて若干低くなりますが、FN の精度よりも高くなっています。 図 3 および 4 は、それぞれ FT および DR に対する DC の結果を表します。輪郭検出結果は左画像にバイナリ輪郭を上書きすることによって作成されています。このとき輪郭検出のしきい値は 3(画素)です。FT では物体内部のエッジが輪郭として検出されていないことが分かります。DR では細かい木の葉の輪郭検出が困難です。FT の視差推定結果では、各物体の形状が明確に示されています。DR では窓越しに背景が現れる車両上部の推定が困難です。

表 1　各ネットワークの推定精度

データ	FN	CN	DN	DC	DC3
FT(C)	3.634	2.376	-	2.804	3.339
FT(D)	2.746	-	2.403	2.571	3.138
DR(C)	6.648	4.365	-	4.516	6.355
DR(D)	5.842	-	4.815	4.958	6.065

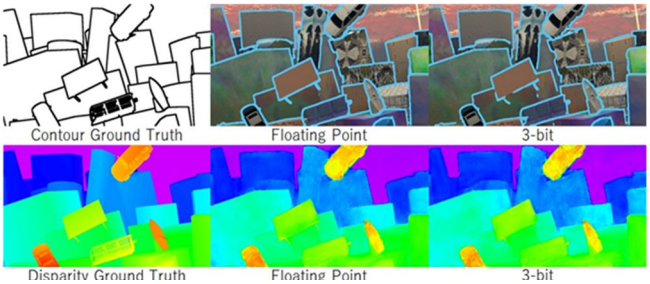

図 3　FlyingThings3D 結果

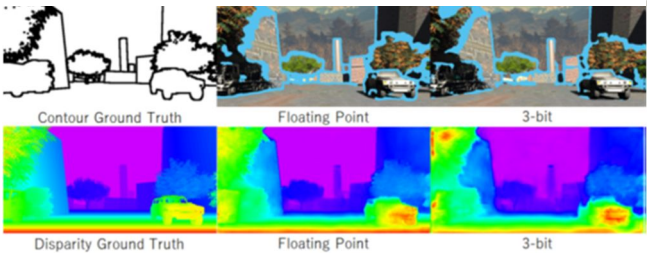

図 4　Driving 結果

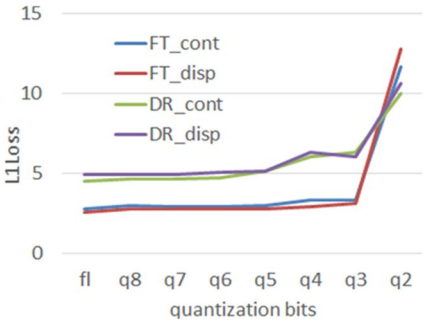

図 5　量子化結果

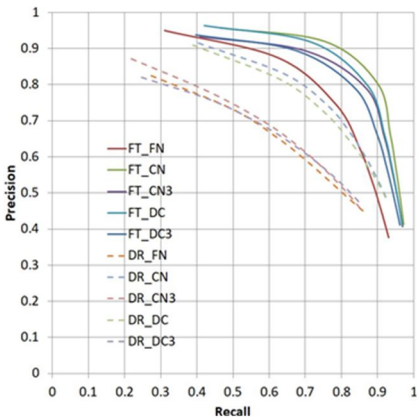

図 6　P-R 曲線

表 2　F-measure

データ	FN	CN	CN3	DC	DC3
FT	0.764	0.849	0.827	0.833	0.813
DR	0.641	0.752	0.654	0.738	0.652

マルチタスクネットワーク DC の量子化の実験結果を図 5 に示します。どちらの推論においても、ビット数が減少するにつれて精度が低下しました。3 ビットの精度は学習率を上げることで良好に保たれました。しかし、学習率を高くしても、2 ビットの精度は悪いままでした。3 ビット量子化の結果を DC3 として表 1、図 3、図 4 に示します。輪郭と視差は 3 ビットでかなり良く推定されました。

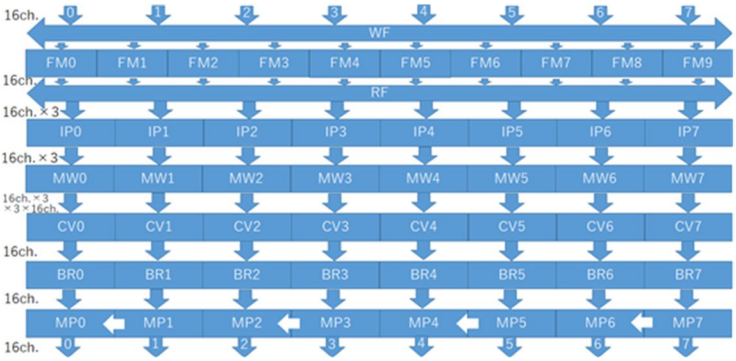

図 7　マルチタスク CNN プロセッサ

表 3　FPGA 諸元

LUT	369,516/1,182,240
URAM	528/960
BRAM	377/2,160
周波数	250 MHz
解像度	480 × 320
スループット	134 fps

輪郭検出の結果を詳細に説明します。推定精度の指標である precision、recall、および F-measure は次のように定義されます。

$$precision = \frac{tp}{tp + fp}$$

$$recall = \frac{tp}{tp + fn}$$

$$F - measure = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

ここで、tp は真陽性の数、fp は偽陽性の数、fn は偽陰性の数です。真陽性とは、境界を境界として正しく判断することです。偽陽性とは、境界以外を境界として誤って判断することです。偽陰性とは、境界を非境界と誤って判断することです。

precision-recall(P-R)曲線の作成方法について説明します。ネットワークからの輪郭推定画像に対して、しきい値以下の値を持つ画素を境界として判定し、最終的なバイナリイメージを作成します。しきい値は 1、2、3、4、5、6、7、8 としました。推定画像ごとにしきい値を変更することで、8 つのバイナリ画像が作成され、precision と recall が計算されました。各しきい値について、すべてのバイナリ画像の precision と recall が平均化され、P-R 曲線が 8 つの平均値で描かれました。

図 6 は、両方のデータセットに対して、FN、CN、3 ビット量子化 CN、DC、および 3 ビット量子化 DC の 5 つのネットワークの P-R 曲線を示しています。曲線がグラフ右上に近いほど、精度が高いことを示します。DC の精度は、CN よりも低いが、FN よりもはるかに優れていました。3 ビット DC の精度は、FT の場合は FN より優れていました。DR の場合は FN の場合と同等かそれ以上でした。両方のデータセットに対する 5 つのネットワークの F-measure を表 2 に示します。DC の F-measure は FT の場合 0.833、DR の場合は 0.738 でした。3 ビット DC の F-measure は、FT では 0.813、DR では 0.652 で、浮動小数点数の FN より高い値でした。

（2）ハードウェアアーキテクチャ

図 7 は物体輪郭・視差推定用マルチタスクプロセッサのブロック図を表しています。FM（Feature Memory）は 10 バンクで構成され、中央の 8 行に上下 2 行を追加した 10 行が連続で同時に読み出されます。各 FM バンクは 48 ビットワードであり、1 ワードは 16 チャンネルの 3 ビットアクティベーションを格納します。連続する 10 行の 1 列が、クロックサイクルごとに FM から読み出されます。RF（Read Feature）は、これを 3 つの連続した行の 8 つのデータに再構成し、IP（Interpolation）に送信します。IP と後ろに続くブロックは 8 つのユニットに分割され、8 行を並列処理します。各 IP ユニットは、高さと幅を 2 倍するバイリニア補間を行います。各 MW（Make Window）ユニットは、連続する 3 列を組み合わせて、3×3 アクティベーション×16 チャネルの畳み込みウィンドウを作成します。この 16 個のコピーが対応する CV（Convolution）ユニットに送信されます。CV は畳み込み計算を実行します。各 CV ユニットは、16 個の CVE（CV Element）回路で構成され、16 出力チャネルを並列処理します。16 個の出力チャネルは BR（Batch Normalize & ReLU）ユニットによって処理され、バッチ正規化と ReLU が実行されます。MP（Max Pooling）ユニットを使用した最大プーリングがこれに続きます。MP ユニットによって生成された 16 チャネルは、WF（Write Feature）を介して 1 ワードとして FM バンクに書き込まれます。BR に接続するパラメータメモリ PM および CV に接続する重みメモリ WM は、このブロック図では省略されています。

（3）FPGA 実装

提案された UNET ベースのマルチタスク CNN を FPGA に Xilinx 社の Alveo U200 へ実装しました。Alveo U200 は、画像処理と AI 推論用の FPGA を含むアクセラレータカードです。開発環境は Xilinx の Vitis 2020.1 を使用しました。プロセッサの諸元を表 3 に示します。

次に、Vitis AI と比較しました。Vitis AI は、ザイリンクスが開発した FPGA ベースの DNN プロセッサと、量子化やコムパイラツールを含む開発環境で構成されています。Caffe や Tensorflow などの機械学習フレームワークを使用して開発されたニューラルネットワークは、8 ビットで量子化され、コンパイルされ、DNN プロセッサ上で実行されます。700 MHz 周波数の DNN プロセッサによる UNET のスループットは 44 fps（フレーム/秒）です。一方、提案されたプロセッサのスループットは 134 fps です。

５．主な発表論文等

〔雑誌論文〕　計1件（うち査読付論文　1件／うち国際共著　0件／うちオープンアクセス　1件）

１．著者名 Miyama Masayuki	４．巻 1729
２．論文標題 FPGA Implementation of 3-bit Quantized CNN for Semantic Segmentation	５．発行年 2021年
３．雑誌名 Journal of Physics: Conference Series	６．最初と最後の頁 012004～012004
掲載論文のDOI（デジタルオブジェクト識別子） 10.1088/1742-6596/1729/1/012004	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 －

〔学会発表〕　計2件（うち招待講演　0件／うち国際学会　1件）

１．発表者名 Miyama Masayuki
２．発表標題 Convolutional Network for Generic Object Contour Detection with Stereo Vision
３．学会等名 2019 2nd International Conference on Information Hiding and Image Processing（国際学会）
４．発表年 2019年

１．発表者名 西沢　拓未, 深山　正幸
２．発表標題 意味分割用DNNの低ビット化
３．学会等名 2019年度電気・情報関係学会北陸支部連合大会予稿集
４．発表年 2019年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

－

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関