Vol.2020-OS-148 No.7
2020/2/28

AI 解析向け動画像符号化方式

中尾鷹詔 [1,a)]　久保田智規 [1]　吉田英司 [1]

概要：近年，動画データの増大が著しく，どのようにサーバやクラウドに転送・蓄積するかが課題となっている．また，ディープラーニング技術の進化に伴い AI による画像解析の需要が高まっている．そこで，我々は動画像を解析可能かつより小さなデータサイズに符号化する手法を研究している．本稿では物体検出手法の一つである YOLOv3 を対象として，AI が物体を検出するのに必要な領域を解析し，不要な領域のみ画質を劣化させる高効率な動画像符号化方式を提案する．本手法を用いることで，人が解析することを想定した符号化方式と比べて圧縮率と検出率が向上することを確認した．

キーワード：物体検出，動画圧縮，深層学習

1. はじめに

近年のデジタルデータの増大は著しく，IDC によると 2025 年には 175ZB になると言われている[1]．それに伴って，必要となるデータ転送帯域やストレージ容量が増大し，通信や蓄積に要する費用は増大する[2]．また，デジタルデータの大部分は動画データと言われており，IT ベンダーはいかに動画を小さなサイズに圧縮するかに注力してきた．例えば 1990 年に登場した動画コーデックである H.261 は順当に進化を続け，世代を追うごとに符号化された動画データのサイズを半分にすることに成功している[3]．最近ではライセンスフリーを特徴とした AV1（Alliance for Open Media Video 1）も登場し H.266 と次世代の標準を狙って争っている[4]．また，増え続ける動画を人が目視で解析するには限界があり，AI（Artificial Intelligence）を利用した動画処理の需要が拡大し AI 解析技術も発達している．例えば，AI を用いて動画像のどこに何が映っているかを検出する技術や[5]，映っている人物の姿勢を検出する技術が公開されている[6]．動画データのサイズを小さくする手法として，画素値の量子化粒度を粗くする方法がある．ただし量子化粒度を粗くすると，画質が劣化して AI による解析精度も悪化する．本稿では動画のデータサイズの縮小と解析精度を両立するために，解析に必要な領域のみ高画質で記録する AI に特化した圧縮方式を提案する．なお，本稿で扱う AI とは深層学習を用いた動画の解析技術になる．

必要な部分だけ高画質にすることで動画サイズを削減する手法は ROI（Region of Interest）圧縮と言われて研究が進んでおり[7]，抽出した人の顔部分のみ高画質にするものや[8]，医療の分野への適用として診断に関わる領域を高画質に記録するもの等様々な分野に応用されている[9]．これらの技術は人が違和感なく着目する対象を認識できる動画像を目指している．しかし人の注目する領域は主観が混じり曖昧になる為，ROI を広めに取って記録することになり，動画データのサイズ削減には限界がある．一方で AI の場合は，同じ動画でも人の注目する領域とは違い，解析する内容によって参照する領域が限定される[10]．その特徴を利用して AI に必要十分な特徴を残すように細かく画質を調整することで，圧縮率を上げてデータサイズを小さくすることが出来る．

本稿では，検出精度を維持しつつ動画全体の圧縮率を人手で解析する目的で符号化したものより高くする手法を提案する．本稿の構成を以下に示す．第二章で提案手法に関連する ROI を用いた圧縮と XAI（eXplainable Artificial Intelligence）について記述し，第三章で提案手法について説明する．第四章で提案手法の効果を示す実験結果を述べる．第五章にまとめと今後の課題を示す．

なお，本稿の図に出てくる画像はハイビジョン・システム評価用標準動画像第二版[11]を利用した．

2. 関連研究

提案する方式の主な技術要素は，動画像を注目領域のみ高画質で符号化する ROI を用いた符号化方式と，AI の動作を解析する技術である XAI の二つである．以下にそれらの動向を概観する．

2.1 ROI 圧縮

ROI 圧縮とは必要な部分だけ高画質にすることで動画サイズを削減する技術である．ROI を検出する技術の一つに，あらかじめ対象となる物体の特徴を定めておいてそれと一致する領域を探す方法がある．例えば人の顔を検出する時には肌の色を探し出してそこを ROI とする[8]．また，Grad-CAM 等の深層学習に利用される CNN（Convolutional Neural Network）の注目箇所を求める技術を利用する場合もある[12]．いずれも目視による解析を想定した技術であり人の注目領域は主観によって変動するため，注目領域を広めに指定することになる．これに対して AI の解析を想定する場合は注目領域を厳密に特定できるため，さらに動画データのサイズを削減する余地があるといえる．

1　(株)富士通研究所
Fujitsu Laboratories Ltd.
a)　nakao.takanori@fujitsu.com

2.2 XAI

AI の動作を解析する技術として XAI が研究されている[13]．本稿で扱う XAI は，深層学習による物体検出において，入力画像の画質劣化に伴い物体の検出精度が劣化した時に，精度劣化の原因領域を求める技術になる．関連する研究としては推論モデルが物体を高い精度で検出できない画像において，原因領域を画素粒度で抽出・可視化する技術が提案されている[10]．[10]の手法は Activation Maximization でスコア最大化した画像を用意して，推論モデルによる物体検出に対して Backpropagation や Grad-CAM 等を用いることでおおよその原因領域を求めた後に，原因領域をスコア最大化画像に差し替えた画像を評価して物体が検出できるかどうかを，原因領域を変えつつ繰り返し試すことで，より正確な原因領域を導出する．本稿では[10]の手法を応用して動画圧縮に利用することを目指す．

3. 提案手法

3.1 概要

深層学習による認識モデルが物体を検出するのに必要な領域を解析し，不要な領域のみ画質を劣化させることで高効率に符号化する手法を提案する．図 1 で示すように最初に動画像に対してフレーム毎に深層学習を用いて物体検出を行い，物体の有無によって動作を変える．物体が存在しないフレームは画素値の量子化粒度を最大限に粗くすることで，低画質になる代わりに高い圧縮率で符号化する．物体が存在するフレームは図 2 に示す流れに従って画素値の量子化粒度を調整して画質を変更する．

図 2 の処理概要を説明する．まず画像全体の圧縮率を上げるために量子化粒度を上げて画質を劣化させつつ物体が検出できるか確認する．検出できなくなった時点でその画像のどの領域を劣化させたことが原因で検出できなくなったのかを確認し，原因となった領域の量子化粒度を元に戻すことで検出できるようにする．これを繰り返すことで，物体を検出できる最低限のデータサイズにすることが出来る．以下，図 2 の流れに沿って詳しく説明する．

3.2 画像全体の圧縮率を上げる

最初に圧縮率を上げるために，入力画像全体の画素値の量子化粒度を，物体が検出できなくなるまで上げて劣化画像を作る．後述する評価では H.265/HEVC（High Efficiency Video Coding）を用いて符号化した動画のデータサイズを比較した．そのため量子化粒度は領域毎に指定できる QP（Quantization Parameter）と呼ばれる値で調節した．符号化する時に QP を大きくすると量子化粒度が上がり圧縮率が大きくなり画質が劣化して，反対に QP を小さくすると量子化粒度が下がり圧縮率が小さくなり画質が向上する．

3.3 重要度マップの作製

次に入力画像と劣化画像の差分を取得する．差分画像は量子化粒度を粗くした時の画像の変化を表し，変化が大き

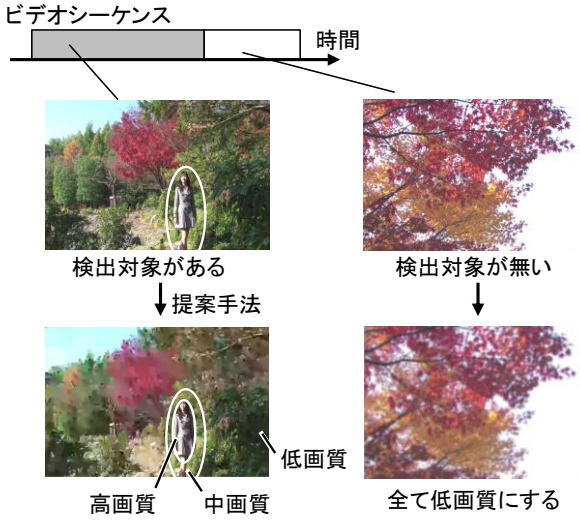

図 1 物体の有無による処理の違い

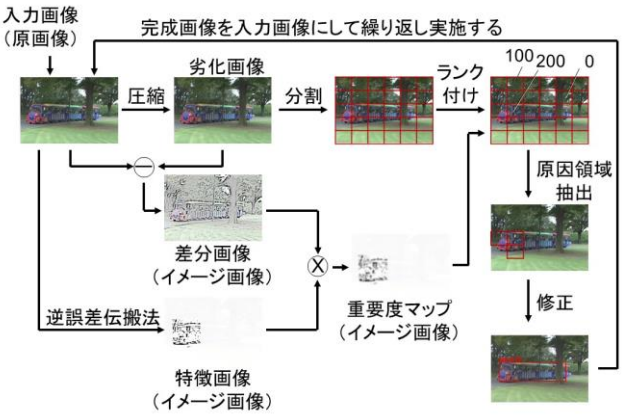

図 2 提案手法

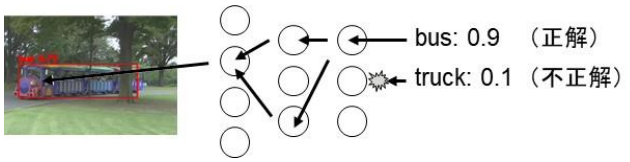

図 3 選択的逆誤差伝搬表

い領域ほど物体が検出できなくなった原因となりやすい．

また入力画像を解析した CNN に対して逆誤差伝搬法を適用して，対象物体を検出するために必要な特徴を求めて特徴画像とする．ただし，逆誤差伝搬法を単純に適用した場合は不正解クラスの特徴も求めてしまう．例えば図 2 ではバスの特徴を表す領域のみ欲しいが，注目する物体に該当しないトラックなどの異なる分類の特徴も含まれてしまう．そこで正解クラスの伝搬のみを行う選択的逆誤差伝搬法用いて，正解クラスの特徴領域のみを求めるようにする（図 3）．

3.4 領域毎に優先度のランク付け

差分画像と特徴画像を重畳することで，量子化粒度を粗くしたことにより大きく変化した領域であり，かつ対象物体を検出するために必要な特徴を表している領域を求める

ことができる．これを重要度マップと定義する．

その後，劣化画像の領域を適切な領域に分割する．この時，分割単位はエンコーダーが画質を変更できる単位を考慮して対応が取れるように決める必要がある．HEVC の場合は 8x8 pixel が最小単位となるため，本稿の評価では同様に 8x8 pixel を分割領域のサイズとした．

この分割した領域の優先度を，重要度マップを元に決定する．これは各領域に含まれるピクセルの重要度を合計することで算出することができる．

3.5 原因領域抽出

続いて領域の優先度を基準として検出できなくなった原因の領域を決定する．その基本は，物体検出できるまで優先度の高い領域から順に量子化粒度を入力画像と等しくして評価することで実現できる．しかし単に最小単位（ここでは 8x8 pixel）の領域を優先度順に差し替える方法では，フレーム毎に処理するには演算量が多い．また優先度順に選択しても必要最小限の領域を選択できるとは限らない．そこで高速化と精度向上のために図 4 のように実装した．

図 4 の上半分は候補領域を広げるステップで，8x8 pixel の領域を結合して 32x32 pixel 等の大きな領域を定義し，優先度の高い順に候補領域として追加する．そして候補領域の量子化粒度を入力画像と等しくして，それ以外の領域の量子化粒度を劣化画像のものにして推論エンジンにかける．対象とする物体が検出されなかったら検出されるまで候補領域の追加を行い，対象とする物体が検出されたら図 4 の下半分のステップに移る．

図 4 の下半分は候補領域を狭めるステップで，領域の分割サイズを最小単位の 8x8 pixel に変更し，優先度の逆順に候補領域を削る．候補領域を削るたびに，前のステップと同様に量子化粒度を調整したものを推論エンジンにかけ，対象の物体が検出されなかったら削った候補領域を元に戻し，検出されたら候補領域をそのままにして，どちらの場合も次の候補領域を削って改めて画像合成から繰り返す．一定回数繰り返したら候補領域を原因領域として終了する．

このように，候補領域を広げる時に大きなサイズを使うことで高速にして，領域を削る候補を再帰的に探すことで精度を向上した．

3.6 画像の修正

前項で劣化画像が物体を検出できなくなった原因領域が判明したため，劣化画像を基準に原因領域の量子化粒度を入力画像と等しく変更することで，物体を検出できる状態を維持したまま圧縮率を上げることが出来る．

3.7 全体の画質が収束するまで繰り返す

以上の，フレーム全体の量子化粒度を粗くしてから，物体検出できなくなった原因領域の量子化粒度を元に戻す処理を繰り返すことで，全ての領域が必要最小限のデータで表現され，動画データのサイズをより少なくすることが出来る．

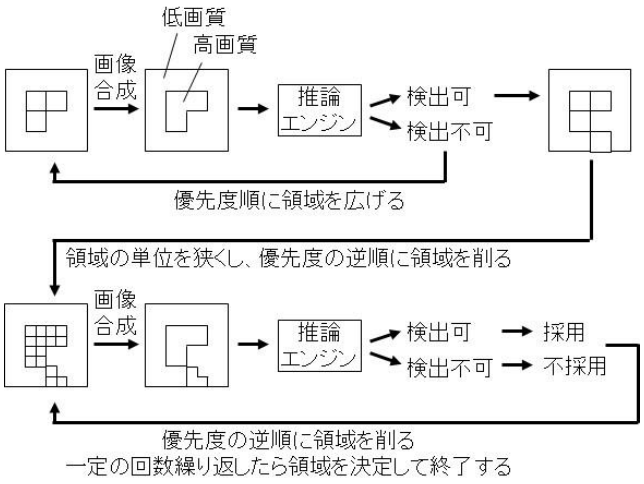

図 4 原因領域抽出

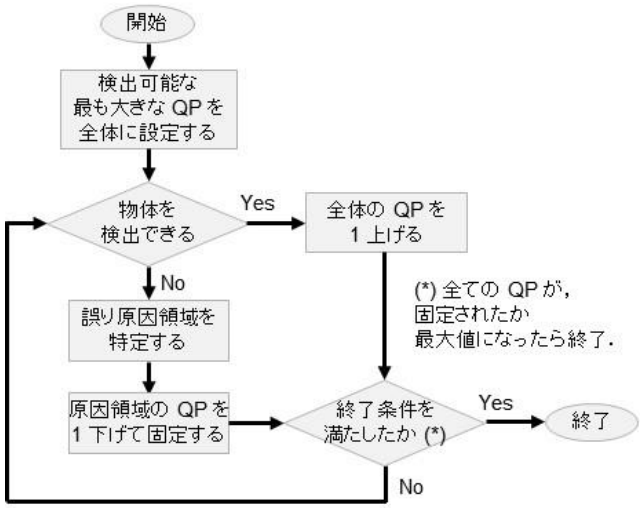

図 5 QP 決定の流れ

3.8 QP（量子化パラメータ）の決定

提案手法で用いた QP 決定の流れを図 5 に示す．最初の検出可能な最も大きな QP の決定は，高速化を目的として二分探索で QP を変えつつ物体を検出できるかどうか確認することで実装した．その後 QP を 1 ずつ上げて，物体を検出できなくなった時に提案手法を用いて原因領域を求めて，原因領域の QP を 1 下げることで物体を検出できる状態を維持している．一度原因領域になった領域の QP は固定して変動させないことで QP の収束速度を速めている．

4. 提案手法の評価

4.1 実験環境

以降の実験では符号化方式として H.265/HEVC[14]を利用した．量子化粒度の変更は QP を利用し，領域毎の細かい画質の変化は H.265/HEVC で符号化した時の CU（Coding Unit）ごとに QP を変えることで実施した．CU の領域が 8x8 pixel より大きくなる時は，CU に含まれる領域の中から提案手法で求めた最も小さな QP を選択した．物体検出には COCO データセット[16]で学習した YOLOv3[5][15]を利用した．実験対象の動画データはハイビジョン・システム評

価用標準動画像第二版[11]から検出可能な物体が映っている動画を 56 本選択し，最初の 33 フレームを抜き出したものを使用した．物体検出を行う時は，YOLOv3 の入力サイズである 416x416 に縮尺して，入力の際に YUV データを RGB データに変換している．本実験では各動画で物体検出を行い，一番高いスコアで検出できた物体を検出可能な状態を維持したまま符号化する手法を試みた．以降では YOLOv3 のスコアが 0.7 以上の物体を検出できたものとして，符号化後の動画データで対象の物体を検出できたフレーム数を，元の動画で対象の物体を検出できたフレーム数で割ったものを検出率と定義した．適用用途に遅延が少ない要件があるため，B フレームを使用せずに I フレーム一つに付き P フレームが 31 続く構成で符号化している．

4.2 比較対象

いくつかの単純な実験結果と提案手法の実験結果を比較する．比較した手法の詳細は後述するが，(A) 全体を均一に人手で解析する場合に適用要件を満たす QP 30 で符号化したものと，(B) 検出対象の物体のみ抽出して QP 30 で符号化したものと，(C) フレーム毎に物体を検出できる最低限の QP で符号化したものである．図 6 が 4 種類の方式で符号化した動画データのサイズを比較したものであり，横軸が先に述べた 56 本の動画それぞれで，縦軸が全体を均一な画質で符号化したものを基準としたデータサイズの比である．図 7 が物体の検出率であり，横軸は図 6 と同様に 56 本の動画それぞれで，縦軸が符号化する前の動画で物体が検出されたフレームのうち，符号化後も物体が検出されたフレームの比率である．なお方式毎に値をソートしてプロットしたため，図 6 と図 7 は同じ縦列にプロットした値であっても同じ動画とは限らない．

(A) 均一な画質で符号化する手法

特に工夫を行わない標準的な動画の符号化方式として，全体を均一な画質で符号化したものを比較対象とした．QP は人手で解析する場合に適用要件を満たす 30 に設定した．

本実験で用いた推論モデルでは，符号化することで 4 割程度の動画で検出できなくなる物体が存在しており（図 7），推論モデルや対象とする動画像に応じて符号化方法を変える必要があると分かる．

(B) 物体の抽出のみ行う手法

物体検出できる範囲で動画データのサイズを下げる最も単純な手法として，物体の存在する領域を抽出して，それ以外の領域を黒で塗りつぶす手法を試した．物体の存在する領域の抽出には，YOLOv3 によって物体が存在する領域と判断された Bounding Box（BB）と呼ばれる長方形の領域を利用した．つまり BB の大小は検出対象とする物体の大小を表している．検出率が物体の大きさによって変わるため，検出対象の物体の大きさで分類し，QP を 0 と 30 にしたもの二通りに対して，BB をそのまま抽出したものと BB に対して上下左右に 32 pixel 広げたものと 64 pixel 広げたもので検出率を確認した（図 8，図 9）．横軸が BB の長方形の大きさで，縦軸が物体の検出率である．この実験結果から検出したい物体のサイズが小さいほど広範囲の背景が必要なことが分かる．また，QP が小さいほど検出率が良くなり，BB を広げる幅が大きいほど検出率は良くなるが，ほとんど画質の劣化が無い QP 0 の場合で BB を 64 pixel 広げて検出したものでも検出できない物体が存在しており，高い検出率を維持して圧縮するには扱う動画や注目する物体に応じて符号化方法を調整する必要がある．

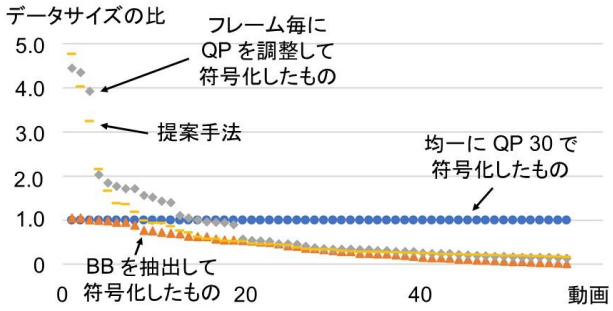

図 6 動画のデータサイズの比較

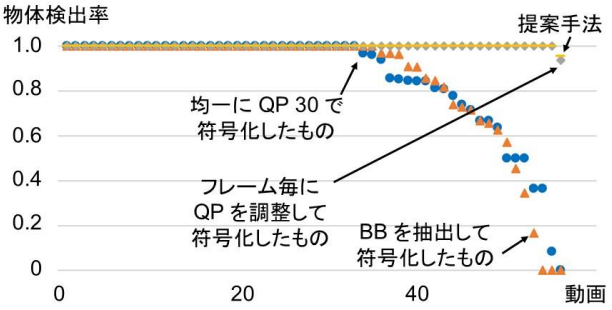

図 7 物体検出率の比較

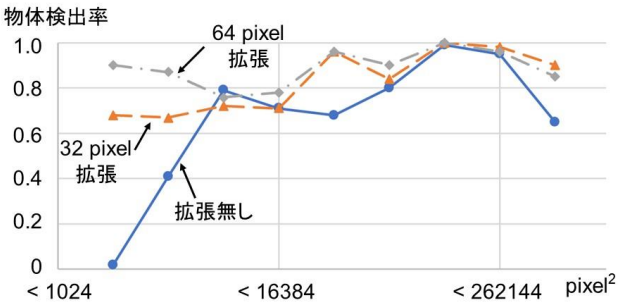

図 8 （B）における物体抽出法の結果（QP 30）

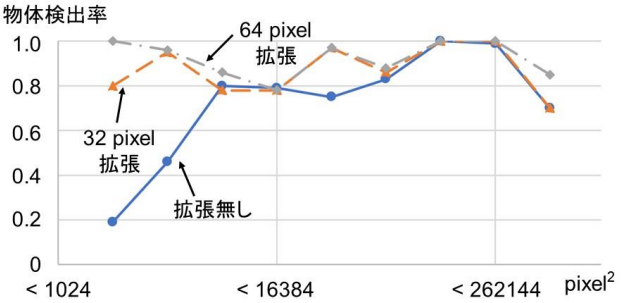

図 9 （B）における物体抽出法の結果（QP 0）

4 つの方式を比較している図 6 と図 7 では QP 30 を用いて BB に対して上下左右に 64 pixel 広げて抽出した結果を採用している．

(C) フレーム毎に QP を変動させる手法

フレーム毎に物体が検出できるぎりぎりの量子化粒度を採用することで，検出率を落とさずにどこまで動画データのサイズを小さくできるか確認した．

図 7 に示したように物体検出率は非常に高い．符号化後の動画では図 10 に写っている人物が検出できなかったが，もともと YOLOv3 のスコアが 0.7 と検出できたかどうかの境界のスコアだったため，許容できる範囲と判断した．動画データのサイズは QP 30 で均一に符号化したデータよりも小さくなるケースの方が多い．ただし図 10 のようなぎりぎり検出出来る物体を苦手としていて，検出するために高画質にすることからサイズが大きくなる傾向にある．

4.3 提案手法の評価

提案手法も物体検出率は非常に高い（図 7）．検出対象となる物体毎に符号化方法を変えることで，QP 30 で均一に符号化したものや，固定幅で BB を抽出して符号化したものでは検出できなかった物体が検出できている．

圧縮率は高い動画から低い動画まで存在する（図 8）．YOLOv3 の検出スコアが高く容易に検出できるものは圧縮率を上げることができ，検出スコアが低く検出が難しいものは圧縮率が悪くなる．フレーム毎に QP を変動させた手法に対しては，検出対象が映っていない領域の量子化粒度を粗くした効果が出ており，ほぼ全ての動画で圧縮率が上回った．均一に QP 30 で符号化した方式に対しては圧縮率が高い動画も低い動画もあるが，8 割程度の動画で圧縮率が向上しており有用と言える．特に図 11 のように検出対象が小さい動画を得意としており，均一に QP 30 で符号化した方式に対して約 0.18 倍までデータサイズが小さくなった．BB を抽出して符号化したものに対しては圧縮率が下がっており，不要な領域を黒塗りにする効果が高いことが分かる．今後，提案手法に対して検出率を下げない範囲で黒塗りにする方法を検討したい．

提案手法は全ての物体を検出できることを目標とした．しかし検出率を犠牲にして圧縮率を向上させることも可能で用途に応じて最適化する必要がある．特に原画像でぎりぎり物体が検出出来たような動画については検出率を維持するために圧縮率が悪くなっており，そういった物体の検出を諦めることで圧縮率が大きく改善すると思われる．

5. システム構成

これまで動画のデータサイズを削減する重要性と，削減するための手法について述べてきた．ここからは提案手法を前提とした動画解析システムの構成を例示する．

5.1 エッジクラウド連携型構成

高度な解析処理を行う場合には，クラウドなどリソース制約が少ないデータセンターと連携する必要がある．しかしデータセンターへの伝送はデータ量に応じた課金やあらかじめ必要な帯域を確保するように契約されることが多く，可能な限り伝送するデータ量を削減したい．

図 10 （C）において検出できない動画

図 11 （D）において圧縮率が高い動画

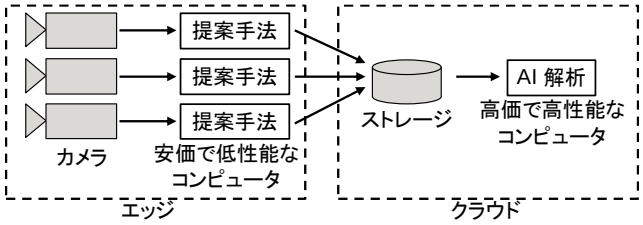

図 12 エッジクラウド環境のシステム構成

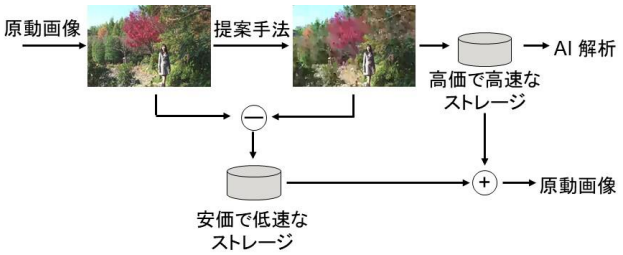

図 13 階層ストレージを用いたシステム構成

一方で，多数のカメラを配置して監視などに利活用するような，大量の動画像を扱う需要は多い．しかし監視範囲が広範囲になるにつれて人手での監視は現実的ではなくなり，AI による監視の需要が高まっている．監視して実施したい内容には不審人物の検知や作業者への支援，老朽化した危険な機器の発見などがある．それらの解析処理を行うには大きな演算能力が必要であり，カメラで撮影した画像をその場（エッジ）で解析するには多数の高性能なコンピュータが必要になる．そのためにクラウドにデータを送って処理したいのだが，例えば 8K カメラを 10 台で一日動かして人が目視で解析するための符号化をした場合，10TB もの膨大なデータ量になり通信コストが高くなる．

そこでカメラの画像を，可能な限りカメラの近傍で提案手法を用いて符号化してからクラウドに転送することで，通信と蓄積に要するコストを低減して安価に広範囲な空間を監視するシステムが構築できる（図 12）．図 6 と図 7 に示すように標準的な符号化方式に比べて高い圧縮率と検出率を実現できる．

5.2 階層ストレージ構成

データを蓄積する際にはデータを選別して，使用頻度や重要度が高いデータは高速にアクセス（読み書き）できるストレージに配置し，使用頻度や重要度が低いデータは相対的に安価なストレージに配置することで蓄積コストをコントロールすることができる．

我々の提案する手法においても，特定の AI 解析を行うだけであれば提案手法で符号化した動画のみで行える．しかし異なる目的の AI 解析を行う場合や，人間が詳細な状況を確認したい場合は，一度元のデータに戻して再度用途に応じた符号化を行う必要がある．

そこで提案手法で符号化した AI 解析を行うためのデータを高価で高速なストレージに置き，復元するための差分データを安価で低速なストレージに置く構成を提示する（図 13）．

6. まとめ

深層学習を用いた物体検出の検出率を維持したまま高い圧縮率で動画を符号化する方法を提案した．提案手法は人手を介することなく検出対象を自動的に見つけて符号化している．単純な手法では検出率を低下させずに圧縮することが難しいことを示し，深層学習の必要とする領域のみ高画質に記録することで，物体検出率を維持したまま 8 割以上の動画で標準的な符号化方式の圧縮率を上回ることが出来た．

今後は実際の現場に運用して，評価と改善を行う必要がある．また，演算負荷を軽くして安価に高速に動作する方式も検討していく．

謝辞 加藤正文氏を始めとして論文執筆に協力していただいた皆様に感謝の意を示す．

参考文献

[1] “The Digitization of the World From Edge to Core”. https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf, (参照 2020-01-24).
[2] 三菱総合研究所, “インターネットトラヒックの現状” https://www.soumu.go.jp/main_content/000534007.pdf, (参照 2020-01-24)
[3] 大久保榮, 鈴木輝彦, 高村誠之, 中條健. H.265/HEVC 教科書. インプレスジャパン, 2013.
[4] Chen, Yue, et al. "An overview of core coding tools in the AV1 video codec." 2018 Picture Coding Symposium (PCS). IEEE, 2018.
[5] J. Redmon et.al., 'YOLOv3: An Incremental Improvement', arXiv:1804.02767, 2018.
[6] Cao, Zhe, et al. "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields." arXiv preprint arXiv:1812.08008 (2018).
[7] Hadizadeh, Hadi, and Ivan V. Bajić. "Saliency-aware video compression." IEEE Transactions on Image Processing 23.1 (2013): 19-33.
[8] Chen, Mei-Juan, et al. "ROI video coding based on H. 263+ with robust skin-color detection technique." IEEE Transactions on Consumer Electronics 49.3 (2003): 724-730.
[9] Doukas, Charalampos, and Ilias Maglogiannis. "Region of interest coding techniques for medical image compression." IEEE Engineering in medicine and Biology Magazine 26.5 (2007): 29-35.
[10] 久保田智規, 中尾鷹詔, 吉田英司. “ディープラーニングによる物体検出において正しく検出できない原因を解析する手法の提案” 信学技報, vol. 119, no. 317, AI2019-30, pp. 1-6, 2019 年 11 月.
[11] ハイビジョン・システム評価用標準動画像 第２版 (ITE/ARIB Hi-Vision Test Sequence 2nd Edition), 映像情報メディア学会
[12] Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." Proceedings of the IEEE International Conference on Computer Vision. 2017.
[13] A. Adadi et al., "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)", IEEE Access 6: 52138-52160, 2018.
[14] ISO/IEC 23008-2:2017, “High efficiency coding and media delivery in heterogeneous environments -Part2: High Efficiency Video Coding,” Oct. 2017. | Recommendation ITU-T H.265(2018), “High Efficiency Video Coding”, 2018.
[15] “YOLO: Real-Time Object Detection”. http://pjreddie.com/yolo/, (参照 2020-01-24)
[16] “COCO Common Objects in Context”. http://cocodataset.org/, (参照 2020-01-24)