修士論文

共有 CNN を利用した高効率な推論向けマルチタスクニューラルネットワークの分割手法と評価

平賀 由利亜

2019 年 1 月 31 日

奈良先端科学技術大学院大学
情報科学研究科

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士 (工学) 授与の要件として提出した修士論文である。

平賀 由利亜

審査委員：

中島 康彦 教授	（主指導教員）
井上 美智子 教授	（副指導教員）
中田 尚 准教授	（副指導教員）
Tran Thi Hong 助教	（副指導教員）
張任遠 助教	（副指導教員）

共有 CNN を利用した高効率な推論向けマルチタスクニューラルネットワークの分割手法と評価*

平賀 由利亜

内容梗概

実用的な深層学習アプリケーションとして、一つの IoT デバイスで取得したセンサーデータをネットワークを経由して複数のサーバーに転送し、それぞれ別のタスクの深層学習の推論を行うモデルを考える。画像のようなデータサイズの大きいセンサーデータを送る場合、転送先の数だけセンサーデータを送る必要があり、ネットワークの輻輳問題が発生するため、行うタスク数に限りが出る。

本稿では深層学習アプリケーションにおけるネットワークの輻輳問題とクラウドへの計算負荷の集中を軽減するため、マルチタスクにおける共有 CNN を用いた深層学習の推論の分割と中間データを圧縮する手法を提案し、提案するモデルにおける RGB 画像に共通の前処理をしたものを入力とした共有可能なタスクと全体計算量、転送データサイズの削減割合について調査を行う。ILSVRC2012 の物体認識タスクと PASCAL VOC の物体検出タスクに対して中間データの圧縮に BPG 圧縮を利用した場合、共有 CNN に VGG16 と MobileNetV2 を使用すると物体認識タスクでは圧縮前のモデルの精度と比較すると Top-1 Accuracy について VGG16 は 2.11%、MobileNetV2 は 2.59%の精度の劣化を示し、物体検出タスクでは SSD300 と比較をすると mAP について VGG16 は 19.93%、MobileNetV2 は 14.91%の精度を劣化を示した。同時に、VGG16 を使用した場合に全体計算量を 42.30%、データ転送量の 36.19%の削減を行った。MobileNetV2 では全体計算量を 9.77%削減し、データ転送量の 68.70%の削減を行った。

*奈良先端科学技術大学院大学 情報科学研究科 修士論文, 2019 年 1 月 31 日.

キーワード

深層ニューラルネットワーク、エッジコンピューティング、共通ニューラルネットワーク、マルチタスクニューラルネットワーク

Divided method and its evaluation of efficient multi-task neural network for inference using shared CNN*

Yuria Hiraga

Abstract

A DNN application has network connections between an IoT device and the third party instances of the different clouds where they take in sensor data from the IoT device to run inference for each task. The IoT device needs to send sensor data to the clouds as much as the number of tasks. It causes trafiic congestion unless the application limits the number of forwarding clouds. I present divided multi-task deep neural network using shared CNN between the edge and the cloud to reduce traffic congestion and concentration of the cloud; preprocess for RGB image as input data is the same procedure as when the shared CNN is trained; and transfer data is compressed by lossy compression algorithms. I examine sharable tasks, the ratio of bit rate reduction and MAC operations on proposed network structure. In my experimental results I use the dataset of ILSVRC2012 for object recognition and PASCAL VOC2007 for object detection, and lossy compression algorithm is BPG. For 224×224 input, while proposed network using VGG16 as a shared CNN achieves 2.11% top-1 accuracy decrease on ILSVRC2012 compared to the trained model's one and 19.93% mAP decrease on PASCAL VOC2007 compared to SSD300 on original paper, the ratio of MAC operation reduction is 42.3%; the ratio of transfer data size reduction is 36.19%. For 224×224 input,

*Master's Thesis, Graduate School of Information Science, Nara Institute of Science and Technology, January 31, 2019.

while proposed network using MobileNetV2 as a shared CNN achieves 2.59% top-1 accuracy decrease on ILSVRC2012 compared to trained model's one and 14.91% mAP decrease on PASCAL VOC2007 compared to SSD300 on original paper, the ratio of MAC operation reduction is 9.77%; the ratio of transfer data size reduction is 68.7%.

Keywords:

Deep Neural Network, Edge-computing, Inference, Shared-CNN, Multi-task neural network

目 次

図 目 次

表 目 次

1. はじめに

インターネットの普及や計算機、センサーテクノロジーの発展により、様々なサービスが我々の生活に浸透している。特にインターネットにつながるモノであるIoT(Internet of Things) デバイスの数は増え続け、パソコンやスマートフォンなどの従来のインターネット接続端末に加え、車や工場製品などの世界中の様々なモノのIoT 化が拡大しており、2016 年時点の 173 億個から 2020 年には 300 億個まで増加することが予想される [33]。IoT デバイスを利用する用途には、デバイス側で取得したデータを演算処理した結果を用いた様々なシステムへの応用が考えられており、IoT デバイスを利用したシステムでは、1. IoT デバイスに接続されたセンサーから取得した入力情報を圧縮し、2. ネットワーク通信路、基地局を介してサーバーへ転送、3. サーバーは入力情報を展開、4. 演算処理より結果を得る、という手順で処理が実行される。サーバーがクラウドに置いてあると仮定すると、増え続けるIoT デバイスからのデータ転送に対してエッジデバイス-クラウド間の通信網が一極集中であるため、トラフィック輻輳が発生する可能性がある。解決策の一つがエッジコンピューティングである。エッジコンピューティングとは端末の近くにサーバーを分散配置するネットワーク技法であり、図 1 のようにユーザーや端末の近くでデータ処理を行うことで上位システムへの演算やネットワークへの負荷を軽減を解消する手法である。

深層学習のような計算量の多い処理は計算資源の豊富なクラウド側で行われることが多かったが、末端デバイスへの搭載を対象とする専用のアクセラレータ [7, 19, 34] も研究されている。末端デバイスでの処理を意識した研究では、推論時のデータ単位の変更や末端デバイスでの処理向けのモデルが提案されている。推論時にデータ単位を変更する手法では、推論であればデータの取りうる表現の幅を落としても精度の低下は小さいため、深層学習の処理で使用されることの多い 32bit の浮動小数点型を固定小数点に変更する手法 [16] や 2 値論理を用いる手法 [21] が提案されている。末端デバイスでの処理向けのモデルの提案では、モデル内の層の数や層ごとのチャネル数を少なく設計することで必要なメモリや計算量を抑えることができる。これらの取り組みにより、末端端末での深層学習の推論を用いたエッジコンピューティングは不可能ではない。

私は実用的な深層学習アプリケーションとして、一つのエッジデバイスで取得したセンサーデータをネットワークを経由して複数のクラウドに転送し、それぞれ別のタスクの深層学習の推論を行うモデルを考える。画像のようなデータサイズの大きいセンサーデータを送る場合、転送先の数だけセンサーデータを送る必要があり、ネットワークの輻輳問題や行うタスク数に限りが出る。ネットワークの輻輳問題に対して、エッジデバイス側で深層学習の一部の処理を行い、出力される特徴量が元のデータサイズよりも小さくなった段階で転送をする手法が考えられる。しかし、エッジデバイスの計算資源では複数のタスクを同時に処理することは難しい。別のアプローチとして、Convolutional Neural Network(CNN) を共有する共有 CNN を用いたマルチタスクの深層学習モデルが考えられる。マルチタスク学習とは異なるタスクをモデルを共有し学習させることによって、タスク単体のデータセットを用いて学習するよりも精度を向上させる手法である。深層学習では、前半層は汎用的な特徴を学習し、後半層はタスクに特化した特徴を学習していると考えている [30]。深層学習を用いたマルチタスク学習では、深層

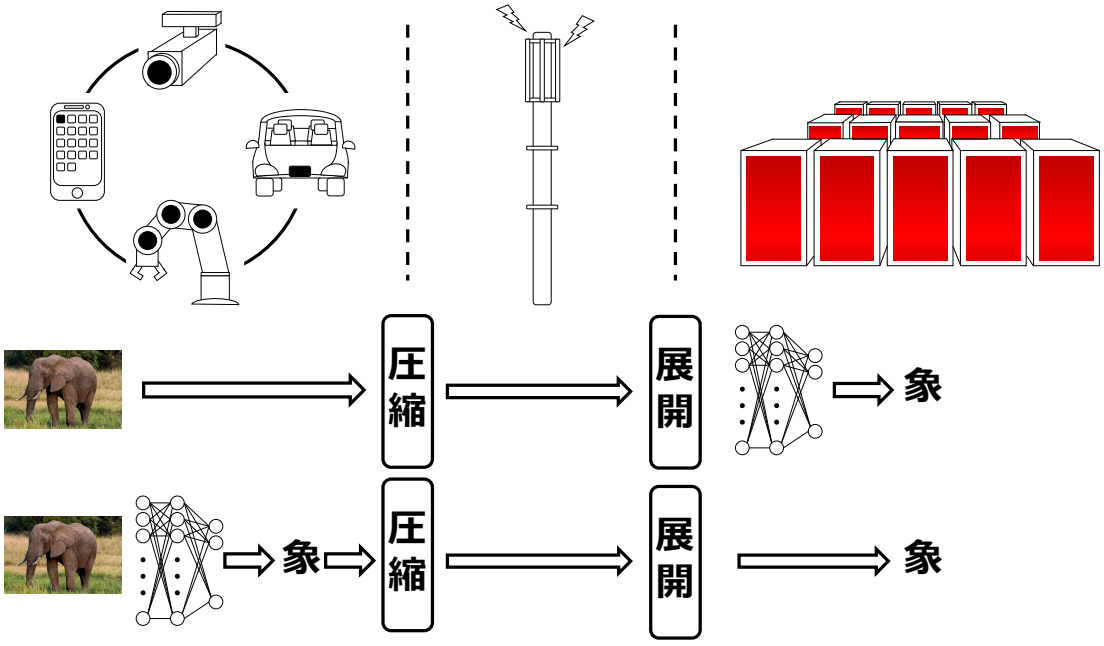

図 1: エッジコンピューティング

学習モデルの前半層をタスク間で共有し、後半層はタスクごとに分岐するモデル構成となっている。マルチタスク学習のモデル構成を応用することで、タスクあたりの計算量を削減し、共有層の処理をエッジデバイス側で行うことができる。しかし、現実のアプリケーションとして不特定多数のユーザーがタスクを追加、取り除くことができる場合に、タスクを追加する度にモデルの再学習を行う必要があるため、再学習の計算コストがかかる。また、特定のタスクに最適化されたマルチタスク学習ではないため、再学習ごとのモデルの精度を保証することができないため、常に精度を保証する必要があるアプリケーションには使うことができない。本稿では深層学習アプリケーションにおけるネットワークの輻輳問題とクラウドへの計算負荷の集中を軽減するため、マルチタスクにおける共有 CNN を用いた深層学習の推論の分割を提案し、提案するモデルにおける RGB 画像を入力とした場合の共有可能なタスクと、中間データの転送を行う際の圧縮手法について調査を行う。提案する分散深層学習モデルは、あらかじめ学習を行いパラメータを固定した単一の CNN を共有し、共有 CNN を用いてタスクごとに再学習を行った複数の深層学習モデルから構成されている。エッジデバイス、または基地局で共有 CNN による推論を行い、元画像よりも要素数が減少した段階で残りの処理を行うクラウドに中間データを圧縮し、転送することで、アプリケーション全体のネットワークトラフィック量を削減する。圧縮手法にはサーバー・クライアントモデルで既に圧縮、展開の取り組みが行われている動画像の可逆・非可逆圧縮を利用する。

2. 諸定義

2.1 Neural Network

ニューラルネットワークとは、生体システムにおける情報処理を数学的に表現しようとする試みにその起源がある [32]。式 1 で表される 1 層のニューラルネットワークは入力 x と重みパラメーター w との線形和を非線形活性化関数 σ への入力とした処理を 1 層とし、深層学習では FCL（Fully Connected Layer）と称される。多層にしたニューラルネットワークの学習では勾配消失により、学習を促進させることは困難であったため、浅い層のモデルが提案されることが多く、今日の深層学習のように層の深いネットワークである Deep Neural Network(DNN) のように精度向上を図ることが難しい。

$$y_j(\mathbf{x}, \mathbf{w}) = \sigma\left(\sum_{\mathbf{i=0}}^{\mathbf{N}} \mathbf{w_{ji}x}\right) \tag{1}$$

2.1.1 Deep Neural Network

深層ニューラルネットワークが注目されるようになった契機は画像認識の大規模競技会である ILSVRC[25] での目覚しい認識精度向上を示した結果によるところが大きい。2012 年の ILSVRC では、5 つの CNN(Convolutional Neural Network) と 3 つの FCL を組み合わせた全 8 層の深層学習モデルである AlexNet[14] が提案された。

2.1.2 Convolutional Neural Network

画像を入力とする DNN(Deep Neural Network) では、認識を役割をもつ全結合層への入力は画像の特徴量になる。そのため、その前段には特徴量抽出を行うニューラルネットワークが設置される。その画像の特徴量を抽出するニューラルネットワークが畳み込みニューラルネットワーク、すなわち CNN と称する。図 3 に示すように、CNN は入力が与えられた時、各畳み込み層が持つ重みパラメータと] 入力からある一定範囲、すなわち図中のカーネルと示された枠から取り出し

た値それぞれの掛け算の総和を求める演算、すなわち畳み込み演算より 1 つの出力が求められ、この畳込み演算を入力全てに対して適用させることにより、畳み込み層が出力する特徴量を得る。出力された特徴量は数層に重ねられた畳み込み層を通過することより、画像からより空間的な特徴が抽出される。しかし、特徴量の中には情報の少ない値もあるため、全ての特徴量を伝搬させると、その全てに対して畳み込み演算やパターン認識が行われる。しかし、小さな値を持つ特徴量は後続の層の出力へ与える影響は少なく、全ての特徴量に対して演算を行うことは非効率であるため、出力された特徴量から代表値だけを伝搬させる役割を持

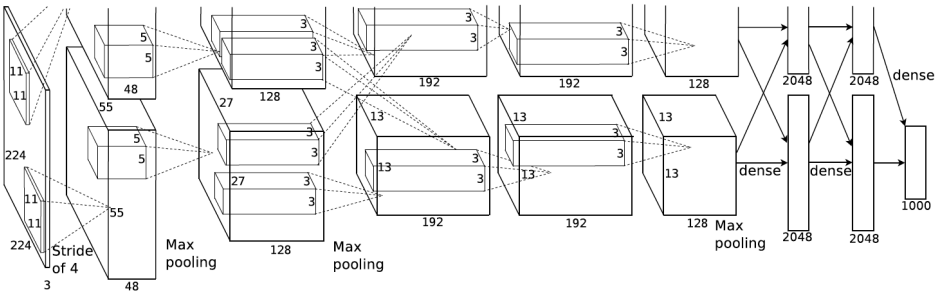

図 2: AlexNet

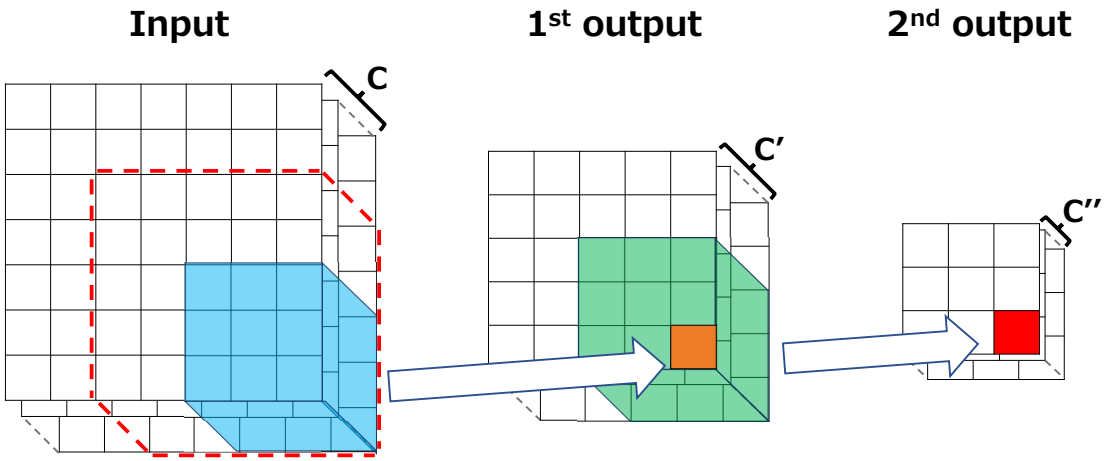

図 3: 2 層の畳み込みニューラルネットワーク

つプーリング層が CNN の間にいくつか挿入されてる。このプーリング層により、ある一定サイズのフィルターを適用することにより代表値だけを伝搬できるため、後続のニューラルネットワークへの入力を小さくできる。CNN の計算量は、重みパラメータのサイズや入力画像、畳み込み演算層の入力範囲であるカーネルの大きさより決定される。重みパラメータの数が増えることで、重みパラメータのデータサイズも大きくなるが何よりも計算量の増大が大きい。

2.1.3 Pooling

プーリング層とは、層間で出力される特徴量マップのそれぞれの値の周辺情報を集約する処理である。Zhou らの研究 [31] で提唱された最大プーリング (Max Pooling) は特徴量マップに対して矩形カーネル内の最大値のみを抽出する処理を示す。図 4 の 2×2 の二次元のカーネルを使った例では、二次元の特徴量マップの要素数を 4 分の 1 に削減する。

2.2 マルチタスク学習

深層学習では汎用的なパラメーターを学習するために十分なデータを用意する必要があるが、行うタスクに対して十分なデータを用意できない場合に、学習用

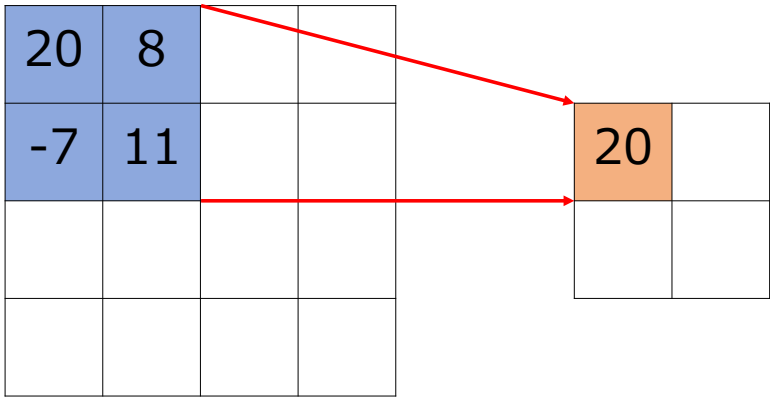

図 4: 2×2 カーネルの Max Pooling

データの水増しを行い、認識精度の向上を図る。これをデータ拡張という。データ拡張のアイデアは深層学習以前から検討されており、ニューラルネットワークでは複数のタスクを同時に学習するマルチタスク学習が提案されている。マルチタスク学習では共通の中間層を用いて多クラス分類を行い、単一クラスを対象とするときよりも大きいデータセットを用いて学習することを可能とした。クラス間の共通した特徴量を学習することによって単一クラスの学習をそれぞれ行うよりも良い識別精度を示している [6]。S. Ruder が行なったマルチタスク学習についてのサーベイ [24] では、Caruana の研究 [6] も考慮し、マルチタスク学習のメリットについて以下の 5 つの項目について言及されている。

Implicit data augmentation　マルチタスク学習においてタスク A、タスク B に共通する特徴量 F があるとすると、特徴量 F はタスク A とタスク B のデータセットを合わせた学習を行うことができるため、データ拡張を行なっていると見なせる。

Attention focusing　データ数が少ない場合に、共通する特徴量を発見することが困難になるが、複数のタスクの学習によってデータを水増しすることで、タスクのデータが少ない場合でも、共通する特徴量を発見し、学習することができる。

Eavesdropping　タスク A のデータよりもタスク B のデータに共通する特徴量 F が含まれている場合、タスク A 単体の学習よりもより良い学習ができることを示す。

Representation Bias　タスク A の局所最適解とタスク B の局所最適解があった場合に共通しない局所最適解は学習されない。

Regularization　学習時のオーバーフィッティングを抑制することで、ラデマッハ複雑度で示されるモデルの複雑さによるリスクを削減し、モデルを汎化させるように学習を行う。

2.3 転移学習・ファインチューニング

深層学習の学習では、学習を始める際の初期値によって学習の進み具合に差が出ることが知られている。教師なし学習を用いた手法では、Autoencoder[20] が使われていた。教師あり学習を用いた手法に転移学習とファインチューニングがある。転移学習 [30] とは対象のデータセット以外のデータセットを用いてモデルパラメーターの事前学習を行い、学習されたモデルのパラメーターを対象のモデルのパラメーターの初期値とし、学習はせずに固定する。パラメーターの転移を行なっていない後半層については適当な初期化を行い、学習を行う。一方で、fine-tuning[30] とは転移学習時に固定されていたパラメータを対象のタスクに対して再学習を行うことを示す。

転移学習やファインチューニングは、学習を早めるパラメーターの初期値の決定を目的とするだけでなく、学習対象の十分なデータが得られない場合にも精度向上の手法として用いられる。J. Yosinski らの研究 [30] では、深層学習では前半層は汎用的であり、後半層はドメイン固有のものであると見なしている。ドメインとは学習するデータの分布の領域を示し、共通のドメインのデータセットを用いて事前学習を行うことでタスクに対するデータセット単体で学習するよりも良い精度を得ることができる [5, 18]。あらかじめ定義された深層学習モデルを異なるデータセットを用いて事前学習し、前半層のパラメータを初期値とし、後半層を初期化させた上で転移学習または fine-tuning[29] を行なっている。単純な学習よりも事前学習と転移学習を行なった場合の実験結果のほうが良かったため、深層学習でも異なるドメイン間の共通部分の学習を行っていると考える。

2.4 深層学習を用いた画像を入力とするタスク

2.4.1 物体認識

深層学習の有用性を広く示す契機となったのは、ILSVRC という画像内の物体認識を対象としたアルゴリズムの大規模競技会である。2012 年に Hinton らによって提案された AlexNet を始めとして、毎年最も良い精度を示した深層学習モデルや学習手法をもとに物体認識やその他の分野への深層学習の応用が研究されてい

る。物体認識のタスクでは一枚の画像あたり、一つのクラス分類を行う。ILSVRC の 1000 クラス分類のようにクラス数が多く、クラス間に類似度の高い画像が多く含まれるデータセットを利用する場合には、分類結果の上位 5 クラス内に正解データが含まれる精度も評価する。

2.4.2 物体検出

物体検出とは、一枚の画像あたり分類するクラスに属する物体と物体を含む領域を選択するタスクである。精度の計算方法として mean Average Precision(mAP) で評価をしている。代表的な物体検出の深層学習モデルに Single Shot Multibox Detector(SSD)[17] を挙げる。SSD の feature map は Conv4_3、Conv7、Conv8_2、Conv9_2、Conv10_2、Conv11_2 のそれぞれの出力ごとに生成され、各層から伝搬させる feature map ごとにあらかじめ設定した Default Bounding Box の検出結果から閾値を越えたバウンディングボックス同士を結合し、結合したバウンディングボックスごとに物体の領域とクラスを決定する。

3. 関連研究

3.1 推論の分割に関する研究

エッジコンピューティングを利用した深層学習の推論の分割では、全体の処理を前半と後半に分割する。処理の前半部をエッジデバイス上で実行し、実行結果である中間データをサーバーへ転送、後半の処理を行う手法を示す。Y. Kang[12] らの研究では 3 種類のタスク、8 つのモデルを対象に、各層で分割したときのレイテンシとエッジデバイスでの処理、データ転送、クラウドでの処理で消費したエネルギーについて計測を行い、最も良い分割点について考察がされている。研究内で使用されているエッジデバイスとクラウドの構成では、画像認識のタスクに対する VGG16[27] を利用した推論において、レイテンシでは pool5 での分割点が最も良く、エネルギー消費量ではクラウド側で全ての処理を行うことが最も良いという結果が出ている。使用する機器、タスクやモデルによって結果は異なるが、使用環境による最適な性能を引き出す手段の一つに推論の分割という選択肢がある。現実的なアプリケーションとして、推論の分割でも転送するデータサイズを考慮すれば、中間データの転送時には圧縮を行うことが考えられる。動画像を利用したサーバークライアントモデルでは、エッジデバイスで動画像を取得して圧縮を行う。その後、クラウドへ転送を行い、圧縮されたデータを解凍し、処理を行う。H. Choi らの研究 [8] では、VGG16 を用いた物体認識、YOLOv2[22] を用いた物体検出に対して推論をエッジデバイス側、クラウド側の二箇所に分散する。エッジデバイス側で生成された中間データをクラウドに転送する際に動画圧縮手法である H265/HEVC[28] を用いた圧縮を行なっている。

これらの手法では入力に使用する画像とタスクが一対一であることが前提である。圧縮条件や学習条件をタスクごとに調節することができる一方で、エッジデバイスで取得したデータを複数のタスクに適用することは考慮されておらず、複数のタスクをこれらの手法で実現するためには、計算資源の乏しいエッジデバイス側で並列に推論を実行するか、順番に推論を繰り返す必要がある。タスク数が多い場合や、リアルタイム性を求めるアプリケーションが多い場合に、アプリケーションの要求を満たすことが難しくなる。また、不特定多数の第三者に配信しそ

れぞれのタスクに対して推論をするようなことは想定されていないため、推論の詳細が明かされていない場合にこれらの手法を採用することはできない。

3.2 パラメーター共有に関する研究

深層学習のマルチタスク学習ではハードパラメーターシェアリングとソフトパラメーターシェアリングの二種類の学習手法がある [24]。ハードパラメーターシェアリングとは、タスク間で前半層を共有し、後半の層はそれぞれのタスクに特化したモデルを用意した状態で学習を行う手法を示す。ソフトパラメーターシェアリングとは、タスク間で層の共有はせず、層の構造のみ共通化する。それぞれのタスクを学習する際に、タスク間の対象の層を層間の L2 距離やノルムを利用し、正規化することによって汎化性能を高める手法を示す。本研究では、マルチタスク学習時の共有層の構造を効率的な推論に応用することを目的にする。また、共有層の学習には単一のデータセットを利用するため、関連研究ではハードパラメーターシェアリングを重視する。

ILSVRC2012 で深層学習が画像認識の精度についてブレイクスルーを起こす前の画像認識のフローでは、画像から特徴量抽出を行い、特徴量を分類器へ入力し分類を行う。画像を入力とする深層学習のタスクでは、ILSVRC の競技会で最も良いモデルをベースに改良されることが多く、処理の組み合わせとして CNN+FC の組み合わせのモデルを使用することが多い。深層学習での転移学習、ファインチューニングという手法では、対象のタスクを学習するためのデータセットを十分に用意できない場合に、他のタスクの十分な量のデータセットで学習されたモデルの前半部を初期値として使用し、後半部のみ学習させるといった手法である。十分な量のデータセットで学習されたモデルにはタスクに関わらない特徴量抽出器としての機能があると考えられる。しかし、特徴量抽出器として深層学習モデルを利用する場合、モデル内の処理に注意しなければならない。深層学習の性能の高さを支えているのは活性化関数によるところが大きい。特に ReLU[9] と呼ばれる、0 以下の値を 0 に変換する活性化関数を取り入れた深層学習モデルは ILSVRC の競技会において認識精度向上に寄与しているだけでなく、ニューラルネットワークの学習において学習するパラメーターの初期値によって学習具合が左右される

問題に対して、設定する初期値によらず最適解に収束することを示す研究 [4] もされている。ReLU を組み込んだ深層学習モデルの場合、順伝搬時に 0 以下の数値に対して情報の欠落が発生する。深層学習の判断根拠を示す研究 [23] では、推論結果から判断材料となった元画像の領域を可視化している。モデルの出力層に近い層では、判断基準にならない部分の情報が欠落していることを示している。特徴量抽出を行うにあたり、深層学習モデルのアーキテクチャや学習方法によって必要な情報を抽出するように工夫しなければならない。S. Kornblith らの研究 [13] では、ILSVRC のような学習が困難な大規模データセットで良い結果を出したモデルは、ハードパラメーターシェアリングによって他のタスクに転移可能な特徴量抽出器として機能することを仮定している。研究では、ILSVRC2012 で使用された IMAGENET のデータセットを用いて学習をした 13 個の深層学習モデルを特徴量抽出器として利用し、得られた特徴量をロジスティック回帰への入力として、検証のために用意した 12 個のタスクについて学習を行なっている。この実験では、ILSVRC2015 で最も良い精度を出した Resnet-152[10] が最も良い結果を出している。また、異なる種類のタスクに対して汎用的なモデルの提案もされている。MobileNet の研究 [11][26] では、スマートフォンのような末端端末でも効率良く動作するようにモデルに必要なパラメーター数や演算回数を減らす試みがされている。MobileNetV2 の研究 [26] では、高効率な演算だけでなく、深層学習で後段の層に伝搬する度に情報の損失が起こる問題に対して反転残差構造を挿入することによって活性化関数による情報の損失を軽減させている。順伝搬時の情報の損失を軽減させることで、画像認識、物体検出、セマンティックセグメンテーションの 3 つのタスクに対してパラメーターが十分に持つモデルと比べて遜色ない結果を示している。

しかしながら、深層学習モデルを特徴量抽出器として利用した実験では単にタスクごとに転移学習を行なったのみであり、特徴量抽出器として利用する共有 CNN の構成について画像認識以外のタスクへの影響は考慮されていない。また、推論の分割を目的としているわけではないため、共有 CNN と後段のタスク別深層学習モデルのデータの受け渡しに圧縮処理等は行われていない。MobileNetV2 の研究では、同一のモデル構造で物体認識、物体検出、領域検出の複数タスクに

対応することを検討しているが、入力画像の大きさが異なることや、物体検出では feature map の転送も行われるため、共有 CNN を用いた推論の分割に直接利用することはできない。

4. 提案手法

本章では、エッジコンピューティングと共有 CNN を利用したマルチタスク深層学習モデルについて共有可能なタスクと精度について調査を行う。

4.1 一対多のデータフローにおける共有 CNN を利用したマルチタスク DNN 向け推論の分割と圧縮手法の概要

入力データに対して複数のタスクを処理する場合、図 5 のようにタスク数分のモデルを動作させることが考えられる。IoT デバイスとクラウドを利用したサーバークライアントモデルを考えた場合に、IoT デバイスで取得したデータをクラウドに転送し、計算リソースが潤沢にある環境で処理が行われる。一方で、エッジコンピューティングにより全ての処理をエッジ側で処理するためには、計算リソースとタスクの処理に必要な深層学習モデルの大きさを考慮する必要がある。リアルタイムに動作させる必要のあるタスクがある場合には、必要なタスク数分のモデルを並列で動作させる必要があるが、計算リソースの乏しいエッジ側では並列で動作させることのできるタスク数には限りがあるため、並列実行可能なタスク数に制限が出る。計算リソースに対して深層学習モデルの大きさを削減する試みには、深層学習モデルの蒸留や使用するデータ単位の縮小が研究されている。しかしながら、入力データに対して並列に動作させる必要のあるタスクが増えるに従い、要求される計算リソースも増え続けるため解決策にはならない。

これらの問題に対して、ハードパラメーターシェアリングを利用したマルチタスク学習のモデル構造を推論に応用し、エッジークラウド間で計算を分散することによってエッジコンピューティングにおける計算リソースが不足する問題に対処することができる。共有可能なタスクについて以下の点を重視する。

1. 共有 CNN に使用するモデルとその学習手法
2. 共有 CNN を使用した際のタスク別モデルの拡張手法
3. 共有 CNN の使用による転送データ容量とモデル全体の演算処理の削減割合

本章では、エッジークラウドを協調した複数タスクに対するニューラルネットワークモデルの推論実行を想定し、マルチタスク DNN 向け推論の分割を提案する。従来における DNN の推論は、以下の手順で処理される。

1. 末端デバイスは接続されたセンサからデータを取得し、圧縮する
2. 圧縮済みデータはネットワーク通信路を介しサーバへ転送される
3. サーバは受け取ったデータを展開後、DNN の演算を行う
4. 処理結果だけを末端デバイスへ転送する

この一連の推論では、末端デバイスの演算は圧縮のみであり、DNN の推論はサーバ側で行っている。そのため、サーバへの計算負荷が集中し、サーバへ至る通信路の輻輳も問題となっている [3]。提案する実行モデルは画像を入力とする、タスク別の DNN を分割実装し推論を行う。提案する実行モデルの概略を図 6 に示す。提案するモデルでは *Hard-shared* までの処理をエッジで行い、圧縮しクラウドへと転送する。転送先の *A*、*B*、*C* では転送されたデータを展開し、それぞれ

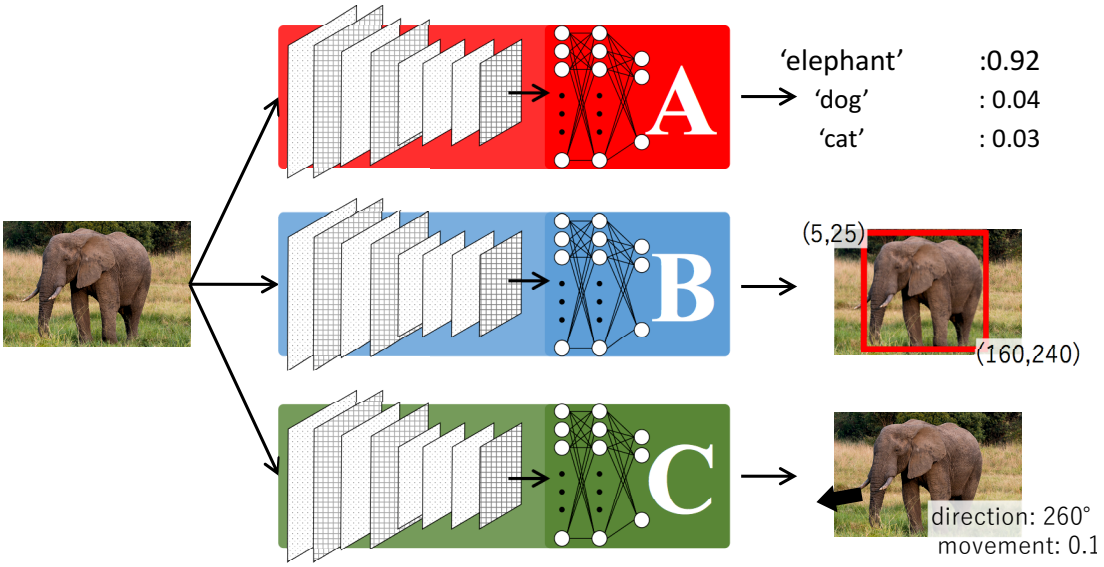

図 5: 一つの入力データに対する複数タスクの推論実行

のタスクに対する後段の処理を行う。共有ニューラルネットワークを利用することで、タスク別に用意された特徴量を抽出するニューラルネットワークが 1 つになる。(タスク数 $\times$ *Hard-shared*) の計算量削減と (対象のタスク $-$ *Hard-shared*) のメモリ使用量が削減され、より多くの計算資源の乏しいエッジ向けハードウェアでの実行も可能となる。また、提案するモデルでは複数タスクに対するモデル全体の学習を行わなわず、タスク別に *Hard-shared* された共有 CNN を用いた転移学習を行うため、タスクの追加に対するネットワーク全体の再学習等を行う必要がなく、不特定多数のユーザーに対して特徴量の転送をすることが可能である。

以降では、提案する実行モデルにおける、画像特徴量の圧縮と共有ニューラルネットワークの詳細を順に述べる。

4.2 効率的な演算を図るニューラルネットワークの共有

提案する共有 CNN を用いたマルチタスク推論のモデルでは、共有 CNN 部をハードパラメーターシェアリングによって共有するため、あらかじめ共有 CNN 部を含んだモデルを学習させる必要がある。本研究では、共有 CNN 部の学習モデルに VGG16、MobileNetV2 の二つのモデルを採用し、物体認識に使用する ILSVRC2012 のデータセットを利用した学習を行い、学習したパラメータを他のタスクにハードパラメーターシェアリングする。共有 CNN を使用したモデルで

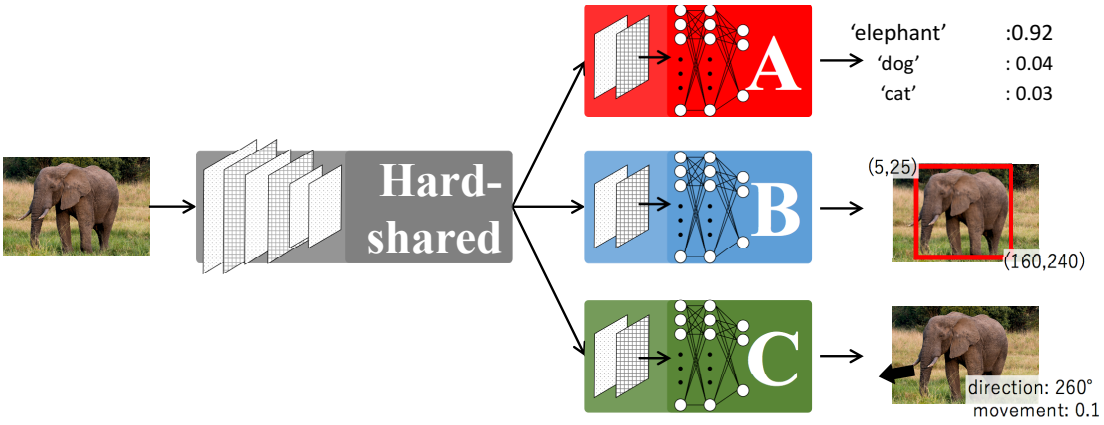

図 6: 共有 CNN を利用した複数タスクにおける推論実行

は、タスクによらず共通の入力画像を利用するため、画像の大きさを揃える必要がある。本研究では、共有 CNN を学習する際の ILSVRC で用いられることの多い (224,224,3) のテンソルになるように画像を縮小する。入力画像の大きさを物体認識のタスクに都合良く定めたため、物体検出のタスクではモデル構造の変更が求められる。

共有 CNN に利用するモデルの分割点であるが、分割点の出力がエッジからクラウドへ転送されるデータとなる。中間データは入力画像を転送するよりデータサイズが小さく、高効率に転送することを目標とするため、本研究では、入力画像よりデータサイズが小さく、また共有 CNN から出力される単一のテンソルのみ転送する。VGG16 の各レイヤーにおける出力要素数を図 7 に示す。入力画像より要素数が小さくなるのは pool4 の出力以降であるため、pool4 以降の出力を中間データとして伝搬させることが望ましい。MobileNetV2 の描くレイヤーにおける出力要素数を図 8 に示す。反転残差構造では、構造内でチャネルの拡大を行うため一時的にデータ量が増える。また、箇所によって反転残差構造への入力を保存する必要があるため、反転残差構造ごとの出力を伝搬させることを考え、図 8 では Bottleneck と示した。入力画像より要素数が小さくなるのは Bottleneck2 の出力以降であるため、Bottleneck2 以降の出力を中間データとして伝搬させることが望ましい。単一のテンソルのみを転送する場合、物体検出のモデルである SSD は中間の feature map を最終層へ伝搬させる必要があるため、転送するテンソルが最終層に伝搬される最大の feature map となる。共有 CNN に VGG16 を用いる場合、SSD のネットワーク構造は変更せずに伝搬する feature map を Conv4_3 の出力から pool4 の出力に変更する。共有 CNN に MobileNetV2 を用いる場合、Bottleneck14 の出力を SSD の Conv5_1 に伝搬するようにネットワーク構造を変更し、伝搬する feature map を Conv4_3 の出力から Bottleneck7 に変更する。

提案する共有ニューラルネットワークついて述べる。文献 [15] では、異なる 2 つのタスクにおいて 1 つの CNN を共有し、後段にタスク別の判別機を設けている。モデル全体の学習はそれぞれの判別機から得られた誤差を共有 CNN まで逆伝搬させている。そのため、共有 CNN は 2 つの誤差を用いて学習されるため、その共有 CNN は 2 つのタスクに特化していると言える。提案手法では特徴量抽

出機として汎化性能の高い重みパラメータを持ったニューラルネットワークを共有する。複数の DNN モデルにおける全体の計算量を見ると、タスク毎に個別のモデルを設けた場合、全体の MAC(Multiply-ACcumulate) 計算量 $MAC(models)$ は式 2 になる。続いて、共有ニューラルネットワークを用いる場合を述べる。まず、あるモデルの分割時におけるモデルの MAC 計算量 $MAC(model)$ は式 3 で表せられる。$model_{former}$ と $model_{latter}$ は、それぞれモデルをある境界で分割した場合において、そのモデルの前段と後段にあたるニューラルネットワークである。特徴量抽出機の役割を持つ前段 $model_{former}$ を共有畳み込みニューラルネットワーク CNN_{Hard_shared} で置き換えたならば、全体の MAC 計算量は式 4 になる。

$$MAC(models) = \sum_{k=1}^{n} MAC(model_k) \tag{2}$$

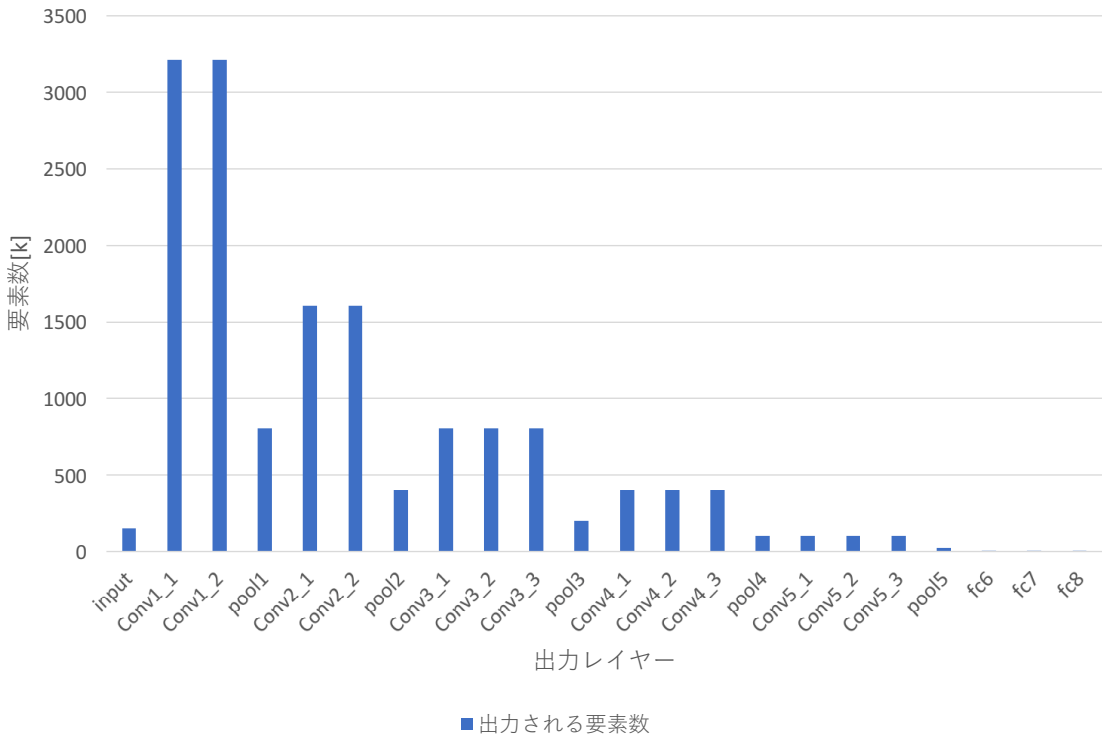

図 7: 1000 クラス分類のための VGG16 の出力要素数

$$MAC(model) = MAC(model_{former}) + MAC(model_{latter}) \tag{3}$$

$$MAC(models) = MAC(CNN_{Hard\_shared}) + \sum_{k=1}^{n} MAC(model_{k,latter}) \tag{4}$$

よって、前段のニューラルネットワークを共有 CNN で置換したモデル全体では式 5 だけ計算量削減が可能になる。一方、モデルの分割境界により、削減できる計算量と各モデルの精度はトレードオフの関係にある。なぜなら、CNN 本来の役割の特徴量の抽出だが、CNN の層が深くなるにつれパラメータは学習したデータセットに特化するためである。そのため、分割境界は分割時における各タスクの精度が、共有ニューラルネットワークを利用しないモデルと比較してどれだけ低下しているかは調査する必要がある。

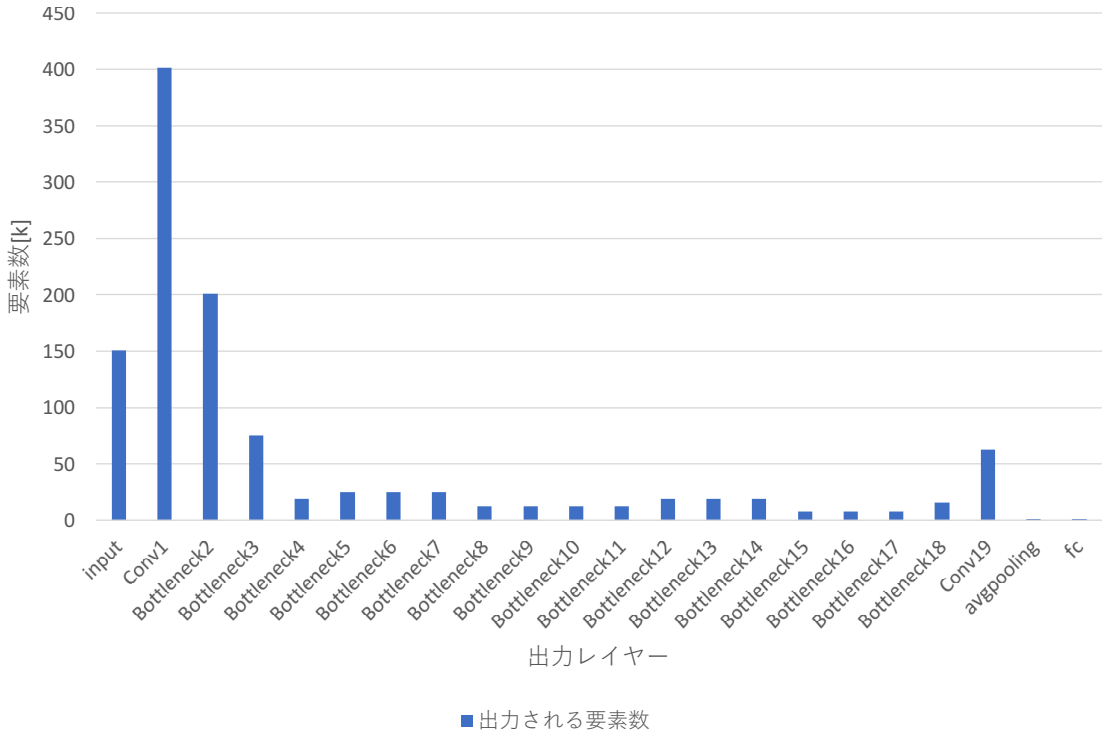

図 8: 1000 クラス分類のための MobileNetV2 の出力要素数

$$diff = MAC(CNN_{Hard\_shared}) - \sum_{k=1}^{n} MAC(model_{k,latter}) \tag{5}$$

4.3 転送データ量を抑える特徴量データ圧縮

エッジとクラウドでの分割推論実行では中間データの転送を行うが、通常のクライアントサーバーアプリケーションと同様に転送するデータを圧縮することが考えられる。画像を入力としたときの VGG16 の pool4 の出力と MobileNetV2 の Bottleneck7 の出力を可視化したものを図 9 に示す。VGG16 の構造では活性化関数に ReLU を利用しているため、出力される中間データは 0 が多く含まれ、圧縮方法によっては圧縮率を高める可能性がある。一方で MobileNetV2 では反転残差構造を採用しているため、VGG16 の中間データよりもスパースな構造ではないため、画像圧縮を行う上では VGG16 の中間データよりも圧縮率が高くなる可能性がある。深層学習の中間出力である特徴量マップに最適化された圧縮手法は現在無いため元データ量が多く、圧縮率向上について多く研究されている動画像の圧縮手法を利用する。動画像の圧縮には可逆圧縮と非可逆圧縮がある。名称のとおり、可逆圧縮は圧縮後にデータの欠損はないが、非可逆圧縮では圧縮手法や圧縮時に指定するパラメーターごとにデータの欠損具合と圧縮率が変化する。本研究では非可逆圧縮のアルゴリズムとして H.265/HEVC[28] を主に使用する。HEVC とは動画圧縮に使用されるコーデックであり、画像圧縮に応用をしたのが BPG[1] である。また、別の画像の非可逆圧縮のアルゴリズムとして WebP[2] を利用する。

5. 実験

本章では、提案した共有 CNN を用いた複数タスクに対する深層学習モデルの学習、学習済みのモデルを分割推論実行する際に中間データを非可逆圧縮する手法について行なった実験を 5.1 節と 5.2 節で述べる。

5.1 共有 CNN

本節では、異なるタスクに対して一部に共通の CNN を用いた深層学習モデルを学習させた場合の精度を調査する実験を行う。タスクには物体認識と物体検出の二つを対象とする。実験で使用したデータセットを表 1 に示す。物体認識は ILSVRC2012 で利用された 1000 クラス分類の画像のデータセットを利用し、物体検出には PASCAL VOC の 2007 年と 2012 年の画像のデータセットを組み合わせたものを利用する。共有 CNN に利用するモデルは物体認識のタスクにおいてモデル構造を考える上で転機となったモデルである VGG16、MobileNetV2 を採用する。共有に使用するモデルは ILSVRC2012 で利用されたデータセットで学習したものを利用した。共有する箇所は、VGG16 は入力から 4 つ目の Max pooling の処理後までを共有し、出力された (512,14,14) のテンソルをタスク別モデルに伝搬させる。MobileNetV2 は入力から 6 つ目の反転残差構造までを共有し、出力された (32,28,28) のテンソルをタスク別モデルに伝搬させる。モデル別の共通の前処

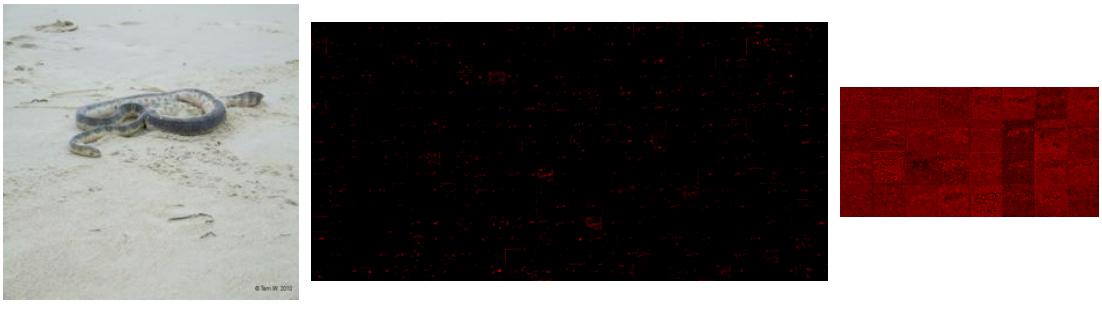

図 9: (1) 元画像 (2)VGG16 の中間データ (3)MobileNetV2 の中間データ

表 1: 実験で用いたデータセット

データセット	train[枚数]	test[枚数]	クラス数	入力画像サイズ
ILSVRC2012	1,281,167	50,000	1,000	224
PASCAL VOC2007+2012	16,551	4,952	20	224

表 2: 各モデルの学習環境

共有 CNN	VGG16		MobileNetV2	
タスク別モデル	VGG16	SSD224	MobileNetV2	SSD224
最適化関数	MomentumSGD	MomentumSGD	MomentumSGD	MomentumSGD
学習率	0.01	0.001	0.01	0.001
WeightDecay	0.1	0.1	0.1	0.1
バッチサイズ	128	32	128	32

理は、VGG16 のときは式 6 で示すように入力データを (224,224,3) に縮小したあと、平均値を引き、255 で割ることによって 0-1 量子化を行なった。MobileNetV2 のときは式 7 で示すように入力データを (224,224,3) に縮小したあと、128 で割り、1 を引いた。共有 CNN の学習時には、初期値を学習済みモデルにより決定する。VGG16 は Caffe の学習済みモデルを利用し、MobileNetV2 は Tensorflow の学習済みモデルのパラメーターを初期値として利用した。物体認識では fine-tuning を行い、物体検出では転移学習を行なっているため、学習時のパラメーターは表 2 に示すとおり、最適化関数には MomentumSGD を利用した。

$$preprocessedData = \frac{resize(input) - mean}{255.} \tag{6}$$

$$preprocessedData = \frac{resize(input)}{128.} - 1. \tag{7}$$

5.2 特徴量圧縮

本節では、前節で述べた共有 CNN の出力を圧縮する手法について述べる。本実験で行うエッジークラウド間の中間データの圧縮展開フローを以下に示す。

1. （エッジ）中間データのテンソルを RGB 画像のフォーマットのテンソルへと変換する
2. （エッジ）データ単位が unsigned int8 になるように Min-Max 量子化を行う
3. （エッジ）量子化されたデータの画像圧縮を行う
4. 圧縮されたデータと Min-Max 量子化で使用された最大値、最小値を転送する
5. （クラウド）圧縮されたデータを展開する
6. （クラウド）最小値、最大値をもとに浮動小数点 32bit のデータ単位になるように逆量子化を行う
7. （クラウド）始めに中間データのテンソルを変換した手法の逆変換を行う

深層学習の推論の途中で生成されるテンソルは画像のように (Height, Width, 3 Channel) のテンソルで表されていないため、画像圧縮をするために変換を行う必要がある。本研究では、表 3 に示すように中間データごとの大きさの画像を用意し、図 10 に示すように中間データのチャネルごとの二次元のテンソルを画像の Red Channel にタイリングを行い、Green Channel と Red Channel を 0 でパディングを行なった。深層学習の推論では浮動小数点 32bit が使用されることが多く、本実験でも使用している。深層学習の推論中の中間データのような特徴量を圧縮する専門の手法はないため、本研究では画像圧縮に用いられる非可逆圧縮の BPG、WebP を用いた圧縮を行う。画像圧縮を行う際にはデータ単位を unsigned int8 または unsigned int16 が多く用いられるため、画像圧縮の前に量子化を行う必要があり、本実験では式 8 で示す Min-Max 量子化と式 9 で示す Min-Max 逆量子化を採用している。Min-Max 逆量子化をクラウドで使用するためには圧縮され

表 3: 中間データのテンソルの変換

モデル	変換前 (Height, Width, Channel)	変換後 (Height, Width, Channel)
VGG16	(512,14,14)	(224,448,3)
MobileNetV2	(32,28,28)	(112,224,3)

たデータの最大値、最小値が必要となるため、転送の際には圧縮されたデータと共に最大値、最小値も転送する必要がある。

$$\tilde{V} = round\left(\frac{V - min(V)}{max(V) - min(V)} \cdot (2^n - 1)\right) \tag{8}$$

$$\hat{V} = min + \tilde{V} \cdot \frac{max - min}{(2^n - 1)} \tag{9}$$

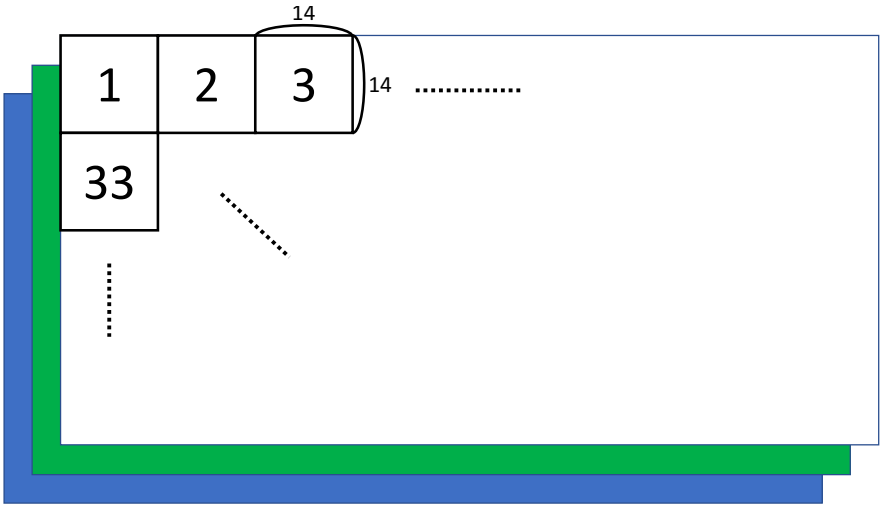

図 10: VGG16 の中間データのタイリング

6. 実験結果

本章では、5 章で述べた実験を行った結果を示す。実験結果は 5 章の節に対応して記述する。

6.1 共有 CNN による認識率と全体計算量の変化

本節では、5.1 節の実験の評価を行う。共有 CNN を作成するために物体認識タスクを利用したモデルの学習をした結果を表 4 に示す。そして、物体検出のタスクに対して 5.1 節で示した共有 CNN 部を用いた SSD モデルの転移学習によって学習した結果を表 5 に示す。SSD の研究では同データセットのタスクに対して 74.3%の精度を示しているため、VGG16 では 17.27%、MobileNetV2 では 13.26%の精度の劣化を確認した。物体検出の研究では入力画像のサイズを物体認識のタスクに比べて大きくする傾向がある。SSD の研究では、入力画像の大きさを (300,300,3) に、feature map の最大の大きさを (512,38,38) に設定している。本実験での入力画像は (224,224,3) であり、feature map の最大の大きさは (512,14,14) になるため、元々の SSD の精度より特徴量が小さいため精度の劣化が生まれたと考えられる。しかしながら入力画像の大きさや feature map の大きさの問題は、入力画像の大きさを変更することで計算量とメモリ使用率のトレードオフと引き換えに解消することができる課題であると考えるため、入力画像サイズの大きさによる精度の向上は今後の課題とする。

共有 CNN に用いる VGG16 と MobileNetV2 のモデルの各レイヤーごとの MAC 計算量を図 11、12 にそれぞれ示す。VGG16 の Conv4_3 までの共有部の計算量は全体の 90.23%を含んでいるため、推論の分割によってクラウドへの計算負荷の集中を抑制することができると考えられる。一方で MobileNetV2 の Bottleneck7 までの共有部の計算量は全体の 25.23%であり、VGG16 と比較をするとエッジで負担する計算の割合は多くないが、全体計算量が VGG16 の三分の一程度であり、ネットワーク全体で見ると計算量が多くないため、このような結果になったと考えられる。物体認識と物体検出の二つのタスクに対して。計算量の削減割合を MAC 計算量を元に計算したものを表 6 に示す。VGG16 を共有 CNN に採用した

表 4: 物体認識を対象とした共有モデルの学習結果

共有モデル	Top-1	Top-5
VGG16	70.50%	89.22%
MobileNetV2	67.86%	88.05%

場合は 42.3%の削減、MobileNetV2 を共有 CNN に採用した場合は 9.77% の削減が可能である。

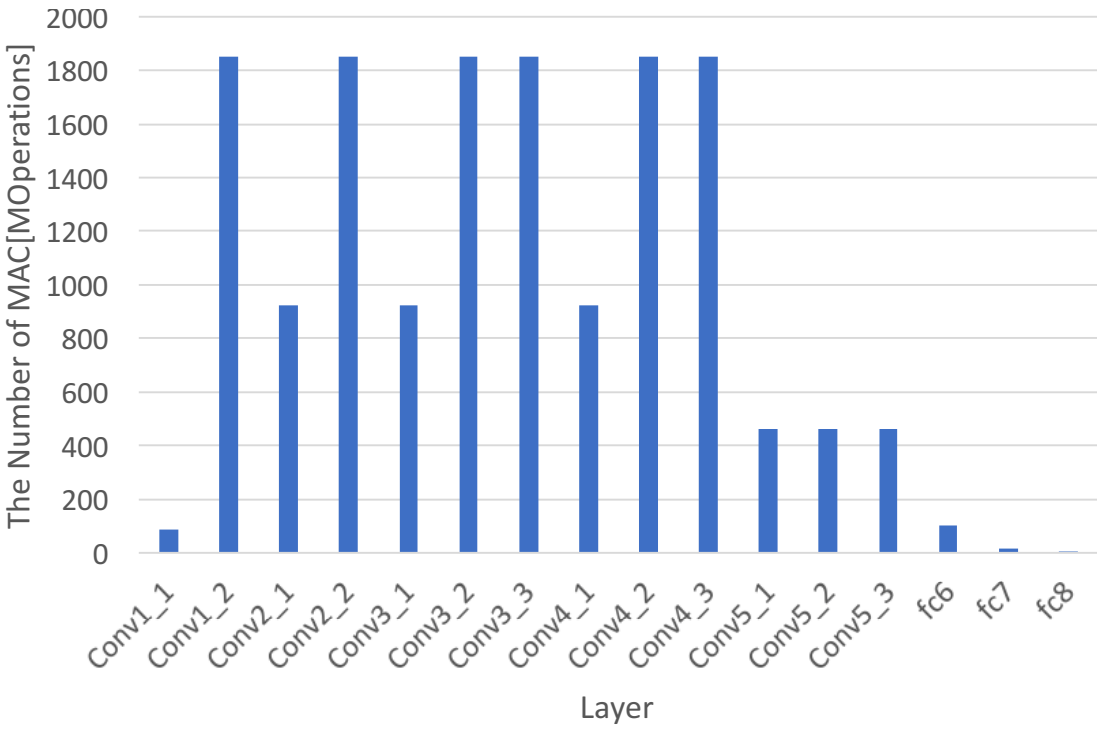

図 11: VGG16 の各レイヤーごとの MAC 計算量

表 5: 共有モデルを利用した物体検出の学習結果

共有モデル	mAP
VGG16-SSD224	57.03%
MobileNetV2-SSD224	61.04%

表 6: 共有 CNN を利用した場合の全体 MAC 計算量の削減割合

	VGG16	MobileNetV2
計算量削減割合	42.30%	9.77%

6.2 特徴量データ圧縮と認識率

本節では、5.2 節の実験の評価を行う。推論の中間データの圧縮を評価するために、入力データとのデータサイズの比較を行う。ILSVRC2012 と PASCAL

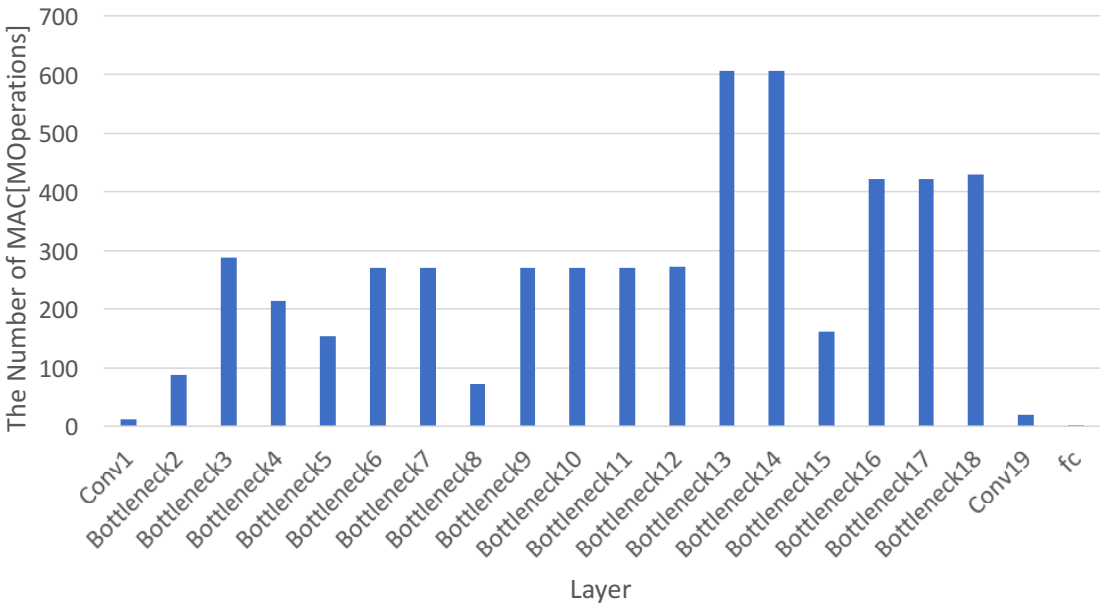

図 12: MobileNetV2 の各レイヤーごとの MAC 計算量

VOC2007 のテストデータは JPEG で提供されているため比較するために (224,224,3) に縮小をする。中間データは 5.2 節での量子化とタイリングを行なったあとのデータを PNG 圧縮した。入力画像、量子化後の中間データのヒストグラムを図 13、14、15 に示し、これらのデータサイズを表 7 に示す。入力画像と PNG 圧縮した中間データのデータサイズを比較すると、PNG 圧縮した中間データは VGG16、MobileNetV2 ともに入力画像よりも大きいことがわかる。量子化後の中間データのヒストグラムを見ると、VGG16 では、ReLU による非線形処理後のためピクセル値が 0 の分布が多くなることが期待されたが、全体の 5%程度にとどまり、その他のピクセル値も際立って局所的な値の集中は見られない。MobileNetV2 では、反転残差構造による処理によるピクセル値の分布の局所的な値の集中は見られず、要素数では入力画像の六分の一程度であるが入力画像よりもデータサイズが大きくなっていることがわかる。

モデルの精度の劣化について述べるが、中間データは PNG 生成過程で量子化を行うため、精度の変化が予想される。量子化後の認識率を表 8,9 に示す。量子化前の認識率である表 4,5 と比較をすると、全てのタスク、モデルに組み合わせにお

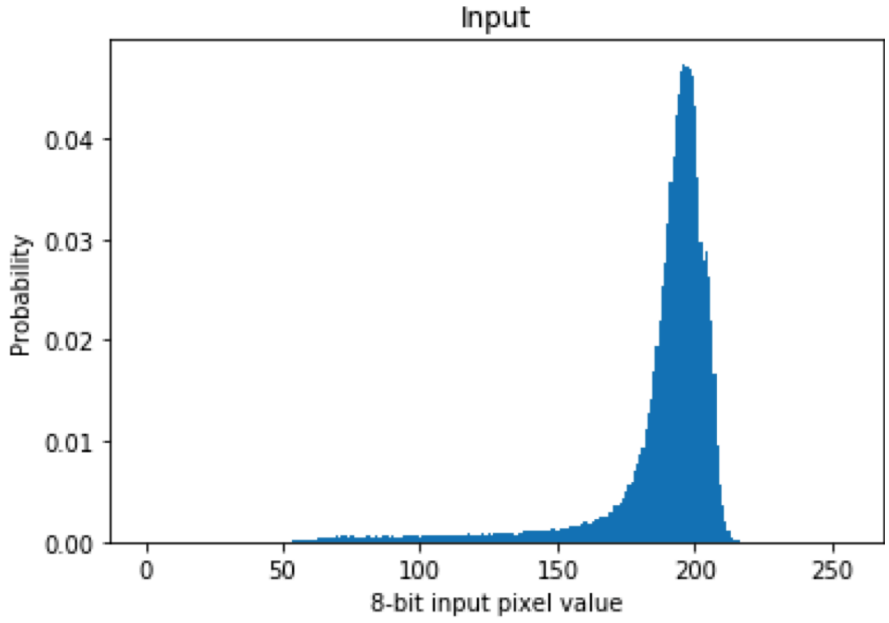

図 13: 入力画像のピクセル値のヒストグラム

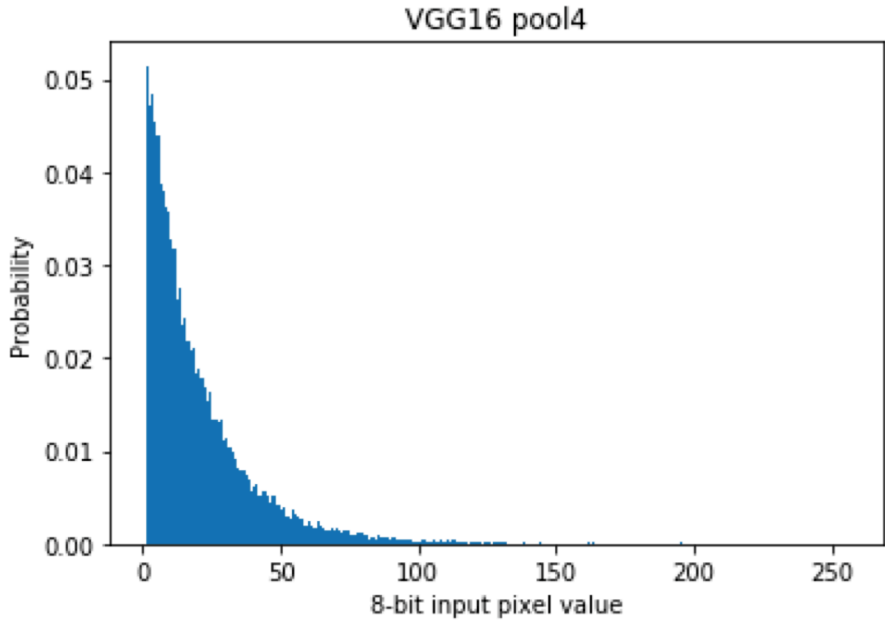

図 14: VGG16 Conv4_3 の出力値のピクセル値のヒストグラム

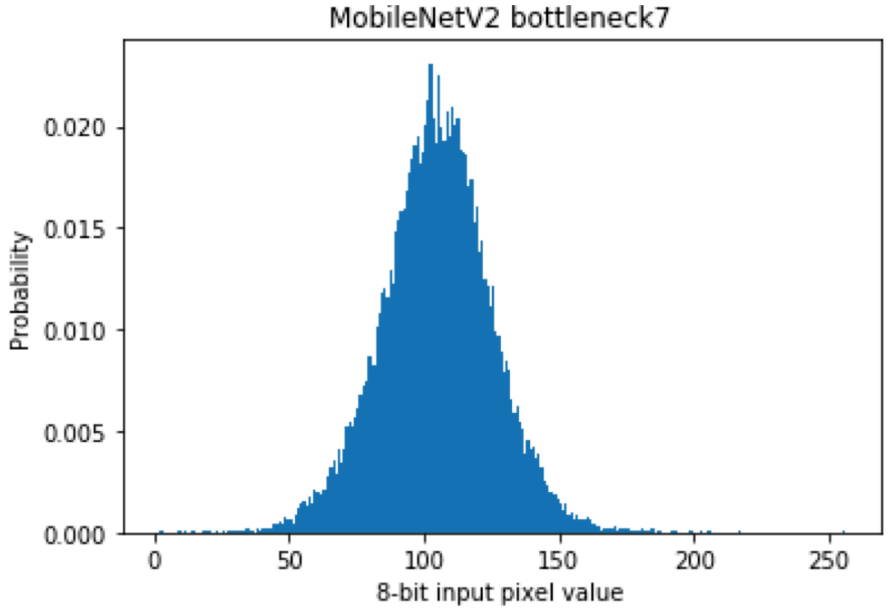

図 15: MobileNetV2 bottleneck7 の出力値のヒストグラム

表 7: 各層の出力データサイズ [Mbytes]

データセット	input[JPEG]	VGG16-conv4-3[PNG]	MobileNetV2-conv7[PNG]
ILSVRC2012	485.44	1742.48	1016.01
PASCAL VOC2007	89.47	158.05	100.31

表 8: 物体認識を対象とした共有モデルの Min-Max 量子化後の認識結果

共有モデル	Top-1	Top-5
VGG16	70.52%	89.22%
MobileNetV2	67.84%	88.05%

表 9: 共有モデルを利用した物体検出の Min-Max 量子化後の認識結果

共有モデル	mAP
VGG16-SSD224	57.07%
MobileNetV2-SSD224	61.07%

いて最大 0.02%の精度の劣化が見られた。次に ILSVRC2012 のデータセットに対して PNG 圧縮された中間データを BPG 圧縮した後の圧縮率、認識率を表 10 に示し、WebP 圧縮した後の圧縮率、認識率を表 10 に示す。次に PASCAL VOC2007 のデータセットに対して PNG 圧縮された中間データを BPG 圧縮した後の圧縮率、認識率を表 12 に示し、WebP 圧縮した後の圧縮率、認識率を表 13 に示す。

非可逆圧縮に BPG を使用した場合の精度と WebP を使用した場合の精度を比較する。BPG を qp35 で使用したときに、物体認識の VGG16 の中間データは 63.81%、MobileNetV2 の中間データは 31.30%の圧縮率を記録している。WebP の qp80 では VGG16 の中間データでは 89.22%、MobileNetV2 の中間データでは 46.55%を示す。物体検出では、BPG が qp35 のとき VGG16 の中間データは 32.66%、16.63%の圧縮率を示した。一方で WebP が qp80 のとき VGG16 の中間

表 10: 物体認識の中間データに BPG 圧縮を行なったときの圧縮率と精度

量子化	VGG16			MobileNetV2		
パラメータ	圧縮率	Top1	Top5	圧縮率	Top1	Top5
qp10	485.23	70.52	89.24	177.80	67.85	88.05
qp20	298.11	70.48	89.20	117.62	67.77	87.99
qp30	131.60	69.98	89.04	61.00	67.16	87.63
qp35	63.81	68.39	88.18	31.30	65.27	86.43
qp40	19.92	57.93	80.94	9.80	49.99	74.41
qp50	1.30	0.99	2.63	1.08	0.73	2.00

表 11: 物体認識の中間データに WebP 圧縮を行なったときの圧縮率と精度

量子化	VGG16			MobileNetV2		
パラメータ	圧縮率	Top1	Top5	圧縮率	Top1	Top5
qp10	7.71	10.37	22.07	4.46	1.294	3.40
qp20	12.84	26.57	48.47	7.47	3.96	9.74
qp30	19.55	40.61	65.60	11.09	11.48	24.86
qp40	26.53	48.75	73.61	15.01	22.60	43.48
qp50	34.91	54.51	78.36	19.62	33.62	57.72
qp60	44.74	58.37	81.25	24.77	41.03	66.39
qp70	57.04	61.09	83.23	31.22	46.40	71.84
qp80	89.22	64.38	85.42	46.55	51.78	76.59
qp90	189.87	66.52	86.90	84.43	54.87	79.05

データは 46.34%、24.72%の圧縮率を示している。物体認識、物体検出ともに同程度の圧縮率の場合に BPG を利用した非可逆圧縮を行うほうが VGG16、MobileNetV2 ともに精度が良い。

これらの結果から、CNN の中間データの特徴量を非可逆圧縮するとき画像の

表 12: 物体検出の中間データに BPG 圧縮を行なったときの圧縮率と精度

量子化	VGG16		MobileNetV2	
パラメータ	圧縮率	mAP	圧縮率	mAP
qp10	233.82	57.16	95.21	61.09
qp20	144.58	57.14	62.90	61.07
qp30	65.17	56.29	32.54	60.59
qp35	32.66	54.37	16.63	59.39
qp40	11.12	42.54	5.19	50.62
qp50	0.73	1.39	0.61	2.37

表 13: 物体検出の中間データに WebP 圧縮を行なったときの圧縮率と精度

量子化	VGG16		MobileNetV2	
パラメータ	圧縮率	mAP	圧縮率	mAP
qp10	4.53	6.80	2.42	3.95
qp20	7.29	12.69	4.01	11.80
qp30	11.08	21.99	5.92	22.32
qp40	14.91	29.77	7.99	33.39
qp50	19.21	36.16	10.43	41.62
qp60	24.37	40.64	13.15	46.35
qp70	30.25	43.76	16.57	50.08
qp80	46.34	47.17	24.72	53.53
qp90	95.82	50.75	44.94	55.12

非可逆圧縮でより良い結果を出したアルゴリズムは中間データの非可逆圧縮においても良い圧縮ができていることがわかる。非可逆圧縮を行なったときの精度の劣化だが、ILSVRC2012 のテストデータセットを用いた 1000 クラスの物体認識タスクに対して BPG 圧縮を用いた推論の分割と中間データの非可逆圧縮では、

qp35 の場合に VGG16 は Top1 で 68.39%、Top5 で 88.18%の認識率を示し、非圧縮の結果と比較して Top1 で 2.11%の精度の劣化を確認した。MobileNetV2 は Top1 で 65.27%、Top5 で 86.43%の認識率を示し、非圧縮の結果と比べて、Top1 で 2.59%の精度の劣化を確認した。BPG 圧縮を用いた推論の分割と中間データの非可逆圧縮では、qp35 の場合に VGG16 は 54.37%の mAP を示し、MobileNetV2 は 59.39%の mAP を示した。非圧縮の結果と比較すると、VGG16 は 2.70%の劣化、MobileNetV2 は 1.68%の劣化を示した。

7. おわりに

本稿では、一対多のエッジコンピューティングにおける共有 CNN を用いたマルチタスクに対する深層学習モデルの推論の分割を提案した。分割推論時に中間データの圧縮には動画像に用いられる非可逆圧縮を提案し、中間データの劣化程度によるモデルの認識精度の変化についてそれぞれ実験を行った。共有 CNN を用いた実験では VGG16 を利用した場合に 42.3%の全体計算量の削減、MobileNetV2 を利用した場合に 9.77%の全体計算量の削減をした。ILSVRC2012 のテストデータセットを用いた 1000 クラスの物体認識タスクに対して BPG 圧縮を用いた推論の分割と中間データの非可逆圧縮では、qp35 の場合に VGG16 は Top1 で 68.39%、Top5 で 88.18%の認識率を示し、MobileNetV2 は Top1 で 65.27%、Top5 で 86.43%の認識率を示した。非圧縮の精度と比較すると Top1 について VGG16 は 2.11%、MobileNetV2 は 2.59%の精度の劣化を確認した。圧縮率では、縮小済みの元画像に対して VGG16 は 63.81%、MobileNetV2 は 31.3%の圧縮率を確認した。PASCAL VOC2007 のテストデータセットを用いた 20 クラスの物体検出タスクに対して BPG 圧縮を用いた推論の分割と中間データの非可逆圧縮では、qp35 の場合に VGG16 は 54.37%の mAP を示し、MobileNetV2 は 59.39%の mAP を示した。実験条件は異なるが、SSD の研究と比較すると VGG16 は 19.93%、14.91%の精度の劣化を確認した。圧縮率では、圧縮済みの元画像に対して VGG16 は 32.66%、MobileNetV2 は 16.63% の圧縮率を確認した。

これらの結果から共有 CNN を用いたネットワークの分割と中間データの非可逆圧縮により、クラウドへの計算負荷の集中、転送データサイズの削減を精度のトレードオフとともに達成した。今後の展望として、本実験で示した物体認識で用いられる画像の前処理、モデル構造ではなく、物体検出や関節点推定等の他のタスクの精度に影響しない統一的な前処理たモデル構造の提案が挙げられる。

謝辞

研究活動や課外活動、日常生活まで多くの指導をして頂いた本学の中島康彦教授に感謝の意を申し上げます。本稿をご精読いただき。発表の場においても貴重なご意見を頂いた本学の井上美智子教授に深く感謝致します。本研究へ多くの助言や、修士生活を通して生徒の自主性を重んじた指導をしていただいた本学の中田尚准教授に感謝の意を表します。また日々の研究活動や本稿の執筆をはじめ多くのご助言を頂いた同研究室の TRAN Thi Hong 助教、張任遠助教に心より御礼申し上げます。研究室の特論や日々の研究、就職活動への助言等でご指導していただいた山野龍祐氏、福岡久和氏、三谷剛正氏、修士二年間を研究から日常生活まで支えてくれた木村睦氏、一倉孝宏氏、上竹規之氏、菊谷雄真氏、山根弘樹氏、吉田怜矢氏、後輩の池田裕哉氏、岩本淳氏、西本宏樹氏、新谷隆太氏、その他コンピューティング・アーキテクチャ研究室 OB の方々からの助力に深く感謝します。

最後に大学院進学をするにあたり、経済的、心理的に支えてくれた家族に感謝の意を申し上げます。

参考文献

[1] https://bellard.org/bpg/.

[2] https://developers.google.com/speed/webp/.

[3] 将来のネットワークインフラに関する研究会（第 4 回）配付資料. http://www.soumu.go.jp/main_sosiki/kenkyu/nwinfra/02kiban05_04000270.html.

[4] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization, 2018.

[5] Tan B., Song Y., Zhong E., and Yang Q. Transitive transfer learning. In KDD ’ 15, 2015.

[6] Rich Caruana. Multitask learning. *Mach. Learn.*, 28(1):41–75, July 1997.

[7] Tianshi Chen, Zidong Du, Ninghui Sun, Jia Wang, Chengyong Wu, Yunji Chen, and Olivier Temam. Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. In *ACM Sigplan Notices*, volume 49, pages 269–284. ACM, 2014.

[8] H. Choi and I. V. Bajić. Near-lossless deep feature compression for collaborative intelligence. IEEE MMSP’18, Vancouver, BC, 2018.

[9] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[11] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017.

[12] Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor N. Mudge, Jason Mars, and Lingjia Tang. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *ASPLOS*, 615–629, ACM, 2017.

[13] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? CoRR, abs/1805.08974, 2018.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[15] Sijin LI, Zhi-Qiang Liu, and Antoni B. Chan. Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2014.

[16] Darryl Dexu Lin, Sachin S. Talathi, and V. Sreekanth Annapureddy. Fixed point quantization of deep convolutional networks. *CoRR*, abs/1511.06393, 2015.

[17] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[18] M. T. Rosenstein Z. Marx, L. P. Kaelbling, and T. G. Dietterich. To transfer or not to transfer. In *In NIPS 2005 Workshop on Transfer Learning*, page volume 898, 2005.

[19] Tony Nowatzki, Vinay Gangadhar, Karthikeyan Sankaralingam, and Greg Wright. Domain specialization is generally unnecessary for accelerators. *IEEE Micro*, 37(3):40–50, 2017.

[20] Masayuki Ohzeki. Statistical-mechanical analysis of pre-training and fine tuning in deep learning. CoRR, abs/1501.04413, 2015.

[21] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. *CoRR*, abs/1603.05279, 2016.

[22] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016.

[23] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. CoRR, abs/1602.04938, 2016.

[24] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[26] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. CoRR, abs/1801.04381, 2018.

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[28] Gary J Sullivan, Jens Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

[29] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee. Cross-language transfer learning for deep neural network based speech enhancement. In *in IEEE Int. Symp. on Chinese Spoken Language Processing (ISCSLP)*, pages 336—-340, 2014.

[30] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *CoRR*, abs/1411.1792, 2014.

[31] Y.T. Zhou and Rama Chellappa. Computation of optical flow using a neural network. pages 71 – 78 vol.2, 08 1988.

[32] C.M ビショップ. パターン認識と機械学習（上）（下）. 丸善出版, 2012.

[33] 総務省. 情報通信白書.

[34] 中島康彦. Emaxv における複数バースト転送と複数ベクトル演算のオーバラップ手法. *CPSY*, 15:71–76, 2016.

業績

1. 平賀由利亜, 三谷剛正, 福岡久和, 中田　尚, 中島康彦: ”エッジコンピューティングによる分散ニューラルネットワークの構想”, 信学技報, vol.117, no.44, CPSY2017-11, pp.62-67, May. (2017)

2. 福岡久和, 三谷剛正, 平賀由利亜, 中田尚, 中島康彦: ”動画圧縮技術を利用した分散機械学習における情報伝達効率化”, 信学技報, vol.117, no.153, CPSY2017-30, pp.151-155, Jul. (2017)

3. 福岡久和, 平賀由利亜, 三谷剛正, 中田尚, 中島康彦: ”分散 CNN における圧縮と集約による情報転送の最適化”, 信学技報, vol.117, no.314, CPSY2017-59, pp.51-54, Nov. (2017)

4. Takamasa Mitani, Hisakazu Fukuoka, Yuria Hiraga, Takashi Nakada and Yasuhiko Nakashima: ”Compression and aggregation for optimizing information transmission in distributed CNN”, CANDAR'17, REGULAR PAPER, pp.112-118, Nov. (2017)

5. 【xSIG:Best M1 Student Award】 平賀由利亜, 福岡久和, 三谷剛正, 中田尚, 中島康彦: ”共有 CNN を用いた高効率な分割推論実行モデル”, xSIG 2018: The 2nd. cross-disciplinary Workshop on Computing Systems, Infrastructures, and Programming, Apr. (2018)

6. 平賀由利亜, 三谷剛正, 中田　尚, 中島康彦: ”共有 CNN を用いた高効率なマルチタスクニューラルネットワーク”, 信学技報, vol.118, no.92, CPSY2018-4 DC2018-4, pp.93-98, June. (2018)