VAEを用いた画像圧縮，異常検知の応用研究
（題名もうちょっと良いのないかな）

今村　優希[†a)]　　　川崎　大雅[†b)]

Yuki IMAMURA[†a)] and Taiga KAWASAKI[†b)]

あらまし　ああ（大体300文字）

キーワード　VAE, FPGA, エッジコンピューティング

1. はじめに

近年，無線通信技術は飛躍的に向上しており，5G通信の普及が進んでいる．5Gは従来の4Gなど通信規格と異なり，「高速大容量」「低遅延」「多数同時接続」の3つの特徴を備えており，その中でも「低遅延」と「多数同時接続」は新たな通信環境を構築する上で重要な軸となっている[1].

従来の4G通信は，人が使用するスマートフォンや携帯に焦点を当てていた．しかし，5Gでは車両，ドローン，センサなどのIoT機器が大量にネットワークに接続されることを前提としている．このような環境においては，従来のクラウド中心の処理方式では，トラヒック増加による遅延や負荷集中が生じる可能性があり，5Gの利点を十分に発揮できない問題がある．また，今現在のIT業界ではクラウドが主流で，処理の多くを一つ（もしくは複数の）コンピュータで行うという構造である．多くの端末から取得したデータをクラウドのみで処理を行うのはある程度限界があり，またトラヒック量が増加して，5Gのメリットを享受できないという問題が発生すると考えられる．

このような課題を解決するため，エッジコンピューティングという技術が近年注目され始めている．エッジコンピューティングとは，従来はクラウドで行っていた処理の一部を，ユーザ端末（スマートフォンやIoT機器）の近い位置である基地局やその至近に設置されているサーバなどでデータ処理を行う技術である[2]. この技術を用いることでクラウドにかかる処理をエッジコンピューティングで分散することが可能で，通信のトラヒック量，5Gの特徴のひとつである「低遅延」に貢献することも可能である．

そこで，エッジコンピューティングの実現をVAEとFPGAを用いて実現することを考えた．(本文を書きながら続きを記述予定)

2. 方　　法

2.1 実行環境

今回の設計において使用したツール及びそのバージョンを表1で示す．

2.2 システム構造

今回のVAEを搭載システムでは，以下2つの機能を搭載した．

† 九州工業大学情報工学部 情報・通信工学科3年
福岡県飯塚市川津 680-4
Kyushu Institute of Technology, School of Computer Science and System Engineering, Department of Computer Science and Networks
a) E-mail: imamura.yuuki475@mail.kyutech.jp
b) E-mail: kawasaki.taiga000@mail.kyutech.jp

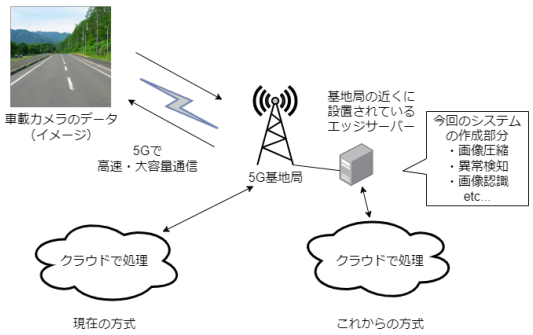

図 1　エッジコンピューティングのイメージと
今回のシステムの作成部分

表 1　使用したツール

用途	使用ツール
VAE シミュレーション用	MATLAB 2024b
ハードウェアシミュレーション	MATLAB/Simulink 2022a
HDL コード生成	HDL Coder
FPGA 設計ソフトウェア	Vivado xxxx
ハードウェアアクセラレーション	Vitis xxxx
評価ボード	DIGILENT 製 ZYNQ-7010

1　画像圧縮

VAE の特徴のひとつである次元圧縮能力を画像に応用する．

2　異常検知

もう一つの特徴である異常検知を，元画像と生成画像とを比較して行う．

上記の機能を実現するために，VAE の構造を 2.2.1 で説明する．また，FPGA の構造の詳細を 2.2.2 にて説明し，最後に SoC FPGA の構造を 2.2.3 にて説明する．

全体の構想について解説する．今回使用する画像は，簡単化のためにグレースケール化したものを使用する．また，JPEG のようにブロックに分割して，それぞれのブロック毎に処理を行う．ブロックサイズは 16 × 16 に設定した．また，画像圧縮や異常検知の判定は，元画像と圧縮後の画像との比較を PSNR を用いて行い

2.2.1　VAE 構 造

VAE の構造の概略を図 2 に示す．16 × 16 の画像を使用することから，入力 256 次元，出力 256 次元で設計を行った．潜在空間の次元は，次元圧縮と異常検知という目的を両立させるために，16 次元で設計を行った．エンコーダ部分の活性化関数に関しては，平均では出力そのままであるが，分散に関してはソフトプラス関数を使用している．また，デコーダ部分ではシグモイド関数を利用している．

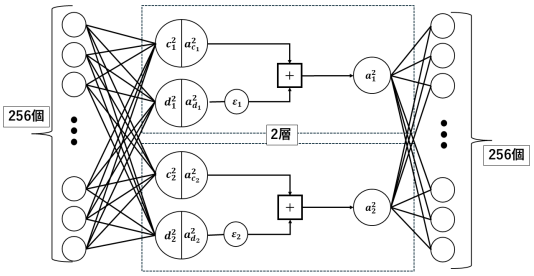

図 2　今回の VAE の構造

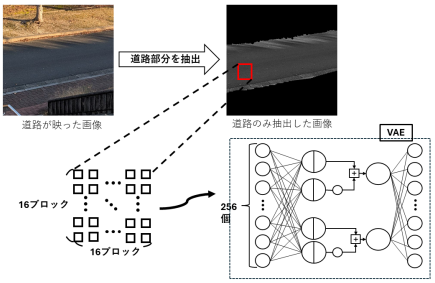

図 3　VAE 学習方法

$$f(x) = \log(1 + e^x) \quad : \text{ソフトプラス関数} \qquad (1)$$

$$f(x) = \frac{1}{1 + e^{-x}} \quad : \text{シグモイド関数} \qquad (2)$$

また，VAE の学習方法を図 3 に示す．まず，道路のみが映った画像を用意する．その画像に対して，道路のみ映った部分だけで 16 × 16 のブロックにする．そのブロックらを教師データとし，VAE を学習させる．今回のシステムでは，MATLAB で VAE の学習のみ行わせ，そこから出力された重みやパラメータを使用して LSI 設計を行っていった．

2.2.2　FPGA 構造

今回使用したボードは, DIGILENT 製の ZYNQ-7010 である．FPGA は，エンコーダ部分とデコーダ部分と大きく分けることができる．

まずは，エンコーダ部分を図 4 に示す．入力データとして，X，$w2$，$b2$ を使用する．赤色の枠で囲われているユニットは，

$$Output_1 = X_1 \times w2_1 \qquad (3)$$

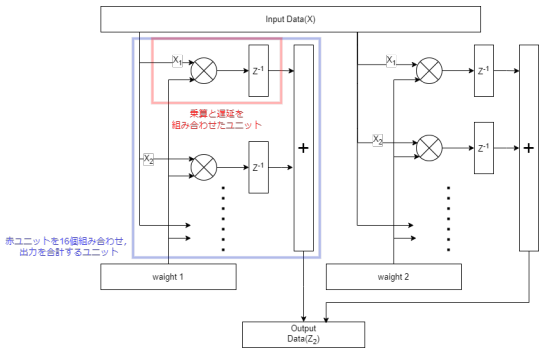

図 4　FPGA エンコーダ部分の構造

の演算を行い，パイプライン処理のために遅延を入れている．そのユニットを 16 個用意したものが，青色の枠線で囲われているユニットである．当初は 256 個を FPGA に載せたかったが，容量に限界があったため，16 個で設計を行った．青色のユニットでは，最終的に以下の計算を行っている．

$$Z_{21} = \sum_{i=0}^{16} X_j \times W_{2j} + b2_1 \tag{4}$$

そのユニットを 2 つ用意することで，2 つの取得を得られるように設計した．

次に，デコーダ部分を図 5 に示す．

2.2.3　SoC FPGA の構造

SoC FPGA のシステム構造の概要を図 6 にて示す．Processing System ユニットがバスを通って様々な処理を行う．

次に，SoC FPGA の処理の概要を図 7 にて示す．SD カードに VAE の学習重みや画像データが格納されているため，CPU が読み取る．その後，作成した FPGA に従って入力データの処理を行い，FPGA にそのデータを送信する．FPGA はデータが格納されると同時に実行し，出力結果を保存する．その出力結果を CPU が読み取りに行き，出力データを処理する．それらの処理を繰り返し行う．

今回の VAE が $256 \times 16 \times 256$ であり，2.2.2 で作成できた FPGA の構造が $16 \times 2 \times 16$ である．したがって，設計した VAE を実現するためには，FPGA に $16 \times 16 = 256$ 回稼働してもらう必要がある．

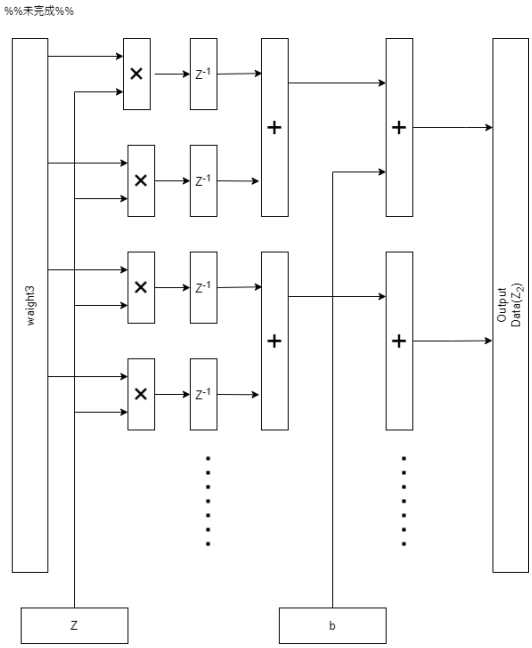

図 5　FPGA デコーダ部分

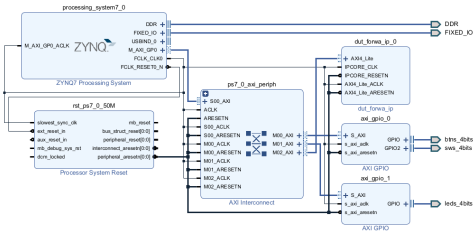

図 6　SoC FPGA の構成図

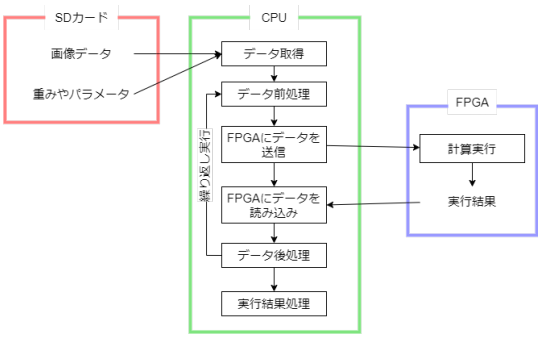

図 7　SoC FPGA の処理概要

3.　実行結果と考察

4.　結論と今後の展望

今回は，エッジコンピューティングを意識した VAE

と FPGA の手法に関して報告を行った．

謝辞　今回のシステム構築に対して，様々な支援を頂いた方々に感謝する．これからも，日本や世界を支えるエンジニアになるために尽力する．

文　献

[1] 森川博之, 5G 次世代移動通信規格の可能性, 岩波書店,

[2] 田中裕也, 高橋紀之, 河村龍太郎, ”IoT 時代を拓くエッジコンピューティングの研究開発”, NTT 技法ジャーナル, vol.27, no.8, pp.59-63, 2015.