性能と軽量さのトレードオフを考慮した画像変換深層学習ネットワークに関する研究

2024 年度

柴崎　圭

学位論文 博士 (工学)

性能と軽量さのトレードオフを考慮した画像変換深層学習ネットワークに関する研究

2024 年度

慶應義塾大学大学院理工学研究科

柴崎　圭

目次

図目次

表目次

第 1 章

序　論

1.1　研究背景

近年，デジタル撮影技術の向上により，デジタル画像や動画の利用が広く普及している．動画像をデジタル上で視覚的に理解する技術をコンピュータビジョンといい，この分野の重要性も高まっている．古典的なコンピュータビジョン技術は，主に手作業で設計された特徴量抽出手法や，従来型の機械学習アルゴリズムに依存していた．Scale-Invariant Feature Transform (SIFT) や Histogram of Oriented Gradients (HOG) といった特徴量抽出手法は，エッジやコーナー，パターンの検出に有効であったが，これらの手法は対象物の複雑な変形や環境の変化に対して頑健ではないという課題があった．また，サポートベクターマシン（SVM）や k-Nearest Neighbors (k-NN) などの分類器は，手作業で抽出された画像の特徴を基にした分類を行っていたが，その精度や汎化能力は限られていた．

しかし，Krizhevsky ら [4] が Convolutional Neural Network (CNN) を用いたニューラルネットワークによって従来の手法を大幅に上回る画像分類手法を提案したことを皮切りに，深層学習によるコンピュータビジョン技術が急速に発展し，当該分野の基盤技術となっている．深層学習は，画像分類に限らず様々なコンピュータビジョンのタスクに応用されてきたが，動画像の変換や復元に関するタスクにも応用されている．Ronneberger ら [5] は，医療用画像のセグメンテーションのための U-Net を提案し，これを基本構造とした様々な派生モデルが提案され，特定の動画像変換タスクに対する精度向上が図られている．[1, 3, 6, 7] さらに，Isora ら [8] は，Generative Adversarial Network (GAN) を活用して，より写実的な画像変換手法を提案し，動画像変換において広く用いられている技術となっている．

近年，Dosovitskiy ら [9] は自然言語処理の分野で有用性が確認されている Transformer [10] を画像処理分野に適用した Vision Transformer (ViT) を提案した．Transformer は画像の長距離依存関係を効率的にとらえることができるため，これまでの手法では難しかった精緻な画像変換を可能にしている．Transformer を活用したネットワークは他にも提案されており [11–14]，CNN とは異なる特徴抽出器ながら優れた性能を達成している．

しかし，深層学習ベースの手法はその有用さとは裏腹に極めて計算資源を要することが知られており，実用的な運用のためには潤沢な計算資源（一般的には GPU）を必要とする．特に，Transformer ベースの手法は高性能を達成可能であるが，処理する画像解像度が大きい場合に計算量が跳ね上がる傾向にある．さらに，深層学習ベースの手法ではその性能と軽量さにはトレードオフが存在し，軽量さを保ちつつ高性能を達成することは困難である．計算資源をあまり必要としない軽量さを重視した手法は性能が低いものとなりがちであり，逆に性能が高い手法は軽量ではないものになりがちである．ここで述べる性能は，定量的な評価指標や定性的な比較によって総合的に判断される．タスクによって適切な評価指標は変わるが，既存手法と公正な比較のため，本研究では基本的には既存手法に従っており，各評価指標の詳細は節 2.3 で解説する．また軽量さは，計算の大部分を占めるニューラルネットワークのパラメータ数，計算量，実測速度から判断されるものであり，詳細は小節 2.3.1 で解説する．

性能と軽量さのトレードオフは多くの研究に見られる現象である．初期の深層学習では，主にフィルタリングやシンプルなアルゴリズムを用いていた [4, 5, 15] ため，計算コストは抑えられていたものの，その性能には限界があった．その後，CNN や Transformer モデル [9, 12, 13] の発展により，その性能は飛躍的に向上した．しかし，それと同時に計算資源の要求も増加し，性能と軽量さのトレードオフが重要な課題となった．

例えば，動画のノイズ除去においては，深層学習の導入により，より高度にフレーム間の相関を捉えることが可能となり，ノイズ除去の精度が大幅に向上した [16, 17]．しかし，高性能なモデルほどその計算コストは増加し，高解像度動画やリアルタイム処理といった実用的な応用が難しくなる．このため，近年では，性能を保ちながら効率性を高める手法 [6, 18] も模索されているが，依然としてトレードオフが完全に解消されたわけではない．この傾向は，特定のタスクに限らず同様の課題が見られ，これらにおいても，深層学習の進化によって精度が大幅に向上する一方で，軽量化と計算効率の確保が難題となっている．

このような問題を解決するためには，GPU で効率的に演算を行うためのフレーム

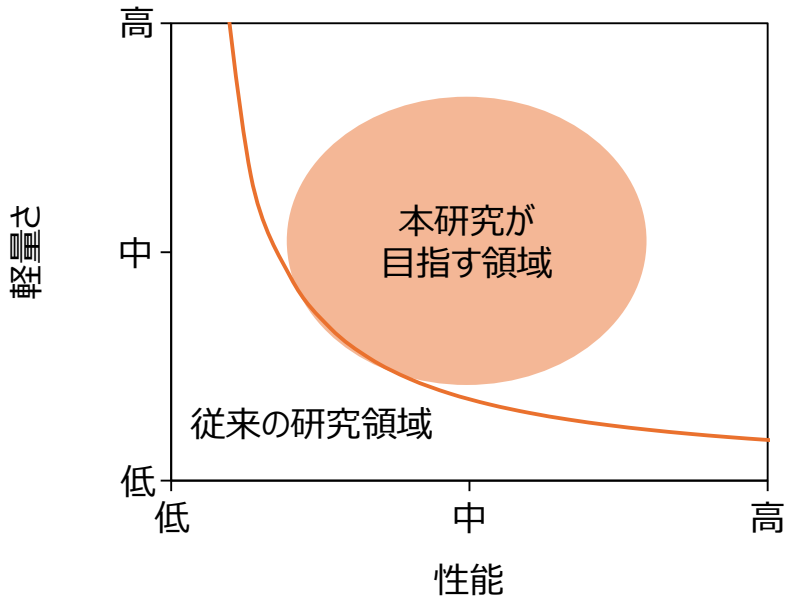


図 1.1　性能と軽量さのトレードオフの概念図．深層学習に置いては，性能が高い手法は軽量さが低く，軽量な手法は性能が低くなる傾向にあるが，本研究では，従来手法よりも性能と軽量さを両立している手法を考案していくことを目指す．

ワーク [19] や，ハードウェアの仕組みを意識したモジュールの開発 [20] などが挙げられるが，本研究では深層学習ネットワークやスキームの改善によって，性能と軽量さのトレードオフを克服することを試みる．図 1.1 は，性能と軽量さのトレードオフの概念図であり，本研究では，従来手法よりも性能と軽量さを両立している手法を考案していくことを目指す．そのためのアプローチは 2 つ考えられる．一つは，「新規モジュールを採用したネットワークを提案する」ことで，もう一つは「効果的な処理スキームや損失関数を提案する」ことである．ここで述べるモジュールとは，ネットワークを構成する要素であり，スキームとはネットワークに何を入力し何を出力するか，ネットワークの出力をどのように最終的な出力にしていくかを表すものである．本研究では，「新規モジュールを採用したネットワークを提案する」においては，既存の深層学習ネットワークアーキテクチャに対して新しいモジュールを追加・改良することで，軽量化と性能向上の両立を図ることを目指す．具体的には，従来のネットワークにおいてボトルネックとなっていた計算負荷を削減しつつ，表現力を維持または強化するための工夫が必要である．一方，「効果的な処理スキームや損失関数を提案する」に関しては，ネットワークの学習過程で使用する損失関数や推論の流れに注目することで，軽量モデルでも

高精度な予測が可能となることを目指す．特に，本研究では取り組むタスクの特性を考慮した構造を考案することで性能と軽量さを両立した手法を提案していく．

本研究では動画像変換タスク，フォトレタッチ，動画のノイズ除去，姿勢変換，非ペア画像変換の 4 つの幅広い画像変換タスクにおいて，「新規モジュールを採用したネットワークを提案する」，「効果的な処理スキームや損失関数を提案する」のいずれかのアプローチを用いて性能と軽量さを両立した手法を提案し，様々な実験検証を通じて課題解決のための，深層学習手法の設計において重要な知見を発見することを目指す．これらのタスクは性能と軽量さのトレードオフの克服が重要なタスクであり，以降はそれぞれのタスクに対する具体的な背景を記述する．

1.1.1 フォトレタッチの背景

フォトレタッチは画像変換タスクの一種であり，入力画像の色調を補正して人にとって自然で美しい色調へと変換することを目的としている．近年，深層学習によるフォトレタッチ技術が発展しており，優れたレタッチ性能を持つ手法が次々と提案されている．Hu ら [21] は，深層強化学習を用いた反復的なフォトレタッチ手法を提案した．Chen ら [22] は，Generative Adversarial Network (GAN) を活用したフォトレタッチ手法を提案した．また，Liang ら [1] は，主に画像の低周波成分が変換されることに着目したフォトレタッチ手法である Laplacian Pyramid Translation Network (LPTN) を提案した．LPTN では，入力画像からラプラシアンピラミッド [23] を作成し，高周波成分と低周波成分に分離し，低周波成分を重点的に変換するネットワークによって効率的な変換を実現した．しかし，LPTN は高速である一方，変換画像に特有のアーティファクトが生じる問題がある．このように，フォトレタッチでは反復的な計算で変換性能を向上させたり，効率的に変換を行う手法が提案されているが，これらは性能と効率を両立していない．

1.1.2 動画のノイズ除去の背景

動画のノイズ除去は，映像に含まれるノイズを取り除くタスクである．動画のノイズ除去は長年研究されている問題であり，GPU や深層学習が普及する以前にも多くの手法が提案されていた．Maggioni ら [24] は，自然な動画の特徴である時間的・空間的な冗長性を利用したフィルタリングアルゴリズムを提案した．この手法は非局所的な領域のグルーピングを通じて，時間的・空間的な相関を活用している．Jovanov ら [25] は，

動き推定とノイズレベル推定に基づいた選択的ウェーブレット収縮アルゴリズムを提案した．Arias ら [26] は，類似した時空間パッチに対してベイズモデルを構築する手法を提案した．

これらの手法は，GPU に最適化されておらず，CPU での動画処理は非常に低速である．また，最近の GPU や深層学習を用いた手法と比較して精度が劣る点も課題である．

近年，GPU を活用した深層学習ベースの手法が動画のノイズ除去において最先端の性能を示している．これらの手法は GPU での演算処理に最適化されているため，高速な推論が可能である．Tassano ら [27] は，空間的な相関を考慮したノイズ除去の後に時間的な方向を考慮したノイズ除去を行うネットワークを提案し，従来の CPU ベースの手法を超えるノイズ除去性能を達成した．Tassano らの手法以降も，多くのノイズ除去手法が考案されており，精度も大きく改善されている．Vaksman ら [28] は，実フレームに類似したパッチクラフトフレームを導入し，動画シーケンスを補強してから CNN ネットワークでノイズ除去を行う手法を提案した．

さらに，Liang ら [16] は，Attention モジュールによってフレーム間の相関を認識する手法を提案した．Buades ら [29] は，正則化オプティカルフロー法を用いて時間的な関係性を捉える手法を提案した．Shen ら [30] は，オプティカルフローに基づく補完アーキテクチャとフロー強調モジュールを組み合わせたネットワークを提案し，空間的および時間的な強調を実現している．Liang ら [17] は，Guided Deformable Attention モジュールを用いて，効率的に時間的相関を利用する手法を提案した．

また，軽量さを重視したネットワーク [6,18,31] もいくつか提案されている．Tassano ら [6] は，2 つの軽量な U-Net [5] で構成される高速なノイズ除去ネットワークを提案した．この手法はフレームをチャンネル方向に結合し，フレームとチャンネルの関係性を融合させることで高速な推論を実現しているが，スライディングウィンドウベースの推論により推論効率が低いという課題がある．Qi ら [18] は，Temporal Shift Module (TSM) [2] に代わる Bidirectional Buffer Block (BBB) を考案し，それを取り入れた Multi-Input Multi-Output (MIMO) ネットワークを提案した．Qi ら [18] は，MIMO ネットワークに特有の時間的なクリッピングによる性能低下に対して，推論時に TSM を BBB に置き換えることで対処している．

これらの手法は，性能と効率を一定のレベルで両立しているものの，両者を高い水準で同時に達成することには至っていない．Liang ら [16] の提案した手法は高い性能を達成しているが極めて大きいネットワークであり，NVIDIA GeForce RTX 3090 24GB において，メモリ不足により 128×128 の解像度の動画を処理できない．また，Qi ら [18] の手法は高速な推論を可能としたが，動画のノイズの除去性能が低く，1080p のような

高解像度の動画を NVIDIA GeForce RTX 3090 24GB で処理できないといった問題がある．

1.1.3 姿勢変換の背景

姿勢変換とは，ソースの人物画像とその姿勢情報，ターゲットの姿勢情報を用いて，ターゲットの姿勢をとった人物画像を生成するタスクである．Ma ら [32] は，まず粗い画像を生成し，その後詳細な画像を作成する手法を提案した．Esser ら [33] は，U-Net [5] を使用して，人物の外観と姿勢を分離する手法を提案した．しかし，純粋な CNN ベースの手法では，複雑な姿勢変換に対応しきれないという問題があった．そこで，Zhu ら [34] は，段階的に人物画像を生成し，姿勢に基づいて外観を調整する手法を提案した．さらに，Tang ら [35] は，姿勢と外観の特徴を交差させるモジュールを用いた手法を提案した．

Li ら [36] や Ren ら [3] は，オプティカルフローを推定して元画像のワーピングを行う手法を提案したが，大きな姿勢変化やオクルージョンが存在する場合，オプティカルフローの推定に失敗し，質の低い画像が生成されるという問題がある．また，オプティカルフローの推定はしばしば生成ネットワーク本体とは別に学習されるため，ハイパーパラメータの調整に手間がかかり，学習時間も増加する．

Ma ら [37]，Li ら [38]，Zhang ら [39] は，追加の人物のパース情報を利用したネットワークを提案し，姿勢変換の品質を向上させた．彼らの手法では，まずパースマップを推定してから最終的な人物画像を生成する．しかし，これらの手法で用いるパースマップの準備には労力がかかり，さらにパースマップ自体の信頼性にも課題がある．Zhang ら [40] は，Siamese Network [41] を活用して恒等写像を学習する補助タスクを設定することで，ネットワークがテクスチャ情報を効率的に取得できる手法を提案した．しかし，補助タスクの学習には追加のハイパーパラメータが必要であり，学習の複雑さが増えるという課題がある．

まとめると，最近の手法の多くは，姿勢変換が抱える問題に対して追加の学習データやタスクを導入して取り組んでいる．しかし，これらのアプローチは，追加データの準備やタスク追加によるハイパーパラメータや学習ステップの増加により，実用性やネットワークの軽量化が妨げられている．

1.1.4 非ペア画像変換の背景

非ペア画像変換は，非ペア画像データセットにおける画像変換タスクであり，ペア画像が存在する場合に比べ，基礎的なコンテンツを維持しつつ変換することが困難である．しかし，近年の研究 [7, 42–47] により，基礎的なコンテンツを維持しつつ変換する優れた手法が提案されている．Zhu ら [42] は，ソースドメインからターゲットドメイン，そして再びソースドメインへ画像を再構成する際に再構成損失を最小化する Cycle Consistency Loss を提案し，深層学習ネットワークが非ペア画像変換を行えるようにした．Liu ら [45] は，ドメイン間に共有潜在空間の仮定を置くことで非ペア画像変換を可能にした．また，Cycle Consistency Loss に代わる学習方法や制約も提案されている．Park ら [43] は，コントラスト学習によって入力画像と変換画像の相互情報量を最大化することで非ペア画像変換を行う手法を提案した．Fu ら [46] は，幾何的な変換に可換となるように Generator を制約する Geometry Consistency を提案した．Xu ら [44] は，入力画像に対する最大敵対的摂動に対して Generator が可換となるような Maximum Spatial Perturbation Consistency を提案した．

既存手法の多くは，非ペアデータにおいて基礎的なコンテンツを維持しつつ画像を変換することにフォーカスしており，形状の変換に対応できないことや背景が崩壊するという問題が残されている．この問題に対処することで，ネットワークの変換能力を変換すべき領域に集中させ，ネットワークを軽量に保ちながら変換性能を維持することが可能となる．Tang ら [7] は，前景を抽出する注意マップにより，変換すべき前景と変換すべきでない背景を分離する AttentionGAN を提案し，背景の崩壊に対応することを試みた．しかし，AttentionGAN の前景と背景の分離精度は低く，しばしば分離に失敗している．これは，AttentionGAN の前景と背景の分離が画像変換と同時に学習されており，バイアスのあるデータセットから悪影響を受けたことが原因だと考えられる．また，AttentionGAN は形状が変化する場合には効果的な注意マップを抽出できない．効果的に前景と背景を分離することで，ネットワークを軽量に保ちながら変換性能を維持できるため，この問題に対処する必要がある．

1.2 研究目的

本研究の取り組む課題と，それに対するアプローチを表 1.1 に示す．本研究では，「新規モジュールを採用したネットワークを提案する」ことと「効果的な処理スキームや損

表 1.1　本研究の取り組む課題とその分類.

課題	軽量化と性能の両立	
新規性の種類	新規モジュールを採用したネットワークを提案する	
具体的なタスク	フォトレタッチ	動画のノイズ除去
具体的な新規性と課題とのつながり	低解像度における Transformer モジュール	3D Conv を必要とせずに時間関係を捉えるモジュール
新規性の種類	効果的な処理スキームや損失関数を提案する	
具体的なタスク	姿勢変換	非ペア画像変換
具体的な新規性と課題とのつながり	タスクを分割し別々のモジュールで効率的に処理する学習	サリエンシーマップの活用によって前景と背景を分離し効率的な学習

失関数を提案する」という 2 つのアプローチを用いて，4 つの幅広いタスクにおいて軽量化と性能の両立という課題に対処可能な手法を提案する．本研究ではフォトレタッチ，動画のノイズ除去，姿勢変換，非ペア画像変換の 4 つのタスクに取り組む．フォトレタッチと動画のノイズ除去では「新規モジュールを採用したネットワークを提案する」アプローチ，姿勢変換と非ペア画像変換では「効果的な処理スキームや損失関数を提案する」アプローチを用いる．それぞれのタスクにおいて提案した新規性と本研究が取り組むべき課題との対応も表 1.1 に記している．それぞれのタスクに対する具体的な目的とその成果は以降の項で記す.

1.2.1　フォトレタッチの研究目的

フォトレタッチに関して，本研究では LPTN をベースに，入力画像をラプラシアンピラミッドで分解して処理するネットワークにおいて，低解像度部分の変換を Axial Transformer [11] ベースのモジュールに置き換えた Laplacian Pyramid Translation Transformer (LPTT) を提案する．これにより，提案手法は LPTN で生じていたアーティファクトを解消しつつ，性能と効率の両立を達成した．具体的には，ラプラシアンピラミッドレベルが 6 の場合，提案手法は LPTN に対して MIT Adobe FiveK データセット [48] における PSNR 値を 3.75 dB 向上させることが可能であり，4K 画像 50 枚を平均して 0.022 秒で処理できるほど高速である.

1.2.2　動画のノイズ除去の研究目的

動画のノイズ除去に関して，本研究では生成動画の品質と推論速度の両方を高い水準で実現することを目的とした，GPU と深層学習ベースの動画ノイズ除去ネットワークである Pseudo Temporal Fusion Network (PTFN) を提案する．また，本研究では新たに Pseudo Temporal Fusion (PTF) というモジュールを提案しており，これは Temporal Shift Module (TSM) [2] と組み合わせることで，時間的な関係性を擬似的に捉えるモジュールである．PTF は Conv3D のような時間軸方向の計算を直接行わないため，性能とネットワークの軽量化に貢献している．さらに，PTFN では ConvNeXt ベースの [49]ConvBlock を採用しており，これも性能と軽量さの両立に寄与している．しかし，[49,50] で提案されている ConvBlock が動画のノイズ除去に適しているかは検証されていないため，本研究では動画ノイズ除去に適した ConvBlock の構造を探索している．

PTFN は従来手法と比較して計算量を大幅に削減しつつ，生成画像の品質を向上させることに成功しており，PyTorch での処理時間も優れている．さらに，PTFN はメモリ消費の面でも優れており，NVIDIA RTX 3090 24GB で 1920×1080 の動画を高速に処理可能である．また，より軽量なバージョンである PTFN Half は，2560x1440 の 2K 動画も処理できる．

1.2.3　姿勢変換の背景

姿勢変換とは，ソースの人物画像とその姿勢情報，ターゲットの姿勢情報を用いて，ターゲットの姿勢をとった人物画像を生成するタスクである．Ma ら [32] は，まず粗い画像を生成し，その後精細な画像を作成する手法を提案した．Esser ら [33] は，U-Net [5] を使用して，人物の外観と姿勢を分離する手法を提案した．しかし，純粋な CNN ベースの手法では，複雑な姿勢変換に対応しきれないという問題があった．そこで，Zhu ら [34] は，段階的に人物画像を生成し，姿勢に基づいて外観を調整する手法を提案した．さらに，Tang ら [35] は，姿勢と外観の特徴を交差させるモジュールを用いた手法を提案した．

Li ら [36] や Ren ら [3] は，オプティカルフローを推定してソース画像のワーピングを行う手法を提案したが，大きな姿勢変化やオクルージョンが存在する場合，オプティカルフローの推定に失敗し，質の低い画像が生成されるという問題がある．また，オプ

ティカルフローの推定はしばしば生成ネットワーク本体とは別に学習されるため，ハイパーパラメータの調整に手間がかかり，学習時間も増加する．

Ma ら [37]，Li ら [38]，Zhang ら [39] は，追加の姿勢情報を利用したネットワークを提案し，姿勢変換の品質を向上させた．彼らの手法では，まずパースマップを推定してから最終的な人物画像を生成する．しかし，これらの手法で用いるパースマップの準備には労力がかかり，さらにパースマップ自体の信頼性にも課題がある．Zhang ら [40] は，Siamese Network [41] を活用して恒等写像を学習する補助タスクを設定することで，ネットワークがテクスチャ情報を効率的に取得できる手法を提案した．しかし，補助タスクの学習には追加のハイパーパラメータが必要であり，学習の複雑さが増えるという課題がある．

まとめると，最近の手法の多くは，姿勢変換が抱える問題に対して追加の学習データやタスクを導入して取り組んでいる．しかし，これらのアプローチは，追加データの準備やタスク追加によるハイパーパラメータや学習ステップの増加により，実用性やネットワークの軽量化が妨げられている．

1.2.4 非ペア画像変換の背景

非ペア画像変換は，非ペア画像データセットにおける画像変換タスクであり，ペア画像を用いる変換に比べ，基礎的なコンテンツを維持しつつ変換することが困難である．しかし，近年の研究 [7, 42–47] により，基礎的なコンテンツを維持しつつ変換する優れた手法が提案されている．Zhu ら [42] は，ソースドメインからターゲットドメインへ変換し，再びソースドメインへ戻す際に再構成損失を最小化する Cycle Consistency Loss を提案し，深層学習ネットワークが非ペア画像変換を行えるようにした．Liu ら [45] は，ドメイン間に共有潜在空間の仮定を置くことで非ペア画像変換を可能にした．また，Cycle Consistency Loss に代わる学習方法や制約も提案されている．Park ら [43] は，コントラスト学習によって入力画像と変換画像の相互情報量を最大化することで非ペア画像変換を行う手法を提案した．Fu ら [46] は，幾何的な変換に可換となるように生成器を制約する Geometry Consistency を提案した．Xu ら [44] は，入力画像に対する最大敵対的摂動に対して生成器が可換となるような Maximum Spatial Perturbation Consistency を提案した．

既存手法の多くは，非ペアデータにおいて基礎的なコンテンツを維持しつつ画像を変換することにフォーカスしており，形状の変換に対応できないことや背景が崩壊するという問題が残されている．この問題に対処することで，ネットワークの変換能力を変換

すべき領域に集中させ，ネットワークを軽量に保ちながら変換性能を維持することが可能となる．Tang ら [7] は，前景を抽出する注意マップにより，変換すべき前景と変換すべきでない背景を分離する AttentionGAN を提案し，背景の崩壊に対応することを試みた．しかし，AttentionGAN の前景と背景の分離精度は低く，しばしば分離に失敗している．これは，AttentionGAN の前景と背景の分離が画像変換と同時に学習されており，バイアスのあるデータセットから悪影響を受けたことが原因だと考えられる．また，AttentionGAN は形状が変化する場合には効果的な注意マップを抽出できない．ネットワークを軽量に保ちながら変換性能を維持できるためにも，Attention GAN が抱える問題に対処する必要がある．

1.3　本論文の構成

本論文は以下のように構成される．2 章では本研究の内容をカバーする基礎理論を解説し，3 章では本研究で取り組むタスクにおける従来手法を解説する．また，4 章，5 章，6 章，7 章ではそれぞれ，本研究が取り組む具体的なタスクである，フォトレタッチ，姿勢変換，動画のノイズ除去，非ペア画像変換における提案手法の紹介とその有効性の検証を行う．最後に，8 章では，各タスクで提案・検証された内容と本研究が提起した問題との関連などを記述しつつ，論文の総括を行う．

第 2 章

基礎理論

2.1　デジタルメディア

2.1.1　デジタル画像

デジタル画像は，デジタル形式で表された画像であり，一般的には 0 から 255 の範囲の整数値を格納した $(H \times W \times C)$ の形のテンソルで表される．ここでは，H は画像の縦幅，W は画像の横幅，C は色を表現するためのチャンネル数を表している．

2.1.2　デジタル動画

デジタル動画は，デジタル形式で表された動画であり，デジタル画像のシーケンスとして表される．一般的にデジタル動画はシーケンス長が T である場合，$(T \times H \times W \times C)$ のテンソルとして表されている．

2.2　深層学習

深層学習は多くの階層型ニューラルネットワークを学習することでデータの特徴を獲得することを目的としている．一般的に深層学習は多層にするほど表現能力が上がるが，単純に層の数を増やすだけでは勾配消失・爆発のような問題が発生してしまう．こういった問題に対処するために層と層の間に正規化層を挿入するといった工夫がなされている．

2.2.1 畳み込みニューラルネットワーク（CNN）

畳み込みニューラルネットワーク（CNN）[51] は，学習可能なフィルタを備えた畳み込み層を有するネットワークである．CNN は入力データから空間的な特徴を抽出する畳み込み層を複数組み合わせることで強力な表現能力を獲得している．CNN を含むニューラルネットワークは効率的に非線形の表現を学習するために正規化層や活性化関数が必要である．

2 次元の畳み込み層は入力データが $x(i,j)$ とし，畳み込みフィルタのパラメータを $w(s,t)$ とした場合，畳み込み層の出力 $y(i,j)$ は以下のように表される．

$$y(i,j) = \sum_{t=-Q(k,2)}^{Q(k,2)} \sum_{s=-Q(k,2)}^{Q(k,2)} x(i+s, j+t) w(s,t) \tag{2.1}$$

ただし，i,j は画像の座標，s,t は畳み込み層のパラメータ内の位置，k は畳み込みフィルタのフィルタサイズ，$Q(a,b)$ は a を b で割った商を表している．2 次元の畳み込み層は，空間的な方向のだけでなくチャンネル方向にも計算が行われるが，その場合は以下のように表される．式 (2.1) で表される計算は Depthwise Convolution と呼称される．[52]

$$y(i,j,c') = \sum_{c=1}^{C} \sum_{t=-Q(k,2)}^{Q(k,2)} \sum_{s=-Q(k,2)}^{Q(k,2)} x(i+s, j+t) w(s,t,c,c') \tag{2.2}$$

ただし，C は入力テンソルのチャンネル数で，c' は出力チャンネルのインデックスである．2 次元の畳み込み層の入力チャンネル数が C，出力チャンネル数が C' である場合，$(H \times W \times C)$ の形のテンソルで表された画像に対して，計算量は $O(k^2HWCC')$ となる．このとき，バイアスの加算由来の計算は計上されていない．デジタル動画に関して，フィルタサイズが $k \times k \times k$ で，入力チャンネル数が C，出力チャンネル数が C' である場合，$(T \times H \times W \times C)$ の形のテンソルで表された画像に対して，計算量は k^3THWCC' となる．

2.2.2 活性化関数

CNN を含むニューラルネットワークは効率的に非線形の表現を学習するために正規化層や活性化関数が必要である．活性化関数には様々な種類が存在し，Sigmoid 関

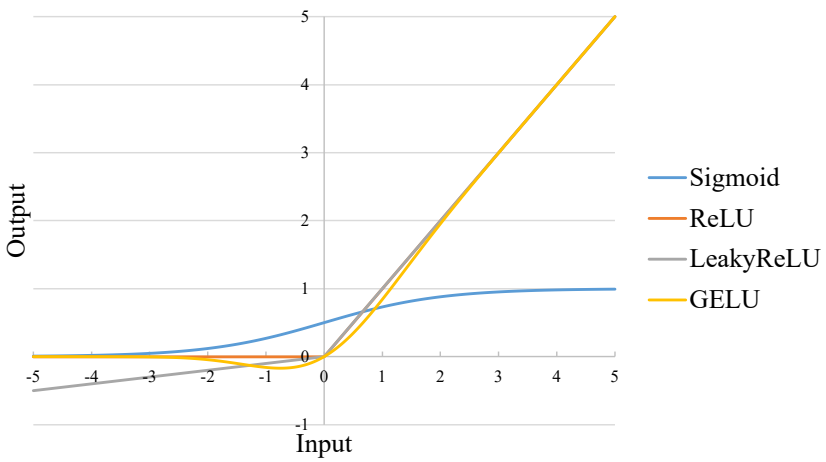


図 2.1　活性化関数の比較.

数 [53]，ReLU 関数 [54] や LeakyReLU 関数 [55]，GELU 関数 [56] のような関数が提案されている．それぞれの活性化関数の比較を図 2.1 に示す.

Sigmoid 関数

Sigmoid 関数は深層学習の初期や 2 値問題の出力，Channel Attention 層 [57] の出力によく用いられている関数であり，以下の式で表される.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

ReLU 関数

ReLU 関数は現在の深層学習分野においてよく用いられている活性化関数であり，Sigmoid 関数が陥っていた勾配消失の問題を軽減できる．ReLU 関数の式は以下に示される.

$$\mathrm{ReLU}(x) = \begin{cases} 0 & (x < 0) \\ x & (x \geq 0) \end{cases} \tag{2.4}$$

LeakyReLU 関数

LeakyReLU 関数は ReLU 関数の派生であり，特に勾配消失の問題が深刻である敵対的生成ネットワーク（2.2.6 で詳述）において用いられている活性化関数であり，以下の

ように示される．

$$\mathrm{LeakyReLU}(x) = \begin{cases} \alpha x & (x < 0) \\ x & (x \geq 0) \end{cases} \tag{2.5}$$

ただし，α は任意の正数であり，一般的には 1.0 以下の正数が用いられる．

GELU 関数

GELU 関数は ReLU 関数の派生であり，ReLU 関数より滑らかな微分関数を有している．GELU の式は以下に示される．

$$\mathrm{GELU}(x) = x\Phi(x) = x \cdot \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right] \tag{2.6}$$

ただし，erf は誤差関数である．誤差関数の値は初等計算で求められないため，実際は以下のように近似した関数を用いている．

$$\mathrm{GELU}(x) \sim \frac{1}{2}x\left(1 + \tanh\left[\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right]\right) \tag{2.7}$$

2.2.3 正規化層

深層学習ネットワークの訓練では，ネットワークパスが長すぎる場合に，逆誤差伝搬によって計算した勾配が爆発したり消失したりする問題がある．こういった問題に対し，中間層の特徴マップを正規化するための層を正規化層と呼称する．本項では深層学習の分野において一般的に用いられている正規化層の紹介をする．

Batch Normalization

Batch Normalization は [58] で提唱された正規化層であり，ミニバッチ単位でデータの正規化を行うことで安定した訓練やより早い収束を可能にした．x を入力された場合の Batch Normalization の出力 y は以下に示される．

$$y = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \odot \gamma + \beta \tag{2.8}$$

ただし，$\odot$ はアダマール積，γ, β は学習可能なパラメータであり，ϵ はゼロ除算を避けるために設定される小さな正の実数である．また，μ_B, σ_B はミニバッチ単位計算した平均と分散であり，ミニバッチ数が N の場合，以下のように計算される．

$$\mu_B = \frac{1}{N}\sum_i^N x_i \tag{2.9}$$

$$\sigma_B = \sqrt{\frac{1}{N}\sum_i^N (x_i - \mu_B)^2} \tag{2.10}$$

Layer Normalization

Layer Normalization は，[59] で提唱された正規化層であり，チャンネル単位でデータの正規化を行っている．この正規化層は，Batch Normalization が抱えていた，ミニバッチ数が小さい場合に学習が不安定になる問題 [50] や，複数 GPU における訓練の問題を解決しているため，現在最先端の手法においてよく用いられている．[49, 50]

$$y = \frac{x - \mu_L}{\sqrt{\sigma_L^2 + \epsilon}} \odot \gamma + \beta \tag{2.11}$$

ただし，$\odot$ はアダマール積，γ, β は学習可能なパラメータであり，ϵ はゼロ除算を避けるために設定される小さな正の実数である．また，μ_L, σ_L はチャンネル単位で計算した平均と分散であり，特徴マップのチャンネル数が N の場合，以下のように計算される．

$$\mu_L = \frac{1}{N}\sum_i^N x_i \tag{2.12}$$

$$\sigma_L = \sqrt{\frac{1}{N}\sum_i^N (x_i - \mu_L)^2} \tag{2.13}$$

2.2.4 損失関数

深層学習のネットワークを訓練する際に，参考となるデータとの乖離の程度をモデル化することで，ネットワークのパラメータ更新の指針を立てる．本項では，参考となるデータとの乖離の程度をモデル化のための損失関数の解説をする．

L1 & L2 Loss

画像から画像への変換のように，入出力のドメインが同じである場合，入出力に対して単純な差分を取る損失関数が広く用いられている．このような損失関数は L1 Loss や L2 Loss と呼ばれ，以下のように計算される．

$$L_{l1} = \frac{1}{HWC}\sum_i^H \sum_j^W \sum_k^C ||y(i,j,k) - x(i,j,k)||_1 \tag{2.14}$$

$$L_{l2} = \frac{1}{HWC} \sum_{i}^{H} \sum_{j}^{W} \sum_{k}^{C} ||y(i,j,k) - x(i,j,k)||_2 \tag{2.15}$$

ただし，H, W, C はそれぞれ画像テンソルの高さ，幅，チャンネル数であり，$x(i,j,k), y(i,j,k)$ はそれぞれネットワークの出力と参照データである．

Perceptual Loss

Perceptual Loss は [60] で提案された損失関数であり，画像復元・変換の分野においてよく用いられている．この損失関数では，よく訓練された深層学習ネットワークは画像の特徴を捉える操作と解釈することが可能であり，その考えに基づいて画像の特徴が近しいかどうかを測定している．具体的には VGG [15] などの訓練済みの深層学習ネットワークを用いて以下のように計算される．

$$L_{perc} = \sum_{i} ||\phi_i(y) - \phi_i(x)||_1 \tag{2.16}$$

ただし，$\phi_i(x)$ は VGG の i 層目の出力を表す．

Style Loss

Style Loss は [60] で提案された損失関数であり，スタイル変換によく用いられている関数であるが，画像復元・変換の分野においても用いられている．Style Loss はネットワークの出力画像と正解画像を VGG [15] などの訓練済みの深層学習ネットワークに通した中間出力に対して，グラム行列を計算し，その差を計算する．グラム行列は正規化された場合の分散共分散行列とみなすことが可能であるため，グラム行列同士の差によって，統計的な特徴がどれだけ似通っているかを測定している．

$$L_{style} = \sum_{i} ||\mathrm{Gram}_i^{\phi}(y) - \mathrm{Gram}_i^{\phi}(x)||_1 \tag{2.17}$$

2.2.5 ニューラルネットワークの学習

勾配降下法

勾配降下法は深層学習ネットワークのパラメータ最適化の問題を解く際に用いられている方法であり，Θ でパラメータが与えられるネットワークに対して，損失関数が $L(\Theta)$ であるとき，以下のように計算される．

$$\Theta^{(t+1)} = \Theta^{(t)} - \eta \nabla L(\Theta^{(t)}) \tag{2.18}$$

$$\nabla L(\Theta^{(t)}) = \frac{\partial L(\Theta^{(t)})}{\partial \Theta^{(t)}} = \left[\frac{\partial L(\Theta^{(t)})}{\partial \Theta_1^{(t)}} \cdots \frac{\partial \Theta_{M-1}^{(t)}}{\partial \Theta_M^{(t)}} \right] \tag{2.19}$$

この時，t はパラメータ更新のイテレーションの番号で，η は学習率であり，M は更新が必要なパラメータの総数である．勾配降下法では損失関数のパラメータに対する微分をもとに，損失関数が小さくなるようにパラメータを更新している．パラメータの微分を求める際は，微分の連鎖律を用いて微分の計算を層ごとに伝播させていく誤差逆伝播法を用いている．

Adam

Adam [61] は，パラメータ最適化のために提案された手法であり，以下のようにパラメータを更新する．

$$\mathbf{m}^{(t+1)} = \beta_1 \mathbf{m}^{(t)} + (1-\beta_1)\nabla L(\Theta^{(t)}) \tag{2.20}$$

$$\mathbf{v}^{(t+1)} = \beta_2 \mathbf{v}^{(t)} + (1-\beta_2)\nabla L(\Theta^{(t)})^2 \tag{2.21}$$

$$\hat{\mathbf{m}} = \frac{\mathbf{m}^{(t+1)}}{1-\beta_1} \tag{2.22}$$

$$\hat{\mathbf{v}} = \frac{\mathbf{v}^{(t+1)}}{1-\beta_2} \tag{2.23}$$

$$\Theta^{(t+1)} = \Theta^{(t)} - \alpha \frac{\hat{\mathbf{m}}}{\sqrt{\hat{\mathbf{v}}} + \epsilon} \tag{2.24}$$

Adam は単純な勾配降下法よりも収束が早く安定しているため，数多くの手法における最適化手法として採用されている．

2.2.6 敵対的生成ネットワーク

敵対的生成ネットワーク（Generative Adversarial Network, GAN）[62] は，敵対的な学習によって学習がされる生成モデルである．GAN の訓練スキームに関して，GAN では Generator と Discriminator という 2 つのネットワークを訓練する．Generator と Discriminator は交互に学習され，それぞれがお互いを欺くように訓練される．具体的には，Generator を $G(\cdot)$，Discriminator を $D(\cdot)$ としたとき，以下のような訓練を行う．

$$\min_{\Theta_G} \max_{\Theta_D} (\mathrm{E}[\log D(y)] + \mathrm{E}[\log(1 - D(G(x)))]) \tag{2.25}$$

GAN は，画像生成や画像変換 [8, 42, 63] によく用いられている手法であり，様々なタスクにおいて優秀な性能を達成している．[1, 40, 43, 44]

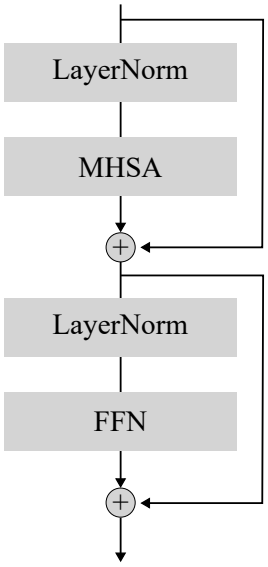


図 2.2　ベーシックな Transformer を用いた特徴抽出モジュール．ここで，Multi Head Self Attention は MSHA，Feed Forward Network は FFN という略称で表されている．

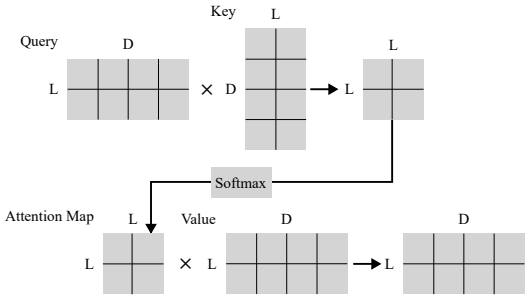


図 2.3　Multi Head Self Attention の詳細．ここで，L, D はそれぞれシーケンス長とチャンネルの深さを表しており，$\times$ はテンソル積を表している．

2.2.7　画像処理における Transformer

Transformer [10] は，自然言語処理分野において広く用いられている [64, 65] 技術であり，近年は画像処理の分野にも応用されている．[12, 13, 66] Transformer は画像処理の深層学習において従来よく用いられていた CNN とは異なり，畳み込みではなく Attention によって主な特徴抽出を行っている．ベーシックな Transformer を用いた特徴抽出モジュールは図 2.2 に示すように表現できる．

Transformer のモジュールは図 2.2 に示される，Layer Normalization (LayerNorm)，Multi Head Self Attention (MHSA)，Feed Forward Network (FFN) が含まれ，レイヤー間には加算がされる．各レイヤーの詳細な解説を以下に記す．

Multi Head Self Attention (MHSA)

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.26}$$

$$\begin{aligned} \mathrm{MultiHead}(Q,K,V) &= \mathrm{Concat}(\mathrm{head}_1,\ldots,\mathrm{head}_h)W^O \\ \text{where } \mathrm{head}_i &= \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.27}$$

Multi Head Self Attention (MHSA) は，(2.27) に示される，Transformer アーキテクチャの中核要素である．MHSA は，入力された特徴マップに対して，チャンネル方向に分割したヘッドごとに，図 2.3 や (2.26) のような自己注意の計算を行う．MHSA は CNN とは異なり，特徴マップ全体にわたって Attention Map の値を算出するため，離れたピクセル同士の特徴関係を捉えられる．

Feed Forward Network

$$FFN(x) = \mathrm{ReLU}(xW_1 + b_1)W_2 + b_2 \tag{2.28}$$

Feed Forward Network (FFN) は，MHSA のあとに計算される層であり，画素ごとに (2.28) に示されるような計算を行う．このとき，W_1, W_2 はネットワークの重みで，b_1, b_2 はバイアスである．実際の実装においてこの操作は，カーネルサイズが 1×1 の畳み込みで実装される．

Positional Encoding

Positional Encoding (PE) は，Transformer の Multi Head Self Attention (MHSA) の前に挿入されるモジュールである．MHSA はシーケンスの位置関係を認識できないため，Positional Encoding では (2.29) および (2.30) のような，シーケンスの位置によって値が変化する関数の値を MHSA の入力に加算することで位置情報を埋め込む操作を行う．

$$\mathrm{PE}_{(l,2i)} = \sin(l/10000^{2i/d_{model}}) \tag{2.29}$$

$$\mathrm{PE}_{(l,2i+1)} = \cos(l/10000^{2i/d_{model}}) \tag{2.30}$$

このとき，l, i はそれぞれ，シーケンス，チャンネルの深さを示すインデックスで，d_{model} はチャンネルの最大深さである．実際に PE で埋め込まれる値は図 2.4 のように表される．

Positional Encoding は，タスクの性能の向上に非常に有効であるが，画像処理の分野においては未知の解像度の画像に対応できないといった問題が存在する．Conditional Positional Encoding (CPE) [67] は，画像のような多次元テンソルに対する位置エンコーディングの手法の 1 つである．CPE は対象画素の周囲の画素値から動的にエンコーディングの値を決定することで，未知の解像度の画像においても妥当な値を埋め込

める手法である．実際の実装においてこの操作は，任意のカーネルサイズの Depthwise Convolution と加算で実装される．

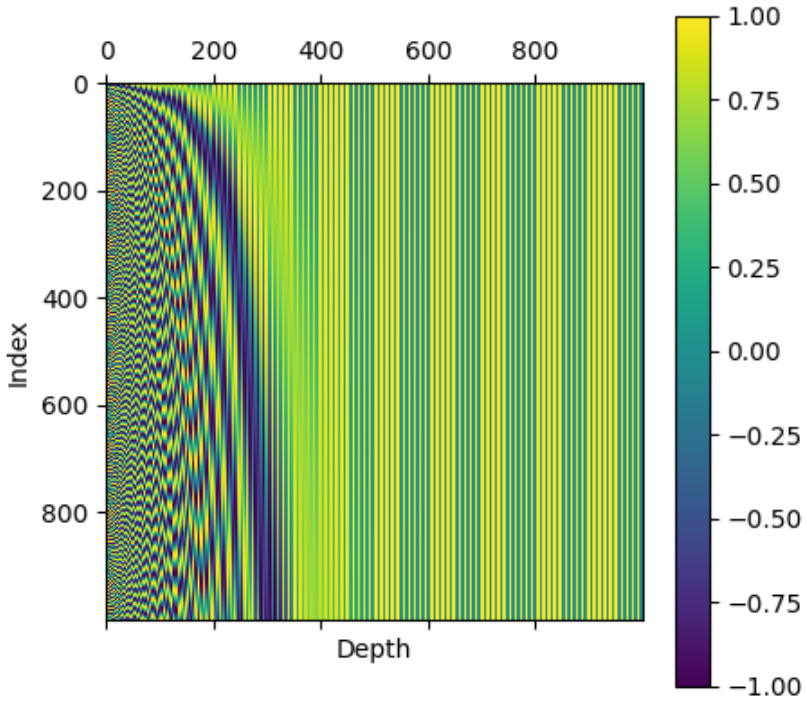


図 2.4　Positional Encoding の値．Index はシーケンス方向，Depth はチャンネル方向を表している．

Transformer の優位点・問題点

Transformer は広い受容野を持つため，離れたピクセル同士の関係性を捉えることが可能である．また，Self Attention を Cross Attention に変更することで，強力な条件付けを行えることも大きな優位点である．しかし，Transformer は解像度に対して計算量が大幅に増大するため，高解像度画像に対する推論が困難であったり，CUDA のような低レベルにおける処理が CNN ほど最適化されていないことから，実際の計算速度が遅くなる傾向にあるといった欠点も存在する．

Axial Transformer

Axial Attention [11] は，多次元テンソルに対する Attention の手法の一つであり，Axial Transformer [11] は，Axial Attention を Multi Head Attention に採用した Transformer である．画像処理分野における Transformer の応用に関して，Vision Transformer (ViT) [9] は, 広い受容野を持った優れた性能のネットワークである．しかし，ViT では Attention の計算のために画像をパッチに分割したあとにテンソルを平坦化する．よって，画像の入力解像度が大きくなると計算量が急速に大きくなるという問題があった．Attention の計算量を減らすための手法として，Image Transformer [66] や Swin Transformer [12] が，提案されているがどちらも縦横方向の受容野を狭めるため，広い受容野という Transformer の利点を損ねている．したがって，大きな姿勢の変換が起こりうる姿勢変換のタスクにおいては品質に影響をもたらす可能性がある．これに対し，Axial Attention は，入力テンソルの列・行毎に Attention の計算を行うため，縦横方向の受容野を狭めることなく，計算量を効率的に減らすことが可能である．具体的には，入力サイズが $R \times R \times C$ のテンソルの場合，通常の Attention における計算量が $O(R^4C)$ であるのに対し，Axial Attention では，$O(2R^3C)$ となる [11].

2.3 評価指標

本項では実験結果を評価するための手法について解説する．実験結果を評価するためには，主に定量評価と定性評価があり，それぞれに特徴が存在する．以下，それぞれの特徴を説明する．

2.3.1 定量評価

定量評価は，実験結果を数値的な指標を用いて評価することであり，取り組むべきタスクに対して適切な指標を用いて手法の良し悪しを評価できる．定量評価はテストデータセット全体を対象に測定されるため，データサイズが非常に大きい深層学習において，客観的な視点から比較する有効な手段である．しかし，全く適切な指標を選択することは難しいため，複数の指標を組み合わせて手法の良し悪しを総合的に考慮することが重要である．以下，画像変換の分野においてよく用いられている評価指標を紹介する．

Peak Signal to Noise Ratio (PSNR)

Peak Signal to Noise Ratio (PSNR) は，深層学習が生成した画像 $\hat{I}$ と，訓練データの正解画像 I との間の歪みを測定した指標であり，以下のように計算される．

$$\mathrm{MSE} = \frac{1}{HW}\sum_{i}^{H}\sum_{j}^{W}\sum_{c}(\hat{I}_c(i,j) - I_c(i,j)) \tag{2.31}$$

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right) \tag{2.32}$$

ここで，H, W は画像の縦・横であり，c はチャンネルを表している．また，MAX は画像テンソルの取りうる最大値であり，一般的には 255 または 1.0 である．PSNR は両画像の MSE が小さい，つまり歪みが小さいほど高い値を取ることを意味する．

Structural Similarity (SSIM)

SSIM は画像の構造的な違いを測定する手法であり，以下のように計算される．SSIM はスライディングウィンドウの小領域ごとに，以下の計算を行う．

$$\mathrm{SSIM} = \frac{(2\mu_{\hat{I}}\mu_I + C_1)(2\sigma_{\hat{I}I} + C_2)}{(\mu_{\hat{I}}^2 + \mu_I^2 + C_1)(\sigma_{\hat{I}}^2 + \sigma_I^2 + C_2)} \tag{2.33}$$

ここで，$\mu_{\hat{I}}, \mu_I, \sigma_{\hat{I}}, \sigma_I$ は $\hat{I}, I$ の画素値の平均，標準偏差を表しており，$\sigma_{\hat{I}I}$ は $\hat{I}, I$ の共分散，C_1, C_2 は定数である．PSNR は MSE を用いて歪みを測定する一方，画像全体の色味などが評価値に大きく影響するが SSIM は小領域ごとの統計的な特徴をもとに指標値を算出しているため，画像の構造などの類似度を反映している．

Learned Perceptual Image Patch Similarity (LPIPS)

Learned Perceptual Image Patch Similarity (LPIPS) は [68] で提案された評価指標であり，人間のような直感的な画質を測定することを目的としている．LPIPS は，よく訓練された CNN が画像の特徴を抽出する能力を利用して，深層学習が生成した画像 $\hat{I}$ と，訓練データの正解画像 I を訓練済みの CNN ネットワークに入力し，その中間出力同士で距離を計算する．訓練済み CNN ネットワークには AlexNet [4] や VGG [15] がよく用いられる．深層学習のネットワークでは入力した画像の視覚特性が反映された特徴マップの間で距離を計算するため，PSNR や SSIM より人間の主観に近い特性を測れる．

Frechet Inception Distance (FID)

Frechet Inception Distance (FID) は，対になった正解画像が得られないタスクにおいても画像の品質を測定することが可能な指標である．FID は LPIPS と同様に，よく訓練された CNN が画像の特徴を抽出する能力を有していることを活用している．FID は生成画像と正解画像それぞれの画像群 $\hat{\mathbb{I}}, \mathbb{I}$ を Inception V3 [69] に通したときに得られた，潜在空間同士の距離を測定することで算出される．$\mu_{\hat{\mathbb{I}}}, \mu_{\mathbb{I}}, \Sigma_{\hat{\mathbb{I}}}, \Sigma_{\mathbb{I}}$ をそれぞれ生成画像群と正解画像群の潜在空間の平均・分散であるとすると，FID は以下のように算出される．

$$\mathrm{FID} = ||\mu_{\hat{\mathbb{I}}} - \mu_{\mathbb{I}}||^2 + \mathrm{Tr}(\Sigma_{\hat{\mathbb{I}}} + \Sigma_{\mathbb{I}} - 2\sqrt{\Sigma_{\hat{\mathbb{I}}}\Sigma_{\mathbb{I}}}) \tag{2.34}$$

FID は LPIPS と同様に深層学習ネットワークを通じて出力間で距離を計算しているため，人間の主観に近い尺度で測定できる．しかし，FID は LPIPS とは異なり，評価データがペア画像になっていない場合でも測定が可能であるといった利点がある．

ネットワークの軽量さを測定するための指標

定量的な評価指標には，生成変換した画像の品質だけではなく，深層学習ネットワークの軽量さを測定するものが存在する．深層学習の研究分野においては，実世界への応用のために軽量なネットワークが求められるため，こういった指標値を用いてそれを評価している．本研究では主に以下の 3 つの指標を用いるが，それぞれに対して利点や欠点が存在する．

ネットワークパラメータ数：深層学習ネットワークのパラメータ数はネットワークのスケールを示す一方法である．ネットワークパラメータはネットワークの大きさを簡単に表せるが，ネットワーク内部の計算の複雑さや実際の推論速度を考慮できないといった欠点も存在する．

処理時間の実測：条件を揃えて実際の処理時間を計測することでそのネットワークのスケールを測定する手法も存在する．実際の処理時間は，実社会への応用可能性を定量化する際に信頼がおける数値になる．しかし，フレームワークのバージョンやコードの書き方，計算サーバの状態のようなネットワーク以外の要素に大きく左右されるため，純粋にネットワークのスケールを示すものではないことに注意が必要である．

計算量：Multiply-accumulation Operations (MACs) や Floating Point Operations (FLOPs) のような計算量によってネットワークのスケールを計算できる．計算量は，ネットワークのパラメータ数よりもネットワーク内部の計算の複雑さを反映することが

可能であるが，必ずしも実測の速度とは比例しない．

2.3.2　定性評価

定性評価は，実験結果を実際の生成画像などを用いて評価することである．定量評価で用いられている指標が完璧に画質を表すものではないことや，提案手法の有効性を視覚を交えて主張するためにこれらは用いられる．定性評価では客観性のために複数人にアンケートを取ることもしばしば行われる．

第 3 章

従来手法

本章では，本研究が取り組む具体的なタスクにおける従来手法を紹介する．

3.1 フォトレタッチの従来手法

3.1.1 Laplacian Pyramid Translation Network (LPTN)

Liang ら [1] は，高解像度画像に対しても高速でフォトレタッチを行うことが可能な，Laplacian Pyramid Translation Network (LPTN) を提案した．LPTN は，ラプラシアンピラミッドによって入力画像を分解し，低周波成分と高周波成分で変換ネットワークの深さを変えることで高解像度の画像を高速に推論することを可能にした．LPTN は，4K 解像度の画像も変換できる軽量さを有しているが，変換画像に粒状のアーティファクトを生成する傾向にあり，定量・定性的な品質を損ねており，軽量さを成立させるために性能を犠牲にしていると言える．LPTN の具体的な変換画像は 4 章の図 4.4 に示されている．

ネットワーク構造

LPTN のネットワークは図 3.1 に示されている．LPTN では，まず入力画像 $I_0 \in \mathbb{R}^{h \times w \times 3}$ をラプラシアンピラミッドに分解し，$H = [h_0, h_1, \cdots, h_{L-1}]$ と低周波残差画像 I_L を得る．そして，低解像度の I_L は深い畳み込みで $\hat{I}_L$ に，より解像度が高い要素は図 3.1 のように，低解像度の出力に対して適応的に変換を行う．最終的に，変換された低周波成分と調整された高周波成分から画像を再構成する．

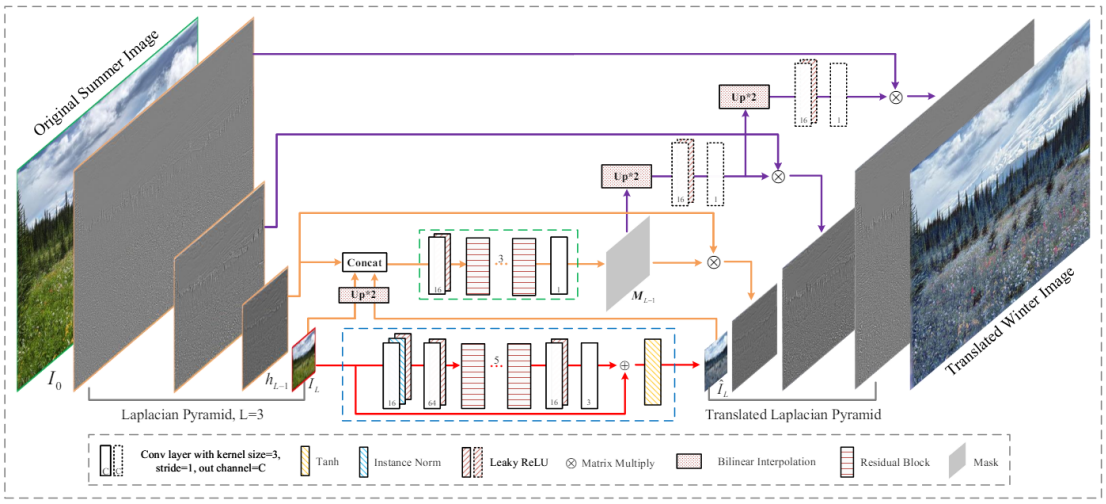


図 3.1　LPTN のパイプライン．高解像度画像 $I_0 \in \mathbb{R}^{h \times w \times 3}$ が与えられたとき，まずこれをラプラシアンピラミッドに分解する（例えば，$L = 3$）．赤い矢印：低周波成分 $I_L \in \mathbb{R}^{\frac{h}{2^L} \times \frac{w}{2^L} \times c}$ について，軽量なネットワークを使用してこれを $\hat{I}_L \in \mathbb{R}^{\frac{h}{2^L} \times \frac{w}{2^L} \times c}$ に変換する．茶色の矢印：高周波成分 $h_{L-1} \in \mathbb{R}^{\frac{h}{2^{L-1}} \times \frac{w}{2^{L-1}} \times c}$ を適応的に変換するために，高周波および低周波成分の両方に基づいてマスク $M_{L-1} \in \mathbb{R}^{\frac{h}{2^{L-1}} \times \frac{w}{2^{L-1}} \times 1}$ を学習する．紫の矢印：より高解像度の他の成分については，学習したマスクを段階的にアップサンプリングし，フォトリアリスティックな再構築の能力を維持するために軽量な畳み込みブロックで微調整する．図は [1] より引用された．

損失関数

LPTN の学習には複数の損失関数が用いられており，まずは入力画像と出力画像の L2 Loss を用いた再構成損失を採用している．また，LSGAN [70] の目的関数とマルチスケール Discriminator を使用した敵対的損失も導入されている．これらの損失を重み付けして組み合わせた損失 $L = L_{recons} + \lambda L_{adv}$ を最小化することで，モデルを最適化する．ただし，λ は二つの損失のバランスを調整するパラメータである．

3.2 動画のノイズ除去の従来手法

3.2.1 FastDVDNet

Tassano ら [6] は，従来の方法と比較して同等以上の性能を達成しつつ，大幅に高速な処理を実現した FastDVDNet を提案した．FastDVDNet は，チャンネル方向にフレームを結合して処理するため，モーション推定を行わずにアーキテクチャの特性によってフレーム間の関係を処理することが可能である．また，単一のネットワークモデルで幅広いノイズレベルに対応し，リアルタイムに近い処理速度を実現するため，実用的なアプリケーションに適している．しかし，FastDVDNet は動画テンソルを直接処理しないため効率的な計算が可能であるが，スライディングウィンドウ方式で動画を処理する冗長な構造をしており，性能と軽量さをより高度に両立させる余地がある．

ネットワーク構造

FastDVDNet はカスケード型の 2 段階の U-Net 構造を採用している．各段階は修正された U-Net 型のブロックで構成されており，5 つの連続フレームをチャンネル方向に結合したものを入力とし，中央フレームのノイズを除去する．また，FastDVDNet は画像サイズと同じ解像度のノイズマップをチャンネル方向に結合した追加入力として使用することで性能を高めている．また，アップサンプリングに Pixel Shuffle [71] を採用することで格子状のアーティファクトを軽減している．

損失関数

FastDVDNet は L2 Loss を使用している．損失関数は $L = \frac{1}{2m_t} \sum_{j=1}^{m_t} ||\hat{I_t^j} - I_t^j||^2$ と定義され，ここで $\hat{I_t^j}$ はネットワークの出力，I_t^j はクリーンなフレーム，m_t は入力されたフレームの総数である．また，最適化には Adam [61] を使用している．

3.2.2 Bidirectional Streaming Video Denoising (BSVD)

Qi ら [18] は，前述した FastDVDNet の弱点を踏まえて，動画の高速なノイズ除去のための手法である Bidirectional Streaming Video Denoising (BSVD) を提案した．BSVD は過去と未来の双方向の時間的特徴伝播が可能な構造をしているため，従来手法よりも高速で高品質なノイズ除去が可能である．また BSVD は，FastDVDNet のよう

なスライディングウィンドウ方式の推論ではなく，Bidirectional Buffer Block (BBB) を用いたパイプライン方式の推論を行うため，オンラインストリーミング処理を可能であり，FastDVDNet が陥っているような冗長性の問題を解決した．しかし，BSVD のネットワークは計算量が高いという問題があり，GPU のメモリ消費も大きい．よって，動画時間が長い場合に実用性が制限されやすい．実際に BSVD は 24GB のメモリを有した GPU で 1080p（1920 × 1080）の動画を処理できず，十分な軽量さの水準に達していないと言える．

ネットワーク構造

BSVD のネットワークは図 3.2 に示される．BSVD はバックボーンとして 2 つの軽量 U-Net を使用する W-Net [72] 構造を採用している．Bidirectional Buffer Block を導入して時間的特徴の融合を行い，Batch Normalization [58] を除去して ReLU の代わりに ReLU6 を使用している．BSVD では，複数の Bidirectional Buffer Block を直列に接続することで大きな時間的受容野を実現し，入力フレームに対して 1 フレームずつ処理を行うパイプライン推論方式を採用している．

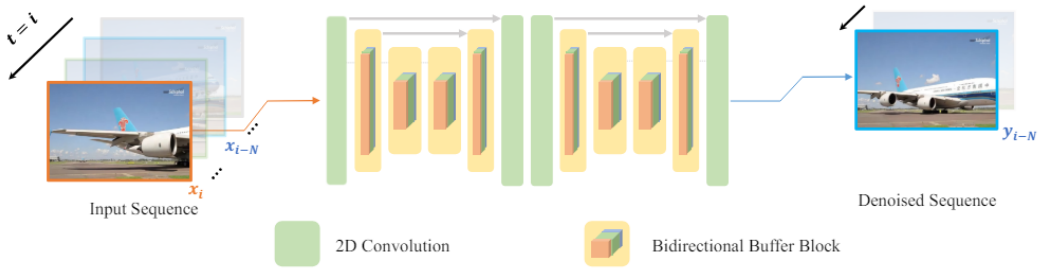


図 3.2　BSVD のパイプライン．BSVD は畳み込み層の間に Tempral Fusion Module (TSM) [2] を挿入した 2 つの軽量 U-Net からなる．推論中の時間ステップ i で，1 つのノイズフレーム x_i とそのノイズマップがネットワークに入力される．そしてネットワークは別のクリーンフレーム y_{i-N} を出力する．図は [2] より引用された．

損失関数

BSVD は FastDVDNet と同様に L2 Loss を使用している．

3.3 姿勢変換の従来手法

3.3.1 Global-Flow Local-Attention (GFLA)

Ren ら [3] は姿勢変換のための Global-Flow Local-Attention (GFLA) という深層学習の手法を提案した．従来の CNN はデータの空間的な変換能力が限られているという課題があったが，GFLA は Global Flow Field Estimator と Local Neural Texture Renderer という 2 つのモジュールから構成されたネットワークによってこの問題を改善した．前者はソース画像とターゲット画像間のグローバルな相関を計算してフローフィールドを予測し，後者は予測されたフローフィールドを使用してソース特徴とターゲット情報を関連付けて姿勢変換をする．GFLA は姿勢変換の他にフロー推定のタスクを追加することで姿勢変換の性能を向上させたが，その結果，追加のネットワークや学習コストが必要になり，大きなネットワークとなる．

ネットワーク構造

GFLA のネットワークは図 3.3 に示される．Global Flow Field Estimator (図 3.3 の下側) は，ソース画像，ソースポーズ，ターゲットポーズを入力として受け取り，フローフィールド w とオクルージョンマスク m を生成する．一方，Local Neural Texture Renderer (図 3.3 の上側) は，ソース画像，ターゲットポーズ，フローフィールド，オクルージョンマスクを入力として受け取り，ローカルアテンション機構を使用して特徴の空間変換を実行する．このプロセスでは，各位置で $n \times n$ のローカルパッチを抽出し，ソースとターゲットの関連付けを行うことで効果的な特徴変換を実現している．

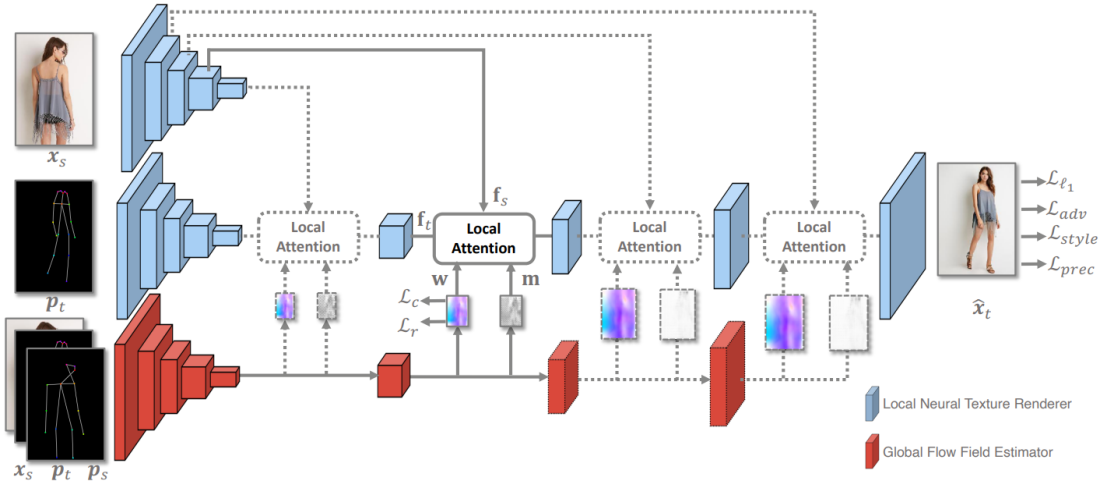


図 3.3　GFLA のパイプライン．図は [3] より引用された．

損失関数

GFLA では Global Flow Field Estimator と Local Neural Texture Renderer をそれぞれ学習していくが，損失関数もそれぞれ異なっている．Global Flow Field Estimator はサンプリング正確性損失と正則化損失を用いている．前者は抽出された特徴とグランドトゥルースの特徴間の相対コサイン類似度を計算し，後者はローカル領域での変換がアフィン変換に近くなるように制約をかけている．Local Neural Texture Renderer は，生成画像とグランドトゥルース間の L1 距離を計算する再構成損失，GAN フレームワークによる敵対的損失，事前学習済みネットワークの活性化マップ間の L1 Loss を計算する知覚損失，そして活性化マップから構築されたグラム行列間の誤差を計算するスタイル損失を使用している．

3.4　非ペア画像変換の従来手法

3.4.1　CycleGAN

Zhu ら [42] は，非ペア画像変換を実現するための手法 CycleGAN を提案した．CycleGAN では 2 つのドメイン間で双方向の変換を学習し，Cycle consistency loss を導入することで，非ペア画像変換を可能にしている．この手法は様々なタスクに適用可能で，タスクによっては教師あり手法に匹敵する結果を達成している．

ネットワーク構造

CycleGAN は 2 つの Generator と 2 つの Discriminator を使用している．Generator には Johnson らの手法 [60] をベースとしたアーキテクチャを採用し，Discriminator にはパッチサイズが 70×70 の PatchGAN [8] を採用している．また，ネットワーク内では Instance normalization [73] を使用している．

損失関数

CycleGAN の損失関数は主に 3 つの要素から構成され，以下のように示される．1 つ目は敵対的損失である．2 つ目は Cycle consistency loss であり，これは循環的に双方向の変換を行って復元された画像が元の画像に戻るような制約を加える損失関数である．さらに，一部のタスクでは追加で Identity mapping のロスを導入している．これはネットワークがターゲットドメインの画像を入力された際に変換せずに出力するという制約を加える損失関数である．これらの要素を組み合わせることで，非ペア画像変換を実現している．

3.4.2 Maximum Spatial Perturbation Consistency (MSPC)

Xu ら [44] は非ペア画像変換タスクのための新しい正則化手法である Maximum Spatial Perturbation Consistency (MSPC) を提案している．MSPC は，最大空間摂動関数 (T) と Generator (G) が可換であることを強制し，空間摂動関数を学習するための 2 つの敵対的訓練を導入している．これにより MSPC は安定した学習で高い性能を達成した．しかし，MSPC は非ペア画像変換の手法が陥りがちな，背景などの変換に関係のない部分が変化されてしまう問題に対処できていない．これにより，ネットワークは変換するべき前景に変換リソースを集中できず，性能の割にネットワークが大きくなってしまう．

ネットワーク構造

本手法のネットワーク構造は，Generator に 9 層の ResNet-Generator を，Discriminator に PatchGAN-Discriminator を，空間摂動関数 T に ResNet-19 を使用している．空間変換のグリッドサイズは 2×2 とし，最適化には Adam（学習率 2×10^{-4}, $\beta = [0.5, 0.999]$）を用いている．

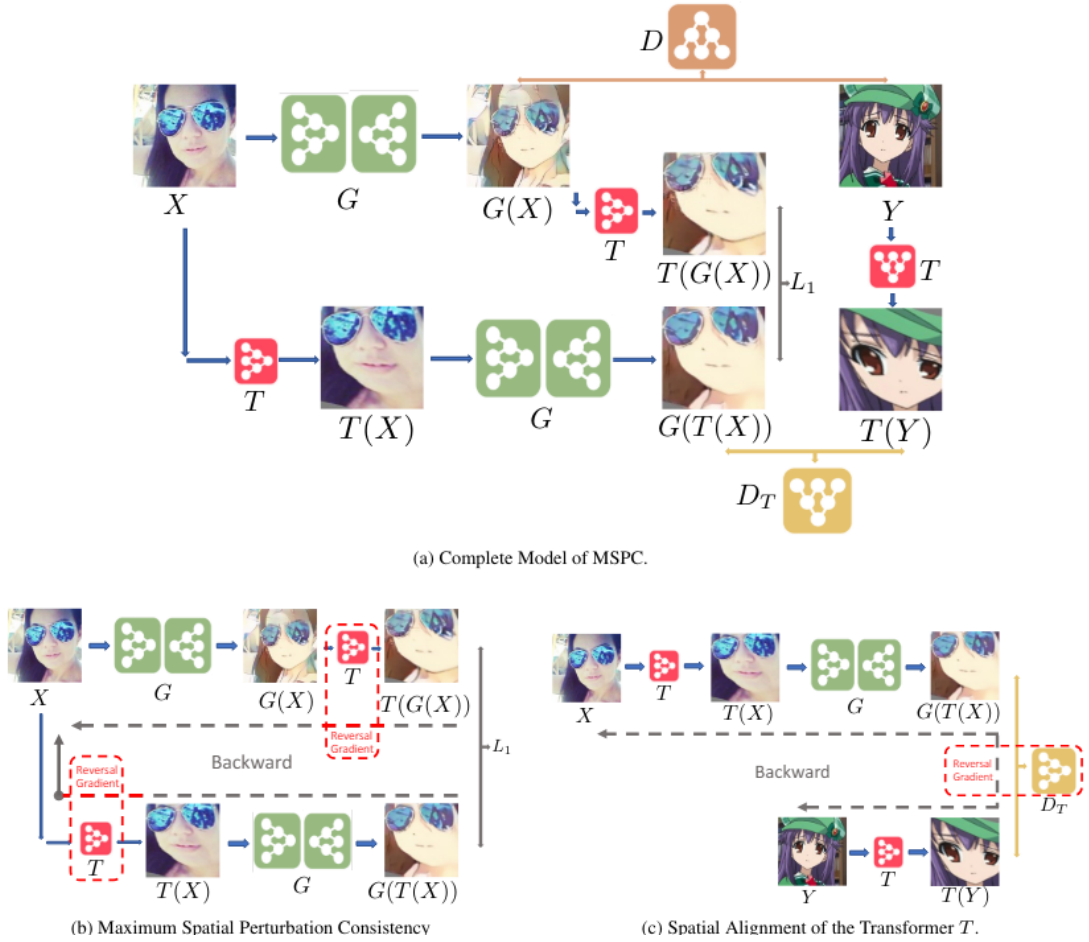


図 3.4　MSPC のパイプライン．図は [2] より引用された．

損失関数

本手法では，通常の敵対的損失，Maximum Spatial Perturbation Consistency Loss を使用している．敵対的損失は，通常の画像だけではなく摂動関数によって摂動が加えられた画像に対しても計算する．Maximum Spatial Perturbation Consistency Loss は Generator を G，空間摂動関数を T とした場合に $||T(G(x)) - G(T(x))||_1$ で定義される．加えて，T に対する制約として $\frac{1}{a} < \frac{|p_i p_j|}{|q_i q_j|} < a, \quad i \neq j \quad \text{and} \quad -b < \sum_{i=1}^{n} p_i < b$ を設けており，ここで $a = 1/3, b = 3, c = 0.25, d = 0.25$ としている．この制約は，摂動関数が大きすぎる変換を行わないようにさせる効果がある．これらの損失関数と制約を組み合わせることで，MSPC は空間摂動の最大化と分布のアライメントを同時に達成し，より安定した画像変換を実現している．

第 4 章

フォトレタッチ

4.1 提案手法

提案手法である Laplacian Pyramid Translation Transformer (LPTT) ネットワークの全体的なアーキテクチャは, 図 4.1 に示される．LPTT は, LPTN をベースとしたフォトレタッチ用の軽量ネットワークである．LPTT において，入力画像 $I_0 \in \mathbb{R}^{H\times W\times 3}$ は，ラプラシアンピラミッドによって周波数成分 $[h_0, h_1, \cdots, h_{L-1}]$ と最も低解像度の画像 I_L に分解される．ここで，L はラプラシアンピラミッドのレベルを表す．周波数成分 h_i や画像 I_i の形状は，I_0 の形状が $H \times W$ の場合，$\frac{H}{2^i} \times \frac{W}{2^i}$ である．h_i や I_i のインデックスの値が高いほど，その成分には低周波情報が多く含まれる．その後，各周波数成分および低解像度画像が，出力画像の周波数成分 $[\hat{h}_0, \hat{h}_1, \cdots, \hat{h}_{L-1}]$ および低解像度画像 $\hat{I}_L$ に変換される．最後に，出力画像 $\hat{I}_0$ はこれらの変換された成分から構成される．Axial Transformer Block は，I_L と h_{L-1} の変換に使用され，軽量 CNN ブロックは $h_0, h_1, \cdots, h_{L-2}$ の変換に使用される．低周波成分を変換するために Axial Transformer Block を使用することには，以下の利点がある．

- Axial Transformer Block は長距離の依存関係を捉えられ，生成される画像の品質が向上する．
- Axial Transformer を含む Transformer ベースのネットワークは，大きな計算コストとメモリサイズの問題がある．しかし，LPTT では低解像度（低周波数）成分のみを変換することでこれを回避できる．Axial Transformer Block を使用して，低解像度の特徴マップ内の周波数成分を変換することで，ネットワークが長距離の依存関係を捉える能力を維持しながら，計算コストとメモリサイズを削減することができる．

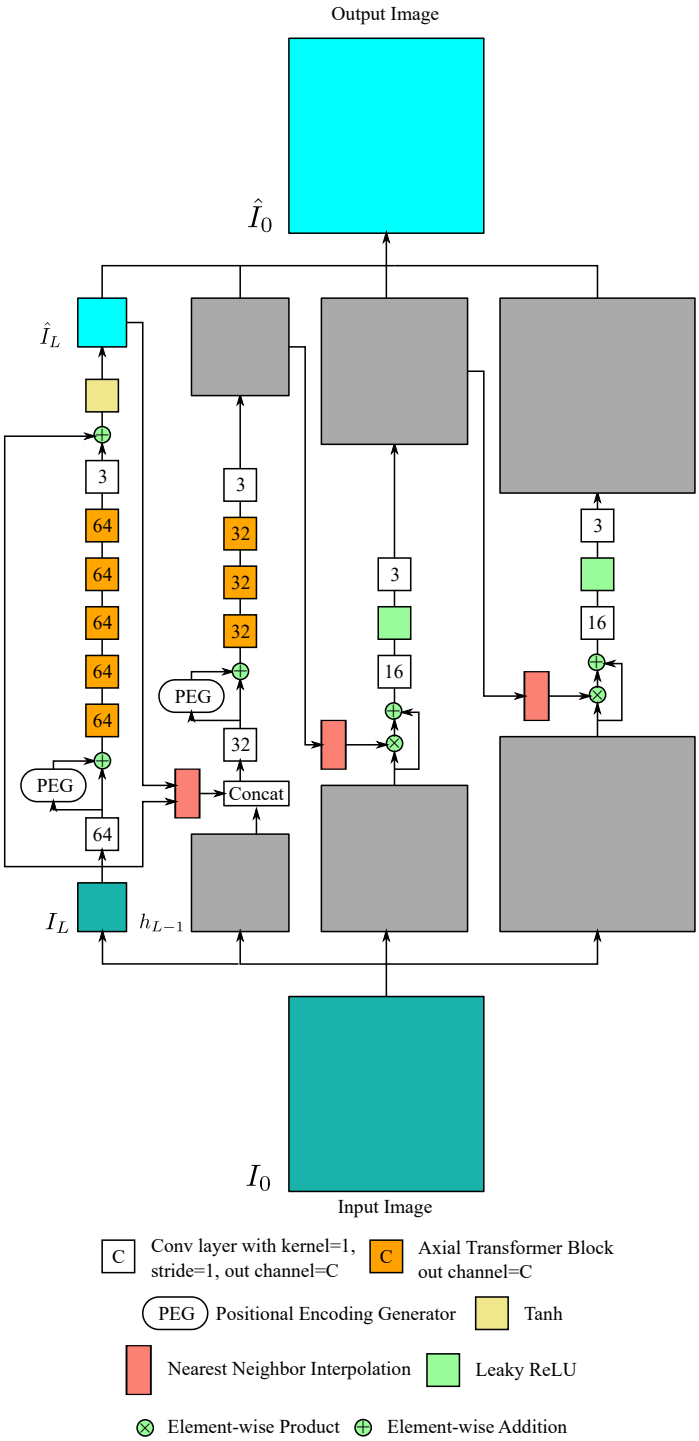


図 4.1　提案手法である LPTT($L = 3$) のネットワーク構造.

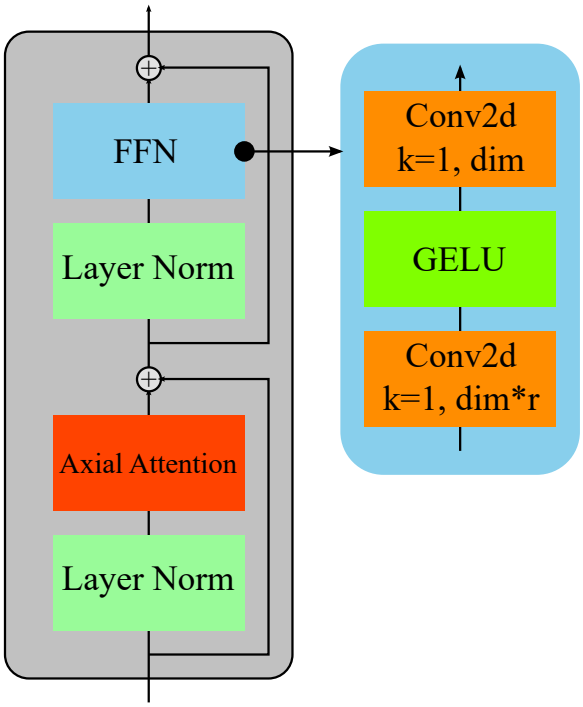


図 4.2　LPTT で使われる Axial Transformer Block の構造．FFN の内容に関して，k は Conv2d のカーネルサイズ，$\mathrm{dim} * r$ は出力チャンネル数を表す．

4.1.1　Axial Transformer Block

Axial Transformer Block は図 4.2 に示すように，Axial Attention と FFN (Feed Forward Network) から構成されている．Axial Attention では，各軸の Attention Map を直列ではなく並列に行い，その結果を合計する．FFN 部分は，カーネルサイズ 1 の Conv2d の層と，その間の GELU 関数からなる．各部分が計算される前に，Layer Normalization が実行され，残差接続が適用される．Conditional Positional Encoding (CPE) が Axial Transformer Block の前に適用され，これは Axial Transformer Block への入力のための位置情報をエンコードする．

4.1.2　低解像度における変換

フォトレタッチでは，主に画像の全体的な色や鮮やかさといった低周波成分が変換される．そのため，画像がラプラシアンピラミッドによって高周波成分と低周波成分に分解され，低周波成分が集中的に変換されたとしても，画像の全体的な品質は保たれる．

また，画像がラプラシアンピラミッドに分解されると，低周波成分は低解像度になるため，画像変換の計算コストは依然として低く抑えられる．

低解像度画像 $I_L \in \mathbb{R}^{\frac{H}{2^L} \times \frac{W}{2^L} \times c}$ は，図 4.1 の左側の経路で変換される．最初に，カーネルサイズ 1 の Conv2d 層を使用し，出力チャネル数 64 でチャンネルを拡張し，CPE を適用する．その後，図 4.2 に示す Axial Transformer Block を 5 つ適用し，カーネルサイズ 1 の Conv2d と 3 つの出力チャネルでチャンネルを調整する．最後に，結果と I_L の和に対して tanh を適用して，低解像度画像 $\hat{I}_L$ を得る．

次に，低解像度の周波数成分 $h_{L-1} \in \mathbb{R}^{\frac{H}{2^{L-1}} \times \frac{W}{2^{L-1}} \times c}$ は，図 4.1 の左から 1，2 番目の経路のように変換される．h_{L-1} はネットワークに直接入力されるのではなく，$\mathrm{concat}[h_{L-1}, up(\hat{I}_L), up(I_L)]$ が入力される．このとき $up(\cdot)$ は最近傍補間である．この入力を処理するネットワークは，I_L を変換するネットワークとほぼ同じ構造を持つ．主な違いは，Conv2d と Axial Transformer Block の出力チャネル数（64 から 32）と Axial Transformer Block の数（5 から 3）である．

4.1.3 高解像度における変換

他の周波数成分 $h_0, h_1, \cdots, h_{L-2}$ は，図 4.1 の左側から 3 番目以降の経路で変換される．レベル i での入力は $up(\hat{h}_{i-1}) \otimes h_i + h_i$ である．これをカーネルサイズ 1 の Conv2d 層 2 つで処理し，それぞれの出力チャネル数は 16 と 3 である．ネットワークは，低解像度の特徴を変換入力に結合することで，高周波成分を適応的に変換できる．これにより，異なる周波数成分間のギャップによって生成された画像に生じるアーティファクトを排除できる．

4.1.4 損失関数

LPTT の実験では，LPTN と同じ損失関数が使用される．まず，入力画像 I_0 と出力画像 $\hat{I}_0$ の間で再構成損失 $\mathcal{L}_{\mathrm{recons}} = |\hat{I}_0 - I_0|_2^2$ が計算される．次に，敵対的損失 $\mathcal{L}_{\mathrm{adv}}$ が式 (4.1) に従って計算される．ここで，I_0 は入力画像，$\tilde{I}_0$ はターゲット画像である．敵対的訓練には 6 層の標準的な Discriminator が使用された．

$$\mathcal{L}_{\mathrm{adv}} = \begin{cases} -\mathbb{E}_{I_0 \sim p_{data}(I_0)}[D(G(I_0))] & \text{(G)} \\ -\mathbb{E}_{\tilde{I}_0 \sim p_{data}(\tilde{I}_0)}[D(\tilde{I}_0))] & \text{(D Real)} \\ \mathbb{E}_{I_0 \sim p_{data}(I_0)}[D(G(I_0))] & \text{(D Fake)} \end{cases} \tag{4.1}$$

さらに，Discriminator の訓練のために WGAN-GP [74] の勾配ペナルティ (gp) が計算される．この項は訓練プロセスを安定させるために採用されている．これらの項から基

準が式 (4.2,4.3) として計算される．$\lambda_1, \lambda_2, \lambda_3$ は，それぞれの項の比率を調整するためのパラメータである．

$$\mathcal{L}_G = \lambda_1 \mathcal{L}_{\text{recons}} + \lambda_2 \mathcal{L}_{\text{adv}} \tag{4.2}$$

$$\mathcal{L}_D = \lambda_2 \mathcal{L}_{\text{adv}} + \lambda_3 \text{gp} \tag{4.3}$$

4.2 提案手法の特長

提案手法は，低周波成分において広い受容野を持つ Axial Transformer を採用することで，DPE [75] のような反復的な計算を行わず，LPTN よりも高い変換性能を持つ，性能と効率を両立したフォトレタッチを実現した．また，提案手法は LPTN が抱える変換画像に特有のアーティファクトが生じるといった問題を解消している．

4.3 実験

4.3.1 セットアップ

データセット

本研究では MIT-Adobe FiveK データセット [48] を用いて，ネットワークの性能を視覚的・定量的に評価した．MIT-Adobe FiveK データセットはフォトレタッチのベンチマークとして一般的に使用されており，過去の研究でも採用されている [1, 21]. MIT-Adobe FiveK データセットには，5000 枚の写真と，それに対応するエキスパートによるレタッチ版が含まれる．実験の正解画像は，先行研究 [1] に従ってエキスパート C がレタッチした画像にした．また，データセットの 5000 枚のうち，4500 枚を 256×256 にリサイズして学習データとし，残りの 500 枚を検証データとする．学習時には LPTN と同じ非ペアデータセットを使用し，対になったデータセットは定量的性能の評価にのみ使用する．MIT-Adobe FiveK データセットでは，入力画像の形式が RAW 形式であり，高解像度の画像を取得するためには Adobe Lightroom という有償ソフトを使用する必要があるため，高解像度画像での比較は，Pixabay からダウンロードした画像を使用する．

学習条件・ハイパーパラメータ

提案手法は LPTN の実験では同じハイパーパラメータを使用した．バッチサイズは 32 であり，訓練画像の解像度は 256×256 に設定されている．学習には学習

表 4.1　MIT-Adobe FiveK データセットの 480p 画像に対する LPTT と他の先行手法の PSNR/SSIM 値．太字は LPTN($L = 3$) の PSNR/SSIM を超える値である．

Methods	PSNR	SSIM
CycleGAN [42]	20.98	0.831
UNIT [45]	19.63	0.811
MUNIT [47]	20.32	0.829
White-Box [21]	21.32	**0.864**
DPE [75]	21.99	**0.875**
LPTN(L=3) [1]	22.91	0.848
LPTN(L=4) [1]	22.19	0.840
LPTN(L=5) [1]	20.00	0.828
LPTN(L=6) [1]	18.94	0.811
LPTT(L=3)	**23.32**	**0.866**
LPTT(L=4)	**23.18**	**0.858**
LPTT(L=5)	**23.04**	**0.860**
LPTT(L=6)	22.70	**0.859**

率 1.0×10^{-4} で $\beta_1 = 0.9, \beta_2 = 0.99$ の Adam を用いる．調整パラメータ λ_i は $\lambda_1 = 1000, \lambda_2 = 1, \lambda_3 = 100$ とした．CPE のカーネルサイズは $L = 3, 4, 5$ で 9, $L = 6$ で 3 である．

4.3.2　評価手法

提案手法では生成画像の品質に関して，既存手法との比較のために PSNR や SSIM を用いている．また，提案手法の軽量さを測定するために異なる GPU において異なる解像度の画像に対する推論速度の比較を行っている．さらに，提案手法の有効性を定性的にも確認をしている．

4.3.3　定量的比較

本節では，PSNR/SSIM 値を LPTT と他の画像間変換手法との間で比較し，推論速度を LPTT と LPTN との間で比較する．

PSNR/SSIM: PSNR/SSIM 値を測定するために使用したデータセットは，LPTN の GitHub リポジトリ（https://github.com/csjliang/LPTN）で提供されている 480p の MIT-Adobe FiveK データセットの検証画像である．MIT-Adobe FiveK データセットの 1080p およびオリジナルスケールの画像を用意できないため，高解像度画像の評価は図 4.5，図 4.6 の視覚的な比較に限られる．

表 4.1 に示されているように，LPTT($L = 3$) の性能は，LPTN($L = 3$) と比較して PSNR で +0.41 dB 向上している．LPTT の PSNR/SSIM は，L が増加するにつれて LPTN と同様に減少する傾向がある．しかし，LPTT のこれらの値の減少は LPTN よりも緩やかであり，LPTT($L = 5$) の性能は依然として LPTN($L = 3$) を上回る．LPTN($L = 6$) は，入力画像のダウンサンプリングが多すぎるため，性能が大幅に低下しているが，LPTT($L = 6$) は同じスケールでありながら競争力のある性能を維持している．LPTT($L = 6$) の PSNR は，LPTN($L = 6$) と比較して 3.75 dB と大きく向上している．

訓練画像の解像度が 256×256 であるため，LPTT($L = 6$) の場合，I2IT 変換は主に非常に低解像度の特徴マップ (4×4 および 8×8) にのみ適用され，他の成分は非常に軽量な 2 層の畳み込みでのみ変換される．驚くべきことに，LPTT($L = 6$) は，解像度が 32×32 および 16×16 の成分を計算する LPTN($L = 3$) と同様の性能を持っている．この結果は，Transformer ベースのアーキテクチャが低解像度で処理する際に CNN を上回る特徴抽出能力を有する可能性があることを示唆している．

また，White-Box や DPE は SSIM において優れた性能を達成しているが，提案手法や LPTN と比較して極めて大きい計算コストを有しており，4K や 8K の解像度を処理できず，性能と軽量さのトレードオフを克服しているとは言えない．

推論速度: 表 4.2 と表 4.3 は，LPTT と LPTN の推論速度を示している．表 4.2 と表 4.3 の太字の値は，LPTN よりも PSNR および推論速度が優れている値である．例えば，2K，4K，8K の画像において，LPTT($L = 5$) は LPTN($L = 3$) よりも速く，かつ PSNR が優れているため，その値は太字で示されている．4K および 8K の画像では，LPTT($L = 6$) が LPTN($L = 4$) よりも速く，かつ PSNR が優れているため，その値は太字で示されている．

LPTT の推論速度は，LPTN と同様に L が増加するにつれて増加する傾向がある．Transformer ベースのアーキテクチャはメモリサイズが大きいため，推論時間は CNN ベースのアーキテクチャよりも長くなることが多い．しかし，表 4.2 に示されているように，LPTT($L = 5$) は，高解像度において，同等の性能を持つ LPTN($L = 3$) と比較して，より速い推論速度を持つ．これは，図 4.3 に示されているように，LPTN と

表 4.2　異なる解像度の画像に対する LPTN と LPTT の推論速度（images/sec）の比較．表 4.1 中の PSNR が 22dB 以上のモデルの推論速度のみを示す．推論速度は 50 枚の平均値である．推論には NVIDIA GeForce RTX 3090 24GB RAM を使用．太字の値は，LPTN から PSNR と推論速度の両方で大きい値である．OOM は Out Of Memory の略.

	L	480p	1080p	2K	4K	8K
LPTN	3	0.004	0.011	0.019	0.041	0.162
	4	0.005	0.007	0.011	0.024	0.096
LPTT	3	0.018	0.053	0.105	0.313	OOM
	4	0.019	0.016	0.027	0.066	0.368
	5	0.013	0.013	**0.014**	**0.028**	**0.121**
	6	0.011	0.012	0.012	**0.022**	**0.083**

LPTT の間の計算コストの差が高解像度画像では大きいためである．この差の影響は，Transformer の使用によるメモリサイズの増加の影響を上回る．

さらに，LPTT では，ラプラシアンピラミッドを使用して入力画像をダウンサンプリングし，Transformer で処理する低解像度画像を作成するため，計算コストとメモリサイズの増加を防げる．例えば，LPTT($L = 5$) で 4K（3840×2160）の入力画像を処理する場合，Axial Transformer Block で処理される特徴マップのサイズは 240×135 および 120×67 である．

表 4.1 と表 4.2 に示されているように，LPTT($L = 6$) は，NVIDIA GeForce RTX 3090 24GB を使用した場合，PSNR が 22.70 dB であり，8K 解像度画像における推論速度は 0.083 s である．したがって，提案手法は NVIDIA GeForce RTX 3090 24GB を使用することで，8K 画像において競争力のある性能を維持しながらリアルタイムに近い推論速度を実現している．

表 4.3　異なる解像度の画像に対する LPTN と LPTT の推論速度（images/sec）の比較．表 4.1 中の PSNR が 22dB 以上のモデルの推論速度のみを示す．推論速度は 50 枚の平均値である．推論には NVIDIA GeForce GTX 1080 Ti 12GB RAM を使用．太字の値は，LPTN から PSNR と推論速度の両方で大きい値である．OOM は Out Of Memory の略．

	L	480p	1080p	2K	4K	8K
LPTN	3	0.006	0.034	0.059	0.135	0.572
	4	0.005	0.025	0.044	0.100	0.428
LPTT	3	0.020	0.167	0.361	OOM	OOM
	4	0.013	0.046	0.087	0.233	OOM
	5	0.012	**0.028**	**0.049**	**0.111**	**0.522**
	6	0.013	0.025	**0.042**	**0.094**	**0.404**

4.3.4　定性的比較

低解像度画像における比較

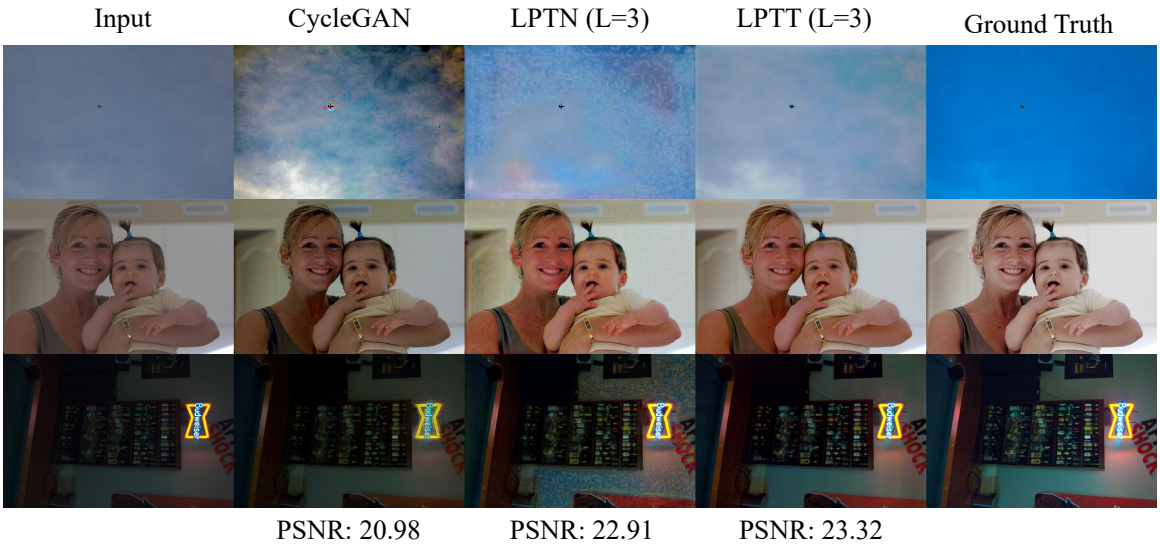


図 4.4　MIT-Adobe FiveK データセットにおける検証画像の生成．2 列目は CycleGAN の生成画像．3 列目は LPTN($L = 3$) の生成画像．4 列目は LPTT($L = 3$) の生成画像．

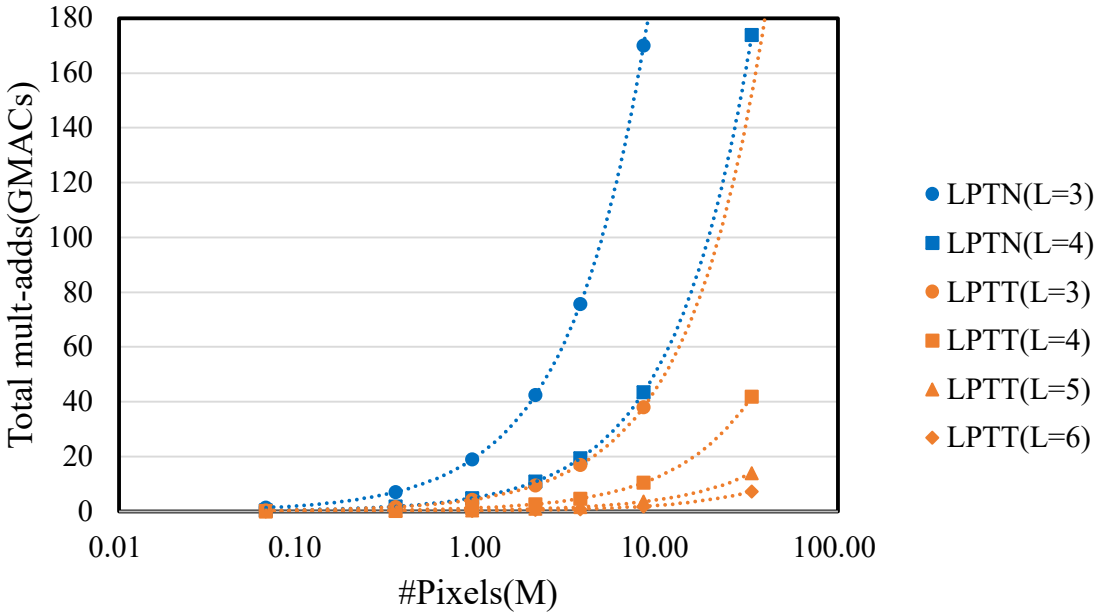


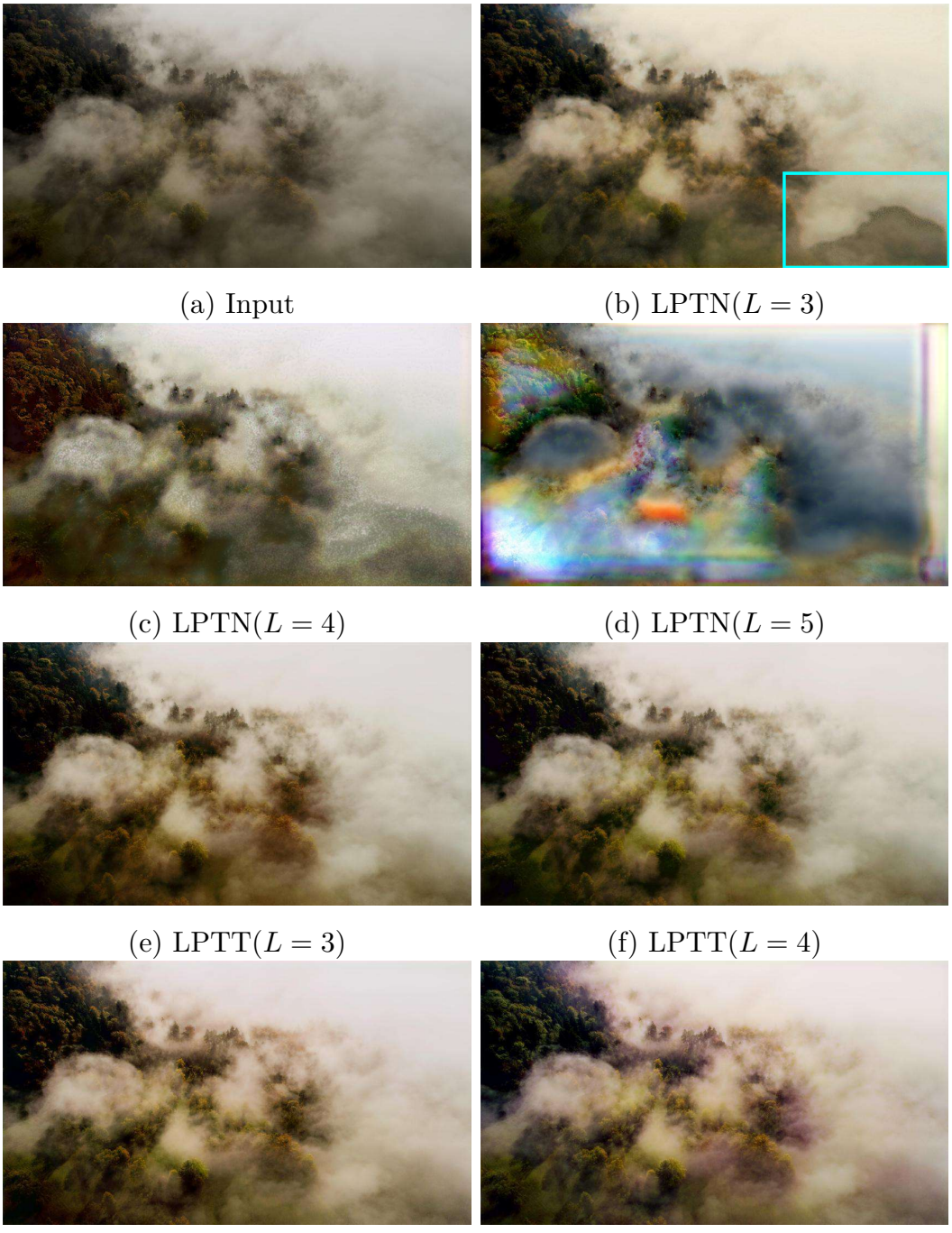
図 4.3　PSNR>22dB の場合の LPTT と LPTN の計算量の比較．横軸は画像の画素数，縦軸は torchinfo で測定した計算量である．

ここでは，CycleGAN，LPTT($L = 3$)，LPTN($L = 3$) による生成画像を，MIT-Adobe FiveK データセット上で比較する．生成された画像を図 4.4 に示す．図 4.4 に示すように，CycleGAN によって生成された画像は，色合いが暗く，若干の乱れがあることがわかる．また，LPTN($L = 3$) では，入力画像中の一様に広がるテクスチャ（空や壁など）の特徴を捉えられず，生成画像に不自然なアーティファクトが発生していることが分かる．これは LPTN が CNN ブロックを用いているためであり，長距離依存性を捉えられない．LPTT は長距離依存性を捉えられる Axial Transformer Block を使用するため，LPTT はアーティファクトのない画像を生成できる．

高解像度画像における比較

LPTT と LPTN は非常に軽量なネットワークであるため，高解像度の画像を処理できる．しかし，表 4.1 中の PSNR/SSIM の値は 480p 画像に対する値であり，高解像度画像に対する性能を示していない．そこで，Pixabay から取得した高解像度画像（5000×2809，1920×1080）で LPTT と LPTN の性能を評価した．その結果を図 4.5 と図 4.6 に示す．図 4.5 に示すように，LPTN($L = 3, 4$) では 480p 画像と同様に，霧の領域でアーティファクトのある画像が生成され，LPTN($L = 5$) では歪んだ画像が

生成される．一方，各 L における LPTT は，アーティファクトや歪みのない画像を生成する．図 4.6 に示すように，LPTN($L = 3, 4, 5$) で生成された画像は不自然な色合いになっている．一方，各 L における LPTT は不自然な色調のない画像を生成する．このことから，高解像度の画像では，LPTN では 480p の画像に見られるようなアーティファクトが発生する問題が残っているが，LPTT ではアーティファクトのない画像を生成できることがわかる．従って，高解像度画像を処理する場合でも，Transformer を用いて低解像度成分を計算することで，長距離依存性を捉え，推論速度を高速に保ったまま性能を向上させることが可能であると言える．



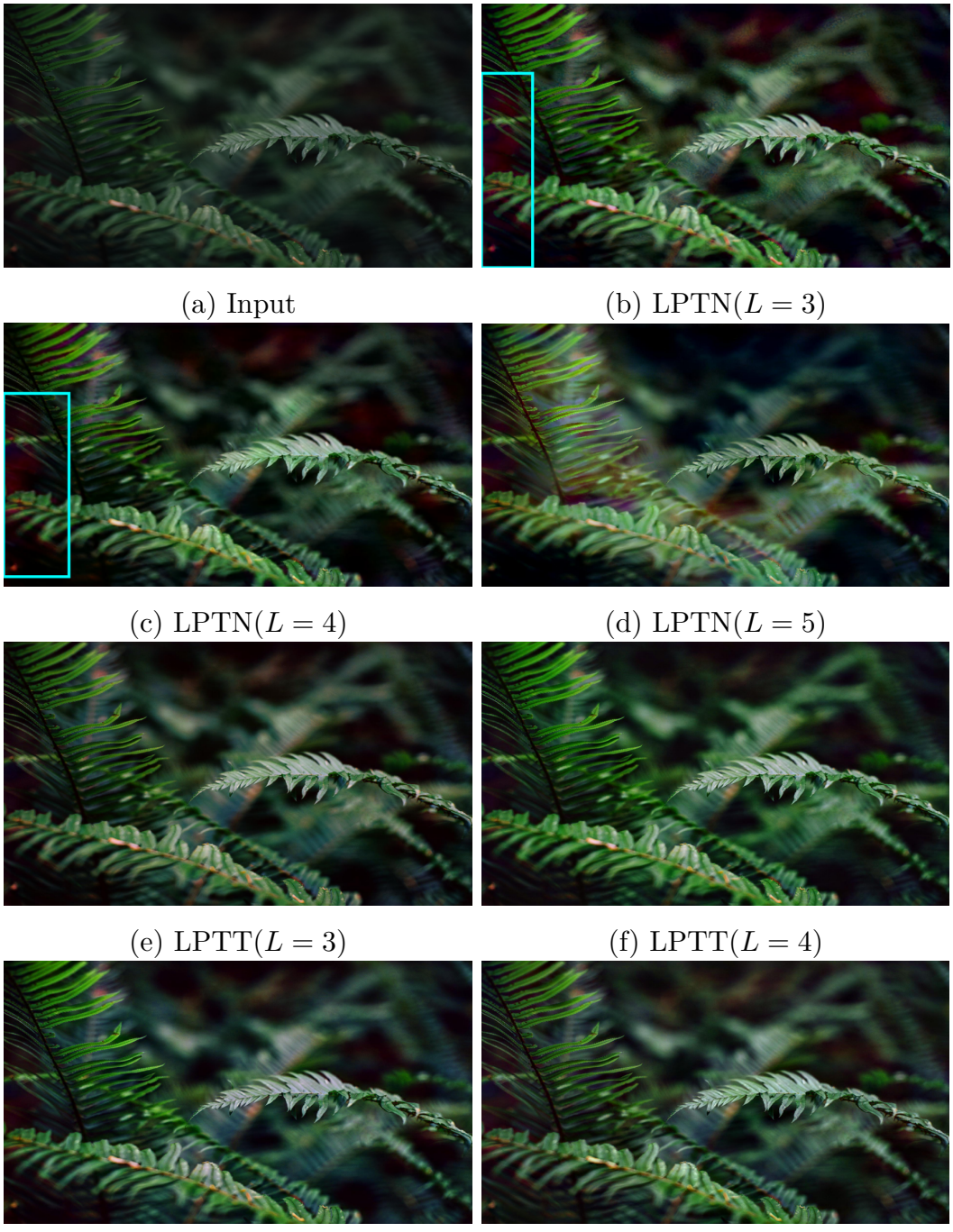
(a) Input　(b) LPTN($L = 3$)

(c) LPTN($L = 4$)　(d) LPTN($L = 5$)

(e) LPTT($L = 3$)　(f) LPTT($L = 4$)

(g) LPTT($L = 5$)　(h) LPTT($L = 6$)

図 4.5　Pixabay から取得した解像度 5000 × 2809 の画像に対する LPTT と LPTN の出力．ファイルサイズの関係で，添付画像は 854 × 480 にリサイズされている．見つけにくいと思われるアーティファクトは矩形で囲んでいる．LPTN で生成した画像は霧の部分に不自然なアーティファクトがあり，LPTN($L = 5$) で生成した画像は歪んでいる．対して，LPTT による生成画像には，このようなアーティファクトや歪みはない．



(a) Input　(b) LPTN($L = 3$)

(c) LPTN($L = 4$)　(d) LPTN($L = 5$)

(e) LPTT($L = 3$)　(f) LPTT($L = 4$)

(g) LPTT($L = 5$)　(h) LPTT($L = 6$)

図 4.6　Pixabay から取得した 1920 × 1080 の解像度の画像に対する LPTT と LPTN の出力．ファイルサイズの関係で，添付画像は 854 × 480 にリサイズされている．見つけにくいアーティファクトは矩形で囲んでいる．LPTN で生成した画像は色合いが不自然である ($L = 3, 4, 5$)．対して，LPTT($L = 3, 4, 5, 6$) ではこのような歪みはない．

表 4.4　CPE の有無による LPTT($L = 5$) の PSNR/SSIM の値.

	LPTT($L = 5$)	
	PSNR	SSIM
w CPE	23.04	0.860
w/o CPE	22.70	0.856

4.3.5　新規性の有効性の検証

Axial Transformer Block の効果

LPTT の LPTN に対する主な改良点は，主にフォトレタッチで変換される低周波成分を処理する CNN を Axial Transformer Block に変更したことである．この改善により，ネットワークは長距離依存性を捉えられ，低周波成分の表現力が向上する．したがって，提案手法は定量的・視覚的評価の向上が期待できる．実際に，表 4.1 に示すように，提案手法は LPTN に比べて PSNR と SSIM の値が向上している．フォトレタッチは主に低周波成分を変換するタスクであるため，低周波成分の表現力が優れていることが評価指標の優れた値につながったと考えられる．したがって，LPTT による評価指標の改善は，Axial Transformer Block が CNN よりも優れた表現力を持つことを示している．さらに，図 4.4 を見ると，従来の LPTN によって生成された画像にはアーティファクトがあることがわかる．これらのアーティファクトは，空や壁のような一様に広がったテクスチャに現れる．これは，ネットワークが一様に分布したテクスチャをキャプチャできないためにアーティファクトが発生していることを示唆している．一方，LPTT によって生成された画像にはアーティファクトがない．このことは，LPTT が LPTN よりも，ネットワークの一様に広がったテクスチャを捉える能力に優れていることを示唆している．

CPE の有無

CPE は，標準的な位置エンコーディングのように，Transformer が画素間の位置関係を認識することを可能にする能力を持っている．したがって，CPE を用いた LPTT は，CPE を用いない LPTT よりも優れた性能を持つことが期待される．実際，表 4.4 に示すように，LPTT($L = 5$) において，CPE を用いたネットワークでは，CPE を用いないネットワークと比較して，PSNR が +0.34 dB，SSIM が +0.003 増加している．

表 4.5　CPE の位置に対する LPTT($L = 3, 5$) の PSNR/SSIM の値.

	LPTT($L = 3$)		LPTT($L = 5$)	
pos	PSNR	SSIM	PSNR	SSIM
-1	23.32	0.866	23.04	0.860
0	23.15	0.863	23.03	0.860

CPE の位置

ここで，pos $= n$ は，PEG(Positional Encoding Generator) を挿入して $(n + 2)$ 番目の Axial Transformer Block の前に CPE を適用することを意味する．表 4.5 に示すように，LPTT($L = 3, 5$) では pos $= -1$ のとき，最高の性能が得られる．pos $= -1$ の場合，次元が拡張された入力画像に対して CPE が適用され，特徴マップがもつれないため，対象画素に対して周辺画素から適切な符号化値を決定できると考えられる．この結果は，元の CPE の論文 [67] で示されたものとは異なる．CPE 原論文の DeiT-tiny with CPE [67] は pos $= 0$ で良い性能を示すが，LPTT は pos $= -1$ で良い性能を示す．CPE による DeiT-tiny は，まず入力画像をパッチに分割し，入力トークンを作成する．次に，入力トークンを画像テンソルのように再形成し，PEG により符号化値を計算する．この方式は LPTT の CPE 方式より複雑である．CPE を用いた DeiT-tiny では，特徴マップの情報は既に -1 でもつれている．そのため，CPE with DeiT-tiny では，特徴マップが Transformer を通ることで実際の受容野を拡大するため，pos $= 0$ のときに性能が向上する．LPTT の場合，分離された特徴マップの利点は，Transformer を通ることで得られる受容野の拡大の影響を上回る．

CPE のカーネルサイズ

CPE は Conv2d を用いて実装されているため，カーネルサイズはハイパーパラメータとなる．表 4.6 に CPE のカーネルサイズ k に対する PSNR/SSIM 値を示す．表 4.6 から分かるように，LPTT($L = 3, 4$) では $k = 9$，LPTT($L = 5$) では $k = 5$ が最も良い性能を示す．これらの結果は 2 つの傾向の均衡点であると考えられる．第一に，カーネルサイズが大きいほど，より広い範囲から計算することで最適な符号化値が決定されるということである．第二に，広い範囲から情報を取りすぎると，最適値を決定することが難しくなる．なお，LPTT ではラプラシアンピラミッドを用いて入力画像のスケールを小さくしているため，k が大きくなると PEG の実質的な受容野は非常に大きくなる．

表 4.6　CPE のカーネルサイズに対する LPTT($L = 3, 4, 5$) の PSNR/SSIM の値．太字がその列で最も高い値を表す．

	LPTT($L = 3$)		LPTT($L = 4$)		LPTT($L = 5$)	
k	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
3	23.30	0.863	23.06	0.860	22.93	0.859
5	23.30	0.865	23.13	0.862	**23.04**	**0.860**
9	**23.32**	**0.866**	**23.18**	0.858	N.A.	N.A.
15	23.25	0.864	23.00	**0.859**	N.A.	N.A.
21	23.27	0.865	N.A.	N.A.	N.A.	N.A.

表 4.7　LPTT と LPTN の各 L におけるネットワークパラメータ数の比較．

L	LPTN	LPTT
3	617567	390431
4	617682	390546
5	617797	385285
6	617912	383864

計算量・パラメータ数の比較

図 4.3 は，LPTT と LPTN の計算量を，入力画像の画素数 L ごとに比較している．LPTT，LPTN ともに L が大きくなるにつれてネットワーク性能は低下する．しかし，PSNR が 22dB 以上のモデルでは，LPTT の方が LPTN よりも計算量が少なく，特に高解像度画像ではその差は大きい．各 L に対する LPTT と LPTN のパラメータ数を表 4.7 に示す．表 4.1 や表 4.7 から，LPTT のパラメータ数は LPTN のパラメータ数の約 60 %であるが，LPTT の性能は LPTN より高いことがわかる．したがって，LPTT は LPTN よりもパラメータ効率が良いと言える．

第 5 章

動画のノイズ除去

5.1 提案手法

PTFN は図 5.1 に示されるような構造をしている MIMO ネットワークである．いくつかの従来の手法 [6,18] に従い，PTFN は Pseudo Temporal Fusion Denoising Block (PTF Denoising Block)（図 5.2）という U-Net [5] 構造のブロックを 2 つ直列につなげたネットワークとなっている．PTFN は，時刻 $t-T$ から $t+T$ までのノイズ動画シーケンス $[f_{\mathrm{noise}(t-T)}, \cdots, f_{\mathrm{noise}(t+T)}]$ を入力し，時刻 $t-T$ から $t+T$ までのノイズ除去動画シーケンス $[\hat{f}_{(t-T)}, \cdots, \hat{f}_{(t+T)}]$ を出力する．さらに，PTFN は PTF Denoising Block を 1 つ適用した場合の中間的な結果 $[\hat{f}_{\mathrm{inter}(t-T)}, \cdots, \hat{f}_{\mathrm{inter}(t+T)}]$ も出力する．中間的な結果を求めるときは，1×1 の Conv2d (図 5.1 の ToRGB) によってチャンネル

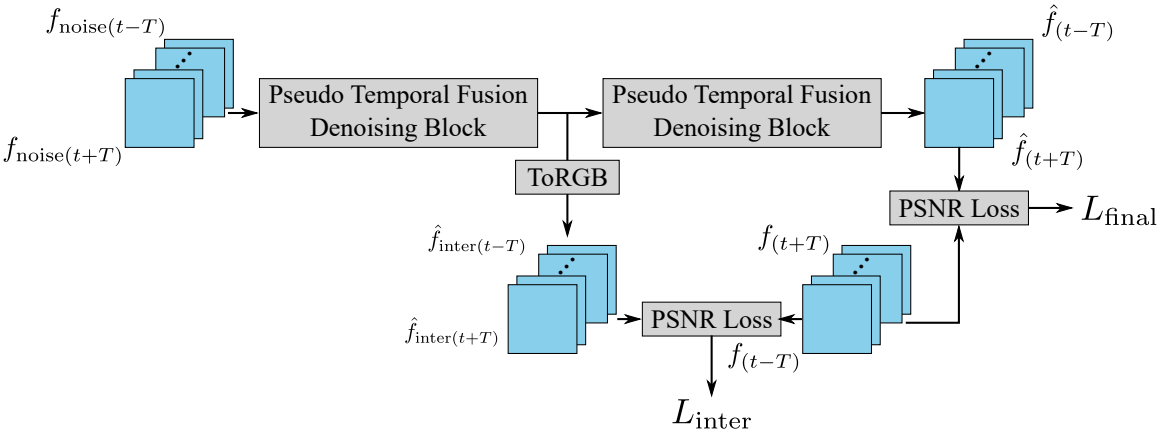


図 5.1　PTFN のネットワーク構造．PTFN では Pseudo Temporal Fusion Denoising Block が 2 つ直列に接続されており，それぞれの出力に対して損失を計算する．

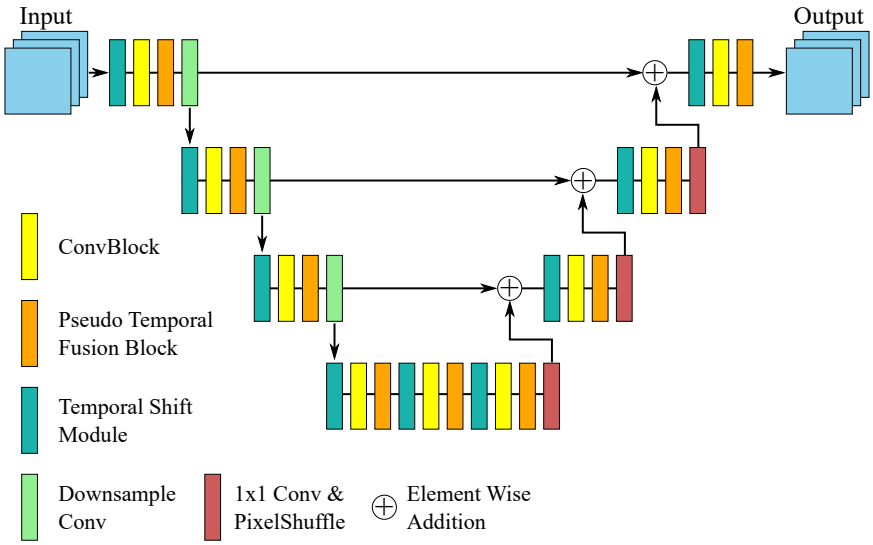


図 5.2　Pseudo Temporal Fusion Denoising Block (PTF Denoising Block) の構造．PTF Denoising Block は，ConvBlock，PSeudo Temporal Fusion Block，Temporal Shift Module で構成される U-Net 型のネットワークであり，ダウンサンプリングはカーネルサイズ 2，ストライド 2 の Conv2d で，アップサンプリングはカーネルサイズ 1 の Conv2d と Pixel Shuffle でおこなっている．

数を調整している．

5.1.1　Pseudo Temporal Fusion Denoising Block (PTF Denoising Block)

PTF Denoising Block は，図 5.2 に示される 4 スケールの U 字型のネットワークである．各スケールにおいて，PTF Denoising Block は，Temporal Shift Module (TSM)，ConvBlock，Pseudo Temporal Fusion Block (PTF Block) からなる層で動画を処理する．ダウンサンプリングはカーネルサイズ 2，ストライド 2 の Conv2d で，アップサンプリングはカーネルサイズ 1 の Conv2d と Pixel Shuffle [71] でおこなっている．また，U-Net と異なり特徴量を融合させる際に，特徴量をチャンネル方向に結合するのではなく要素ごとに加算をしている．さらに提案手法では，[18] に倣い，推論時には TSM を Bidirectional Buffer Block に置き換えている．

5.1.2 Temporal Shift Module (TSM)

TSM [2] は，動画を処理する MIMO ネットワークにおいて，時間的な情報をチャンネル的な情報と融合するためのモジュールである．時間的な情報をチャンネル的な情報と融合することで，非常に計算量が大きい Conv3d の使用を避け，軽量なネットワークを構築できる．TSM は，隣接フレームの特徴マップの一部のチャンネルを前後フレームにシフトさせる動作を行っており，サイズが (B, T, C, H, W) の特徴マップ f を入力した場合，TSM の出力 f' は式 (5.1) のように表される．

$$\begin{aligned} f'[:,t,:,:,:] = \mathrm{Concat}\Big(& f\left[:,t-1,:\frac{c}{2r},:,:\right], \\ & f\left[:,t+1,\frac{c}{2r}:\frac{c}{r},:,:\right], \\ & f\left[:,t,\frac{c}{r}:,:,:\right]\Big) \end{aligned} \tag{5.1}$$

ただし，バッチサイズを B，フレーム数を T，チャンネル数を C，縦幅を H，横幅を W，r をシフトさせる割合を決定する定数とする．また，$\mathrm{Concat}(x_1, x_2, \cdots, x_n)$ は，チャンネル方向に $[x_1, x_2, \cdots, x_n]$ を結合する操作を意味している．

5.1.3 ConvBlock

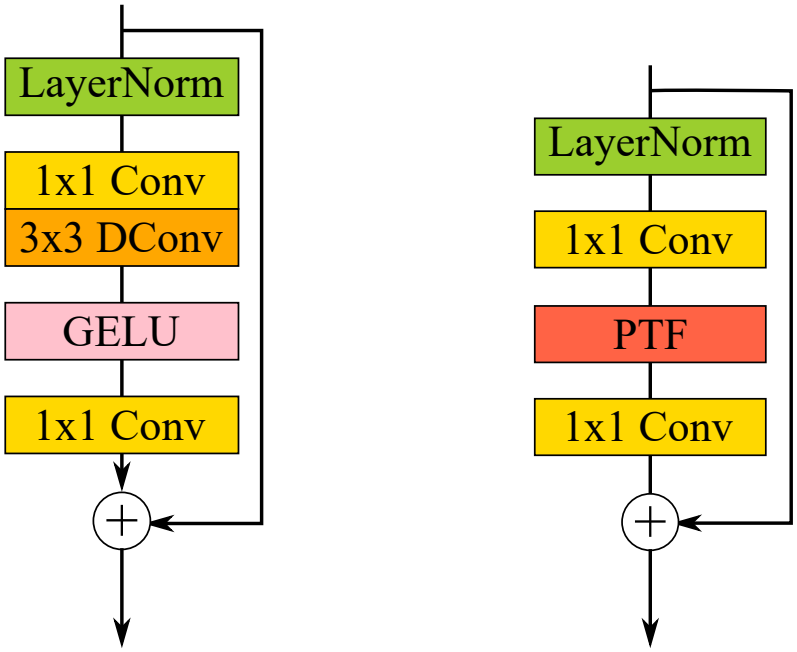
PTFN では，従来の CNN ネットワークに用いられているような ConvBlock ではなく，図 5.3 (a) に表されるような構造の ConvBlock を採用している．ConvBlock は，Layer Normalization [59]，1×1 Conv2d，3×3 Depthwise Conv2d [52]，GELU [56]，1×1 Conv2d から成る構造をしており，入力と出力の間で残差接続をしている．

5.1.4 Pseudo Temporal Fusion Block (PTF Block)

PTF Block は，図 5.3 (b) に示されるような，Layer Normalization，1×1 Conv2d，Pseudo Temporal Fusion (PTF)，1×1 Conv2d から成る構造をしている．PTF は，図 5.4 に示されるようなモジュールであり，TSM と合わせて動画シーケンスの時間的相関を捉えることを目的としている．PTF の具体的な操作は，式 (5.2) と式 (5.3) のように計算される．

$$X_{t-1}, X_t, X_{t+1} = \mathrm{Split}(X, 3) \tag{5.2}$$

$$Y = \mathrm{Like}(0.5, X_t) \otimes ((X_{t-1} \otimes X_t) \oplus (X_t \otimes X_{t+1})) \tag{5.3}$$



(a) ConvBlock　　(b) PTF Block

図 5.3　ConvBlock と Pseudo Temporal Fusion Block (PTF Block) の構造.

ただし，Split(X, n) はチャンネル方向に X を n 等分する操作であり，Like(a, X) は，値がすべて a である X と同様のサイズのテンソルを意味している．PTF では，チャンネル方向に入力を 3 等分しており，これは，ある時刻 t の情報を含む特徴マップと，その前後 1 フレームの情報を持った特徴マップとを分離することを目的としている． 通常は，チャンネル方向に特徴マップを区切る操作によって，時間的な情報を分離することはできないため，モジュール単体では時間的な関係性を捉えられない．しかし，PTF Denoising Block には TSM が含まれており，TSM によって時間的な相関はチャンネル的な相関と融合されている．よって，PTF ではチャンネル方向に特徴マップを区切り，それぞれの時刻の情報を持った特徴マップ間で乗算や加算をすることで，擬似的に時間的な相関を捉えることが可能となっている．また，PTF は Conv3d のように直接的に時間軸で計算を行わないため軽量であることが特徴である．実際に，PTF Denoising Block の計算量と 3×3 の Conv3d の計算量の比は式 (5.4) のように表され，前者の方が計算量が少ないことがわかる．

$$\frac{\underbrace{3hwc^2}_{\text{1x1 Conv2d}} + \underbrace{3hwc}_{\text{PTF}} + \underbrace{hwc^2}_{\text{1x1 Conv2d}}}{\underbrace{9fhwc^2}_{\text{3x3 Conv3d}}} \sim \frac{4}{9f} \tag{5.4}$$

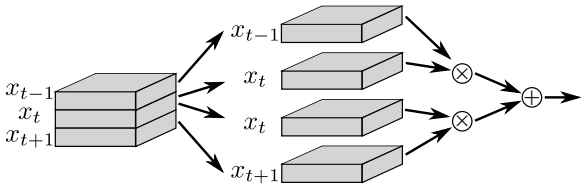


図 5.4　Pseudo Temporal Fusion の構造．$\otimes$ と $\oplus$ は要素ごとの乗算と要素ごとの加算を意味する．

ただし，h, w, c, f はそれぞれ特徴マップの縦幅，横幅，チャンネル数，フレーム数を表している．

5.1.5　損失関数

学習時に用いた損失関数は式 (5.5) のように設定した．

$$\frac{1}{(2T+1)(1+\alpha)}\left(\sum_{\tau=t-T}^{t+T}\mathrm{PSNRLoss}(\hat{f}_\tau, f_\tau)+\right.$$
$$\left.\alpha\sum_{\tau=t-T}^{t+T}\mathrm{PSNRLoss}(\hat{f}_{\mathrm{inter}(\tau)}, f_\tau)\right) \tag{5.5}$$

式 (5.5) における記号の意味は図 5.1 に倣っており，t, T はフレームのインデックス，$\hat{f}$ はネットワークの最終的な出力画像，$\hat{f}_{\mathrm{inter}}$ はネットワークの中間出力画像，f はクリーン画像を表している．また，$\mathrm{PSNRLoss}(x, y)$ は，x と y の間の PSNRLoss を表しており，α は比率を調節するための係数である．

最新手法の多く [6, 18] はノイズ除去ネットワークを 2 つ直列につないだ構造であることから，多段階のノイズ除去が高品質のノイズ除去に有用であると考えた．よって，式 (5.5) において，多段階のノイズ除去学習を補助するために中間出力とクリーン画像との間でも損失を取った．また，今回使用する PSNR Loss は大きな絶対値をとるため，項の追加によって損失関数の値が大きく変化し，勾配の値も大きく変化してしまう．よって，損失値を $1+\alpha$ で割ることで α の値によって起こる損失値の変化を軽減している．

5.2　提案手法の特長

提案手法は，TSM と組み合わせて，チャンネル方向の演算のみで擬似的に時間関係を捉える PTF を提案・採用しているが，これにより Conv3d のような非常に計算コストが高い操作を行わずに高性能の変換をすることを可能とした．また，提案手法はノイズ除去ネットワークに ConvNeXt [49] や NAFNet [50] を参考にした ConvBlock（モダンな ConvBlock）を採用しており，従来手法と比較して同等以上の性能を保ちながら計算量を大幅に削減している．

5.3　実験

5.3.1　セットアップ

データセット

提案手法と従来法を比較するために FastDVDNet [6] に倣って，DAVIS データセット [76] と Set8 テストセット [6, 77] を用いて定量的・定性的な比較検証を行う．DAVIS データセットは 90 の RGB 動画シーケンスを含む訓練セットと 30 の RGB 動画シーケンスを含むテストセットから構成されており, 動画の解像度は 480p (854×480) である．Set8 テストセットは Derf テストセット [77] からの動画シーケンス 4 つ，GOPRO で撮影された動画シーケンス 4 つから構成されており，動画の解像度は 540p (960 × 540) である．

学習条件・ハイパーパラメータ

提案手法の訓練時のバッチサイズは 16，入力フレーム数は 11 で，図 5.1 の表記に従うと $T = 5$ である．入力動画は 96 × 96 にランダムクロップされ，動画シーケンスごとにランダムフリッピングが適用されている．動画シーケンスには Additive White Gaussian Noise (AWGN) が付加され，ノイズレベル σ は一様分布 $U(5, 55)$ から選択される．

学習は DAVIS 訓練セットに対して行い，400,000 イテレーションの訓練を行う．最適化アルゴリズムには $\beta_1 = 0.9, \beta_2 = 0.9$ の Adam [61] を用いている．また，学習率スケジューラとして，$\eta_{max} = 1.0 \times 10^{-3}, \eta_{min} = 1.0 \times 10^{-7}$，$T_{max} = 400,000$ の Cosine Annealing [78] を用いている．学習時の損失関数 (5.5) の中間出力の損失の比

率は $\alpha = 0.1$ としている．学習時には勾配の値を 0.1 でクリッピングしている．

本研究では図 5.2 の PTF Denoising Block を使用したモデル (PTFN) の他に，PTF Denoising Block の各ステージにおける PTF Block の数を 1 つから 2 つに増やしたモデル (PTFN-L) でも比較評価をしている．また，PTF Denoising Block が 1 つのみのモデル (PTFN Half, PTFN-L Half) についても比較評価をしている．これらのモデルに関しては PTFN や PTFN-L を訓練した後に，その重みを用いてファインチューニングしている．ファインチューニングは DAVIS 訓練データセットに対して行い，100,000 イテレーションの訓練を行う．最適化アルゴリズムには $\beta_1 = 0.9, \beta_2 = 0.9$ の Adam を用いている．また, 学習率スケジューラとして, $\eta_{max} = 1.0 \times 10^{-4}, \eta_{min} = 1.0 \times 10^{-7}$, $T_{max} = 100,000$ の Cosine Annealing を用いている．

さらに，ノイズレベルが既知である場合のノイズ除去の他に，ノイズレベルが既知ではない場合の動画のノイズ除去に関しても実験を行った．この場合の訓練条件はノイズレベルが既知の場合と同じである．

また，ネットワークのチャンネル数は，上のスケールから，32, 64, 128, 256 となっており，全ての TSM において $r = 8$ としている．

5.3.2 評価手法

提案手法と既存手法のノイズ除去性能は，動画に付加された AWGN のノイズレベル σ が $10, 20, 30, 40, 50$ の場合における PSNR や SSIM の値を測定することで比較している．また，定量的な比較以外にも，定性的にも比較をしている．提案手法と既存手法の推論速度は，480p, 720p, 1080p の動画に対する実測と，480p, 720p の動画に対する GMAC を計算している．また，本章の研究では，性能と軽量さのトレードオフの観点から比較をするために，計算量とノイズ除去性能（PSNR）の関係を比較した図を作成している．

5.3.3 定量的比較

生成画像の品質: 表 5.1 は，ノイズレベルが既知である場合の提案手法と従来手法における DAVIS テストセットと Set8 テストセットにおける生成画像の PSNR の値を比較している．表 5.1 は提案手法が従来手法と比較して幅広いノイズレベルに対して優れた性能を記録していることを示している．PTFN，PTFN-L は従来の高速な手法の最先端である BSVD に対してノイズレベルが 50 の場合，DAVIS テストセットでそれぞ

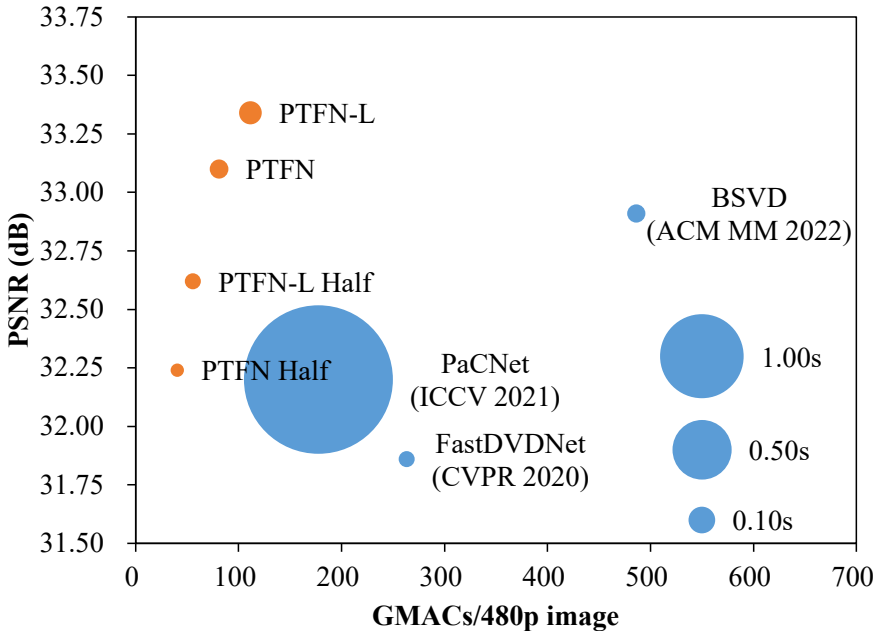


図 5.5　既存手法（青）と提案手法（オレンジ）の比較．横軸は 480p 画像 1 枚あたりの計算コスト（Giga Multiply-Accumulate Operations （GMAC)），縦軸は DAVIS テストセットのノイズレベル 50 に対する PSNR，バブルの大きさは PyTorch での処理時間を表す．提案手法は，既存の軽量な最先端手法である BSVD の約 16.7% の計算コストしか有していないが性能が上回っている．

れ 0.19 dB，0.43 dB，Set8 テストセットで 0.05 dB，0.20 dB の改善をしている．また，PTFN Half，PTFN-L Half は FastDVDNet に対してノイズレベルが 50 の場合，DAVIS テストセットでそれぞれ 0.38 dB，0.75 dB，Set8 テストセットで 0.10 dB，0.38 dB の改善をしている．

表 5.2 は，ノイズレベルが既知ではない場合において，提案手法と従来手法における DAVIS テストセットと Set8 テストセットにおける生成画像の PSNR の値を比較している．ノイズレベルが既知である場合と同様に，PTFN は従来手法と比較して優れた性能を記録している．

モデルの軽量さ: 図 5.5 は，480p 画像 1 枚あたりの計算量 (GMAC) を横軸，$\sigma = 50$ における DAVIS テストセットの PSNR の値を縦軸，PyTorch [19] における 480p 画像 1 枚あたりの推論時間をバブルの大きさとしたバブルチャートである．表 5.3 や図 5.5 から，PTFN は従来法と比較して極めて少ない計算量で高品質なノイズ除去をすることが可能であることがわかる．具体的には PTFN は BSVD の約 16.7% の計算量である

表 5.1　ノイズレベルが既知の場合の DAVIS テストセットと Set8 テストセットにおける PSNR (dB) の値．σ は AWGN のノイズレベルを表しており，Avg は各ノイズレベルの PSNR の平均値を記している．各項目において最も優れた値は太字にしてあり，2 番目に優れた値には下線を引いている．

	DAVIS PSNR (dB)					
Method	$\sigma = 10$	$\sigma = 20$	$\sigma = 30$	$\sigma = 40$	$\sigma = 50$	Avg
VNLB [26]	38.85	35.68	33.73	32.32	31.13	34.34
V-BM4D [24]	37.58	33.88	31.65	30.05	28.80	32.39
DVDNet [27]	38.13	35.70	34.08	32.86	31.85	34.52
FastDVDNet [6]	38.71	35.77	34.04	32.82	31.86	34.64
PaCNet [28]	**39.97**	36.82	34.79	33.34	32.20	35.42
BSVD [18]	39.81	36.82	35.09	33.86	32.91	35.70
PTFN Half	39.18	36.11	34.37	33.16	32.24	35.01
PTFN	39.72	<u>36.86</u>	<u>35.20</u>	<u>34.02</u>	<u>33.10</u>	<u>35.78</u>
PTFN-L Half	39.44	36.48	34.77	33.56	32.62	35.37
PTFN-L	<u>39.86</u>	**37.05**	**35.41**	**34.24**	**33.34**	**35.98**
	Set8 PSNR (dB)					
Method	$\sigma = 10$	$\sigma = 20$	$\sigma = 30$	$\sigma = 40$	$\sigma = 50$	Avg
VNLB [26]	**37.26**	33.72	31.74	30.39	29.21	32.46
V-BM4D [24]	36.05	32.19	30.00	28.48	27.33	30.81
DVDNet [27]	36.08	33.49	31.79	30.55	29.56	32.29
FastDVDNet [6]	36.44	33.43	31.68	30.46	29.53	32.31
PaCNet [28]	<u>37.06</u>	<u>33.94</u>	32.05	30.70	29.66	32.68
BSVD [18]	36.74	33.83	32.14	30.97	30.06	32.75
PTFN Half	36.35	33.42	31.72	30.54	29.64	32.33
PTFN	36.68	33.85	<u>32.21</u>	<u>31.06</u>	<u>30.18</u>	<u>32.80</u>
PTFN-L Half	36.52	33.64	31.97	30.81	29.91	32.57
PTFN-L	36.82	**33.99**	**32.36**	**31.22**	**30.34**	**32.95**

にもかかわらず BSVD を上回る性能を記録しており，PyTorch における推論速度も同等の値を持っている．BSVD と PTFN はどちらも U-Net ライクな構造をしているが，

表 5.2　ノイズレベルが既知ではない場合の DAVIS テストセットと Set8 テストセットにおける PSNR (dB) の値．σ は AWGN のノイズレベルを表しており，Avg は各ノイズレベルの PSNR の平均値を記している．各項目において最も優れた値は太字にしてあり，2 番目に優れた値には下線を引いている．

	DAVIS PSNR (dB)					
Method	$\sigma = 10$	$\sigma = 20$	$\sigma = 30$	$\sigma = 40$	$\sigma = 50$	Avg
ReMoNet [31]	38.97	35.77	33.93	32.64	31.65	34.59
BSVD-blind [18]	39.68	36.66	34.91	33.68	32.72	35.53
PTFN-blind	<u>39.65</u>	<u>36.76</u>	<u>35.08</u>	<u>33.90</u>	<u>32.98</u>	<u>35.67</u>
PTFN-L-blind	**39.79**	**36.96**	**35.31**	**34.14**	**33.22**	**35.88**
	Set8 PSNR (dB)					
Method	$\sigma = 10$	$\sigma = 20$	$\sigma = 30$	$\sigma = 40$	$\sigma = 50$	Avg
ReMoNet [31]	36.29	33.34	31.59	30.37	29.44	32.21
BSVD-blind [18]	36.54	33.70	32.02	30.85	29.95	32.61
PTFN-blind	<u>36.58</u>	<u>33.78</u>	<u>32.14</u>	<u>30.99</u>	<u>30.11</u>	<u>32.72</u>
PTFN-L-blind	**36.64**	**33.89**	**32.28**	**31.14**	**30.26**	**32.84**

BSVD はクラシカルな CNN ネットワークを用いているのにに対して，PTFN では計算量が少なくとも優れた特徴抽出能力を持つモダンな ConvBlock や PTF Block を用いている．実際に BSVD に用いられている ConvBlock は 2 層の Conv2d であり，計算量は式 (5.6) で表されるが，PTFN の ConvBlock の計算量は式 (5.7) で表される．ただし，h, w, c はそれぞれ特徴マップの縦幅，横幅，チャンネル数を表している．実際に $h, w, c = 480, 856, 32$ のとき，BSVD の ConvBlock の計算量は 7.57×10^9 であるのに対し，PTFN で用いられている ConvBlock では 1.92×10^9 と大幅に計算量が削減されている．

$$\underbrace{9hwc^2}_{\text{3x3 Conv2d}} + \underbrace{9hwc^2}_{\text{3x3 Conv2d}} = 18hwc^2 \tag{5.6}$$

$$\begin{aligned}\underbrace{2hwc^2}_{\text{1x1 Conv2d}} + \underbrace{18hwc}_{\text{3x3 DWConv2d}} + \underbrace{2hwc^2}_{\text{1x1 Conv2d}} \\ = 4hwc^2 + 18hwc^2\end{aligned} \tag{5.7}$$

また，PTFN Half に関しても，FastDVDNet の約 15.4% の計算量で，より優れた性能を示しており，推論速度も上昇している．

表 5.3　提案手法と従来手法のモデルの軽量さに関する比較．Runtimes (s/image) は Python または PyTorch 実装における画像 1 枚あたりの処理時間，GMACs/image は画像 1 枚あたりの計算量を表している．480p は 854 × 480，720p は 1280 × 720，1080p は 1920 × 1080，の解像度を意味する．処理時間について，GPU ベースの手法は NVIDIA RTX 3090 24GB，CPU ベースの手法は Intel Xeon CPU E5-1650 v4 を計算機に用いた．処理時間と性能のトレードオフの観点から比較するために，DAVIS テストセットの $\sigma = 50$ における PSNR の値も表 5.3 に掲載している．OOM は Out Of Memory の略であり，画像あたりの計算量は GPU ベースの手法のみを算出している．

		PSNR (dB)	Runtimes (s/image)			GMACs/image	
Method	Hardware	$\sigma = 50$	480p	720p	1080p	480p	720p
VNLB [26]	CPU	31.13	128.97	264.18	583.13	N/A	N/A
V-BM4D [24]	CPU	28.80	151.34	335.81	763.12	N/A	N/A
DVDNet [27]	CPU+GPU	31.85	3.71	6.31	OOM	N/A	N/A
FastDVDNet [6]	GPU	31.86	0.037	0.081	0.18	263.20	498.11
PaCNet [28]	GPU	32.20	3.16	OOM	OOM	177.42	OOM
BSVD [18]	GPU	32.83	0.047	0.10	OOM	486.35	1090.88
PTFN Half	GPU	32.24	0.026	0.056	0.13	40.60	91.06
PTFN	GPU	33.10	0.051	0.11	0.25	81.16	182.03
PTFN-L Half	GPU	32.62	0.037	0.082	0.18	55.74	125.04
PTFN-L	GPU	33.34	0.074	0.16	0.37	111.45	249.98

提案手法はメモリ消費の観点からも優秀である．最先端手法の BSVD は，NVIDIA RTX 3090 24GB では 1080p 画像を処理できなかったが，PTFN は 1080p 画像を処理することが可能である．また，より軽量な PTFN Half はさらに高解像度の画像を処理することが可能であり，同じ GPU で 2K 解像度 (2560 × 1440) の画像を 1 枚あたり 0.22 s で処理することが可能である．

5.3.4　定性的比較

図 5.6 と図 5.7 は，それぞれ Set8 テストセットに含まれる Snowboard と Tractor という動画のある 1 フレームにおけるノイズ除去画像の比較をしている．図 5.6 の空の部分に注目するとわかるように，PTFN は従来手法と比較して綺麗にノイズ除去ができており，優れたノイズ除去性能を持っていることがわかる．

図 5.7 のノイズ画像では，文字が完全に潰れてしまっているため単一の画像のみでの

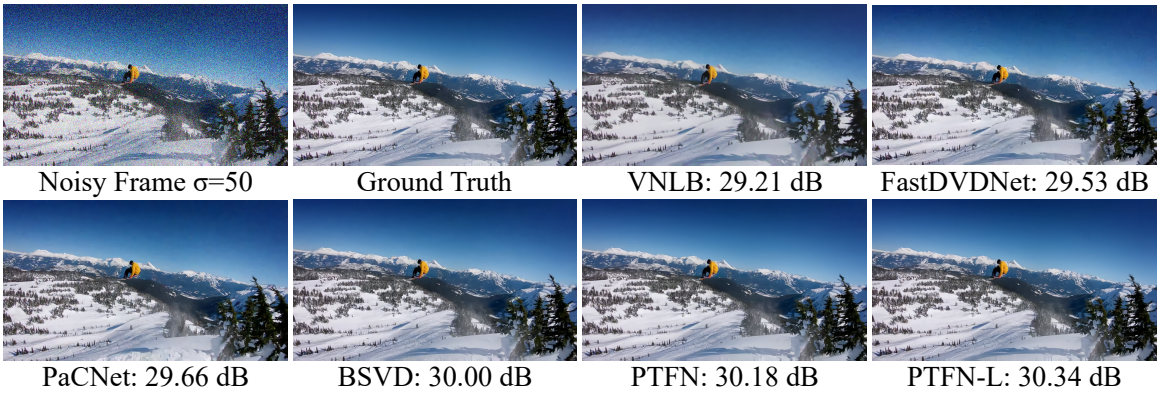


図 5.6　Set8 テストセットの動画 Snowboard の 1 フレームにおける提案手法と従来法のノイズ除去画像の比較.

復元は不可能である．実際に図 5.7 において現在，単一画像のノイズ除去で最先端の性能を持つ Restormer [14] による復元画像は文字を復元できていない．よって，文字まで復元ができている画像を出力するためには，異なる動画フレームからの情報を集約して復元する必要があり，ネットワークが動画シーケンスの時間的な関係性を十分に捉えている必要がある．PTFN の復元画像は，実際に読めるぐらいに文字の復元ができているため，十分に時間的な関係性が捉えられていると考えられる．

5.3.5　新規性の有効性の検証

表 5.4 は，PTFN のネットワーク構造のアブレーションスタディに関する表であり，ノイズ除去性能や推論の速さといった観点から検証をしている．

ConvBlock の構造: [49, 50] は，クラシカルな ConvBlock より優れた性能を持つモダンな ConvBlock を提案しているが，それらが動画のノイズ除去に適したものであるかといった検証はされていない．よって，本研究では動画のノイズ除去に適した ConvBlock を探し出すための検証も行った．

表 5.4 の①は PTFN の ConvBlock に含まれる Depthwise Conv のカーネルサイズを 3 から 7 に変更した場合を表している．この場合 DAVIS データセットのノイズレベル $\sigma = 50$ における PSNR の値が 0.13 dB 向上している．これは，カーネルサイズが増加することで，画像内の空間的な冗長性をより捉えることが可能になり精度が向上したと考えられる．しかし，PyTorch における 1080p 画像の 1 枚あたりの処理時間が 24% 増加してしまっている．

表 5.4 の②，③は，ConvBlock の Depthwise Conv の位置を繰り上げた場合を表し

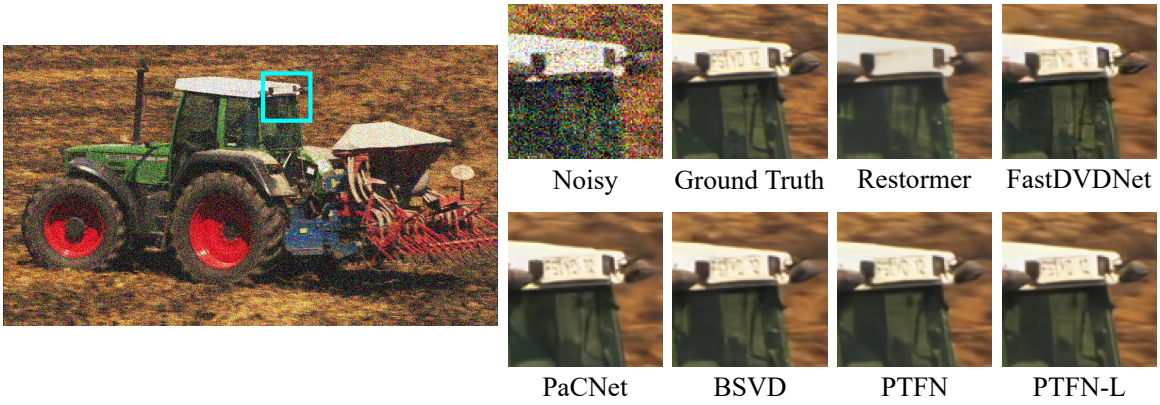


図 5.7　Set8 テストセットの動画 Tractor の 1 フレームにおける提案手法と従来法のノイズ除去画像の比較．シアンの枠の部分を拡大して比較している．また，比較検証のため，最先端の単一画像のノイズ除去手法である Restormer によるノイズ除去結果も掲載している．

ており，②は，カーネルサイズが 3，③はカーネルサイズが 7 の場合を表している．Depthwise Conv の位置を繰り上げることは，[49] において導入されており，画像分類のタスクにおいてはその有効性が示されている．②は PSNR の値が 0.04 dB 減少しているが速度も 4% 減少しており，③は PSNR の値が 0.04 dB 増加しているが速度は 8% 増加している．

図 5.8 は，表 5.4 の④以外の項目に対して性能と速度のトレードオフを検証するために描いたグラフである．図 5.8 から，カーネルサイズが 3 で，Depthwise Conv の位置を繰り上げないモデルである PTFN が，速度と性能のトレードオフの観点から最適なモデルであると解釈できる．

Pseudo Temporal Fusion (PTF): 本研究で提案している PTF の有効性を検証するために，PTF Block の PTF を GELU に変更した場合④のネットワークに対しての比較検証を行った．表 5.4 から，PTF を導入することによって DAVIS テストセットの PSNR の値が 0.32 dB 上昇していることがわかる．PTF の追加によってネットワークの時間的な関係性を捉える能力が向上したことで，生成動画の品質が向上したと考えられる．

Intermediate Loss: 表 5.5 では，学習時の損失関数 (5.5) における α の値を変更した際の性能を比較している．$\alpha = 0$ は，Intermediate Loss を用いていない場合を表している．表 5.5 からは，$\alpha = 0.1$ で訓練した場合が最も性能が良くなることがわかる．これは，多段階のノイズ除去が高品質のノイズ除去に有用であり，Intermediate Loss によ

表 5.4　PTFN のネットワーク構造のアブレーションスタディ．PSNR の値はノイズレベル $\sigma = 50$ の DAVIS テストセットにおける値である．速度測定に使用した GPU は NVIDIA RTX 3090 24GB で，使用したフレームワークは PyTorch である．

Model	Moved Up	Kernel Size	PSNR (dB)	Runtimes (s/1080p image)	× Speed Up
①	No	7	33.23	0.31	-24.00%
②	Yes	3	33.06	0.24	4.00%
③	Yes	7	33.14	0.27	-8.00%
④	No	3	32.78	0.21	16.00%
PTFN	No	3	33.10	0.25	0.00%

表 5.5　損失関数の Intermediate Loss の項の係数を変化させたときの，DAVIS テストセットにおける PSNR の値．最も優れた値は太字にしてある．

α	PSNR (dB)
0.0	33.02
0.1	**33.10**
1.0	32.91

り多段階のノイズ除去というタスクをより意識した訓練が可能となったことが要因であると考えられる．

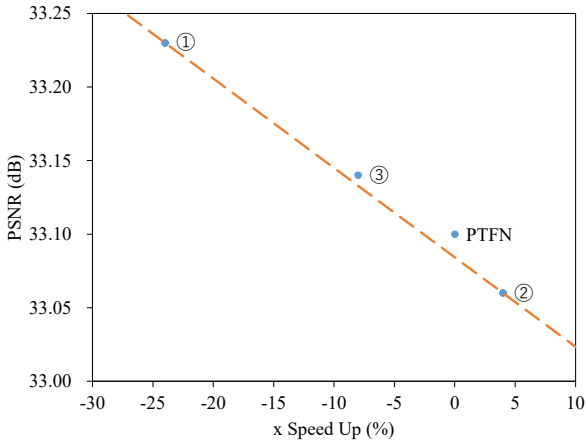


図 5.8　表 5.4 の④以外の項目に対して横軸を PTFN に対する速度の変化，縦軸を DAVIS テストセットにおけるノイズレベル $\sigma = 50$ の PSNR の値としたグラフ．オレンジの直線は①と②をつなぐ直線である．

第 6 章

姿勢変換

6.1 提案手法

提案ネットワークの全体は図 6.1 に示される．多くの従来法ではソースの人物画像，その姿勢情報，ターゲットの姿勢情報以外にも追加のデータ・タスクが必要である．推論時にもこのデータは必要となるため，ハイパーパラメータの調節が難化，学習時間が増加し，実用性が制限される．それに対し，提案手法は追加のデータ・タスクを必要としない．

ネットワークは，ソースの人物画像，その姿勢情報を結合したもの I_s を入力とし，Shallow Feature Extraction モジュールで，マルチスケールの浅い特徴を抽出する．提案手法は，姿勢変換というタスクを「大まかな姿勢の変換」と「詳細なテクスチャの生成」というタスクに分離して取り組んでいる．抽出された特徴のうち，大域的な特徴を有している低解像度の特徴は Axial Transformer Transformation Block (ATTB) によって「大まかな姿勢の変換」が行われる．高解像度の特徴は CNN Transformation Block (CTB) によって「詳細なテクスチャを生成」が行われる．このように姿勢変換のタスクを 2 つのサブタスクに分割することで，ネットワークが学習するべきタスクを明確化している．

以下，図 6.1 において，F_{si} が出力される解像度と同じ解像度で計算をしている部分をレベル $L = i$ と表記する．

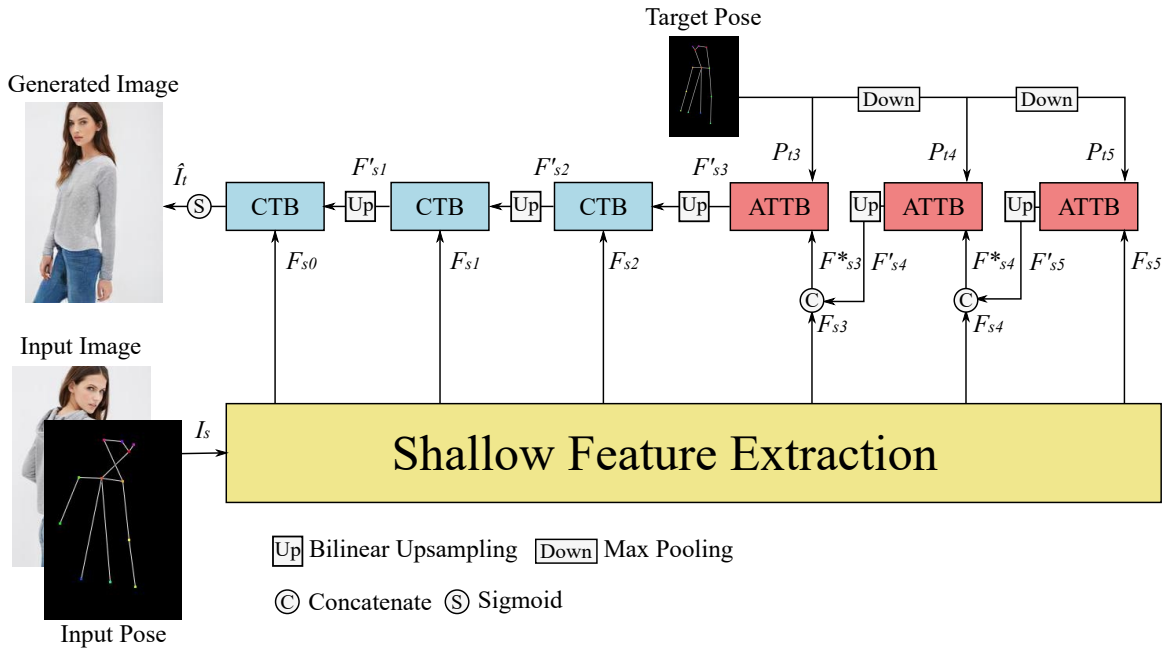


図 6.1　提案ネットワークの全体図．提案手法のネットワークは Shallow Feature Extraction, Axial Transformer Transformation Block (ATTB), CNN Transformation Block (CTB) で構成されている．

6.1.1　Shallow Feature Extraction

Shallow Feature Extraction は図 6.2 の (a) に示される入力の情報 I_s の浅い特徴を抽出するためのモジュールで，Conv2d, Layer Normalization, ReLU からなる層で構成されている．Shallow Feature Extraction はマルチスケールのモジュールであり，出力は $F_{si}(i = 0, 1, 2, 3, 4, 5)$ となっている．I_s のサイズが $H \times W$ の場合，F_{si} のサイズは $\frac{H}{2^i} \times \frac{W}{2^i}$ となる．各層のチャンネル数と深さは $L = 0, 1$ の場合は $128, 2$ であり，$L = 2, 3, 4$ の場合は $64, 4$ である．

6.1.2　Axial Transformer Transformation Block (ATTB)

Axial Transformer Transformation Block (ATTB) は，図 6.2 の (b) に示されるモジュールである．ATTB は N 層の Axial Transformer の Encoder-Decoder ブロックで構成されており，Shallow Feature Extraction から抽出された特徴量 F_{si}，もしくは下の解像度の ATTB からの出力 $F'_{s(i-1)}$ と F_{si} を結合した F^*_{si} を Encoder の入力と

する．ターゲットのポーズ情報 P_{ti} を Decoder の入力とする．Axial Transformer は，Axial Attention を用いており，縦横方向に受容野を制限しないため，CNN よりも画像の全体的な特徴を捉えることに適している．Axial Transformer の持つこの特徴は，「大まかな姿勢の変換」というタスクに有用である．また，Encoder-Decoder ブロックを採用することでネットワークがより姿勢情報を有効に活用できるため，ネットワークの性能や安定性が増す．各入力には位置エンコーディングとして Conditional Positional Encoding (CPE) を用いる．

Axial Transformer の Encoder は，Layer Normalization，Multi Head Axial Self Attention からなる層と Layer Normalization，カーネルサイズが 1×1 の Conv2d，GELU からなる層で構成されている．それぞれの層は Residual Connection で接続されている．

Axial Transformer の Decoder は，Layer Normalization，Multi Head Axial Self Attention からなる層，Layer Normalization，Multi Head Axial Attention からなる層と Layer Normalization，カーネルサイズが 1×1 の Conv2d，GELU からなる層で構成されている．Multi Head Axial Attention は P_{ti} または前の Decoder ブロックの出力を query とし，Encoder からの出力を key, value とする．それぞれのブロックは Residual Connection で接続されている．

Encoder や Decoder のチャンネル数と N の値について，$L = 3$ のときは $64, N = 2$ で，$L = 4$ のときは $128, N = 2$ で，$L = 5$ のときは $128, N = 4$ である．

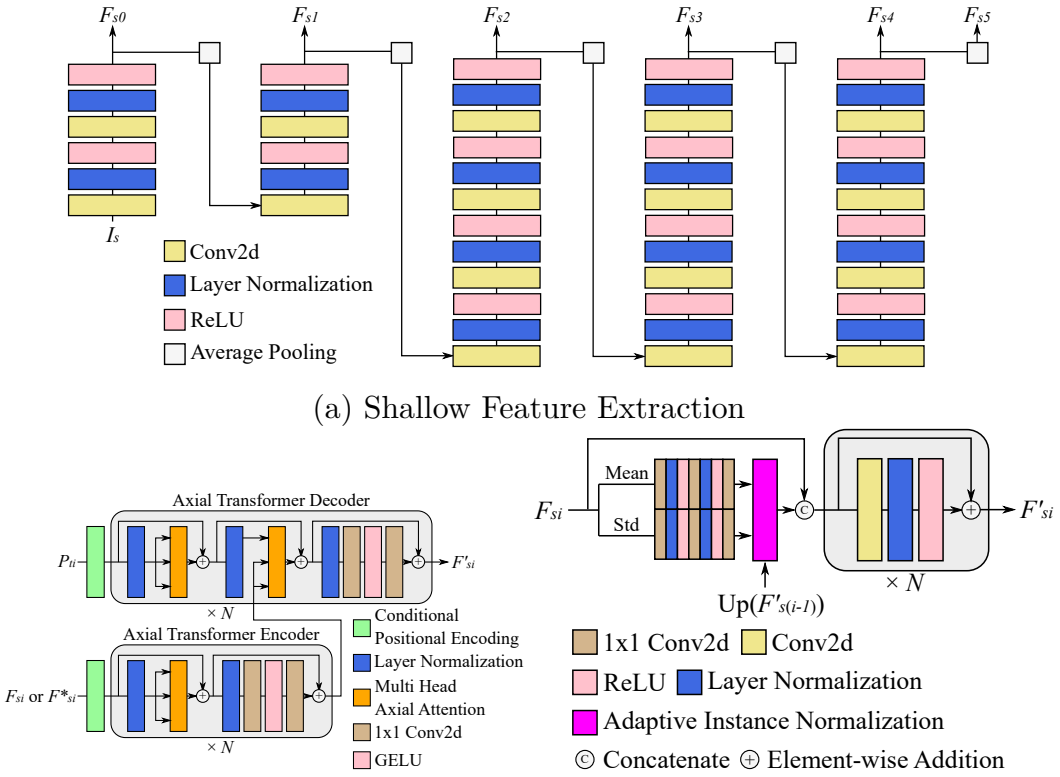
また，Axial Transformer の Decoder の query である P_{ti} について，入力画像の解像度が $H \times W$ の場合，P_{ti} の解像度は $\frac{H}{2^i} \times \frac{W}{2^i}$ となる．このように，提案手法はフル解像度の姿勢情報を使用せず，Max Pooling でダウンサンプリングした姿勢情報をネットワークに入力している．具体的には，訓練時のソース情報 I_s の解像度が 256×256 であるのに対し，ATTB に入力されるのは P_{t3}, P_{t4}, P_{t5} であり，それぞれ解像度は $32 \times 32, 16 \times 16, 8 \times 8$ である．フル解像度の姿勢情報を使用しないようにネットワークを設計したのは，ATTB が取り組むべきタスクである「大まかな姿勢の変換」を達成するには，低解像度の特徴マップのみを変換すれば十分であるという仮説に基づいている．この仮説の妥当性はアブレーションスタディの「ソースとターゲットの分離」で検証する．さらに，ATTB を含む Transformer ベースのモジュールは高解像度の特徴マップに対する計算量が著しく増加するという問題があるが，提案手法では低解像度の特徴マップのみを ATTB で処理しているため，計算量を削減できる．

6.1.3 CNN Transformation Block (CTB)

CNN Transformation Block (CTB) は，図 6.2 の (c) に示されるモジュールである．CTB は Adaptive Instance Normalization (AdaIN) [79] と，Conv2d，Layer Normalization，ReLU からなる N 個の Residual Block から構成される．姿勢変換タスクにおいて，テクスチャはほとんど変化がないため，そのことを有効に活用するために，AdaIN を採用した．

AdaIN のコンテント入力は，下の解像度のブロックの出力 $F'_{s(i-1)}$ であり，スタイル入力は F_{si} の平均と分散を 3 層のネットワークで計算した出力である．

CTB のチャンネル数と N の値について，$L = 0, 1$ のときは $64, N = 4$ で，$L = 2$ のときは $64, N = 6$ である．



(a) Shallow Feature Extraction

(b) Axial Transformer Transformation Block (ATTB)

(c) CNN Transformation Block (CTB)

図 6.2　提案ネットワークの各モジュールの詳細.

6.1.4 Loss Function

提案手法は，式 (6.1), (6.2) のような損失関数を用いて学習している．L_{adv} は敵対的損失であり，WGAN [80] と同様の目的関数を用いている．Discriminator は標準的な 6 層の CNN ネットワークを用いている．

L_{l1} はネットワークの推論画像とグラウンドトゥルースの間の絶対誤差である．L_{perc} は VGG19 [15] をバックボーンとしている，Perceptual Loss [60] である．L_{style} は，VGG19 [15] をバックボーンとしている，Style Loss [60] である．gp は，Gradient Penalty [74] であり，学習の安定化のために採用されている．$\lambda_i (i = 1, 2, 3, 4, 5)$ は，損失の比率を調節するためのハイパーパラメータである．

$$L_G = \lambda_1 L_{adv} + \lambda_2 L_{l1} + \lambda_3 L_{perc} + \lambda_4 L_{style} \tag{6.1}$$

$$L_D = \lambda_1 L_{adv} + \lambda_5 \mathrm{gp} \tag{6.2}$$

6.2 提案手法の特長

提案手法は姿勢変換を 2 つのサブタスクに分離し，それぞれに対応した処理をさせる構造を採用している．また，低周波成分を変換するための ATTB に Encoder-Decoder 構造を採用しており，姿勢の情報をより意識した変換を可能にしている．これらによって提案手法は競争的な性能を保ちつつ，追加のタスクやデータを必要としない軽量な変換を可能とした．

6.3 実験

6.3.1 セットアップ

データセット

定性的・定量的な比較のために 2 つのデータセット，Deep Fashion [81], Market-1501 [82] を用いた．これらのデータセットは姿勢変換タスクにおいて一般的に用いられているデータセットである [3, 35, 38, 40]．Deep Fashion は服飾した人物の高品質な画像 52,712 枚からなるデータセットであり，白い無地の背景で統一されている．画像の解像度は 256×176 である．Market-1501 は 263,632 枚の人物画像からなるデータ

表 6.1　提案手法といくつかの最新手法との定量的な比較．1 番良い値は太字，2 番目に良い値には下線を引いている．

	DeepFashion				Market-1501				
	SSIM↑	LPIPS↓	IS↑	FID↓	SSIM↑	LPIPS↓	IS↑	FID↓	#Params
XingGAN [35]	0.687	0.2929	2.878	44.808	<u>0.352</u>	0.3059	**3.201**	37.510	44.84M
GFLA [3]	**0.725**	<u>0.1869</u>	2.856	**7.332**	0.328	0.2815	0.2849	<u>28.042</u>	14.04M
MUST [37]	0.685	0.2467	**2.971**	17.220	-	-	-	-	51.45M
SPGNet [38]	0.702	0.2109	2.711	11.964	**0.378**	<u>0.2777</u>	2.942	30.520	117.13M
PISE [39]	0.691	0.2084	2.815	9.905	-	-	-	-	64.00M
DPTN [40]	0.707	0.1966	2.867	9.683	0.332	**0.2711**	2.965	28.678	<u>9.79M</u>
Ours	<u>0.718</u>	**0.1849**	<u>2.944</u>	<u>8.034</u>	0.311	0.2939	<u>3.091</u>	**23.307**	**8.83M**

セットであり，様々な視点や明度，背景の画像が存在する．画像の解像度は 128×64 である．人物のポーズ情報（18 ジョイントのキーポイント）は，OpenPose [83] によって抽出されている．公平な比較をするために，それぞれのデータセットに関して，訓練とテストの分け方は [3] と同様にした．

学習条件・ハイパーパラメータ

損失関数について，$\lambda_1 = 1, \lambda_2 = 2.5, \lambda_3 = 0.25, \lambda_4 = 250, \lambda_5 = 10$ とした．バッチサイズは 8 で，学習ステップ数は Deep Fashion では 500,000, Market-1501 では 100,000 である．両データセットにおいて，学習率は Generator と Discriminator 双方において 1.0×10^{-4} である．本手法では学習率減衰が使用されており，Deep Fashion においては，250,000 Step と 400,000 Step において学習率に 0.1 を掛けており，Market-1501 においては，50,000 Step と 80,000 Step において学習率に 0.1 を掛けている．最適化アルゴリズムは Adam [61] を使用しており $\beta_1 = 0.5, \beta_2 = 0.999$ である．

6.3.2　評価手法

既存の手法 [3, 35] に倣い，評価指標として Structural Similarity Index Measure (SSIM) [84], Learned Perceptual Image Patch Similarity (LPIPS) [68], Inception Score (IS) [85], Fréchet Inception Distance (FID) [86] を採用した．

6.3.3 定量的比較

提案手法といくつかの最新の手法 (XingGAN [35], GFLA [3], MUST [37], SPGNet [38], PISE [39], DPTN [40]) とを比較する．生成画像とモデルサイズに関する定量的な比較の結果は表 6.1 のようになっている．表 6.1 から，提案手法は最新の手法にも劣らない性能であり，8 つの評価指標の内 6 つが，1 番目または 2 番目に優れた性能を持っている．表 6.1 における従来法の多くは，追加のパース情報が必要 (MUST, SPGNet, PISE) であったり，追加のタスクを設定・学習 (GFLA, SPGNet, DPTN) する必要がある．しかし，提案手法はこれらの追加要素を用いていないにもかかわらず，優れた性能を記録している．この結果は，姿勢変換というタスクの特性を考慮してネットワークを構築する場合，パース情報やタスクを追加するよりも，Encoder-Decoder 構造のような，ネットワーク自体がポーズの情報をより有効に活用できる構造にしたり，Axial Transformer のような大域的な特徴を捉えるのに適した特徴抽出器を使用するほうが重要であることを示唆している．さらに，提案手法ではパース情報やタスクの追加による，データを用意する労力の増大や，ハイパーパラメータの調節や難化が起こらないので，実用性が高く，応用がしやすい．さらに，提案ネットワークのパラメータ数は 8.83M であり，これは表 6.1 にある最新手法のどれよりも少ない．提案ネットワーク (8.83M) は SPGNet (117.13M) の 7.54% のパラメータ数でネットワークが構成されている非常に小さいネットワークであるにもかかわらず，優れた性能を出せる．

6.3.4 定性的比較

図 6.3 は，比較したネットワークにおける Deep Fashion と Market-1501 の検証データに対する推論画像を示している．図 6.3 の左側が Deep Fashion の画像で，右側が Market-1501 の画像である．

提案手法と同じく，追加のパース情報やタスクを必要としていない XingGAN はぼやけた画像を生成してしまっているにもかかわらず，提案手法では他の最新手法に劣らないクオリティの画像を生成できている． 図 6.3 の 4 行目の Deep Fashion の画像において，提案手法と DPTN 以外は半ズボンか長ズボンかが曖昧な画像が生成されてしまっている．また，5 行目の画像においても不自然なテクスチャが生成されてしまっている場合がある．これは，これらの手法が主に近隣ピクセルからしか特徴抽出できない CNN をソース情報とターゲット情報の間の特徴抽出に用いており，画像全体

図 6.3　提案手法といくつかの最新手法との定性的な比較．左側は Deep Fashion での生成画像で，右側が Market-1501 での生成画像である．

の整合性を考慮することが困難であることが原因である．それに対し，受容野が広い Transformer を使用している DPTN と提案手法は，画像全体の整合性を考慮して画像を生成することが可能となっている．これが，Transformer ベースのネットワークを特徴抽出器にすることの利点である．

Market-1501 の画像に関しても，提案手法は他の手法に対して劣らないクオリティの画像を生成している．

図 6.3 の結果から，提案手法は従来手法のように大きな姿勢の変化にも対応可能であり，さらに画像全体の整合性も考慮できていることがわかる．

6.3.5　新規性の有効性の検証

特徴抽出器としての ATTB: 表 6.2 は，提案ネットワークにおいて特徴抽出器として，ATTB を用いた場合と，CNN, Swin Transformer を用いて学習した場合の精度を比較した表である．表 6.2 では，Axial Transformer を使用した場合のほうが他の特徴抽出器を用いたときよりも定量的に優れた性能となることを示している．CNN のフィルタ演算，Swin Transformer の Window Attention は，縦横方向の受容野を狭めるため，画像全体の整合性を考慮できない．それに対して，ATTD は縦横方向の受容野を狭め

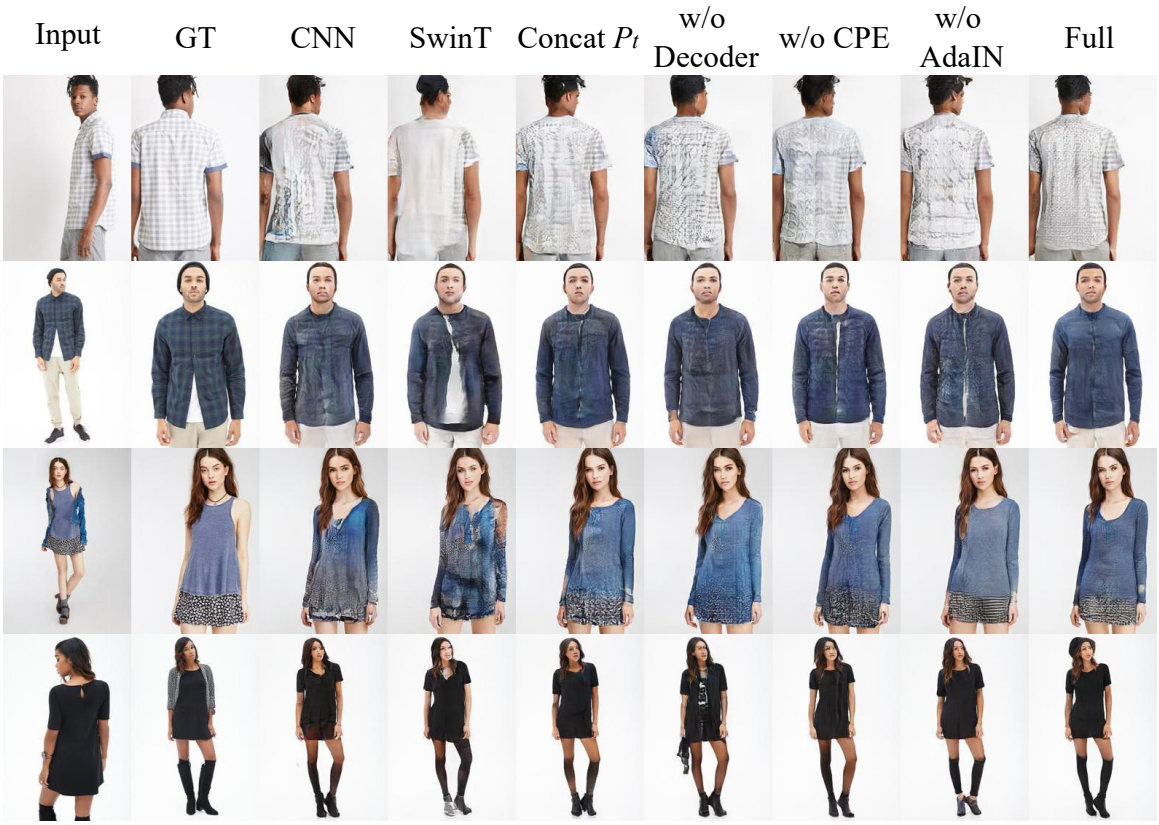


図 6.4　Deep Fashion におけるアブレーションスタディの定性的な比較.

ないため，画像全体の整合性が取れた画像が生成されているためだと考えられる．実際に，図 6.4 より，CNN や Swin Transformer を特徴抽出器に用いている場合，生成画像の全体の整合性が取れていない事がわかる．

ソースとターゲットの分離: 表 6.3 の concat P_t はターゲットの姿勢情報 P_t を入力情報 I_s に結合して学習した場合を指し，これはソース情報とターゲット情報を分離しない場合を表している．このとき，ATTB の構造は変化させていない．ATTB におけるソース情報とターゲット情報の分離の妥当性は「Encoder-Decoder 構造」で検証している．また，提案手法では ATTB が取り組むべきタスクである「大まかな姿勢の変換」を達成するには，低解像度の特徴マップのみを変換すれば十分であるという仮説を元に，低解像度のみで姿勢変換を行っている．concat P_t の場合は，ターゲットの姿勢情報が CTB にも入力されるため，低解像度のみならず高解像度でも姿勢の変換を行っている場合を表しているとも解釈できる．

表 6.3 では，concat P_t より，I_s と P_t を完全に分離してネットワークを学習したほうが優れた性能を出せることを示している．言い換えると，ソース情報とターゲット情

表 6.2　ATTB の Axial Transformer の部分を CNN, Swin Transformer (SwinT), ATTB にした場合の Deep Fashion における評価指標の値．精度が一番良い値は太字にしている．

	SSIM↑	LPIPS↓	IS↑	FID↓
CNN	0.707	0.2020	2.846	8.935
SwinT	0.701	0.2187	2.814	9.541
ATTB	**0.718**	**0.1849**	**2.944**	**8.034**

表 6.3　Deep Fashion におけるアブレーションスタディの定量的な比較．1 番良い値は太字，2 番目に良い値には下線を引いている．

	SSIM↑	LPIPS↓	IS↑	FID↓
concat P_t	<u>0.717</u>	0.1901	2.914	8.935
w/o Decoder	0.710	0.1990	2.918	8.440
w/o CPE	0.715	0.1941	2.931	8.317
w/o AdaIN	0.716	<u>0.1882</u>	**2.951**	<u>8.035</u>
Full	**0.718**	**0.1849**	<u>2.944</u>	**8.034**

報を結合して CNN で演算することで精度が向上しないことを示している．これは，近隣ピクセルからしか特徴抽出できない CNN では，ソース情報とターゲット情報間から有用な特徴を捉えられないことが理由と考えられる．さらに，ソース情報とターゲットの情報を分離することで，ネットワークが学習するべきタスクを明確化する効果があり精度が向上すると考えられる．また，concat P_t の場合にネットワークの精度が向上していないことから，低解像度の特徴マップのみを変換すれば十分であるという仮説は，妥当性があると考えられる．

Encoder-Decoder 構造: 表 6.3 の w/o Decoder は，ATTB のブロック数 N の Encoder と Decoder をブロック数 $2N$ の Encoder のみに置き換えた場合を表している．表 6.3 では，ブロック数 N の Encoder と Decoder を用いたほうがブロック数 $2N$ の Encoder のみを用いた場合よりも精度が向上していることを示している．これは，Encoder-Decoder 構造により，ネットワークがより強く姿勢情報を意識するため，姿勢情報を有効に活用して学習ができたからだと考えられる．

また，Encoder-Decoder 構造には，ネットワークが単なるコピーを出力し，学習を

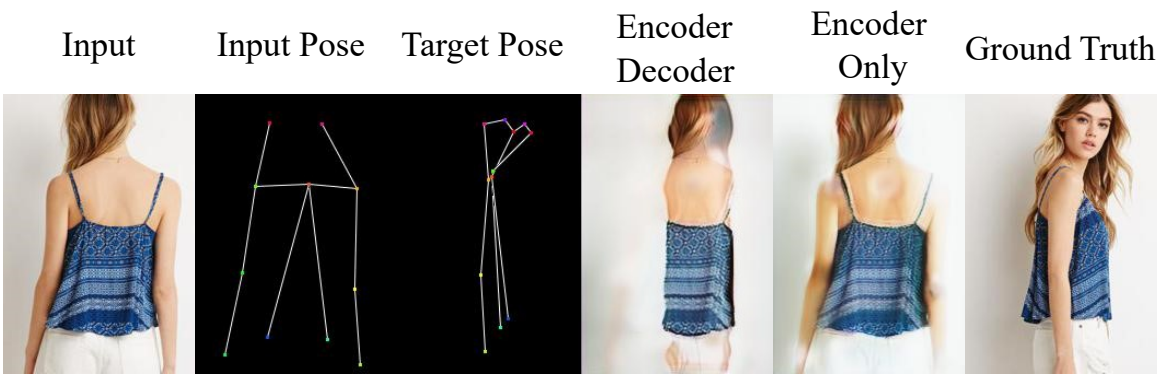


図 6.5　学習の初期段階 (Step 1,000) における Encoder-Decoder 構造の場合と Encoder のみの場合の Deep Fashion における生成画像.

妨げることを防ぐ効果がある．図 6.5 に示されるように，学習の初期段階 (全 500,000 Steps の内 Step 1,000) において，Encoder のみのネットワークは入力画像のコピーを作成してしまっているが，Encoder-Decoder 構造をもつモデルの場合はターゲットの姿勢を反映した画像を出力できている．

図 6.6 は，学習の初期段階 (全 500,000 Steps の内 Step 20,000 まで) における Encoder-Decoder 構造の場合と Encoder のみの場合のネットワークの FID の値を表している．Encoder のみの場合は，学習の初期段階において入力のコピーを生成してしまっているため，FID が低い値となっているが，その後 FID が急激に上昇している．その後は Encoder-Decoder 構造の場合の FID より高い FID となってしまっている．これは，Encoder のみの場合は，コピーを生成することをやめて，姿勢の変換を学習し始める際に大きく生成画像が崩れてしまうという学習の不安定さが原因だと思われる．図 6.5 に示されるように，Encoder-Decoder 構造を採用した場合は，最初から姿勢情報を反映して学習を行うため，安定した学習が可能であり，精度的にも優れたネットワークになる．

CPE の有無: 表 6.3 の w/o CPE は，ATTB の Conditional Positional Encoding (CPE) を取り除いて学習した場合を表している．表 6.3 は，CPE を追加したほうが精度が向上することを示している．CPE は，位置エンコーディングの手法の 1 つであり，ATTB の入力に対して適用されている．このモジュールは Axial Attention が位置情報を考慮できないのに対して，位置情報を特徴マップに埋め込む役割がある．よって，CPE の適用による位置情報の考慮によって，精度が上がったと考えられる．

また，図 6.4 の 4 行目の，CPE がない場合の生成画像は靴下かタイツかが曖昧になっている．CPE が存在しない場合は位置情報を考慮できないため，服の境目の情報がう

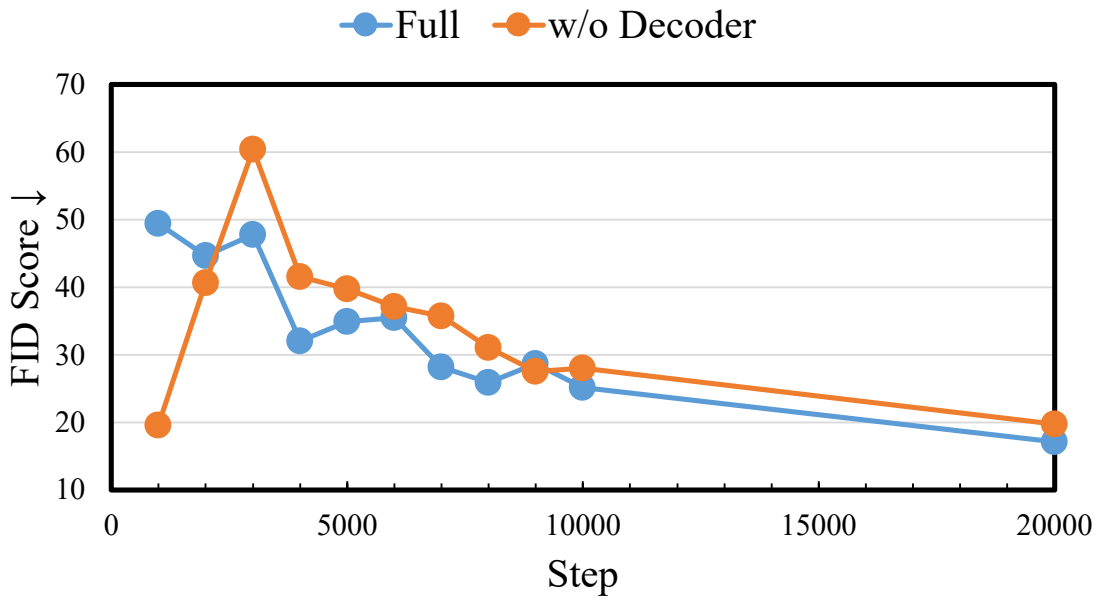


図 6.6　Encoder-Decoder 構造 (Full) の場合と Encoder のみの場合 (w/o Decoder) の，学習の初期段階 (Step 20,000 まで) における Deep Fashion の生成画像に対する FID の比較.

まく捉えられず，生成画像の整合性に悪影響が生じると考えられる．逆に，これは CPE のような位置エンコーディングを採用することで，生成画像の整合性に良い影響を与えられる可能性があることを示唆している．

AdaIN の有無: 表 6.3 の w/o AdaIN は，CTB の Adaptive Instance Normalization (AdaIN) を取り除いて学習した場合を表している．表 6.3 は，AdaIN を追加したほうが精度が少し向上することを示している．AdaIN は，スタイル変換でよく用いられているモジュールである．姿勢変換は，入力とターゲットで全体的なテクスチャがほとんど変化しない．AdaIN によって，入力のテクスチャを良く出力に反映できたため精度が向上したのだと考えられる．実際に，図 6.4 の 1 行目より，AdaIN が存在する場合の方が自然なテクスチャを生成できているとわかる．

第 7 章

非ペア画像変換

7.1　提案手法

提案手法は図 7.1 のように表される．提案手法では，画像ドメインにおける変換の他に，事前学習済みの Salient Object Detection ネットワークで抽出したサリエンシード

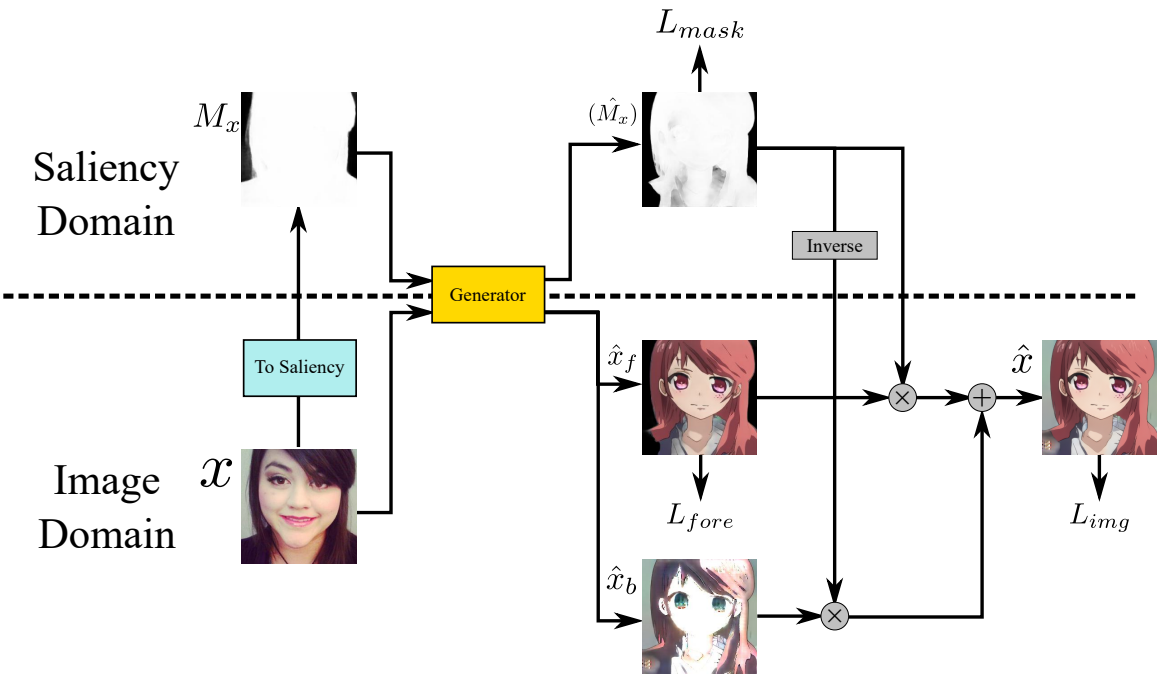


図 7.1　提案手法のネットワーク図．提案手法は学習済みの U2Net でサリエンシーマップを抽出してから，変換したサリエンシーマップ，変換画像の前景，変換画像の背景を出力している．

メインの変換をする．まずは，入力画像 x が与えられたとき，事前学習済みの Salient Object Detection ネットワークである U2Net+ [87] で前景マスク M_x を抽出する．Generator は x, M_x をもとに変形した前景マスク $(\hat{M_x})$，前景画像 $\hat{x}_f$，背景画像 $\hat{x}_b$ を出力する．サリエンシードメインは変換するべきオブジェクトとそれ以外を区別しているが，$M_x \cap \overline{(\hat{M_x})}$ の部分を出力するために $\hat{x}_b$ を出力している．最終的な出力画像 $\hat{x}$ は $\hat{x} = (\hat{M_x}) \otimes \hat{x}_f + \mathrm{Inv}((\hat{M_x})) \otimes \hat{x}_b$ で計算される．ただし，$\otimes$ は画素ごとの乗算，Inv はマスクの反転を意味する．

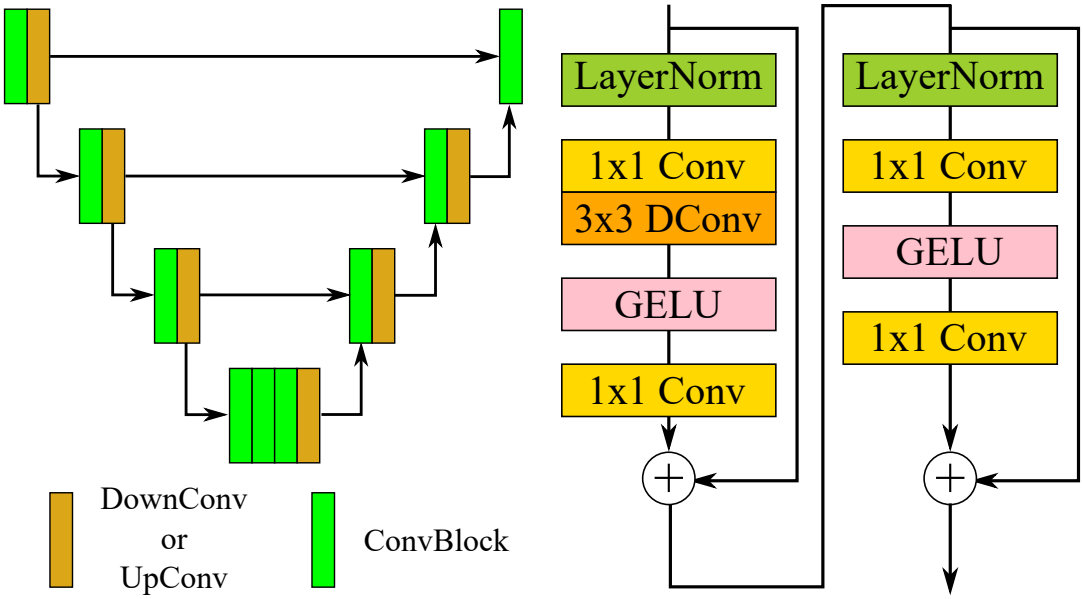
提案手法では画像ドメインの他にサリエンシードメインの変換を行い，サリエンシーに対する損失を計算している．また，変換されたサリエンシーマップを出力画像に明示的に反映している．サリエンシーマップは変換するべき物体の位置・形状を表現するため，サリエンシードメインでの変換を学習することで，変換した画像の形状や，空間配置が改善される．提案手法では背景部分についても変換後の画像を生成しているが，これは変換前のサリエンシーには含まれ，変換後のサリエンシーには含まれない領域を生成するためである．

変換に応じて形状が大きく変化しない場合は，サリエンシードメインのマスク変換や背景の生成が不要であるため，Generator は $\hat{x}_f$ のみを生成し，出力画像は $\hat{x} = M_x \otimes \hat{x}_f + \mathrm{Inv}(M_x) \otimes x$ で表される．

以下の節では提案手法を構成する，Generator，Salient Object Detection ネットワークの解説をする．

7.1.1 Generator

提案手法で使用する Generator は図 7.2 (a)，Generator の中の Convblock は図 7.2 (b) に示される．Generator は U-Net [5] ライクなネットワークであり，各スケールで ConvBlock を適用している．ダウンスケールにはストライドが 2 の 2×2 Conv，アップスケールには 1×1 Conv でチャンネル数を 4 倍にした後に Pixel Suffle [71] を用いている．ConvBlock は Layer Normalization [59]，3×3 Depthwise Conv [52]，1×1 Conv，GELU [56] から構成される 2 つの残差ブロックから構成される．ConvBlock について，GELU の前の 1×1 Conv でチャンネル数が 2 倍になり，GELU の後の 1×1 Conv でチャンネル数が元に戻される．提案する Generator は，従来手法でよく用いられている CycleGAN と同型のネットワークよりも計算量が小さく軽量である．しかし，本手法は変換に集中すべき前景部分に変換リソースが集中される構造であるため，性能も向上している．



(a) Generator の構造.　　(b) Generator の ConvBlock 構造.

図 7.2　Generator とそれに含まれる ConvBlock の構造.

7.1.2　サリエンシードメインへの変換

サリエンシードメインへの変換には事前学習済みの U2Net+ [87] を用いている．U2Net+ は軽量であることが特長であり，サリエンシードメインへの変換がボトルネックとなってモデルが肥大化することを防いでいる．

7.1.3　損失関数

実験では，サリエンシードメインのマスク，画像ドメインの前景部分，画像全体に対してそれぞれ敵対的な学習を行う．画像全体だけでなく，画像ドメインの前景部分に対する敵対的学習によって，よりメインオブジェクトを強調した訓練ができる．提案手法は LSGAN [70] を用いて敵対的な学習を行っている．提案手法では [44] と同様に，Maximum spatial perturbation consistency を利用した非ペア画像変換を行っており，摂動を加えた画像に対する損失項もある．

Generator の損失は (7.1) に示される．

$$L_G = \lambda_{maskG} L_{maskG} + \lambda_{foreG} L_{foreG} + \lambda_{imgG} L_{imgG} \tag{7.1}$$

ただし，$L_{maskG}, L_{foreG}, L_{imgG}$ はそれぞれサリエンシードメインのマスク，画像ドメインの前景部分，画像全体に対する損失を表しており，$\lambda_{maskG}, \lambda_{foreG}, \lambda_{imgG}$ は係数である．(7.1) の L_{maskG} は以下のように構成される．

$$\begin{aligned} L_{maskG} = L_{madv} + L_{madvpert} + L_{midt} \\ + L_{midtpert} + L_{mmspc} \end{aligned} \tag{7.2}$$

ただし，$L_{madv}, L_{madvpert}, L_{midt}, L_{midtpert}$ はサリエンシードメインのマスクとその摂動画像の敵対的損失と Identity mapping loss [42] であり，L_{mmspc} はサリエンシードメインのマスクにおける Maximum spatial perturbation consistency loss [44] である．(7.1) の L_{foreG} は以下のように構成される．

$$\begin{aligned} L_{foreG} = L_{fadv} + L_{fadvpert} + L_{fidt} \\ + L_{fidtpert} + L_{fmspc} \end{aligned} \tag{7.3}$$

ただし，$L_{fadv}, L_{fadvpert}, L_{fidt}, L_{fidtpert}$ は画像ドメインの前景部分とその摂動画像の敵対的損失と Identity mapping loss であり，L_{fmspc} は画像ドメインの前景部分における Maximum spatial perturbation consistency loss である．(7.1) の L_{imgG} は以下のように構成される．

$$\begin{aligned} L_{imgG} = L_{iadv} + L_{iadvpert} + L_{iidt} \\ + L_{iidtpert} + L_{imspc} + L_{irec} \end{aligned} \tag{7.4}$$

ただし，$L_{iadv}, L_{iadvpert}, L_{iidt}, L_{iidtpert}$ は画像全体とその摂動画像の敵対的損失と Identity mapping loss，L_{imspc} は画像全体における Maximum spatial perturbation consistency loss，L_{irec} は入力画像から抽出したサリエンシードメインのマスク M_x とネットワークで変換したサリエンシードメインのマスク $(\hat{M_x})$ のどちらにも含まれない領域に対する再構成損失である．

Generator の損失における Identity mapping loss や再構成損失は学習の安定化のために採用されている．

Discriminator の損失は (7.5) に示される．

$$\begin{aligned} L_D = \lambda_{maskD} L_{maskD} + \lambda_{foreD} L_{foreD} \\ + \lambda_{imgD} L_{foreD} + \lambda_{const} L_{const} \end{aligned} \tag{7.5}$$

ただし，$L_{maskD}, L_{foreD}, L_{imgD}$ はそれぞれサリエンシードメインのマスク，画像ドメインの前景部分，画像全体に対する損失を表しており，$\lambda_{maskD}, \lambda_{foreD}, \lambda_{imgD}$ は係数である．$L_{const}, \lambda_{const}$ は [44] で用いられている，画像に摂動に関する制約項とその係

数である．(7.5) の L_{maskD} は以下のように構成される．

$$L_{maskD} = L_{madv} + L_{madvpert} + L_{mmspc} \tag{7.6}$$

ただし，$L_{madv}, L_{madvpert}$ はサリエンシードメインのマスクとその摂動画像の敵対的損失であり，L_{mmspc} はサリエンシードメインのマスクにおける Maximum spatial perturbation consistency loss である．(7.5) の L_{foreD} は以下のように構成される．

$$L_{foreD} = L_{fadv} + L_{fadvpert} + L_{fmspc} \tag{7.7}$$

ただし，$L_{fadv}, L_{fadvpert}$ は画像ドメインの前景部分とその摂動画像の敵対的損失であり，L_{fmspc} は画像ドメインの前景部分における Maximum spatial perturbation consistency loss である．(7.5) の L_{imgD} は以下のように構成される．

$$L_{imgD} = L_{iadv} + L_{iadvpert} + L_{imspc} \tag{7.8}$$

ただし，$L_{iadv}, L_{iadvpert}$ は画像全体のマスクとその摂動画像の敵対的損失であり，L_{imspc} は画像全体における Maximum spatial perturbation consistency loss である．

それぞれの損失関数の具体的な式は以降の小節で解説する．

7.1.4 L_{maskG}, L_{maskD} の各項の詳細

L_{maskG} と L_{maskD} の各項の詳細は，(7.9)，(7.10)，(7.11)，(7.12) および (7.13) に示されている．M_x と M_y は，入力画像 x およびターゲットドメインの画像 y に対するサリエンシーマップである．$\hat{}$ は Generator で変換したあとの出力を表し，^{p} は摂動関数によって行われる変換を表す．(7.13) のように，摂動関数と Generator が可換であるという制約は，[44] によって提案された非ペア画像変換を実現するための制約である．

$$L_{madv} = \begin{cases} \frac{1}{2}||D((\hat{M_x})) - 1||_2 & \text{for G} \\ \\ \frac{1}{2}||D(M_y) - 1||_2 + \frac{1}{2}||D((\hat{M_x}))||_2 & \text{for D} \end{cases} \tag{7.9}$$

$$L_{madvpert} = \begin{cases} \frac{1}{2}||D((\hat{M_x})^p) - 1||_2 & \text{for G} \\ \\ \frac{1}{2}||((M_y)^p) - 1||_2 + \frac{1}{2}||D((\hat{M_x})^p))||_2 & \text{for D} \end{cases} \tag{7.10}$$

$$L_{midt} = ||(\hat{M_y}) - M_y||_1 \tag{7.11}$$

$$L_{midtpert} = ||(\hat{M_y})^p - (M_y)^p||_1 \tag{7.12}$$

$$L_{mmspc} = ||(\hat{M_x})^p - M_x^p||_1 \tag{7.13}$$

7.1.5 L_{foreG}, L_{foreD} の各項の詳細

L_{foreG} と L_{foreD} の各項の詳細は，(7.14)，(7.15)，(7.16)，(7.17) および (7.18) に示されている．$\hat{x}_f$ は変換された画像の前景にで，$\times$ は要素ごとの乗算を表す．

$$L_{fadv} = \begin{cases} \frac{1}{2}||D(\hat{x}_f) - 1||_2 & \text{for G} \\ \frac{1}{2}||D(y \otimes M_y) - 1||_2 + \frac{1}{2}||D(\hat{x}_f)||_2 & \text{for D} \end{cases} \tag{7.14}$$

$$L_{fadvpert} = \begin{cases} \frac{1}{2}||D((\hat{x}_f)^p) - 1||_2 & \text{for G} \\ \frac{1}{2}||(y \otimes M_y)^p) - 1||_2 + \frac{1}{2}||D((\hat{x}_f)^p)||_2 & \text{for D} \end{cases} \tag{7.15}$$

$$L_{fidt} = ||\hat{y}_f - y \otimes M_y||_1 \tag{7.16}$$

$$L_{fidtpert} = ||(\hat{y}_f)^p - (y \otimes M_y)^p||_1 \tag{7.17}$$

$$L_{fmspc} = ||(\hat{x}_f)^p - \hat{x^p}_f||_1 \tag{7.18}$$

7.1.6 L_{imgG}, L_{imgD} の各項の詳細

L_{imgG} と L_{imgD} の各項の詳細は，(7.19)，(7.20)，(7.21)，(7.22) および (7.23) に示されている．$\hat{x}$ は Generator によって変換された画像である．L_{irec} は，M_x および $(\hat{M_x})$ のいずれにも関連しない領域，すなわち変換と無関係な背景に対する再構成損失である．

$$L_{iadv} = \begin{cases} \frac{1}{2}||D(\hat{x}) - 1||_2 & \text{for G} \\ \frac{1}{2}||D(y) - 1||_2 + \frac{1}{2}||D(\hat{x})||_2 & \text{for D} \end{cases} \tag{7.19}$$

$$L_{iadvpert} = \begin{cases} \frac{1}{2}||D((\hat{x})^p) - 1||_2 & \text{for G} \\ \frac{1}{2}||y^p - 1||_2 + \frac{1}{2}||D((\hat{x})^p)||_2 & \text{for D} \end{cases} \tag{7.20}$$

$$L_{iidt} = ||\hat{y} - y||_1 \tag{7.21}$$

$$L_{iidtpert} = ||(\hat{y})^p - y^p||_1 \tag{7.22}$$

$$L_{imspc} = ||(\hat{x})^p - \hat{x^p}||_1 \tag{7.23}$$

$$L_{irec} = ||\hat{x} \otimes \overline{M_x \cup (\hat{M_x})} - x \otimes \overline{M_x \cup (\hat{M_x})}||_1 \tag{7.24}$$

7.2 提案手法の特長

提案手法では，事前訓練済みの Salient Object Detection ネットワークにより抽出した，サリエンシーマップにおける変換を学習することで，変換するべき前景と変換するべきではない背景を分ける．また，変換したサリエンシーマップ，変換した前景，変換した背景それぞれに対して損失を計算して学習を行う．これらにより，ネットワークの変換能力を前景の部分に集中することが可能となり，変換性能を高い水準で維持しつつ効率的な変換ができる．さらに，多くの既存手法が悩まされている，変換後に背景が崩壊してしまうといった問題を解決できる．

7.3 実験

7.3.1 セットアップ

データセット

提案手法の性能を評価するため，selfie2anime [88]，front2profile [44]，horse2zebra [42]，apple2orange [42] の 4 つのデータセットに対して非ペア画像変換の性能を評価した．selfie2anime は，女性の自撮りの画像からアニメドメインの女性の画像へ変換するタスクのデータセットであり，非ペア画像変換で広く用いられている．訓練データは 3400 枚の対になっていない画像ペア，検証データは 100 枚の非整列画像ペアで構成されており，画像解像度は 256×256 である．front2profile は [44] で導入された，正面顔から側面顔に変換するタスクのデータセットである．訓練データは正面顔 4320 枚，横顔 3680 枚の非整列画像ペア，検証データは 1960 枚の整列画像ペアで構成されており，画像解像度は 128×128 である．front2profile データセットは撮影条件や表情を変えてはいるが，1 人の顔から 40 枚ほどの画像を作成しているため，モード崩壊が起こりやすく，学習が安定しづらい．horse2zebra は画像中のウマをシマウマに変換するタスクの

データセットであり，非ペア画像変換で広く用いられている．訓練データはウマ 1067 枚，シマウマ 1334 枚の非整列画像ペア，検証データはウマ 120 枚，シマウマの画像 140 枚の非整列画像ペアで構成されており，画像解像度は 256×256 である．apple2orange は画像中のりんごをオレンジに変換するタスクのデータセットであり，非ペア画像変換で広く用いられている．訓練データはりんご 995 枚，オレンジ 1019 枚の非整列画像ペア，検証データはりんご 266 枚，オレンジ 248 枚の非整列画像ペアで構成されており，画像解像度は 256×256 である．

非ペア画像変換の課題である，形状の変化，背景の崩壊といった独立した問題に対して，selfie2anime と front2profile は前者，horse2zebra や apple2orange は後者に対応している．horse2zebra や apple2orange は，変換の前後でメインオブジェクトの形状が変化しないが，メインオブジェクトの形状が変化しない場合でないと背景の崩壊に対する定量的な指標を測れないため，これらのデータセットでの評価を行っている．

訓練条件の詳細

非ペア画像変換は，データセットによって変換の度合いや難易度が大きく異なるため，全てのデータセットに対して一律の学習条件を適用することは不適切である．そのため，本節では，提案手法において各データセットに適用される個別の学習条件を紹介する．selfie2anime データセットでは，総学習ステップ数は 700K とした．front2profile データセットでは，総学習ステップ数は 220K とした．horse2zebra データセットでは，総学習ステップ数は 220K とした．apple2orange データセットでは，総学習ステップ数は 30K とした．これは，提案手法が先に学習済みの Salient Object Detection ネットワークによって前景を抽出するため，学習の収束が早くなったことが起因している．

学習条件

非ペア画像変換のデータセットは，その種類によってやるべきタスクや難しさが大幅に変わるため，画一的な訓練条件を用いることは適切ではない．

提案手法では，$\beta_1, \beta_2 = 0.5, 0.999$ の Adam [61] を用いて，バッチサイズ 1 で訓練を行った．学習率は，学習の初期 10% では，5×10^{-5} から 2×10^{-4} までの線形増加，10% から 50% までは 2×10^{-4}，50% 以降は Cosine Annealing [78] によって，0 まで減少させている．

ハイパーパラメータ

ネットワークのハイパーパラメータ: ネットワークのチャンネル数は，上のスケールから $32, 64, 128, 256$ となっている．

損失関数のハイパーパラメータ: selfie2anime や front2profile のような形状が大きく変化する場合は $\lambda_{maskG} = 0.5, \lambda_{foreG} = 1.0, \lambda_{imgG} = 0.5$ とした．horse2zebra や apple2orange のような形状が変化しないが背景が画像に占める割合が大きい場合は，$\lambda_{maskG} = 0.0, \lambda_{foreG} = 1.0, \lambda_{imgG} = 0.0$ とした．形状が変化しない場合はマスクの変換を行う必要がないため $\lambda_{maskG} = 0.0$ とし，背景は学習に含める必要がないため $\lambda_{imgG} = 0.0$ としている．

7.3.2 評価手法

評価指標には，Frèchet Inception Distance (FID) [86]，Kernel Inception Distance (KID) [89] を用いた．これらの指標はペア画像が用意されていない場合の，非ペア画像変換においてよく採用されている評価指標であり，ターゲットドメインの画像群と変換画像群を Inception V3 [69] で処理した埋め込み表現の分布距離である．FID や KID は，非ペア画像変換でよく用いられている指標値であるが，ターゲットドメインの画像群と変換画像群の分布がどれだけ近しいかを表しているので，形状や空間配置，背景の自然さが評価値に直接的には影響しない．

さらに，horse2zebra，apple2orange データセットのような背景が画像に占める割合が大きいデータセットにおいて，背景がどれだけ崩壊しているかを測る定量的に測定するために，新たな指標である Background Reconstruction PSNR (BR PSNR) 考案した．BR PSNR は入力画像と変換画像において，背景がどれだけ保存されているかを表す指標であり，入力画像と変換画像の背景部分の再構成 PSNR を計算する．BR PSNR は背景が保存されているほど高い値となり，背景が崩壊し望まない変換が起こっているほど低い値となる．BR PSNR を求める際の前景マスクは，手作業でラベリングした．BR PSNR が高いほど背景部分が保存されていることを表しているが，指標値が優れていれば無条件に優れた変換ができているということではない．例えば，入力画像を全く変換しなかった場合，BR PSNR は無限大になる．BR PSNR の指標値は，FID や KID の評価値と合わせて見るべきであることを注意されたい．

7.3.3 定量的比較

表 7.1 は，提案手法と既存手法 (CycleGAN [42], FastCUT [43], CUT [43], AttentionGAN [7], MSPC[*1] [44]) の各データセットにおける評価指標の値を記している．提案手法は，全てのデータセットにおける全ての指標において，最も優れた値か，2 番目に優れた値を達成している．FID や KID は形状や空間配置に特化した指標ではないため，提案手法が形状の変換や空間配置に効果的に対処できているかは表 7.1 では分からないが，FID や KID という観点からも提案手法は優れた指標値を達成している．提案手法は horse2zebra や apple2orange のような背景が画像に占める割合が大きいデータセットにおいても優れた指標値を達成している．一般的に視覚品質と歪みにはトレードオフが存在し，例えば horse2zebra データセットにおいて CUT は視覚品質に関連する指標 (FID，KID) は優れているが，歪みに関する指標値は極めて悪い値となっている．それに対して提案手法は視覚品質に関連する指標 (FID，KID)，歪みに関する指標 (BR PSNR) ともに優れた値を達成しており，視覚品質と歪みにはトレードオフの観点からも極めて優れた性能を達成している．apple2orange においては，AttentionGAN が極めて高い BR PSNR を達成しているが，これは AttentionGAN が前景の抽出に失敗し，入力画像をほとんど変化させない出力をしているからである．実際に，提案手法と AttentionGAN において，抽出した前景マスクと正解マスクの MIoU のヒストグラムを比較した図 7.6 を見るとわかるように，AttentionGAN は全く前景を認識していないサンプルが非常に多い．そういったサンプルでは入力画像を全く変化させずに出力するため背景が保存され，BR PSNR が高い値となる．しかし先述したように，BR PSNR が高いほど優れた手法であるとは言えず，FID や KID の評価値も合わせて考慮すると AttentionGAN より提案手法のほうが優れているといえる．

ネットワークの軽量さの比較に関して，表 7.2 は提案手法と既存手法の変換ネットワークの計算量とパラメータ数の比較である．表 7.2 から，提案手法で採用されている Generator は計算量が小さく，サリエンシーマップを抽出する U2Net+ の計算量を含めても計算量やパラメータ数が小さい．これは，前景と背景の分離のためにネットワークを追加した結果，計算量が増大した AttentionGAN とも対照的である．

[*1] MSPC のソースコードは公開されているが，実験条件が論文に記載されていたものと大きく異なるため，論文 [44] で記されていた設定をもとに再現実験を行った．表 7.1 や図 7.3 の MSPC の結果は再現実験の結果であるため文献値と異なる．

表 7.1　各データセットにおける定量的な比較．最も良い値はオレンジに色づけており，2 番目に良い値には青に色づけている．KID は (平均) ± (分散) の様に表記をしている．

	selfie2anime		horse2zebra		
	FID (↓)	KID (↓)	FID (↓)	KID (↓)	BR PSNR (↑)
CycleGAN [42]	84.30	2.08 ± 0.26	77.02	1.97 ± 0.55	19.26
FastCUT [43]	285.23	24.71 ± 0.64	73.81	2.11 ± 0.60	17.26
CUT [43]	80.34	1.51 ± 0.29	47.19	0.75 ± 0.32	15.61
Attention GAN [7]	86.66	1.98 ± 0.22	73.36	1.89 ± 0.35	30.68
MSPC [44]	112.14	5.96 ± 0.53	76.16	2.35 ± 0.57	16.38
Ours	81.21	1.79 ± 0.31	53.43	0.69 ± 0.23	30.81
Input	289.44	25.37 ± 0.80	235.95	19.95 ± 0.74	Inf
Target	0	-3.91 ± 0.13	0	-1.74 ± 0.17	NaN
	front2profile		apple2orange		
	FID (↓)	KID (↓)	FID (↓)	KID (↓)	BR PSNR (↑)
CycleGAN [42]	96.14	16.96 ± 0.27	174.08	10.44 ± 0.87	7.24
FastCUT [43]	120.32	23.03 ± 0.41	156.77	9.34 ± 0.94	24.67
CUT [43]	105.78	19.38 ± 0.31	177.83	10.68 ± 0.95	20.40
Attention GAN [7]	87.07	14.65 ± 0.29	168.71	11.01 ± 0.81	41.50
MSPC [44]	94.13	18.45 ± 0.45	205.84	14.97 ± 1.01	21.32
Ours	45.95	10.22 ± 0.31	156.76	9.63 ± 0.89	29.44
Input	108.78	21.79 ± 0.59	183.05	13.79 ± 0.76	Inf
Target	0	-1.03 ± 0.02	0	-5.10 ± 0.21	NaN

7.3.4　定性的比較

図 7.3 は，提案手法と既存手法 (CycleGAN [42], FastCUT [43], CUT [43], AttentionGAN [7], MSPC [44]) の変換画像の比較である．図 7.3 の 2 列目 “Target"では，検証データが整列している場合は正解画像，非ペアの場合はターゲットドメインの 1 サンプルを表示している．

selfie2anime について，提案手法の変換画像は既存手法と比較して明らかに顔の形状や空間配置が改善されていることがわかる．これは，提案手法が空間的な形状を明示的にガイドすることで顔の形状がよりアニメドメインに近づいたものになり，その結果顔

表 7.2　提案手法と既存手法の変換ネットワークの計算量とパラメータ数の比較.

	GMACs/256 × 256 image	# Params (M)
CycleGAN	56.86	11.38
FastCUT	56.86	11.38
CUT	56.86	11.38
AttentionGAN	71.51	11.82
MSPC	56.86	11.38
Ours	$\underset{\text{Generator}}{8.05} + \underset{\text{U2Net+}}{12.75}$	$\underset{\text{Generator}}{2.65} + \underset{\text{U2Net+}}{1.13}$

のパーツの空間配置も自然な配置となったのだと考えられる．CUT は定性的な評価指標の値は優れているが，実際には形状の崩壊などが度々生じてしまっており，直観的には好ましくない画像を生成してしまっている．front2profile について，提案手法以外の手法はモード崩壊が生じていたりして貧弱な画像を生成している．それに対して提案手法は有効な変換ができている．horse2zebra や apple2orange についても，提案手法は極めて優れた背景保存性を達成できていることが見受けられる．

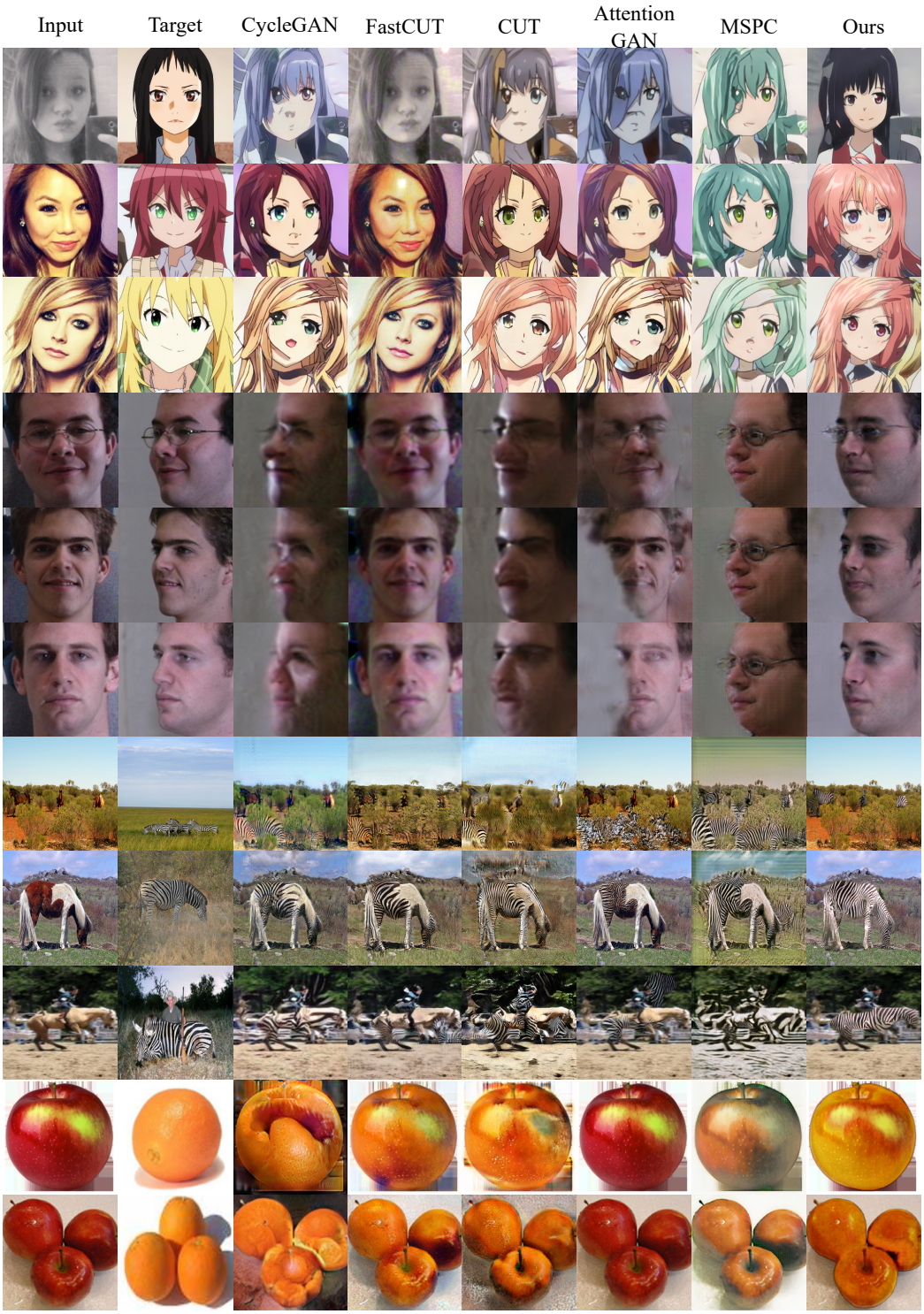


図 7.3　提案手法と既存手法との定性的な比較.

表 7.3　selfie2anime データセットにおいて，サリエンシードメインにおける変換を行わなかった場合と行った場合における評価値の比較.

	FID (↓)	KID (↓)
w/o Saliency Transformation	121.27	4.03 ± 0.64
w Saliency Transformation	81.21	1.79 ± 0.31

表 7.4　selfie2anime において，(7.1) と (7.5) の各損失項が評価値に与える影響の比較.

	L_{mask}	L_{fore}	L_{img}	FID (↓)	KID (↓)
Case A			✓	89.67	2.51 ± 0.35
Case B	✓		✓	87.14	2.21 ± 0.30
Case C		✓	✓	87.92	1.86 ± 0.45
Ours	✓	✓	✓	81.21	1.79 ± 0.31

7.3.5　新規性の有効性の検証

本項で行うアブレーションスタディの定性的な比較は図 7.4 に示される.

サリエンシードメインの変換：提案手法では，形状の変換に対処するために事前学習済みの Salient Object Detection ネットワークで抽出したサリエンシーマップに対する変換を同時に行っているが，これがどのような効果をもたらすかを検証する．表 7.3 は，selfie2anime においてサリエンシードメインの変換を同時に行った場合と行わなかった場合の評価値の比較である．表 7.3 から，サリエンシードメインにおける変換を同時に行った場合に性能が向上することが確認された．これは，変換の際にメインオブジェクトの位置や形状が変化する場合は，画像ドメインだけではなく，メインオブジェクトの形状や位置を表すサリエンシードメインの変換も同時に学習するべきであるということを示している．実際に，図 7.4 の列 “w/o Saliency Transformation"では，変換後のオブジェクトの形状や顔のパーツの空間配置がおかしくなってしまっているが，列 “Ours" ではそのような問題が解消されている.

損失項：提案手法はサリエンシードメイン，変換したサリエンシーマップを元とした画像の前景部分，画像全体の 3 つにおいて敵対的な学習を行っているが，それぞれがどのような影響を与えているかを検証する．表 7.4 は，(7.1)，(7.5) の $L_{mask}, L_{fore}, L_{img}$ をそれぞれを採用した場合とそうでない場合の比較である．表 7.4 は，selfie2anime で

図 7.4　selfie2anime における，アブレーションスタディの定性的な比較.

は $L_{mask}, L_{fore}, L_{img}$ を全て使用した場合，つまり変換したサリエンシーマップを元とした画像の前景部分，画像全体の 3 つにおいて敵対的な学習を行う場合に評価値が最も優れた値になることが確認された．L_{mask}, L_{fore} はメインオブジェクトに特化した損失項であるため，これらの損失を加えることでより適切な形状変換が可能となり，生成画像の品質が向上したと考えられる．実際に，図 7.4 では，形状や空間配置といった点で提案手法の生成画像が優れていることを示している．

Salient Object Detection の性能：図 7.5，図 7.6 はそれぞれ horse2zebra と apple2orange における，抽出したサリエンシーマップの IoU のヒストグラムである．図 7.5，図 7.6 から提案手法は AttentionGAN より優れた IoU を持つサンプルが多く，MIoU の値も高くなっている事がわかる．また，AttentionGAN は前景の抽出の精度が低く，特に apple2orange では前景が全く分離できていない場合もしばしば起こっているが，提案手法では合理的な分離をすることが可能となっている．

図 7.7 は，horse2zebra と apple2orange における，提案手法と AttentionGAN の生成画像と前景マップの比較である．図 7.7 の 2 列目の Mask GT は手作業でウマの部分をラベリングした画像であり，Background Reconstruction PSNR の計算にも用いら

れている．図 7.7 から，提案手法は AttentionGAN よりも正解に近いマップを抽出できており，それが生成画像の品質の向上につながっていることが確認できる．

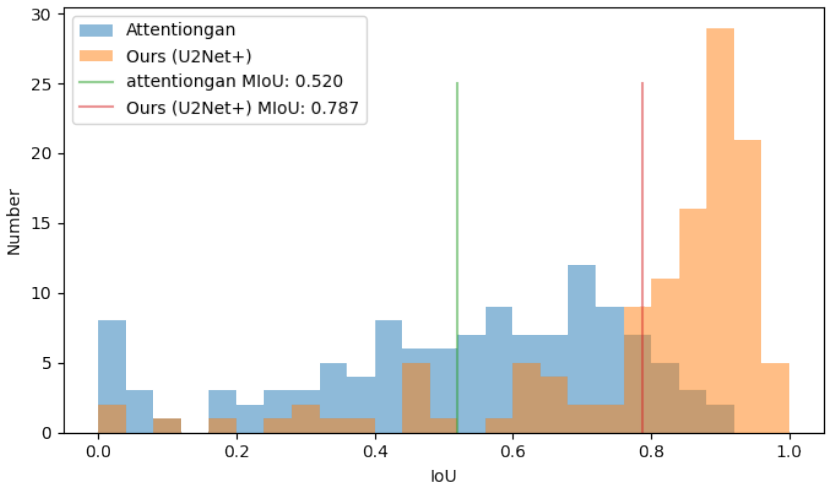


図 7.5　horse2zebra における提案手法と AttentionGAN の前景マスクの IoU ヒストグラム．

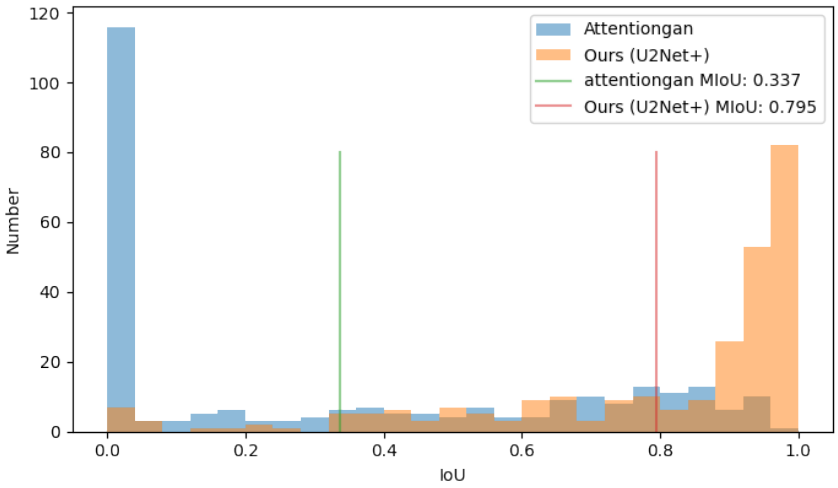


図 7.6　apple2orange における提案手法と AttentionGAN の前景マスクの IoU ヒストグラム．

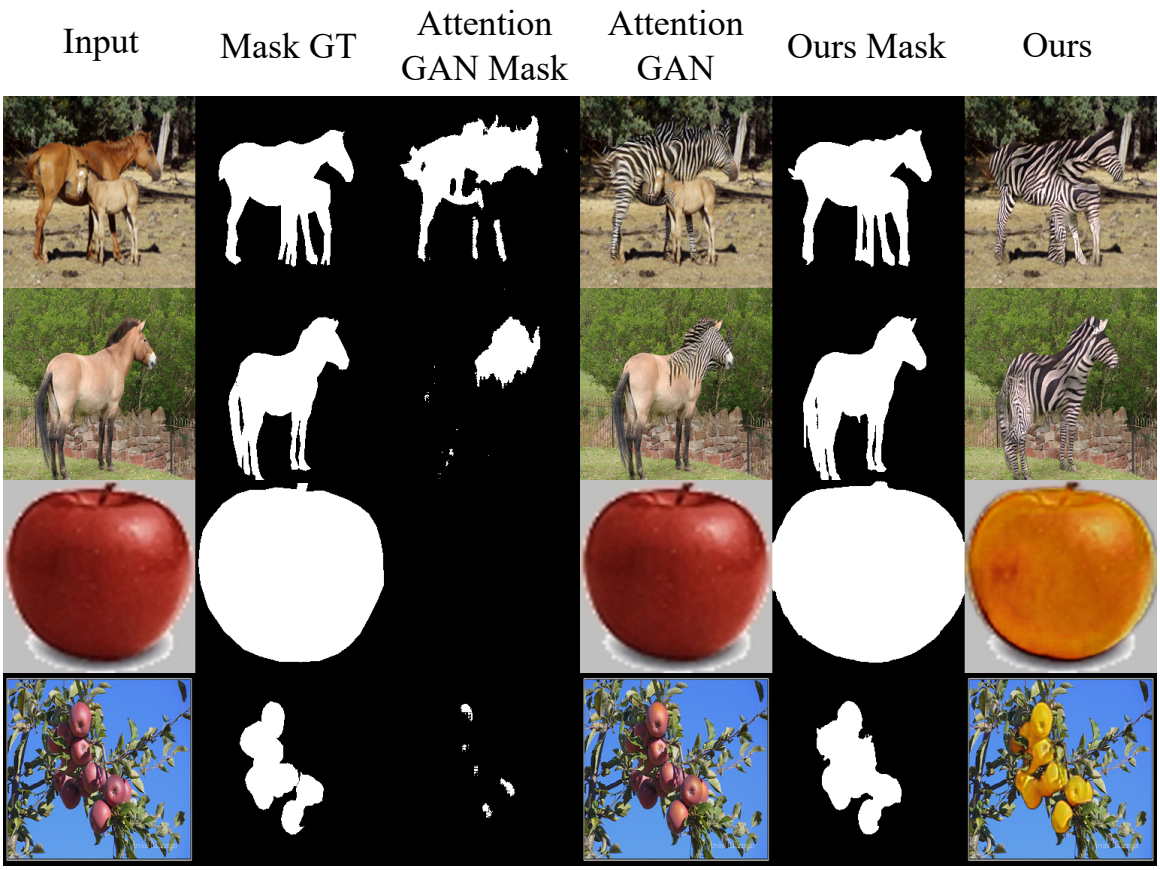


図 7.7　horse2zebra と apple2orange における提案手法と AttentionGAN の前景マスクの比較.

図 7.8　selfie2anime において，提案手法が変換に失敗しているケースの例.

7.3.6　制限及び今後の取り組み

提案手法は，非ペア画像変換における形状の変換や背景の崩壊といった問題に対する優れた解法であるが，残された課題が存在する．提案手法はサリエンシードメインにおける変換を同時に学習しており，サリエンシードメインへの変換のために事前学習済みの Salient Object Detection ネットワークを使用している．事前学習済みのネットワークを使用することで，訓練データのバイアスに影響されにくいといった利点はあるが，Salient Object Detection ネットワークの性能が最終的な出力画像の深刻なボトルネックとなる．実際に，図 7.8 は Salient Object Detection の失敗によって生成画像の品質が低下したサンプルである．非ペアデータセットにおいて，効果的に Salient Object Detection を学習できる手法が確立されたり，より優れた Salient object detection ネットワークを採用することで，この問題は解決されるだろう．ただし，精度と処理速度のトレードオフに関しては配慮する必要がある．

第 8 章

結　論

8.1　提案手法のまとめ

本研究では，深層学習の「性能と軽量さの両立」という課題に対して，「新規モジュールを採用したネットワークを提案する」，「効果的な処理スキームや損失関数を提案する」という 2 つのアプローチで取り組んだ．その中でも，本研究で具体的に取り組むタスクについて，「新規モジュールを採用したネットワークを提案する」ではフォトレタッチと動画のノイズ除去，「効果的な処理スキームや損失関数を提案する」では姿勢変換と非ペア画像変換に取り組んだ．その結果，本研究では幅広い 4 つの画像変換分野において性能と軽量さの両立した手法を提案し，当該分野における実用的な手法の開発に貢献した．以下ではそれぞれの研究タスクにおける結論を記す．

8.1.1　フォトレタッチ

入力画像をラプラシアンピラミッドに分解し，低周波成分を Axial Transformer Block，高周波成分を軽量 CNN で変換する LPTT を提案した．LPTT は Transformer の表現力を維持したまま 4K 以上の解像度の画像を推論することが可能であり，LPTT($L = 6$) は 8K の画像をリアルタイムで処理できる．また，LPTT は先行研究と比較して，生成画像の品質と高解像度画像に対する推論速度の両方を向上させた．実験結果は，低解像度成分の処理に Transformers を用いることで，モデルが長距離依存性を捉えられ，その結果，推論速度を維持したまま性能が向上することを示している．

8.1.2　動画のノイズ除去

本研究では，高い性能と軽量さを両立した動画のノイズ除去のためのネットワークである Pseudo Temporal Fusion Network (PTFN) を提案した．PTFN は従来の軽量なノイズ除去手法に対して遥かに少ない計算量で性能を大幅に向上させている．また，PTFN はメモリ消費の観点からも優れており，24GB RAM の GPU で 1080p 画像の処理が可能である．PTFN には Pseudo Temporal Fusion という新たなモジュールを採用しており，単体では時間的な関係性を捉えられないが，Temporal Shift Module と組み合わせて擬似的に時間的な関係性を捉えられ，性能に貢献している．また，本研究ではモダンな ConvBlock を動画のノイズ除去に応用する際に，より有用な構造の探索を行っている．PTFN は動画のノイズ除去において様々な観点から有用なネットワークであることを実証した．

8.1.3　姿勢変換

本論文では，姿勢変換のためのシンプルなネットワークを提案する．提案手法は，姿勢変換を「大まかな姿勢の変換」と「詳細なテクスチャの生成」に分けるという考えのもとネットワークが設計されており，オプティカルフローや身体位置を表現するセマンティックなマップのような余分なデータやタスクを必要としない．さらに，Encoder-Decoder 構造を採用した Axial Transformer Transformation Block によって，姿勢情報を活用した学習をすることで，学習が安定化し，精度も向上する．また，提案手法は追加のタスクやデータを用いていない軽量なネットワークながら，既存手法と競争的な性能を達成しており，性能と軽量さを高い水準で両立している．

8.1.4　非ペア画像変換

本論文では，非ペア画像変換における，形状の変化や背景の崩壊といった問題に対して有効な解決策となる手法を提案した．提案手法は画像だけではなく，事前学習済みの Salient Object Detection ネットワークによって抽出されたサリエンシードメインにおいても敵対的な学習による変換を行う．サリエンシーマップは画像中のメインオブジェクトの形状や位置を表すので，サリエンシーマップを同時に変換することによって形状の変化や背景の崩壊といった問題に対処することが可能である．さらに，提案手法では変換するべき前景に変換リソースを集中できるため，軽量なネットワークながら既存手

法より高い変換性能を達成できた．提案手法は比較検証の結果，定量的・定性的に優れた性能を達成しており，非ペア画像変換という分野における確かな進歩となった．

8.2 まとめ・今後の展望

本研究では深層学習を用いた，フォトレタッチ，動画のノイズ除去，姿勢変換，非ペア画像変換といった，異なる動画像処理タスクに対して，革新的な手法と実践的な成果を提案した．具体的には，性能および軽量さの観点から，従来手法を上回る効率的かつ実用的な解決策を提供した．本研究は，多様な動画像処理分野のタスクにおける重要な技術的進展を示しており，大きな学術的な貢献をしている．さらに，本論文で提案した手法は，軽量かつ高性能なモデル設計に特徴があり，限られたリソース環境下でも高いパフォーマンスを発揮する．よって，商業アプリケーションやリアルタイム処理システムへの適用可能性が拡大され，実用的な応用においても大きな貢献を果たした．

本研究の今後の展望としてはまず，変換性能と軽量さのトレードオフをさらに洗練させる研究が挙げられる．コンピュータビジョン分野では，深層学習の進化が加速しており，特にモデルの軽量化と性能向上の両立に向けた新たなアプローチの開発が今後の重要な課題となる．例えば，Transformer ベースの手法においては，Restormer [14] のように，Attention の計算をチャンネル方向に最適化する新しいアプローチや，CNN と Transformer を組み合わせた CMT [90] のようなハイブリッド手法が次々と提案されており，これらはさらなる性能向上と計算効率の改善をもたらす可能性がある．本研究で提案した手法も，これら最新技術と組み合わせることで，さらに効果的なモデル設計が可能となると考えられる．

さらに，研究を他の画像処理タスクに拡張し，各タスクにおける入力と出力の構造的変化の度合いと，それに応じた最適なアプローチの選定を比較・検証することも有意義である．例えば，Latent Space Models [91, 92] のような，CNN や Transformer とは異なる特徴抽出メカニズムや，Diffusion Models [93] のように従来の GAN や教師あり学習に依存しない推論スキームが近年注目されており，これら新技術が性能と軽量化のトレードオフにどう寄与するかは，今後の研究において興味深いテーマとなるだろう．特に，これらの新手法が既存のタスクやアプローチにどのように統合されるかは，さらなる理論的発展および応用展開に寄与すると期待される．

総括すると，本研究では性能と軽量さというトレードオフを克服することをテーマとした深層学習手法の設計において重要な知見を得られた．本研究で取り組んだ各タスクはそれぞれ実務的な応用可能性が高いものであり，これらのタスクにおける本研究の成

果を用いて，産業応用やエッジデバイスでの活用可能性までを見据えた応用を考えられる．実践的な応用につなげやすいといった点においても，本研究の性能と軽量さのトレードオフを様々な動画像処理タスクにおいて考案するというものの価値は大きいといえる．今後，最新技術との融合や，より多様なタスクへの応用を通じて，本研究の成果がさらに広範な分野で活用され，深層学習モデルの実用性を向上させるための一助となることを期待している．本研究が，動画像変換分野における効率的かつ高性能なモデル開発の分野を前進させたことを確信し，ここに結論とする．

参考文献

[1] J. Liang, H. Zeng, and L. Zhang, “High-resolution photorealistic image translation in real-time: A laplacian pyramid translation network,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.9392–9400, 2021.

[2] J. Lin, C. Gan, and S. Han, “Tsm: Temporal shift module for efficient video understanding,” Proceedings of the IEEE international conference on computer vision, pp.7083–7093, 2019.

[3] Y. Ren, X. Yu, J. Chen, T.H. Li, and G. Li, “Deep image spatial transformation for person image generation,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.7690–7699, 2020.

[4] A. Krizhevsky, I. Sutskever, and G.E. Hinton, “Imagenet classification with deep convolutional neural networks,” Communications of the ACM, vol.60, no.6, pp.84–90, 2017.

[5] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” International Conference on Medical image computing and computer-assisted intervention, pp.234–241, 2015.

[6] M. Tassano, J. Delon, and T. Veit, “Fastdvdnet: Towards real-time deep video denoising without flow estimation,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1354–1363, 2020.

[7] H. Tang, H. Liu, D. Xu, P.H. Torr, and N. Sebe, “Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks,” IEEE Transactions on Neural Networks and Learning Systems, 2021.

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A.A. Efros, “Image-to-image translation with conditional adversarial networks,” Proceedings of the IEEE Conference

on Computer Vision and Pattern Recognition, pp.1125–1134, 2017.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," ICLR, 2021.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, pp.5998–6008, 2017.

[11] J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans, "Axial attention in multidimensional transformers," arXiv preprint arXiv:1912.12180, 2019.

[12] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," Proceedings of the IEEE international conference on computer vision, pp.10012–10022, 2021.

[13] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," Proceedings of the IEEE international conference on computer vision, pp.1833–1844, 2021.

[14] S.W. Zamir, A. Arora, S. Khan, M. Hayat, F.S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.5728–5739, 2022.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[16] J. Liang, J. Cao, Y. Fan, K. Zhang, R. Ranjan, Y. Li, R. Timofte, and L. Van Gool, "Vrt: A video restoration transformer," arXiv preprint arXiv:2201.12288, 2022.

[17] J. Liang, Y. Fan, X. Xiang, R. Ranjan, E. Ilg, S. Green, J. Cao, K. Zhang, R. Timofte, and L. Van Gool, "Recurrent video restoration transformer with guided deformable attention," arXiv preprint arXiv:2206.02146, 2022.

[18] C. Qi, J. Chen, X. Yang, and Q. Chen, "Real-time streaming video denoising with bidirectional buffers," Proceedings of the 30th ACM International Conference on Multimedia, pp.2758–2766, 2022.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen,

Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," Advances in Neural Information Processing Systems 32, pp.8024–8035, Curran Associates, Inc., 2019.

[20] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré, "Flashattention: Fast and memory-efficient exact attention with io-awareness," Advances in Neural Information Processing Systems, vol.35, pp.16344–16359, 2022.

[21] Y. Hu, H. He, C. Xu, B. Wang, and S. Lin, "Exposure: A white-box photo post-processing framework," ACM Transactions on Graphics (TOG), vol.37, no.2, pp.1–17, 2018.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," Advances in neural information processing systems, vol.27, 2014.

[23] P.J. Burt and E.H. Adelson, "The laplacian pyramid as a compact image code," Readings in computer vision, pp.671–679, Elsevier, 1987.

[24] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms," IEEE Transactions on Image Processing, vol.21, no.9, pp.3952–3966, 2012.

[25] L. Jovanov, A. Pizurica, S. Schulte, P. Schelkens, A. Munteanu, E. Kerre, and W. Philips, "Combined wavelet-domain and motion-compensated video denoising based on video codec motion estimation methods," IEEE Transactions on Circuits and Systems for Video Technology, vol.19, no.3, pp.417–421, 2009.

[26] P. Arias and J.-M. Morel, "Video denoising via empirical bayesian estimation of space-time patches," Journal of Mathematical Imaging and Vision, vol.60, no.1, pp.70–93, 2018.

[27] M. Tassano, J. Delon, and T. Veit, "Dvdnet: A fast network for deep video denoising," IEEE international conference on image processing, pp.1805–1809, 2019.

[28] G. Vaksman, M. Elad, and P. Milanfar, "Patch craft: Video denoising by deep modeling and patch matching," Proceedings of the IEEE international conference on computer vision, pp.2157–2166, 2021.

[29] A. Buades and J.-L. Lisani, "Enhancement of noisy and compressed videos by optical flow and non-local denoising," IEEE Transactions on Circuits and Systems for Video Technology, vol.30, no.7, pp.1960–1974, 2019.

[30] W. Shen, M. Cheng, G. Lu, G. Zhai, L. Chen, M.S. Asif, and Z. Gao, "Spatial temporal video enhancement using alternating exposures," IEEE Transactions on Circuits and Systems for Video Technology, vol.32, no.8, pp.4912–4926, 2021.

[31] L. Xiang, J. Zhou, J. Liu, Z. Wang, H. Huang, J. Hu, J. Han, Y. Guo, and G. Ding, "Remonet: Recurrent multi-output network for efficient video denoising," Proceedings of the AAAI Conference on Artificial Intelligence, vol.36, pp.2786–2794, 06 2022.

[32] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," Advances in neural information processing systems, vol.30, 2017.

[33] P. Esser, E. Sutter, and B. Ommer, "A variational u-net for conditional appearance and shape generation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.8857–8866, 2018.

[34] Z. Zhu, T. Huang, B. Shi, M. Yu, B. Wang, and X. Bai, "Progressive pose attention transfer for person image generation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.2347–2356, 2019.

[35] H. Tang, S. Bai, L. Zhang, P.H. Torr, and N. Sebe, "Xinggan for person image generation," Proceedings of the European conference on computer vision, pp.717–734, 2020.

[36] Y. Li, C. Huang, and C.C. Loy, "Dense intrinsic appearance flow for human pose transfer," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.3693–3702, 2019.

[37] T. Ma, B. Peng, W. Wang, and J. Dong, "Must-gan: Multi-level statistics transfer for self-driven person image generation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.13622–13631, 2021.

[38] Z. Lv, X. Li, X. Li, F. Li, T. Lin, D. He, and W. Zuo, "Learning semantic person image generation by region-adaptive normalization," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.10806–10815, 2021.

[39] J. Zhang, K. Li, Y.-K. Lai, and J. Yang, "Pise: Person image synthesis and editing with decoupled gan," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.7982–7990, 2021.

[40] P. Zhang, L. Yang, J.-H. Lai, and X. Xie, "Exploring dual-task correlation for pose guided person image generation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.7713–7722, June 2022.

[41] G. Koch, R. Zemel, R. Salakhutdinov, et al., "Siamese neural networks for one-shot image recognition," ICML deep learning workshop, vol.2, pp.1–30, 2015.

[42] J.-Y. Zhu, T. Park, P. Isola, and A.A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," Proceedings of the IEEE international conference on computer vision, pp.2223–2232, 2017.

[43] T. Park, A.A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," Proceedings of the European conference on computer vision, pp.319–345, 2020.

[44] Y. Xu, S. Xie, W. Wu, K. Zhang, M. Gong, and K. Batmanghelich, "Maximum spatial perturbation consistency for unpaired image-to-image translation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.18311–18320, 2022.

[45] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," Advances in neural information processing systems, vol.30, pp.700–708, 2017.

[46] H. Fu, M. Gong, C. Wang, K. Batmanghelich, K. Zhang, and D. Tao, "Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.2427–2436, 2019.

[47] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," Proceedings of the European conference on computer vision, pp.172–189, 2018.

[48] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, "Learning photographic global tonal adjustment with a database of input/output image pairs," CVPR 2011, pp.97–104, 2011.

[49] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet

for the 2020s," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.11976–11986, 2022.

[50] L. Chen, X. Chu, X. Zhang, and J. Sun, "Simple baselines for image restoration," Proceedings of the European conference on computer vision, pp.17–33, 2022.

[51] K. O'shea and R. Nash, "An introduction to convolutional neural networks," arXiv preprint arXiv:1511.08458, 2015.

[52] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1251–1258, 2017.

[53] S. Narayan, "The generalized sigmoid activation function: Competitive supervised learning," Information sciences, vol.99, no.1-2, pp.69–82, 1997.

[54] A.F. Agarap, "Deep learning using rectified linear units (relu)," arXiv preprint arXiv:1803.08375, 2018.

[55] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," arXiv preprint arXiv:1505.00853, 2015.

[56] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," arXiv preprint arXiv:1606.08415, 2016.

[57] S. Woo, J. Park, J.-Y. Lee, and I.S. Kweon, "Cbam: Convolutional block attention module," Proceedings of the European conference on computer vision, pp.3–19, 2018.

[58] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," International conference on machine learning, pp.448–456, 2015.

[59] J.L. Ba, J.R. Kiros, and G.E. Hinton, "Layer normalization," arXiv preprint arXiv:1607.06450, 2016.

[60] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," Proceedings of the European conference on computer vision, pp.694–711, 2016.

[61] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[62] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," Communica-

tions of the ACM, vol.63, no.11, pp.139–144, 2020.

[63] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv preprint arXiv:1511.06434, 2015.

[64] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[65] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., "Language models are few-shot learners," Advances in neural information processing systems, vol.33, pp.1877–1901, 2020.

[66] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," International conference on machine learning, pp.4055–4064, 2018.

[67] X. Chu, Z. Tian, B. Zhang, X. Wang, X. Wei, H. Xia, and C. Shen, "Conditional positional encodings for vision transformers," arXiv preprint arXiv:2102.10882, 2021.

[68] R. Zhang, P. Isola, A.A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.586–595, 2018.

[69] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.2818–2826, 2016.

[70] X. Mao, Q. Li, H. Xie, R.Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," Proceedings of the IEEE international conference on computer vision, pp.2794–2802, 2017.

[71] W. Shi, J. Caballero, F. Huszár, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1874–1883, 2016.

[72] X. Xia and B. Kulis, "W-net: A deep model for fully unsupervised image segmentation," arXiv preprint arXiv:1711.08506, 2017.

[73] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The miss-

ing ingredient for fast stylization," arXiv preprint arXiv:1607.08022, 2016.

[74] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," arXiv preprint arXiv:1704.00028, 2017.

[75] Y.-S. Chen, Y.-C. Wang, M.-H. Kao, and Y.-Y. Chuang, "Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.6306–6314, 2018.

[76] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation," arXiv:1704.00675, 2017.

[77] Xiph.Org Foundation, "Xiph.org : Derf's test media collection," 2022. https://media.xiph.org/video/derf/.

[78] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," arXiv preprint arXiv:1608.03983, 2016.

[79] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," Proceedings of the IEEE international conference on computer vision, pp.1501–1510, 2017.

[80] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," International conference on machine learning, pp.214–223, 2017.

[81] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1096–1104, 2016.

[82] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," Proceedings of the IEEE international conference on computer vision, pp.1116–1124, 2015.

[83] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.7291–7299, 2017.

[84] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Transactions on Image Processing, vol.13, no.4, pp.600–612, 2004.

[85] S. Barratt and R. Sharma, "A note on the inception score," arXiv preprint

arXiv:1801.01973, 2018.

[86] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," Advances in neural information processing systems, vol.30, 2017.

[87] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O.R. Zaiane, and M. Jagersand, "U2-net: Going deeper with nested u-structure for salient object detection," Pattern recognition, vol.106, p.107404, 2020.

[88] J. Kim, M. Kim, H. Kang, and K.H. Lee, "U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation," International Conference on Learning Representations, pp.0–0, 2019.

[89] M. Bińkowski, D.J. Sutherland, M. Arbel, and A. Gretton, "Demystifying mmd gans," arXiv preprint arXiv:1801.01401, 2018.

[90] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, "Cmt: Convolutional neural networks meet vision transformers," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.12175–12185, 2022.

[91] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, and Y. Liu, "Vmamba: Visual state space model," arXiv preprint arXiv:2401.10166, 2024.

[92] S. Yamashita and M. Ikehara, "Image deraining with frequency-enhanced state space model," arXiv preprint arXiv:2405.16470, 2024.

[93] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," Advances in neural information processing systems, vol.33, pp.6840–6851, 2020.

謝辞

本研究は筆者が慶應義塾大学大学院理工学研究科後期博士課程在学中に行ったものである．本論文の作成にあたり，多くのご意見，ご享受を賜りました指導教員および，本論文の主査である慶應義塾大学理工学部の池原雅章教授に深く感謝の意を示します．また，ご多忙な中本論文の副査を引き受けてくださった萩原将文教授，青木義満教授，久保亮吾教授に厚く御礼申し上げます．

また，本研究を進めるうえで数々の助言をくださり，様々なご指導をいただいた池原研究室の諸氏にも深く御礼申し上げます．最後にはなりますが，私がこのような研究を行えたのは家族の支えがあってこそでした．常日頃から私を支えてくださった家族の皆様に心から感謝申し上げます．