

# AIを用いた低ビットレート動画の高精細化技術

洞井 晋一<sup>1</sup> 増田 貴大<sup>1</sup> 東田 光裕<sup>1</sup>

**概要:** LPWA の通信方式はセンサーの情報収集に主眼が置かれているため, 画像や動画のような容量の多いデータを送受信することは難しい. 我々は低ビットレートに変換した動画を送信し, 受信側で AI を用いて復元することで, LPWA での送信に耐えうる画像を視聴に耐えうる画像への変換する技術に取り組んだ. 特に監視カメラや周回するバスに取り付けたドライブレコーダのように撮影対象が限定される動画に適用することで, 従来では難しかった  $128 \times 128$  から  $512 \times 512$  への拡大や, グレースケール画像のカラー化などを実現した.

## Application of Low Bitrate Video Super-Resolution using Deep Learning

Shinichi Doi<sup>1</sup> Takahiro Masuda<sup>1</sup> Mitsuhiro Higashida<sup>1</sup>

### 1. はじめに

IoT 技術の拡がりと共に, 広範囲かつ低消費電力で利用できる LPWA (Low Power Wide Area) が注目されている. LPWA は特にセンサーなどを用いて情報収集することに適しており, 温度や湿度といった環境に関する情報であったり, ガスや水道のメーター情報の収集にも用いられている. これらのセンサデータはデータ容量が少ないため, LPWA では数十 kbps からせいぜい数百 kbps 程度の狭帯域しか利用できない. 一方で, データ量が多い画像や動画を用いて現場の確認を行ったり, 画像解析によってセンサーの代わりとするような場面が増えてきている. 特に AI 技術の拡がりによって画像解析の精度は飛躍的に向上したため, LPWA などを用いて広範囲から画像や動画の情報を収集する需要は日々高まっているが, 前述のとおり数十 kbps から数百 kbps 程度では送受信することが難しい.

我々は数百 kbps 程度の動画を, AI を用いて高精細化する技術に取り組んだ [1][2]. 特に農場に設置された定点カメラや, 同じ道路を周回するバス, 軌道上を走るバスなど, 撮影対象が限定されたものに焦点を当てることで, 従来では難しかった非常に低い解像度の画像を高解像度に復元することや, 色情報が無いグレースケール画像からのカラー

画像復元などを実現した. これにより, LPWA での送信に耐えられる程度の低ビットレート動画を視聴に耐えられる程度の高精細動画に復元することが可能となった. 本研究の有効性を示すために, 以下の 2 つのデモンストレーションを作成した.

- 監視カメラを想定した, 定点カメラの映像復元
- 移動体カメラを想定した, 鉄道模型映像の復元

### 2. AI による低ビットレート動画の高精細化技術

ニューラルネットワークを用いて画像を生成する方法としては GAN (Generative Adversarial Network) による方式のものが注目されており, グレースケール画像や線画から着色を行う pix2pix[3] や, 低解像度の画像から高解像度の画像を生成する enhanceNet[4] などがある. 本研究ではこれらの実装を組み合わせることで改良を加えることで低ビットレート動画を高精細にする AI を作成した [1]. 本研究で作成した AI は Python を用いて実装し, 主に Tensorflow[5] を利用している. また, 動画をフレーム毎に分割して画像として処理を行い, それらの画像を連結することで動画の生成を行う. 実行時に何倍に拡大するかを指定する必要がある. 通常は 2 倍か 4 倍を指定する. なお, この倍数は 2 の累乗の必要がある. また, 入力画像も 2 の累乗の必要があ

<sup>1</sup> 西日本電信電話株式会社

るため、例えば  $128 \times 128$  や  $256 \times 256$  のサイズに縮小して用意する必要がある。学習を行う際には教師データとして高精細な画像が必要となるが、画像サイズは十分大きなものを用意すれば良い。例えば入力画像が  $128 \times 128$  で倍数が4倍の場合、出力画像は  $512 \times 512$  となるため、与えられた教師データの画像を  $512 \times 512$  に縮小して利用する。

学習時には教師データの画像を保管したディレクトリ名、入力画像を保管したディレクトリ名を指定し、epoch 数や batch 処理数などの各種設定値と、パラメータ値などを与えて実行する。画像生成時は入力画像のディレクトリ名を指定して、ディレクトリ内にある画像を高精細化するローカルモードと、TCP の接続を待ち受け、ソケット上から画像を受け取って高精細な画像を生成するサーバモードがある。AI の検証作業等を行う場合にはローカルモードを利用し、デモンストレーションや実際の利用時にはサーバモードを利用する。生成した画像は Websocket 経由で閲覧者に提供する。閲覧者は閲覧用の Web ページをブラウザで閲覧し、ブラウザは Web ページに書かれた Javascript に従って Websocket を AI が駆動しているサーバに接続し、生成された画像を順次受け取り、表示する。連続的に画像を受け取って順次表示することで動画として再生を行う。

学習時と画像生成時の両方で高性能な GPU を1台必要としており、検証では NVIDIA 社製の GeForce 1080Ti を利用した。尚、 $256 \times 256$  サイズの動画で 20fps の性能を出すために GPU が1台必要となるため、10fps の性能で良ければ1台の GPU で2本の動画ストリームを高精細化することが可能である。

また、フル HD 動画 ( $1920 \times 1080$ ) を始めとして一般的に動画は横長であることが多い。そのため、例えば出力動画のサイズが  $1024 \times 1024$  であった場合でも元の  $1920 \times 1080$  のサイズに戻すには横に2倍程度大きくする必要があり、拡大する際に画質の劣化が起きてしまう。そこで、入力画像を左右で分割し、左右それぞれの AI で高精細化を行い、出力画像を結合して出力することで横長の動画であっても高精細に復元できるように改良を加えた。この場合、入力画像が  $256 \times 128$ 、倍数が4であった場合、出力画像は  $1024 \times 512$  となる。なお、GPU は2台必要となる。

### 3. 定点カメラの映像復元

監視カメラ映像の高精細化を行うデモンストレーションとして、定点の Web カメラ映像の復元を行うデモンストレーションを作成した。Web カメラはノートパソコンを接続し、Web カメラで撮影した画像をそのまま AI 処理用のサーバへと転送する。AI 処理用サーバでは転送されてきた画像を  $256 \times 128$  に縮小し、4倍に拡大することで  $1024 \times 512$  の大きさの画像を生成する。

本来であれば低ビットレート化するために、H.264 などのコーデックを用いてフレーム間の圧縮を行う必要があるが、

デモンストレーションでは簡単のために JPEG 形式の画像のまま転送している。そのため、実際には低ビットレートに圧縮した際に発生するブロックノイズ等によって品質が劣化すると考えられるが、デモンストレーション上では理想的な画像しか入力されないため、最も綺麗に復元されたフレームしか表示されないことに注意されたい。

また、デモンストレーション用の機能として、閲覧時に画像上をマウスホバーすることで、マウスを中心とした小領域だけが元の Web カメラの画像が表示されるように実装した。これは Websocket を使って復元画像をブラウザに送る際、元の Web カメラの画像も合わせて転送しておき、ブラウザ側でマウスホバーした際にマウス位置の小領域だけ元の画像を表示している。これにより、実際にはどうなっているのかを確認することで AI による復元がどの程度確からしいのかを確認できる。

また、将来的な用途として閲覧者の注目点だけは元の画像を表示するという方式が考えられる。AI で生成した画像はあくまでも生成した画像であるため、実際の画像とは全く異なる画像が生成される可能性がある。ミッションクリティカルな場面での利用を考えると、閲覧者が注目した点に関しては元の画像を表示したいという需要はあると考えられる。そこで、閲覧者の注目点部分の小領域だけは元の画像を送ることでトラフィック量を減らしつつ、閲覧者の注目点だけは元の画像を表示することができる。本来であればマウスホバーに対して Web カメラ側から元の画像が送られてくるまでは遅延があるが、今回の実装方法では元の画像の全てをそのまま送っているため、そうした遅延は表現されなく、あくまでもコンセプトの展示としている。

### 4. 鉄道模型映像の復元

ドライブレコーダのような移動体に取り付けられたカメラを想定して、鉄道模型に取り付けたカメラ映像の復元を行った。カメラが取り付けられた鉄道模型としてタカラトミー社が販売している「プラレール スマホで運転! ダブルカメラ ドクターイエロー」を用いた。本製品はプラレール上を走る鉄道模型であるため、デモンストレーションの構築が容易かつ低価格であるばかりか、運搬にも優れており、玩具店等でも入手できるため調達することも容易いという利点がある。

本製品は WiFi を使って映像を転送するように設計されており、車両本体の撮影部を改良して低ビットレートでの転送を行うことは困難であったため、車両本体から送信される映像は高ビットレートの高精細な映像となっている。こちらも本来であれば車両から低ビットレートの映像を転送すべきであるが、あくまでも映像の高精細化を目的としたデモンストレーションであることを注意されたい。

送信された映像は定点カメラの映像と同様に復元されるが、移動体であるため固定的な部分がなく、4倍の拡大は難

しいと考えられる。そのため、デモでは2倍の拡大を行い、さらに横長の映像再現ではなく正方形の映像再現を行った。

一方で、AIを用いた復元では高精細化だけでなく着色を行うことも可能である。入力画像をグレースケール画像とし、教師データをカラー画像として学習すれば、被写体の特徴点などから着色が行われる。実際には映像をグレースケール化しても低ビットレートになるわけではない。YCbCrの色情報から帯域削減のために4:2:2や4:2:0などが用いられているように、そもそも画像情報は輝度情報が支配的であるため、そこからさらに色情報を落としても劇的に低ビットレート化するわけではない。グレースケール画像から着色された画像へ変換するにはAIを用いた高度の処理が必要となり、コストもかかることから低ビットレート化には色情報を落とすよりも解像度を落とす方が効率的であるといえる。ただし、例えば赤外線カメラなどを用いた夜間の映像であったり、出来る限り帯域を削減するために色情報を削減することも考慮し、着色を行う手法についても検討を行っている。デモンストレーションではグレースケールからの着色も行うことで、その可能性についても議論したい。

## 5. まとめ

本研究ではLPWAなどの狭帯域通信を用いて画像や動画を送信する方法として、低解像度やグレースケールの低容量の画像をあらかじめ学習しておいたAIを持ちいて復元する方法を提案した。デモンストレーションとして監視カメラを想定した定点カメラの映像復元と、移動体カメラを想定した鉄道模型カメラの映像復元を作成した。今後は閲覧者の注目点のみを送信する方式や、グレースケール画像の着色技術についての適用先などを議論していきたい。

## 参考文献

- [1] 洞井晋一, 増田貴大, 東田光裕. 深層学習を用いた低ビットレート動画の高精細化技術. マルチメディア, 分散, 協調とモバイル (DICOMO2018) シンポジウム, 2018.
- [2] 増田貴大, 洞井晋一, 東田光裕. ドライブレコーダ映像の低ビットレート配信とAIを使った高精細化. マルチメディア, 分散, 協調とモバイル (DICOMO2018) シンポジウム, 2018.
- [3] Daniel Michelsanti and Zheng-Hua Tan. Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification. 2017.
- [4] Mehdi SM Sajjadi, Bernhard Schölkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. 2016.
- [5] Tensorflow. <https://www.tensorflow.org/>.