

クラウド型顔画像解析エンジンにおけるレイテンシを考慮した DCNN の自動分割

○猪子弘康 山内悠嗣 山下隆義 藤吉弘亘 (中部大学)

1. はじめに

人の生活を支援するロボットと共存する安心安全な社会の構築に向け、サービスロボットの研究開発が積極的に進められている。一般にロボットの計算リソースは限られているため、クラウド上のコンピュータで計算処理を行うクラウドロボティクス [1, 2, 3, 4] への関心が高まっている。我々はこれまでに、Deep Convolutional Neural Network(DCNN)に基づくクラウドロボティクスのための顔画像認識エンジン [5] を提案している。本手法は、DCNN による顔画像を解析するクラウド型の認識エンジンである。DCNN を 2 つに分割して、ロボット側で計算した途中結果である特徴マップをクラウドサーバに送信し、DCNN の解析結果をロボットに転送する。これにより、ロボット側の計算負荷や転送量を小さくできる。また、顔画像を転送する必要がないため、プライバシーに配慮できるという特長がある。しかし、この手法では DCNN を分割する層を事前に決定する必要があり、負荷分散を効率的に行うには、ロボットの性能や通信環境の状態に応じて動的に分割層を決定する必要がある。

そこで、本稿ではクラウド型顔画像解析エンジンにおけるレイテンシを考慮した DCNN の自動分割法を提案する。提案するシステムでは、ユーザが画像を入力してから出力結果を得るまでの希望時間（要求レイテンシ）を予め設定する。DCNN の分割層は、クライアントの処理時間、クラウドサーバの処理時間、ネットワークの伝送時間、レイテンシの 4 つの要素から自動で決定する。

2. レイテンシを考慮した DCNN の自動分割

本章では、クラウド型顔画像解析エンジンにおけるレイテンシを考慮した DCNN の自動分割法について述べる。

2.1 システムの構成

本システムの構成を図 1 に示す。ロボット側では入力画像から検出した顔画像を DCNN の上層に投入し、出力される特徴マップを、ユーザが設定した要求レイテンシと共にクラウドサーバに送信する。クラウドサーバ側では送信されてきた特徴マップを DCNN の下層に投入し、顔画像解析結果（人種、年齢、性別、表情、器官点）を得る。また、要求レイテンシを元に分割層を決定し、最後に得られた顔画像解析結果と分割層をロボットに出力する。

また、システムは Robot Operating System(ROS) を基盤に開発しており、ROS 対応ロボットであれば使用することができる。従来手法では、ロボット側の DCNN

の層とクラウド側の DCNN の層をどのように分割するか、予めユーザが決めておく必要がある。

2.2 分割層の自動決定

この節では、DCNN を自動的に分割する方法とモデル化について述べる。

2.2.1 分割層自動決定の方針

DCNN の分割層を自動的に決定する方法は幾つか考えられる。例えば、応答時間の最小化、クライアントもしくはロボットの計算量を最小化、通信量を最小化という観点から分割層を自動的に決定することができる。しかしながら、これらの方法ではクライアントもしくはクラウドサーバの双方において、計算量等を制御することができない。そこで、我々はクライアント及びサーバの計算時間、通信時間の他に要求レイテンシの合計 4 つの要素から、DCNN を自動的に分割することを考える。これにより、ユーザのリクエストに可能な限り応えられるシステムとする。ただし、サーバの負担を小さくするために、要求レイテンシを満たしている場合には、サーバの計算量を小さくするように DCNN の分割層を決定する。

2.2.2 サーバの負担低減を優先した分割層の決定

2.2.1 で述べた方針をモデル化する。まず第一にユーザからの要求レイテンシを満たす分割層を条件とする。クライアントの処理時間を t_c 、クラウドサーバの処理時間を t_s 、ロボット-クラウドサーバ間の通信時間を t_t とし、決定される分割層 i は、以下となる。

$$\begin{aligned} i &= \arg \max_i t_c(i) \\ \text{s.t. } (t_c(i) + t_s(i) + t_t(i)) &< R \end{aligned} \quad (1)$$

表 1 に要求レイテンシと総処理時間から分割層が決定される例を示す。式 (1) の条件により、まず総処理時間の中から要求レイテンシを満たす分割層が選択される。例えば要求レイテンシが 60[ms] であれば、55[ms] の層が選択される。要求レイテンシが 100[ms] であれば、90, 70, 55, 65[ms] が条件を満たすが、このうち選択されるものは分割層の下層である 90[ms] の 3 層目である。分割層を決定する際、どの分割層も総処理時間が要求レイテンシを上回ってしまい、最適な分割層を決定できない場合がある。この場合は総処理時間が最小となる層を分割層として選択する。

2.3 システムの実装

2.2.2 で説明したモデルを基にシステムを実装する。このシステムのシーケンス図を図 2 に表す。(1) でクライアントとクラウドサーバ間の通信時間を測定する。こ

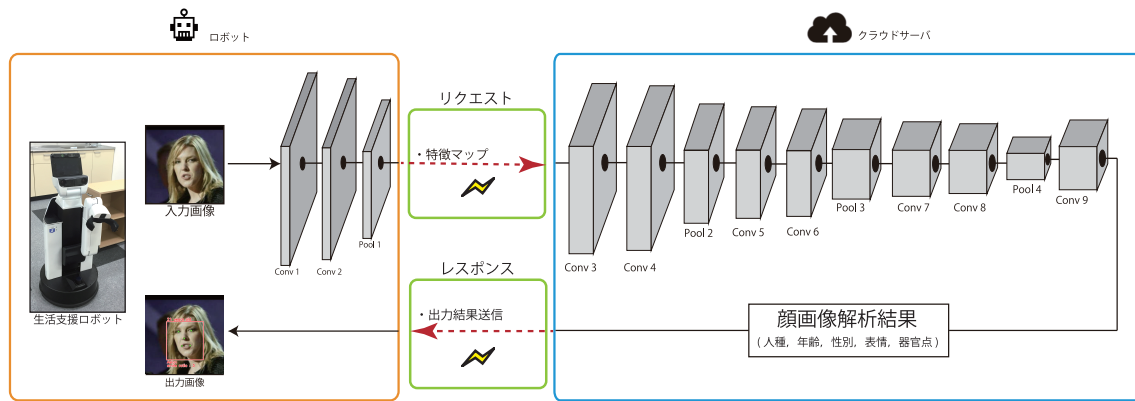


図 1 クラウドによる顔画像の属性推定の流れ (第 2 層目で分割した例)

表 1 要求レイテンシと総処理時間から分割層が決定される例
○は要求レイテンシを満たすもの、✓は選択される分割点

分割層	総処理時間 [ms]	要求レイテンシ [ms]				
		60	100	120	300	500
4	115			✓	✓	✓
3	90		✓	○	○	○
2	70		○	○	○	○
1	55	✓	○	○	○	○
0	65		○	○	○	○

の計測は (2)～(5) までの処理と平行して定期的に、かつ非同期で行われる。(2) ロボット側で DCNN の上層の処理を行い、出力される特徴マップを要求レイテンシと共に (3) でクラウドサーバに送信する。(4) クラウドサーバでは、DCNN の下層の処理を行い、出力される顔画像解析結果を得る。同時に分割層の決定を行う。(5) でロボットに対して顔画像解析結果と決定した分割層の情報を返す。

分割層の決定は、通信時間の測定の結果と、ロボット・クラウドサーバ双方の DCNN の処理時間によって左右される、そこで、次のように測定する。ロボットの処理時間は、各分割層において、ダミーデータを入力し、特徴マップが出力されるまでの時間を測定する。クラウドサーバの処理時間についても、各分割層において、特徴マップを入力し、顔画像解析結果が出力されるまでの時間を測定する。

本システムは ROS 上で開発しているが、ロボット・クラウドサーバ間の通信は ROS のサービスコールという通信方式を利用している。開発言語は Python であり、特徴マップをはじめ多くの処理を numpy の ndarray 型で扱っている。通信を伴う場合においては、ROS の通信プロトコルに合わせる必要がある。本システムを実装するにあたって ndarray の多次元配列を ROS の Float32MultiArray 形式に相互変換するメソッドを実装した。

3. 検証実験

提案システムの有効性を検証するために、評価実験を行う。実験で使用する DCNN の構造を表 2 に示す。

表 2 DCNN の構成

各層	関数	値
入力層	-	100×100
畳み込み層 1	フィルタ	$16 \times 3 \times 3$
	MaxOut	2
畳み込み層 2	フィルタ	$16 \times 3 \times 3$
	MaxOut	2
	MaxPool	2×2
畳み込み層 3	フィルタ	$32 \times 3 \times 3$
	MaxOut	2
畳み込み層 4	フィルタ	$32 \times 3 \times 3$
	MaxOut	2
	MaxPool	2×2
畳み込み層 5	フィルタ	$64 \times 3 \times 3$
	MaxOut	2
畳み込み層 6	フィルタ	$64 \times 3 \times 3$
	MaxOut	2
	MaxPool	2×2
畳み込み層 7	フィルタ	$128 \times 3 \times 3$
	MaxOut	2
畳み込み層 8	フィルタ	$128 \times 3 \times 3$
	MaxOut	2
	MaxPool	2×2
畳み込み層 9	フィルタ	$256 \times 2 \times 2$
	MaxOut	2
全結合層	DropOut	0.5
	ユニット数	200
	活性化関数	Sigmoid
出力層	ユニット数	17

3.1 実験環境

実験に使用するクライアント、サーバの機器の性能を表 3 に示す。クライアント、クラウドサーバ共に OS は Ubuntu14.04LTS で ROS は Indigo を使用した。それぞれの PC はルータを介し 1Gbps で接続した。通信帯域は Ubuntu の tc コマンドによって通信帯域を制限した。クライアントは、様々な性能の計算機が搭載されたロボットを想定するため、CPU のクロック周波数を変化させた。CPU のクロック周波数は Ubuntu の libcpufreq というツールを利用してクロック周波数を変化させる。

3.2 分割層の自動決定の有効性の検証

まず、処理時間及び通信時間を変化させた際の処理時間と通信時間について検証する。図 4 に 3 つの条件

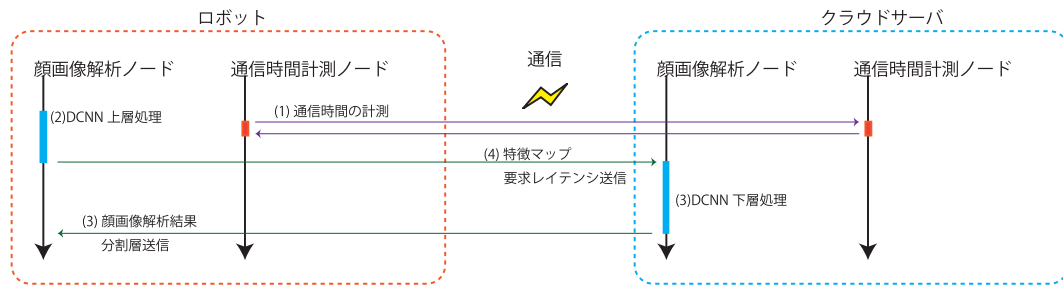


図2 本システムのシーケンス図

表3 使用する端末のスペック

端末	CPU	メモリ	GPU
クライアント	Intel Core i7 810MQ@2.8GHz	8GB	-
クラウドサーバ	Intel Core i7 6700K@4.0GHz	32GB	GeForce GTX1080

における総処理時間の変化を示す。図4(a), (b) から、クライアントの処理性能の差が総処理時間に大きな影響を与えていることがわかる。特に、クライアントの処理性能が低い場合には、分割層が下位になるに従い、処理時間が増加する。次に、図4(b), (c) から、通信速度の差が総処理時間に大きな影響を与えていることがわかる。最後に、図4の3つの図から、処理時間と通信時間の条件が変わると、最も総処理時間が短くなる分割層が異なること読み取れる。要求レイテンシを考慮し、総処理時間が短くなる分割層を選択する必要があることがわかる。

次に、提案するモデルにより、要求レイテンシを考慮しながら理想的に分割層を決定することができるか検証する。表4, 表5, 表6に処理時間及び通信時間を変化させた際の提案モデルにより分割層を決定した結果を示す。表中の通信量とは、各分割層により送信された特徴マップのデータ量である。 t_c , t_s , t_t は、それぞれクライアント及びクラウドサーバの処理時間、通信時間を表しており、Total は総処理時間を表す。

クライアントの処理時間が長く、通信時間が短い場合の表4を見ると、要求レイテンシを満たしながら、可能な限り下位の分割層を選択できていることがわかる。表5に示すクライアントの処理時間が短く、通信時間が短い場合においても同様の結果が得られている。表6に示すクライアントの処理時間が短く、通信時間が長い場合では、要求レイテンシを満たすことができない場合がある。このような場合には、応答時間が最も短くなるような分割層が決定されていることがわかる。

図3はこのシステムによって出力される出力結果と、特徴マップの例である。特徴マップは、第3層目からの出力を可視化した。このとき送信されるデータサイズは132.2kBである。

4. おわりに

本稿では、幾つかあるDCNNの分割層を自動的に決定する方法からユーザのリクエストに可能な限り応え、クラウドサーバの負荷を減らすシステムを提案した。今後の展開として、分割に適したDCNNの畳み込み層やプーリング層などの構造を検証すること、また

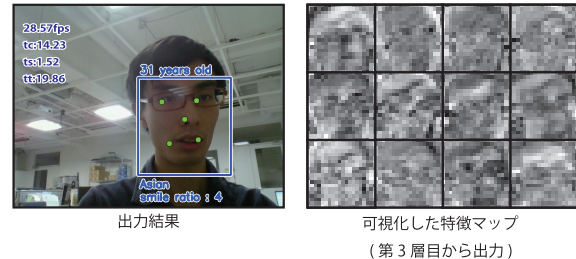


図3 出力結果と可視化した特徴マップの例

実際のロボット上での実装をし、実環境での展開などが挙げられる。

参考文献

- [1] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-Based Robot Grasping with the Google Object Recognition Engine", IEEE International Conference on Robotics and Automation, 2013.
- [2] O. Zweigle, R. van de Molengraft, R. d'Andrea, and K. Haussermann, "Roboearth: connecting robots worldwide", International Conference on Interaction Sciences: Information Technology, Culture and Human. ACM, pp. 184-191, 2009.
- [3] W. J. Beksi, J. Spruth and N. Papanikolopoulos, "CORE: A Cloud-based Object Recognition Engine for Robotics", International Conference on Intelligent Robots and Systems, pp. 4512-4517, 2015.
- [4] K. Sugiura and K. Zettsu, "Rospeex, A Cloud Robotics Platform for Human-Robot Spoken Dialogues", International Conference on Intelligent Robots and Systems, pp. 6155-6160, 2015.
- [5] 山内悠嗣, 加藤優, 山下隆義, 藤吉弘亘, "クラウドロボティクスのための画像認識エンジンの提案", 電子情報通信学会技術報告, パターン認識・メディア理解, vol. 115, no. 456, pp. 91-96, 2016.

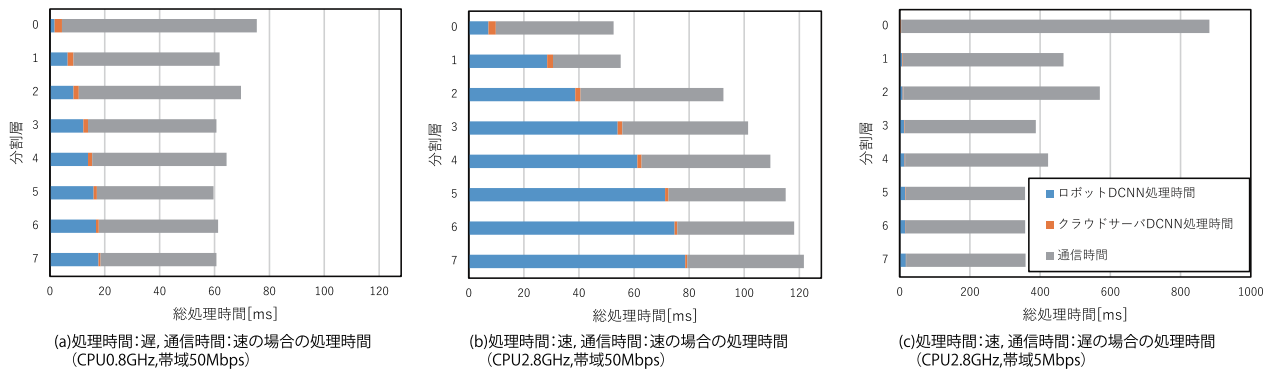


図 4 各条件における総処理時間の変化

表 4 クライアントの処理時間：長、通信時間：短の場合 (CPU 0.8GHz, 50Mbps)
○は要求レイテンシを満たすもの、√は選択される分割点

分割層 i	特徴マップ データサイズ [kB]	t_c [ms]	t_s [ms]	t_t [ms]	Total[ms]	要求レイテンシ [ms]				
						60	100	120	300	500
7	2.2	78.56	0.73	42.34	121.64				√	√
6	12.2	74.70	0.97	42.44	118.12			√	○	○
5	10.1	71.25	1.23	42.55	115.05			○	○	○
4	50.0	61.22	1.52	46.74	109.49			○	○	○
3	30.2	54.04	1.69	45.67	101.41			○	○	○
2	132.2	38.67	1.93	51.84	92.45		√	○	○	○
1	72.0	28.46	2.13	24.55	55.15	√	○	○	○	○
0	301.1	7.16	2.67	42.72	52.56	○	○	○	○	○

表 5 クライアントの処理時間：短、通信時間：短の場合 (CPU 2.8GHz, 50Mbps)

分割層 i	特徴マップ データサイズ [kB]	t_c [ms]	t_s [ms]	t_t [ms]	Total[ms]	要求レイテンシ [ms]				
						60	100	120	300	500
7	2.2	17.72	0.73	42.26	60.71		√	√	√	√
6	12.2	16.81	0.97	43.48	61.27		○	○	○	○
5	10.1	15.90	1.23	42.49	59.63	√	○	○	○	○
4	50.0	13.89	1.52	48.97	64.39		○	○	○	○
3	30.2	12.20	1.69	46.79	60.69		○	○	○	○
2	132.2	8.58	1.93	59.11	69.62		○	○	○	○
1	72.0	6.47	2.13	53.23	61.84		○	○	○	○
0	301.1	1.71	2.67	71.00	75.38		○	○	○	○

表 6 クライアントの処理時間：短、通信時間：長の場合 (CPU 2.8GHz, 5Mbps)

分割層 i	特徴マップ データサイズ [kB]	t_c [ms]	t_s [ms]	t_t [ms]	Total[ms]	要求レイテンシ [ms]				
						60	100	120	300	500
7	2.2	17.72	0.73	339.86	358.317					√
6	12.2	16.81	0.97	339.70	357.495					○
5	10.1	15.90	1.23	339.99	357.137	√	√	√	√	○
4	50.0	13.89	1.52	407.34	422.758					○
3	30.2	12.20	1.69	373.61	387.509					○
2	132.2	8.58	1.93	559.66	570.181					
1	72.0	6.47	2.13	458.12	466.731					○
0	301.1	1.71	2.67	877.67	882.062					