撮影環境起因で劣化した画像の特徴を考慮した深層学習による画像復元

2024 年度

松井 拓郎

学位論文 博士 (工学)

撮影環境起因で劣化した画像の特徴を考慮した深層学習による画像復元

2024 年度

慶應義塾大学大学院理工学研究科

松井 拓郎

目次

図目次

表目次

第 1 章

序　論

1.1　研究背景

現代のデジタル画像は，個人や法人を問わず様々な場面で広く使用されており，情報の可視化やコミュニケーションにおいて重要な役割を果たしている．個人では，スマートフォンの普及により写真を撮影することが当たり前となった．また，Instagram や TikTok を始めとする SNS(Social Network Service) の流行により，動画像を共有する機会が増えており，より美しい景色や人物が精細に写っている写真が求められている．法人では画像から何らかの情報を抽出し業務やサービスの効率化や高度化に利用されることが多い．例えば，セキュリティカメラや自動車の車載カメラでの異常検知や物体検出や，人間が作業しづらい環境でのドローンカメラによる点検作業の代替などデジタル画像の応用範囲は広い．

しかし，撮影された画像はさまざまな要因によって劣化が生じてしまう．画像劣化の要因は「撮影時の環境条件によるもの」と「情報処理時に起きるもの」に大別できる．要因別に劣化画像の種類を整理したものを図**??**に示す．

一つ目の撮影時の環境条件とは，照度，天候，撮影場所，被写体・撮影者がある．低照度環境下では，画像にノイズが発生し，暗い箇所の情報が欠落してしまう．一方で高照度環境下では白飛びにより情報が欠落してしまう．天候も画像を劣化させる要因であり，雨や霧が画像にぼやけやコントラストの低下を生じさせてしまい色温度や色彩バランスが変化してしまう．撮影場所による画像劣化として，フェンスのような障害物や観光地での群衆など，注目したい被写体以外の物体が映り込んでしまうことが挙げられる．被写体・撮影者による画像劣化としては，被写体自体の動きや撮影者の技能によってぶれやぼけが発生してしまうなどが挙げられる．

図 1.1　要因別画像劣化の種類と本研究の範囲

二つ目の情報処理時に生じる画像劣化として，圧縮や伝送，ストレージ時の処理による情報の損失や，ノイズやアーチファクトの発生がある．例えば，画像の圧縮では情報の削減が行われ，ブロックノイズや輪郭のぼやけが生じることがある．また，画像の伝送中にはノイズが混入する可能性があり，画像の復元において問題となることがある．

このような背景のもと，画像劣化の要因を取り除き，画像の視認性を向上させ，画像に含まれる有用な情報を抽出しやすく強調することは，画像処理の最も重要な役割の一つと言える．最終的な目標は，どのような劣化画像であっても自動的に鮮明な画像を得ることである．

複数の画像復元タスクをこなせる手法として，Deep Image Prior [1] がある．この手法は深層学習ベースの手法であり，大量のラベル付きデータを必要とせずにネットワークの学習特性を活かして画像復元を行う．ランダムノイズを入力して生成した画像と劣化画像との間の差異を定義する損失関数が最小となるようにネットワークを学習させる．学習を途中で止めることで劣化のない画像を復元することができ，ノイズ除去，超解像，インペインティングなど様々なタスクに応用ができる．しかし，1 枚の画像ごとに学習する必要があるため，計算時間が長く，人間の手で学習回数やハイパーパラメータを調整しなければならない．全自動で様々な劣化画像を復元する技術は実現されていないのである．よって，現在は劣化画像の特徴に応じてシステムを個別に構築していく研究が一般的である．

本論文では，画像劣化の中でも撮影時の環境条件が要因で劣化した画像の復元につい

て取り扱う．具体的には，雨すじの除去，フェンス除去，低照度画像強調である．いずれもすでに撮影された画像に対してソフトウェア的に後処理することによって画像の復元を行う．また，一般的に高精度とされる深層学習を用いた手法に焦点を当てる．図**??**で示すように本研究対象の三テーマは，ノイズ，ブラー，コントラスト低下，解像度低下，画像欠損という複数の画像劣化が関与するテーマである．よって，本研究は，劣化画像の種類によらず一つのネットワークで自動的に鮮明な画像を得るという最終目標を達成するための要素技術になり得ると考える．

雨すじ除去とは，雨天時に撮影された画像から雨すじを取り除く処理のことであり，大きく分けて二つの方法がある．動画ベースの手法 [2–4] と静止画ベースの手法 [5–8] である．動画ベースの手法は，2004 年に初めて Garg ら [2] によって提案された．動画ベースの手法は隣接フレームの情報を利用するので精度は高くなるが時間情報も考慮するので計算コストは高くなってしまう．一方，静止画ベースの手法は 2012 年頃からは事前情報を基にした手法が提案されるようになった．スパース性に基づく手法 [5, 6] や低ランク表現に基づく手法 [7]，ガウス混合モデルに基づく手法 [8] などが提案されている．2017 年頃からは深層学習をベースにした手法が提案されるようになった．シンプルな CNN(Convolutional Neural Networks) ベースの手法 [9,10] や GAN(Generative Adversarial Networks) ベースの手法 [11,12] などがある．辞書学習ならびに深層学習を用いた手法は精度が高いが，学習データセットや問題設定に精度が依存してしまう．雨すじを残してしまう，過平滑化されてしまう，色彩が変わってしまうという問題点を残している．

フェンス除去とは，画像内にあるフェンスを取り除き，フェンスで隠れていた部分を補間する処理のことであり，大きく分けて三つの方法がある．動画ベースの手法 [13–19] と複数画像ベースの手法 [20, 21] と単一画像ベースの手法 [22–26] である．2008 年に Liu [22] らが初めて自動でフェンス除去をするアルゴリズムを提案した．動画ベースの手法とは，複数のフレーム情報を利用してフェンスを検出し除去するというものである．動きが大きすぎない動画に対しては高精度だが，動画の処理を行うため計算コストが高くなってしまう．複数画像ベースの手法は焦点や方角などを変えた複数の画像を合成してフェンス除去画像を生成する．フェンス除去以外のタスクにも応用できるが，ある特定の条件下で撮影をする必要があるため実用性は低い．単一画像ベースの手法では，規則性をもとにした手法と色ベースの手法がある．Liu ら [22] と Park ら [23] はフェンスの規則性に着目したが，歪んだフェンスや画像内にフェンスの一部しか写っていない場合はうまく検出できない．また，パッチ単位での処理のため計算コストが高い．一方，Farid ら [24] は色ベースの手法を提案したが，ユーザ入力が必要なうえ，フェンスの色

と背景の色が似ていると検出に失敗してしまうという問題点がある．

低照度画像強調とは，低照度条件で撮影された画像を明るく鮮明にするための処理のことであり大きく分けて二つの方法がある．モデルベースの手法 [27–36] と深層学習ベースの手法 [37–51] である．2004 年頃からヒストグラム平坦化ベースの手法 [27–30] が主流であったが，一定以上に暗い画像に対してはノイズの増幅や変色が生じてしまうという問題点があった．そこで，2013 年頃より人間の視覚メカニズムを参考にした Retinex 理論に基づいた手法 [31–33, 35, 36] が提案されるようになった．これらの手法は極端な照明条件や非均一な照明条件に対する効果は限定的で，見た目も不自然なものが多く，計算コストが高いという問題点があった．2017 年頃には深層学習に基づく手法 [37–51] が提案され，現在の主流となっている．画像を複数の成分に分割してそれぞれを推論するためのサブネットワークを構築している手法が多く，複雑な構造故に計算コストが高い．また，不自然な照度や色彩，ディテールの損失，強調時のノイズやヘイズの生成など多くの問題点を残している．

1.2 研究目的

本研究の目的は，劣化画像の特徴を考慮した深層学習に基づく画像復元手法の開発と性能向上である．深層学習の精度を高めるには，劣化画像の特徴を考慮したネットワーク構造，前処理や後処理，データセットの構築が重要となる．

劣化画像の特徴を考慮したネットワーク構造の開発により画像の復元性能を高めることを目指す．モデルの構築には，CNN や GAN などの深層学習アーキテクチャを活用する．先行研究で有用とされているようなアーキテクチャをそのまま使うのではなく，劣化画像の特徴に合わせてカスタマイズを行う．また，深層学習では劣化画像の問題設定と解き方が重要であり，損失関数と損失係数は最適な設定を見つける必要がある．

次に，劣化画像の特徴を考慮した前処理や後処理の導入により画像の復元性能を高めることを目指す．深層学習ベースの従来法の多くは，古典的な画像処理理論を用いずに大量のデータセットを学習させることで精度を高めようするものが多い．深層学習という最先端の手法と空間フィルタリングなどの古典的な画像処理手法を組み合わせることで，お互いに足りない要素を補完する．

劣化画像の特徴を考慮した深層学習モデルを構築するためには，適切な学習データセットが必要である．教師あり学習では，劣化画像と正解画像のペアが必要となる．しかし，本研究で対象とするような撮影時の環境が要因で生じた劣化画像は正解画像を用意するのが困難な場合がある．近年では学習用の画像データセットが公開されているも

のもあるが，撮影者や撮影場所によってデータセットの偏りが生じてしまう．そこで，劣化画像を適切にモデル化した合成画像を作成することで自然画像に対してもロバストなモデルを作成することを目指す．

最後に，開発した深層学習モデルの実用性と性能を評価するために，さまざまな劣化画像に対する復元実験を行う．各タスクで高い精度を達成している従来法と提案モデルの比較実験を行い，提案手法の有用性と性能の高さを示すことを目指す．具体的には，復元品質の客観的な評価指標を用いた性能評価や主観的な画像の視覚的比較などを行う．

上記の研究目的の達成により，劣化画像の復元における深層学習の有用性と可能性を示し，劣化画像の品質向上や情報の回復に貢献することを目指す．

1.3　本論文の構成

第 2 章では，画像劣化の種類について説明し，空間フィルタリング，画質品質評価手法，深層学習の基礎理論について概説する．第 3 章では雨すじ除去，第 4 章ではフェンス除去，第 5 章では低照度画像強調に関して，それぞれ従来法の問題点を克服する手法を提案する．最後に第 6 章で全体を総括して結論を述べる．

第 2 章

劣化画像の復元に関する基礎理論

2.1 画像劣化

画像劣化は，さまざまな要因によって引き起こされる．画像劣化の種類と代表的な復元手法を表 2.1 に示す．復元手法は近年主流となっている深層学習ベースと伝統的な統計・ルールベースの手法を記載している．

表 2.1 画像劣化の種類と代表的な復元手法

画像劣化		代表的な復元手法	
種類	発生原因	深層学習ベース	統計・ルールベース
ノイズ	センサノイズ，伝送ノイズ	DnCNN，Noise2Noise，Deep Image Prior	ウィナーフィルタ，BM3D
ブラー	カメラや被写体の動き，焦点不良	DeepDeblur，DeblurGAN	ブラインドデコンボリューション
コントラストの低下	照明条件，露出不良	DeepRetinex，KinD++	ヒストグラム平坦化，トーンマッピング
解像度低下	画像の縮小，伝送ロス	SRGAN，EnhanceNet	バイリニア補間，バイキュービック補間
画像欠損	伝送エラー，物理的損傷や障害物の存在	Context Encoders，CM-GAN	テクスチャ合成，Poisson Image Editing

2.1.1 ノイズ

ノイズは画像に不要なランダムな信号であり，画像の品質を低下させる．ノイズの主な原因は，撮影時のセンサのノイズや伝送時のノイズがある．ノイズのない画像を $\boldsymbol{x}$，ガウシアンノイズやインパルスノイズなどの加法性ノイズを $\boldsymbol{n}$ とするとノイズあり画像 $\boldsymbol{y}$ は次のように表される．

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n} \tag{2.1}$$

ノイズ $\boldsymbol{n}$ を推定しノイズなし画像 $\boldsymbol{x}$ を復元する古典的手法としてウィナーフィルタや BM3D (Block-Matching and 3D filtering) [52] が挙げられる．BM3D は，ブロックマッチングと 3D フィルタリングを組み合わせることで，画像内の類似ブロックを見つけ出しノイズを除去する．深層学習ベースの手法では，2017 年に Zhang らによって提案された DnCNN (Denoising Convolutional Neural Network) [53] が代表的である．DnCNN は畳み込み層と Batch Normalization と ReLU (Rectified Linear Unit) からなるシンプルなネットワークである．ノイズを含んだ画像を入力して，残差画像としてのノイズを出力することによってノイズ除去性能を上げることに成功した．

2.1.2 ブラー

ブラーは画像がぼやけたり不鮮明になったりすることである．ブラーの主な原因は，カメラや被写体の動きによるブレや被写体との焦点不良である．鮮明な画像を $\boldsymbol{x}$，ブラーカーネルを $\boldsymbol{k}$ とするとブレ画像は次のように表される．

$$\boldsymbol{y} = \boldsymbol{x} * \boldsymbol{k} + \boldsymbol{n} \tag{2.2}$$

ここで，$*$ は畳み込み演算子を表し，$\boldsymbol{n}$ は加法性ノイズである．ブラインドデコンボリューションとは，ブラーカーネル $\boldsymbol{k}$ を推定し逆畳み込みをすることによって鮮明な画像を復元するという手法である．深層学習ベースの手法では，Wang らが 2017 年に DeepDeblur [54] を提案した．カーネルを推定することなく，ブレ画像から直接鮮明画像を復元することに成功した．

2.1.3 コントラストの低下

コントラストの低下とは，画像の明るさや濃淡の差が減少してしまうことである．主な原因は，照明条件や露出の調整不良である．古典的な手法では，ヒストグラム平坦化

やトーンマッピングがある．図 2.1 のようなトーンカーブを用いて，画像の輝度値の分布を調整することでコントラストを調整する．トーンカーブは様々な関数があり次式はその一例である．

$$\boldsymbol{x} = \frac{1}{\exp(0.03 \times (-\boldsymbol{y} \times 255 + 170))} \tag{2.3}$$

深層学習ベースの手法は人間の視覚メカニズムである Retinex 理論に基づいた手法等 [37, 39–41, 55] が提案されている．

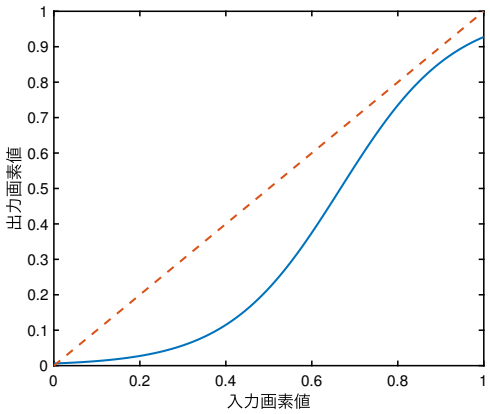


図 2.1　コントラスト強調に用いるトーンカーブの例

2.1.4　解像度低下

解像度低下は，画像の詳細情報が欠落し，画像が鮮明でなくなる現象である．画像の縮小，圧縮，伝送時の情報損失によって引き起こされる．バイリニア補間は，周囲のピクセルの値を線形補間して新しいピクセルの値を推定する．バイキュービック補間は，周囲の 16 個のピクセルを重み付けして補間する．深層学習ベースである EnhanceNet [56] は Sajjadi らによって 2017 年に提案された．End-to-end のシンプルなネットワークに低解像度画像を通し，出力と正解高解像度画像の損失関数が最小となるように学習する．

2.1.5　画像欠損

画像欠損は，画像データの一部が失われたり損傷したりした結果，画像に欠落や不連続な部分が生じる現象である．画像欠損の主な原因は，物理的な損傷や障害物の存在などがある．画像のインペインティングとは，画像内の欠損領域の画素を補間すること

で，自然で違和感のないように画像を修復する技術のことである．テクスチャ合成手法の一つにパッチベースのインペインティング手法がある．未欠損領域内から自然な見た目になりそうなパッチを探索して欠損部分に貼り付けて修復する．深層学習ベースの手法では，2016 年に Pathak らによって Context Encoders [57] が提案された．入力画像にランダムにマスキングを施し，そのマスクされた領域を補間するように学習を行う．

2.2 空間フィルタリング

領域ベースの濃淡変換では，入力画像の対応する画素だけでなく，周囲の画素も含めた領域内の画素値を使用して計算する．この手法は空間フィルタリングと呼ばれ，それに使用されるフィルタは空間フィルタとして知られている．さらに，空間フィルタは一般的に線形フィルタと非線形フィルタに分類される．線形フィルタでは，入力画像を $\boldsymbol{x}(i,j)$，出力画像を $\boldsymbol{y}(i,j)$ として，以下の式で計算される．

$$\boldsymbol{y}(i,j) = \sum_{n=-w}^{w} \sum_{m=-w}^{w} \boldsymbol{x}(i+m, j+n) h(m,n) \tag{2.4}$$

ただし，$h(m,n)$ はフィルタの係数を表す配列で，$(2w+1) \times (2w+1)$ の大きさを持つ．この空間フィルタのうち，本論文で使用するエッジ検出フィルタと平滑化フィルタについて以下で紹介する．

2.2.1 エッジ検出フィルタ

エッジ検出は，画像分割や画像強調のタスクにおいて，画像内の物体の境界を見つけるための画像処理技術である．エッジ検出フィルタの基本的なものの一つに，隣接する画素値の差を計算する微分フィルタが挙げられる．しかし，微分フィルタはエッジだけでなくノイズにも反応する傾向がある．エッジを抽出したながら画像のノイズを減らすために，いくつかの方法が提案されている．ソーベルフィルタ [58] とは，平滑化フィルタと一次微分フィルタを組み合わせたものである．ソーベルフィルタは加重平均を利用して，中心画素からの距離に応じて画像を平滑化する．式 (2.5) において，h_{SX} と h_{SY} はそれぞれ垂直ソーベル演算子と水平ソーベル演算子を表す．

$$h_{SX} = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, \qquad h_{SY} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \tag{2.5}$$

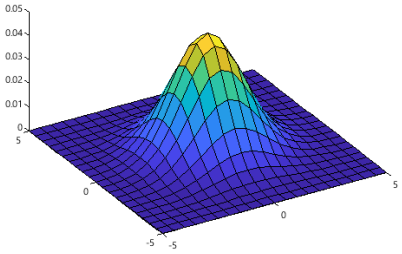


図 2.2　2 次元ガウス分布の例

一方，ラプラシアンフィルタ [59] とは二次微分フィルタである．一般に，関数 $f(x, y)$ のラプラシアンは以下のように定義できる．

$$\nabla^2 f(x_1, x_2) = \frac{\partial^2}{\partial x_1^2} f(x_1, x_2) + \frac{\partial^2}{\partial x_2^2} f(x_1, x_2) \tag{2.6}$$

デジタル画像においては，垂直方向の二次微分と水平方向の二次微分を足し合わせることによってラプラシアン値を求める．式 (2.7) において，h_{L4} と h_{L8} はそれぞれ，4 近傍ラプラシアンフィルタと 8 近傍ラプラシアンフィルタである．

$$h_{L4} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \qquad h_{L8} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{2.7}$$

2.2.2　平滑化フィルタ

写真を撮影するとき，ピントが合っていないぼけたような画像になることがある．そのような画像の濃淡変化は，ぼけていない画像と比べて滑らかである．画像処理によって濃淡変化を滑らかにする処理を平滑化という．画像に含まれるノイズなどを軽減するために用いられる．平滑化フィルタの最も簡単なものに平均化フィルタがある．下式は 3×3 画素の平均化フィルタである．フィルタサイズが大きくなるほど，平滑化の効果は大きくなる．

$$h_{average} = \begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix} \tag{2.8}$$

一方，単純な平均値ではなく，フィルタの原点に近いほど大きな重みを付ける加重平均フィルタもある．その重みをガウス分布に近づけたものをガウシアンフィルタとい

う．平均 0，分散 σ^2 のガウス分布は次式で表される．

$$h_g(x) = \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{x^2}{2\sigma^2}} \tag{2.9}$$

それらを 2 次元に拡張したガウス分布は次式で表され，デジタル画像処理ではこれが用いられる．

$$h_g(x_1, x_2) = \frac{1}{2\pi\sigma} \mathrm{e}^{-\frac{x_1^2+x_2^2}{2\sigma^2}} \tag{2.10}$$

図 2.2 は式 (2.10) において $\sigma = 8$ の時のグラフである．ガウシアンフィルタは，単純な平均化フィルタに比べ，より滑らかで自然な平滑化の効果が期待できる．

2.3 画像品質評価手法

劣化画像の復元性能を評価するにあたり，画像品質評価は重要である．画像品質評価手法は主観的手法と客観的手法に分けられる．主観的手法では，劣化画像と性能を比較したい復元画像を並べて，より自然な見た目の画像や，正解画像がある場合は正解画像に近い画像が優れた手法であると評価する．複数の被験者に順位付けをさせてその平均点を定量比較する方法が一般的である．

一方，客観的手法は定量的かつ公平に画像復元性能を比較するものである．いくつかの調査論文 [60, 61] で述べられているように，客観的な画像品質評価指標は完全参照（Full Reference），部分参照（Reduced Reference），非参照（No Reference）に分けられる．以下では劣化画像復元でよく用いられる完全参照指標と非参照指標の代表的な手法について述べる．

2.3.1 完全参照指標

完全参照法の画像品質評価指標は，評価対象の画像と参照画像との間の差異を評価する．評価対象の画像と参照画像を比較し，画像間のピクセルレベルの差異や統計的な特徴の類似性を解析する．代表的な完全参照指標には，ピーク信号対雑音比（PSNR: Peak Signal to Noise Ratio）や構造類似性指標（SSIM: Structural Similarity）[62] がある．

PSNR は，画像復元によく用いられ，

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}} \right) = 20 \log_{10} \left(\frac{\mathrm{MAX}}{\sqrt{\mathrm{MSE}}} \right) \tag{2.11}$$

と定義される．単位はデシベル [dB] である．ここで，MAX は画素値の最大値を表し，1 画素 8 bit の画像なら画素値は 0 から 255 の間の値をとるので最大値は 255 である．また，MSE(Mean Square Error) は平均二乗誤差を表しており，

$$\mathrm{MSE} = \frac{1}{MN}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\|\boldsymbol{x}(i,j) - \hat{\boldsymbol{x}}(i,j)\|_2^2 \tag{2.12}$$

と求められる．M, N は画像の縦横のサイズである．ただし，カラー画像の場合は RGB チャネル方向も考えれば同様に求められる．式 (2.11) より，PSNR が大きいと MSE が小さいことを意味しているので，処理画像 $\hat{\boldsymbol{x}}$ が原画像 $\boldsymbol{x}$ に近いことがわかる．一般に高画質とされるのは 35〜45 dB とされており，0.2 dB 程度異なれば視覚的にもかなり変化があるとされる．

SSIM [62] は画像の構造的な情報から画質を評価する手法である．PSNR が符号化にともなって生じたノイズ成分の知覚感度に基づく指標であるのに対し，SSIM は画像構造の類似度が人間の画質劣化の知覚に寄与するという仮説に基づいて定義されている．数式で表すと，

$$\mathrm{SSIM} = \frac{(2\mu_{\boldsymbol{x}}\mu_{\hat{\boldsymbol{x}}} + C_1)(2\sigma_{\boldsymbol{x}\hat{\boldsymbol{x}}} + C_2)}{(\mu_{\boldsymbol{x}}^2 + \mu_{\hat{\boldsymbol{x}}}^2 + C_1)(\sigma_{\boldsymbol{x}}^2 + \sigma_{\hat{\boldsymbol{x}}}^2 + C_2)} \tag{2.13}$$

となる．$\mu_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}\hat{\boldsymbol{x}}}$ はそれぞれ $\boldsymbol{x}$ の画素値の平均と標準偏差，$\boldsymbol{x}$ と $\hat{\boldsymbol{x}}$ の共分散である．また，C_1, C_2 は分母の値が非常に小さくなった時に評価値が不安定にならないための定数で，$C_1 = (0.01 \cdot \mathrm{MAX})^2, C_2 = (0.03 \cdot \mathrm{MAX})^2$ を用いることが多い．SSIM の値は最大が 1 で値が大きいほど処理画像 $\hat{\boldsymbol{x}}$ が原画像 $\boldsymbol{x}$ に近いことを意味する．

他にも，MAD (Most Apparent Distortion measure) [63] や VSI (Visual Saliency Induced quality index) [64]，FSIM (Feature Similarity index) [65]，GMSD (Gradient Magnitude Similarity Deviation) [66]，LPIPS (Learned Perceptual Image Patch Similarity) [67]，IQT(Image Quality Transformer) [68] などの手法がある．低照度画像強調では，低照度画像の性能を評価するのに特化した指標もある．例えば，LIEQAIndex(Low-light Image Enhancement Quality Assessment Index) [69] は，輝度強調，色再現，ノイズ評価，構造保存の観点から性能を評価する．

2.3.2 非参照指標

非参照法の画像品質評価指標は，参照情報（オリジナル画像や理想的な画像）を使用せずに画像の品質を評価する手法である．画像の特徴や統計情報を解析して品質を推定する．参照情報を必要としないため，リアルタイムの応用や大規模な画像データセット

の評価に有用だが，完全参照指標と比べて推定の正確性や一貫性に制約がある場合が多い．

多くの論文で用いられるのは NIQE(Natural Image Quality Evaluator) [70] や BRISQUE(Blind/Referenceless Image Spatial Quality Evaluator) [71] である．NIQE は画像の自然さや真実性を評価する指標である．評価対象の画像自体に含まれる統計的な特徴を分析し，画像のノイズレベル，シャープネス，エッジの質などを考慮して品質を評価する．BRISQUE は，画像の空間的な品質を評価する無参照法の指標である．画像の統計的特徴，コントラスト，エッジの鮮明さ，ノイズの存在などを解析して品質を推定する．

非参照指標には, ほかにも NFERM(No-reference Free Energy based Robust Metric) [72], MANIQA(Multi-dimension Attention Network for No-reference Image Quality Assessment) [73] などがある．低照度画像強調の分野では Zhang ら [74] が，低照度画像強調専用の非参照指標を提案している．彼らの手法では，光強調，色比較，ノイズ測定，構造評価の 18 の特徴を，ラベル付きのペアデータから学習している．

2.4 深層学習

2.4.1 Convolutional Neural Network

深層学習は，層を深くしたニューラルネットワークであり，画像処理の分野では主に畳み込みニューラルネットワーク（CNN）が広く使われている．CNN は，入力層，出力層，そして複数の隠れ層で構成されている．. 第 l 層のデータ $\boldsymbol{x}_{l-1}$ は

$$\boldsymbol{A}_l = \boldsymbol{W}_l * \boldsymbol{x}_{l-1} + \boldsymbol{b}_l \tag{2.14}$$

として計算される．ここで，$*$ は畳み込み演算を表し，$\boldsymbol{W}_l$ と $\boldsymbol{b}_l$ はそれぞれ，重みとバイアスであり，ネットにより学習されるパラメータである．この $\boldsymbol{A}_l$ を活性化関数 $\phi(\cdot)$ に入力した時の応答

$$\boldsymbol{Z}_l = \phi(\boldsymbol{A}_l) \tag{2.15}$$

が第 l 層の出力となり，次の層へと伝播される．近年の研究で画像認識や復元タスクで CNN ベースの手法が多く提案されている．[57, 75–78]

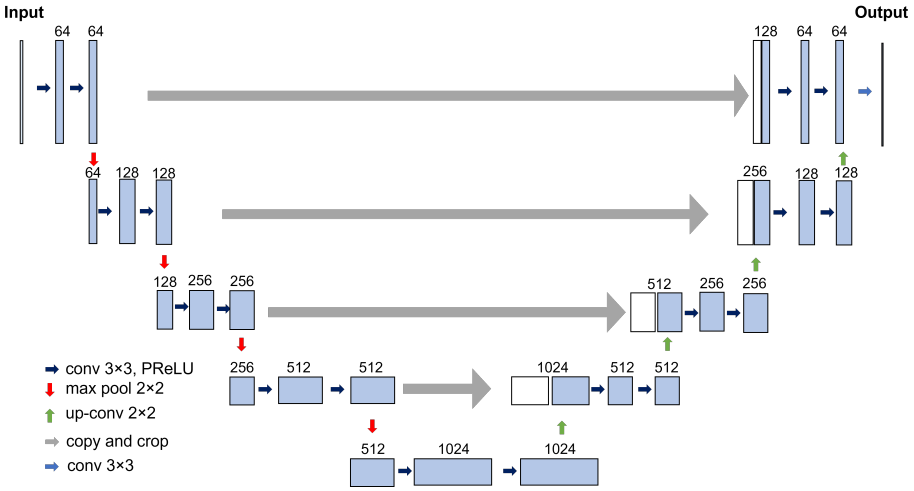


図 2.3　U-Net

2.4.2　U-Net

CNN ベースの画像分類タスクでは，畳み込み層は局所的な特徴を抽出し，プーリング層は位置情報を曖昧にする役割を果たす．一方，画像分類タスクでは，物体のサイズや位置のずれに対してロバストである必要があるが，領域分割タスク [79–81] では大局的な特徴と局所的な特徴を組み合わせる必要がある．そこで，Ronneberger ら [81] によって 2015 年に U-Net が提案された．U-Net は図 2.3 のようにエンコーダとデコーダの二つのパスで構成されている．エンコーダでは，畳み込み層とプーリング層からなり，画像の特徴を低次元の表現に変換する．デコーダでは，アップサンプリングと畳み込み層からなり，低次元表現を元の入力サイズに復元し，セグメンテーションマップを生成する．エンコーダとデコーダの間にあるスキップコネクションでは，エンコーダの特徴マップをデコーダの対応する層に連結することで，大局的な特徴と局所的な特徴を組み合わせることができる．

2.4.3　U-Net++

U-Net++ は，U-Net の改良版として 2018 年に Zhou ら [82] によって提案されたモデルである．U-Net++ の構造を図 2.4 に示す．U-Net のエンコーダとデコーダの代わりにより小さな U-Net モジュールを使用し多層の構造をもつことによってより低次元の表現を生成することができる．また，スキップコネクションの数を増やすことで異な

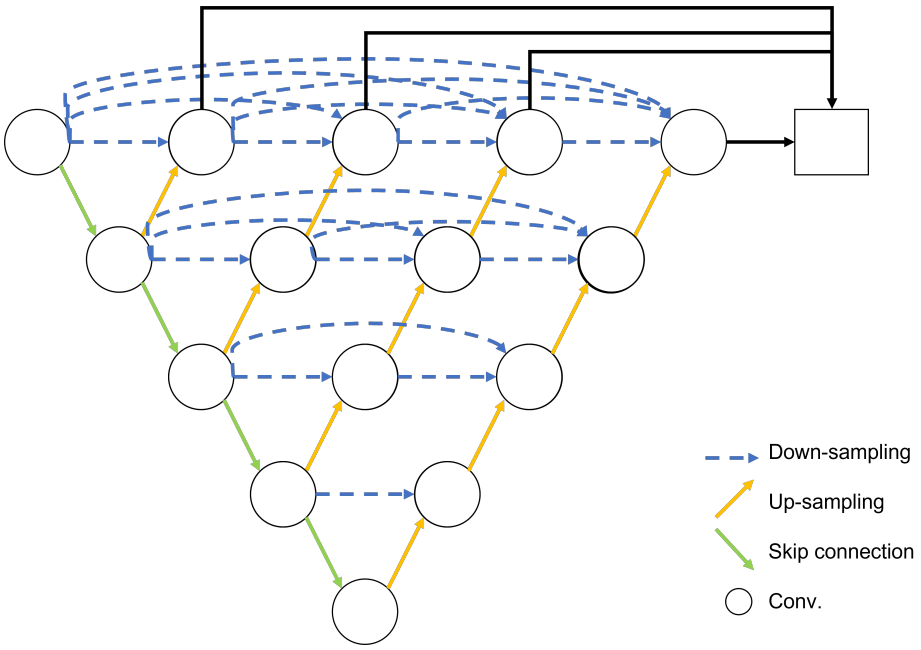


図 2.4　U-Net++

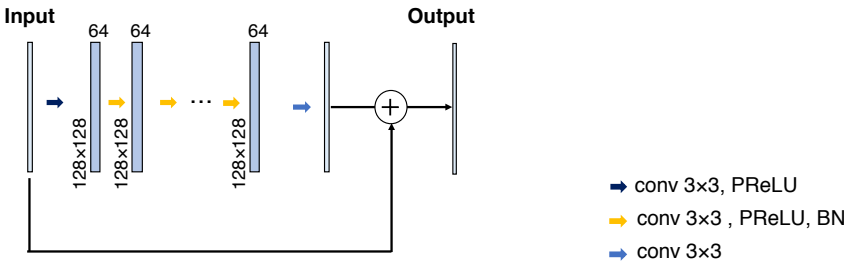


図 2.5　ResNet

る解像度の特徴マップを結合しより多様な情報を考慮することができる．

2.4.4　ResNet

CNN の残差学習 [83] とは，層数が増えると学習の精度が低くなるというトレードオフを解消するために提案されたものである．図 2.5 の ResNet で見られるように，ネットワークはスキップ接続を持ち，出力が入力に加算されることによって最終出力が生成される．これは，残差を学習することが，入力と出力の関係を直接学習させるよりも効果的であるという仮説に基づいている．この残差学習ベースの手法は多くの画像修復タ

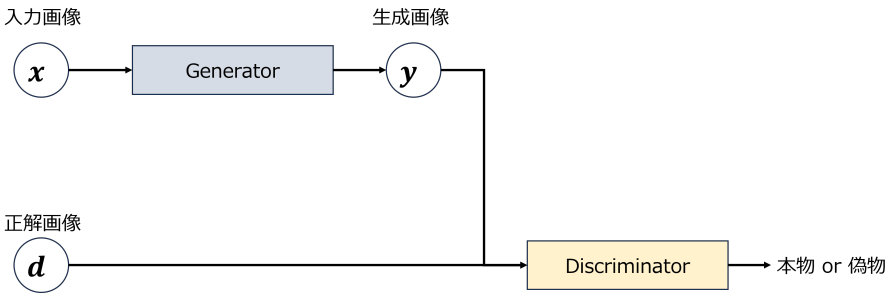


図 2.6　GAN のアルゴリズムの概要

スクで成果を出している [53, 84].

2.4.5　GAN

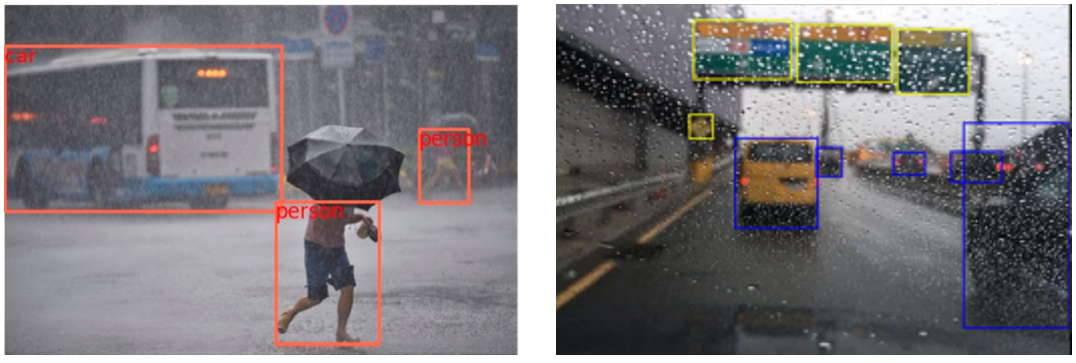
GAN は 2014 年に Goodfellow ら [85] によって提唱された深層学習ネットワークである．ゲーム理論に着想を得て，GAN は生成器（Generator）と識別機（Discriminator）の二つのニューラルネットワークが相互に競い合う構造を持っている．図 2.6 に GAN のアルゴリズムの概要を示す．Generator を $G(\cdot)$ すると，$\boldsymbol{y} = G(\boldsymbol{x})$ は入力画像 $\boldsymbol{x}$ から画像 $\boldsymbol{y}$ を生成する役割を果たす．Discriminator を $D(\cdot)$ とすると，$s = D(\boldsymbol{y})$ は生成画像 $\boldsymbol{y}$ がどれだけ訓練データ $\boldsymbol{d}$ の分布に近いかを識別する．GAN は学習中に不安定になる傾向があり不自然なアーチファクト生成してしまうことが多い．この問題を解決するために，多くの研究者が GAN を最適化する方法を模索してきた．CGAN [86] では，条件変数が導入されている．Salimans ら [87] は，学習を改善するためにミニバッチ単位での Discriminator を提案している．Creswell ら [88] はあるタスクに特化し損失関数を導入している．EBGAN [89] では，Discriminator をエネルギー関数と見なして処理をしている．GAN が多くの研究者の関心を集めている現在，多くの画像処理タスクに GAN が用いられている．例えば，テキストから画像への合成 [90]，単一画像の超解像 [91]，画像のインペインティング [75] などである．

第 3 章

複数の雨モデルを考慮した GAN に基づく雨すじ除去

3.1　雨すじ除去について

本章では画像劣化のうち天候環境が要因となり劣化した画像の復元について扱う．監視カメラや車載カメラなど屋外に設置されることの多い装置では天候によって画像の見た目が大きく変わってしまう．雨が降っているときには，雨すじが画像のブレを引き起こすだけでなく，光の散乱による霞（ヘイズ）をも引き起こし画質を下げてしまう．肉眼では雨すじを無視して物体を認識することができても，画像認識や物体追跡などの応用を考えると，ノイズである雨すじが足かせとなる．図 3.1 は雨天時に屋外で撮影された画像における雨すじ除去の応用例である．



(a) 屋外カメラ画像での物体認識　　(b) 車載カメラ画像での物体追跡

図 3.1　雨すじ除去の応用シーン

そこで，今必要とされている技術の一つが，画像から雨すじを除去すること (derain) である．ここ数十年で，雨すじ除去手法が多く提案されている．画像から雨すじを除去する方法は大きく分けて 2 種類あり，動画ベースの手法と静止画ベースの手法である．静止画ベースの手法は，動画ベースの手法と比べると少ない情報しか得られないので困難な問題であるとされているが，提案されている手法が少なく改善の余地が多くある．

3.2 雨すじ除去の従来法

画像から雨すじを除去する方法は大きく分けて 2 種類あり，動画ベースの手法と静止画ベースの手法であり，本論文では静止画ベースの手法に焦点を当てる．

3.2.1 動画ベースの手法

動画ベースの手法は 2004 年に初めて Garg ら [2] によって提案された．Garg らの手法では，相関モデルを用いて雨すじを検出し，隣接フレームからとった平均画素値を使って雨すじを除去する．他にも，雨の方向のヒストグラムを用いて雨すじを検出しガウス混合モデルにより雨すじを除去する手法 [3] やマルチスケール畳み込みスパース符号化を用いた手法 [4] などがある．動画ベースの手法は隣接フレームの情報を利用するので精度は高くなるが時間情報も考慮するので計算コストは高くなってしまう．例えば車載カメラでは衝突回避などに用いる場合はリアルタイム性が求められるので動画ベースの手法は適さない．

3.2.2 静止画ベースの手法

動画ベースの雨すじ除去と比べて，静止画ベースの雨すじ除去は情報が少ないため困難なタスクである．2012 年頃から事前情報をベースとした手法が提案されている．スパース性に基づく手法 [5, 6]，低ランク表現に基づく手法 [7]，ガウス混合モデルに基づく手法 [8] がある．これらの手法は，雨すじを残してしまうこととや過平滑化されてしまう傾向がある．

2017 年頃から深層学習をベースにした手法が提案されるようになった．シンプルな CNN ベースの手法 [9, 10] や GAN ベースの手法 [11, 12] などがある．合成された雨画像と雨なし画像のペアを集めたデータセットを使用しネットワークを学習させる．

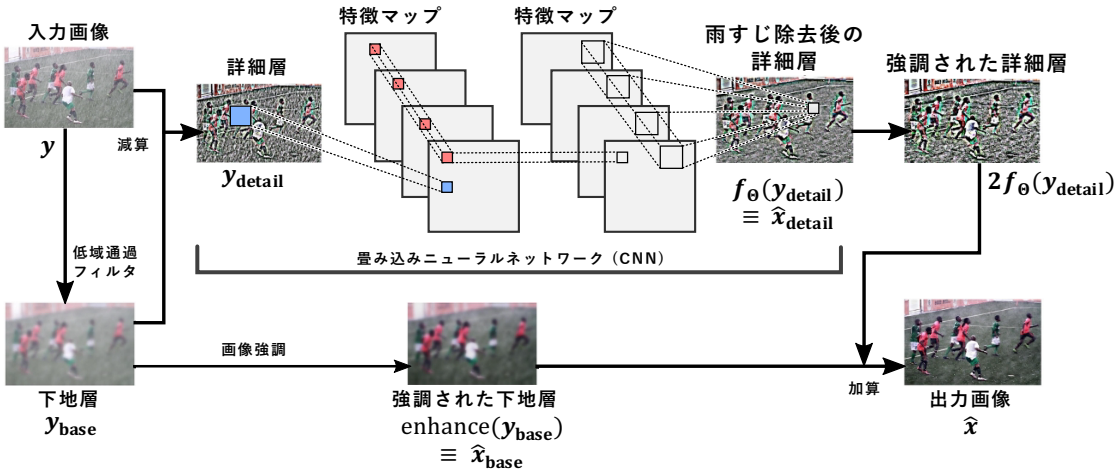


図 3.2　DerainNet [9] の概要

DSC (ICCV'15) [6]

DSC(Discriminative Sparse Coding) は識別的スパース符号化を用いたという辞書ベースでの雨すじ除去手法である．雨画像を雨層と背景層に分離するために，学習済みの辞書とスパース係数を使用する．具体的には，雨画像をパッチに分割し，雨層のパッチの集合 Y_1 と背景層のパッチの集合 Y_2 の組み合わせとみなす．Y_1 と Y_2 が辞書 D の下でスパースに近似できると仮定する．

$$Y_1 \approx DC_1, Y_2 \approx DC_2, \tag{3.1}$$

ここで，C_1 と C_2 はスパース符号である．次に，学習済みの辞書を使用して，パッチをスパース符号化することで，各パッチが辞書のアトムの線型結合として表現される．雨層はスパースな性質を持つため，雨層のパッチのスパース符号化には少ない辞書アトムが使用される．最終的に，スパース符号化された雨層の係数と，雨画像から計算された背景層を組み合わせることで，雨の取り除かれた画像を再構築する．

DerainNet (TIP'17) [9]

DerainNet は CNN と画像分解を組み合わせた手法である．概要を図 3.2 に示す．次式のように，入力した雨画像 $\boldsymbol{y}$ をガイデッドフィルタのような低域通過フィルタを用いて下地層 $\boldsymbol{y}_{\text{base}}$ と詳細層 $\boldsymbol{y}_{\text{detail}}$ に分解する．

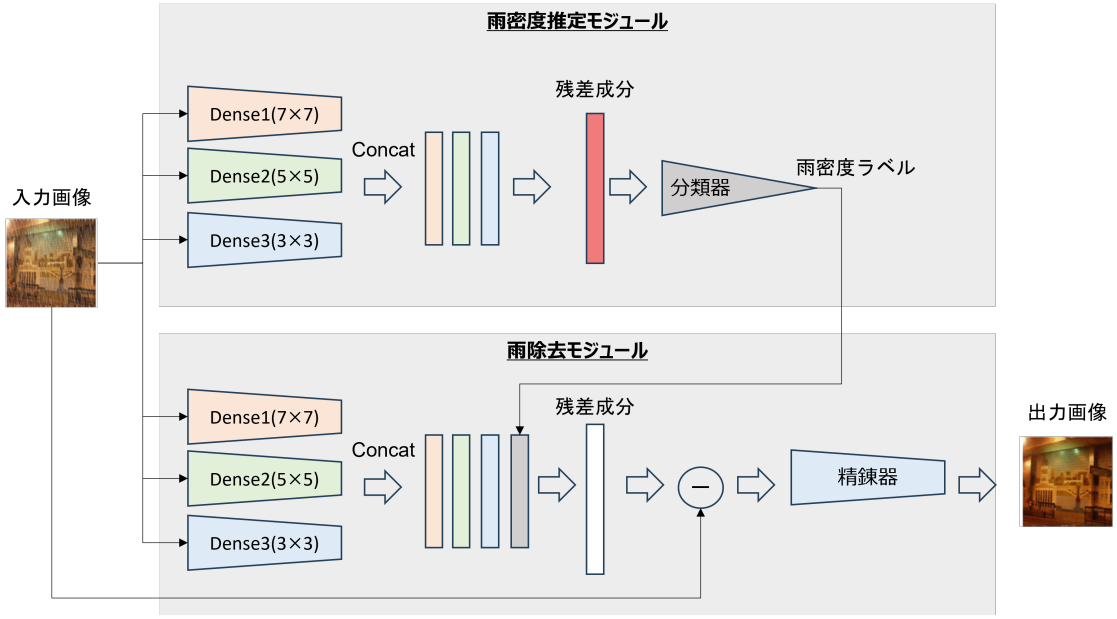


図 3.3　DID-MDN [11] の概要

$$\begin{aligned} \boldsymbol{y} &= \mathrm{LPF}(\boldsymbol{y}) + (\boldsymbol{y} - \mathrm{LPF}(\boldsymbol{y})) \\ &= \boldsymbol{y}_{\mathrm{base}} + \boldsymbol{y}_{\mathrm{detail}} \end{aligned} \tag{3.2}$$

LPF(·) は低域通過フィルタ処理を示す．このように画像を周波数分解すると，詳細層（高周波部）に画像のエッジや雨すじが残り，下地層（低周波部）にはぼやけた画像が残る．DerainNet [9] では，低周波部分である下地層には雨すじがないと仮定し，高周波部分である詳細層に 3 層の CNN を用いて雨すじを除去させる．

$$\begin{aligned} \boldsymbol{z}_0 &= \boldsymbol{y}_{\mathrm{detail}}, \\ \boldsymbol{z}_l &= \phi\left(\boldsymbol{W}_l * \boldsymbol{z}_{l-1} + \boldsymbol{b}_l\right), \quad l = 1, 2 \\ \hat{\boldsymbol{x}}_{\mathrm{detail}} &= \boldsymbol{W}_3 * \boldsymbol{z}_2 + \boldsymbol{b}_3 \end{aligned} \tag{3.3}$$

ここでは，$\phi(\cdot)$ は活性化関数として tanh 関数を用いている．下地層と CNN に通した詳細層を足し合わせることで最終的な雨なし画像を生成する．

$$\hat{\boldsymbol{x}} = \mathrm{enhance}(\boldsymbol{y}_{\mathrm{base}}) + \alpha f_\Theta(\boldsymbol{y}_{\mathrm{detail}}) \tag{3.4}$$

ここで，α は見た目を良くするために調整する係数である．

DID-MDN (CVPR'18) [11]

DID-MDN(Density-aware Single Image De-raining using a Multi-stream Dense Network) は 2 段階に分かれており，雨の密度を推定するモジュールと，推定した雨密度と入力雨画像から雨すじ除去をするモジュールで構成されている．概要を図 3.3 に示す．DID-MDN では，雨画像を加算合成モデルとして捉えている．

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{r} \tag{3.5}$$

ここで，$\boldsymbol{y}$ が雨画像，$\boldsymbol{x}$ が雨なし画像，$\boldsymbol{r}$ が雨ノイズ成分である．一つ目の雨の密度を推定するモジュールでは，DenseBlock を束ねて出力した残差成分 $\boldsymbol{r}$ を分類器にかけて雨密度を 3 段階で評価する．残差成分を出力するネットは，$3 \times 3, 5 \times 5, 7 \times 7$ の異なる三つのカーネルサイズの DenseBlock 群で構成され，六つの DenseBlock で構成されたストリームを 3 種（カーネルサイズがそれぞれ $3 \times 3, 5 \times 5, 7 \times 7$）からなる．分類器は三つの畳み込み層と平均プーリング，二つの全結合層で構成されている．

二つ目の雨すじ除去モジュールでは，まず雨密度推定と同様の DensBlock のストリームの出力と雨密度から残差 $\boldsymbol{r}$ を推定する．入力画像から残差成分を引いた $\boldsymbol{y} - \boldsymbol{x}$ を二つの畳み込み層と ReLU からなるネットに通して最終的な雨すじ除去画像を生成する．

3.2.3 従来法の問題点と提案法における改良点

従来法で様々なアプローチが提案されているが次の問題点がある．

- 多様な強さや方向，長さの雨すじを考慮しきれておらず，自然画像では雨すじが残ってしまう（主に DSC [6])
- 低周波領域に残った雨成分の影響で全体的に白っぽい見た目になってしまう（主に DerainNet [9])
- 雨すじ除去のための平滑化が強く効きすぎており，被写体のエッジやテクスチャが失われてしまう (主に DID-MDN [11])

これらを解決するために，複数の雨すじモデルを考慮し GAN を用いた雨ノイズレベルや画像に対してロバストなモデルを提案する．

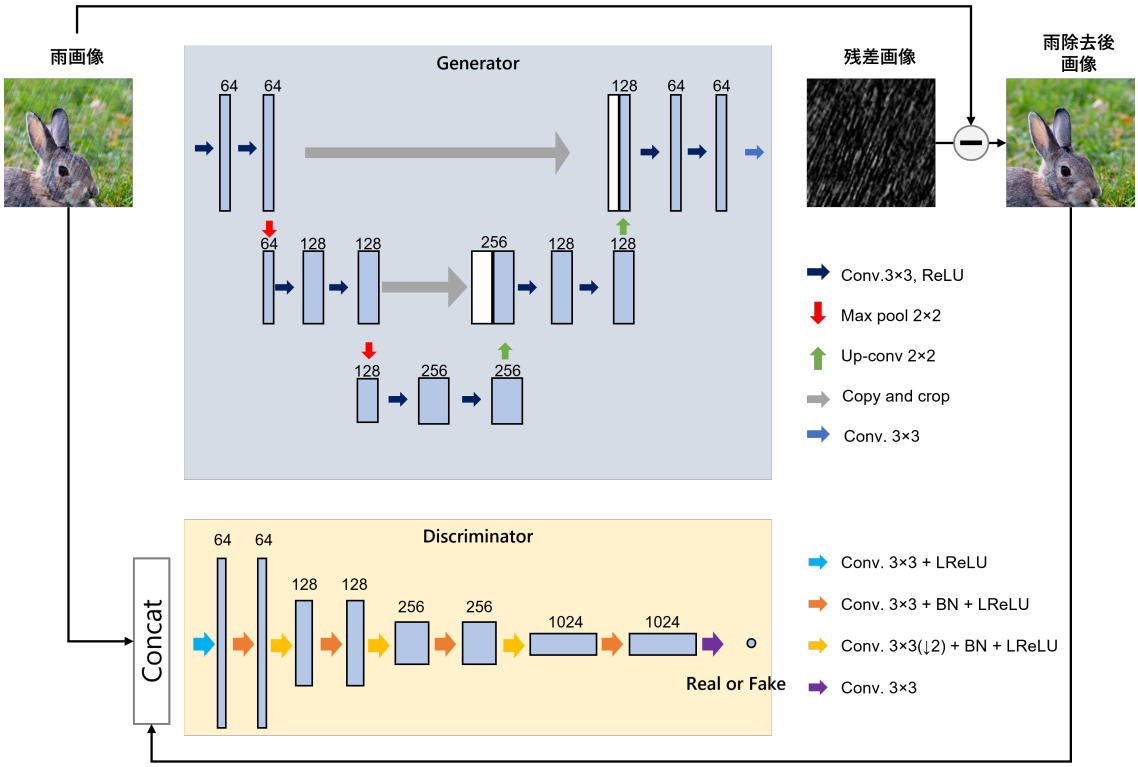


図 3.4　提案法のネットワークの全体像

3.3　提案法

3.3.1　ネットワーク構造

提案法の全体像を図 3.4 に示す．雨画像 $\boldsymbol{y}$ を入力して雨すじ除去後画像 $\tilde{\boldsymbol{x}}$ を出力する．従来法と異なる点が四つある．

まず第一に，画像分解手法を使わない．従来の多くの手法は，画像を高周波領域と低周波領域に分け，それぞれを独立して処理する．その結果，低周波領域に雨のノイズが残り，復元された画像が白っぽくなってしまう．

第二に，残差学習を採用した．CNN ベースの従来法の多くは直接雨すじ除去処理された画像を出力するが，被写体のエッジと雨のすじを正確に区別するのが難しい．それらを区別することに特化させ，ネットワークの出力は雨すじ除去後画像 $\tilde{\boldsymbol{x}}$ ではなく，残差画像 $\boldsymbol{y} - \tilde{\boldsymbol{x}}$ とする．雨すじ除去をするネットを $G(\cdot)$ とすると，

$$\tilde{\boldsymbol{x}} = \boldsymbol{y} - G(\boldsymbol{y}) \tag{3.6}$$

のように表せる．

第三に，U-Net をベースとしたネットワークで雨すじを検出した．雨すじは画像を大局的に見ると似たパターンで分布する一方，局所的には強さや大きさが変わる．雨すじと画像の背景を区別するためには，大局的な特徴と局所的な特徴の両方を考慮する必要があるのではないかという仮説のもと，U-Net を採用した．U-Net は階層が深くなるほど精度は高くなる傾向にはあるが過学習することもあり，また計算コストも高まってしまう．実験の結果深さは三層が適切であった．U-Net のデコーダでは 3 回マックスプーリングによるダウンサンプリングを行い，各階層では 3×3 の畳み込みと ReLU を 2 回ずつ適用している．デコーダも同様に 3 回アップサンプリングを行い，各階層では 3×3 の畳み込みと ReLU を 2 回ずつ適用した後，最終層では 3×3 の畳み込みを 1 回適用する．実験的にも U-Net の有用性は確認できており，ResNet と U-Net，U-Net++ による精度比較をした．表 3.1 は精度比較結果であり，赤字が一番精度が高く，青字が二番目に精度が高いことを示す．精度比較結果からもわかるように，U-Net 構造が雨すじ除去には最適である．一般的には U-Net よりもその改良版である U-Net++ の方が精度が高い傾向にある．しかし，精度比較結果から，BSD100 や Urban100 のような画像内の大部分を被写体が占めるような場合には，U-Net++ よりもむしろシンプルな U-Net の方が過学習しづらくロバストであるのではないかと考えられる．

第四に，GAN 構造を採用した．Generator では，雨すじ除去後画像を生成させる．Discriminator では，雨なし画像 x または生成画像 $\tilde{\boldsymbol{x}}$ と雨画像 y を連結させたものを入力させ，雨なし画像（本物）であるか生成された画像（偽物）であるかを識別する．Discriminator では，第一層目で 3×3 の畳み込みと LeakyReLU を適用する．その後合計 3 回のダウンサンプリングと合計 7 回の 3×3 の畳み込みと Batch Normalization と LeakyReLU を繰り返す．最終層では 3×3 の畳み込みを 1 回適用し，0 から 1 のスカラーを出力させる．

3.3.2 学習データセット

雨ノイズの作成

雨あり画像から雨すじを除去するために，雨あり画像と雨なし画像をデータセットにネットワークに通して学習させる必要がある．しかし，現実世界で雨の日と晴れの日で同じ場所の画像を取得することは困難なので，雨なし画像 $\boldsymbol{x}$ に雨のノイズ $\boldsymbol{v}$ を人工的に付加させることでネットワークを学習させる．従来法（ [9,11]）では，Photoshop を使って雨ノイズを合成していたが，時間と手間がかかる．そこで，乱数から雨ノイズ

表 3.1　ネットワーク構造による雨すじ除去精度の定量比較

	PSNR↑			SSIM↑		
アーキテクチャ	Rain100	BSD100	Urban100	Rain100	BSD100	Urban100
雨画像	21.08	23.31	22.54	0.760	0.830	0.822
ResNet	22.09	29.55	29.85	0.819	0.942	0.950
U-Net	22.11	30.76	30.38	0.812	0.953	0.955
U-Net++	22.16	29.01	29.42	0.821	0.934	0.946

$\boldsymbol{v}_{\text{rain}}$ を生成し，雨の密度や角度などのパラメータを調整できるようにした．図 3.5 に示すように，雨ノイズ生成のプロセスは三つのステップに従う．

まず，雨なし画像と同じサイズの一様乱数 $\boldsymbol{u} \in \mathcal{U}(0,1)$ を生成する．0 を境に正負に値を分布させるために λ だけシフトさせ，ノイズの量 σ_a を調節し，λ だけ逆シフトさせてから 0 から 1 の範囲にクリッピングをする．ランダムノイズの要素 $v_{\text{rand},i} \in \boldsymbol{v}_{\text{rand}}$ は以下のように計算される．

$$v_{\text{rand},i} \leftarrow \max\left(\min\left(\sigma_a(u_i - \lambda) + \lambda, 1\right), 0\right) \tag{3.7}$$

ここで，λ は 0.5 である．生成されたランダムノイズを図 3.5 (a) に示す．

次に，生成されたノイズにガウシアンフィルタ $\boldsymbol{\mathcal{F}}_g$ をかけてぼかす．フィルタ適用後の出力に対し，$\sigma_{T_{\min}}$ と $\sigma_{T_{\max}}$ の値を使用して正規化する．これらの閾値を用いることで，ノイズの量を減らし，そのコントラストを強調することができる．以下のように計算し，値を 0 から 1 の範囲にクリップする．

$$v_{\text{med},i} \leftarrow \max\left(\min\left(\frac{\hat{v}_{\text{rand},i} - \sigma_{T_{\min}}}{\sigma_{T_{\max}} - \sigma_{T_{\min}}}, 1\right), 0\right) \tag{3.8}$$

ここで，$\hat{\boldsymbol{v}}_{\text{rand}} = \boldsymbol{\mathcal{F}}_g \boldsymbol{v}_{\text{rand}}$ である．中間生成部としての雨ノイズを図 3.5 (b) に示す．

最後に，一定の方向をつけることで自然な雨すじを再現する．モーションフィルタ $\boldsymbol{\mathcal{F}}_m(\sigma_l, \sigma_\varphi)$ を適用した後，雨のスケール σ_s を調整する．フィルタ $\boldsymbol{\mathcal{F}}_{\boldsymbol{m}}$ は長さ σ_l と角度 σ_φ の関数である．

$$\boldsymbol{v}_{\text{rain}} \leftarrow \sigma_s \boldsymbol{\mathcal{F}}_m(\sigma_l, \sigma_\varphi) \boldsymbol{v}_{\text{med}} \tag{3.9}$$

最終的な雨のノイズを図 3.5 (c) に示す．

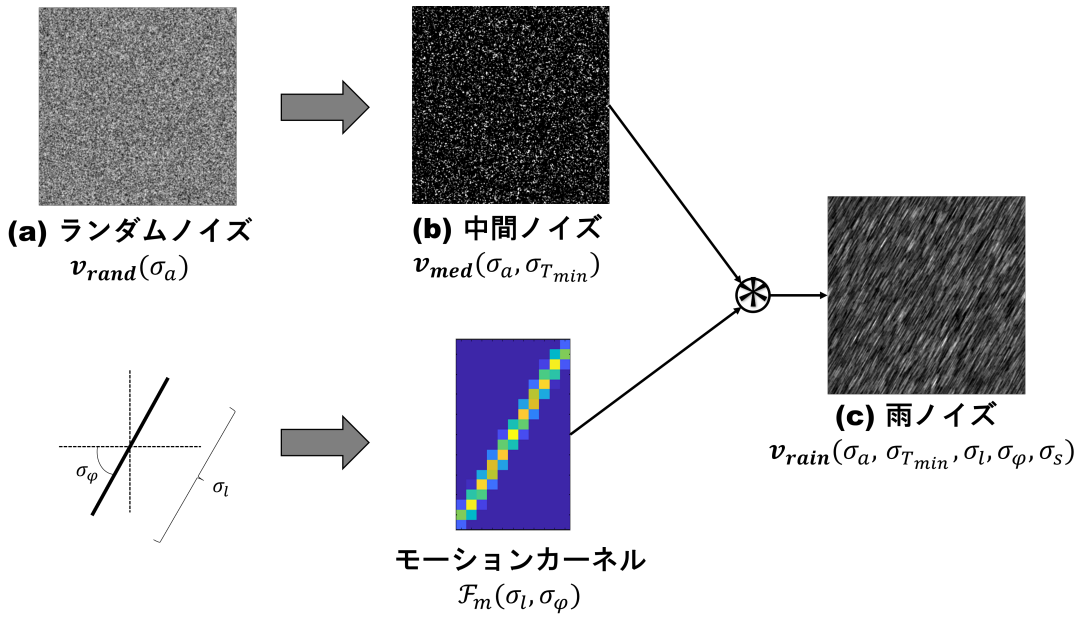


図 3.5　雨ノイズの作成フロー

雨なし画像と雨ノイズの合成方法

生成した雨ノイズを雨なし画像に合成する方法は線形加算合成モデルとスクリーン合成モデルがある．線形加算合成モデルは最も単純なモデルである．雨の画像 $\boldsymbol{y}_{\text{add}}$ が背景部分 $\boldsymbol{x}$ と雨ノイズ部分 $\boldsymbol{v}$ に分解できるという仮説に基づき，次式で表される．

$$\boldsymbol{y}_{\text{add}} = \boldsymbol{x} + \boldsymbol{v} \tag{3.10}$$

画像処理では，$\boldsymbol{y}_{\text{add}}$ の計算値は 0 から 1 の範囲にクリッピングするため，次のように書き換えられる．

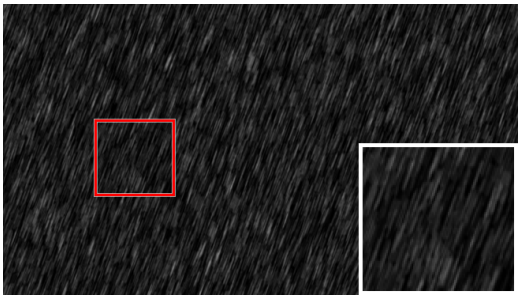
$$\boldsymbol{y}_{\text{add}} = \min\left(\boldsymbol{x} + \boldsymbol{v}, \boldsymbol{1}\right) \tag{3.11}$$

ここで，$\boldsymbol{1}$ は 1 で満たされたベクトルであり，$\min\left(\boldsymbol{X}, \boldsymbol{Y}\right)$ は $\boldsymbol{X}$ または $\boldsymbol{Y}$ から最も小さい要素を取る操作を示す．図 3.6 の (a) 原画像に対して (b) 雨ノイズを線形加算モデルで合成した画像が (c) である．図からもわかるように，線形加算モデルは，元の画像に雲や白いオブジェクトなどの明るい領域が含まれる場合には不自然な結果を生成する傾向にある．しかし，非常に激しい雨を再現する場合には適した合成手法である．

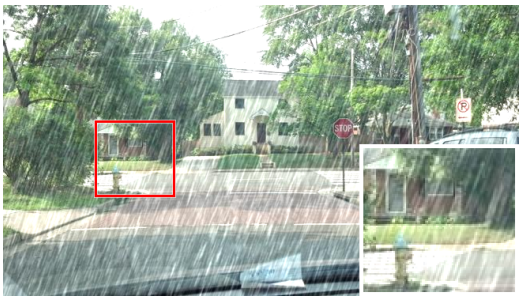
一方，スクリーン合成モデルは，加算モデルの不自然な見た目を解消するために提案されたもので，加算した結果に対し $\boldsymbol{x}$ と $\boldsymbol{v}$ の積を引くことで計算される．

$$\boldsymbol{y}_{\text{blend}} = \boldsymbol{x} + \boldsymbol{v} - \boldsymbol{x} \circ \boldsymbol{v} \tag{3.12}$$

(a) 原画像



(b) 雨ノイズ



(c) 線形加算モデル

(d) スクリーン合成モデル

図 3.6　雨合成モデルによる雨画像の見た目の比較

ここで，$\circ$ は要素ごとの乗算演算子を示す．スクリーン合成モデルによって生成された画像は，画像に明るい部分と暗い部分の両方が含まれる場合に自然に見える（図 3.6(d)）．

データ拡張

雨画像の学習データセットを作成するために，屋外で撮影された雨なし画像を集める必要がある．従来法の DerainNet [9] で使用された 900 枚の画像と DID-MDN [11] で使用された 4000 枚の画像の合計 4900 枚の画像を用いた．それらをランダムに $384 \times 384 \times 3$ のパッチに切り取り，画像を水平または垂直方向にランダムに反転させることでデータ拡張を行った．

現実世界における様々な雨のパターンを想定するために，角度，スケール，密度を完全にランダムに調整し雨ノイズを作成した．具体的には，雨のノイズパラメータ $\boldsymbol{v}_{\text{rain}}(\sigma_a.\sigma_{T_{\min}}, \sigma_l, \sigma_\varphi, \sigma_s)$ を $\sigma_{T_{\min}} \in [0.54, 0.62]$，$\sigma_l \in [5, 15]$，$\sigma_\varphi \in [50, 130]$，$\sigma_s \in [1.1, 1.5]$，および $\sigma_a \in [0, 2]$ に設定した．ここで，$\sigma \in [\sigma^a, \sigma^b]$ は，σ が σ^a から

σ^b の範囲内の値を取ることを表している．パッチごとに，雨なし画像と雨ノイズは線形加算合成モデルまたはスクリーン合成モデルの両方で合成された画像が含まれるようにした．また，出力画像を入力画像と同じサイズに保つために，畳み込み前にゼロパディングを行った．

3.3.3 損失関数

GAN では，Generator と Discriminator の損失関数を定義する必要がある．Gerenator のパラメータは次の損失関数を最小化することで最適化される．

$$\mathcal{L}_G = \mu_1 \mathcal{L}_{\mathrm{MSE}} + \mu_2 \mathcal{L}_{\mathrm{adv}} \tag{3.13}$$

ここで，μ_1 と μ_2 は実験的に決定した係数である．第一項の $\mathcal{L}_{\mathrm{MSE}}$ は，残差画像 $\boldsymbol{y_n} - \boldsymbol{x_n}$ と出力画像 $G(\boldsymbol{y}_n; \boldsymbol{\Theta_G})$ の平均二乗誤差である．

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{2N} \sum_{n=1}^{N} \| (\boldsymbol{y}_n - \boldsymbol{x}_n) - G(\boldsymbol{y}_n; \boldsymbol{\Theta_G}) \|_2^2 \tag{3.14}$$

ここで，n と N は画像のインデックスと画像の総数を示す．$\boldsymbol{\Theta_G}$ は生成器の学習されたパラメータである．第二項の $\mathcal{L}_{\mathrm{adv}}$ は Adversarial ロスであり，Discriminator の入力が真の雨なし画像か偽の雨なし画像かを評価し最適化される．Adversarial ロスは次のように表される．

$$\mathcal{L}_{\mathrm{adv}} = \frac{1}{2N} \sum_{n=1}^{N} \| 1 - D(\boldsymbol{x}_n, \boldsymbol{y}_n; \boldsymbol{\Theta_D}) \|_2^2 \tag{3.15}$$

一方，Discriminator のパラメータは，以下の損失関数を最小化することによって最適化される．

$$\mathcal{L}_D = \frac{1}{2N} \sum_{n=1}^{N} \left(\| D(\tilde{\boldsymbol{x}}, \boldsymbol{y}; \boldsymbol{\Theta_D}) \|_2^2 + \| V - D(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\Theta_D}) \|_2^2 \right) \tag{3.16}$$

ここで，V は平均値が 1 のガウス分布に従うランダムな値行列を示す．Discriminator を効果的に学習するために，定数値ではなくランダムな値を使用している．

3.3.4 学習パラメータ

学習には Pytorch フレームワークを用いた．ベースの学習率を 0.0002 に設定し，ソルバーは Adam [92] を用いた．Adam では二つの学習率を $\beta_1 = 0.5$，$\beta_2 = 0.999$ と設

表 3.2　Rain4 の合成雨画像生成に用いたパラメータ

画像名	サイズ	$\sigma_{T_{\min}}$	σ_l	σ_φ
Umbrella	350×550	0.65	20	100
Bird	480×576	0.66	22	120
Rabbit	640×494	0.57	22	70
Dock	768×512	0.66	22	70

定した．バッチサイズは 4，イタレーション回数は 4900×40 である．また，損失関数（式 (5.15)) の Adversarial ロスの比率は $\mu_1 = 1$，$\mu_2 = 0.001$ とした．

3.4　雨すじ除去の実験と比較

3.4.1　実験内容

合成雨画像と自然雨画像に対して三つの従来法との雨すじ除去性能比較を行った．

3.4.2　比較対象

本研究では深層学習ベースでの雨すじ除去に焦点を当てているため，次の三つの手法と比較した．

- DSC (ICCV'15) [6]：スパース符号化と辞書を用いて雨を検出し除去する．
- DerainNet (TIP'17) [9]：雨画像を高周波数領域と低周波領域に分割し，高周波領域に 3 層の CNN を通して雨を除去する．
- DID-MDN (CVPR'18) [11]：マルチスケールの DenseBlock を結合したネットワークで雨密度の推定と残差成分の推定して雨を除去する．

3.4.3　評価手法

合成画像の客観的評価として，PSNR と SSIM [62] を使用して雨すじ除去性能を定量的に比較した．テスト画像として，多くの論文でベンチマークに用いる Rain12 [9] と Rain100 [11] を採用した．また，表 3.2 の通りのパラメータを設定した合成雨画像

Rain4 [9] もテスト画像として使用した．さらに，より多くの屋外環境での雨すじ除去性能を比較するため，屋外画像のデータセット BSD100 と Urban100 に対してランダムに雨ノイズを付加したものもテストデータセットに用いた．合成画像の主観的評価では，複数の合成雨画像に対して見た目による比較を行った．

自然画像に対しては正解画像がないため主観的評価のみ行った．インターネットや [9] で使用されたテストデータセットから，現実世界の雨画像を収集した．合成画像には雨すじのみが含まれているが，実世界の画像にはヘイズと雨すじが含まれている．クリアな見た目とするために，すべての手法の出力結果に対して後処理としてデヘイジング手法 [93] を適用した．

3.4.4　合成画像での実験結果

主観評価結果

合成画像に対して雨すじ除去を実施した結果を図 3.7 と図 3.9 に示す．また，それぞれに対応するヒストグラムは図 3.8 と図 3.10 である．雨すじ除去結果を見ると，DSC [6] は相当量の雨すじが残っている．図 3.9 の Bird を見ると，DeraiNet [9] は雨すじ除去はうまくいっているが，全体的な画像が白っぽくなっている．特に鳥の羽が影響を受けている．図 3.10 のヒストグラムは全体的に右にシフトしてしまっている．この色の変化は，低周波成分に残った雨が原因と考えられる．一方，DSC [11] はほぼ完全に雨すじ除去されているが過平滑化されてしまい重要なディテールが失われている．提案法は，残差学習と U-Net の組み合わせにより，画像のコントラストおよびディテールを保持しながら雨すじをうまく除去できている．

客観評価結果

各テストデータセットに対して雨すじ除去を実施した結果の PSNR と SSIM をそれぞれ表 3.3 と表 3.4 に示す．提案法は Rain100 の SSIM を除き，各テストデータセットで最高の PSNR と SSIM を実現できている．DerainNet [9] は，SSIM は比較的高いが PSNR が低い傾向にある．画像全体が白っぽくなってしまうことから PSNR が低いのではないかと考えられる．DSC [6] と DID-MDN は [11] はデータセットによって性能にばらつきがあり，入力雨画像よりも PSNR または SSIM が下回ることもある．学習するデータセットの屋外画像に偏りがあることが原因ではないかと考えられる．

表 3.3　合成雨画像における雨すじ除去結果の PSNR

手法	PSNR↑				
	Rain4	Rain12	Rain100	BSD100	Urban100
雨画像	24.61	28.82	21.15	22.48	22.71
DSC [6]	27.93	28.71	21.04	26.65	26.21
DerainNet [9]	23.46	28.96	21.71	22.89	22.71
DID-MDN [11]	27.36	26.58	21.08	23.78	21.27
提案法	32.02	31.38	22.11	30.76	30.38

3.4.5　自然画像での結果

主観評価結果

図 3.11～図 3.13 に自然画像に対する雨すじ除去結果を示す．DSC [6] と DID-MDN [11] は，過平滑化されて重要なディテールが失われている．DerainNet [9] は，雨すじを除去することができるが，テクスチャとエッジがぼやけてしまっている．提案法では，ディテールとテクスチャを保持しながら雨を除去することができおり，合成画像だけでなく実世界の画像にもうまく対応していることがわかる．

3.4.6　複数の雨モデルを組み合わせることが学習結果に及ぼす影響

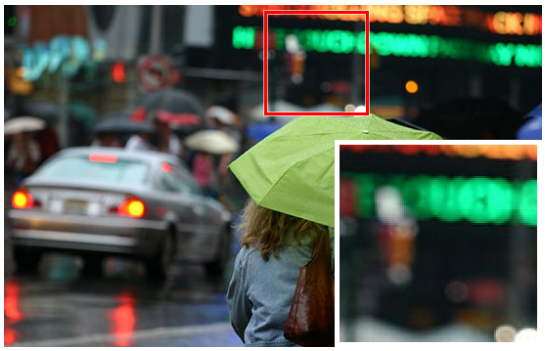
提案法では，複数の雨モデルを考慮したデータセットを作成して学習させた．提案法の複合雨モデルが雨すじ除去タスクに有効であることを実証するため，三つのトレーニングデータセットを用意した．一つ目は加算合成モデル (式 (3.11)) のみで雨画像を生成したデータセット，二つ目はスクリーン合成モデル (式 (3.12)) のデータセット，三つ目はそれらを組み合わせたデータセットである．図 3.14 に 3 種類のデータセットで学習した場合の雨すじ除去画像を示す．いずれか一方の合成モデルで学習したものに比べ，二つの合成モデルの混合で学習させた提案法は雨すじやアーチファクトを残さないことがわかる．

表 3.4　合成雨画像における雨すじ除去結果の SSIM

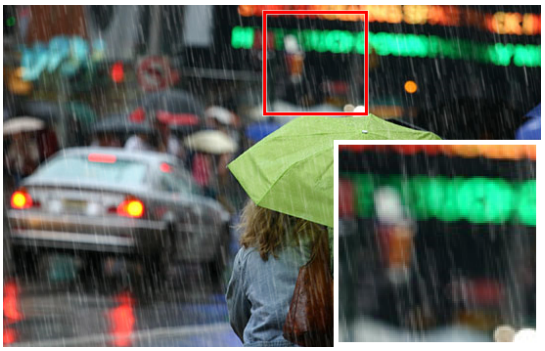
手法	SSIM↑				
	Rain4	Rain12	Rain100	BSD100	Urban100
雨画像	0.868	0.910	0.768	0.841	0.888
DSC [6]	0.921	0.916	0.764	0.878	0.891
DerainNet [9]	0.904	0.939	0.831	0.898	0.888
DID-MDN [11]	0.918	0.918	0.886	0.854	0.752
提案法	0.958	0.942	0.812	0.953	0.955

3.5　雨すじ除去のまとめ

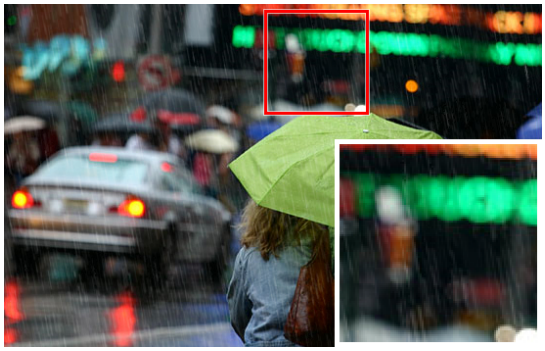
雨すじ除去では雨すじを残してしまうこと，過平滑化されてしまうこと，色彩が変わってしまうという問題点があった．提案法では，現実世界における雨の特徴を捉えた GAN ベースのネットワークを構築した．雨すじの大局的な特徴と局所的な特徴を捉えるために，Generator には U-Net 用い，残差画像を学習させた．データセットにおいては，雨の強さ，長さ，方向などを自動で調整できる雨ノイズ生成器を導入した．また，二つの雨画像合成モデルを組み合わせることで現実世界の雨を幅広く捉える工夫をした．実験では合成画像と自然画像の両方で従来法より雨すじ除去性能を上回ることができた．多様な雨の特徴を考慮したデータセットで学習させることにより，画像のディテールを失うことなく雨すじを除去することができた．今回は雨すじ除去精度を上げることを主目的にしていたため，実行時間の短縮については今後の課題である．



(a) 正解画像

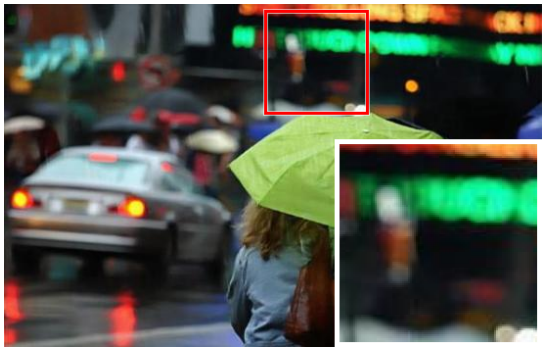


(b) 雨画像



(c) DSC [6]

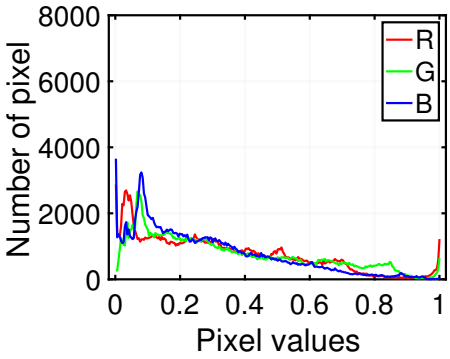
(d) DerainNet [9]



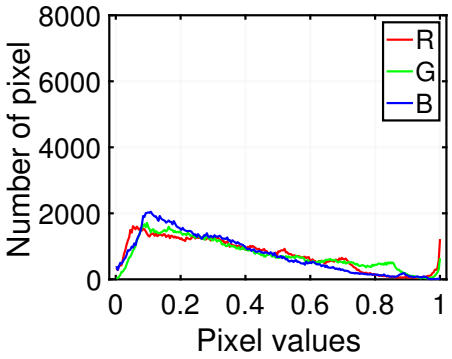
(e) DID-MDN [11]

(f) 提案法

図 3.7　合成画像での雨すじ除去結果 1（Umbrella）

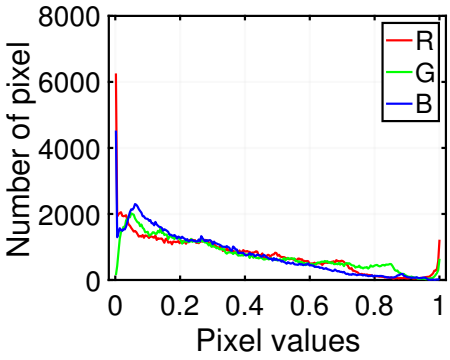


(a) Histogram of (a)

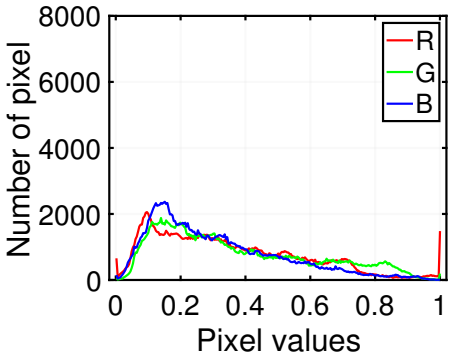


(c) 雨画像

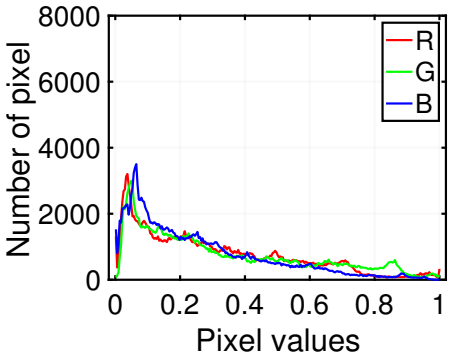
(b) 正解画像



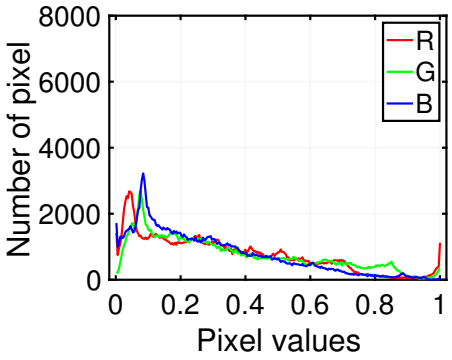
(d) DSC [6]



(e) DerainNet [9]

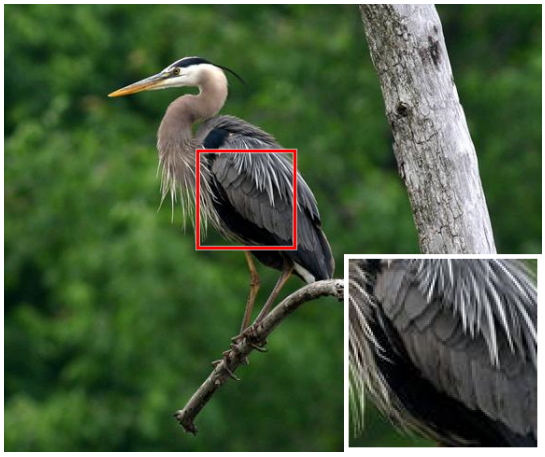


(f) DID-MDN [11]



(g) 提案法

図 3.8　合成画像での雨すじ除去画像のヒストグラム 1（Umbrella）

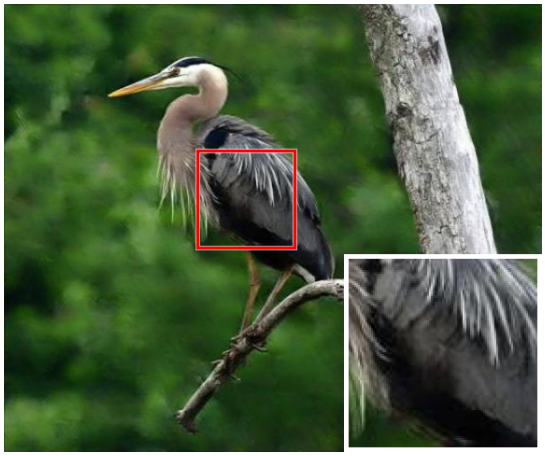


(a) 正解画像

(b) 雨画像

(c) DSC [6]

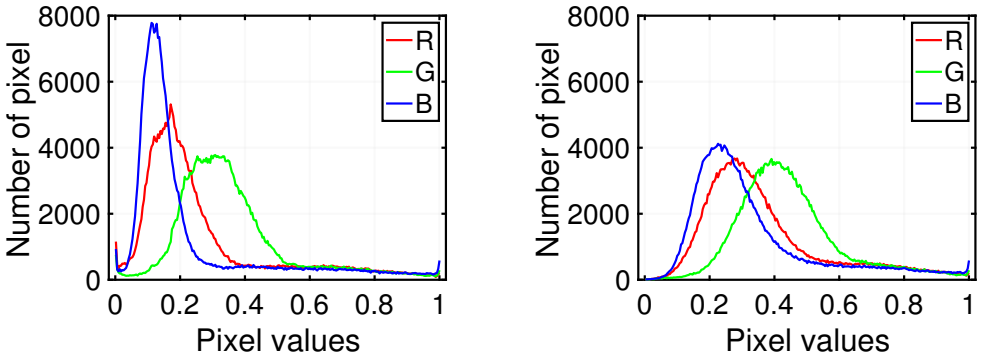
(d) DerainNet [9]



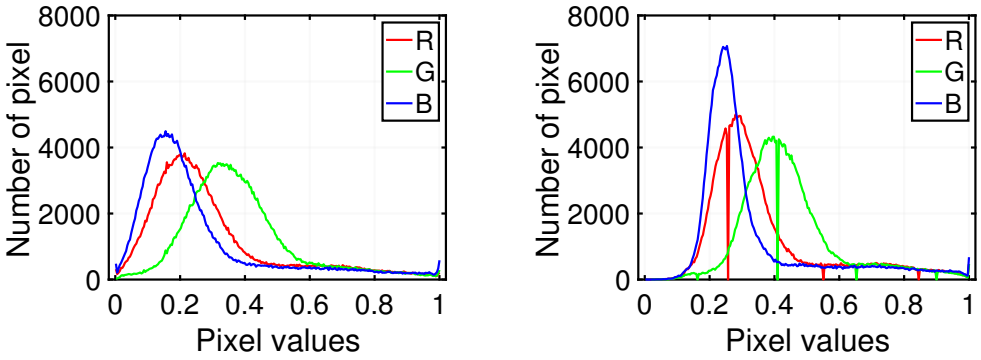
(e) DID-MDN [11]

(f) 提案法

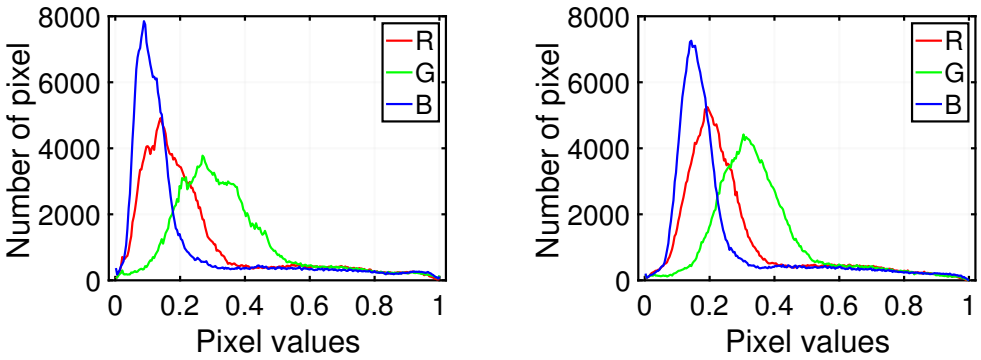
図 3.9　合成画像での雨すじ除去結果 2（Bird）



(a) 正解画像　(b) 雨画像



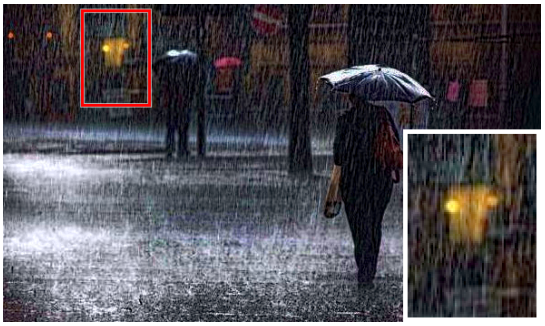
(c) DSC [6]　(d) DerainNet [9]



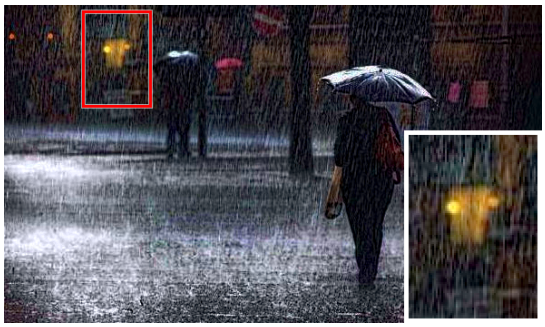
(e) DID-MDN [11]　(f) 提案法

図 3.10　合成画像での雨すじ除去結果のヒストグラム 2（Bird）

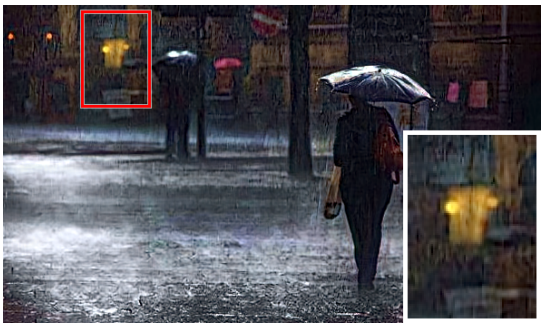
(a) 雨画像



(b) ヘイズ除去後画像

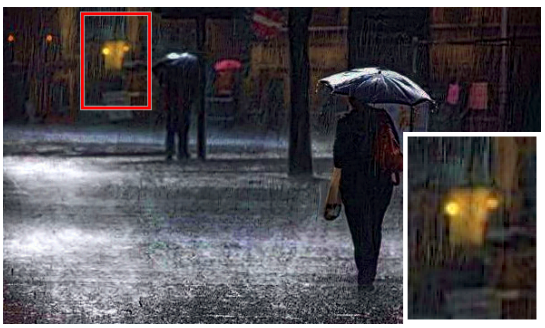


(c) DSC [6]



(d) DerainNet [9]

(e) DID-MDN [11]



(f) 提案法

図 3.11　自然画像での雨すじ除去結果 1（Street）

(a) 雨画像

(b) ヘイズ除去後画像

(c) DSC [6]

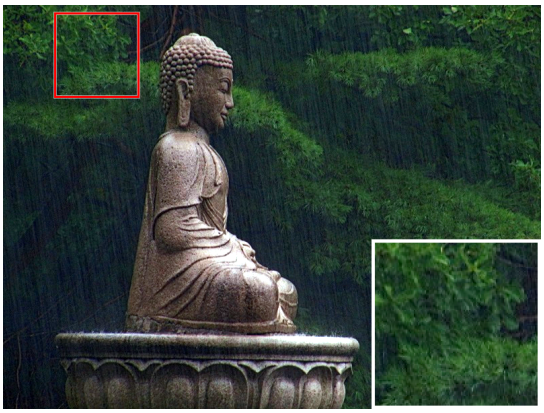
(d) DerainNet [9]

(e) DID-MDN [11]

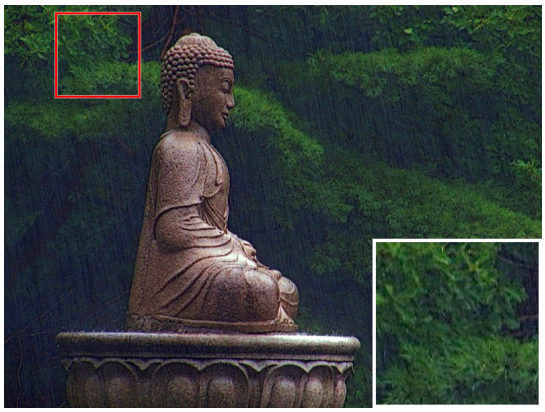
(f) 提案法

図 3.12　自然画像での雨すじ除去結果 2（Soccer）

(a) 雨画像



(b) ヘイズ除去後画像



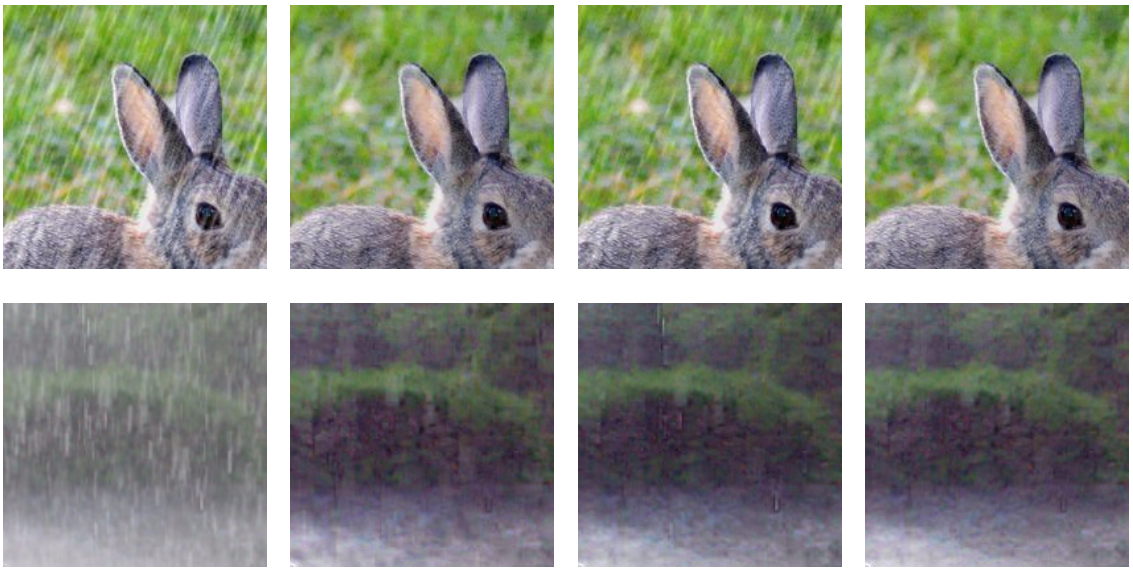
(c) DSC [6]

(d) DerainNet [9]

(e) DID-MDN [11]

(f) 提案法

図 3.13　自然画像での雨すじ除去結果 3（Buddha）



(a) 雨画像　(b) 加算合成のみ　(c) ブレンド合成のみ　(d) 提案法

図 3.14　雨合成モデルが異なるデータセットによる雨すじ除去結果の比較

第 4 章

エッジ抽出フィルタを用いた深層学習に基づくフェンス除去

4.1 フェンス除去について

本章では画像劣化のうち撮影場所が要因となり劣化した画像の復元について扱う．撮影時のノイズや障害物は画像処理の妨げとなってしまうことがしばしばある．

例えば，旅行者やプロのカメラマンは，動物園やスポーツ施設などにおいて，フェンスの存在が被写体を見えづらくして邪魔に感じてしまう．危険な場所から遠ざけるためにはフェンスを設置しなければならないので，フェンスを物理的に取り除くことはできない．フェンスの穴から撮影をすると被写体と近すぎて全体を写すことができなくなってしまう．他にも，歴史的な建造物を観光する際，しばしば工事中の足場として鉄柵が付けられていることがある．せっかく旅行をしたのにそのような観光地で写真を撮るのは気が引けてしまうこともある．このように，様々なシーンにおいてフェンスや鉄柵などを画像処理的に除去したい (De-fencing) という需要がある．もちろん Photoshop のような画像加工ソフトウェアを用いればそれらを除去することはできるが，時間と手間がかかりすぎてしまう．自然画像のフェンスは様々な形状や色のものが存在するため，フェンス除去はかなり難しい処理である．さらに，フェンスの中には規則的な構造になっていないものや壊れたり歪んだりしているものもある．これらの理由から，ロバストかつ自動でフェンス除去を行う手法が必要である．

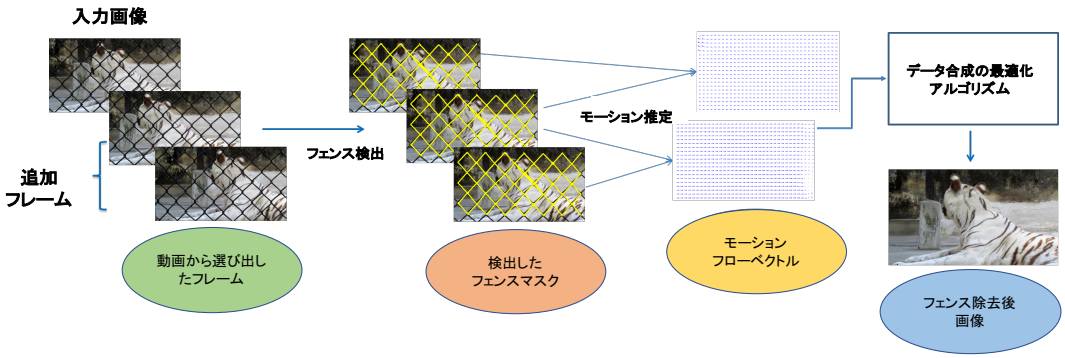


図 4.1　動画ベースの手法のフローチャート [18]

Liu [22] らが 2008 年に初めて自動でフェンス除去をするアルゴリズムを提案し，それを “De-fencing” と名付けた．彼らは画像内のフェンスは規則的なパターンを持つだろうという仮説のもとでフェンスの検出を行う．その後フェンス部分とフェンスでない部分を識別するマスクを作成し，フェンスで隠れた部分を古典的なインペインティング手法 [94] により補間する．このように，De-fencing のタスクはフェンスを検出する段階と欠損部分を修復する段階の二つの段階に分けて考えることができる．この二段階のアプローチで様々な手法が十数年間にわたって研究されている．それらの手法を三種類に分類することができると考える．一つ目は，動画ベースの手法でフレーム間情報を用いて検出と除去を行う．二つ目は，複数画像ベースの手法で，一つの被写体に対して角度や焦点などが異なる画像を合成させて行う．三つ目は，単一画像ベースの手法で，たった一枚の画像からフェンスの検出と除去を行う．動画ベースの手法は多く提案されているが，画像ベース，特に，単一画像ベースのフェンス除去手法はまだ数種類の手法しか提案されていない．その理由は，動画と違い画像は時間成分が無く，検出や欠損部分を埋める情報が少ないからである．

4.2　フェンス除去の従来法

4.2.1　動画ベースの手法

動画ベースの手法 [13–19, 95, 96] とは，複数のフレーム情報を利用してフェンスを検出し除去するというものである．例えば，手法 [13] では，Affine-SIFT [97] を用いて注目点を追跡し，全体的な背景の動きを推定する．動きの小さな動画では，あるフレー

ムでフェンスにより隠れた部分は別のフレームでは見えるようになる．手法 [14,15] では動きの小さな動画だけでなく，動きの大きなシーンを撮影した場合に対してもフェンス除去を試みている．また近年では，深層学習ベースの手法も提案されている．Jonna ら [16] や Yi ら [17]，Nakka ら [18] は単純な CNN 構造でフェンスのセグメンテーションができることを提案した．図 4.1 は Nakka らの手法のアルゴリズムである．Du ら [19] は画像領域分割でよく用いられる FCN [79](Fully Convolutional Network) を使用している．以上のように使用する CNN の構造は変わっていても，フェンスの交差点を含むパッチとフェンスの交差点を含まないパッチを分類するように学習させているという点は共通している．

最後に補足として，動画の RGB 成分に加えて深度マップを用いてフェンス検出を行う手法 [98] も存在する．深度マップがフェンスマスクの推定精度を高めてくれるというものである．

以上のように，動画ベースのフェンス除去は，特に動きの大きすぎないものに対してはかなり高精度である．しかし，動画の処理を行うため計算コストが高くなってしまうという欠点はある．

4.2.2　複数画像ベースの手法

複数画像ベースの手法 [20,21,99] とは，ある特定条件下において撮影された複数枚の画像を合成してフェンス除去画像を生成するというものである．手法 [20] では，図 4.2 のように一つの背景に対して固定カメラで 3 種類の画像を撮影する．(A) 被写体に焦点を当てた画像と (B) フェンスに焦点を当てた画像，そして (C) フェンスに焦点を当ててフラッシュを焚いた画像である．この 3 枚の画像から (D) フェンスマスクを作成する．フェンス画像がアルファ合成でモデル化できるという仮説の基で，フェンスを除去するというアルゴリズムである．一方，手法 [21] では 2 枚のステレオ画像を使う．人間の目と同様に少し離れた二方向から撮影した画像のペアである．CNN を用いて画像ペアに対応する視差マップを計算する．視差マップから 2 値マスクを生成し，TV(Total Variation) 正則化問題を解くことによってフェンス除去画像を出力するというアルゴリズムである．これらの複数画像ベースの手法は，フェンス除去以外のタスクにも応用することができる．しかし，フェンス除去をするというタスクにおいては，あらかじめ望ましい条件下で撮影をする必要があるため実用性は低いと言える．

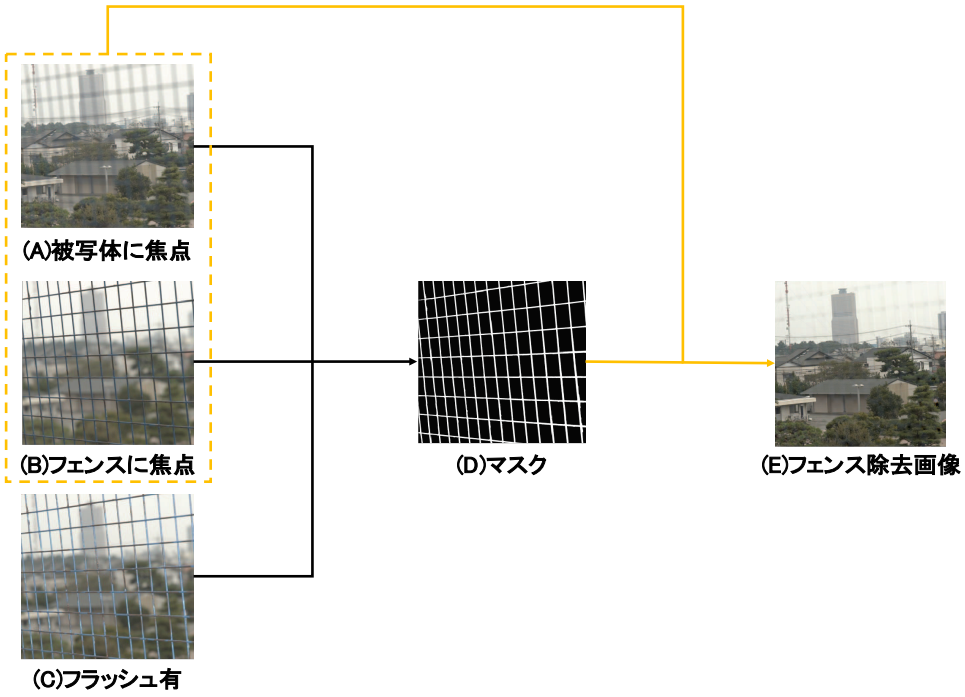


図 4.2　複数ベースの手法のフローチャート [20]

4.2.3　単一画像ベースの手法

単一画像ベースの手法 [22–26] とは，一枚の画像だけからフェンスの検出と除去を行うというものである．動画ベースの手法と違い，従来法の数が少ない．

パッチの規則性を基にした手法

Liu ら [22] の手法は，規則性のある前景を自動で見つけ出し [100]，欠損部分を Criminisi らのインペインティング手法 [94] によって補間するという流れである．図 4.3 のように，まずコーナー検出をした大量の点からフェンスの交差点に相当するものを見つけ出す．その交差点を中心とした $p \times p \times 3$ のパッチ N_p 枚を切り出して，標準偏差 σ を RGB それぞれのチャネルに対して計算する．それによってできたベクトル $[r, g, b, \sigma_r, \sigma_g, \sigma_b]^T$ を K-means クラスタリングを用いて，フェンスマスクを作成する．

このフェンス検出方法は Park ら [23] によって改良された．ベクトルを作成するときに，RGB の画素値ではなく，平均値 μ を使って K-means クラスタリングを行う．つま

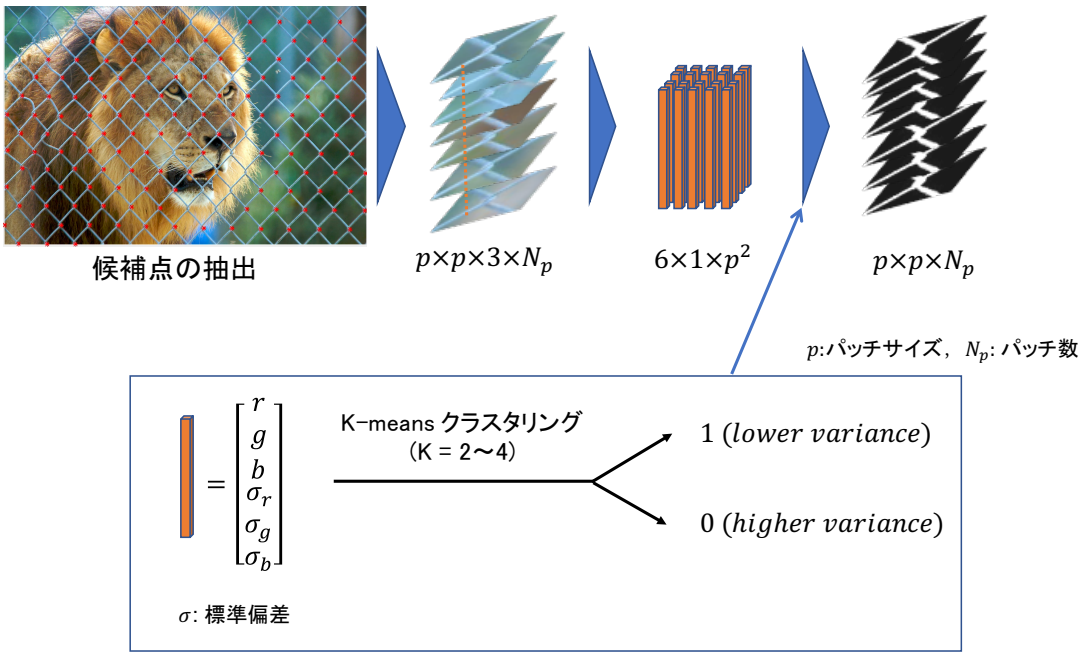


図 4.3　Liu らによる単一画像ベースのフェンスマスク生成アルゴリズム [22]

り，ベクトル $[\mu_r, \mu_g, \mu_b, \sigma_r, \sigma_g, \sigma_b]^T$ を計算しクラスタリングするということである．その後にサポートベクターマシンによってフェンスマスクの再補正を行うことによって規則性が分かりづらいようなフェンス画像にも対応できるようになった．しかしながら，Park らの手法は似た見た目のパッチを切り出すため，フェンスが歪んだり壊れたりしている時には効果を発揮しないという欠点がある．

色ベースの手法

そこで，Farid ら [24] が色ベースのフェンス検出アルゴリズムとハイブリットインペインティングアルゴリズムを提案してこの問題を解決した．まず，ユーザが画像からフェンスに相当する点を n 箇所（1 0〜2 0）入力する．その n 個の点とその k 近傍を集めたサイズ $n(k+1) \times 3$ の行列 $\mathscr{P}$ を考える．RGB のチャネルごとに平均 μ と共分散 Σ を以下のように計算する．

$$\mu_c = \frac{1}{n(k+1)} \sum_{i=1}^{n(k+1)} \mathscr{P}(i,c), \quad c = \{r,g,b\} \tag{4.1}$$

$$\Sigma = \begin{pmatrix} \sigma_{(r,r)} & \sigma_{(r,g)} & \sigma_{(r,b)} \\ \sigma_{(g,r)} & \sigma_{(b,g)} & \sigma_{(g,b)} \\ \sigma_{(b,r)} & \sigma_{(b,g)} & \sigma_{(b,b)} \end{pmatrix} \tag{4.2}$$

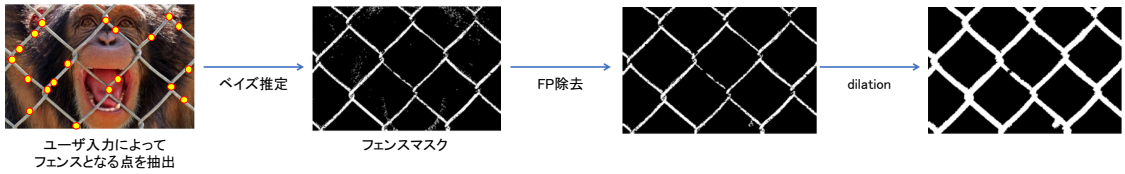


図 4.4　Farid らによる単一画像ベースのフェンスマスク生成アルゴリズム [24]

ただし，$\sigma_{(\alpha,\beta)}$ は

$$\sigma_{(\alpha,\beta)} = \frac{1}{n(k+1)} \sum_{i=1}^{n(k+1)} (\mathscr{P}(i,\alpha) - \mu_\alpha)(\mathscr{P}(i,\beta) - \mu_\beta) \tag{4.3}$$

により求められる．これらに従うガウシアン分布を想定し，ベイズの定理を用いることによって，フェンスマスクを作成する．画像 $\boldsymbol{x}$ の各画素がフェンスに含まれる確率は以下の式で求められる．

$$P'(\mu_f, \Sigma_f|\boldsymbol{x}) = \frac{1}{\sqrt{|\Sigma_f|}} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}-\mu)^T \Sigma_f^{-1}(\boldsymbol{x}-\mu)} p_f \tag{4.4}$$

同様に，フェンスに含まれない確率は

$$P'(\mu_n f, \Sigma_n f|\boldsymbol{x}) = \frac{1}{\sqrt{|\Sigma_n f|}} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}-\mu)^T \Sigma_{nf}^{-1}(\boldsymbol{x}-\mu)} p_{nf} \tag{4.5}$$

のように書ける．以上から得られたフェンスマスクは最終出力ではない．そこで，False Positive(FP) を除去するため，マスク内で一定の点群が繋がってない箇所を削除する．さらに False Negative(FN) を除去するため，モルフォロジー膨張を行う．これによって完成したフェンスマスクを用いてインペインティングを行う．この際，画像をアップサンプリングすることによってサイズの異なる画像 4 種類に対してそれぞれインペインティングを行い，最後にそれらを結合するというハイブリッドインペインティングをしている．

従来法の弱点を克服したが，ユーザ入力が必要でかなり手間がかかるという問題点を残している．さらに，画像の中にフェンスの色と似た色の被写体があるとうまくフェンスを検出できないという問題もある．

4.3　従来法の問題点と提案法における改良点

第 4.2 節で紹介した従来手法の概要とそれぞれの長所と短所を表 4.1 にまとめた．本節では単一画像ベースのフェンス除去手法における問題点を述べる．まず，Liu ら [22]

表 4.1　フェンス除去手法のまとめ

	動画ベース	画像ベース				
		複数画像	単一画像			
方法	[13–19, 95, 96, 98]	[20, 21, 99]	Liu ら [22]	Park ら [23]	Farid ら [24]	提案法
概要	フレーム間情報から補間を行い合成	複数の画像から補間を行い合成	特徴点抽出と教師なし学習により検出	オンライン学習でフェンス検出	ユーザ入力をもとに似た色をフェンスと認識	CNN と空間フィルタリングにより検出と除去
長所	* フレーム間情報を使うので精度は高い	* 他分野へ応用可能	* 規則的なフェンスで検出から除去まで一貫処理	* 準規則的なフェンスでも検出から除去まで一貫処理	* 不規則な形状のフェンスでも検出と除去が可能	* フェンスの色や形によらずフェンス検出 * 自然な見た目
短所	* 動画のみに対応 * 高計算コスト	* 撮影に手間がかかる	* 歪んだ，不規則なフェンスはできない * 高計算コスト	* 不規則なフェンスはできない * 高計算コスト	* ユーザ入力にコツがいる * フェンスと背景の色を見分けられない	* 一定の角度や形には弱い

と Park ら [23] の手法は，規則性をもとにフェンスを検出するため，フェンスが歪んでいたり，画像内にフェンスの一部しか写っていなかったりすると検出がうまくいかない．また，フェンス検出が不完全だと最終出力であるインペインティング後の画像も崩れてしまう．さらに，全てパッチ単位のローカル処理を行なっているので，計算コストが高くなってしまうという問題点がある．一方，Farid ら [24] は色ベースの手法を提案したが，ユーザ入力が必要でコツがいるというのが最大の欠点であると言える．加えて，フェンスの色と背景の色が似ているとそれらの区別ができず，検出に失敗してしまうという問題点がある．従来手法の問題点を解決するため，以下の方向で提案手法を考えた．

- フェンスの形状や色，画像に依存しない高精度な除去手法
- ユーザ入力なしでフェンス検出と除去を実現
- 特にフェンス検出に注力する
- 計算コストを抑える

自然画像からフェンスの特徴を捉えるため，ディープネットワークによる学習ベースの手法を提案する．フェンス検出とフェンス除去を別問題として扱い，それぞれの問題で最適なネットワーク構造を用いる．それだけでなく，古典的な画像処理のフィルタリ

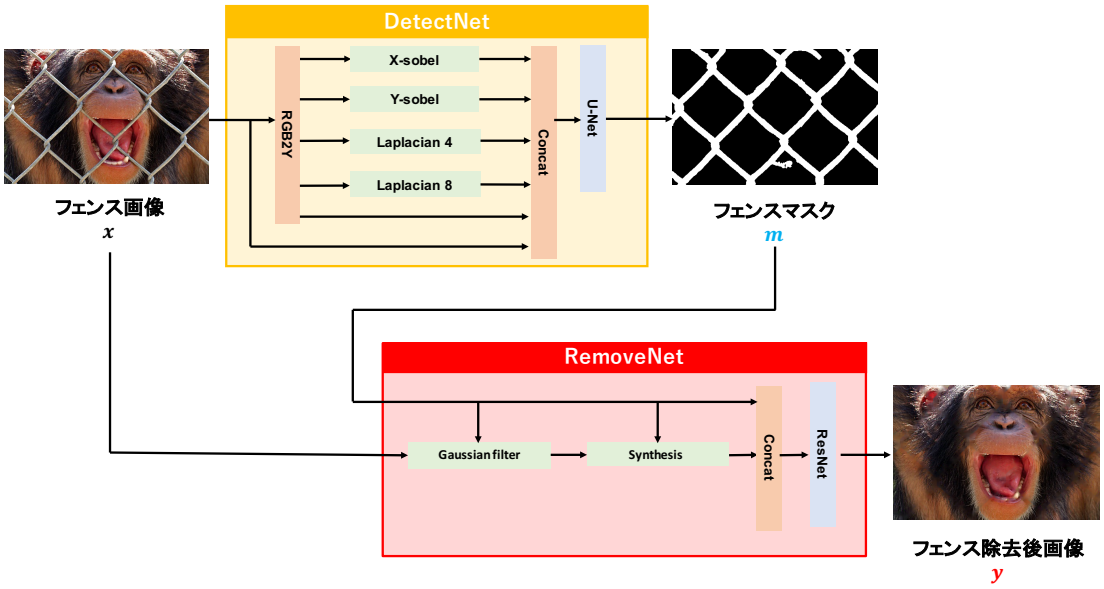
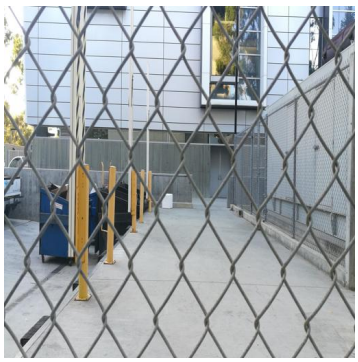


図 4.5　DefenceNet のフローチャート

ングを用いることによってディープネットワークの過学習を防ぎ，様々なフェンスに対応可能なアルゴリズムを提案する．提案手法の特徴をまとめると以下のようになる．

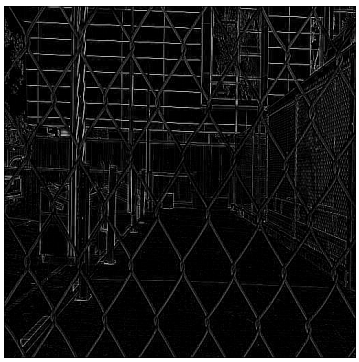
1) ディープネットワークと古典的な画像処理の知見を融合させることで，一枚の画像から全自動でフェンスの検出と除去を可能にする．
2) フェンス検出タスクは分類問題としてではなく回帰問題として扱うことで，不規則なフェンスの形でもフェンスを検出することができるようにする．
3) フェンス補間では，ネットワークを学習させるために合成フェンス画像を作成することで，様々なフェンス画像に対してロバストにさせる．

提案法である DefenceNet は図 4.5 のように二段階に分けて処理される．第一段階では，入力画像 $\boldsymbol{x}$ から 2 値フェンスマスク $\boldsymbol{m}$ を生成する．第二段階では，フェンス画像 $\boldsymbol{x}$ と先の検出ネットで推定したフェンスマスク $\boldsymbol{m}$ から，フェンス除去後画像 $\boldsymbol{y}$ を得る．



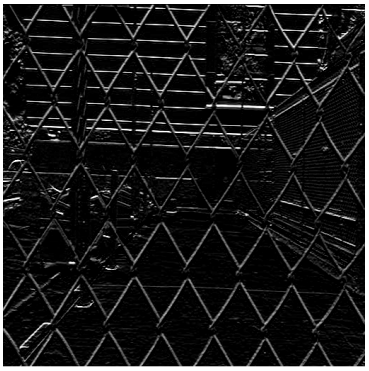
(a) RGB 画像

(b) 水平ソーベルフィルタ

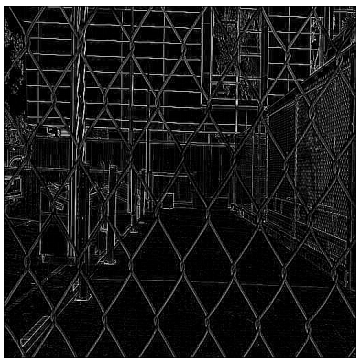


(c) 4 近傍ラプラシアンフィルタ

(d) Y 画像



(e) 垂直ソーベルフィルタ



(f) 8 近傍ラプラシアンフィルタ

図 4.6　フェンス検出パート（U-Net）の入力に用いる前処理後画像群

4.4　提案法（DefenceNet）の検出パート

4.4.1　ネットワーク構造

フェンス検出ネットは図 4.5 上部の黄色い部分（DetectNet）である．提案法のフェンス検出ネットワークは U-Net の構造 [81] を基にしている．しかしながら，U-Net に直接フェンス画像を入力してもうまくいかず，フェンス検出に適したものにするため，以下の手順にしたがって入力データと出力データを工夫した．

4.4.2　前処理

フェンス検出手法は二つに大別することができる．一つ目は，フェンスの金網の色を使う手法である．フェンスの色は画像内で似た色になることを利用し，似た色同士でグルーピングを行い，色パターンからフェンス部分とそうでない部分を分類する．しかし，フェンスの色と背景色が似ている場合などにおいては，フェンスとしての特徴をうまく捉えきれないという問題点がある．二つ目は，入力画像から規則的に繰り返されるパターンを認識するという手法である．コーナー検出により抽出した特徴点から，一定の距離にあり似た特徴のある点をフェンスの交差点であるとして集めてくるというものである．この規則性を基にしたアプローチは一定の効果があるが，規則性が少ないフェンスや歪んだフェンスに対しては効果を発揮しないという問題点がある．以上から，フェンス検出タスクは「フェンスは周りと比べて勾配が大きい」という局所的特徴と，「規則性のあるパターンを繰り返す」という大局的特徴の両方を捉える必要があるということがわかる．U-Net は画像をそのまま畳み込む部分とアップサンプリングする部分の両方を持つので，先に述べたような局所的特徴と大局的特徴の両方を抽出することができる．しかし，U-Net に直接フェンス画像を入力すると過学習を起こしてしまう．そこで，旧来より使われる画像処理の知見と融合するべきだと考えた．まず，以下の式で表される画像の輝度成分（Y チャネル）を入力（RGB）に加えることでフェンス検出ネットが色に大きく依存することを避ける．

$$f_Y(\boldsymbol{x}) = (0.299 \quad 0.587 \quad 0.114)\,\boldsymbol{x} \tag{4.6}$$

ここで，$\boldsymbol{x}$ は RGB フェンス画像を列ベクトルとして記述したものとする．次に，エッジ検出フィルタであるソーベルフィルタとラプラシアンフィルタを掛けた画像を入力し，U-Net がフェンスの大きさや形や色に対してロバストになるようにする．まとめると，RGB-Y 成分とエッジ検出後画像の合計 8 チャネルが U-Net に入力されることになる．図 4.6 に U-Net の入力となる画像の例を示す．

4.4.3　後処理

学習データやパラメータを工夫しても，自然画像に対してのフェンス検出は不完全である．例えば，図 4.8(a) のようにフェンスでない部分をフェンスであると認識してしまうこと（FP) や，フェンスである部分をフェンスでないと認識してしまうこと（FN）がある．これらは後のフェンス除去結果に大きく影響を与えてしまうため，後処理として

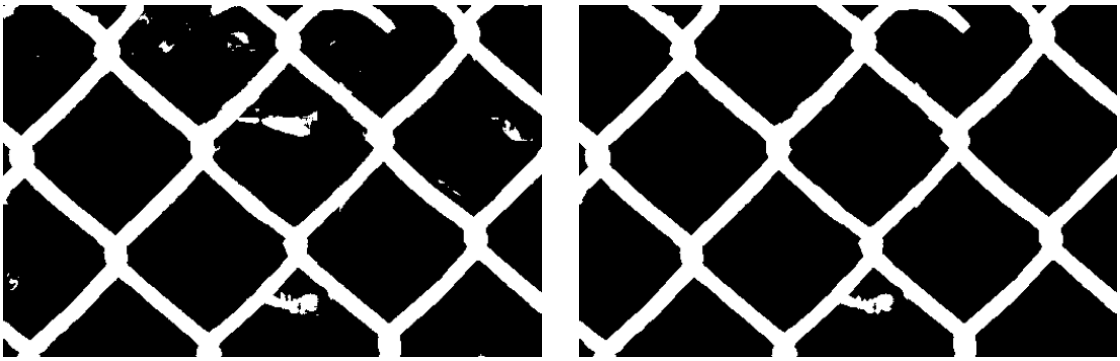
(a) フェンスパッチ　　(b) マスクパッチ

図 4.7　フェンス検出パート（U-Net）の学習データサンプル

FP と FN を取り除く必要がある．具体的には，モルフォロジー変換を用いる．まず，バイナリ画像を膨張させる．近傍のピクセルの中に値 1 のものが存在すると 1 になるような演算を行う． モルフォロジー膨張により，オブジェクト内の小さな穴が埋まる，つまり，FN が消えることになる．次に，モルフォロジーオープニングを行う．フェンスは画像内で繋がっているという仮説のもと，一定画素数以上繋がっていない部分を削除する．これによりフェンスでない部分（FP）を取り除くことができる．最後に，モルフォロジー膨張で太くなった分を細くするため，モルフォロジー収縮を行い，最終的なフェンスマスクが生成される（図 4.8(b)）．

4.4.4　学習データセット

U-Net の学習データセットとして，Du ら [19] の自然フェンス画像から 545 枚を集め，$128 \times 128 \times 3$ のパッチに切り分けた．さらに学習データをフェンスの向きや形状に対してロバストにするため，パッチをランダムに反転，回転，拡大縮小，輝度変化させて合計 27088 枚のパッチを学習に用いた．さらに，より多くの種類のフェンス画像に対応できるようにするため，学習データセットはフェンス画像をパッチとして切り出し，反転，回転，拡大縮小，輝度変化をしてデータの拡張を行った．学習に用いたパッチのサンプルを図 4.7 に示す．



(a) 処理前フェンスマスク　　(b) 処理後フェンスマスク

図 4.8　モルフォロジー変換によるバイナリーマスクの比較

4.4.5　学習パラメータ

DefenceNet の学習はそれぞれ独立させて学習を行った．フレームワークは Caffe を用いた．学習率は $\alpha_0 = 0.001$ から始め，$\alpha(t) = \alpha_0(1 + \gamma t)^p$ に従って減衰するようにさせた．ここで，$\gamma = 0.0001$，$p = 0.75$ とした．

フェンス検出ネットの学習には，RGB-Y 画像とエッジ検出フィルタ処理された画像の合計 8 チャネルを入力させた．局所的特徴と大局的特徴の両方を捉えるため，畳込み後の特徴とダウンサンプルした特徴を結合させる．そこで，ダウンサンプリングは 2×2 のマックスプーリングフィルタを用いた．畳込み層のカーネルサイズは 3×3 とし，重みは Xavier [101] で初期化した．学習時間短縮のため，バッチサイズ 8 で学習を行い，10 万回のイテレーションに 2 時間程度かかった．

4.4.6　損失関数

ここ数年で様々な CNN ベースのフェンス検出手法が提案されている．機械学習における教師あり学習は「分類」と「回帰」の二つに分けることができる．分類タスクは出力の数が限られている時に使われることが多い．例えば，画像のフェンスを認識するというタスクにおいて，フェンス部分を「1」，フェンスでない部分を「0」とするものである．多くの従来法ではフェンスの X の形の交差点部分を中心としたパッチをポジティブデータ「1」と定義し，交差点が中心にないパッチをネガティブデータ「0」と定義して学習させている．このアプローチはフェンスが歪んでいる時には十分ではない．した

がって，このフェンス検出を「分類」問題ではなく「回帰」問題として解くことにした．つまり，フェンスの交差点がパッチ内に 1 個以上あろうとなかろうと関係なく，パッチ内のどの部分がフェンスに相当するかを領域分割させるということである．ネットの出力は 0 から 1 の間の連続値をとるので，閾値からの大小により 2 値化してフェンスマスク $\boldsymbol{m}$ を推定する．フェンス検出ネットのパラメータ $\boldsymbol{\Theta}_D$ は以下の損失関数 $E_D(\boldsymbol{\Theta_D})$ を最小とするように学習させた．

$$E_D(\boldsymbol{\Theta_D}) = \frac{1}{2N}\sum_{n=1}^{N} \|\boldsymbol{m}_n - f_D(\boldsymbol{x}_n; \boldsymbol{\Theta}_D)\|_2^2 \tag{4.7}$$

ただし，n は画像のインデックスであり，N は合計パッチ数である

4.5 提案法（DefenceNet）の除去パート

フェンスというのは画像全体に広がっており，背景の大事な部分を覆ってしまうので，欠損部分の復元（インペインティング）はかなり難しいタスクである．本論文では，ResNet [83] にデータを入力する前に前処理を行う．

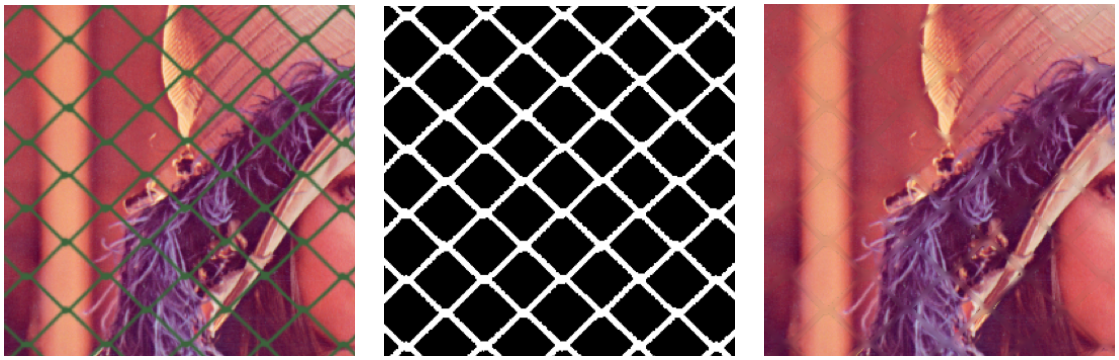
4.5.1 ネットワーク構造

最終フェンス除去後画像は，学習済みの ResNet に前処理後画像を入力した時の出力から得ることができる．ここでネットワーク構造として ResNet を用いたのは，前処理後の画像は周囲画像から推定した低周波成分であり，入力と出力の残差を学習する ResNet がその高周波成分を補うのに最適なのではないかという仮説に基づいている．

4.5.2 前処理

フェンスにより隠れた部分を埋めるために，周囲の画素情報を基に推測する必要がある．そこで，単純なインペインティング手法であるガウシアンフィルタを用いる．具体的には，推測したい画素 (i_c, j_c) を中心とした 11×11 の窓に対して，標準偏差 σ のガウシアン分布に従う重みで加重平均を取り，その値と対象画素を置き換える．その際，窓内のフェンスに相当する部分はマスクを用いることで重みを 0 にして計算する．画素 (i, j) におけるガウシアンカーネル $g(i, j)$ は以下の式で表される．

$$g(i,j) = \frac{1}{2\pi\sigma}\mathrm{e}^{-\frac{(i-i_c)^2+(j-j_c)^2}{2\sigma^2}}\left(1 - m(i,j)\right), \tag{4.8}$$



(a) フェンス画像 $\boldsymbol{x}$　(b) フェンスマスク $\boldsymbol{m}$　(c) フィルタ処理後画像 $\hat{\boldsymbol{x}}$

図 4.9　ガウシアンフィルタによる前処理画像

ここで，標準偏差は実験的に $\sigma = 2$ とした．ガウシアンフィルタ $\boldsymbol{\mathcal{F}_g}$ を用いることによって，復元後画像 $\hat{\boldsymbol{x}}$ は式 4.9 のように書くことができる．

$$\hat{\boldsymbol{x}} = \boldsymbol{x} \circ (\boldsymbol{1} - \boldsymbol{m}) + \boldsymbol{\mathcal{F}_g x} \circ \boldsymbol{m} \tag{4.9}$$

ただし，$\circ$ は要素ごとに積をとる演算子である．これにより得られる画像の例を図 4.9 に示す．この時点でかなり復元できているが，高周波成分が失われており，跡が残って見える．

4.5.3　学習データセット

ネットワークを学習する際，自然画像のフェンス画像とそれに対応するフェンスなし画像を入手するのは難しい．そこで，式 (4.10) を用い学習データセットに用いる合成画像を生成した．

$$\boldsymbol{x} = \boldsymbol{y} \circ (\boldsymbol{1} - \boldsymbol{m}) + (\boldsymbol{c} + \boldsymbol{n}) \circ \boldsymbol{m}. \tag{4.10}$$

ここで，$\boldsymbol{c}$ はフェンスの色に相当する画素値を表す．ダークグレー，ライトグレー，ダークグリーン，ライトグリーン，茶色と白色の 6 種類色を使用した．色に対してロバストにするため，色付けされたフェンスに対してガウシアンノイズ $\boldsymbol{n}$ を加算した．学習に用いたデータセットのサンプルは図 4.10 に示す．

ResNet の学習には $128 \times 128 \times 3 \times 30944$ 枚のパッチを学習させた．フェンス画像とそれに対応するフェンス無し画像を手に入れるのは困難なので，UCID データセット [102] と BSD データセット [103] から集めた 900 枚の画像 [9] に対して，Du ら [19] のフェンスマスクを合成させることによってデータセットを作成した．

(a) 原画像パッチ　(b) フェンスパッチ　(c) マスクパッチ

図 4.10　フェンス除去パート（ResNet）の学習データサンプル

4.5.4　学習パラメータ

フレームワークとして Caffe を用い，学習率は $\alpha_0 = 0.001$ から始め，$\alpha(t) = \alpha_0(1+\gamma t)^p$ に従って減衰するようにさせた．ここで，$\gamma = 0.0001$，$p = 0.75$ とした．

フェンス除去ネットの学習には，ガウシアンフィルタ処理された RGB 画像とフェンスマスクの合計 4 チャネルを入力させた．ResNet の層数は $L = 20$ で，3×3 の畳み込み層で重みは Xavier [101] で初期化した．バッチサイズは 12 とし，2 時間かけて 1 0 万回学習させた．

4.5.5　損失関数

提案法のフェンス除去ネットの損失関数 $E_R(\boldsymbol{\Theta_R})$ は以下のように定義できる．

$$E_R(\boldsymbol{\Theta_R}) = \frac{1}{2N}\sum_{n=1}^{N} \|\boldsymbol{y}_n - f_R(\hat{\boldsymbol{x}}_n, \boldsymbol{m}_n; \boldsymbol{\Theta}_R)\|_2^2, \tag{4.11}$$

ただし，$\boldsymbol{\Theta}_R$ は重みとバイアスを含む学習パラメータである．

表 4.2　フェンス検出と除去における比較手法と評価方法のまとめ

比較手法	目的	コード	交差点数	主観的評価	PSNR/SSIM
提案法	検出 + 除去	有	○	○	○
CVPR’08 [22]	検出 + 除去 [94]	無	-	○	-
ACCV’10 [23]	検出 + 除去 [94]	無	-	○	-
ISIVP’16 [24]	検出 + 除去	有	○	○	○
TPAMI’09 [104]	検出のみ	有	○	○	-
TIP’04 [94]	除去のみ	有	-	○	○

4.6　フェンス除去の実験と比較

4.6.1　実験内容

提案法である DefenceNet のフェンス検出・フェンス除去性能をそれぞれ評価する．そのために，いくつかの自然画像と合成画像に対してフェンス検出・フェンス除去を行い，従来法とその精度を比較する．

4.6.2　比較対象

表 4.2 に示したように，フェンス検出性能は交差点数と見た目の比較を [24, 104] と比較した．フェンス除去性能に関しては，PSNR と SSIM の客観的評価は [24, 94] と行い，見た目の比較は [22–24] と行った．

4.6.3　評価手法

フェンス検出とフェンス除去でそれぞれ客観的評価と主観的評価を行う．

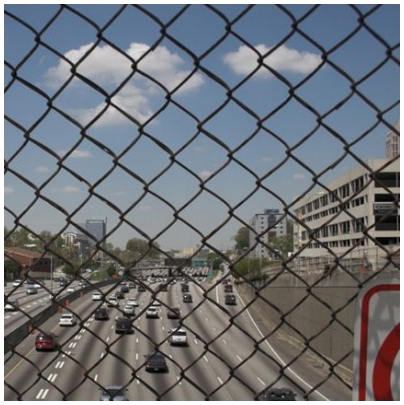
フェンス検出　フェンス検出において客観的評価を行うことは難しい．そこで，フェンスの交差点の数の検出率で客観的評価を行う．Liu らが提供 [22, 104] している NRT (Near-Regular Texture) データセットから 28 枚の自然画像と，インターネット等から収集した 9 枚の自然フェンス画像を用いる．合計 37 枚のフェンス画像にある交差点の数を数え，各手法でどれだけ交差点を検出できたかを割り算によって求める．したがって最小値は 0 で最大値は 1 となる．主観的評価としては，フェンスマスクの見た目で比較を行う．

フェンス除去　次に，フェンス除去の客観的比較は PSNR と SSIM により比較をする．BSD100 の画像に対して Du らのフェンスマスク [19] を合成させ，それに対するそれぞれのフェンス除去性能を比べる．主観的評価としては，インペインティングの見た目を見て行う．

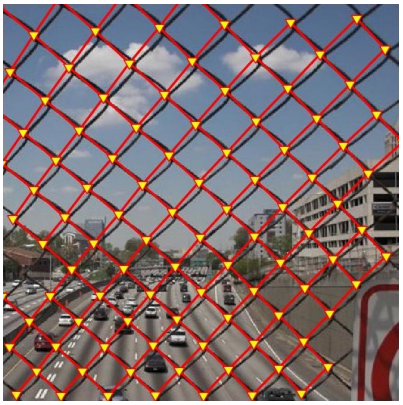
4.6.4　実験結果

フェンス検出

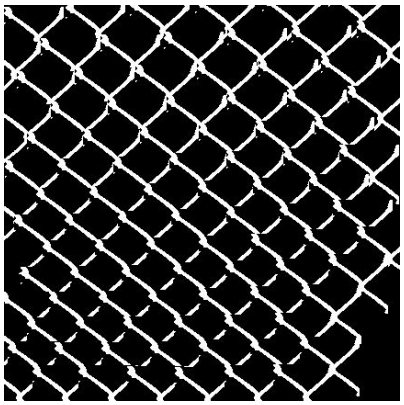
まず，フェンス検出の見た目を比較した結果を図 4.11，図 4.12，図 4.13 に示す．Park らの手法 [104] では画像から規則性のあるパッチを集めてくるので，図 4.11 の “Road” のようにフェンス間隔が一定で背景の変化が少ない画像ではうまくフェンス検出ができる．しかし，図 4.12 の “Lion” や図 4.13 の “Prefab” のようにフェンスが歪んでいる画像や，背景の変化が大きい画像に対してはフェンスを検出しきれていないことがわかる．一方で，Farid らの手法 [24] は，色ベースのフェンス検出をしているので，フェンス検出の取りこぼしは比較的少ない．しかし，例えば図 4.13 の “Prefab” のようにフェンスとプレハブの屋根の色が似ているため，両方ともフェンスであると誤検出してしまっているのがわかる．提案手法は CNN と空間フィルタを組み合わせた手法であるため，色や形を総合的に判断してフェンスを検出する．よって，それぞれの画像においてきれいにフェンスを検出できていることがわかる．



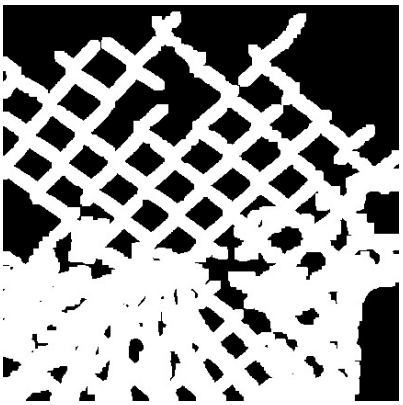
(a) フェンス画像



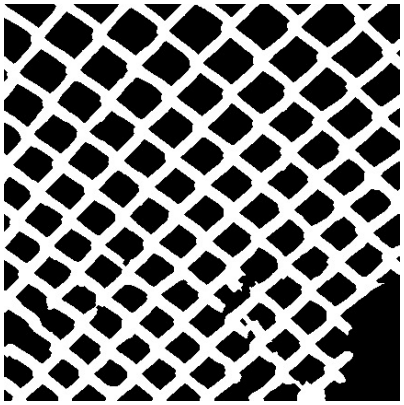
(b) Park らのフェンス検出 [104]



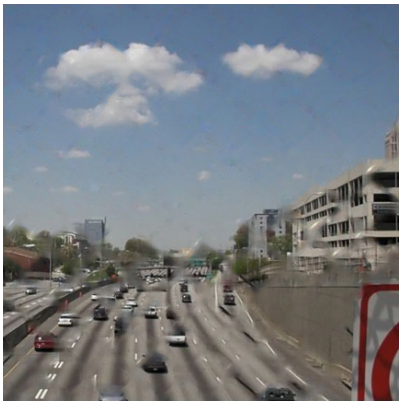
(c) Park らの推定マスク [104]



(d) Farid らの推定マスク [24]



(e) 提案法の推定マスク

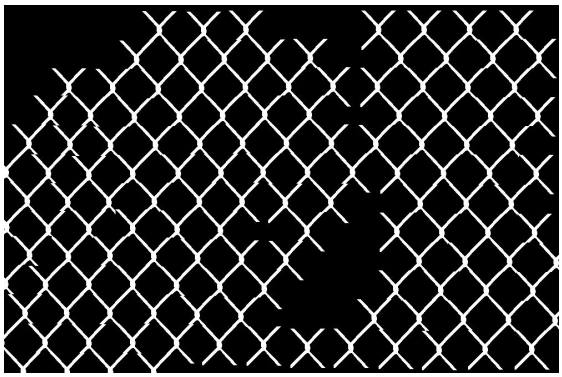


(f) 提案法のフェンス除去後画像

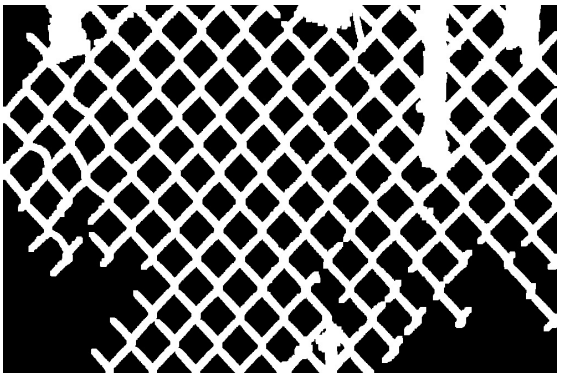
図 4.11　自然画像に対するフェンス検出結果（Road）

(a) フェンス画像

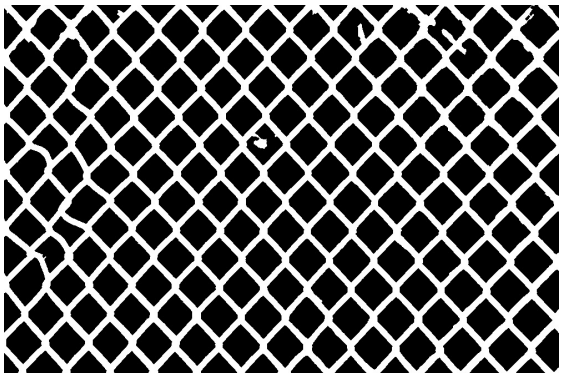
(b) Park らのフェンス検出 [104]



(c) Park らの推定マスク [104]



(d) Farid らの推定マスク [24]



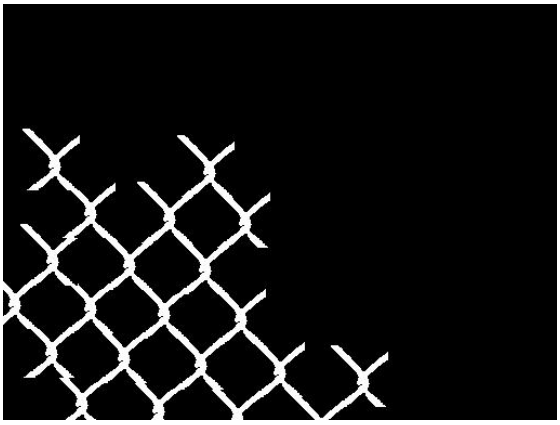
(e) 提案法の推定マスク

(f) 提案法のフェンス除去後画像

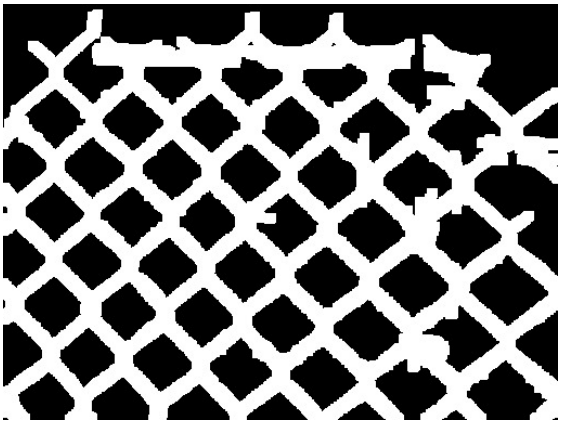
図 4.12　自然画像に対するフェンス検出結果（Lion）

(a) フェンス画像

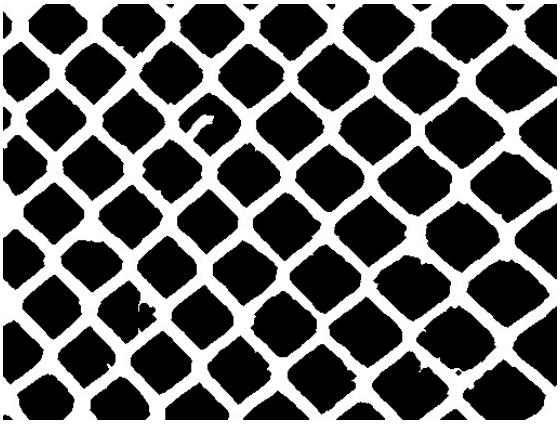
(b) Park らのフェンス検出 [104]



(c) Park らの推定マスク [104]



(d) Farid らの推定マスク [24]



(e) 提案法の推定マスク

(f) 提案法のフェンス除去後画像

図 4.13　自然画像に対するフェンス検出結果（Prefab）

表 4.3　フェンス交差点の数の検出率比較

手法	NRT	Others
Park ら [104]	0.756	0.277
Farid ら [24]	0.737	0.571
提案法	0.837	0.898

次に，客観的評価として検出したフェンスの交差点の数を 0 から 1 の割合で比較する．その結果を表したのが表 4.3 である．表から分かるように，NRT データセットに対しては従来法も提案法も検出率が 7 割を超えている．しかし，NRT データセットに含まれない画像で実験をしてみると，特に Park らの手法 [104] は検出率が大幅に下がっていることがわかる．一方で提案手法はいずれのデータセットにおいても 8 割以上の高い検出率を出しているということが実験から示された．以上から，フェンス検出というタスクにおいて，客観的にも主観的にも提案手法は従来手法を上回っていることが示された．

フェンス除去

フェンス除去の結果を三つの従来手法 [22–24] と比較した．そのうちの二つ [22, 23] は 4.2.3 節で述べたような手順でフェンス検出を行い，Criminisi らのインペインティング手法 [94] を使って最終出力を得ている．一方で，Farid ら [24] は画像をダウンサンプリングした後に結合するというハイブリッドインペインティング手法によって，最終フェンス除去画像を生成している．

見た目による比較結果を図 4.14 と図 4.15 に示す．Liu らの手法 [22] は，図 4.15 の “Duck” においてフェンス検出がうまくいっていないため，インペインティング後の画像が潰れてしまっている．Park らの手法 [23] はいずれの画像においてもかなりうまくフェンス除去ができていることが分かる．ただ，図 4.15 の “Duck” の右上で芝の部分に羽が滲み出てしまっている．Farid らの手法 [24] はフェンスを少しだけ残してしまっている．それに対して，提案手法はフェンスを残すことなく除去できているということが分かる．

その他の画像に対して，Farid らの手法 [24] とフェンス検出・除去結果を比較したものを図 4.16，図 4.17，図 4.18，図 4.19 に示す．図 4.16 の “Chimpanzee” と図 4.17 の “House” では，フェンス検出はうまくできている．しかし，従来手法のインペインティ

ングは多少滑らかではないように見える．実際のフェンスはこれらのように規則的な形をしていないこともある．例えば，図 4.18 の “Warning” のようにフェンスの前にポスタが掛かっているような場合である．Farid らの手法 [24] は色ベースの検出を行っているので，建物とフェンスの色の違いを見分けられていない．一方で提案手法では，建物の部分のフェンスもしっかり検出しているだけでなく，ポスタを避けて検出しているということが分かる．次に，図 4.19 の “Garden” のようにフェンスが特殊な形状をしている場合もあり得る．従来手法は色ベースのアプローチにもかかわらず，光の当たり具合からフェンスを全て検出できていない．それに対し提案手法は，背景との差分をうまく認識してフェンスを検出できていることが分かる．これらの結果から，提案手法のデータ拡張は，様々なイレギュラーな状況下にも対応し得るということが示された．

(a) フェンス画像

(b) Liu らのフェンス除去後画像 [22]

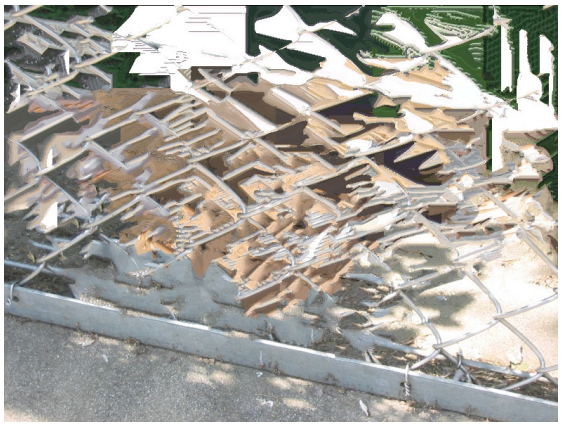
(c) Park らのフェンス除去後画像 [23]

(d) Farid らのフェンス除去後画像 [24]

(e) 提案法のフェンス除去後画像

図 4.14　自然画像に対するフェンス除去結果（Bird）

(a) フェンス画像



(b) Liu らのフェンス除去後画像 [22]

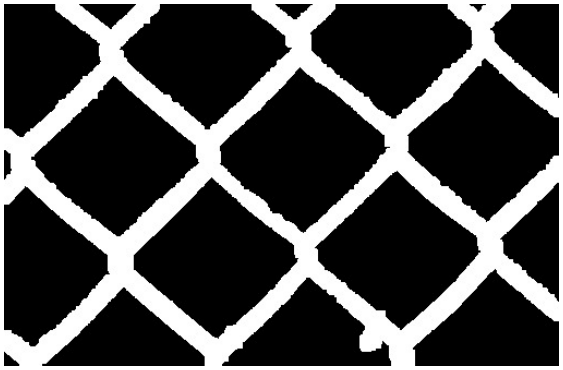
(c) Park らのフェンス除去後画像 [23]

(d) Farid らのフェンス除去後画像 [24]

(e) 提案法のフェンス除去後画像

図 4.15　自然画像に対するフェンス除去結果（Duck）

(a) フェンス画像



(b) Farid らの推定マスク [24]

(c) Farid らのフェンス除去後画像 [24]

(d) 提案法の推定マスク

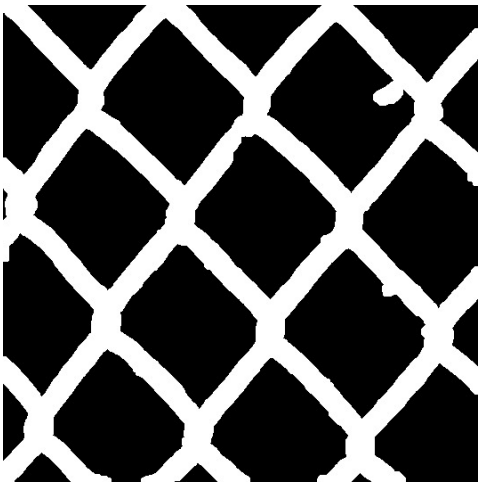
(e) 提案法のフェンス除去後画像

図 4.16　自然画像に対するフェンスマスクとフェンス除去の結果（Chimpanzee）

(a) フェンス画像

(b) Farid らの推定マスク [24]

(c) Farid らのフェンス除去後画像 [24]



(d) 提案法の推定マスク

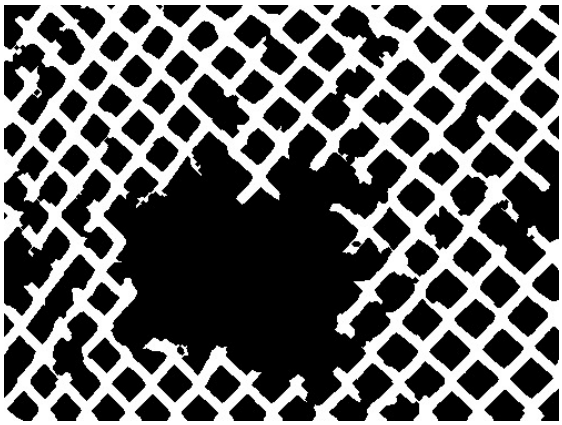
(e) 提案法のフェンス除去後画像

図 4.17　自然画像に対するフェンスマスクとフェンス除去の結果（House）

(a) フェンス画像

(b) Farid らの推定マスク [24]

(c) Farid らの手法 [24]



(d) 提案法の推定フェンス

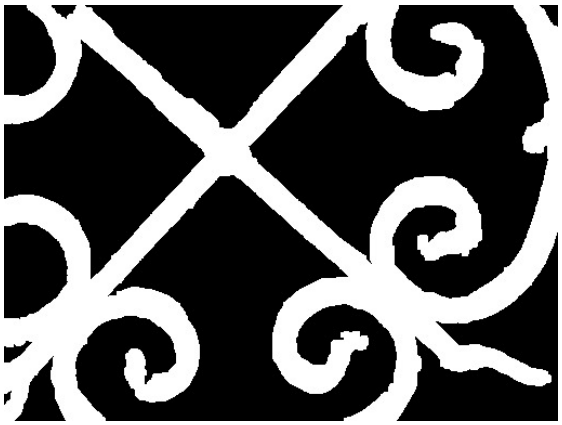
(e) 提案法のフェンス除去後画像

図 4.18　自然画像に対するフェンスマスクとフェンス除去の結果（Warning）

(a) フェンス画像

(b) Farid らの推定マスク [104]

(c) Farid らのフェンス除去後画像 [24]



(d) 提案法の推定マスク

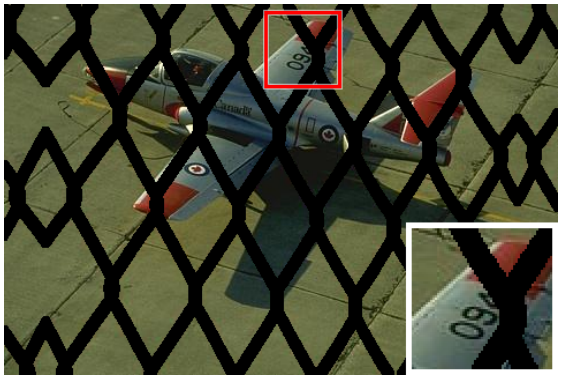
(e) 提案法のフェンス除去後画像

図 4.19　自然画像に対するフェンスマスクとフェンス除去の結果（Garden）

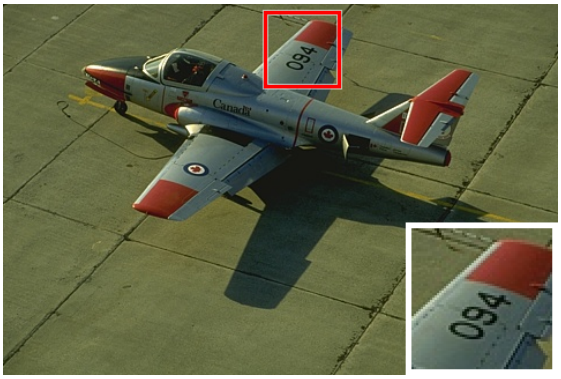
表 4.4　フェンス除去後画像の PSNR と SSIM の比較

手法	PSNR↑	SSIM↑
Criminisi *et al.* [94]	22.72	0.847
Farid *et al.* [24]	23.22	0.856
提案法	27.11	0.906

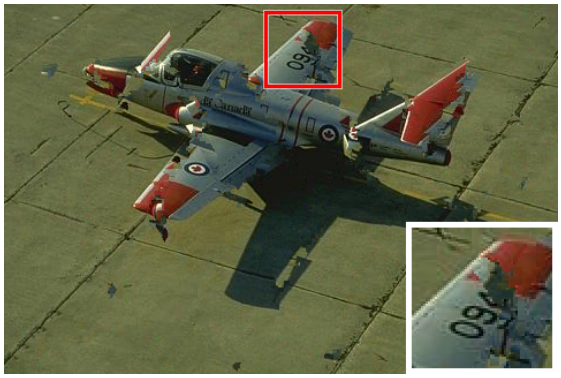
次に，合成フェンス画像に対するインペインティングの結果を二つの従来法 [24,94] と比較する．BSD100 のデータセットに含まれる 100 枚の画像に対して PSNR と SSIM を計算して比較したものを表 4.4 に示す．表から分かるように，提案手法は PSNR が 4 dB 程度，SSIM が 0.05 程度も上回っている．実際の画像で比較してみると，図 4.20 では飛行機の羽の部分で修復結果に大きな差が見て取れる．また，図 4.21 では，従来手法において顔と帽子が潰れてしまっているのに対し，提案手法はうまく修復できていることが分かる．これは，ガウシアンフィルタによって近傍のパッチのみから参照して埋めた後に ResNet で滲み出しを補正するようにしていることで，詳細やエッジを壊さずに修復できているのだと考えられる．



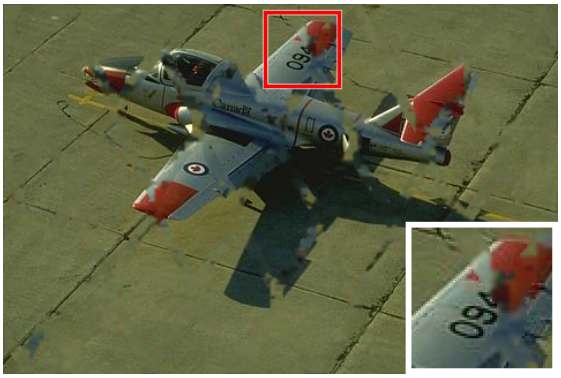
(a) フェンス画像



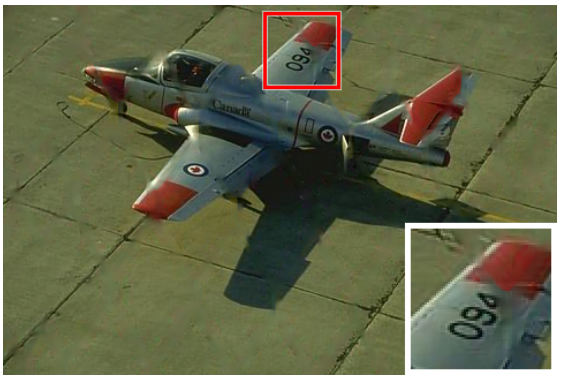
(b) Groundtruth



(c) Criminisi らの手法 [94]
PSNR(25.25)/SSIM(0.894)



(d) Farid らの手法 [24]
PSNR(25.90)/SSIM(0.899)



(e) 提案法
PSNR(29.26)/SSIM(0.937)

図 4.20　合成画像に対するフェンス除去の比較（Plane）

(a) フェンス画像

(b) Groundtruth

(c) Criminisi らの手法 [94]
PSNR(20.61)/SSIM(0.866)

(d) Farid らの手法 [24]
PSNR(20.73)/SSIM(0.866)

(e) 提案法
PSNR(25.85)/SSIM(0.933)

図 4.21　合成画像に対するフェンス除去の比較（Horse）

4.7　提案法の課題

提案手法は様々な自然フェンス画像に対して検出と除去ができ，従来手法よりも性能が上回っていることが分かった．しかし，図 4.22 のようにフェンス検出が不十分な画像も存在する．斜めから撮影されて奥側のフェンスが潰れてしまっている場合や，フェンスの素材や形状が異なると上手くフェンスを検出できないことがある．これは入手可能な自然フェンス画像とそのマスクだけでは，全てのフェンスに対応できず，ある程度の限界があるということである．そのために，回転や拡大縮小だけでない最適なデータ拡張を考える必要がある．また，後処理による悪影響も課題である．今回検出ネットの出力後にモルフォロジー変換を行ったが，図 4.23 に示すように，不要な部分をなくすことができる一方で必要な情報を落としてしまうこともある．本研究ではフェンス検出を回帰タスクとして扱ったが，分類タスクとして扱い，最終層の活性化関数をシグモイド関数にすることで後処理なくフェンスマスクの出力精度が上げられるのではないかと考える．

4.8　フェンス除去のまとめ

ユーザ入力なしで全自動で，一枚のフェンス画像からフェンスを検出し除去するアルゴリズムを提案した．ロバスト性が高くないという従来法の問題を解決するため，この問題をフェンス検出タスクとフェンス除去タスクに分けて取り組んだ．それぞれにおいて，CNN の最新知見と空間フィルタという画像処理の古典的知見をうまく融合させることによって最適なネットワークを作成した．

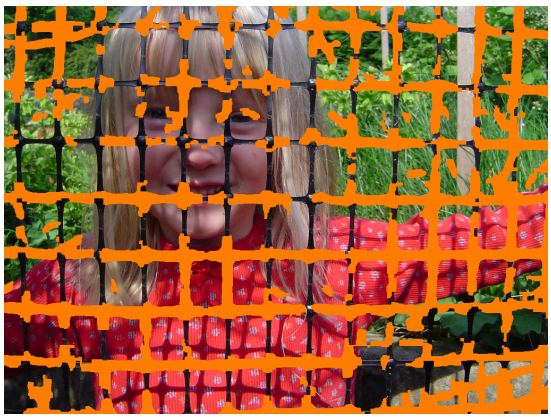
まず，フェンス検出タスクにおいて，画像の大局的特徴と局所的特徴を捉えるためにネットワーク構造は U-Net をベースにした．それだけでなく，入力にグレースケール画像と RGB カラー画像の両方を含めることで色依存を低減させた．また，エッジ検出フィルタ処理をした画像を入力することで，フェンス検出精度を高めた．これらの工夫によってフェンス検出において，提案手法が従来手法と比べて客観的にも主観的にも上回っていることを示した．

次に，フェンス除去タスクにおいて，前処理としてガウシアンフィルタを用いた後に ResNet によって高周波成分の補間を行った．ネットワークを学習する際に，フェンス画像とそれに対応するフェンスなし画像を入手することが困難であるので，新たにフェンス画像を合成させることによってデータセットを作成した．実験結果から，主観的評

(a) フェンス画像

(b) 推定フェンスマスク

(c) フェンス画像



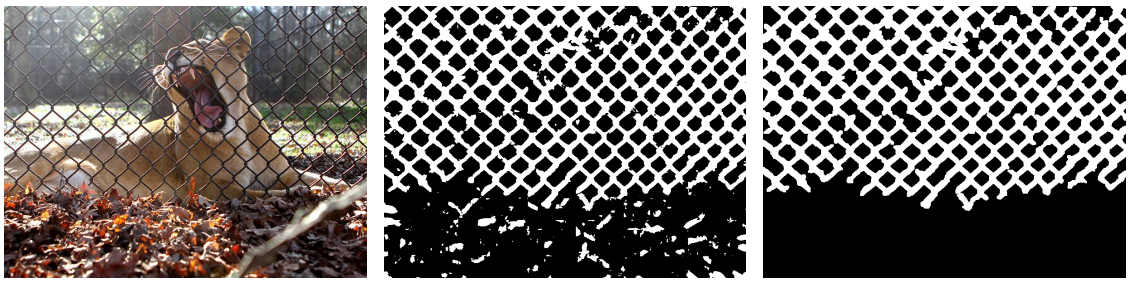
(d) 推定フェンスマスク

図 4.22　提案法の失敗例

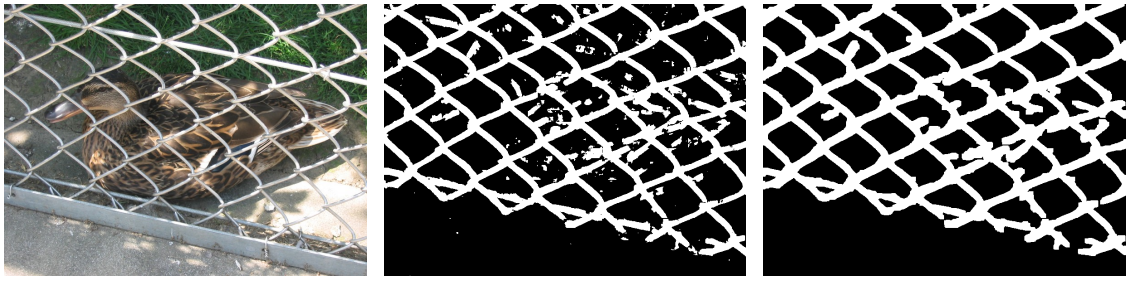
価と客観的評価において従来手法よりも精度が高くなっているということが示された．

4.8.1　様々なフェンス形状への対応

今回の提案手法では，一般的な形のフェンスを想定していたが，様々な形状や特殊な撮影下のフェンスに対応できるために，空間フィルタと CNN を最適に組み合わせてきた．また，データセットの回転，反転，拡大縮小，輝度変化を行うことにより，学習データセットの拡張を行った．しかし，実世界のあらゆるフェンスに対応させるためには，アフィン変換等のさらなるデータセット拡張をする必要がある．



(a) フェンス画像　(b) フェンス検出ネットの出力　(c) 後処理後



(d) フェンス画像　(e) フェンス検出ネットの出力　(f) 後処理後

図 4.23　後処理による悪影響の例

4.8.2　実行時間の短縮

提案手法は画像サイズにもよるが，GPU(Graphics Processing Unit) を用いれば数秒程度で処理ができるため，数十秒かかる従来手法と比べたら実行時間は速いと言える．しかし，実際のアプリケーションとして使用する場合には，さらなる実行時間の短縮が求められる．ネットワークの構造や複雑な計算を簡略化させることで，高い精度を保ったまま実行時間を短縮させるということを今後の展望としたい．

第 5 章

効果的なモジュールを用いた GAN に基づく低照度画像強調

5.1 低照度画像強調について

本章では劣化画像のうち照度環境が要因で劣化した画像の復元について扱う．低照度環境で撮影された画像は，周囲の光量不足により画像が暗くなり，撮影したい物体の視認性が下がる．例えば，屋内パーティーや夜間の風景撮影など，暗いシーンでの鮮明な画像は，個人のユーザにとっても望ましいものである．また，監視カメラや自動運転システムでは，暗い環境下での正確な物体検出や識別が求められる．被災した建物などの危険な場所でのロボットによる作業代替を想定すると，暗所であることが原因で対象物の見落としや認識ミスは許されない．図 5.1 は屋外で撮影された低照度画像における応用シーンの一例である．

従来は撮影時の工夫として，フラッシュ撮影やカメラのナイトモードなど露光時間を調整してきた．しかし，対象物に光や熱が加わることによる不具合，コストや電池消耗などの実用面での問題，露光過多による白飛びやノイズなどの問題が発生してしまう．また，撮影後に Photoshop などのアプリを用いて人間が加工することも可能である．しかし，画像の品質を保ちながら照度を調整するのには高度な専門性が必要な上に時間と手間がかかってしまう．そこで，すでに撮影された低照度画像を自然に復元する低照度画像強調（Low-light image enhancement）が長らく研究されている．ノイズやアー

(a) 屋外カメラ画像での顔検出

(b) 車載カメラ画像での車番認識

図 5.1　低照度画像強調の応用シーン

チファクトの削減，エッジやテクスチャの保存，自然な明るさと色の再現を実現しながら画像復元することが求められている．

5.2　低照度画像強調の従来法

5.2.1　モデルベースの手法

低照度画像強調は 20 年以上前から研究されている分野であり，2014 年頃まではヒストグラム平坦化ベースの手法が主流であった．ヒストグラム平坦化とは，画像のコントラストを強調する古典的な手法である．この方法では，0 から 255 のすべての輝度値においてヒストグラムの高さが等しくなるように画像全体の濃淡分布を変換する．低照度画像の特徴を考慮し様々な手法 [27–30] が提案されてきた．しかし，一定以上に暗い画像に対してはノイズの増幅や変色が生じてしまうという問題点がある．そこで，人間の視覚メカニズムを参考にした Retinex 理論に基づいた低照度画像強調の手法が [31–36] 提案されるようになった．Retinex ベースの手法では，画像 S を反射成分 R と照度成分 I に分割する．

$$S = R \odot I. \tag{5.1}$$

これらの手法は極端な照明条件や非均一な照明条件に対する効果は限定的で，見た目も不自然なものが多い．また，画像を反射成分と照度成分に分割して処理をすることから計算コストが高いという問題点がある．

5.2.2　深層学習ベースの手法

深層学習技術の発展に伴い，2017 年ころから多くの深層学習を用いた手法が提案されている．Retinex 理論に基づく手法 [37–40] や画像のグレー成分とカラー成分を分けて処理をする手法 [41] などがある．また，End-to-end で処理をする CNN ベースの手法 [42–44]，GAN ベースの手法 [105–107] や半教師あり学習や教師なし学習の手法 [48–51] も存在する．また，2022 年以降は Transformer ベースの手法 [45, 46] や拡散モデルベースの手法 [108, 109] も提案されている．

本研究で精度比較に使っている三つの従来法について手法の概要を紹介する．

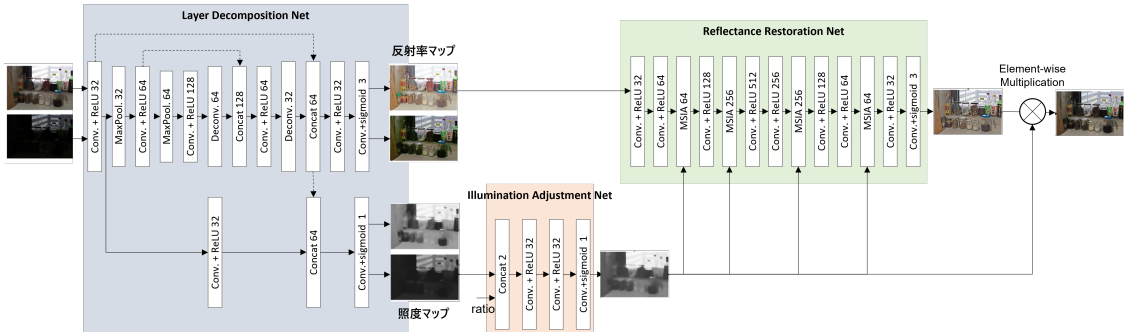


図 5.2　KinD++ [39]

KinD++ (IJCV'22) [39]

KinD++ は，画像の様々なスケールでの照明調整を行う機構を持つ Retinex 理論に基づく手法である．図 5.2 のように，レイヤ分解ネット，反射率復元ネット，照度調整ネットの三つのサブネットワークから構成されている．

一段目のレイヤ分解ネットでは，最初の Conv+ReLU 層を通した後，パスが二つに分岐される．上段では初期反射マップ R_0 を，下段では初期照度マップ L_0 を出力する．反射マップを出力する部分はエンコーダ・デコーダの構造をとっている．

二段目の反射率復元ネットでは，初期反射マップを入力としネットワークを通すことでより自然な反射マップへと変換される．二つの Conv+ReLU 層と MSIA(Multi-Scale Illumination Attention) と呼ばれるマルチスケール照度注意機構を 4 回繰り返すことで最終的な画像が出力される．MSIA は Conv+Sigmoid 層からなる照度注意機構とマルチスケールで Conv+BatchNormalization+ReLU を行うマルチスケールモジュールで構成されている．

三段目の照度調整ネットは単純な構造で，三つの Conv+ReLU 層と Conv+Sigmoid 層で構成されている．

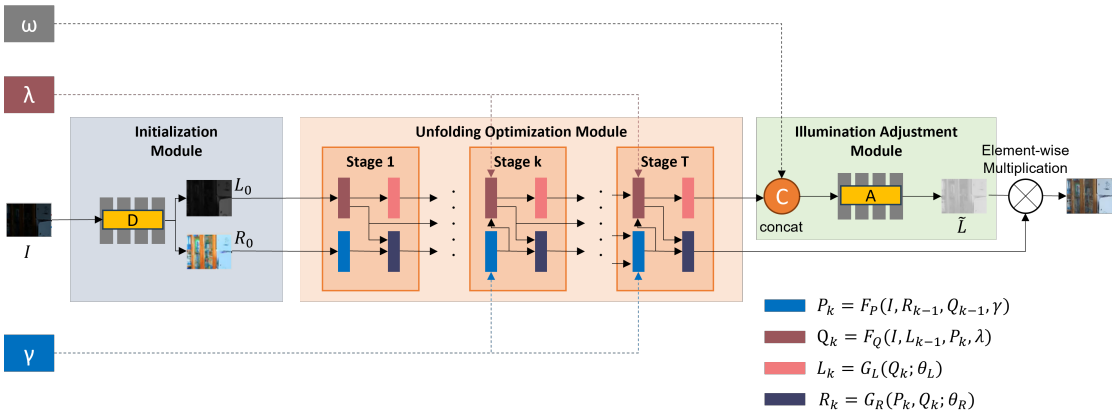


図 5.3　Uretinex-Net [40]

URetinex-Net (CVPR'22) [40]

URetinex-Net はアンフォールディング最適化を用いた Retinex 理論に基づく手法である．図 5.3 のように，初期化モジュール，アンフォールディング最適化モジュール，照明調整モジュールと三つのモジュールから構成される．

一段目の初期化モジュールでは，画像を照度マップ L_0 と反射マップ R_0 に分解しそれを初期マップとする．二つの Conv+LeakyReLU 層と一つの Conv+ReLU 層により構成される．

二段目のアンフォールディング最適化モジュールでは，次式の最適化問題を解く．

$$\min_{P,Q,R,L} \quad \|I - P \cdot Q\|_F^2 + \alpha\Phi(R) + \beta\Psi(L) \text{s.t.} \quad P = R, Q = L, \tag{5.2}$$

ここで，$\|\cdot\|_F^2$ はフロベニウスノルム，$\Phi(R)$ と $\Psi(L)$ は R および L の事前分布を示す正則化項，α と β はトレードオフパラメータである．最適な P, Q, R, S を探索すべく，T 回の処理を繰り返して最適化問題を解いていく．

三段目の照度調整モジュールでは，ユーザが指定した強調率 ω に従い強調画像を出力する．四つの Conv 層から構成されており，2 段目で出力された照度マップ L_T を入力に調整後の照度マップ $\tilde{L}$ が出力される．

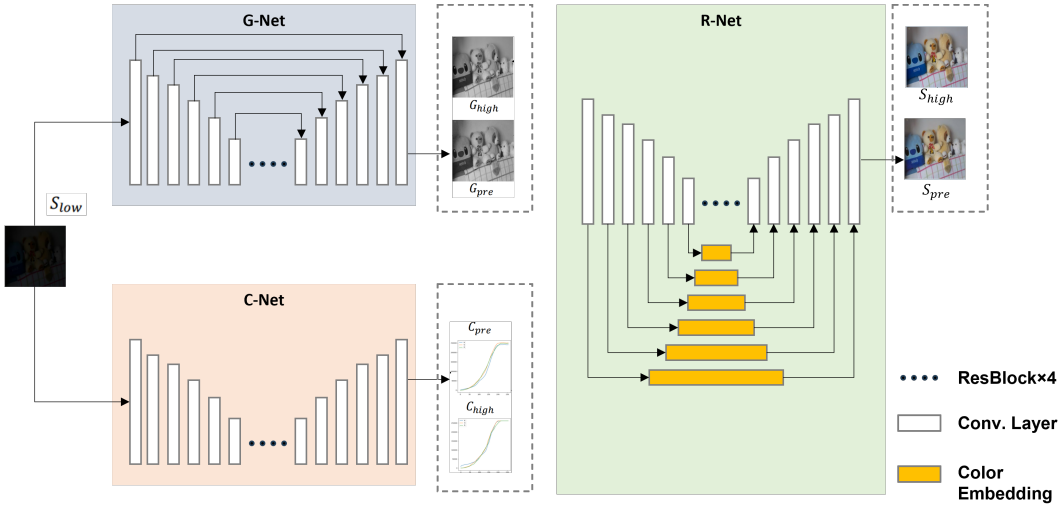


図 5.4　DCC-Net [41]

DCC-Net (CVPR'22) [41]

DCC-Net は画像の構造的特徴と色の特徴を分けて処理する手法である．図 5.4 のように，G-Net，C-Net，R-Net の三つのサブネットで構成されている．

一段目の G-Net は，入力された低照度画像をグレースケールに変換し，画像の構造やテクスチャを処理するサブネットである．U-Net と同様にエンコーダとデコーダで構成されるパイプライン構造を採用している．

二段目の C-Net は，色分布の学習に焦点を当てており，スキップコネクションを持たないがエンコーダとデコーダで構成されている．入力を低照度画像とし，カラーヒストグラムを出力としている．

三段目の R-Net は，G-Net で出力されたグレー画像と C-Net で出力されたカラーヒストグラムを組み合わせて画像を復元する．エンコーダとデコーダの畳み込み層に加え，R-Net の出力であるヒストグラムから空間的特徴を補間して色を埋め込む PCE モジュールを組み込んでいる．PCE はピラミッド構造を持つ六つのカラーエンベッディング（CE）モジュールで構成されている．

5.3　従来法の問題点と提案法における改良点

低照度画像強調は，ノイズやアーチファクトの削減，エッジやテクスチャの保存，自然な明るさと色の再現という三つの主要なタスクが含まれる．深層学習ベースの従来法

は好ましい条件下では高精度だが，次のような問題点がある．

- ノイズ除去のための平滑化が強く効きすぎており，被写体のエッジやテクスチャが失われている
- 屋外など照度が一定でない環境で撮影された画像では，アーチファクトや不自然な色味となる
- 複数のサブモジュールを組み合わせた複雑な構造を持つため，処理時間が長い

スマートフォンなどの個人利用やセキュリティカメラなどの商用利用を考えると，精度を保ちつつ処理速度を向上させる必要がある．そこで，本研究では，ノイズやアーチファクトの削減，エッジやテクスチャの保存，自然な明るさと色の再現を実現しながら，高速かつシンプルなネットワークを提案する．

5.4 提案法

5.4.1 ネットワークの全体構造

提案法のネットワーク構造を図 5.9 に示す．提案法は GAN を用いており，低照度画像強調を行う Generator と，入力が本物の通常光画像か人工的に強調された画像かを評価する Discriminator で構成される．画像分解することで最終出力が中間出力の学習精度に依存してしまう従来手法とは異なり，提案法は End-to-end のアーキテクチャを採用している．

5.4.2 Generator の構造

入力となる低照度画像を S_{low}，出力の強調後画像を S_{enhanced} とすると，Generator のネットワーク G は次のように表せる．

$$\boldsymbol{S}_{\mathrm{enhanced}} = G(\boldsymbol{S}_{\mathrm{low}}) \tag{5.3}$$

低照度画像強調では，画像の大局的特徴と局所的特徴を捉えることが有効とされており U-Net を使用されることが多い [110]．提案法でも U-Net の構造をベースとした．通常の U-Net ではエンコーダで 4 回のダウンサンプリングを行うが，計算量を低減するためダウンサンプリング回数を 2 回に削減した．

最終層は残差成分と入力画像を足し合わせる事により最終出力画像としている．残差学習は出力と入力のマッピングを直接学習するよりも，残差のマッピングを学習するほ

うがよりロバストなモデルになることが先行研究から明らかになっている [84,111,112].

5.4.3 前処理としての空間フィルタリング

低照度画像はエッジやテクスチャに関する情報が少ないため，強調された画像がぼやけてしまう傾向がある．この問題を解決するため，U-Net への入力の前処理として，RGB 入力画像をグレースケールに変換したあとエッジ強調フィルタをかけた．

$$\boldsymbol{S}_{\text{low,gray}} = 0.299\boldsymbol{S}^{R}_{\text{low}} + 0.587\boldsymbol{S}^{G}_{\text{low}} + 0.114\boldsymbol{S}^{B}_{\text{low}} \tag{5.4}$$

ここで，$\boldsymbol{S}^{R,G,B}_{\text{low}}$ はそれぞれ入力低照度画像の RGB 成分である．エッジ検出には，画像の上下左右の勾配に加え斜め方向の勾配も検出できる 8 近傍ラプラシアンフィルタを用いた．ラプラシアンフィルタはノイズ強く反応してしまうため，前処理としてブラーフィルタとガウシアンフィルタを適用した．

$$\boldsymbol{S}'_{\text{edge}} = ((\boldsymbol{S}_{\text{low,gray}} \otimes \boldsymbol{K}_{\text{blur}}) \otimes \boldsymbol{K}_{\text{Gauss}}) \otimes \boldsymbol{K}_{\text{Lap8}} \tag{5.5}$$

ここで，$\boldsymbol{K}_{\text{blur}}, \boldsymbol{K}_{\text{Gauss}}, \boldsymbol{K}_{\text{Lap8}}$ はそれぞれブラーカーネル，ガウシアンカーネル，8 近傍ラプラシアンカーネルを表す．エッジ抽出された画像を図 5.5 に示す．先述の通り，ラプラシアンフィルタはノイズに敏感である．ブラーフィルタとガウシアンフィルタによるノイズ低減をしないとエッジやテクスチャ以外のノイズまで強調してしまうということが図からもわかる．

5.4.4 学習モジュールの導入

精度と処理時間のトレードオフの問題を解決するため，三つのモジュールを導入した．

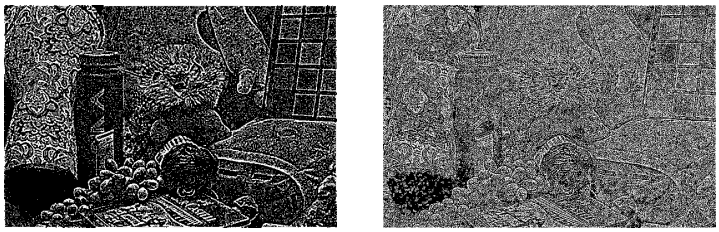
Res FFT-ReLU

低照度画像に含まれる高周波成分を補強しつつノイズを低減させるために，Res FFT-ReLU ブロック [113] を導入した．エンコーダでは 1 段目と 2 段目で 2 回ずつ，3 段目では 5 回，デコーダでは 2 段目と 1 段目で 1 回ずつ ResFFT-ReLU ブロックをかけた．Res FFT-ReLu ブロックとは Deep らによって提案されたモジュールで，図 5.6 のような構造を持つ．入力を $\boldsymbol{Z}$ とすると，出力 $\boldsymbol{Y}$ は次のように表される．

$$\boldsymbol{Y} = \boldsymbol{Y}^{\text{fft}} + \boldsymbol{Y}^{\text{res}} + \boldsymbol{Z} \tag{5.6}$$

第一項目の $\boldsymbol{Y}^{\text{fft}}$ は，入力 $\boldsymbol{Z}$ に対して二次元高速フーリエ変換 $\mathcal{F}$ をかけたもので

(a) グレースケール画像　(b) エッジ検出画像（提案法）　(c) ブラーフィルタなし



(d) ガウシアンフィルタなし　(e) ブラーフィルタとガウシアンフィルタなし

図 5.5　低照度画像のエッジ強調

ある．

$$\tilde{\boldsymbol{Z}} = \mathcal{F}(\boldsymbol{Z}) \tag{5.7}$$

二次元 FFT によって画像を周波数領域に変換した後，1×1 の畳み込み，活性化関数 ReLU をかけて再度畳み込みを行う．

$$h(\tilde{\boldsymbol{Z}}; K_1, K_2) = \mathrm{ReLU}(\tilde{\boldsymbol{Z}} \otimes K_1) \otimes K_2 \tag{5.8}$$

ここで，K_1 と K_2 は畳み込みカーネルである．次に，逆二次元 FFT を実行して出力を画像空間に戻す．

$$\boldsymbol{Y}^{\mathrm{fft}} = \mathcal{F}^{-1}(h(\tilde{\boldsymbol{Z}}; K_1, K_2)) \tag{5.9}$$

第二項目の $\boldsymbol{Y}^{\mathrm{res}}$ は，3×3 の畳み込み，ReLU，そして 3×3 の畳み込みを通して出力される．

$$\boldsymbol{Y}^{\mathrm{res}} = \mathrm{ReLU}(\tilde{\boldsymbol{Z}} \otimes K_3) \otimes K_4 \tag{5.10}$$

Res FFT-ReLU ブロックの導入により，画像の広範な特徴を抽出しオブジェクトの境界を明確にするだけでなく，ノイズを減少させ，鮮やかな色の再現する効果があると考える．

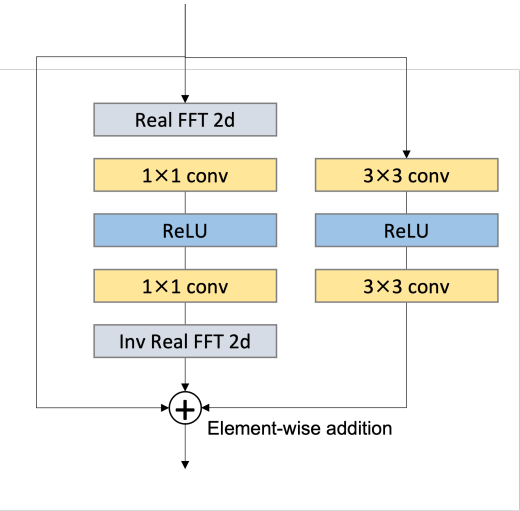


図 5.6　Res FFT-ReLU

Channel Attention

中間層の出力に対して，チャネル間の重要度で重み付けをすることで精度が向上するのではないかという仮説のもと，Channel Attention を導入した（図 5.7）．

アテンション機構は元々自然言語処理の分野で用いられてきた [114] が，現在では画像処理の領域でも応用が見られるようになっている．画像処理のタスクにおいては，二つのタイプのアテンション機構が存在する [115]．Channel Attention は特徴マップ内のチャネルの重要性を調整する一方，Spatial Attention は画像内の位置や領域の重要性を調整する．

U-Net が複数のスケールにおいて局所的な特徴と大局的な特徴を捉える役割を果たしていることを考慮すると，Channel Attention を各スケールに組み込むことが画像の色と照明の復元に効果的である考えた．ただし，計算のコストを抑えるため，Channel Attention はエンコーダのみに導入した．

Pixel Shuffler

U-Net のデコーダでは，アップサンプリングと逆畳み込みがよく使用される．高速な処理かつ高解像度の画像を得るために，Pixel Shuffler を組み込んでいる [116]．図 5.8 に示すように，Pixel Shuffler の関数を P とすると，P は入力マップのサイズが $C_{out} \cdot r^2 \times H_{in} \times W_{in}$ である入力マップを，サイズが $C_{out} \times rH_{in} \times rW_{in}$ である出力マップに変換する処理である．

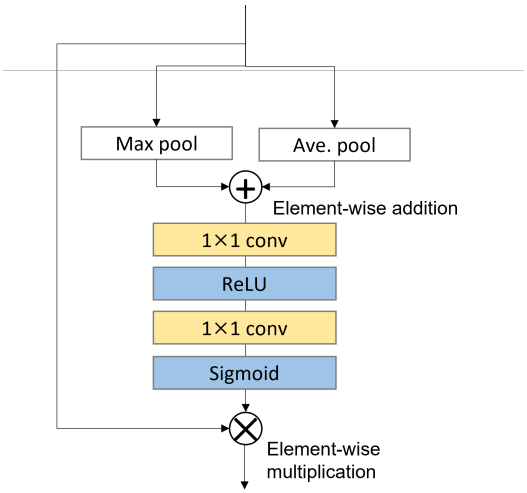


図 5.7　Channel Attention

数式的には，この操作は以下のように表される．

$$\begin{pmatrix} c' \\ y' \\ x' \end{pmatrix} = \begin{pmatrix} C_{out} \cdot r \cdot \mathrm{mod}(y, r) + C_{out} \cdot \mathrm{mod}(x, r) + c \\ \lfloor \frac{y}{r} \rfloor \\ \lfloor \frac{x}{r} \rfloor \end{pmatrix} \tag{5.11}$$

ここで，c，r，y，x は出力マップのチャネル，拡大率，y 座標，x 座標を表す．なお，$\lfloor \cdot \rfloor$ は床関数である．

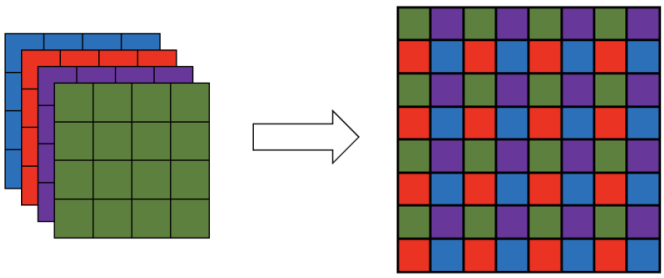


図 5.8　Pixel Shuffler

5.4.5　Discriminator の構造

Discriminator では，通常光量画像 I または強調後画像 $\boldsymbol{S}_{\mathrm{enhanced}}$ と低照度画像 $\boldsymbol{S}_{\mathrm{low}}$ を連結させたものを入力させ，通常光量画像（本物）であるか強調後の画像（偽物）で

あるかを識別する．Discriminator は五つのブロックからなる．第一ブロックは 3×3 の畳み込みと LeakyReLU で構成される．第二から第四ブロックは 3×3 の畳み込みと LeakyReLU のあとに Batch Normalization を適用し，3×3 の畳み込みとダウンサンプリング，LeakyReLU と Batch Normalization を行う．第五ブロックは 3×3 の畳み込みと Batch Normalization と LeakyReLU のあとに再度 3×3 の畳み込みをして 0 から 1 の間の値をとるスカラーを出力させる．

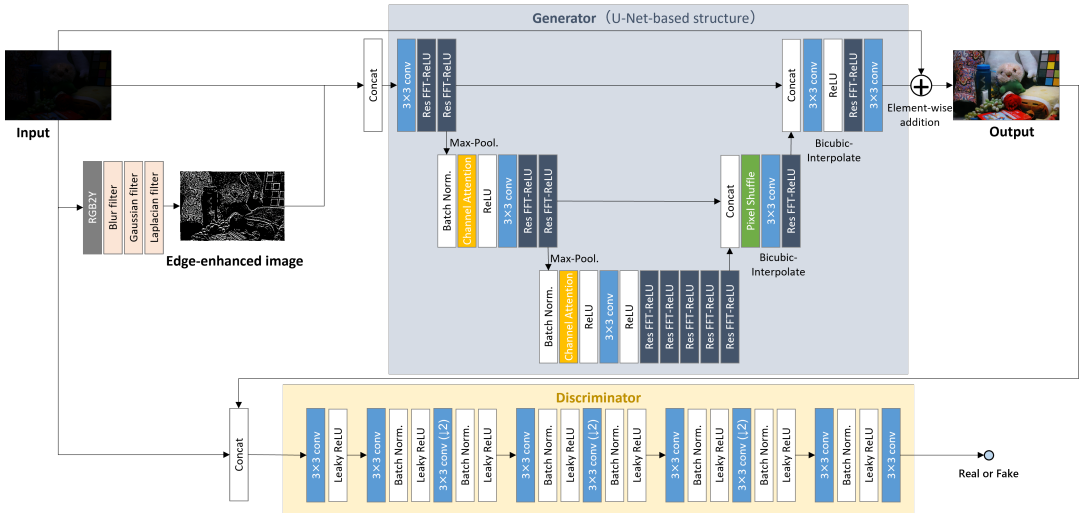


図 5.9　提案法のネットワークの全体像

5.4.6　損失関数

GAN では，Generator と Discriminator の損失関数を定義する必要がある．Generator のパラメータは次の損失関数を最小化することによって学習した．

$$\mathcal{L}_{\mathrm{G}} = \mu_1 \mathcal{L}_{\mathrm{L1}} + \mu_2 \mathcal{L}_{\text{MS-SSIM}} + \mu_3 \mathcal{L}_{\mathrm{adv}} \tag{5.12}$$

ここで，μ_1, μ_2, μ_3 は実験的に決定する係数である．

第一項である $\mathcal{L}_{\mathrm{L1}}$ は入力画像 $\boldsymbol{S}_{\mathrm{low}}$ と生成画像 $G(\boldsymbol{S}_{\mathrm{low}})$ の L1 ノルムを表し，以下のように定義される．

$$\mathcal{L}_{\mathrm{L1}} = \|\boldsymbol{I} - G(\boldsymbol{S}_{\mathrm{low}})\|_1 \tag{5.13}$$

式 (5.12) の第二項である $\mathcal{L}_{\text{MS-SSIM}}$ は MS-SSIM (Multi-Scale Structural Similarity) ロス [117] である．MS-SSIM は，SSIM の発展形であり，異なるスケールの複数の画像ペアを作成するために，低域通過フィルタリングとダウンサンプリングを繰り返し行

うことで計算される．正解画像 $\boldsymbol{I}$ と生成画像 $G(\boldsymbol{S}_{\text{low}})$ に対して MS-SSIM を計算して，すべての要素が 1 である行列 $\mathbf{1}$ から減算する．

$$\mathcal{L}_{\text{MS-SSIM}} = \mathbf{1} - \text{MS-SSIM}(\boldsymbol{I}, G(\boldsymbol{S}_{\text{low}})) \tag{5.14}$$

式 (5.12) の第三項である $\mathcal{L}_{\text{adv}}$ は Adversarial ロスである．これは Discriminator が入力画像を本物の通常光量画像か低光量画像から生成された画像か識別する能力を測るものである．

$$\mathcal{L}_{\text{adv}} = \|\mathbf{1} - D(G(\boldsymbol{S}_{\text{low}}))\|_2^2 \tag{5.15}$$

他方で，Discriminator のパラメータは以下の損失関数を最小化することにより学習した．

$$\mathcal{L}_D = \|D(G(\boldsymbol{S}_{\text{low}}))\|_2^2 + \|V - D(\boldsymbol{I})\|_2^2 \tag{5.16}$$

ここで，V は平均が 1 の正規分布に従うランダムな値を持つ行列である．

5.4.7 学習データセットと学習パラメータ

学習にあたり，合成画像と自然画像のデータセットを用いた [37]．合成画像データセットは 1000 枚の低光量画像とそれに対応する通常光量画像，自然画像データセットは 485 枚の低光量画像とそれに対応する通常光量画像を含んでいる．モデルのロバストさを高めるために，大きさが $256 \times 256 \times 3$ のパッチを切り出し，ランダムに左右上下反転，回転をすることでデータ拡張をした．パラメータは Adam を用いて最適化を行い，学習率 l_r は 0.0002，重みの減衰率の係数 β_1 と β_2 はそれぞれ 0.5 と 0.999 に設定した．バッチサイズは 8 とし，1045 エポック学習させた．また，損失関数の係数 μ_1, μ_2, μ_3 はそれぞれ $1, 1, 0.01$ に設定した．

5.5 低照度画像強調の実験と比較

5.5.1 実験内容

低照度画像強調の五つの従来手法に対して客観的評価と主観的評価を行った．

5.5.2 比較対象

本研究では深層学習ベースでの低照度画像強調に焦点を当てているため，2021 年以降に提案された四つの教師あり学習ベースの手法と比較をする．

- KinD++ (IJCV'21) [39] : KinD++ は，画像の様々なスケールでの照明調整を行う機構を持つ Retinex 理論に基づく三つのサブネットワークから構成される．
- URetinex-Net (CVPR'22) [40] : URetinex-Net は，アンフォールディング最適化手順を用いた Retinex 理論に基づく三つのサブネットワークから構成される．
- DCC-Net (CVPR'22) [41] : DCC-Net は，グレー成分の予測，カラー成分の予測，両方から画像を復元するための三つのサブネットワークから構成される．
- MIR-Netv2 (TPAMI'22) [55] : MIR-Netv2 は，マルチスケール残差ブロックを中心とした再帰的残差フレームワークで構成される．
- LLFormer (AAAI'23) [46]: LLFormer は，Transformer ブロックを組み合わせたネットワークである．

5.5.3 評価手法

室内低照度画像は LOL データセット，屋外低照度画像は DARK FACE データセットを用いた．客観的な評価は，低照度画像強調でよく用いられる画像品質評価指標で評価をした．LOL データセットは低光量画像と通常光量画像の 15 ペアが含まれており，正解画像が存在するため完全参照による評価が可能である．PSNR, SSIM, LPIPIS を用いて評価をした．また，処理時間は NVIDIA GeForce GTX 1080 Ti GPU 上で計測をした．DARK FACE データセットは夜に屋外で撮影された 100 枚の画像で構成される．正解画像にあたる通常光量画像がないため，非完全参照による評価を行った．主観的な評価は，LOL データセットと DARK FACE データセットそれぞれに対して視覚的に比較をした．

5.5.4 実験結果

客観評価結果

表 5.1 は室内低照度画像 15 枚に対して三つの画像品質評価指標と処理時間を平均した値である．赤字は一番高い精度，青字は二番目に高い精度であることを表す．提案法は従来法と比較して各指標にて一番高い精度かつ高速な処理時間を実現できている．

表 5.2 に屋外低照度画像 100 枚に対して二つの画像品質評価指標を平均した値を示す．従来法は提案法と比較して各指標で最も高い精度である．NIQE はノイズや鮮鋭度などの統計的特徴を分析することで画像の自然さを評価し，BRISUE はコントラストやエッジの鮮鋭度，テクスチャなどを抽出して画像の空間的な品質を評価する．屋外で

撮影された画像は画像内での明るさの分布が異なり人工的な構造物と自然のものが混在しているためより複雑な状況である．GAN では Discriminator が Generator の出力画像が本物か偽物かを識別するため，複雑な画像に対しても自然な色合いやテクスチャの画像を出力できるようになると考えられる．

以上の結果から，提案法は室内低照度画像と屋外低照度画像の両方でロバストであることが示された．

表 5.1　室内低照度画像強調の精度および処理時間の比較

手法	PSNR↑	SSIM↑	LPIPS↓	処理時間↓
KinD++ [39]	21.80	0.829	0.158	18.639
URetinex-Net [40]	21.33	0.833	0.121	0.108
DCC-Net [41]	22.98	0.848	0.143	0.040
MIR-Netv2 [55]	24.74	0.847	0.116	0.365
LLFormer [46]	23.65	0.816	0.169	0.862
提案法	25.08	0.859	0.107	0.021

主観評価結果

図 5.10 と図 5.11 に室内低照度画像における画像強調結果のうち二枚を示す．KinD++ [39] は鮮やかな画像を生成しているが，不自然な色のアーチファクトが発生してしまっている．URetinex-Net [40] は全体的に白っぽい印象の画像が生成されてしまう．DCC-Net [41]，MIR-Netv2 [55]，および LLFormer [46] は比較的自然な画像を生成するが，画像内の暗い部分でノイズが発生してしまい輪郭がぼやけて見える．提案手法では自然な色を保ちながら，細部までしっかりと表現した画像を生成できていることがわかる．

次に，図 5.12 と図 5.13 に屋外低照度画像における画像強調結果のうち二枚を示す．KinD++ [39] は滑らかなテクスチャで鮮やかだが，やや人工的で不自然な画像が生成されてしまっている．URetinex-Net [40] は木の枝などが赤みがかっており輪郭もぼけている．DCC-Net [41] と MIR-Netv2 [55] は画像全体が緑がかってしまい，アーチファクトが発生してしまっている．LLFormer [46] は画像全体がぼやけて輪郭が不鮮明

表 5.2　屋外低照度画像強調の精度比較

手法	NIQE↓	BRISQUE↓
KinD++ [39]	2.81	29.90
URetinex-Net [40]	3.06	20.10
DCC-Net [41]	3.03	30.64
MIR-Netv2 [55]	3.03	32.32
LLFormer [46]	2.74	30.03
提案法	2.49	18.73

である．提案手法では，画像の明瞭さが大幅に向上しており，電子掲示板の文字を鮮明に復元できているため，画像内で明るい場所と暗い場所が混在する画像に対しても効果的に強調できることがわかる．

5.6　低照度画像強調において前処理やモジュールの導入が学習結果に及ぼす影響

導入された前処理とモジュールの効果を確認するために，GAN，エッジ強調フィルタ，Channel Attention，Res FFT-ReLU および MS-SSIM ロスを除いた場合の性能を客観的および主観的に検証して，提案法の効果を確認した．各要素を取り除いた場合の出力画像を図 5.14 と図 5.15 に，画像品質評価指標の平均値を表 5.3 に示す．

GAN を用いた場合とそうでない場合を比較すると，GAN を用いたほうが精度が高くなっている．GAN では Discriminator が Generator の出力画像が本物か偽物かを識別させているため，複雑な画像に対しても自然な色合いやテクスチャの画像を出力できるようになると考えられる．エッジ強調フィルタを除くと，画像のエッジと高周波成分が失われて画像がぼやけていることがわかる．Channel Attention を除くと，画像の色が明るすぎたり暗すぎたりしているだけでなくテクスチャなどの細部の情報が失われている．Res FFT-ReLU を除くと画像の輪郭がぼやけてしまっている．Res FFT-ReLU はグローバルな特徴を捉えることができるため，画像のノンローカル処理を導入することで精度を最大化することができるのではないかと考える．MS-SSIM ロスを除くと，

SSIM と PSNR の値は最も低くなっている．構造的類似性を捉えることで輪郭やテクスチャを捉えることができるのではないかと考える．

以上の実験結果から，各前処理とモジュールの組み合わせによって低照度画像強調の精度を高めるのに役立っているということがわかった．

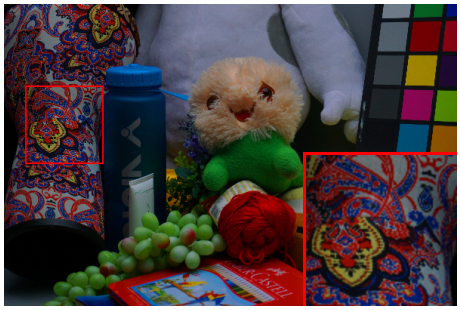
表 5.3　前処理やモジュールの有無による画像強調の精度比較

	室内画像			屋外画像	
モデル	PSNR↑	SSIM↑	LPIPS↓	NIQE↓	BRISQUE↓
提案法	25.84	0.863	0.106	2.49	18.73
GAN なし	24.31	0.854	0.108	2.92	26.20
エッジ強調フィルタなし	21.49	0.829	0.134	2.52	28.76
Channel Attention なし	21.71	0.836	0.134	2.80	23.57
Res FFT-ReLU なし	21.41	0.818	0.142	3.76	19.07
MS-SSIM ロスなし	21.85	0.791	0.182	3.09	28.52

5.7　低照度画像強調のまとめ

低照度画像強調には，不自然な照度や色彩，ディテールの損失，強調時のノイズやヘイズの生成など多くの問題がある．本研究では，従来の低照度画像強調手法のように複雑な構造にはせず，GAN を用いた U-Net ベースのシンプルなネットワークを提案した．従来法の問題を解決するために，適切な前処理とモジュールを導入した．具体的には，前処理としてエッジ強調フィルタを導入し，モジュールとして Res FFT-ReLU，Channel Attention，および Pixel Shuffler を導入した．さらに，損失関数には L1 ロスと MS-SSIM ロスを組み合わせる．従来法とは異なり End-to-end かつ畳み込みの数を減らすことで，軽量なネットワークを実現することができた．実験結果より，従来法と比べて高い精度かつ処理時間を大幅に短縮することができた．従前の手法では画像内に明るい部分と暗い部分が混在するような画像が苦手であったが，GAN を導入することで克服し自然な色合いの画像を生成できることがわかった．

(a) 低照度画像



(b) 通常照度画像

(c) KinD++ [39]

(d) URetinex-Net [40]

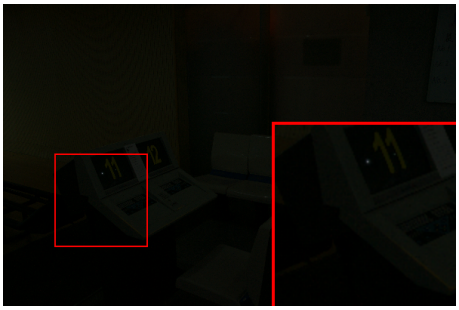
(e) DCC-Net [41]

(f) MIR-Netv2 [55]

(g) LLFormer [46]

(h) 提案法

図 5.10　室内低照度画像に対する強調結果 1



(a) 低照度画像

(b) 通常照度画像

(c) KinD++ [39]

(d) URetinex-Net [40]

(e) DCC-Net [41]

(f) MIR-Netv2 [55]

(g) LLFormer [46]

(h) 提案法

図 5.11　室内低照度画像に対する強調結果 2

(a) 低照度画像

(b) KinD++ [39]

(c) URetinex-Net [40]

(d) DCC-Net [41]

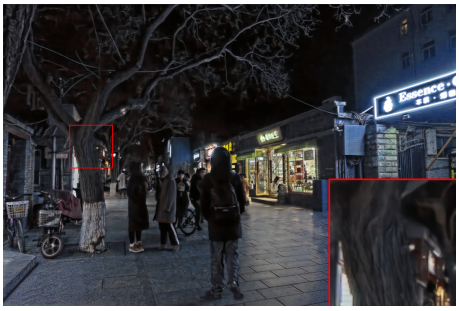
(e) MIR-Netv2 [55]

(f) LLFormer [46]

(g) 提案法

図 5.12　屋外低照度画像に対する強調結果 1

(a) 低照度画像



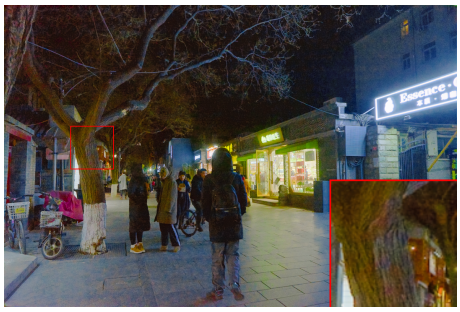
(b) KinD++ [39]

(c) URetinex-Net [40]

(d) DCC-Net [41]

(e) MIR-Netv2 [55]

(f) LLFormer [46]



(g) 提案法

図 5.13　屋外低照度画像に対する強調結果 2

(a) 低照度画像

(b) 提案法

(c) GAN なし

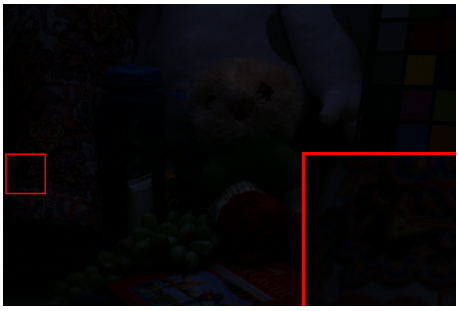
(d) エッジ強調フィルタなし

(e) Channel Attention なし

(f) Res FFT-ReLU なし

(g) MS-SSIM ロスなし

図 5.14　前処理やモジュールの有無による画像強調精度比較 1



(a) 低照度画像

(b) 提案法

(c) GAN なし

(d) エッジ強調フィルタなし

(e) Channel Attention なし

(f) Res FFT-ReLU なし

(g) MS-SSIM ロスなし

図 5.15　前処理やモジュールの有無による画像強調精度比較 2

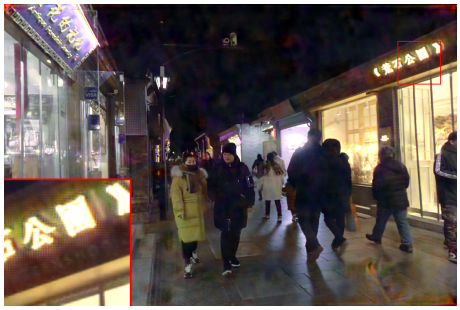
(a) 低照度画像

(b) 提案法

(c) GAN なし

(d) エッジ強調フィルタなし

(e) Channel Attention なし



(f) Res FFT-ReLU なし

(g) MS-SSIM ロスなし

図 5.16　前処理やモジュールの有無による画像強調精度比較 3

(a) 低照度画像

(b) 提案法

(c) GAN なし

(d) エッジ強調フィルタなし

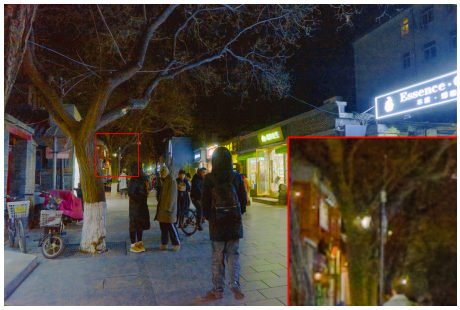
(e) Channel Attention なし

(f) Res FFT-ReLU なし

(g) MS-SSIM ロスなし

図 5.17　前処理やモジュールの有無による画像強調精度比較 4

(a) 低照度画像



(b) 提案法

(c) GAN なし

(d) エッジ強調フィルタなし

(e) Channel Attention なし

(f) Res FFT-ReLU なし

(g) MS-SSIM ロスなし

図 5.18　前処理やモジュールの有無による画像強調精度比較 5

第 6 章

結　論

本論文では，画像に対して発生する劣化のうち，撮影時の環境条件によって発生する劣化画像の復元について研究を行った．具体的には，雨すじの除去（天候環境），フェンス除去（撮影場所），低照度画像強調（照度環境）である．また，手法は一般的に高精度とされる深層学習を用いた手法に焦点を当てた．従来の深層学習ベースの手法は合成画像に対する復元性能は高いものの，効果のある画像の条件が限られていた．複数のサブネットワークを組み合わせた手法や大量のデータセットを学習させた手法が多く，劣化画像の種類と特徴を考慮しきれていないという問題があった．そこで，ネットワークの構造，前処理や後処理，データセットの構築において，雨，フェンス，低照度環境それぞれの特徴を考慮した深層学習ベースの手法を提案した．提案法と従来法の精度比較は，合成画像にて画像品質評価指標を用いた客観的な比較，合成画像および自然画像での復元後の見た目による主観的な比較を行った．実験結果より，提案法は従来法と比べて，高い精度で劣化画像を復元することに成功した．

雨すじ除去

本論文では，複数の雨すじモデルを想定した GAN に基づく手法を提案した．従来の雨すじ除去には，雨すじ除去不足，過平滑化によるディテールの消失，色彩の変化という問題が存在する．提案法の Generator には U-Net を採用し，雨すじ画像の大局的な特徴と局所的な特徴を捉え，雨すじ画像と雨すじなし画像の残差成分を学習させた．合成雨すじ画像のデータセットでは，現実世界での雨すじを想定し，雨の強さ，長さ，方向を変えた雨すじノイズを作成し，加算合成モデルとスクリーン合成モデルを組み合わて合成した．実験結果から，提案法は合成画像と自然画像の両方で従来法より雨すじ除去性能を上回ることができた．多様な雨すじの特徴を考慮したデータセットで学習させることにより，画像のディテールを失うことなく雨すじを除去することができた．今回

は雨すじ除去精度を上げることを主目的にしていたため，実行時間の短縮については今後の課題である．

フェンス除去

本論文では，ユーザ入力なしで一枚の画像からフェンスの検出と除去ができるネットワークを提案した．フェンス検出ネットでは，U-Net を採用し画像の大局的特徴と局所的特徴を捉えた．それだけでなく，前処理にエッジ検出フィルタをかけることでフェンス検出性能を高めた．実験結果より，提案手法が従来手法と比べて精度が上回っていることが確認できた．フェンス除去ネットでは前処理でガウシアンフィルタをかけたあとに ResNet によって画像の高周波成分を補間した．様々な色や形，大きさのフェンスを想定したデータセットを用いてネットワークを学習させた．実験結果から，従来手法よりも精度が高くなっているということが示された．ただし，斜めから撮影されたフェンスやフェンスの素材や形が学習データセットに含まれない場合はフェンス検出・除去の精度が限定的である．

低照度画像強調

本論文では，低照度画像の強調において高速かつ高性能なネットワークを提案した．まず，高速なネットワークを実現するために従来法とは異なり，End-to-end でありながら畳み込みの数を減らすことで，軽量なネットワークを実現した．次に性能の面では，適切な前処理とモジュールを導入することで従来法の問題を解決した．従来の低照度画像強調は，ノイズやアーチファクト発生，エッジやテクスチャの消失，不自然な明るさと色など多くの問題が存在する．提案法では，前処理としてエッジ強調フィルタを使用し，モジュールとして Res FFT-ReLU，Channel Attention，および Pixel Shuffler を導入した．さらに，損失関数には L1 ロスと MS-SSIM ロスを組み合わせた．実験結果から，従来法と比較して高い精度を実現し，処理時間を大幅に短縮することが確認できた．しかし，逆光画像のように暗い部分と明るい部分が混在する場合や，ガラスなどへの映り込みがある画像においては，精度が限定的であった．そこで，GAN ベースのネットワークとすることで画像の空間的特徴および意味的な特徴を捉えるようにした．Generator が低照度画像強調を行い，Discriminator が入力画像が本物の通常光量画像か生成された強調後画像かを識別する役割を果たす．実験では客観的手法と主観的手法で GAN の導入有無による低照度画像の精度を比較した結果，GAN を導入することで自然な色合いで鮮明な画像に復元できるということがわかった．

今後の展望

様々な劣化画像の復元をできるマルチタスクシステムとして，近年注目されている手法は，MIR-Netv2 [55]，TransWeather [118]，Air-Net (All-in-one Image Restoration Network) [119] や IDR (Ingredients-oriented Degradation Reformulation framework) [120] などが挙げられる．MIR-Netv2 [55] は，学習データ次第で単一のタスクが実行できるものであり，ノイズ除去，ぶれ除去，超解像，低照度画像強調のタスクをこなすことができる．TransWeather [118] は天候条件の劣化画像に特化しており，画像から雨・霧・雪を自動で判別して除去することができる．Air-Net [119] は雨・霧除去などの天候条件だけでなく，ぶれ除去にも対応している．IDR [55] は幅広いタスクを自動で判別して実行でき，ノイズ除去，ぶれ除去，雨・霧除去，低照度画像強調をこなすことができる．

最終的には入力された劣化画像の特徴を捉えて一つのネットワークで自動的に鮮明な画像が得られるようなシステムができることを目指したい．そのために三つのステップを踏んで研究をしていく必要があると考える．

1. 想定される劣化画像の種類を洗い出して適切な問題設定をする．
2. 学習データをタスクごとに変えることで単一のタスクであれば実行できるような軽量かつ高精度なネットワーク構造を見つける．
3. ユーザ入力なしに入力画像から自動で実行すべきタスクを判別できるような学習方法を見つける．

本論文は全自動劣化画像復元システムの実現に向けて，特に一つ目と二つ目に対して貢献した．

まずは撮影環境に起因する劣化画像を四つの要因と代表的な劣化の種類で整理したことである．撮影環境に起因する劣化画像は，照度，天候，場所，被写体・撮影者の四つの要因によって発生する．代表的な劣化の種類としては，ノイズ，ブラー，コントラスト低下，解像度低下，画像欠損がある．IDR は多くのタスクをこなせるが，場所起因の画像欠損，本研究で扱ったフェンス除去などには対応ができない．このようにあらかじめ想定される劣化パターンを整理しておくことが重要である．

次に，軽量かつ高精度に単一タスクをこなせるネットワーク構造を見つけるうえでの本研究の示唆は次の通りである．大局的特徴と局所的特徴を捉える U-Net をベースに Channel Attention や Res FFT-ReLU など学習を効率化させるモジュールを導入することで高速かつ高精度を実現できる．また，深層学習という最先端の手法と空間フィル

タリングなどの古典的な画像処理手法を組み合わせることで相互補完ができる．エッジ検出フィルタにより残したい画像のエッジやテクスチャを強調してからネットワークに通す，平滑化フィルタで前処理をしてからネットワークに通すことで高精度を実現できる可能性がある．最後に，対象とする劣化画像が撮影される場面を想定し，多様な特徴を考慮したデータセットで学習させることで自然画像に対してもロバストなモデルとなる．劣化画像の数式モデルの種類や，劣化箇所の大きさ，密度，方向，色などをのパラメータを組み合わせて作成した合成画像を数百から数千枚用意しデータ拡張することでデータセットを作成できる．

上記で設定した問題とネットワーク構造をベースに，自動で実行すべきタスクを認識するための中間層を導入することで全自動劣化画像復元システムの実現に近づくことができると考える．

参考文献

[1] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Deep image prior,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.9446–9454, 2018.

[2] K. Garg and S.K. Nayar, “Detection and removal of rain from videos,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol.1, pp.I–I, 2004.

[3] J. Bossu, N. Hautiere, and J.-P. Tarel, “Rain or snow detection in image sequences through use of a histogram of orientation of streaks,” International Journal of Computer Vision (IJCV), vol.93, pp.348–367, 2011.

[4] M. Li, Q. Xie, Q. Zhao, W. Wei, S. Gu, J. Tao, and D. Meng, “Video rain streak removal by multiscale convolutional sparse coding,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.6644–6653, 2018.

[5] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, “Automatic single-image-based rain streaks removal via image decomposition,” IEEE Transactions on Image Processing (TIP), vol.21, no.4, p.1742, 2012.

[6] Y. Luo, Y. Xu, and H. Ji, “Removing rain from a single image via discriminative sparse coding,” IEEE International Conference on Computer Vision (ICCV), pp.3397–3405, 2015.

[7] Y.-L. Chen and C.-T. Hsu, “A generalized low-rank appearance model for spatio-temporally correlated rain streaks,” IEEE International Conference on Computer Vision (ICCV), pp.1968–1975, 2013.

[8] Y. Li, R.T. Tan, X. Guo, J. Lu, and M.S. Brown, “Rain streak removal using layer priors,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.2736–2744, 2016.

[9] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," IEEE Transactions on Image Processing (TIP), vol.26, no.6, pp.2944–2956, 2017.

[10] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3855–3863, 2017.

[11] H. Zhang and V.M. Patel, "Density-aware single image de-raining using a multi-stream dense network," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.695–704, 2018.

[12] Z. Li, J. Zhang, Z. Fang, B. Huang, X. Jiang, Y. Gao, and J. Hwang, "Single image snow removal via composition generative adversarial networks," IEEE Access, vol.7, pp.25016–25025, 2019.

[13] V.S. Khasare, R.R. Sahay, and M.S. Kankanhalli, "Seeing through the fence: Image de-fencing using a video sequence," IEEE International Conference on Image Processing (ICIP), pp.1351–1355, 2013.

[14] S. Jonna, K.K. Nakka, and R.R. Sahay, "My camera can see through fences: A deep learning approach for image de-fencing," IAPR Asian Conference on Pattern Recognition (ACPR), pp.261–265, 2015.

[15] S. Jonna, K.K. Nakka, and R.R. Sahay, "Deep learning based fence segmentation and removal from an image using a video sequence," Computer Vision – ECCV 2016 Workshops, pp.836–851, Springer International Publishing, 2016.

[16] S. Jonna, K.K. Nakka, and R.R. Sahay, "Towards an automated image de-fencing algorithm using sparsity," arXiv preprint arXiv:1612.03273, 2016.

[17] R. Yi, J. Wang, and P. Tan, "Automatic fence segmentation in videos of dynamic scenes," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.705–713, 2016.

[18] K.K. Nakka, "Automatic image de-fencing system," arXiv preprint arXiv:1610.06924, 2016.

[19] C. Du, B. Kang, Z. Xu, J. Dai, and T.Q. Nguyen, "Accurate and efficient video de-fencing using convolutional neural networks and temporal information," IEEE International Conference on Multimedia and Expo (ICME), pp.1–6, 2018.

[20] A. Yamashita, F. Tsurumi, T. Kaneko, and H. Asama, "Automatic removal of

foreground occluder from multi-focus images," IEEE International Conference on Robotics and Automation (ICRA), pp.5410–5416, 2012.

[21] S. Jonna, S. Satapathy, and R.R. Sahay, "Stereo image de-fencing using smartphones," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.1792–1796, 2017.

[22] Y. Liu, T. Belkina, J.H. Hays, and R. Lublinerman, "Image de-fencing," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1–8, 2008.

[23] M. Park, K. Brocklehurst, R.T. Collins, and Y. Liu, "Image de-fencing revisited," Asian Conference on Computer Vision (ACCV), pp.422–434, 2010.

[24] M.S. Farid, A. Mahmood, and M. Grangetto, "Image de-fencing framework with hybrid inpainting algorithm," Signal, Image and Video Processing (SIVP), vol.10, pp.1193–1201, 2016.

[25] M. Khalid, M.M. Yousaf, K. Murtaza, and S.M. Sarwar, "Image de-fencing using histograms of oriented gradients," Signal, Image and Video Processing (SIVP), vol.12, no.6, pp.1173–1180, 2018.

[26] B. Kang, "Learning robust representations in random forest and deep neural networks for semantic segmentation," PhD thesis, UC San Diego, 2018.

[27] A.M. Reza, "Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement," Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology, vol.38, pp.35–44, 2004.

[28] T. Celik and T. Tjahjadi, "Contextual and variational contrast enhancement," IEEE Transactions on Image Processing (TIP), vol.20, no.12, pp.3431–3441, 2011.

[29] C. Lee, C. Lee, and C.-S. Kim, "Contrast enhancement based on layered difference representation of 2D histograms," IEEE Transactions on Image Processing (TIP), vol.22, no.12, pp.5372–5384, 2013.

[30] G. Yadav, S. Maheshwari, and A. Agarwal, "Contrast limited adaptive histogram equalization based enhancement for real time video system," IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp.2392–2397, 2014.

[31] S. Wang, J. Zheng, H.-M. Hu, and B. Li, "Naturalness preserved enhancement

algorithm for non-uniform illumination images," IEEE Transactions on Image Processing (TIP), vol.22, no.9, pp.3538–3548, 2013.

[32] X. Fu, D. Zeng, Y. Huang, X.-P. Zhang, and X. Ding, "A weighted variational model for simultaneous reflectance and illumination estimation," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.2782–2790, 2016.

[33] X. Guo, "LINE: a method for low-light image enhancement," ACM International Conference on Multimedia, pp.87–91, 2016.

[34] X. Guo, Y. Li, and H. Ling, "LIME: Low-light image enhancement via illumination map estimation," IEEE Transactions on Image Processing (TIP), vol.26, no.2, pp.982–993, 2016.

[35] M. Li, J. Liu, W. Yang, X. Sun, and Z. Guo, "Structure-revealing low-light image enhancement via robust retinex model," IEEE Transactions on Image Processing (TIP), vol.27, no.6, pp.2828–2841, 2018.

[36] X. Ren, W. Yang, W.-H. Cheng, and J. Liu, "LR3M: Robust low-light enhancement via low-rank regularized retinex model," IEEE Transactions on Image Processing (TIP), vol.29, pp.5862–5876, 2020.

[37] C. Wei, W. Wang, W. Yang, and J. Liu, "Deep retinex decomposition for low-light enhancement," arXiv preprint arXiv:1808.04560, 2018.

[38] Y. Zhang, J. Zhang, and X. Guo, "Kindling the darkness: A practical low-light image enhancer," ACM International Conference on Multimedia, pp.1632–1640, 2019.

[39] Y. Zhang, X. Guo, J. Ma, W. Liu, and J. Zhang, "Beyond brightening low-light images," International Journal of Computer Vision (IJCV), vol.129, pp.1013–1037, 2021.

[40] W. Wu, J. Weng, P. Zhang, X. Wang, W. Yang, and J. Jiang, "Uretinex-net: Retinex-based deep unfolding network for low-light image enhancement," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.5901–5910, 2022.

[41] Z. Zhang, H. Zheng, R. Hong, M. Xu, S. Yan, and M. Wang, "Deep color consistent network for low-light image enhancement," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.1899–1908, 2022.

[42] K.G. Lore, A. Akintayo, and S. Sarkar, "LLNet: A deep autoencoder ap-

proach to natural low-light image enhancement," Pattern Recognition, vol.61, pp.650–662, 2017.

[43] C. Li, J. Guo, F. Porikli, and Y. Pang, "LightenNet: A convolutional neural network for weakly illuminated image enhancement," Pattern Recognition Letters, vol.104, pp.15–22, 2018.

[44] W. Ren, S. Liu, L. Ma, Q. Xu, X. Xu, X. Cao, J. Du, and M.-H. Yang, "Low-light image enhancement via a deep hybrid network," IEEE Transactions on Image Processing (TIP), vol.28, no.9, pp.4364–4375, 2019.

[45] Z. Cui, K. Li, L. Gu, S. Su, P. Gao, Z. Jiang, Y. Qiao, and T. Harada, "Illumination adaptive transformer," arXiv preprint arXiv:2205.14871, 2022.

[46] T. Wang, K. Zhang, T. Shen, W. Luo, B. Stenger, and T. Lu, "Ultra-high-definition low-light image enhancement: A benchmark and transformer-based method," AAAI Conference on Artificial Intelligence, vol.37, pp.2654–2662, 2023.

[47] Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen, J. Yang, P. Zhou, and Z. Wang, "EnlightenGAN: Deep light enhancement without paired supervision," IEEE Transactions on Image Processing (TIP), vol.30, pp.2340–2349, 2021.

[48] C. Guo, C. Li, J. Guo, C.C. Loy, J. Hou, S. Kwong, and R. Cong, "Zero-reference deep curve estimation for low-light image enhancement," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.1780–1789, 2020.

[49] C. Li, C. Guo, and C.C. Loy, "Learning to enhance low-light image via zero-reference deep curve estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol.44, no.8, pp.4225–4238, 2021.

[50] L. Zhang, L. Zhang, X. Liu, Y. Shen, S. Zhang, and S. Zhao, "Zero-shot restoration of back-lit images using deep internal learning," ACM International Conference on Multimedia, pp.1623–1631, 2019.

[51] A. Zhu, L. Zhang, Y. Shen, Y. Ma, S. Zhao, and Y. Zhou, "Zero-shot restoration of underexposed images via robust retinex decomposition," IEEE International Conference on Multimedia and Expo (ICME), pp.1–6, 2020.

[52] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," IEEE Transactions on

Image Processing (TIP), vol.16, no.8, pp.2080–2095, 2007.

[53] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," IEEE Transactions on Image Processing (TIP), vol.26, no.7, pp.3142–3155, 2017.

[54] L. Wang, Y. Li, and S. Wang, "DeepDeblur: fast one-step blurry face images restoration," arXiv preprint arXiv:1711.09515, 2017.

[55] S.W. Zamir, A. Arora, S. Khan, M. Hayat, F.S. Khan, M.-H. Yang, and L. Shao, "Learning enriched features for fast image restoration and enhancement," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol.45, no.2, pp.1934–1948, 2022.

[56] M.S. Sajjadi, B. Scholkopf, and M. Hirsch, "Enhancenet: Single image super-resolution through automated texture synthesis," IEEE International Conference on Computer Vision (ICCV), pp.4491–4500, 2017.

[57] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A.A. Efros, "Context encoders: Feature learning by inpainting," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.2536–2544, 2016.

[58] N. Kanopoulos, N. Vasanthavada, and R.L. Baker, "Design of an image edge detection filter using the sobel operator," IEEE Journal of solid-state circuits, vol.23, no.2, pp.358–367, 1988.

[59] A. Huertas and G. Medioni, "Detection of intensity changes with subpixel accuracy using laplacian-gaussian masks," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol.PAMI-8, no.5, pp.651–664, 1986.

[60] G. Zhai and X. Min, "Perceptual image quality assessment: a survey," Science China Information Sciences, vol.63, pp.1–52, 2020.

[61] X. Min, K. Gu, G. Zhai, X. Yang, W. Zhang, P. Le Callet, and C.W. Chen, "Screen content quality assessment: overview, benchmark, and beyond," ACM Computing Surveys (CSUR), vol.54, no.9, pp.1–36, 2021.

[62] 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 (TIP), vol.13, no.4, pp.600–612, 2004.

[63] E.C. Larson and D.M. Chandler, "Most apparent distortion: full-reference image quality assessment and the role of strategy," Journal of Electronic Imaging, vol.19, no.1, pp.011006–011006, 2010.

[64] L. Zhang, Y. Shen, and H. Li, "VSI: A visual saliency-induced index for perceptual image quality assessment," IEEE Transactions on Image Processing (TIP), vol.23, no.10, pp.4270–4281, 2014.

[65] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," IEEE Transactions on Image Processing (TIP), vol.20, no.8, pp.2378–2386, 2011.

[66] W. Xue, L. Zhang, X. Mou, and A.C. Bovik, "Gradient magnitude similarity deviation: A highly efficient perceptual image quality index," IEEE Transactions on Image Processing (TIP), vol.23, no.2, pp.684–695, 2013.

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

[68] M. Cheon, S.-J. Yoon, B. Kang, and J. Lee, "Perceptual image quality assessment with transformers," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.433–442, 2021.

[69] G. Zhai, W. Sun, X. Min, and J. Zhou, "Perceptual quality assessment of low-light image enhancement," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol.17, no.4, pp.1–24, 2021.

[70] A. Mittal, R. Soundararajan, and A.C. Bovik, "Making a "completely blind" image quality analyzer," IEEE Signal Processing Letters, vol.20, no.3, pp.209–212, 2012.

[71] A. Mittal, A.K. Moorthy, and A.C. Bovik, "No-reference image quality assessment in the spatial domain," IEEE Transactions on Image Processing (TIP), vol.21, no.12, pp.4695–4708, 2012.

[72] K. Gu, G. Zhai, X. Yang, and W. Zhang, "Using free energy principle for blind image quality assessment," IEEE Transactions on Multimedia (TMM), vol.17, no.1, pp.50–63, 2014.

[73] S. Yang, T. Wu, S. Shi, S. Lao, Y. Gong, M. Cao, J. Wang, and Y. Yang, "MANIQA: Multi-dimension attention network for no-reference image quality assessment," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.1191–1200, 2022.

[74] Z. Zhang, W. Sun, X. Min, W. Zhu, T. Wang, W. Lu, and G. Zhai, "A no-reference evaluation metric for low-light image enhancement," IEEE Inter-

national Conference on Multimedia and Expo (ICME), pp.1–6, 2021.

[75] R.A. Yeh, C. Chen, T. Yian Lim, A.G. Schwing, M. Hasegawa-Johnson, and M.N. Do, "Semantic image inpainting with deep generative models," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.5485–5493, 2017.

[76] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T.S. Huang, "Generative image inpainting with contextual attention," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.5505–5514, 2018.

[77] G. Liu, F.A. Reda, K.J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," European Conference on Computer Vision (ECCV), pp.85–100, 2018.

[78] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T.S. Huang, "Free-form image inpainting with gated convolution," IEEE International Conference on Computer Vision (ICCV), pp.4471–4480, 2019.

[79] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3431–3440, 2015.

[80] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol.39, no.12, pp.2481–2495, 2017.

[81] 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 (MICCAI), pp.234–241, 2015.

[82] Z. Zhou, M.M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (DLMIA ML-CDS), pp.3–11, 2018.

[83] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.770–778, 2016.

[84] J. Kim, J.K. Lee, and K.M. Lee, "Accurate image super-resolution using very deep convolutional networks," IEEE Conference on Computer Vision and Pat-

tern Recognition (CVPR), pp.1646–1654, 2016.

[85] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," Communications of the ACM, vol.63, no.11, pp.139–144, 2020.

[86] M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv preprint arXiv:1411.1784, 2014.

[87] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," Advances in Neural Information Processing Systems, vol.29, 2016.

[88] A. Creswell and A.A. Bharath, "Task specific adversarial cost function," arXiv preprint arXiv:1609.08661, 2016.

[89] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," arXiv preprint arXiv:1609.03126, 2016.

[90] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," International Conference on Machine Learning (ICML), pp.1060–1069, 2016.

[91] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., "Photo-realistic single image super-resolution using a generative adversarial network," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4681–4690, 2017.

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

[93] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol.33, no.12, pp.2341–2353, 2011.

[94] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," IEEE Transactions on Image Processing (TIP), vol.13, no.9, pp.1200–1212, 2004.

[95] Y.-L. Liu, W.-S. Lai, M.-H. Yang, Y.-Y. Chuang, and J.-B. Huang, "Learning to see through obstructions with layered decomposition," IEEE transactions on pattern analysis and machine intelligence, vol.44, no.11, pp.8387–8402, 2021.

[96] S. Tsogkas, F. Zhang, A. Jepson, and A. Levinshtein, "Efficient flow-guided

multi-frame de-fencing," IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp.1838–1847, 2023.

[97] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," SIAM Journal on Imaging Sciences, vol.2, no.2, pp.438–469, 2009.

[98] S. Jonna, V.S. Voleti, R.R. Sahay, and M.S. Kankanhalli, "A multimodal approach for image de-fencing and depth inpainting," International Conference on Advances in Pattern Recognition (ICAPR), pp.1–6, 2015.

[99] H. Adeel, M.M. Riaz, and S.S. Ali, "De-fencing and multi-focus fusion using markov random field and image inpainting," IEEE Access, vol.10, pp.35992–36005, 2022.

[100] J. Hays, M. Leordeanu, A.A. Efros, and Y. Liu, "Discovering texture regularity as a higher-order correspondence problem," European Conference on Computer Vision (ECCV), pp.522–535, 2006.

[101] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," International Conference on Artificial Intelligence and Statistics (AISTATS), pp.249–256, 2010.

[102] M.S. Gerald Schaefer, "UCID: an uncompressed color image database," Proc. SPIE, vol.5307, pp.5307–5307–9, 2003.
http://dx.doi.org/10.1117/12.525375

[103] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," IEEE International Conference on Computer Vision (ICCV), vol.2, pp.416–423, 2001.

[104] M. Park, K. Brocklehurst, R.T. Collins, and Y. Liu, "Deformed lattice detection in real-world images using mean-shift belief propagation," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol.31, no.10, pp.1804–1816, 2009.

[105] Y. Liu, Z. Wang, Y. Zeng, H. Zeng, and D. Zhao, "PD-GAN: perceptual-details gan for extremely noisy low light image enhancement," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.1840–1844, 2021.

[106] Y. Fu, Y. Hong, L. Chen, and S. You, "LE-GAN: Unsupervised low-light

image enhancement network using attention module and identity invariant loss," Knowledge-Based Systems, vol.240, p.108010, 2022.

[107] X. Wang, Y. Zhai, X. Ma, J. Zeng, and Y. Liang, "Low-light image enhancement based on GAN with attention mechanism and color constancy," Multimedia Tools and Applications, vol.83, no.1, pp.3133–3151, 2024.

[108] Y. Wang, Y. Yu, W. Yang, L. Guo, L.-P. Chau, A.C. Kot, and B. Wen, "ExposureDiffusion: Learning to expose for low-light image enhancement," IEEE/CVF International Conference on Computer Vision (ICCV), pp.12438–12448, 2023.

[109] X. Yi, H. Xu, H. Zhang, L. Tang, and J. Ma, "Diff-retinex: Rethinking low-light image enhancement with a generative diffusion model," IEEE/CVF International Conference on Computer Vision, pp.12302–12311, 2023.

[110] C. Li, C. Guo, L. Han, J. Jiang, M.-M. Cheng, J. Gu, and C.C. Loy, "Low-light image and video enhancement using deep learning: A survey," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol.44, no.12, pp.9396–9416, 2021.

[111] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," IEEE Transactions on Image Processing (TIP), vol.26, no.7, pp.3142–3155, 2017.

[112] T. Matsui, T. Fujisawa, T. Yamaguchi, and M. Ikehara, "Single-image rain removal using residual deep learning," IEEE International Conference on Image Processing (ICIP), pp.3928–3932, 2018.

[113] X. Mao, Y. Liu, W. Shen, Q. Li, and Y. Wang, "Deep residual fourier transformation for single image deblurring," arXiv preprint arXiv:2111.11745, 2021.

[114] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.

[115] S. Woo, J. Park, J.-Y. Lee, and I.S. Kweon, "CBAM: Convolutional block attention module," European Conference on Computer Vision (ECCV), pp.3–19, 2018.

[116] 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," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1874–1883, 2016.

[117] Z. Wang, E.P. Simoncelli, and A.C. Bovik, "Multiscale structural similarity for image quality assessment," IEEE Asilomar Conference on Signals, Systems & Computers (ACSSC), vol.2, pp.1398–1402, 2003.

[118] J.M.J. Valanarasu, R. Yasarla, and V.M. Patel, "TransWeather: transformer-based restoration of images degraded by adverse weather conditions," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.2353–2363, 2022.

[119] B. Li, X. Liu, P. Hu, Z. Wu, J. Lv, and X. Peng, "All-in-one image restoration for unknown corruption," IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp.17452–17462, 2022.

[120] J. Zhang, J. Huang, M. Yao, Z. Yang, H. Yu, M. Zhou, and F. Zhao, "Ingredient-oriented multi-degradation learning for image restoration," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.5825–5835, 2023.

謝辞

本研究は，著者が慶應義塾大学大学院理工学研究科の後期博士課程在学中に行ったものである．本論文の作成に当たり，幾多のご意見とご指導を賜りました指導教授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心より感謝申し上げます．また，ご多忙な中，本論文の副査を引き受けてくださった慶應義塾大学理工学部の萩原将文教授，青木義満教授，久保亮吾教授に深く御礼申し上げます．