特別研究報告

題目

分散映像分析システムの消費電力最小化方式の実装と評価

指導教員
村田 正幸 教授

報告者
川口峻平

令和 5 年 2 月 7 日

大阪大学 基礎工学部 情報科学科

令和 4 年度 特別研究報告

分散映像分析システムの消費電力最小化方式の実装と評価

川口峻平

内容梗概

近年，VR やデジタルツインなどの技術により，目の前にあるものを操作する感覚で遠隔にあるものやデジタル空間上のものを制御するサービスが普及してきており，リアルタイムな応答時間とともに精度の高い映像データ処理が追求されている．一方，世界で使用される電子機器の多くがインターネットに接続されることにより，ネットワークを流れるデータ量が増加し，通信や計算のために消費される電力の増加が大きな問題となっている．

従来，画像や映像などの分析は，クラウドに配備された人工知能技術を用いて行われることが想定されており，ネットワーク上を大量の映像データが流れることにより，通信や計算に要する電力の増加が問題視されている．近年では，ユーザーに近い場所にコンピューティングリソースを配置するエッジコンピューティング技術により，クラウドまで流れるトラフィック量を削減することで消費電力を抑制でき，かつレイテンシを削減できる．一方クラウドでは豊富なコンピューティングリソースを利用して，高精度な映像分析ソフトウェアを用いたとしても短時間で分析を行い，分析結果を取得することが可能である．

本報告では，ネットワーク通信や人工知能による分析を統合して扱うことで，分散映像分析システムに要する全体の消費電力削減を行う．システムを構成する計算機にそれぞれ最適な割合で分析タスクを分散させることにより，遅延と認識精度といった制約を満たしつつ電力消費を最小化する組み合わせ最適化問題を定式化する．しかし，一般に組合せ最適化問題は，問題の規模が大きくなると解の導出が著しく困難となり，大規模かつ環境が変化するネットワークにおいて，リアルタイムに最適解を導出することは困難である．

大規模で複雑な最適化問題に対しては，近似解を導出するメタヒューリスティクスが用いられることが多く，その一つに遺伝的アルゴリズムがある．遺伝的アルゴリズムは生物の進化のプロセスが，環境に対して遺伝的物質を最適化するという概念に着想を得たアルゴリズムであり，環境の変化に順応して最適化することが可能である．本報告では，遺伝的アルゴリズムを用いて解を導出し，映像分析のタスクをエッジ・クラウドコンピューティングシステムを構成する計算機に適切に割り振ることで，遅延および分析精度の制約条件を満たしつつ，消費電力を最小化する．

本報告では映像分析システムを実機を用いて構築し，映像分析タスクの割り当てを行い，消費電力最小化の効果を示す．さらに，消費電力予測モデルの予測値と実測値をもとに，予測モデルを改善し，誤差を低減する．実機による評価を行い，平均絶対パーセント誤差を3.5%に抑えられることを示すとともに，消費電力を 24.5%削減できることを示した．

主な用語

遺伝的アルゴリズム
エッジコンピューティング
クラウドコンピューティング
消費電力最小化
消費電力予測モデル

目 次

図 目 次

表 目 次

1　はじめに

Internet of Things (IoT) の普及により家電製品から企業のデータセンターに至るまであらゆるものがインターネットに接続して使用されるようになり，多くのサービスが出現するとともに，ネットワークを流れるデータ量が増大している．近年では，自動運転技術やVirtual Reality (VR) など，第 5 世代無線通信システム（5G）を前提としたアプリケーションの開発が進んでおり，ネットワーク通信におけるトラフィック量は今後，さらに増大すると見込まれている [1,2]．一方で 2030 年には通信技術に使用される電力量が世界で使用される総電力量の 51%を占めると試算されており，情報通信における省電力化の重要性が高まっている [3,4].

5G の展開に伴い，VR やデジタルツイン技術を利用したアプリケーションへの期待が特に高まっている．目の前にあるデバイスを操作する際と同じ感覚で，インターネットに接続された “モノ” や遠隔デバイスを操作できるアプリケーションにおいては極めて短時間で，高精度の映像分析結果が取得できることが必要となる．従来，画像や映像などの分析は，クラウド上の計算機に配備された人工知能技術を用いて行われることが想定されており，そのためには，ネットワーク上を大量の映像データが流れることとなる．5G に関する研究開発の中では，クラウド上の計算機において処理されていたデータの一部を基地局上の計算資源（エッジ）に移行させ処理を行う，エッジコンピューティング技術の開発が進められており，これによりクラウドまで流れるトラフィック量の大幅な削減が期待されている．エッジコンピューティングを活用することで電力消費の抑制，レイテンシの削減が期待される．一方で，エッジ上の計算資源はクラウド上の計算資源と比較して小さく，アプリケーションの要求するレイテンシや分析精度に応じて，クラウド上の豊富な計算資源を利用することも重要となる．すなわち，データソースと計算処理を行う機器との距離を短くすることのできるエッジコンピューティング技術と，コンピューティングリソースの豊富なデータセンターを利用するクラウドコンピューティング技術を適切に選択し，使用することが重要となる．

我々の研究グループでは，このような分散型の映像分析システムを対象とした省電力化技術の開発を行っている．文献 [5] では，エッジ・クラウド上の計算資源を利用する映像分析システムを対象とした省電力化方式が提案されている．この方式では，カメラによって撮影された映像の分析は，そのカメラを備えた端末上，あるいはエッジ上，クラウド上の計算機に分割して割り当てられる．分析処理を行うタスクの総量は一定であり，その分析場所が分散されたとしても，分析処理を行う計算機が計算に要する消費電力の総量は大きくは変化しないと考えられる．一方で，前述の通り，ネットワーク上を流れるトラフィック量は大きく異なり，端末やエッジ上の計算資源を活用することで省電力化を図ることができる．このように，ネットワーク通信と映像分析処理を統合して考慮することで，分散映像分析システム

全体の省電力化を図ることが可能である．

文献 [5] では前述の省電力化を組み合わせ最適化問題として定式化し，最適解の導出を行っている．一般に組合せ最適化問題は，問題の規模が大きくなると，厳密な最適解を導出することが著しく困難となる場合が多い [6]．大規模で複雑な最適化問題に対しては，近似解を導出するメタヒューリスティクスが用いられることが多く，代表的なアルゴリズムのひとつに，生物の進化の仕組みを応用した進化的アルゴリズムがある．その中で最も一般的なものである遺伝的アルゴリズムは，ランダムに生成された個体集団に対し，優れた個体を選び，交叉や突然変異と呼ばれる遺伝的操作を繰り返すことで最適化問題を解くものであり，これまでに遺伝的アルゴリズムを用いた最適化の研究が様々に行われている [7,8]．遺伝的アルゴリズムは，その進化的プロセスが環境に対して遺伝物質を最適化するという概念に着想を得たアルゴリズムであり，環境が変化しても，状況に応じて適応することが可能である [9]．また，非常に大規模で複雑な最適化問題に対しても，適応度の高い個体集団を迅速に発見することができる [10]．分散映像分析システムは，さまざまな性能や特性を持った計算機や通信機器で構成されることが想定され，厳密な最適解の導出が現実的な時間で行うことが困難なことから，文献 [5] では遺伝的アルゴリズムを用いた解の導出を行っている．

本報告では文献 [5] で提案された分散映像分析システムの実装を行い，その省電力効果を実機を用いて評価する．文献 [5] では実際の機器を用いて映像分析を実施し，その際の消費電力を計測することで，端末，エッジサーバ，クラウドサーバの消費電力のモデル化を行い，遺伝的アルゴリズムを用いた最適化に利用している．しかしながら，あらゆる条件を想定した計測は困難であり，実際の映像分析を行う状況下では，モデルと実測値の間には誤差が発生する．このような誤差が何に起因して発生するのか，また，どのような分布として出現するのかを明らかにするとともに，誤差を改善する方法を検討することが本報告の目的である．

2　対象とする分散映像分析システムと消費電力最小化方式

2.1　分散映像分析システム

2.1.1　構成

本報告では文献 [5] において，提案されている分散型の映像分析システムを実装し，実計算機上における消費電力を評価することで省電力効果を示すとともに，文献における消費電力モデルと実測値の誤差について詳細な調査を行う．分散映像分析システムは，図 1 に示すように，分析対象の映像を複数のフレームに分割し，端末やエッジサーバー，クラウドサーバーに分散して分析する．端末，エッジサーバー，クラウドサーバーには性能差があり，端末，エッジサーバー，クラウドサーバーの順に高性能である．また，映像分析には CNN をベースとしたモデルが用いられ，その精度にも差があり，端末，エッジサーバー，クラウドサーバーの順に高精度な CNN モデルが用いられる．

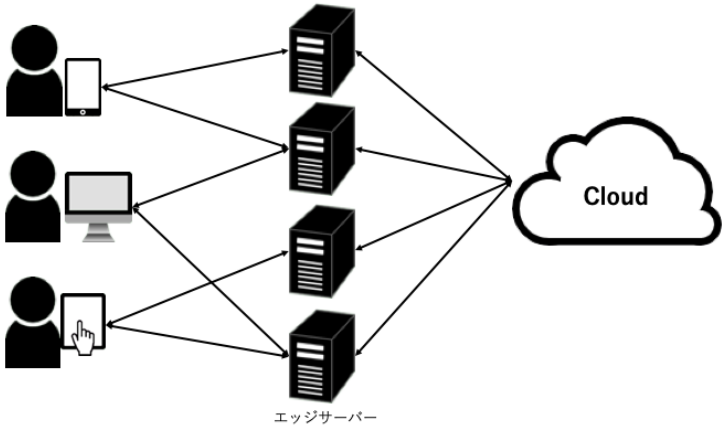

図 1: 想定するシステム構成

2.1.2　分散映像分析システムのモデル

本システムでは端末，エッジサーバー，クラウドサーバーが互いにネットワークで接続されており，端末に繋がったカメラで録画された映像がそれぞれの機器上で分析される．映像を複数のフレームに分割した際，その何割を端末が，残りをどのようにエッジサーバーとクラウドサーバーが分担するのか，それぞれの機器上でどの CNN モデルを利用するのか，この組み合わせを，複数のセッションを対象に同時に最適化する．文献 [5] では，分散映像分析システムの消費電力最小化方式を提案しており，この問題を組み合わせ最適化問題の形で

示している．文献では遺伝的アルゴリズムによって最適解の導出を行っており，本報告でも同様の解法を用いる．

この最適化問題では 2 つの制約条件を設定している．一つ目が，端末，エッジサーバー，クラウドサーバーでの映像分析の処理遅延の合計と，アクセスネットワーク，コアネットワークの伝送遅延の合計である，エンド・ツー・エンド遅延の制約であり，もう一つが端末，エッジサーバー，クラウドサーバーでの映像分析の認識精度をそれぞれの処理割合で加重平均した認識精度の制約である．この制約条件を満たす中で，映像分析を行うシステムの消費電力を最小化する．映像分析を行うシステムの消費電力は端末，エッジサーバー，クラウドサーバー，アクセスネットワーク，コアネットワークの消費電力の合計であり，これを最小化するように端末，エッジサーバー，クラウドサーバーでの処理割合が最適化される．

2.2 消費電力モデル

文献 [5] では，表 1 に示す機器を想定して以降のモデル化を行っている．

表 1: 想定する機器

Device	GWS-i9/4G
CPU	Core i9-10940X
CPU TDP	165 W
GPU	Nvidia RTX A5000
GPU FP32	27.77 Tflops
GPU TDP	230 W
アイドル時消費電力	98.27W

文献 [5] において映像分析には [11] の Yolo-v3 を用いている．また，CNN モデルは表 4 に示す 3 つの学習済みモデルを使用している．また，評価用映像および正解ラベルには， [12] の “Object Detection in Video Segments-validationset” のデータセットから表 2 の映像を使用している．データセットには映像の一部のシーンについて 1 秒毎に 1 件の正解ラベルが付与されており，当該のフレームのみ認識精度 の計算を行い，平均値 (mAP) を記録することで映像分析の認識精度の実測を行う．

表 2: 用いた映像

検出対象	映像時間	URL	アスペクト比
person	3:51	`https://www.youtube.com/watch?v=AJbQP-rIwCY`	16:9

文献 [5] では表 1 の機器と表 4 の CNN モデルを用いて映像セッション数およびフレーム処理割合をさまざまに変化させて映像分析を行い，GPU 使用率，処理時間，認識精度 (mAP)，消費電力を実測している．これらの実測値をもとに，GPU 負荷率を推定する．GPU 負荷率 $L^d(t)$ は式 (1) のように推定される．

$$L^d(t) = \frac{\sum_{s \in S^d} O_s^d(t)}{C^d E^d} \tag{1}$$

S^d は機器 d を使用する映像セッションの集合，C^d を表 1 の FP32 処理性能，E^d は機器 d の処理効率である．前述の通り，実測値から回帰分析によって求めた値が表 3 で示されている．

表 3: GPU パラメーターの推定

	GWS-i9/4G
GPU efficiency (E^d)	0.48
Floating operations A (O^A)	8.1 B
Floating operations B (O^B)	0.4 B

また，時刻 t において映像セッション s が機器 d にかけている処理負荷 $O_s^d(t)$ は以下の式で定義されている．

$$O_s^d(t) = \big((O^{M_s^d(t)} + O^A)W_s^d(t) + O^B\big)FPS \tag{2}$$

式 (2) において映像セッション s が機器 D で選択している CNN モデルの計算量 $O^{M_s^d(t)}$ は表 4 の Floating operations の値を用いている．FPS は 1 秒あたりのフレーム数で文献 [5] では 30fps のサンプル映像が用いられている．O^A および O^B は計算負荷の係数であり，実測値から求められたものが表 3 に示されている．

文献 [5] では表 3 をもとに 1 フレームあたりの処理時間 $T_s^D(t)$ を式 (3) によって推定している．

$$T_s^D(t) = \frac{O^{M_s^d(t)} + O^A + O^B}{C^d E^d L^d(t)} \tag{3}$$

また，機器 d について時刻 t における消費電力 $P^d(t)$ は式 (4) によって推定される．

$$P^d(t) = P_{IDLE}^d + P_{CPU-TDP}^d \alpha^p + P_{GPU-TDP}^d L^d(t) \beta^p \tag{4}$$

式 (4) において，P_{IDLE}^d はアイドル時消費電力，$P_{CPU-TDP}^d$ は CPU TDP，$P_{GPU-TDP}^d$ は GPU TDP であり，表 1 に記載した値が用いられている．また，CPU 負荷率は実測によって明確な値を求めることが困難であったと述べられており，すべての場合で共通の値を用いることとして，回帰分析によって係数 α^p を求めている．GWS-i9/4G における (α^p, β^p) は (0.97, 0.67) となっている．

また，文献 [5] ではネットワークおける伝送遅延のモデルが式 (5) のように定義されている．

$$T_s^n(t) = \frac{M\left(1 - \sum_{d \in D_{pass}^{s,n}} W_s^d(t)\right)}{B^n(t)} \tag{5}$$

同様に，ネットワークにおける消費電力モデルが式 (6) のように定義されている．

$$P^n(t) = \sum_{s \in S} M\left(1 - \sum_{d \in D_{pass}^{s,n}} W_s^d(t)\right) p^n \tag{6}$$

式 (6) において，M は 1 フレームあたりのビット数，$B^n(t)$ はネットワーク n の時刻 t での利用可能帯域，p^n はネットワーク n での 1 ビットの通信あたりの消費電力としている．文献 [5] では M を 470Kbit，p^n をアクセスネットワークとコアネットワークで 193×10^{-9}，60×10^{-9} としている．式 (5) と式 (6) において $D_{pass}^{s,n}$ は，映像セッション s がノード n に到達するまでに通過してきた機器の集合としている．

また，エンド・ツー・エンド遅延の制約について 1 フレームあたりのエンド・ツー・エンドでの処理時間の制約条件を式 (7) で定義する．

$$T_s = \sum_{d \in D^s} T_s^d(t) + \sum_{n \in (N^s)} T_s^n(t) \leq T_s^{max}, \quad \forall s \in S \tag{7}$$

式 (7) において S はすべての映像セッションの集合，D^s は映像セッション s が使用する機器（端末，エッジサーバー，クラウドサーバー）の集合，N^s は映像セッション s が使用するネットワーク（アクセスネットワーク，コアネットワーク）の集合である．T_s は映像セッション s のエンド・ツー・エンドの分析時間であり，その許容される最大値 T_s^{max} が制約として与えられる．$T_s^d(t)$ は時刻 t における映像セッション s の機器 d での 1 フレーム分の映像分析時間，$T_s^n(t)$ は時刻 t におけるネットワーク n での 1 フレーム分の転送遅延とする．

また，映像分析精度の制約については，システムを構成する各機器で行なった映像分析の精度の加重平均値に対して与えられ，式 (8) で定義される．

$$A_s = \frac{\sum_{d \in D^s} A^{M_s^d(t)} W_s^d(t)}{|D^s|} \leq A_s^{min}, \quad \forall s \in S \tag{8}$$

式 (8) において A_s は映像セッション s の映像認識精度であり，その許容される最小値 A_s^{min} が制約として与えられる．$M_s^d(t)$ は映像セッション s が機器 d で選択している CNN モデルとし，その CNN モデルの精度は以下の表 4 の mAP の値を用いる． $W_s^d(t)$ は映像セッション s が機器 D で処理している映像フレームの処理割合とし，$|D^s|$ は映像セッション s が使用するデバイスの総数である．

表 4: CNN モデル

Model	mAP	Floating operations
Yolov3-tiny	33.1%	$5.6B$
Yolov3	55.3%	$65.9B$
Yolov3-spp	60.6%	$141.5B$

以上によってシステム全体の消費電力 P はすべての映像セッションが使用するすべての機器 D^s とネットワーク N^s の消費電力の合計となる．$P^d(t)$ は機器 d の消費電力，$P^n(t)$ はネットワーク n の消費電力であり，式 (9) のように定式化されている．

$$P = \sum_{d \in D^s} P^d(t) + \sum_{n \in N^s} P^n(t) \tag{9}$$

式 (9) は各映像セッションのそれぞれのデバイスでのフレーム処理割合 $(W_s^d(t))$ と CNN モデル $(M_s^d(t))$ の関数となっている．本報告では式 (7) と式 (8) を共に満たし，式 (9) を最小化するようにフレーム処理割合と CNN モデルを決定するために遺伝的アルゴリズムを用いる．遺伝的アルゴリズムの出力した解によって，機器に割り当てるフレーム処理割合と CNN モデルが決定する．

2.3 消費電力最小化手法

文献 [5] では遺伝的アルゴリズムを用いて最適解を探索し，各機器での映像分析処理タスクの処理割合と各機器で使用する CNN モデルを決定している．遺伝的アルゴリズムにおいて，各個体の持つ遺伝子を，全映像セッションに対する各機器でのフレーム処理割合と CNN モデルの選択の組 $[W_s^d(t)\ \forall s, \forall d \in D^s$,$M_s^d(t)\ \forall s, \forall d \in D^s]$ と定義している．このような遺伝子を持つ個体の適合度 F は式 (10) のように求めている．

$$F = -P + \sum_{s \in S} (T_s^{max} - T_s)\alpha^f + \sum_{s \in S} (A_s - A_s^{min})\beta^f \tag{10}$$

式 (10) で表現される適合度 F はシステム全体の消費電力 P が小さいほど，映像分析に要する処理時間 T_s がその上限より小さいほど，映像分析の認識精度 A_s がその下限より大きいほど大きな値となる．ここで，本報告における実装では，端末，エッジサーバー，クラウドサーバーそれぞれで固定の CNN モデルを用いるため，用いてはいけない CNN モデルを選択した場合には適合度 F が小さくなるように設定する必要がある．P はシステム全体の消費電力のモデルであり，T_s は映像分析に要する処理時間，A_s は映像分析の認識精度のモデ

ルである．α^f と β^f は制約条件に関するハイパーパラメーターであり，消費電力よりも制約条件の充足を重視するために，これらは非常に大きな値を設定するものとしている．遺伝的アルゴリズムを用いた最適化では以下の 1~5 の手順を繰り返すことで最適化を行う．

1. 映像分析システムを構成する全ての要素の状態をシステムの状態としてその状態を測定し，これを環境条件として与える．ここではネットワークの回線速度，帯域が相当する
2. 1. を環境条件の下で各遺伝子個体の適合度（式 (10)）を計算する．ここで，初期個体数は 500 と設定されている
3. 集団の中から適合度の高い個体を抜き出し，それらとそれらの子からなる新しい集団を形成する．このときの遺伝操作として，2 点交差を採用している
4. 子を生成する際には，一部の遺伝子を突然変異させる．突然変異率は 0.2 が設定されている
5. 最も適合度の高い個体の遺伝子の情報によって各映像セッションの端末，エッジサーバー，クラウドサーバーにおける処理割合と CNN モデルを決定する

3　分散映像分析システムの実装

3.1　機器構成

実装するシステムは端末 1 台，エッジサーバー 2 台 (エッジサーバー A，エッジサーバー B)，クラウドサーバー 1 台で構成する (図 2)．本報告では表 6 の機器を使用して消費電力の実測を行った．

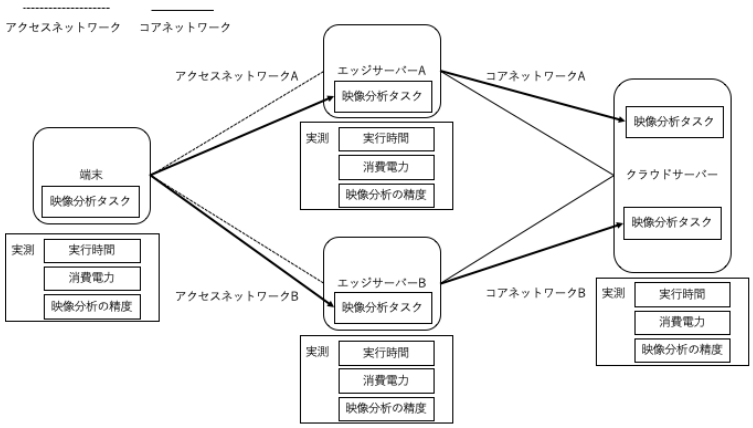

図 2: 評価用のシステム構成

なお，システムを構成するすべての機器について実測値から回帰分析を行なって機器のモデルを構築するためには，測定時に考慮すべき変数の種類・値の組み合わせが多数であり，本実装ではいずれも同一の機器を用いることとした．ただし，端末・エッジサーバー・クラウドサーバーの性能の差を考慮して，以下の図 3 のように仮想的にシステム性能に差が生じるようにした．このシステムを用いて消費電力最小化の効果の評価を行うこととする．アクセスネットワークとコアネットワークにはそれぞれ無線ネットワークと有線ネットワークによる接続を行うこととし，その消費電力値はトラフィック量に応じて決定するモデルを利用する．

端末は映像分析のタスクを割り当てて消費電力を実測した計算機 (表 6) 1 台分，エッジサーバー A は 2 台分，エッジサーバー B は 3 台分，クラウドサーバーは 5 台分の処理性能を持つ計算機であると仮定し，各端末に割り当てられた映像分析のタスクの要求が到着した際に，それぞれの機器に割り当てるフレーム処理割合を決定し，消費電力の評価を行う．また，端末，エッジサーバー，クラウドサーバーで映像分析に使用する CNN モデルは表 5 の通りである．

表 6 で示す計算機上に Yolo-v3 を導入し，分析用映像を保存して，割り当てられた映像分析のフレーム処理割合に応じた映像の分析を行うスクリプトを実行することで，その際の消

表 5: 映像分析に用いる CNN モデル

	CNN モデル
端末	Yolov3-tiny
エッジサーバー A	Yolov3
エッジサーバー B	Yolov3
クラウドサーバー	Yolov3-spp

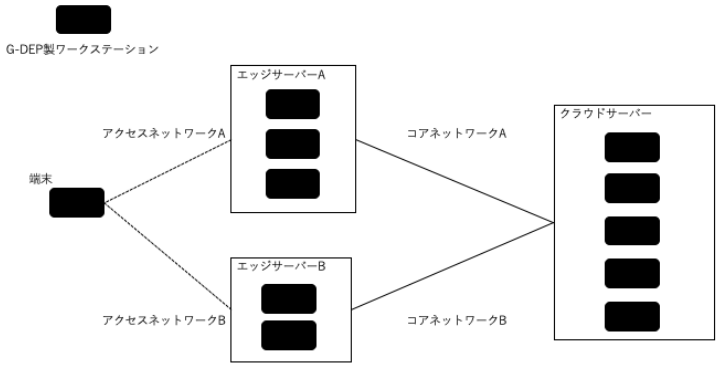

図 3: 評価用の仮想的なシステム構成

費電力，遅延時間，認識精度が得られる．

表 6: 評価に用いた計算機

	G-DEP 製ワークステーション GWS-i9/4G
CPU	Intel Corei9 10940X
RAM	128 GB (DDR4)
GPU	NVIDIA RTX A5000
OS	Ubuntu 20.04 LTS

表 6 の機器のモデルについては，2.2 において説明したものと同様であり，これは映像分析タスクのフレーム処理割合をさまざまに変更させた上で，CPU 利用率，GPU 利用率，分析に要した時間，消費電力量を実測し，線形回帰分析によってモデルパラメータを決定している．映像分析の処理時間のモデルも同様に 2.2 のものが用いられている．分析する映像データセットについては 2.2 と同様のものを使用している．各機器の映像分析タスクのフレーム処理割合については 2.3 に基づいてフレーム処理割合が決定される．

3.2 消費電力の計測方法

図 2 で示した映像分析システムの消費電力は，各機器に映像分析が割り当てられた際の個別の消費電力を実測し，その後，式 (9) を用いることで得られる．エッジサーバー A，エッジサーバー B，クラウドサーバーについてはそれぞれテストベンチを実行する計算機 3 台分，2 台分，5 台分の処理性能をもつものと想定しており，これについては，実測時の映像分析タスクのフレーム処理割合をそれぞれ実際に割り当てられた量の $\frac{1}{3}$ 倍，$\frac{1}{2}$ 倍，$\frac{1}{5}$ 倍とすることで近似する．なお，シミュレーションプログラムから出力されるフレーム処理割合は各映像セッションに対して定められているが，今回使用したテストベンチでは GPU のメモリの都合上，各映像セッションに対してフレーム処理割合を決定することができなかったため，近似的な値を導出する．すなわち，映像分析において機器にかかる負荷は分析する映像のフレーム数に比例すると仮定し，テストベンチにおけるフレーム処理割合は各映像セッションのフレーム処理割合を平均したものと設定した．また，ネットワークで消費する電力については 2.2 でモデル化されたものから，遺伝的アルゴリズムで出力された値を評価時に使用することとした．

3.3 遅延の計測方法

図 2 で示した映像分析システムの遅延については，各機器における映像分析時の処理遅延を実測した値を用いて導出する．式 (3) に実測値を与えることで各映像セッションの遅延が得られる．また，ネットワークの伝送遅延については 2.2 でのモデルを用いることとし，遺伝的アルゴリズムの計算時に導出された値を評価時に使用する．

3.4 認識精度の計算方法

図 2 で示した映像分析システムの認識精度については，各機器の映像分析時の認識精度を実測した値を，式 (8) に与えることで導出する．

4　分散映像分析システムの評価と考察

4.1　消費電力モデルによる予測値と実測値の比較

4.1.1　映像セッションの構成と制約条件

評価をおこなう際のセッションの設定を表 7 のように定めた．4 つの映像セッションを持ち，セッション 0 とセッション 1 はエッジサーバーとしてエッジサーバー A を，セッション 2 とセッション 3 はエッジサーバー B を用いる．また，セッション 0 とセッション 1 ではアクセスネットワークとしてアクセスネットワーク A，コアネットワークとしてコアネットワーク A，セッション 2 とセッション 3 ではアクセスネットワークとしてアクセスネットワーク B，コアネットワークとしてコアネットワーク B を用いる．この際，端末とクラウドサーバーは 4 つの映像セッション全てにおいて同じ機器を用いる．

各映像セッションについてエンド・ツー・エンド遅延制約と認識精度制約を表 8 のように設定した．以上の設定において，エンド・ツー・エンド遅延制約と認識精度制約を満たした上で消費電力を最小化する組み合わせ最適化問題を遺伝的アルゴリズムにより解き，各機器に割り当てる処理の割合，CNN モデルの決定を行う．

表 7: 各セッションの構成機器と制約条件

	端末の機器	エッジサーバーの機器	クラウドサーバーの機器	アクセスネットワーク	コアネットワーク
セッション 0	端末	エッジサーバー A	クラウドサーバー	アクセスネットワーク A	コアネットワーク A
セッション 1	端末	エッジサーバー A	クラウドサーバー	アクセスネットワーク A	コアネットワーク A
セッション 2	端末	エッジサーバー B	クラウドサーバー	アクセスネットワーク B	コアネットワーク B
セッション 3	端末	エッジサーバー B	クラウドサーバー	アクセスネットワーク B	コアネットワーク B

表 8: 制約条件

	エンド・ツー・エンド遅延制約	認識精度制約
セッション 0	0.134 s	0.51
セッション 1	0.045 s	0.36
セッション 2	0.154 s	0.53
セッション 3	0.065 s	0.48

4.1.2 遺伝的アルゴリズムにより得られた解

遺伝的アルゴリズムを実行した結果，得られた最適解により，各機器へ割り当てる映像分析タスクのフレーム処理割合は表 9 のように決定された．本報告の実測に使用した計算機は処理性能が高く，分析を行う映像の多くを端末で分析できるため，エッジサーバーには映像分析タスクが割り当てられたが，クラウドサーバーには映像分析タスクが割り当てられない結果となった．

表 9: 各機器の映像分析タスクの処理割合

	端末	エッジサーバー A	エッジサーバー B	クラウドサーバー
セッション 0	20%	80%	-	0%
セッション 1	86%	14%	-	0%
セッション 2	0%	-	96%	0%
セッション 3	32%	-	68%	0%

表 9 のフレーム処理割合をもとに 3.2 で記述した手順で，実際に機器に割り当てる映像分析タスクのフレーム処理割合を決定した．表 10 がその割合である．

表 10: テストベンチ実行時に機器に割り当てる処理割合

	処理割合
端末	35%
エッジサーバー A	16%
エッジサーバー B	41%
クラウドサーバー	0%

端末の各映像セッションのフレーム処理割合はセッション 0 で 20%，セッション 1 で 86%，セッション 2 で 0%，セッション 3 で 32%であり，4 つの映像セッションのフレーム処理割合の平均は 35%であった．エッジサーバー A の各映像セッションのフレーム処理割合はセッション 0 で 80%，セッション 1 で 14%であり，2 つの映像セッションのフレーム処理割合の平均は 48%となり，エッジサーバー A は表 6 の機器 3 台分の処理性能を持つと仮定しているため，フレーム処理割合を $\frac{1}{3}$ 倍した 16%が割り当てられるフレーム処理割合となる．同様に，エッジサーバー B の各映像セッションのフレーム処理割合はセッション 2 で 96%，セッション 1 で 68%であり，2 つの映像セッションのフレーム処理割合の平均は 82%となり，エッジサーバー B は表 6 の機器 2 台分の処理性能を持つと仮定しているため，フレーム処理割合を $\frac{1}{2}$ 倍した 41%が割り当てられる．

4.1.3 消費電力モデルを用いた予測値の結果

ここまでの節で得られた結果をもとに式 (4) を用いて消費電力が導出される．各機器と各ネットワークでの消費電力の予測値は表 11 となった．

表 11: 消費電力の予測値

	消費電力
端末	336.41W
エッジサーバー A	1102.18W
エッジサーバー B	735.83W
クラウドサーバー	0W
アクセスネットワーク A	2.57 W
アクセスネットワーク B	4.57 W
コアネットワーク A	0 W
コアネットワーク B	0.035 W
システム全体	2181.59W

また，遺伝的アルゴリズムの出力として得られる各映像セッションのエンド・ツー・エンド遅延と認識精度の予測値は表 12 に示す結果となった．

表 12: エンド・ツー・エンド遅延と認識精度の予測値

	エンド・ツー・エンド遅延	認識精度
セッション 0	0.039 s	0.51
セッション 1	0.033 s	0.36
セッション 2	0.040 s	0.53
セッション 3	0.041 s	0.48

4.1.4 計算機を用いた実測値の結果

表 6 の機器を用いて，各機器に与えた映像分析タスクを実施した際の消費電力は表 13 に示す通りである．

この際の各映像セッションの遅延時間と認識精度の実測値は表 14 に示す通りである．

図 2 のシステムを構成する端末，エッジサーバー，クラウドサーバーにランダムに映像分析タスクを割り当てたとき，システム全体の消費電力は 2968.84W となり，本システムを適用することで消費電力を 24.5%削減できた．

表 13: 消費電力の実測値

	消費電力
端末	376.37 W
エッジサーバー A	1116.81 W
エッジサーバー B	741.88 W
クラウドサーバー	0 W
アクセスネットワーク A	2.57 W
アクセスネットワーク B	4.57 W
コアネットワーク A	0 W
コアネットワーク B	0.035 W
システム全体	2242.24 W

表 14: エンド・ツー・エンド遅延と認識精度の実測値

	エンド・ツー・エンド遅延	認識精度
セッション 0	0.034 s	0.74
セッション 1	0.034 s	0.42
セッション 2	0.030 s	0.78
セッション 3	0.034 s	0.73

4.2 消費電力モデルの予測値と実測値の誤差の検証

消費電力の実測値と予測値の間には誤差が生じていることが確認された．システムのモデル化の際には仮定や近似が行われるため，誤差の発生を避けることは困難である．誤差が大きいと，理論的に最適な解を現実の機器に割り当てた際の最適性を確保することが困難になる．そのため，誤差がどのような原因で生じるのか，どのような分布で生じるのかを明らかにし，誤差を改善することが望ましい．本報告では実測値と予測値の誤差を検証し，誤差低減のための消費電力予測モデルの改善を行う．以降では，表 6 の機器を用いて映像分析タスクを割り当て，式 (4) で表される消費電力予測モデルと 3.2 で実測した消費電力を比較する．

4.2.1 分散映像分析システム全体での消費電力の誤差

図 2 に示したシステムに映像分析タスクを割り当て消費電力の実測を行った際に，消費電力の実測値と予測モデルから算出した予測値には，各機器個別・システム全体において表 15 に示すような誤差が生じた．表 15 からは，端末での消費電力において誤差が大きいことが確認でき，このような誤差の大小は映像分析に使用した CNN モデルの種類や映像セッション数の違いによって生じていると考えられる．そこで 4.2.2 において検証実験を行い，どのような場合に誤差が大きくなるのか，その分布を調査した．

表 15: 実測値と予測値の誤差

	誤差
端末	39.97 W
エッジサーバー A	14.63 W
エッジサーバー B	6.04 W
システム全体	60.64 W

4.2.2 単一の機器の消費電力の誤差

表 4 に示す Yolo-v3 の各モデルを 3 通り，フレーム処理割合を 10%~100%の 10 通り，セッション数を 1,2,4 の 3 通りの各組み合わせ 90 パターンでの映像分析を行い，その際の消費電力を実測した．この実測を 5 回繰り返し，合計 450 個の消費電力の実測値データを取得し，消費電力予測モデルから算出される予測値との誤差を導出した．また，映像分析時に消費する電力の多くは GPU で消費されていることが想定されるため，使用している機器に搭載されている GPU 管理用コマンド（nvidia-smi コマンド）を実行して，映像分析時の GPU 消費電力も同時に計測した．GPU における消費電力が，機器における消費電力において大き

な割合を占めていると考えられるため，図 4 に示すように，横軸を GPU の消費電力とし，縦軸に計算機の消費電力の実測値を設定した散布図を作成した．

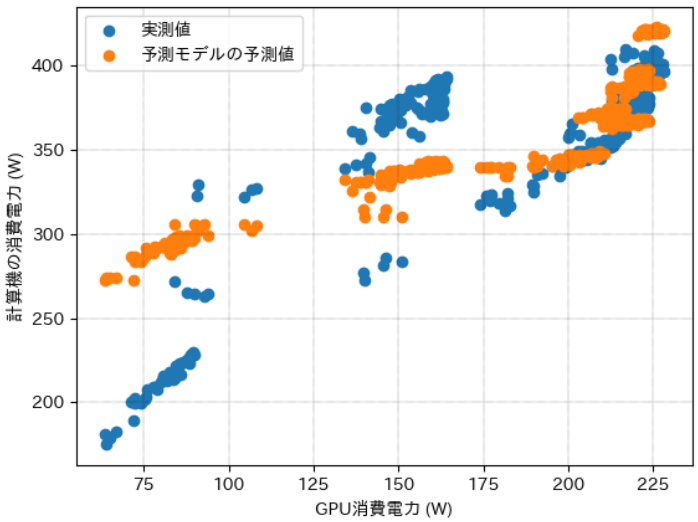

図 4: GPU 消費電力と消費電力の実測値，予測値の関係

図 4 より GPU 消費電力が小さい時には，予測モデルから算出される消費電力の予測値と消費電力の実測値の誤差が大きくなっていることが確認できる．これは映像分析時に GPU にかかる負荷が小さく，GPU で消費される電力に比べて，CPU で消費される電力が大きいことで生じている誤差であると考えられる．また，実測値について，単純に考えると GPU 消費電力と計算機の消費電力は線形（一次関数）の形で表されると考えられるが，GPU 消費電力が 150W となる周辺において非線形な変化が生じていることがわかる．GPU 消費電力が 150W に近くなるのは，映像分析に使用した CNN モデルが Yolov3-tiny，映像セッション数が 2 と 4 の場合であり，このときの記録から，GPU 利用率に比べて CPU 利用率が高くなっていることが確認された．すなわち映像分析に使用した CNN モデルが Yolov3-tiny，映像セッション数が 2 と 4 の場合は 1 フレームの映像分析に要する遅延に対して CPU が行うプロセスに要する時間が大きくなり，GPU の利用効率が悪くなるために計算機で消費される電力の多くが CPU で消費されていると考えられる．

4.3　誤差を低減するための予測モデルの改善

4.2 で行った誤差の検証において，GPU の消費電力に着目すると，137W および 164W を境界として，消費電力の傾向が変化していることが確認できた．これらが普遍的な傾向であ

るかについては確認ができないものの，GPU 消費電力と実測値との分布に対するクラスタリング分析により，モデルの傾向を大別することが有効となる可能性がある．そこで，本報告では簡易なモデルとして，GPU 消費電力が 137W∼164W の場合とその他の場合に分けた予測モデルを構築する．

$$P^d(t) = \begin{cases} P^d_{GPU}(t)\gamma^p + P^d_{fixed_1} & \left(137 \leq P^d_{GPU}(t) \leq 164\right) \\ P^d_{GPU}(t)\delta^p + P^d_{fixed_2} & \left(else\right) \end{cases} \tag{11}$$

式 (11) が新たな消費電力予測モデルである．この予測モデルは GPU 消費電力の実測値を説明変数，消費電力の実測値を目的変数として GPU 消費電力が 137W∼164W の場合とその他の場合に分けて回帰分析を行い，係数 γ^p と δ^p，GPU 消費電力に依らない固定の消費電力 $P^d_{fixed_1}$，$P^d_{fixed_2}$ を定めている．(γ^p, δ^p) は (2.04, 1.16) となり，($P^d_{fixed_1}$， $P^d_{fixed_2}$) は (55.81, 125.60) となる．

4.4 改善した消費電力予測モデルと実測値の誤差の検証

ここでは 4.3 で説明した消費電力予測モデルと消費電力の実測値との誤差を評価した．表 4 に示す Yolo-v3 の各モデルを 3 通り，処理割合を 10%∼100%の 10 通り，セッション数を 1,2,4 の 3 通りの各組み合わせ 90 パターンで映像解析を行って消費電力を実測した．この実測を 5 回繰り返して合計 450 個の消費電力の実測値データを取得し，実測値との誤差を調査した．図 5 は横軸が gpu の消費電力，縦軸が計算機の消費電力の散布図である．

式 (4) で表される消費電力予測モデルと比べて，GPU 消費電力が低い場合の計算機の消費電力の誤差が低減されていることがわかる．また，GPU 消費電力が 137W∼164W の場合の計算機の消費電力も誤差が低下している．消費電力の実測値と式 (4) と式 (11) で表される消費電力予測モデルの予測値との平均二乗誤差を計算した結果，式 (4) で表される消費電力予測モデルの予測値では 35.19W であったのに対し，式 (11) で表される消費電力予測モデルの予測値では 17.44W となった．また，図 2 のシステムの消費電力の予測値と実測値の誤差は表 16 に示す通りである．すべての機器において誤差が低減されており，システム全体としての消費電力の誤差は 70.8%低減された．特に端末の機器で大きく誤差が低減されており，62.7%低減することができた．

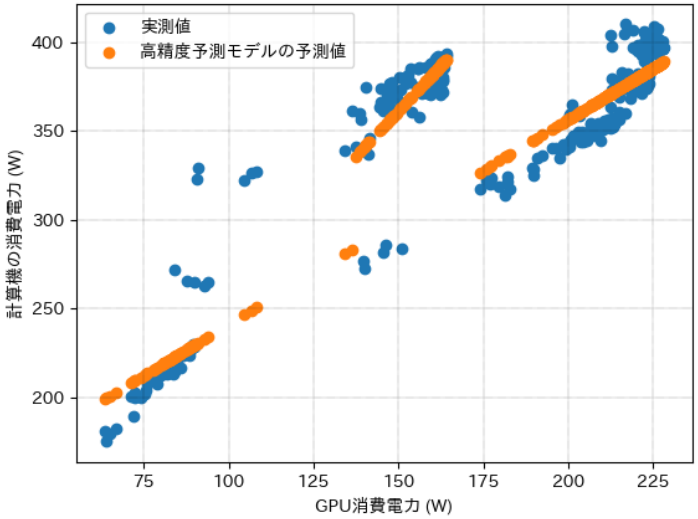

図 5: GPU 消費電力と消費電力の実測値，新たな予測モデルの予測値の関係

表 16: 実測値と新たなモデルの予測値の誤差

	誤差
端末	14.90 W
エッジサーバー A	8.13 W
エッジサーバー B	5.35 W
システム全体	17.69 W

5　おわりに

近年，精度の高い映像データ処理が必要とされるサービスの普及に伴い，映像分析に要する消費電力を削減することが課題となっている．本報告ではエッジ・クラウドコンピューティング技術を用いた分散映像分析システムを構築し消費電力の最小化方式を実装したシステムに適用し，評価した．消費電力の最小化方式を適用することで実機を用いたシステムにおいて消費電力を 24.5%削減できることを示した．また，消費電力の最小化を行う上で用いられている関連研究の消費電力の予測モデルから算出される予測値と消費電力の実測値の間で生じる誤差を検証し，高精度に予測を行うために予測モデルを改善した．関連研究の消費電力の予測モデルでは実測値との平均絶対パーセント誤差が 9.3%であったが，改善を行った予測モデルでは実測値との平均絶対パーセント誤差を 3.5%に削減した．

本報告では機器のモデルがすでに構築されている計算機を用いた仮想的なシステムに対して消費電力の最小化方式を適用した．今後は，実アプリケーションを想定した上で端末，エッジサーバー，クラウドサーバーを選定して実システムを構築し最小化方式を適用させてその評価を行う必要がある．映像分析に要する消費電力には GPU の消費電力以外に CPU の消費電力が大きく関わっていることが確認できた．CPU 消費電力と GPU 消費電力を実測し，消費電力の予測モデルに反映することでより高精度な予測が可能であると考える．また，現実的にはシステムを構成する計算機の特性や性質はさまざまに異なるため，モデルの構築自体が困難な場合も考えられる．そのような場合のために，オンラインでモデルを構築していく方法の検討も重要である．

謝辞

本報告の遂行にあたり，終始多大かつ貴重なご指導を賜りました，大阪大学大学院情報科学研究科の村田正幸教授に心より深く感謝いたします．また，平素から丁寧にご指導くださいました，大阪大学大学院情報科学研究科の小南大智助教，多大なるご助言をくださいました，大阪大学サイバーメディアセンターの下西英之教授に深く感謝いたします．最後に日頃から支えてくださった家族，友人，研究室の皆様に感謝の意を表して謝辞といたします．

参考文献

[1] Gerhard Fettweis, Holger Boche, Thomas Wiegand, et al.: “The Tactile Internet”, ITU-T Technology Watch Report (2014).

[2] K. Rao, G. Coviello, W.-P. Hsiung and S. Chakradhar: “Eco: Edge-cloud optimization of 5g applications”, 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)IEEE, pp. 649–659 (2021).

[3] Andres S. G. Andrae: “Prediction Studies of Electricity Use of Global Computing in 2030”, International Journal of Science and Engineering Investigations, **8**, (2019).

[4] A. S. G. Andrae and T. Edler: “On Global Electricity Usage of Communication Technology: Trends to 2030”, Challenges, **6**, 1, pp. 117–157 (2015).

[5] 下西　英之, 村田　正幸, 長谷川　剛：“分散映像分析システムの消費電力最適化方式の検討”, 信学技報, **122**, 275, pp. 28–33 (2022).

[6] 田中俊二：“大規模組合せ最適化問題に対する数理アプローチの基礎”, 計測と制御, **56**, 12, pp. 967–972 (2017).

[7] M. S. Hoque: “An implementation of intrusion detection system using genetic algorithm”, International Journal of Network Security & Its Applications.

[8] M. Bielli, M. Caramia and P. Carotenuto: “Genetic algorithms in bus network optimization”, Transportation Research Part C: Emerging Technologies, **10**, 1, pp. 19–34 (2002).

[9] A. Van Soest and L. Casius: “The merits of a parallel genetic algorithm in solving hard optimization problems”, Journal of biomechanical engineering, **125**, 1, pp. 141–146 (2003).

[10] K. De Jong: “Learning with genetic algorithms: An overview”, Machine learning, **3**, 2, pp. 121–138 (1988).

[11] J. Redmon and A. Farhadi: “Yolov3: An incremental improvement”, arXiv preprint arXiv:1804.02767 (2018).

[12] "YouTube Bounding Boxes", https://research.google.com/youtube-bb/download.html.