

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 2022.1
ハードウェアアクセラレーション	Vitis xxxxx
評価ボード	DIGILENT 製 ZYNQ-7010

として、DIGILENT 製の SoC を搭載した ZYNQ-7010 を使用する。

## 2.2 システム構造

今回の VAE を搭載したシステムでは、以下 2 つの機能を搭載した。

### 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。

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  $16 \times 16$  に設定した。

画像圧縮のアルゴリズムは、ブロック毎に元画像と圧縮後の画像との比較を PSNR を用いて行う。PSNR が設定した閾値以上だった場合は精度良く圧縮ができているので圧縮した潜在空間の値を送信する。それ以下だった場合は、元の画像データを送信することで全体としての PSNR を維持する目的にした。異常検知のアルゴリズムは、今から通過するであろう部分に対して PSNR を用いて判定を行い、閾値以下だった場合

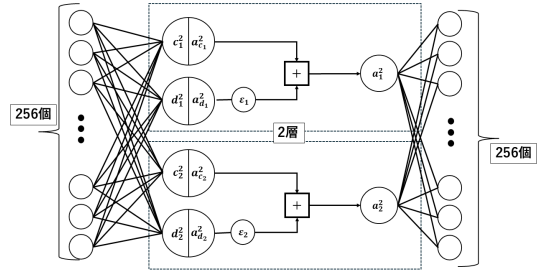


図2: 今回の VAE の構造

は道路以外の可能性が高いため異常と判定する。

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。

### 2.2.1 VAE 構造

VAE の構造の概略を図 2 に示す。  $16 \times 16$  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、圧縮空間いたとしてもある程度判別がつくように 16 次元で作成を行った。エンコーダ部分の活性化関数に関しては、平均では ReLU 関数を使用し、分散で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。

$$f(x) = x \quad : \text{ReLU 関数} \quad (1)$$

$$f(x) = \log(1 + e^x) \quad : \text{Softplus 関数} \quad (2)$$

$$f(x) = \frac{1}{1 + e^{-x}} \quad : \text{Sigmoid 関数} \quad (3)$$

また、VAE の学習方法を図 3 に示す。まず、道路のみのブロックを大量に用意する。そのブロックを教師データとし、VAE を学習させる。学習の条件を表 2 にまとめた。今回は道路が映った写真を用意し 3、道路のみ映った部分をブロック化することで教師データの用意を行った。また、MATLAB で VAE の学習のみを行い、そこから出力された重みやパラメータを使用して LSI 設計を行った。

### 2.2.2 FPGA 構造

使用したボードは、DIGILENT 製の ZYNQ-7010 で

表2: 学習条件

epoch	10000
eta	0.0005
Layer2(潜在空間の数)	16

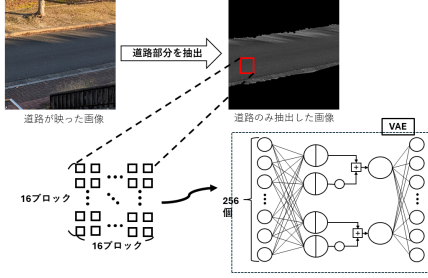


図 3: VAE 学習方法

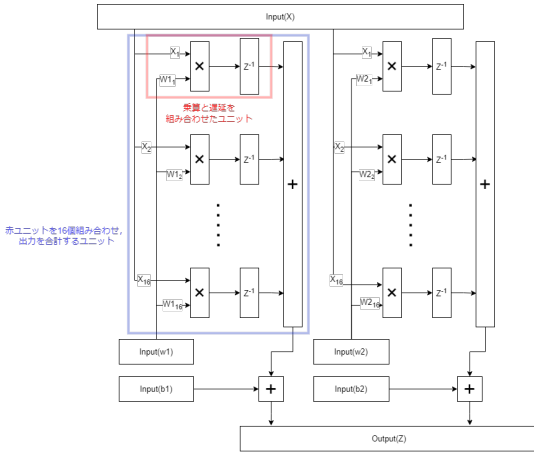


図 4: FPGA の構造

ある．今回の設計では，図 4 に示すように，入力データとして， $X(16)$ ， $W(16 \times 2)$ ， $b(16)$ ，出力として  $Z(16)$  を用意した（() 内は次元数）．赤色の枠で囲われているユニットは出力を  $Output_1$  とすると，

$$Output_1 = X_1 \times W1_1 \quad (4)$$

のような，入力と重みのパラメータを乗算する演算を行っている．その演算ユニットを 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．最終的な一つの  $Z$  の出力は，

$$Z_1 = \sum_{i=0}^{16} X_i \times W1_i + b1 \quad (5)$$

の計算を行っている．

当初は入力 256，出力 256 の演算を FPGA に載せたかったが，容量に限界があった．したがって，今回設計した  $256 \times 16 \times 256$  の VAE を実現しやすい， $X$  の入力が 16 になるように設計を行った．

表 3: FPGA のリソース利用率

Resource	Utilization	Available	Utilization[%]
LUT	3301	17600	18.76
LUTRAM	62	6000	1.03
FF	3297	35200	9.37
DSP	64	80	80.0
IO	12	100	12.0
BUFG	1	32	3.13

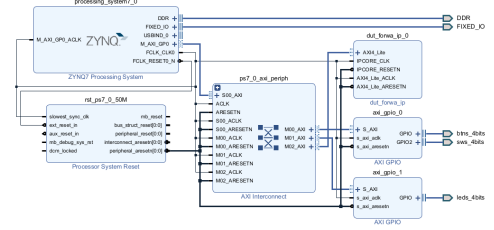


図 5: SoC FPGA の構成図

### 2.2.3 SoC FPGA の構造

SoC FPGA のシステム構造の概要を図 5 にて示す．Processing System ユニットがバスを通して様々な処理を行う．

次に，SoC FPGA の処理の概要を図 6 にて示す．SD カードに格納されている VAE の学習重みや画像データを 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す．

今回の VAE が  $256 \times 16 \times 256$  であり，2.2.2 で作成した FPGA の構造が  $16 \times 2$  である．設計した FPGA でエンコーダを計算する際は，図のような動作を行う必要がある．一つの出力で  $z_{mean}^2$  を，もう一つの出力で  $z_{var}^2$  を計算させる．1 つの潜在空間を計算させるためには，FPGA を 16 回使用する必要がある．その潜在空間が 16 個あるので，FPGA は合計で  $16 \times 16 = 256$  回稼働する．デコーダ部分を計算する際は，FPGA を一回稼働させるだけで，3 層目  $Z_i^3$  の出力を 2 つ得られる．したがって，デコーダでは 128 回 FPGA を稼働させる．このような制御を CPU で作成した．

実装した後のリソース利用評価を表 3 に示す．

### 2.3 実験方法

今回は，図 7 のような車載カメラから取得した画像をイメージとしたものを利用する．また，プロトタイプの作成として，画像の一部のブロックに対して VAE

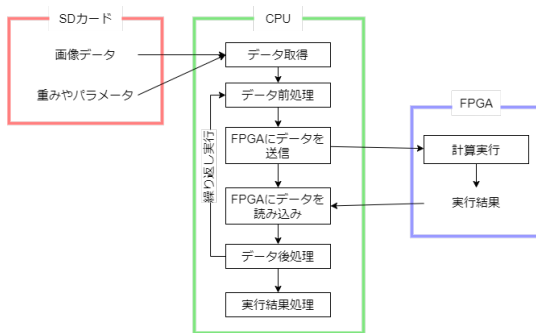
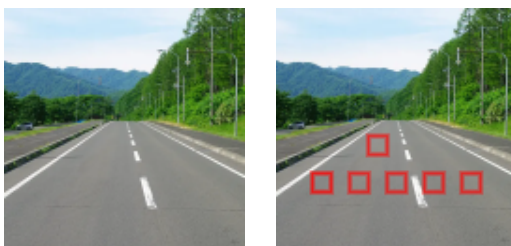


図 6: SoC FPGA の処理概要

(a) 車載カメラからの  
イメージ図

(b) 評価対象例

赤枠の部分を VAE にて評価する

図 7: 今回のシステムで評価する方法  
これいらんかもな

(a) 評価する画像

(b) 評価画像をグレイに変更

図 8: 今回テストする評価画像

を活用した画像圧縮、異常検知を行うようにした。一部のブロックの選出については図 7a のように設定しており、車がこれから通過するであろう部分を判定するように設定した。図 7a における赤枠の大きさは、 $16 \times 16$  である。

実際に、図のようなテスト画像を用意した。実際に VAE で判定させるものは図の白黒画像である。この図は、道路に落下物が落ちていることを想定しており、今回のシステムにおける異常検知の実施を行う。

### 3. 実験と考察

#### 3.1 実験結果

20250130:1600 時点で FPGA を用いたシステムを作成することができた。本日中に実験結果、考察をまとめる予定である。

#### 3.2 考察

出力された画像とシミュレーションした画像の比較を行う。

第一に、MATLAB でシミュレーションした結果と、FPGA を用いて出力した値が異なっていること点である。これに関しては、使用変数を見直したり、固定小数点数を変更することで解決が可能な検証していく。課題の 2 つ目としては、実行時間が長すぎることである。実行時間に関しては、評価ボードの制約やバスを用いた通信によって遅延が生じていると思われる。したがって、実際に産業応用を考える際は、より高性能な FPGA 等を用いることで解決を図ることが可能だと思う。ただ、VAE を FPGA を用いて実現し、画像圧縮、異常検知を行えることが示すことができたため、これからの研究・開発に活用できるのではないかと考える。

### 4. 結論と今後の展望

今回は、VAE を FPGA を上に実装し、道路画像の圧縮及び異常検知を行うシステムを構築した。その結果、FPGA を用いたエッジコンピューティングにおける画像処理が実現可能であることを示した。

本システムにはいくつかの課題があるものの、適切な改善により解決可能であると考えられる。したがって、将来的な産業応用の可能性も秘めていると思う。

今回のコンテストで FPGA を用いた LSI 開発を体験することができた。FPGA の将来性を感じたし、AI が日常的に使用されている現在に計算を効率よく行うために必要なデバイスであることを認知した。これからも FPGA の能力を発揮できるような開発を行っていききたいし、高位合成等の FPGA を開発するための技術も日々進化しているので、様々なことに挑戦していきたい。

**謝辞** 今回のコンテスト開催にあたり、主催して頂いた委員会の皆さん、協賛して頂いた企業・自治体の皆さん、本当にありがとうございました。また、研究室にも配属されていないのに、様々な支援をして頂いた先生方、ありがとうございました。

## 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.