

天候要素を考慮したビデオストリーミングの ノイズ除去フレームワークの設計と実装

陳 明康¹ 孫 静涛² 佐賀 一繁² 丹生 智也² 合田 憲人^{2,1}

概要: スマートシティの実現において、屋外カメラから収集されるデータの精度の高い分析が多くのアプリケーションで必要とされるが、悪天候の影響を受け、ビデオストリームデータの視認性の低下により、AI 解析における物体検知や物体認識をはじめとするデータ分析の精度が下がるという課題がある。本稿では、天候変化のある環境下でも、ビデオストリーミングデータから天候によるノイズを除去することにより、精度の高い物体認識や物体追跡を可能とするシステムを提案するとともに、そのプロトタイプ実装について報告する。本プロトタイプでは、センサカメラを搭載した IoT デバイスを用いて、著者らが開発している IoT アプリケーション開発を支援するためのメッセージングライブラリ SINETStream を利用し、IoT デバイスと、Edge サーバおよびクラウド上のサーバとの間でのデータ通信を実現した。また、各種の天気条件によるノイズの含まれる入力データに対して、天気分類モデルおよびノイズ除去アルゴリズムをオンラインに適用することにより、雨や雪、霧などの悪天候に関わるノイズを除去し、鮮明なデータに復元できるフレームワークを設計・実装した。さらに、現実の動画を用いた画質評価ベンチマークの実験により、提案システムの有用性を確認した。

Design and Implementation of Video Streaming Noise Removal Framework Under Influence of Weather Conditions

1. はじめに

5G ネットワークの提供開始、AI 技術の普及、センサデバイスの低価格化などにより、IoT は様々な研究分野へ広く活用されている。IoT デバイスから収集されたデータを安全にクラウド計算基盤上に蓄積・処理・解析するのが一般的なものとなりつつあり、そのクラウド計算基盤上で、AI 技術を駆使し、物体認識や物体追跡のような IoT アプリケーションが実世界で稼働し始めている。しかし、実世界では、IoT デバイスから収集された画像やビデオといった非構造化データの解析処理に関して、デバイスのモビリティやネットワーク接続の動的変化により、データサイズやフォーマットなどに関する様々な課題があり、処理の結果次第で、機械学習による AI アルゴリズムの精度が著しく向上できるため、近年様々な提案手法が行っている [1], [2].

しかし、ほとんどの既存研究では、1) 用意されている鮮明なデータセットしか対象としておらず、ぼやけのようなノイズまったく考慮していない。2) ノイズの元となる雨、雪、霧という天候要素を自動的に識別し、天候に応じたノイズ除去が可能な総合的なシステムに関する研究が欠けている。

代表的な既存研究では、センサカメラおよびクラウドの計算資源を用い、不審者を特定するビデオ監視システムが提案されている [3]. 彼らの研究では、屋内の環境に適しているが、天候が変化する屋外環境においては、大雨や雪などのノイズが含まれた際に、不審者を識別するのが困難で、悪天候下における物体認識の解決方法を示されていないといった問題があり、実世界への応用に向いていない。また、近年、雨筋や雪片や霧の空間特徴を抽出することで、悪天候におけるノイズ除去に関する研究が多くの提案がされている [4], [5]. ノイズ除去の研究では、雨や霧や雪という特定の天気を中心に展開されており、天候を自動的に分類し、総合的にノイズ除去を行うことができない。

本稿では、IoT デバイスから 24 時間収集されたビデオ

¹ 総合研究大学院大学情報学専攻
Department of Informatics, The Graduate University for Advanced Studies, SOKENDAI

² 国立情報学研究所
National Institute of Informatics

データから、晴、雨、雪、霧の天候を自動的に判断し、天候要素によるノイズを自動的に除去できるビデオ監視システムを提案する。実装上では、IoT デバイスから安全にデータ収集するために、本研究が開発した SINETStream^{*1} という Python ライブラリを用いて Raspberry Pi と Jetson TX2 の二種類 IoT デバイスから、Edge サーバおよびクラウド上のサーバ（以下、Cloud サーバと略）の三者間でのデータ通信を実装した。さらに、IoT アプリケーションの開発者を支援するために、天候要素を動的に分類してラベル付けし、分類された天気の種類に従い、ノイズを除去できるアルゴリズムを動的に選択できるツールを開発した。最後に、ノイズを除去した鮮明なデータのもとで、動的に物体検知可能な監視システムのプロトタイプを実装し、BRISQUE, SSEQ, NIQE という三種類のベンチマークで提案手法の有用性を検証した。

本稿の構成は、以下のように構成している。2 節において関連研究を紹介し、3 節において、監視システムのシナリオと提案手法を述べる。また、4 節では、本研究のシステムアーキテクチャの設計とプロトタイプシステムの実装方法を説明する。5 節で複数のベンチマークを利用し、提案したビデオ監視システムにおける評価を議論し、最後に、4 節で、今後の課題を述べる。

2. 関連研究

雨や霧、雪などの天候要素により劣化したビデオストリームデータの復元問題を解決することを目的としたノイズ除去技術が多く研究されており、周波数領域変換と空間領域特徴に基づいて雨、霧、雪の部分を分離させる手法が検討されている [4], [5], [6]。しかし、復元された画像の品質は降雨・降雪方向と霧濃度に依存し、復元画像は肉眼で見えるほどのぼやけてしまう。さらに、降雨・降雪量が多い場合は、計算量が非常に多くなり、処理時間が長いため、リアルタイムが求められている IoT アプリケーションの実現には困難である。

また、近年ディープラーニングを用いたノイズ除去の研究も増加している。降雨・降雪量や霧濃度が復元効果に影響を与えてしまう問題を克服するため、雨、霧、雪のない鮮明な画像と異なる程度のノイズを混入させた人工合成画像により構成された学習データセットが利用されている。Huang ら [7], Fan ら [8], Liu ら [9] などは、それぞれ雨、霧、雪の人工合成画像データセットから背景情報と雨、霧、雪の特徴を学習して自動的に識別・分離することに焦点を当てており、背景から雨、霧、雪のノイズ除去をすることができる。しかし、上記のような手法は各学習データへの依存度が高く、特定の天候しか対応できないため、スマートシティにおける IoT アプリケーションへの適用が難しい。

そのため、現実世界から収集されたビデオストリームデータの解析精度を向上するためには、自動的に天候を分類し、適応的にノイズ除去することが必要となっている。

3. 提案手法

本節では、ビデオストリームデータの収集や天候要素による画質劣化の問題点を解決するために提案するビデオストリーミングのノイズ除去フレームワークについて、その利用シナリオとともに述べる。

3.1 IoT ビデオストリーミングのシナリオ

我々の検討では、図 1 に示す、屋外の IoT デバイスに搭載されたカメラから収集されるビデオストリーミングデータに対して動画解析を行うことが可能なオンライン監視システムを想定した。

- シナリオの展開環境：学校キャンパスを想定した。その校舎、道路脇などにはセンサカメラを搭載したの静的 IoT デバイスが設置され、またキャンパス内を移動するドローンや車などにはセンサカメラを搭載しモバイル IoT デバイスが装備され、屋内外に複数の IoT デバイスが分散して存在している。
- ビデオストリームデータの収集：これらセンサカメラを搭載した IoT デバイスはキャンパス内のデータを 24 時間 365 日収集している。したがって、雨や霧、雪などの悪天候によって、しばしば視認性の低いビデオストリームデータを収集する。
- ビデオ解析：収集されたデータを Edge サーバまたは Cloud サーバのデータセンターにアップロードし、物体認識を代表とする AI アルゴリズムを用いてビデオ解析やデータの可視化を実現する。

上記のようなシナリオにおいて、好天時の鮮明なデータのみを学習させた場合、悪天候時のノイズの多いビデオストリームデータを高精度に処理することは難しいとされている。悪天候による鮮明さが低下したビデオの解析精度を向上するためには、自動的に天候状況を分類し、適切なノイズ除去や明るさの補正を行うことで、鮮明な動画画像を生成できると考える。

3.2 提案手法

本節では、我々が提案する天候要素を考慮したビデオストリーミングの適応的なノイズ除去の提案手法について紹介する。我々の手法は、各種の天候状況における屋外カメラから撮影されたビデオストリームデータに対して、天気条件を分類し、適切なノイズ除去モデルを採用することで、自動的に様々な天気条件の変化への対応を可能とする。天気条件の分類やノイズ除去の処理は、処理の計算量や計算資源の能力を考慮して、適切な Edge または Cloud サーバで実行する。この手法を用いて、AI アルゴリズムの精度向

^{*1} <https://www.sinetstream.net/>

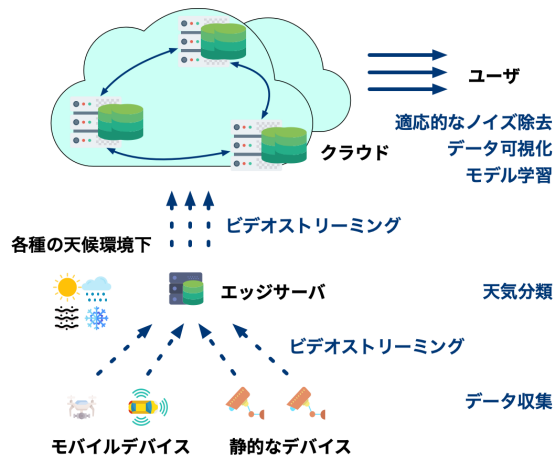


図 1: IoT ビデオストリーミングのシナリオ

Fig. 1 Scenario: Video streaming from IoT devices

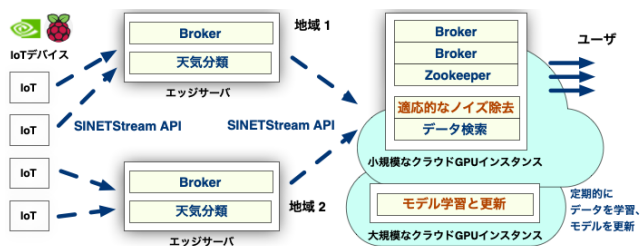


図 2: ノイズ除去フレームワークのアーキテクチャ

Fig. 2 Architecture of our noise removal framework.

上のため、異なる天候状況において鮮明なビデオストリームデータ収集を提供することに大きく貢献できると考える。さらに、開発するフレームワークを汎用性の高いソフトウェアとすることにより、各種の IoT アプリケーションへの適用を可能とすることを旨とする。

4. プロトタイプシステムの設計と実装

4.1 システムアーキテクチャの設計

本節では、提案するノイズ除去フレームワークのシステムアーキテクチャの設計について紹介する。構成を図 2 に示す。本フレームワークは、IoT デバイス、Edge サーバ、Cloud サーバの三者から構成されている。Edge サーバでは天気分類のような遅延時間が短く計算量が少ないタスクを、Cloud サーバではノイズ除去やデータ可視化といった計算量が大きいタスクを処理する。

IoT デバイスからのビデオストリーミングデータの収集には、IoT ストリームデータの処理のためのメッセージング API を提供する SINETStream を利用する [10]。具体的には、SINETStream が提供するトピックベースの Pub/Sub 型モデルの API を用いて IoT デバイスからのデータの書き込みやサーバのデータ読み出しを実装する。また、IoT デバイスとサーバ間でデータを仲介するメッセージブローカとして Kafka を利用する。IoT デバイス側の Publisher とデータを収集する Edge サーバや Cloud サーバの Sub-

scriber は、ブローカの通信パラメータやサービスを指定するだけでブローカに接続してメッセージの送受信ができる。両者は疎結合されており、相手の存在を知る必要もないため、IoT デバイスの大規模分散に適している。

Cloud 側では、二種類のクラウド GPU インスタンスを用意する。一方は小規模な GPU インスタンスであり、Cloud サーバとして無停止運転し、ビデオストリーミングを受信、天気分類されたビデオストリームデータから適切にノイズを除去し、ユーザーにデータ可視化や動画画像解析といったサービスを提供する。もう一方は、大規模な GPU インスタンスであり、周期的に運転し、GPU の高い演算性能を活用して、収集された大量のビデオストリームデータを学習して、ノイズ除去モデルの精度を向上させる役割を担う。

4.2 プロトタイプシステムの実装

プロトタイプシステムでは、IoT と Edge と Cloud の三者連携アーキテクチャに基づいて、IoT と Cloud 間、Edge と Cloud 間のデータの受け渡しは、それぞれ SINETStream を用いて実装した。また、Edge サーバおよびクラウドインスタンス上での処理は、それぞれ Docker コンテナ上に実装した。IoT デバイス側として Raspberry Pi 3 や Jetson TX2、Edge サーバとして 2.3 GHz Intel Core i5 を搭載した MacBook Pro、および Cloud サーバとして Amazon EC2 での g4dn.2xlarge と p3.2xlarge タイプの二種 GPU インスタンスを利用する。

ビデオストリーミングデータは、以下の手順で処理される：

- (1) Edge サーバは定期的にビデオストリームからビデオフレームを取得し、天気分類を実行する。
- (2) Edge サーバは、天気分類結果とともにビデオストリームデータを Cloud サーバへ転送する。
- (3) Cloud サーバでは、受信したビデオストリームデータに対して、天気分類結果に適切なノイズ除去処理を実行し、ノイズが除去されたデータを生成する。

4.2.1 天気分類

天気分類では、まず複雑な天気分類のタイプを定義する必要がある。我々の天気分類モデルでは、複雑な天気条件を「晴」「雨」「霧」「雪」の 4 つのタイプに分類している。異なるタイプの条件に対応可能な天気分類を実現するために、天気ごとにそれぞれ約 1 万枚の画像から構成されているオープンソースの多種天気画像データセット^{*2}を作成した。雨、霧、雪のような異なる悪天候は非常に似ているため、天気分類研究には、カテゴリの不均衡や過学習 (オーバー・フィッティング, overfitting) といった問題を抱えている。現在の研究では、損失関数の重み付けやデータセットのリサンプリングなどのいくつかの解決策が提案され

^{*2} <https://drive.google.com/file/d/13zJ65SBv5s40HS0kUbF1jT54TX7-awCB/view>

ていたが、処理時間や効果を大幅に改善する必要がある。よって、現在、最先端のニューラルネットワークフレームワーク (CNN) の一つである EfficientNet を利用することにして、わずか 1,900 万個程度のパラメータを用いる同時に、優れた Top-1 の精度を達成することができるモデルを採用した。さらに、進化的アルゴリズム (evolutionary algorithm) を用いて、カテゴリの不均衡や過学習の問題を解決し、多種類天気分類を実現した。

4.2.2 適応ノイズ除去

適応ノイズ除去モデルでは、天気分類の結果に応じて、異なる画像強調モデルとノイズ除去モデルに切り替わる手法を利用する [11]。我々の適応ノイズ除去は、晴天の画像に対して、生画像を直接出力するか、明るさの補正モデルに切り替わる。また、雨・霧・雪天候の場合は、雨や霧や雪の除去モデルに切り替わり、それぞれ入力画像の雨筋や霧や雪片のぼやけを除去する。

ノイズ除去モデルの代表例を挙げて、雨除去アルゴリズム選択の方法を説明する。まず、既存の五つの代表的な雨除去アルゴリズム：PReNet, DDN, GP-based SSL, LPNet, DerainNet の処理時間と結果の画質を評価し、最適なアルゴリズムを見つけ出した。画質の定量的評価では、人工合成画像データセットに対して、2つのフルリファレンス画像品質評価指標 (PSNR, SSIM) を利用し、定量的に画質を評価した。一方、実世界の雨画像データに対して、一般的に使用されている3つのノンリファレンス画像品質評価指標 (BRISQUE, SSEQ, NIQE) を利用し、定量的画質を評価した。最終的に、雨ノイズ除去モデルに処理時間が短くて画質が優れたアルゴリズムである Progressive ResNet (PReNet) [12] アルゴリズムを選択した。PReNet は、ResNet と多段再帰を単純に組み合わせた再帰型ニューラルネットワークであり、画像の雨除去に有利な性能を提供している。霧、雪除去アルゴリズムもほぼ同様に画質の定量的評価したため、本稿では説明を割愛する。霧除去アルゴリズムとして DMPHN [13] を、雪除去アルゴリズムとして JSTASR [14] を選択した。

5. 評価実験

本章では、現実の屋外で撮影された各種悪天候のビデオに対して、画質評価ベンチマーク実験を行い、ノイズ除去効果を評価した。さらに、SINETStream を介して、IoT デバイスから送信されるビデオストリームの遅延時間を計測し、提案手法の有用性を検証した。

5.1 実験環境

表 1 に、本実験の環境をまとめた。IoT デバイスとして Camera Module V2.1 に接続した Raspberry Pi 3 ならびにオンボードカメラを搭載した NVIDIA Jetson TX2 を用

表 1: 実験環境
Table 1 Experimental environment.

IoT デバイス	[rasp] Raspberry Pi 3 Model B または [tx2] NVIDIA Jetson TX2
Edge サーバ	MacBook Pro, 2.3 GHz Intel Core i5
Cloud サーバ	Amazon EC2, g4dn.2xlarge, (vCPUs=8, GPU=NVIDIA Tesla T4), AZ=ap-northeast-1d, Ubuntu18.04.1

表 2: 実験環境での ping (RTT) の結果 (単位: ミリ秒)
Table 2 Ping latency in experiment network (Unit: ms).

src → dest	最小値	平均値	最大値	標準偏差
rasp → edge	3.802	5.113	7.601	0.369
tx2 → edge	3.954	5.192	11.443	0.726
rasp → cloud	10.632	23.669	35.216	9.181
tx2 → cloud	13.802	25.899	39.452	10.727

表 3: ノイズ除去のノンリファレンス画像品質評価結果
Table 3 Non-reference image quality evaluation results of noise removal.

	BRISQUE	SSEQ	NIQE
雨	28.33	62.30	3.027
雨 (除去後)	29.84	57.65	2.485
霧	30.48	61.74	5.864
霧 (除去後)	43.20	57.00	3.734
雪	23.08	52.70	4.824
雪 (除去後)	38.11	69.40	3.289

い、Picamera^{*3}や GStreamer^{*4}を利用してビデオストリームを取得した。Edge サーバや Cloud サーバの計算ライブラリとして、PyTorch=1.1.0^{*5}, tensorflow-gpu=1.9.0^{*6}を用いた。IoT デバイスと Edge サーバは同一のサブネットに Wi-Fi で接続し、クラウドインスタンスとはインターネットで接続した。測定結果によると、実験環境でのインターネットの最大ダウンロード帯域は 551.22Mbps、最大アップロード帯域は 424.29Mbps となっていた。実験環境のベースライン通信性能を ping で測定した。表 2 に、レイテンシは ping で 100 回測定した RTT (Round-Trip-Time) の最小値、平均値、最大値、標準偏差を示す。

また、ノイズ除去効果の評価実験で利用したテストデータは、雨、霧、雪天候における都市街道を撮影したビデオまたは各種天候の画像データセット [12], [14] 構成されたものであり、天候ごとにそれぞれ 900 枚以上の写真に相当する。

^{*3} <https://picamera.readthedocs.io/>

^{*4} <https://gstreamer.freedesktop.org/>

^{*5} <https://pytorch.org/>

^{*6} <https://www.tensorflow.org/>



図 3: ノイズ除去モデルを用いたサンプル画像

Fig. 3 Sample images with noise removal models.

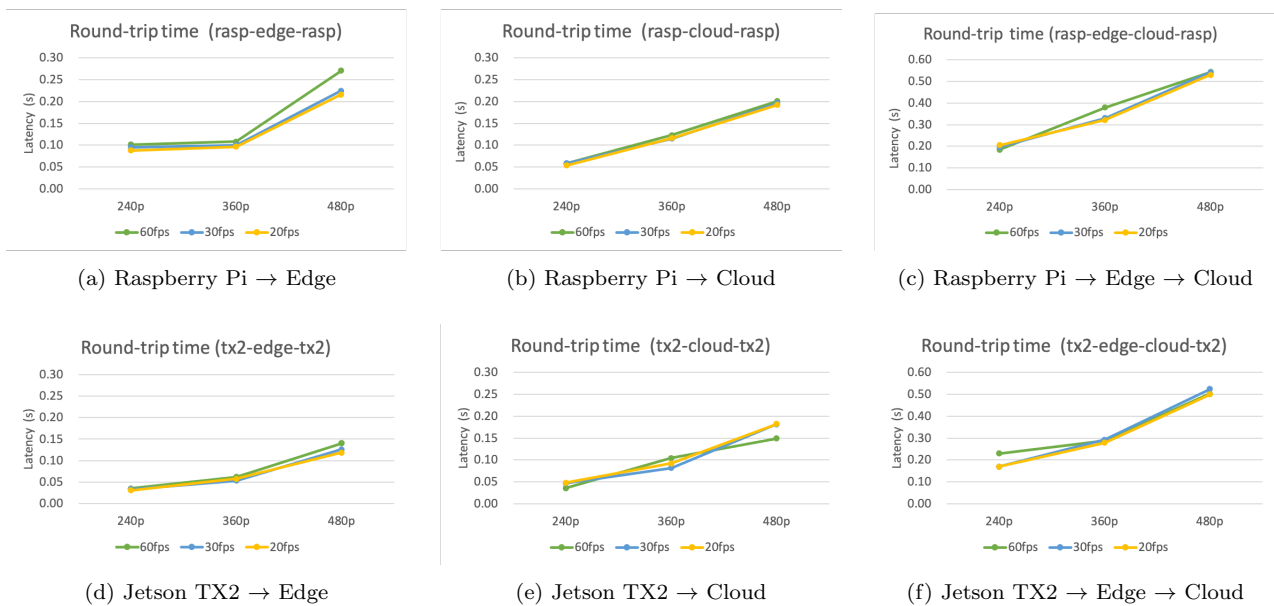


図 4: 遅延時間結果

Fig. 4 Results of latency

5.2 ノイズ除去効果評価実験

本実験では、ノンリファレンス画像品質評価指標 (SSEQ, BRISQUE, NIEQ) を用い、ノイズ除去効果を評価した。原理は、モデル統計量からのパラメータの偏差を自然画像に回帰させることで、画像劣化の度合いを計算するものである。その目的は、人間の観察者から得られた主観的品質スコア、すなわち差平均オピニオン評点 (Difference Mean Opinion Score, DMOS) を予測することである。ノンリファレンス画像品質評価は個人的なバイアスを持つため、本実験では個人的なバイアスの少ない評価指標として BRISQUE, SSEQ, NIQE の 3 つのモデルを用いて実験を実施した。表 3 に三種のノイズ除去のノンリファレンス画像品質評価の平均結果を示す。各項目で最も良い評価を太字で表示している。天気条件を問わず、ノイズ除去後の画像品質は、多くの項目でより良く評価されたことがわかつ

た。図 3 に評価画像の一部を示す。(a), (c), (e) は雨、霧、雪の元画像であり、(b), (d), (f) はそれぞれノイズ除去モデルで画質改善された画像を示す。

5.3 遅延時間測定実験

プロトタイプの下記通信経路のレイテンシを測定した：

- (1) IoT デバイス (Raspberry Pi 3, または Jetson TX2) から Edge サーバへの通信である。
- (2) IoT デバイスから Cloud サーバへの直接通信である。
- (3) IoT デバイスから Edge サーバを経由した Cloud サーバへの通信である。

また、フレームレート設定の違いによる影響を評価するために、3 種類のフレームレートで、連続ビデオストリームにおける 100 フレームの平均遅延時間を測定した。また、異なるノード間での原子時計非同期の影響を避けるため、ラ

ウンドトリップタイム (Round-Trip Time, RTT) を用い、ノード間でのビデオストリーミングの遅延時間を測定することにした。図 4 は Raspberry Pi 3 と Jetson TX2 で、上記 3 種類の通信経路における、3 種類の解像度、3 種類のフレームレート設定で測定した平均遅延時間を示す。横軸は送信する動画の解像度 240p, 360p, 480p, 縦軸は平均遅延時間を (秒) である。図 4 の結果から：

- (1) カメラの API で設定したフレームレートは遅延時間にほとんど影響を与えないが、動画の解像度が大きくなると遅延時間では長くなる傾向が示された。
- (2) IoT デバイスとして Raspberry Pi 3 と Jetson TX2 の同一条件での遅延時間の差は小さいが、Raspberry Pi 3 より Jetson TX2 は遅延時間が短い。
- (3) 当然のことではあるが、IoT デバイスの種類を問わず、IoT デバイスから Edge サーバを経由した Cloud サーバまでの遅延時間は、IoT デバイスと Cloud サーバ間の直接通信の遅延時間より長くなる
- (4) Edge と Cloud までの通信遅延時間の差が小さいことがわかった。

6. まとめと今後の課題

本稿では、天候要素を考慮したビデオストリーミングのノイズ除去のフレームワークを提案した。本フレームワークでは、自動的に天気条件を分類し、適応的にノイズ除去モデルを採用することで、分散の IoT デバイスから収集し、物体認識や物体追跡といった AI アルゴリズムの精度を向上することを可能とする。また、Raspberry Pi 3 と Jetson TX2 を用いて、IoT, Edge, Cloud の三者連携アーキテクチャを構成し、SINETStream を用いて提案フレームワークのプロトタイプシステムを実装した。

プロトタイプシステム上での実験として、現実の動画画像を対象として、3 つのよく使われているノンリファレンス画像品質評価指標を用いて、天候要素のノイズ除去の画質改善効果を実証した。また、収集されたビデオストリームデータの解像度やフレームレートの変化に注目し、構築したプロトタイプシステムの通信遅延時間を測定した。今後の課題としては、LTE 通信環境等の他の環境要素の変化による遅延時間への影響の調査や、Edge サーバ間のロードバランシングや異なるタスクのスケジューリング手法の導入を検討する。

参考文献

- [1] Ali, M., Anjum, A., Yaseen, M. U., Zamani, A. R., Balouek-Thomert, D., Rana, O. and Parashar, M.: Edge enhanced deep learning system for large-scale video stream analytics, *2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC)*, IEEE, pp. 1–10 (2018).
- [2] Kanzaki, H., Schubert, K. and Bambos, N.: Video

- streaming schemes for industrial IoT, *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, IEEE, pp. 1–7 (2017).
- [3] Zeng, X., Fang, B., Shen, H. and Zhang, M.: Distream: scaling live video analytics with workload-adaptive distributed edge intelligence, *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pp. 409–421 (2020).
- [4] Xu, J., Zhao, W., Liu, P. and Tang, X.: An improved guidance image based method to remove rain and snow in a single image, *Computer and Information Science*, Vol. 5, No. 3, pp. 49–55 (2012).
- [5] He, K., Sun, J. and Tang, X.: Single image haze removal using dark channel prior, *IEEE transactions on pattern analysis and machine intelligence*, Vol. 33, No. 12, pp. 2341–2353 (2010).
- [6] Kang, L.-W., Lin, C.-W., Lin, C.-T. and Lin, Y.-C.: Self-learning-based rain streak removal for image/video, *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, pp. 1871–1874 (2012).
- [7] Huang, D.-A., Kang, L.-W., Yang, M.-C., Lin, C.-W. and Wang, Y.-C. F.: Context-aware single image rain removal, *2012 IEEE International Conference on Multimedia and Expo*, IEEE, pp. 164–169 (2012).
- [8] Fan, X., Wang, Y., Tang, X., Gao, R. and Luo, Z.: Two-layer Gaussian process regression with example selection for image dehazing, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 27, No. 12, pp. 2505–2517 (2016).
- [9] Liu, Y.-F., Jaw, D.-W., Huang, S.-C. and Hwang, J.-N.: DesnowNet: Context-aware deep network for snow removal, *IEEE Transactions on Image Processing*, Vol. 27, No. 6, pp. 3064–3073 (2018).
- [10] 竹房あつ子, 孫 静涛, 藤原一毅, 吉田 浩, 合田憲人: IoT ストリームデータ処理のためのソフトウェアライブラリ SINETStream の開発, 研究報告インターネットと運用技術 (IOT), Vol. 2020, No. 19, pp. 1–8 (2020).
- [11] Chen, M., Sun, J., Saga, K., Tanjo, T. and Aida, K.: An adaptive noise removal tool for IoT image processing under influence of weather conditions, *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pp. 655–656 (2020).
- [12] Ren, D., Zuo, W., Hu, Q., Zhu, P. and Meng, D.: Progressive image deraining networks: A better and simpler baseline, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3937–3946 (2019).
- [13] Zhang, H., Dai, Y., Li, H. and Koniusz, P.: Deep stacked hierarchical multi-patch network for image deblurring, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5978–5986 (2019).
- [14] Chen, W.-T., Fang, H.-Y., Ding, J.-J., Tsai, C.-C. and Kuo, S.-Y.: JSTASR: Joint Size and Transparency-Aware Snow Removal Algorithm Based on Modified Partial Convolution and Veiling Effect Removal, *European Conference on Computer Vision*, pp. 754–770 (2020).