卒 業 論 文 概 要 書

Summary of Bachelor's Thesis

Date of submission: 02/01/2022 (MM/DD/YYYY)

学科名 Department	情報通信	氏名 Name	飯野景	指 導 教 員 Advisor	渡辺 裕 印
研究指導名 Research guidance	オーディオビジュアル 情報処理研究	学籍番号 Student ID number	1w183008-8		
研究題目 Title	Collaborative intelligence における中間層特徴マップの情報圧縮 Compression of Feature Maps in the Middle Layer for Collaborative Intelligence				

1. まえがき

近年，DNN モデルを分割し，前段をディバイスに，後段をサーバに配置する Collaborative Intelligence [1]と呼ばれるコンピュータアーキテクチャが注目されている．多くの DNN モデルではサーバに送信される中間層特徴マップが入力のサイズよりも大きくなるという問題がある[2,3,4]．また，既存の中間層特徴マップの圧縮に関する研究では，入力を静止画に限定した研究が行われている．そこで，本研究では動画像，とりわけ定点映像を入力としたときの中間層特徴マップの圧縮手法を提案する．

2. 関連研究

2.1 ビデオコーデックを用いた圧縮

特徴マップのチャネル全体をタイル状に並べるタイリングや，隣接するピクセルが異なるチャネルで構成されるキルティングという方法で再配置し，ビデオコーデックを用いて圧縮する[2]．

2.2 ボトルネック構造による圧縮

ネットワークの分割点に学習可能なボトルネック構造を入れ込むことで圧縮を図る方式[3]で，他手法との併用が容易となっている．例えば，蒸留を行うことで効率的に学習する方法がある[4]．

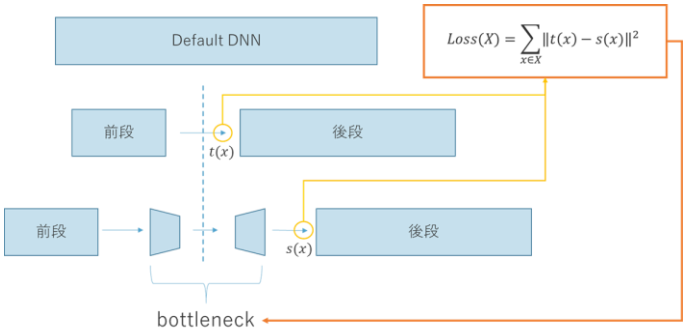

図 2.2.1 ボトルネック構造およびその学習法

2.3 クリップ処理を用いた圧縮

Cohen らにより提案された手法で，前段からの出力に対して，クリップ，量子化，二値化，エントロピー符号化を順に施すことで圧縮を行う[5]．

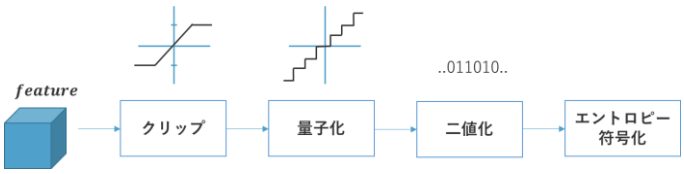

図 2.3.1 ボトルネック構造およびその学習法

3. 提案手法

時間方向の高い相関を利用するために，前処理の後に隣接する復元した中間層特徴マップとの差分を生成する．この差分特徴マップに対して二段階のクリップ，量子化を行う．最後に差分特徴マップのスパース性とチャネル方向の相関を生かすために，チャネル方向のゼロラン圧縮を施す．

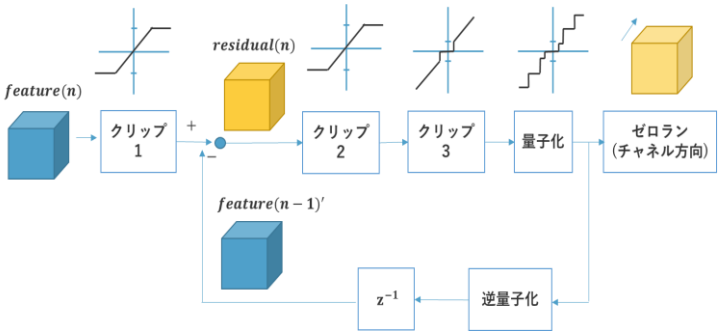

図 3.1 提案手法

4. 評価実験

物体検出における精度と圧縮率という観点で，入力動画の HEVC 圧縮と，中間層特徴マップの HEVC 圧縮（2.1 を動画像に拡張したもの）および提案手法による圧縮の比較を行った．HEVC では crf を変化させ，提案手法ではクリップ 3 のパラメ

タを変化させ，その他のパラメタは固定して実験を行った．評価用データセットとしては符号化難易度が異なる 5 系列の動画を用意した．YOLOv3，Faster R-CNN を使用し．それぞれに二種類（bottleneck1,2），1 種類（bottleneck3）のボトルネック構造を学習させ，実験で使用した．特徴マップの HEVC 圧縮は bottleneck1,3 とのみ併用して実験を行った．各ネットワークにおける全動画の平均の情報量対精度のグラフを図 4.1~4.4 に示す．

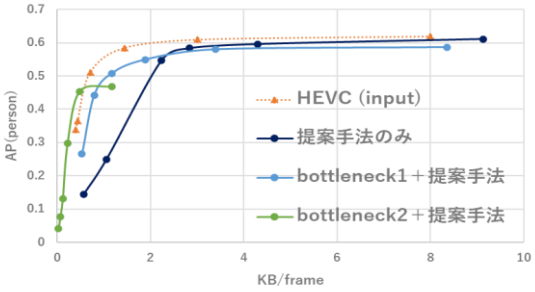

図 4.1 入力動画の HEVC 圧縮との比較（YOLOv3）

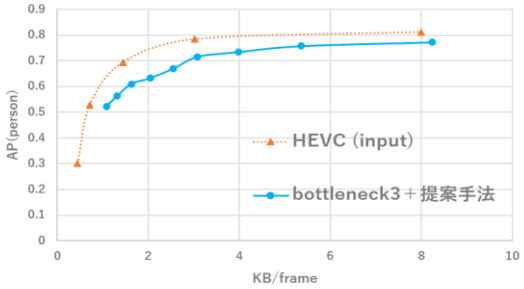

図 4.2 入力動画の HEVC 圧縮との比較（Faster R-CNN）

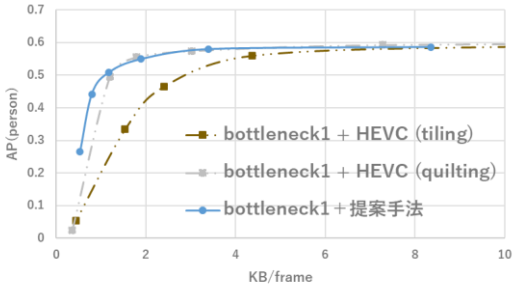

図 4.3 特徴マップの HEVC 圧縮との比較（YOLOv3）

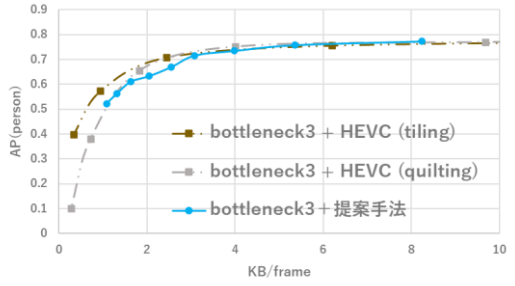

図 4.4 特徴マップの HEVC 圧縮との比較（Faster R-CNN）

図 4.1,4.2 からみられるように，一部を除いて圧縮率と精度の観点で入力動画の HEVC 圧縮より良い性能を示すことはできなかった．しかしながら，精度の大幅な低下はあるものの YOLOv3 の bottleneck2＋提案手法は狭帯域化において入力動画の HEVC 圧縮より同精度で高い圧縮率を示している．また，既存手法の拡張である中間層特徴マップの HEVC 圧縮と比較すると YOLOv3 においてはやや良い性能を示し，Faster R-CNN においてはやや劣る結果となった．これは各ネットワークの中間層特徴マップのサイズや中間層の深さに起因するものであると推察する．

5. 結論

本研究では，Collaborative Intelligence における動画像（定点映像）入力時の中間層特徴マップの圧縮手法を提案した．既存の研究では入力を静止画像に限定していたため，動画像の中間層特徴マップ圧縮は初めての試みであると考えられる．入力動画の HEVC 圧縮，および中間層特徴マップの HEVC 圧縮（既存手法の拡張）と情報量対精度の観点で比較し，評価を行った．入力動画の HEVC 圧縮に対しては限られた条件下でのみ優位性を確認できた．中間層特徴マップの HEVC 圧縮との比較ではネットワークによって異なる結果を得た．実用化に対しての有効性は不十分であるため，圧縮率および精度の向上に努めたい．また，本手法は明らかに HEVC より単純な圧縮手法であるため，計算量の比較も今後の課題としたい．

参考文献

[1] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang: “Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge,” ASPLOS '17: Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 615-629, Apr. 2017.

[2] H. Choi and I. V. Bajić: “Deep Feature Compression for Collaborative Object Detection,” IEEE International Conference on Image Processing (ICIP), pp. 3743–3747, Oct. 2018.

[3] A. E. Eshratifar, A. Esmaili, and M. Pedram: “BottleNet: A Deep Learning Architecture for Intelligent Mobile Cloud Computing Services,” IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), pp. 1-6, Jul. 2019.

[4] Y. Matsubara, D. Callegaro, S. Baidya, M. Levorato, and S. Singh: “Head Network Distillation: Splitting Distilled Deep Neural Networks for Resource-Constrained Edge Computing Systems,” IEEE Access, vol. 8, pp. 212177-212193, Nov. 2020.

[5] R. A. Cohen, H. Choi, and I. V. Bajić: “Lightweight Compression Of Neural Network Feature Tensors For Collaborative Intelligence,” IEEE Open Journal of Circuits and Systems, vol. 2, pp. 350-362, May 2021.

2021 年度　卒業論文

Collaborative intelligence における
中間層特徴マップの情報圧縮

Compression of Feature Maps in the Middle Layer for Collaborative Intelligence

提出日：2022 年 2 月 1 日

指導教員　渡辺 裕 教授

早稲田大学基幹理工学部 情報通信学科

1W183008-8

飯野 景

目次

第１章　序論

１．１　研究の背景

近年, 機械学習とその効率的な実装の進歩に伴い, 自動運転, スマートシティなど様々なアプリケーションにディープニューラルネットワーク（DNN）が活用されるようになっている．しかしながら，高精度な DNN モデルは非常に多くの層とパラメタを有しているため，ディバイス側にそのままモデルを配置することは非常に困難となる．そのため，このようなアプリケーションは，軽量な DNN モデルを設計し，ディバイスに配置して処理を行う Local Computing と，エッジサーバ / クラウドサーバに通常の DNN モデルを配置してディバイスと通信して処理を行う，Edge Computing / Cloud Computing の大きく分けて二つの方法を採用している[1]．軽量な DNN モデルは通常，本来のモデルより精度が落ちる．一方で通常のモデルを用いるためにサーバと通信して処理を行う場合，通信チャネルの影響によりリアルタイム性が大きく損なわれる可能性がある．

今日，これらに並ぶ新たなコンピューティングアーキテクチャとして，Collaborative Intelligence（Split Computing）[2]と呼ばれる手法が注目されつつある．Collaborative Intelligence は，先の二つの間を取るようなアーキテクチャで，DNN モデルを前段と後段に分割し，前段をディバイスに，後段をエッジサーバ / クラウドサーバに配置するという手法である．さらに，ディバイス側に簡易的な後段を加えることで，簡単な入力はディバイスで処理し，難しい入力はサーバに回すといった，入力の難易度に応じた適応的な処理を実現することができる（Early Exiting[3]）．

Collaborative Intelligence では，前段からの出力である中間層の特徴マップがサーバに送信されることになるが，多くの DNN モデルの初期層ではこの特徴マップが入力のサイズよりも大きくなるという傾向がある[4,5,6,7]．そのため，中間層特徴マップの情報圧縮が Collaborative Intelligence の実用化の鍵を握っていると考えられる．

１．２　関連研究と問題点，および研究目的

コンピュータビジョンの設定課題における既存の中間層特徴マップ圧縮に関する研究では，入力を静止画に限定した研究[4,5,6,7]が行われており，動画像を入力の対象とした研究はほぼ皆無である．コンピュータビジョンに関連する多くのアプリケーションは動画像を入力とするため，Collaborative Intelligence の実用化には，動画像を対象とした研究が必要不可欠であると考える．

そこで本論文では，動画像，とりわけ監視カメラのような定点カメラで撮影した映像を入力としたときの中間層特徴マップの圧縮手法を提案する．

１．３　本論文の構成

以下に本論文の構成を示す.

第１章　本章であり，本研究の背景，関連研究と問題点，および研究目的について述べる.

第２章　コンピュータアーキテクチャの種類，および既存の中間層特徴マップの圧縮手法について述べる.

第３章　本研究の提案手法について述べる.

第４章　本研究で行った実験の概要，結果および考察について述べる.

第５章　結論と今後の課題について述べる.

第２章　　関連研究

２．１　　まえがき

本章では, いくつかのコンピュータアーキテクチャおよびコンピュータビジョンタスクにおける代表的な中間層特徴マップの圧縮手法の概要について述べる．ここでのコンピュータアーキテクチャは，DNN モデルの配置方法という意味合いで分類しており，Local Computing, Edge Computing / Cloud Computing, 本研究のベースである Collaborative Intelligence についての簡単な説明を行う．本来，Edge Computing が Local Computing を含んだ意味で定義されることが多いが，ここでは本研究のテーマである Collaborative Intelligence の立場が明確になるように，先述のような分類をする．中間層特徴マップの圧縮手法としては，代表的な三つの手法である，ビデオコーデックを用いた圧縮，ボトルネック構造による圧縮，クリップ処理を用いた圧縮について解説する．

２．２　　コンピューティングアーキテクチャ

２．２．１　　Local Computing

ディバイス内に DNN モデルを配置し，処理・分析を完結するアーキテクチャである（図 2.2.1.1）．サーバに入力データを転送する必要がないため，低遅延で出力を得ることができ，セキュリティリスクの低減，リアルタイム性に優れるといった利点がある．その一方で，ディバイスの計算能力やメモリの制約上，パラメタ数の多い高精度のモデルを配置することが困難となる．そこで，大きなモデルを刈り込みや量子化によって圧縮する方法や，より軽量に設計されたモデルが提案されている．

図 2.2.1.1 Local Computing

２．２．２　Edge Computing / Cloud Computing

ディバイス外，つまりエッジサーバ / クラウドサーバに DNN モデルを配置し，ディバイスと通信を行いながら処理・分析を行うアーキテクチャである（図 2.2.2.1）．基本的にサーバはディバイスよりリソースが豊富なため，高精度な DNN モデルを配置することが可能である反面，入力データをネットワークを介して受け取る必要があるため，収集可能なデータ数に制限があり，通信環境によっては大幅な遅延が生じる可能性がある．

図 2.2.2.1 Edge/Cloud Computing

２．２．３　Collaborative Intelligence

DNN モデルを特定の層で分割して，前段をディバイスに，後段をサーバに配置するというアーキテクチャである（図 2.2.3.1）．近年注目されているアーキテクチャであり，ディバイスとサーバでの計算負荷の分散，タスク指向の圧縮を確立しデータの転送遅延を低減することが期待されている．また，図 2.2.3.2 のようにディバイス側に軽量な後段を追加することで簡単な入力はディバイスで処理し，難しい入力はサーバに回すといった，入力の難易度に応じた適応的な処理を実現することができる(Early Exiting[3])．Collaborative Intelligence では，前段からの出力である中間層特徴マップがサーバに送信されることになるが，多くの DNN モデルの初期層ではこの中間層特徴マップが入力のサイズよりも大きくなるという傾向がある[4,5,6,7]．そのため，後述する中間層特徴マップの圧縮が重要な研究テーマとなる．

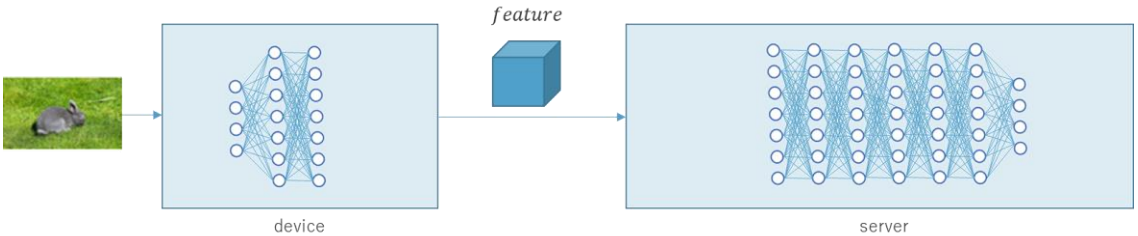

図 2.2.3.1 Collaborative Intelligence

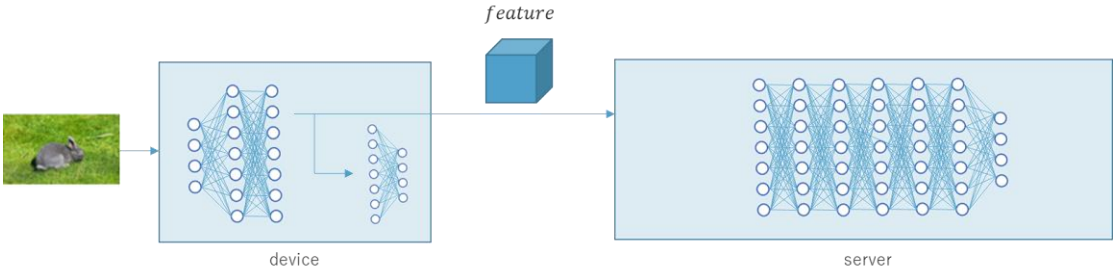

図 2.2.3.2 Early Exiting

２．３　中間層特徴マップの圧縮

前述したように単純な分割を行うと，前段からの出力である中間層特徴マップのサイズは入力画像のサイズより基本的に大きくなる．そのため可逆圧縮は有効ではなく，精度の低下を抑えた非可逆圧縮の検討が重要となる．興味深いことに，中間層特徴マップは 8bit 以上の量子化であればほとんど精度を落とさないことが分かっている[8]．以下ではより高度な圧縮法について述べる．

２．３．１　ビデオコーデックを用いた圧縮

特徴マップは縦×横×チャネル（H*W*C）の形状を取る．H*W を静止画，C を動画像のフレーム数とみなすことで，ビデオコーデックを用いて圧縮することが可能となる．また，複数のフレームを一枚の静止画として統合することにより，C フレーム以下の動画像を作ることができ，すべてのチャネルを一枚の画像に統合することもできる．チャネルを静止画に統合する方法としては，チャネル全体をタイル状に並べるタイリングや，隣接するピクセルが異なるチャネルで構成されるキルティングという方法が挙げられる．[4]では，Yolo9000 を使用し入力が静止画の場合，タイリングによって一枚の静止画を作り代表的な高効率ビデオコーデックである HEVC で圧縮することで，入力画像の JPEG 圧縮より良い情報量対精度を得ることが示されている．

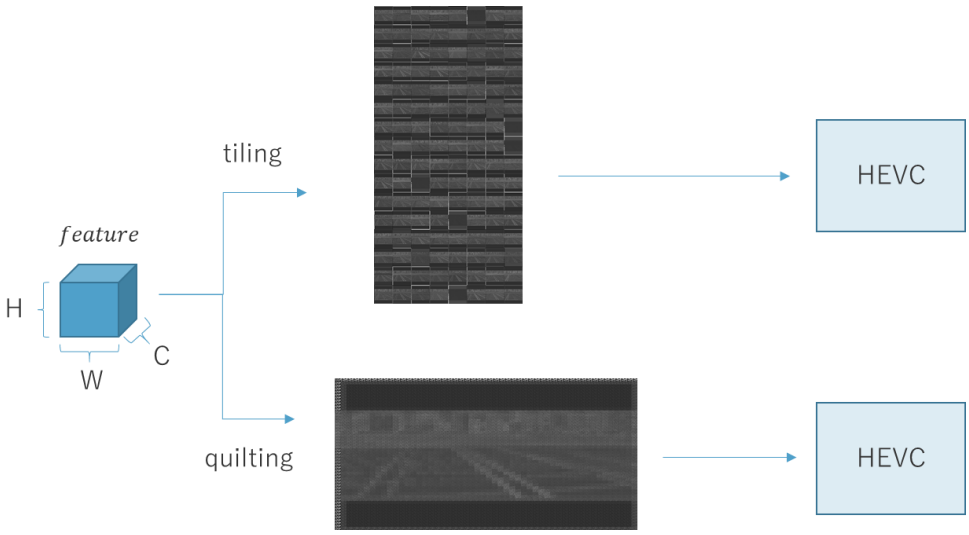

図 2.3.1.1 ビデオコーデックを用いた圧縮の様子

２．３．２　ボトルネック構造による圧縮

ネットワークの分割点にボトルネック構造を入れ込むことで圧縮を図る方式である（図 2.3.2.1）．A. E. Eshratifar らの研究ではボトルネック構造は畳み込み層，バッチ正規化層，活性化関数で構成されるため，学習可能な構造となっており，フィルタサイズやチャネル数を変えることで出力される特徴量のサイズを調整する[5]．また，他の圧縮手法と容易に併用できる点も特徴である．Matsubara らは，追加したボトルネック構造を学習させる際に全体のネットワークを再学習させることなく，元のモデルの前段を教師モデル，ボトルネック追加後の前段を生徒モデルとして蒸留を行い学習させることを提案している[6]．これにより大幅な学習時間の短縮を実現している．本研究においてもこの手法を採用してボトルネックの学習を行った．

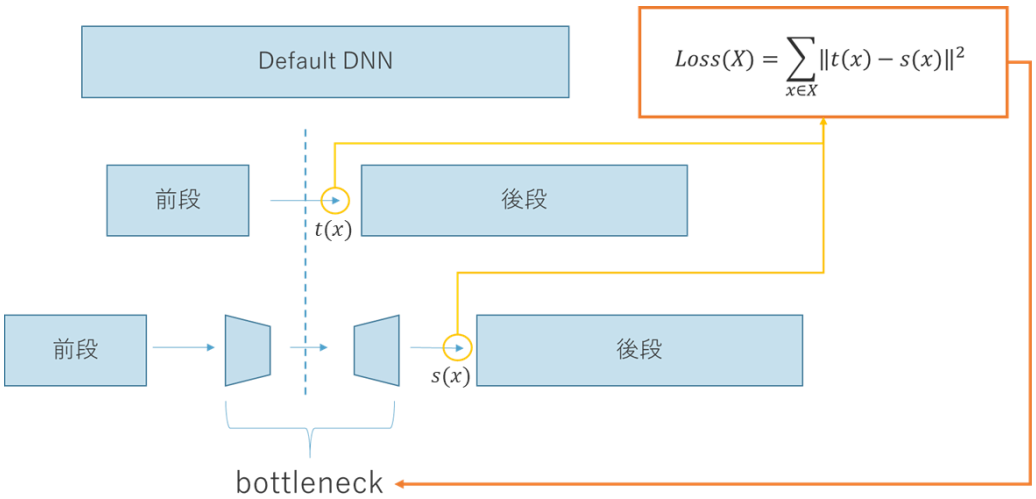

図 2.3.2.1 ボトルネック構造および学習の様子

２．３．３　クリップ処理を用いた圧縮

Cohen らにより提案された手法で，前段からの出力に対して図 2.3.3.1 のように，クリップ，量子化，二値化，エントロピー符号化を順に施すことで圧縮を行う[7]．この手法では再学習を必要とせず，シンプルながらも効果的な圧縮を実現している．本研究の提案手法においても，この手法を参考にクリップ処理を取り入れている．

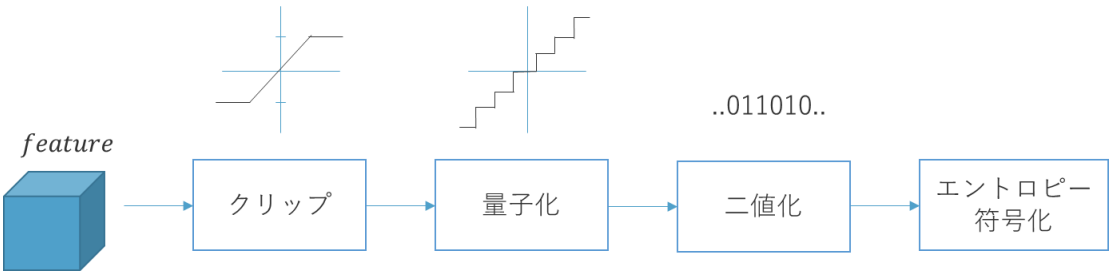

図 2.3.3.1 クリップ処理を用いた圧縮

２．４　むすび

本章では，Local Computing，Edge Computing / Cloud Computing，本研究のベースである Collaborative Intelligence についての簡単な説明を行った．また，代表的な中間層特徴マップの圧縮手法である，ビデオコーデックを用いた圧縮，ボトルネック構造による圧縮，クリップ処理を用いた圧縮の概要を述べた．ビデオコーデックやクリップ処理を用いた圧縮はネットワークの構造には手を加えないため，ボトルネック構造による圧縮と併用が可能である．また，後述する提案手法も同様であるため，ボトルネック構造と併用した実験も行った．

第３章　提案手法

３．１　まえがき

本章では，定点映像を入力としたときの中間層特徴マップの圧縮手法を提案する．時間方向の高い相関を利用するために，前処理をした後に隣接する中間層特徴マップの差分を生成し，クリップ，量子化，チャネル方向のゼロラン圧縮を施す．本手法は再学習を必要としないため，ボトルネック構造と併用することが可能である．比較対象となるのは入力動画のビデオコーデック（HEVC[9]）による圧縮であるが，本手法はこれらと比較して非常に軽量な圧縮手法であると考えられる．

３．２　提案手法

以下の(1)~(5)の手順で中間層特徴マップの圧縮を図る．ブロック図を図 3.2.1 に示す．

（１）前処理として，クリップを行う． $feature(n)$をn番目のフレームの中間層特徴マップ，$feature(n,p)$を中間層特徴マップの特定のピクセル，クリップの上限値を $clip_{max}$，下限値を $clip_{min}$とすると

$$if\ \ feature(n,p) \geq clip_{max} : feature(n,p) = clip_{max} \tag{3.2.1}$$
$$if\ \ feature(n,p) \leq clip_{min} : feature(n,p) = clip_{min} \tag{3.2.2}$$

（２）あるフレームの中間層特徴マップと，（４）の後に逆量子化をした 1 フレーム前の中間層特徴マップ$feature(n-1)'$との差分をとる．$residual(n)$を差分特徴マップとすると，

$$residual(n) = \begin{cases} feature(n) & when\ n\ mod\ T = 0 \\ feature(n) - feature(n-1)' & when\ n\ mod\ T \neq 0 \end{cases} \tag{3.2.3}$$

となり，差分特徴マップを得る．この操作により時間方向の高い相関の除去を行う．

（３）2 段階のクリップ処理を行う．まず，上限値，下限値を定めクリップ処理を行う．クリップの上限値を $clip_{2_{max}}$，下限値を$clip_{2_{min}}$とすると，

$$if\ \ residual(n,p) \geq clip_{2_{max}}\ \ then\ \ \ residual(n,p) = clip_{2_{max}} \tag{3.2.4}$$
$$if\ \ residual(n,p) \leq clip_{2_{min}}\ \ then\ \ \ residual(n,p) = clip_{2_{min}} \tag{3.2.5}$$

次に，差分特徴マップの値が 0 に近いピクセルは 0 に置き換える．閾値を$clip_3$とすると，

$$if \quad |residual(n,p)| \leq clip_3 \quad then \quad residual(n,p) = 0 \tag{3.2.6}$$

（4）量子化を行う．量子化レベル数をkとすると，

$$\widetilde{residual}(n) = round\left(\frac{residual(n) - clip_{2_{min}}}{clip_{2_{max}} - clip_{2_{min}}} \cdot (k-1)\right) \tag{3.2.7}$$

（5）チャネル方向にゼロラン圧縮を行い，チャネル方向の相関除去を行う．

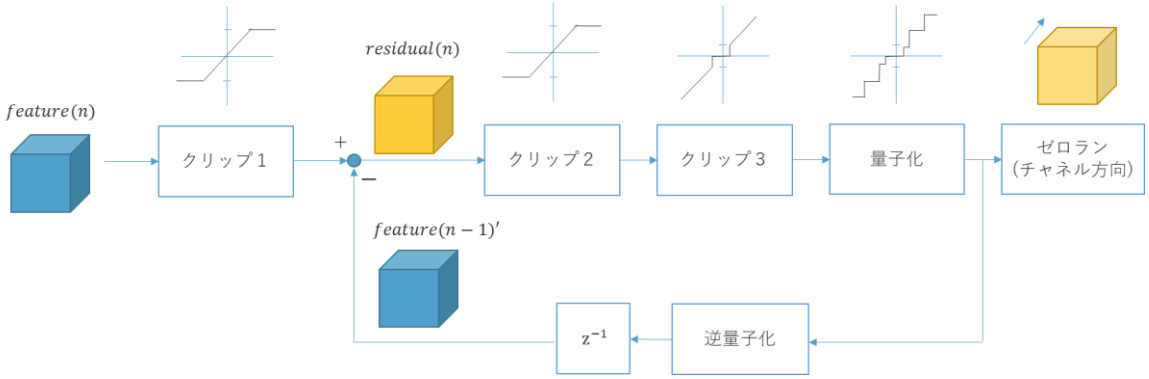

図 3.2.1 提案手法のブロック図

3．3　むすび

本章では，本研究で提案する定点映像を入力としたときの中間層特徴マップの圧縮手法について述べた．提案手法では，時間方向の高い相関を生かすために，隣接フレームの特徴マップの差分を生成し，その差分特徴マップに対して二段階のクリップ処理，量子化を行う．最後にゼロラン圧縮をチャネル方向に施すことで，差分特徴マップのスパース性を生かしたチャネル方向の相関除去を図る．

第４章　実験結果・考察

４．１　まえがき

本章では, 本研究での提案手法に基づく実験の概要, 結果および考察について述べる. 実験では物体検出時の精度と圧縮性能を調べることで評価を行った.

４．２　使用したネットワーク

物体検出に使用するネットワークとして, YOLOv3[10]と Faster R-CNN[11]を選択した. 提案手法を既存手法のボトルネックと併用した実験を行うために, YOLOv3 では独自でボトルネック層の追加，学習を行った．Faster R-CNN については Matsubara らが公開している学習済みのボトルネックを持ったネットワーク[12,13]を本実験に使用した.

４．２．１　YOLOv3

1 段階の物体検出器となっており，基幹ネットワークとして Darknet53[10]を採用している．Cohen らの研究[7]を参考に，第 12 層でネットワークを切断し，ボトルネックはその直後に圧縮層と復元層を追加することで構成した．ボトルネックは２種類用意し，第 12 層から出力される中間層特徴マップのチャネル数のみを 1/2 にする bottleneck1，チャネル数を 1/2 かつ縦横もそれぞれ 1/2 の計 1/8 に圧縮する bottleneck2 となっている．ボトルネックの学習には，COCO minitrain[14]という，COCO 向けに厳選されたミニトレーニングセット（COCO の約 20%のデータ数）を使用した．学習方法は[]で提案された方法を採用しており，元の YOLOv3 の第 12 層から出力を教師データ，ボトルネックからの出力を生徒データとして，損失関数に平均二乗誤差を設定し，蒸留を行うことで学習を行った.

４．２．２　Faster R-CNN

2 段階の物体検出器となっており，本実験では基幹ネットワークの ResNet50[15]の conv2_x を改変しボトルネックを組み込んだモデル[12]を使用した. 公開されているコード[13]には中間層の出力チャネル数が 12,6,9,3 の学習済みボトルネックがあるが，本実験では圧縮率の最も高い出力チャネル数 3 のボトルネック（bottleneck3）を使用した.

４．３　評価に使用したデータセット

本研究では定点映像を対象としているため，WILDTRACK[16]という複数視点の定点映像からなるデータセットから，符号化難易度が異なるように 5 系列の動画を切り出し評価用のデータセットを作成した（図 4.3.1）．各動画は，1920×1080 pixel，20~30 秒，60fps，レート歪み曲線は図 4.3.1 に示すとおりである．WILDTRACK からは正解ラベルを得ることができなかったため，物体検出において高い精度を誇る SwinTransformer2[17]からの出力結果を疑似的な正解ラベルとして評価を行った．

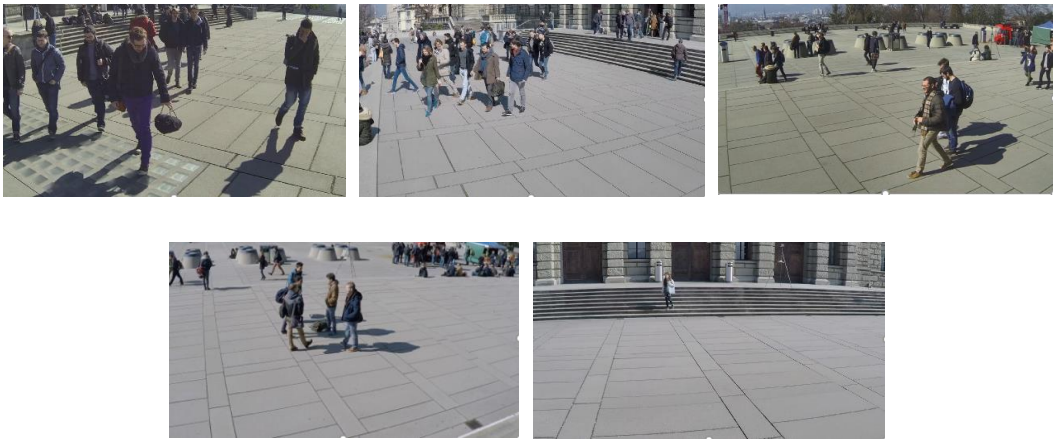

図 4.3.1 各動画の様子（左上：動画 1 ～ 右下：動画 5）

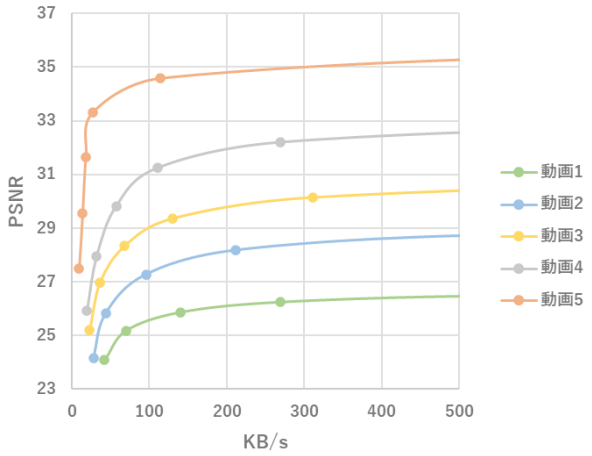

図 4.3.2 各動画のレート歪み

４．４　評価実験

評価用データセットの各動画の ffmpeg[18]を使用した HEVC 圧縮と，中間層特徴マップの HEVC 圧縮および提案手法による圧縮を，精度と圧縮率の観点で比較した．

特徴マップの HEVC 圧縮については既存手法[4]の入力を静止画から動画像に拡張したものであり，YOLOv3 では bottleneck1，Faster R-CNN では bottleneck3 と組み合わせて実験をした．具体的には，まず提案手法と同様の前処理（クリップ 1）を行った後，8bit の量子化を施す．その後，先行研究同様にタイリング（tiling）およびキルティング（quilting）で 1 フレーム上に再配置し，時間軸方向に並べたものを HEVC で圧縮した[4]．また，先行研究にならい RDOQ は使用せず、CTU（Coding Tree Unit）のサイズは 16×16 に設定した[4].

精度については, 動画内にはほとんど人物しか現れないため, person クラスの Average Precision（AP）のみを評価に使用した．また，提案手法においてはゼロラン圧縮後のエントロピーから情報量を試算し，評価を行っている．

今回，提案手法において周期 T＝ ∞で実験を行ったため，条件を揃えるために HEVC でも先頭の I フレームを除き，残りのフレームのすべてが P フレームになるように ffmpeg のオプションを設定した．また，P フレームの参照距離も 1 に設定することで提案手法同様に参照フレームをひとつ前のフレームのみに限定した．

HEVC 圧縮では映像品質のパラメタである crf[18]を変化させ，提案手法では（3.2.6）の$clip_3$ を変化させることで評価を行った．4.4.1，4.4.2 に YOLOv3 と Faster R-CNN の各動画における実験結果を示す．

４．４．１　YOLOv3

図 4.4.1.1，4.4.1.2 に YOLOv3 における平均の情報量対精度のグラフを，図 4.4.1.3 に動画 5 入力時の結果を示す．図 4.4.1.1 は入力動画の HEVC 圧縮との比較，図 4.4.1.2 は特徴マップの HEVC 圧縮との比較を行っている．また，HEVC と提案手法の各パラメタについては表 4.4.1.1，4.4.1.2 に示す．

表 4.4.1.1 HEVC における各パラメタ

圧縮手法	crf
HEVC (input)	[25,30,35,40,45,50]
bottleneck1＋HEVC (quilting, tiling)	[10,15,20,23,25,30]

表 4.4.1.2 提案手法における各パラメタ

圧縮手法	$(clip_{max}, clip_{min})$	$(clip_{2_{max}}, clip_{2_{min}})$	$clip_3$	k
提案手法のみ	(4, -1)	(1, -1)	[0.5,0.8,1.0,1.1,1.5,2.0]	14
bottleneck1＋提案手法	(3.5, -1)	(1, -1)	[0.1,0.2,0.3,0.4,0.5,0.6]	14
bottleneck2＋提案手法	(3.5, -1)	(1, -1)	[0.1,0.2,0.3,0.4,0.5,0.6]	14

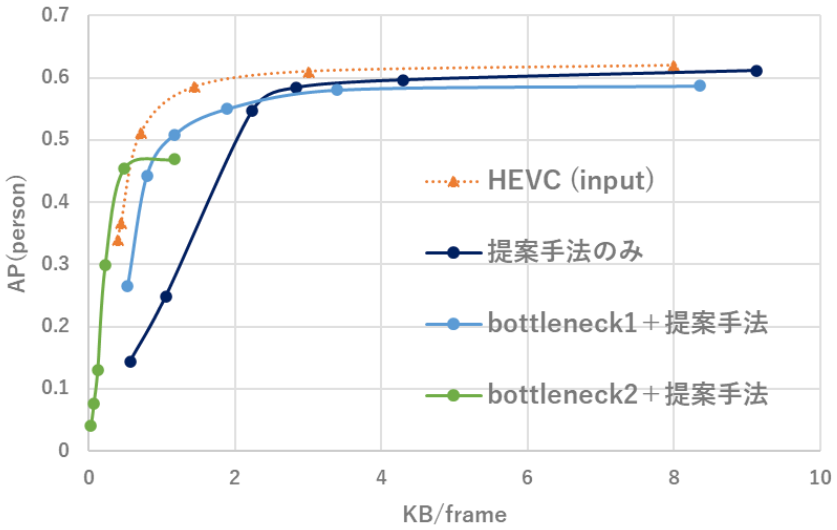

図 4.4.1.1 情報量対精度（平均）：入力動画の HEVC 圧縮との比較

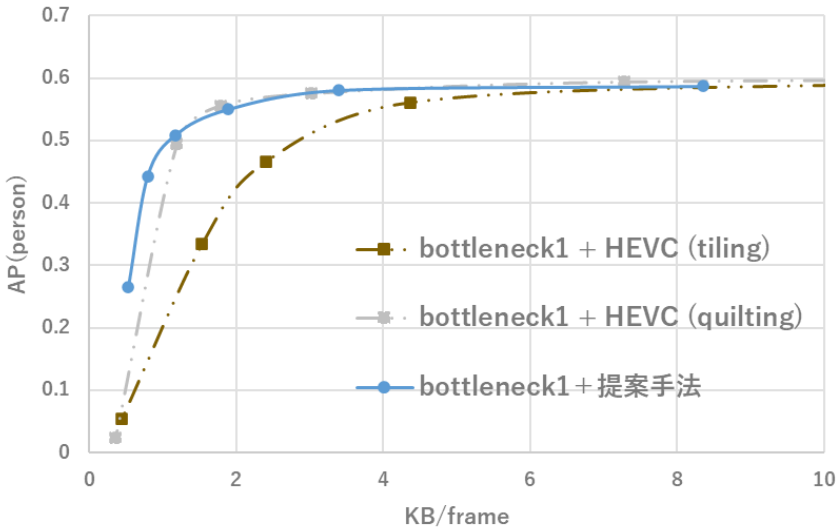

図 4.4.1.2 情報量対精度（平均）：特徴マップの HEVC 圧縮との比較

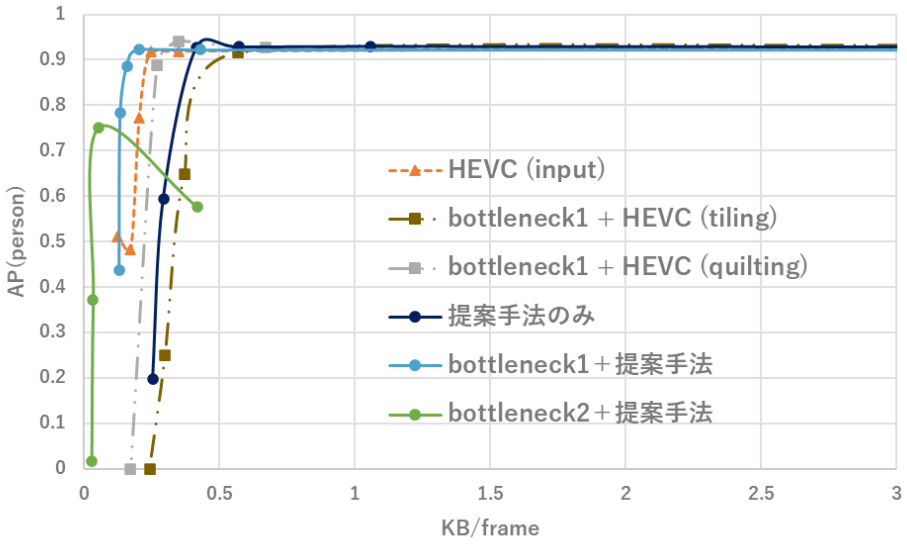

図 4.4.1.3 情報量対精度（動画 5）

４．４．２　Faster R-CNN

図 4.4.2.1，4.4.2.2 に Faster R-CNN における平均の情報量対精度のグラフを，図 4.4.2.3 に動画 5 入力時の結果を示す．図 4.4.2.1 は入力動画の HEVC 圧縮との比較，図 4.4.2.2 は特徴マップの HEVC 圧縮との比較を行っている．また，HEVC と提案手法の各パラメタについては表 4.4.2.1，4.4.2.2 に示す．

表 4.4.2.1 HEVC における各パラメタ

圧縮手法	crf
HEVC (input)	[25,30,35,40,45,50]
bottleneck3＋HEVC (quilting, tiling)	[5,10,15,20,25,30]

表 4.4.2.2 提案手法における各パラメタ

圧縮手法	$(clip_{max}, clip_{min})$	$(clip_{2_{max}}, clip_{2_{min}})$	$clip_3$	k
bottleneck3＋提案手法	(50, -50)	(10, -10)	[2.0,3.0,4.0,5.0,6.0, 7.0,8.0,9.0,10]	14

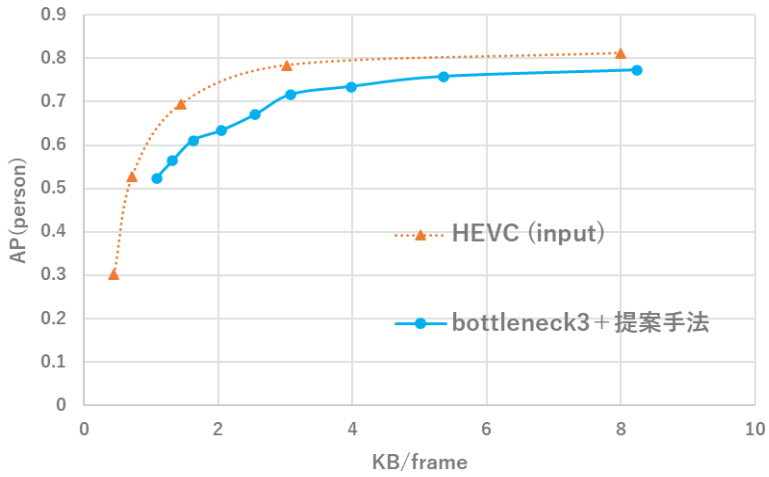

図 4.4.2.1 情報量対精度（平均）：入力動画の HEVC 圧縮との比較

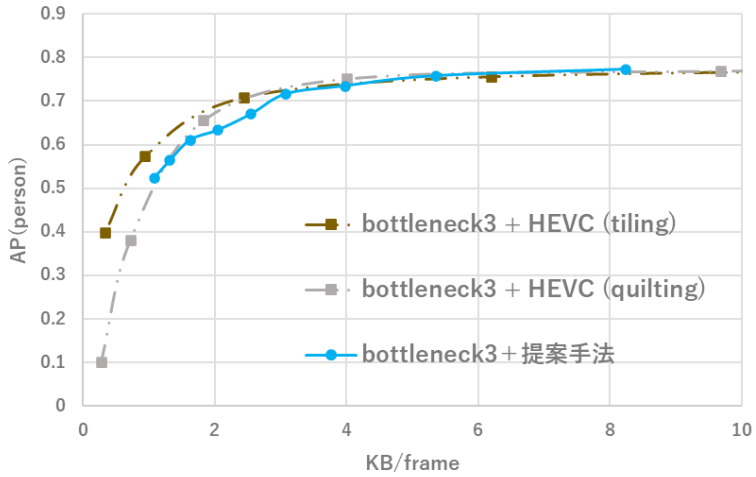

図 4.4.2.2 情報量対精度（平均）：特徴マップの HEVC 圧縮との比較

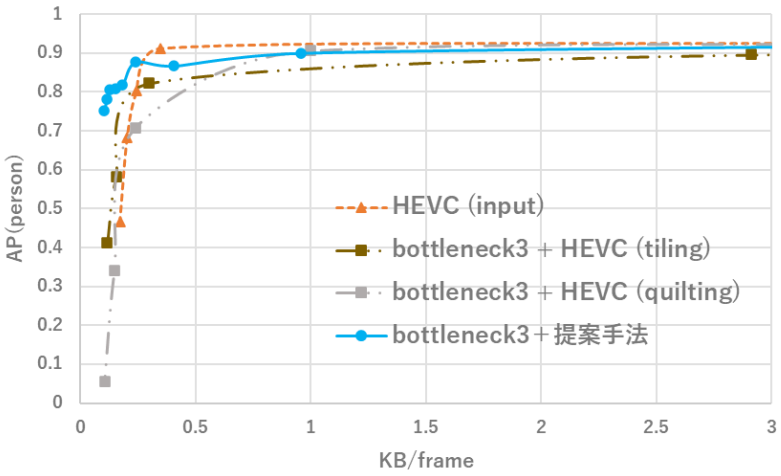

図 4.4.2.3 情報量対精度（動画 5）

４．５　考察

図 4.4.1.1，4.4.2.1 にみられるように，一部を除いて圧縮率と精度の観点で入力動画の HEVC 圧縮より良い性能を示すことはできなかった．しかしながら，映像にほとんど変化のない動画 5 での bottleneck1,3＋提案手法（図 4.4.1.3,，4.4.2.3）や，精度の大幅な低下はあるものの YOLOv3 の bottleneck2＋提案手法（図 4.4.1.1）は HEVC より同精度で高い圧縮率を示している．入力動画と異なり，中間層特徴マップは前段の処理の過程でタスクに必要な情報のみが抽出される．入力動画に対象物が少ないときや，ボトルネックからの出力サイズを小さいときは，この特性が HEVC の圧縮性能より支配的になりこのような結果が得られたと考えられる．さらに，提案手法は明らかに HEVC より単純な圧縮手法ではあるが，部分的に良い性能を示していることを考えると，中間層特徴マップの圧縮が，入力動画の符号化に勝る可能性があると期待できる．

また，既存手法の拡張である中間層特徴マップの HEVC 圧縮と比較すると，YOLOv3 ではやや良い性能を示しており（図 4.4.1.2），Faster R-CNN ではやや劣っていることがわかる（図 4.4.2.2）．これは各ネットワークの中間層特徴マップが異なる特性を有しているためであると推察する．YOLOv3 の中間層特徴マップの 1 チャネル当たりのサイズは 52×52，Faster R-CNN は 196×340 であり，切断した層の浅さからも Faster R-CNN の中間層特徴マップの方が入力画像に近く HEVC が効果的に機能したと思われる．また，YOLOv3 においてタイリングの性能が劣っているのは，HEVC のブロックサイズに対して特徴マップ 1 チャネル当たりのサイズが小さく，認識に重要な情報が削減されてしまうためであると考えられる．

４．６　むすび

本章では，評価実験の設定および結果について述べた．評価実験では，物体検出における精度と圧縮率という観点で，提案手法による中間層特徴マップ圧縮，入力動画の HEVC 圧縮，中間層特徴マップの HEVC 圧縮（既存手法の拡張）の三つを比較した．その結果，提案手法は限られた条件下でのみ入力動画の HEVC 圧縮に対して優位性を示すことがわかった．また，中間層特徴マップの HEVC 圧縮に対しては使用するネットワークによって異なる結果となった．

第５章　結論と今後の課題

５．１　結論

本研究では，Collaborative Intelligence における動画像（定点映像）入力時の中間層特徴マップの圧縮手法を提案した．既存の研究では入力を静止画像に限定していたため，動画像の中間層特徴マップ圧縮は初めての試みであると考えられる．提案手法では，時間方向の相関除去のために前フレームとの差分生成，二段階のクリップ処理，量子化，チャネル方向のゼロラン圧縮という流れで中間層特徴マップの圧縮を図っている．

実験では，符号化難易度の異なる 5 系列の動画を用意し，YOLOv3，Faster R-CNN を用いて物体検出を行った．入力動画の HEVC 圧縮，および中間層特徴マップの HEVC 圧縮（既存手法の拡張）と情報量対精度の観点で比較し，評価を行った．その結果，動画内の変化がほとんどないときや，狭帯域下では入力動画の HEVC 圧縮より良い性能を示すことが確認できた. また, 中間層特徴マップの HEVC 圧縮との比較では, YOLOv3 においてはやや良い性能を示し，Faster R-CNN においてはやや劣る結果となった．

５．２　今後の課題

圧縮率と精度の観点において, 大幅な精度低下を許したときと映像にほとんど変化がないときを除いて，提案手法の有効性が示すことができなかった．これは実際のユースケースにはあまり適さないため，圧縮率および精度の向上に努めたい．提案手法において，隣接フレームの差分という方法でのみ時間方向の相関を利用しているので，参照フレーム数の増加や動き補償予測などの利用, HEVC との計算量の比較も今後検討したい. また，提案手法内のクリップ 3＋量子化を非線形量子化に置き換えることでより詳細に圧縮率を操作できると考えられるため，こちらも今後検討したい．

謝辞

本研究に際して，丁寧かつ素晴らしいご指導をしてくださり，実験環境および快適な研究環境を与えてくださった渡辺教授に心より感謝いたします.

共同で研究をさせてくださり，多くの知識や示唆をしてくださった，NTT 株式会社の江田毅晴様，榎本昇平様，坂本啓様，史旭様，森永一路様に感謝いたします.

日頃から貴重な意見をくださり，研究室における温かい環境を提供してくださった渡辺研究室の皆様に感謝いたします.

最後に，私をここまで育ててくださり，常に心を支えてくださり，生活を支えてくださっている家族に感謝いたします.

参考文献

[1] Y. Matsubara, M. Levorato, and F. Restuccia: “Split Computing and Early Exiting for Deep Learning Applications: Survey and Research Challenges,” arXiv preprint arXiv:2103.04505, Nov. 2021.

[2] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang: “Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge,” ASPLOS '17: Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 615-629, Apr. 2017.

[3] S. Teerapittayanon, B. McDanel, and H.-T. Kung: “BranchyNet: Fast Inference via Early Exiting from Deep Neural Networks,” International Conference on Pattern Recognition (ICPR), pp. 2464-2469, Dec. 2016.

[4] H. Choi and I. V. Bajić: “Deep Feature Compression for Collaborative Object Detection,” IEEE International Conference on Image Processing (ICIP), pp. 3743–3747, Oct. 2018.

[5] A. E. Eshratifar, A. Esmaili, and M. Pedram: “BottleNet: A Deep Learning Architecture for Intelligent Mobile Cloud Computing Services,” IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), pp. 1-6, Jul. 2019.

[6] Y. Matsubara, D. Callegaro, S. Baidya, M. Levorato, and S. Singh: “Head Network Distillation: Splitting Distilled Deep Neural Networks for Resource-Constrained Edge Computing Systems,” IEEE Access, vol. 8, pp. 212177-212193, Nov. 2020.

[7] R. A. Cohen, H. Choi, and I. V. Bajić: “Lightweight Compression Of Neural Network Feature Tensors For Collaborative Intelligence,” IEEE Open Journal of Circuits and Systems, vol. 2, pp. 350-362, May 2021.

[8] A. E. Eshratifar, M. S. Abrishami, and M. Pedram: “JointDNN: an efficient training and inference engine for intelligent mobile cloud computing services,” IEEE Transactions on Mobile Computing, vol. 20, no. 2, pp. 565-576, Feb. 2021.

[9] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand: “Overview of the high efficiency video coding (HEVC) standard,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.

[10] J. Redmon and A. Farhadi: “YOLOv3: An incremental improvement,” arXiv preprint arXiv:1804.02767, Apr. 2018.

[11] S. Ren, K. He, R. Girshick, and J. Sun: “Faster R-CNN: Towards realtime object detection with region proposal networks,” Advances in neural information processing systems 28 (NIPS 2015), pp. 91–99, Dec. 2015.

[12] Y. Matsubara and M. Levorato: “Neural Compression and Filtering for Edge-assisted Real-time Object Detection in Challenged Networks,” International Conference on Pattern Recognition (ICPR), pp. 2272-2279, Jan. 2021.

[13] https://github.com/yoshitomo-matsubara/hnd-ghnd-object-detectors

[14] N. Samet, S. Hicsonmez, and E. Akbas: “HoughNet: Integrating near and long-range evidence for bottom-up object detection,” ECCV 2020, pp.406-423, Jul. 2020.

[15] K. He, X. Zhang, S. Ren and J. Sun: “Deep Residual Learning for Image Recognit ion,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7 70-778, June 2016.

[16] T. Chavdarova et al.: “WILDTRACK: A Multi-camera HD Dataset for Dense Unscr ipted Pedestrian Detection,” IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5030-5039, June 2018.

[17] Z. Liu et al.: “Swin Transformer V2: Scaling Up Capacity and Resolution,” arXiv p reprint arXiv:2111.09883, Nov. 2021.

[18] Suramya Tomar: “Converting video formats with FFmpeg,” Linux Journal, 2006(146), p.10, June 20016. available from https://www.ffmpeg.org/

図一覧

表一覧