平成 29 年度　修士論文

CNN を用いた画風変換による画像鮮明化 および物体検出精度改善

早稲田大学大学院　基幹理工学研究科　情報理工・情報通信専攻

5116F054－7

高橋　沙季

指導　甲藤 二郎　教授

2018 年 1 月 30 日

指導教授印	受付印

目次

第 1 章　序論 3

1.1　はじめに 3
1.2　研究目的 3
1.3　本論文の構成 4

第 2 章　関連技術 5

2.1　画風変換 5
2.1.1　画風変換 5
2.1.2　画風変換の歴史 6
2.1.3　代表的な画風変換手法① 8
2.1.4　代表的な画風変換手法② 10
2.1.5　Microsoft COCO dataset 13
2.2　物体検出 14
2.2.1　物体検出 14
2.2.2　物体検出の歴史 15
2.2.3　HOG 特徴量を用いた手法 15
2.2.4　CoHOG 特徴量を用いた手法 18
2.2.5　Joint HOG 特徴量を用いた手法 19
2.2.6　R-CNN を用いた手法 20
2.2.7　SSD を用いた手法 21
2.2.8　YOLOv2 を用いた手法 21

第 3 章　一般的な画像鮮明化手法 23

3.1　線形濃度変換を用いた鮮明化手法 23
3.2　非線形濃度変換（ガンマ補正）を用いた鮮明化手法 24
3.3　ヒストグラム平均化を用いた鮮明化手法 26
3.4　ダークチャネルを用いた鮮明化手法 27
3.5　鮮明化機器を用いた鮮明化手法 31
3.6　Local Contrast 強調処理を用いた鮮明化手法 32

第 4 章　画風変換を用いた画像鮮明化の提案 34

4.1　概要 34
4.2　画風変換を用いた画像鮮明化 34

第 5 章　実験.. 36
5.1　概要 36
5.2　実験環境と実験動画..36
5.2.1　実験環境 36
5.2.2　実験使用画像 36
5.3　実験１：画風変換による不鮮明画像の鮮明化..36
5.3.1　HAZE 画像の鮮明化実験結果 37
5.3.2　夜間画像の鮮明化実験結果 38
5.3.3　実験結果のまとめ 39
5.4　実験２：鮮明化後画像を用いた物体検出 40
5.4.1　HAZE 画像の鮮明化画像の物体検出結果..40
5.4.2　夜間画像の鮮明化画像の物体検出結果...41
5.4.3　実験結果のまとめ 42
5.5　実験３：鮮明化後画像の画質評価結果..43
5.5.1　不鮮明画像を基にした鮮明化後画像の画質評価....................................43
5.5.2　本来の鮮明画像（HAZE 無し画像）を基にした鮮明化後画像の画質評価....44
5.6　実験４：スタイル画像の違いが及ぼす実験結果への影響...........................45
5.6.1　画素位置の違いによる影響 45
5.6.2　カラーバリエーションの違いによる影響 48
5.6.3　ヒストグラム分布の違いによる影響 51
第 6 章　総括.. 54
6.1　まとめ..54
6.2　今後の課題..54
謝辞 55
参考文献.. 56
発表文献リスト 59

第1章　序論

1.1 はじめに

近年、機械学習や Deep Learning の研究開発が活発であり、囲碁プログラムの AlphaGo[1]や音声認識など様々な分野に Deep Learning は応用されている。そのような Deep Learning の応用例に画風変換という技術がある。画風変換とは、ある画像に写る物体の配置を保持しながら、その画像の画風を別の画像の画風に似せて変換する技術である。画風変換は 2015 年 9 月に研究発表[2]された比較的新しい分野であり、生成される画像の美しさを楽しむために使用されている。

Deep Learning による恩恵を受けた研究分野は他にも多数存在する。その一例として、物体検出分野が挙げられる。物体検出は、初期の研究では物体画像と背景画像の認識問題として扱われており、Wavelet 特徴量と Support Vector Machine （SVM）の組み合わせによる検出手法が提案されていた[3]。その後、物体の大きさや位置を推定する検出問題へと変わり、2005 年に Dalal らが物体の勾配方向に着目する Histogram of Oriented Gradient （HOG）特徴量と SVM の組み合わせによる検出手法[4]を提案したことで、物体検出精度が更に向上した。しかし、それでもなお物体検出における未検出率や誤検出率には改善の余地があった。それを大幅に改善したのが Deep Learning を用いた手法である。2012 年に ILSVRC と呼ばれる一般物体認識のコンペティションで Krizhevsky らの Deep Learning をベースにした手法[5]が優勝して以降は、Deep Learning の一種である CNN をベースとした物体検出手法が数多く提案されている。

1.2 研究目的

物体検出に関わる研究はおよそ 50 年前の 1960 年代から行われており[6]、現在も研究が活発に行われている。2 年後の 2020 年に東京オリンピックを控えている今、さまざまな分野で高い精度の物体検出技術が求められている。

画像に写る物体の検出精度は Deep Learning を用いた手法[5]が初めて提案された 2012 年以降、飛躍的に向上している。しかし、夜間や悪天候時に撮影した不鮮明画像の場合には、物体検出処理を行う前に画像の鮮明化を行う必要がある。一般的な鮮明化手法として、晴天時の実験データに基づいた仮定を利用した手法[7]や鮮明化機器を用いた手法等が挙げられるが、画像ごとに最適なパラメータ値が異なる場合など、上記の手法では画像の鮮明化効果が得られない場合がある。

そこで本研究では、従来とは異なる鮮明化手法のアプローチとして、画風変換技術を用いた不鮮明画像の鮮明化手法を提案する。本来、画風変換技術は生成される画像の美しさを楽しむために使用される。しかし本研究では、不鮮明画像を晴れの日の昼間に撮影した画像風に変換することを目的として画風変換を使用することにより、不鮮明画像の鮮明化を図る。また、不鮮明画像の鮮明化だけでなく、不鮮明画像に写る物体の検出精度向上も含めて研究目的とする。

実際に画風変換を用いた鮮明化実験を行い、本手法により物体検出精度が向上する例を確認した。

1.3 本論文の構成

本論文は、6 章で構成されている。第 1 章では、研究背景および本研究の目的について述べた。第 2 章では関連技術として、画風変換、物体検出について述べる。第 3 章では不鮮明画像の一般的な鮮明化手法について述べる。第 4 章では画風変換を利用した不鮮明画像の鮮明化手法を述べ、第 5 章ではその有効性を示すための実験について述べる。そして、第 6 章ではまとめを述べる。

第2章　　関連技術

2.1 画風変換

2.1.1　画風変換

画風変換は､2015 年 9 月に Gatys 氏とその研究グループによって発表された[2]研究分野である。図 1 のように、画風変換処理を用いることにより、ある画像（コンテンツ画像）の物体の配置を保持しながらその画像の画風を別の画像（スタイル画像）に似せて変換することが可能となる。Gatys 氏らが提案した画風変換モデルでは、Deep Learning の一種である CNN（Convolutional Neural Network）を用いており、誤差逆伝播法を用いて出力画像を少しずつ修正しながら画風変換後画像を生成する。

コンテンツ画像　　スタイル画像　　出力画像

図 1　画風変換例[8]

一般的な画風変換モデルの一つとして Johnson 氏らが 2016 年 3 月に発表したモデル[9]が挙げられる。この画風変換モデルでは、事前に数万枚の学習用画像を用いて画風変換ネットワークを学習させることにより、任意の画像の高速な画風変換が可能となる。この点が、画風変換の提唱者である Gatys 氏らの手法との大きな相違点である。

画風変換ネットワークの学習には､1 枚のスタイル画像と数万枚の学習用コンテンツ画像を用いており､Johnson 氏らは学習用コンテンツ画像に Microsoft COCO dataset[10]の約 8 万枚の画像を使用している。また､画風変換ネットワークはスタイル画像ごとに学習する必要があり、1 枚につき学習時間は 18 時間程度である。しかし、一度画風変換ネットワークを学習すれば､任意の画像を画風変換ネットワークに入力するだけで高速な画風変換が可能となる。実際に変換速度は Gatys 氏らの手法と比べて 1000 倍程度[9]とされている。

本章は文献[高橋沙季 et.: "CNN を用いた画風変換による不鮮明画像の改善"]及び文献[高橋沙季 et.: ”CNN を用いた画風変換による物体検出精度の改善"]に基づいている

2.1.2　画風変換の歴史

2015 年 9 月に Gatys 氏らが画風変換について公に発表して以降、画風変換に関する論文が次々と発表されている。表 1 にその代表例を示す。①②の論文に関しては代表的な画風変換手法として次節で述べるとし、③④⑤の論文に関しては以下に簡単にまとめる。

表 1　画風変換に関する論文の代表例

	発表年月	論文タイトル	第一著者
①	2015.9	A Neural Algorithm of Artistic Style [2]	Leon A.Gatys
②	2016.3	Perceptual Losses for Real-Time Style Transfer and Super-Resolution [9]	Justin Johnson
③	2016.7	Preserving Color in Neural Artistic Style Transfer [11]	Leon A. Gatys
④	2016.10	Artistic Style Transfer for Videos [12]	Manuel Ruder
⑤	2017.4	Deep Photo Style Transfer [13]	Fujun Luan

論文①： A Neural Algorithm of Artistic Style (Leon A.Gatys, 2015.9)
論文②： Perceptual Losses for Real-Time Style Transfer and Super-Resolution (Justin Johnson , 2016.3)

これらの論文の画風変換手法に関しては、次節で詳細に述べるとする。

論文③： Preserving Color in Neural Artistic Style Transfer [11] (Leon A.Gatys, 2016.7)

この論文では、画像の画風変換をするにあたって変換前後で画像の色味を保つような変換手法を提案している。通常の画風変換では、変換後画像の色味はスタイル画像の色味にも影響を受けるが、本手法ではコンテンツ画像の色味のみが出力画像に表れる。

本論文では以下の 2 種類の手法を提案している。これらの 2 手法を用いた場合の画風変換例を以下の図 2 に示す。

・手法１　Color histogram matching 手法

スタイル画像の色ヒストグラムをコンテンツ画像の色に近づけ、色ヒストグラムを近づけた画像を新たなスタイル画像として画風変換を行う手法[14]

・手法２　Luminance-only transfer 手法

スタイル画像とコンテンツ画像から輝度を抽出して輝度空間のみの画風変換を行い、カラーチャンネルを画風変換の出力に後付けする手法[15]

コンテンツ画像

スタイル画像

色ヒストグラムを近づけた
スタイル画像

Color histogram matching 手法による
出力画像

Luminance only transfer 手法による
出力画像

図 2　論文③の Color histogram matching 手法および Luminance-only transfer 手法を用いた場合の画風変換例［14］

論文④：　Artistic style Transfer for videos [12] (Manuel Ruder, 2016.10)

この論文は、Gatys 氏らの画風変換手法[2]を動画に適用するためのアルゴリズムを提案している。動画を静止画の連続と捉えて、1 フレームごとに Gatys 氏らの画風変換手法を適用するだけでは不自然な切れ目が出てしまう。そのため、このアルゴリズムではオプティカルフローを使用することにより動画をスムーズに見れるよう工夫している。

論文⑤：　Deep Photo Style Transfer [13] (Fujun Luan , 2017.4)

この論文は、Gatys 氏らの画風変換手法[2]と比較してよりフォトリアリスティックな出力画像が得られる手法を提案している。Gatys 氏らの手法では、コンテンツ画像やスタイル画像に写真を用いた場合でも、出力画像に歪みが発生し写真に見えないという問題点が存在する。本手法では、画風変換する際に色変換の局所的なアフィンのみ変換するように制限することで、フォトリアリスティックな画像の出力を可能としている[16]。

本手法を用いた場合の画風変換例を以下の図 3 に示す。

コンテンツ画像　　スタイル画像　　出力画像

図 3 論文⑤の手法を用いた場合の画風変換例 [16]

2.1.3　代表的な画風変換手法①

2.1.2 で述べた画風変換に関する論文①は、画風変換について初めて書かれた論文であり、その後追随した研究に大きな影響を与えている。以下に、論文①の画風変換手法について詳細に述べる。

論文①：A Neural Algorithm of Artistic Style [2] (Leon A.Gatys , 2015.9)

本論文は、初めて画風変換について述べられた論文である。Gatys 氏らが 2015 年 9 月に発表し、CNN を用いて入力画像のコンテンツと別の入力画像の画風を結びつけるような手法を提案している。本論文で提案されている画風変換モデルの概略図を図 4 に示す。このモデルでは以下の流れに従って画像の画風変換を行う。

1. コンテンツ画像（C:Content Image）、スタイル画像（S:Style Image）、損失計算用ネットワーク（Loss Network）を用意する
2. 出力画像（T:Target Image）として、乱数で作成したノイズ画像を用意する
3. Content Image、Style Image、Target Image を Loss Network に入力する
4. Loss Network の中間層における Content Image と Target Image の損失（Content Loss）を計算する。損失の計算は後述の式に従って行う。
5. Loss Network の中間層における Style Image と Target Image の損失（Style Loss）を計算する。損失の計算は後述の式に従って行う。
6. Content Loss と Style Loss の和(Total Loss)を損失関数とし、損失関数を最小化する方向に Target Image を少しずつ更新する（逆伝播による勾配降下）
7. Total Loss が閾値以下になるまで 3〜7 を繰り返す
8. Total Loss が閾値以下になった時点での Target Image を画風変換後画像とする

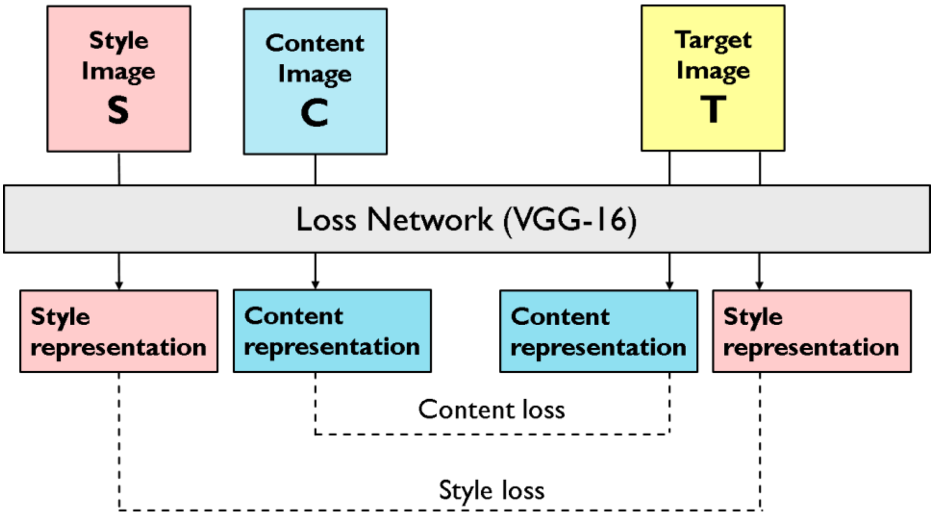


図 4　論文①の画風変換モデルの概略図［17］

なお上記の損失計算ネットワークには VGG-16[18]を使用する。

また、上記の手順 4、5、6 で計算する各損失に関しては、Gatys 氏らの論文[2]より以下の式で計算することとする。

・中間層における Content Image と Target Image の損失(Content Loss)の計算式

$$L_{content} = \frac{1}{2}\sum_{i,j}\left(F_{ij}^{l} - P_{ij}^{l}\right)^2 \qquad \text{式(1)}$$

F_{ij}^{l}: Content Image を入力した時のレイヤlの値

P_{ij}^{l}: Target Image を入力した時のレイヤlの値

・中間層における Style Image と Target Image の損失(Style Loss)の計算式

$$E = \frac{1}{4{N_l}^2{M_l}^2}\sum_{i,j}\left(G_{ij}^{l} - A_{ij}^{l}\right)^2 \qquad \text{式(2)}$$

$$L_{style} = \sum_{l=0}^{L}\omega_l E_l \qquad \text{式(3)}$$

N_l: レイヤlにおけるフィルタの数

M_l:レイヤlにおけるフィルタサイズ

G_{ij}^{l}: Style Image を入力した時のレイヤlの値

A_{ij}^{l}: Target Image を入力した時のレイヤlの

ω_l: レイヤl における重み

L: レイヤ数

・中間層における損失(Total Loss)の計算式

$$L_{total} = L_{content} + L_{style} \qquad \text{式(4)}$$

　論文①の上記の処理手順に従って画風変換を行った結果を以下の図 5 に示す。図 5 の入力コンテンツ画像は猫の画像、入力スタイル画像はゴッホが描いた星月夜の画像である。図 5 のように、これらの画像を画風変換モデルに入力すると、出力画像としてゴッホの星月夜に似せた猫の画像が得られる。

この画風変換モデルをプログラムした一連のコードは”chainer-gogh”という名で GitHub 上に公開されている[19]。図 5 は実際に chainer-gogh を用いて画風変換処理を行った結果である。

コンテンツ画像

スタイル画像

出力画像

図 5　論文①の画風変換モデルを用いた画風変換例[19]

2.1.4　　代表的な画風変換手法②

2.1.2 で述べた画風変換に関する論文②は、2.1.3 で述べた論文①の手法と比べて約 1000 倍高速な画風変換手法である。以下に、論文②の画風変換手法について詳細にまとめる。

論文②： Perceptual Losses for Real-Time Style Transfer and Super-Resolution[9] (Justin Johnson , 2016.3)

この論文は 2016 年 3 月に Johnson 氏らによって発表されたものであり、論文①の画風変換手法の高速化を図った手法を提案している。事前に数万枚の学習用コンテンツ画像と 1 枚のスタイル画像を用いて、入力画像を特定の画風に変換する画風変換ネットワークを学習することにより、論文①の手法と比べてより高速な画風変換が可能となる。本論文で紹

介されている画風変換モデルの概略図を図 6 に示す。このモデルでは以下の流れに従って画像の画風変換を行う。

1. 数万枚の学習用コンテンツ画像（ここでは Microsoft COCO dataset に含まれる 8 万枚の画像を使用）、スタイル画像（S:Style Image）、損失計算用ネットワーク（Loss Network）、画風変換ネットワーク（Image Transformation Network）を用意する
2. 学習用コンテンツ画像から 1 枚選び、コンテンツ画像（C: Content Image）とする
3. 出力画像(O: Output Image)として乱数で作成したノイズ画像を用意する
4. Content Image、Style Image、Output Image を Loss Network に入力する
5. Loss Network の中間層における Content Image と Target Image の損失（Content Loss）を計算する。損失の計算は後述の式に従って行う。
6. Loss Network の中間層における Style Image と Target Image の損失（Style Loss）を計算する。損失の計算は後述の式に従って行う。
7. Content Loss と Style Loss の和(Total Loss)を損失関数とし、損失関数を最小化する方向に Image Transformation Network の重み付けを更新する
8. Total Loss が閾値以下になるまで 3～7 を繰り返す
9. Total Loss が閾値以下になったら、2.で次のコンテンツ画像を選び 3~8 を繰り返す
10. 数万枚の学習用コンテンツ画像に対して 9 を繰り返し、Image Transformation Network の学習を行う
11. 任意のコンテンツ画像を学習済みの Image Transformation Network に入力すると画風変換後画像が高速に得られる

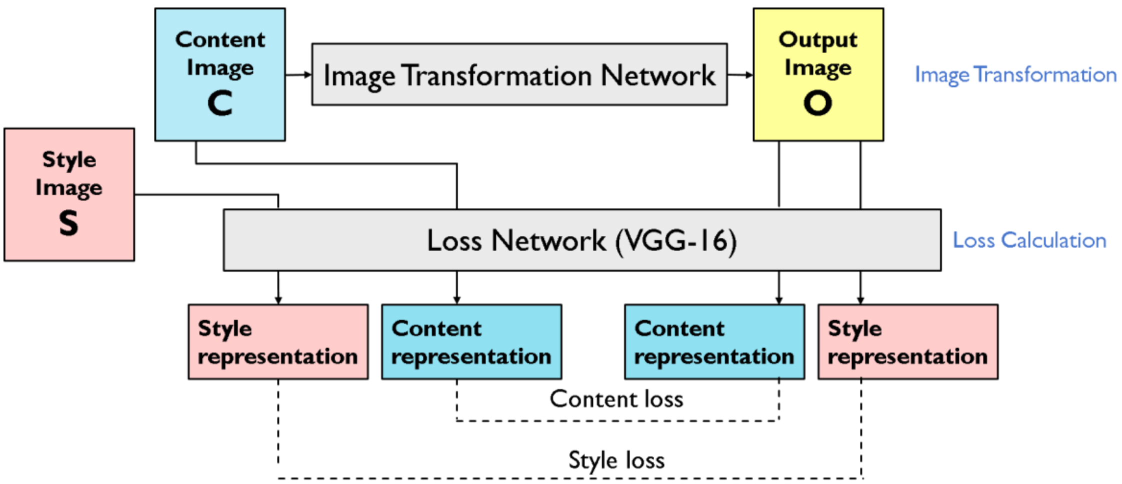


図 6　論文②の画風変換モデルの概略図[20]

なお上記の損失計算ネットワークには、論文①と同様に VGG-16 を使用する。画風変換ネットワークには、畳み込みと活性化関数 ReLU の 3 層＋5 つの Residual block+畳み込みと活性化関数 ReLU の 3 層からなるニューラルネットワークを用いる[20]。

また、上記の手順 5、6、7 で計算する損失に関しては、Johnson 氏らの論文[9]より以下の計算式で計算することとする。

・中間層における Content Image と Target Image の損失(Content Loss)の計算式

$$L_{content}^{\emptyset,j}(\hat{y},y_c)=\frac{1}{C_jH_jW_j}\left\|\emptyset_j(\hat{y})-\emptyset_j(y_c)\right\|^2 \qquad 式(5)$$

$\hat{y}$: yを Image transformation Network で変換した画像

y_c: コンテンツ画像

y_s:スタイル画像

$C_jH_jW_j$: レイヤ j のチャンネル数、高さ、幅

$\emptyset_j$: Loss Network に x を入力した時のレイヤ j の値

・中間層における Style Image と Target Image の損失(Style Loss)の計算式

$$G_j^{\emptyset}(x)_{c,d}=\frac{1}{C_jH_jW_j}\sum_{h=1}^{H_j}\sum_{w=1}^{W_j}\emptyset_j(x)_{h,w,c}\,\emptyset_j(x)_{h,w,c'} \qquad 式(6)$$

$$L_{style}^{\emptyset,j}(\hat{y},y_s)=\left\|G_j^{\emptyset}(\hat{y})-G_j^{\emptyset}(y_s)\right\|^2 \qquad 式(7)$$

$\hat{y}$: yを Image transformation Network で変換した画像

y_c: コンテンツ画像

y_s:スタイル画像

$C_jH_jW_j$: レイヤ j のチャンネル数、高さ、幅

$\emptyset_j$: Loss Network に x を入力した時のレイヤ j の値

c: あるチャンネル

c':別のチャンネル

・中間層における損失(Total Loss)の計算式

$$L_{total}=L_{content}+L_{style} \qquad 式(8)$$

論文②の上記の処理手順に従って画風変換を行った結果を以下の図 7 に示す。図 7 の入力コンテンツ画像は猫の画像、入力スタイル画像はゴッホが描いた星月夜の画像である。出力画像として、ゴッホの星月夜風の猫の画像が得られている。

このアルゴリズムを用いた一連のコードは”chainer-fast-neuralstyle”という名で GitHub 上に公開されている[8]。図 7 は実際に chainer-fast-neuralstyle を用いて画風変換を行った結果である。

コンテンツ画像　　スタイル画像　　出力画像

図 7　論文②の画風変換モデルを用いた画風変換例[8]

2.1.3 の論文①の画風変換モデルを用いた場合の変換結果（図 5）と 2.1.4 の論文②の画風変換モデルを用いた場合の変換結果（図 7）を比較すると、論文②の手法を用いた場合のほうがより鮮明な変換後画像が得られている。論文②では論文①と異なる損失関数を使用しており、損失関数をより細かく忠実に定義している。そのため、Total Loss（Style Loss と Content Loss の和）の減少に繋がり、このような結果になったのではないかと考える。

また、Johnson 氏らの画風変換モデルでは上述の手順からもわかる通り、画風を変換するニューラルネットワーク（画風変換ネットワーク）を事前に学習する必要がある。実際に学習を行ったところ、学習所要時間はスタイル画像 1 枚につきおよそ 18 時間程度であった。この学習所要時間は学習コンテンツ画像の枚数や学習スタイル画像のサイズ等により増減する。スタイル画像ごとに画風変換ネットワークを学習させる必要があり、1 枚につき十数時間かかるのは難点であるが、ひとたび学習を終えれば任意の画像の画風変換を高速に行える点がこの画風変換モデルの最大の特徴である。

2.1.5　Microsoft COCO dataset

2.1.4 で述べた通り、画風変換ネットワークの学習時には数万枚の学習用コンテンツ画像を必要とする。Johnson 氏らは、学習用コンテンツ画像に Microsoft COCO dataset[10]を使用している。実際にデータセットの一例を以下の図 8 に示す。また、Microsoft COCO dataset に含まれる夜間画像例を図 9 に示す。

Microsoft COCO dataset は画像の認識、セグメンテーション、キャプショニングがされているデータセットであり、82783 枚の画像が含まれている[10]。本データセットは Microsoft COCO のホームページにて無償で公開されている。

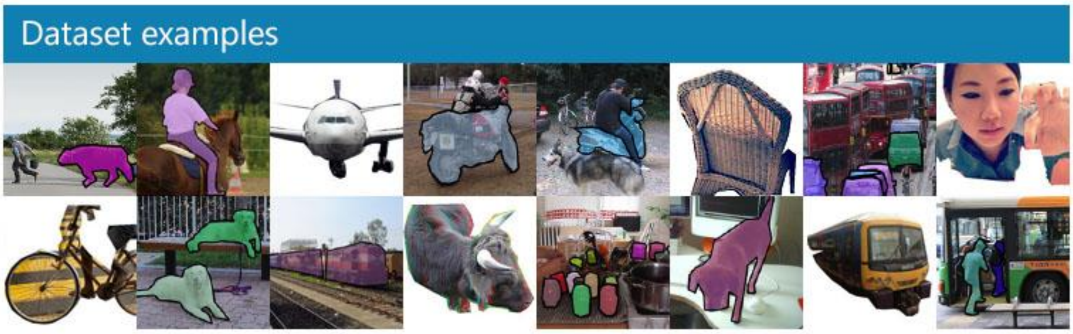


図 8　Microsoft COCO dataset のデータセット例[10]

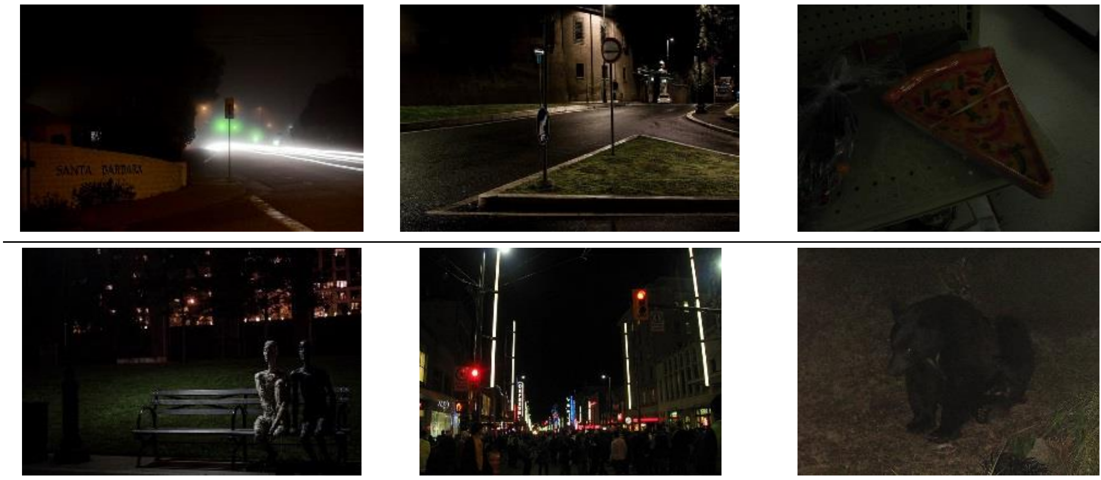


図 9　Microsoft COCO dataset の夜間画像例[10]

2.2 物体検出

2.2.1　　物体検出

物体検出に関わる研究はおよそ 50 年前の 1960 年代から行われており[6]、現在も活発に研究がおこなわれている。2 年後の 2020 年に東京オリンピック・パラリンピックも控えていることから、現在はさまざまな分野で高い精度の物体検出技術が求められている。不審物検出によるテロ防止や歩行者・車両検出による自動運転の実現などがその一例である。

2.2.2　物体検出の歴史

物体検出に関わる研究は 1960 年代から始まり、初期の研究では、物体検出は切り出した物体画像と背景画像の認識問題として扱われていた。その実現方法は、Wavelet 特徴量と Support Vector Machine （SVM）の組み合わせによるものであった[3]。その後、物体の大きさや位置を推定する検出問題へと変わり、2005 年に Dalal らが物体の勾配方向に着目する Histogram of Oriented Gradient（HOG）特徴量と SVM の組み合わせを提案[4]したことで、物体検出の性能を大幅に向上させた。

その後は Co-HOG 特徴量や Joint HOG 特徴量等の HOG の派生である特徴量が提案されたが、2012 年に物体検出手法は大きな転換期を迎える。それは 2012 年に ILSVRC と呼ばれる一般物体認識のコンペティションで、Krizhevsky らの Deep Learning を用いた手法[5]が優勝し、これまでの手法とは全く異なる Deep Learning を用いた手法の有効性が示されたためである。それ以降は Deep Learning の一種である CNN をベースとした手法が数多く提案されている。

以下に代表的な人物検出手法である HOG 特徴量を用いた手法、Co-HOG 特徴量を用いた手法、Joint HOG 特徴量を用いた手法、R-CNN を用いた手法、SSD を用いた手法、YOLOv2 を用いた手法についてそれぞれ簡単に述べる。

2.2.3　HOG 特徴量を用いた手法

HOG 特徴量は、画像の局所領域における輝度の勾配方向の分布を表す特徴量である。HOG 特徴量の HOG とは Histogram of oriented gradients の略であり、直訳すると勾配方向ヒストグラムである。HOG 特徴量は輝度値の変化に影響されにくい特徴量である。HOG 特徴量は、人物や車両等を画像から検出する際に用いられることが多い[21]。HOG 特徴量を用いた人物検出結果例を図 10 に示す。

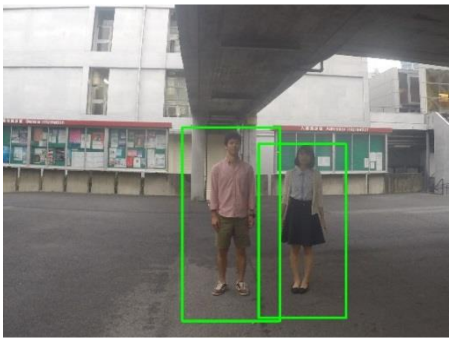


図 10　HOG 特徴量を用いた人物検出結果例

また、2.2.3.1 に HOG 特徴量の算出方法について、2.2.3.2 に HOG 特徴量の描画について簡単に述べる。

2.2.3.1　HOG 特徴量の算出方法

HOG 特徴量を算出するためには、画像から勾配の強さ及び方向を計算し、それらの情報から輝度の勾配方向ヒストグラムを作成し、正規化をブロックごとに行う。各ステップの詳細を以下にまとめる。

(ⅰ)各ピクセルの輝度から勾配方向・勾配強度を計算

勾配強度と勾配方向は、次に示す式(9)～式(12)から算出することができる。式中における$m(x,y)$は点(x,y)における勾配強度、$\theta(x,y)$は点(x,y)における勾配方向、$L(x,y)$は点(x,y)における輝度値をそれぞれ示している。[21]

$$m(x,y)=\sqrt{L_x(x,y)^2+L_y(x,y)^2} \quad \text{式(9)}$$

$$\theta(x,y)=\tan^{-1}\frac{L_y(x,y)}{L_x(x,y)} \quad \text{式(10)}$$

$$L_x(x,y)=L(x+1,y)-L(x-1,y) \quad \text{式(11)}$$

$$L_y(x,y)=L(x,y+1)-L(x,y-1) \quad \text{式(12)}$$

(ⅱ)勾配方向を 9 方向に分類してセルごとに勾配方向ヒストグラムを作成

ステップ(ⅰ)で算出された勾配強度mと勾配方向θを用いて、$a\times a$ ピクセルを 1 セルとした領域において輝度勾配方向ヒストグラムを作成する。この勾配方向は0から180度を20度ずつに分割した 9 方向である。輝度勾配方向を表した画像を図 11 に示す。

また、輝度勾配方向を 9 方向に分類し、セルごとに勾配方向ヒストグラムを作成する。勾配方向ヒストグラムの例を図 12 に示す。

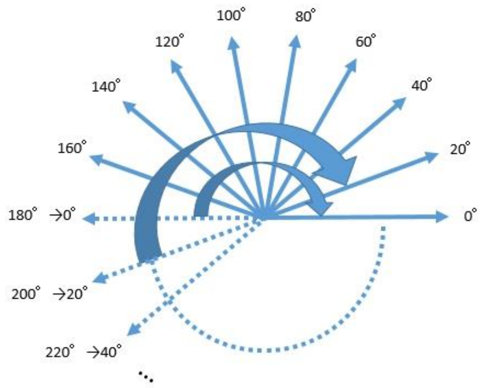


図 11　輝度勾配方向(9 方向)[21]

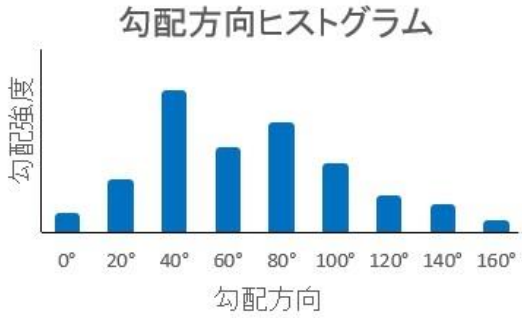


図 12　勾配方向ヒストグラム[21]

(ⅲ)ブロックごとにヒストグラムを正規化したものを特徴量とする

ステップ(ⅱ)で作成した輝度の勾配方向ヒストグラムを 1 ブロック$b \times b$セルとして正規化する。正規化は、ブロックを 1 セルずつ移動させることによって行う。i行j列のセルの特徴量をF_{ij}、k番目のブロックの特徴ベクトルを$\boldsymbol{V}_k$、ブロック内で位置(i,j)にあるセルのヒストグラムをh_{ij}、正規化後のヒストグラムをh'_{ij}としたとき、正規化の式は次式(13)~(15)で表すことができる。また、分母のϵは分母がゼロになり割り切れなくなることを避けるための定数である。[21]

$$F_{ij} = [f_1, f_2, f_3, \dots, f_9] \quad \text{式(13)}$$

$$\boldsymbol{V}_k = \left[F_{ij}, F_{i+1\ j}, F_{i+2\ j}, F_{i\ j+1}, \dots, F_{i+2\ j+2}\right] \quad \text{式(14)}$$

$$h'_{ij} = \frac{h_{ij}}{\sqrt{\|\boldsymbol{V}_k\|^2 + \epsilon^2}} \quad (\epsilon = 1) \quad \text{式(15)}$$

勾配方向が 9 方向であるためF_{ij}は 9 次元であり、$\boldsymbol{V}_k$は 9 b^2次元($b \times b \times 9$, $b \times b$は 1 ブロックあたりのセル数、9 はF_{ij}の次元数)となる。

2.2.3.2　　HOG 特徴量の描画

2.2.3.1 では HOG 特徴量の算出方法について述べた。2.2.3.2 では実際に画像の HOG 特徴量を描画したものを示す。図 13 は原画像、図 14～図 16 は原画像の HOG 特徴量を描画した画像である。HOG 特徴量を計算するにあたって、図 14、図 15、図 16 ではそれぞれ 1 セルを 30×30 ピクセル、20×20 ピクセル、10×10 ピクセルとしている。

図 13　原画像

図 14　HOG 特徴量を描画した画像
（1cell=30×30 pixel、1block＝3×3 cell）

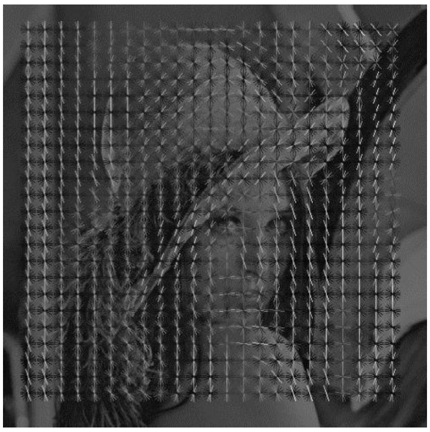


図 15　HOG 特徴量を描画した画像
（1cell=20×20 pixel、1block＝3×3 cell）

図 16　HOG 特徴量を描画した画像
（1cell=10×10 pixel、1block＝3×3 cell）

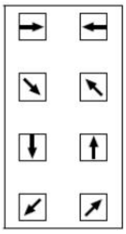
図 14〜図 16 の通り、確かに輝度の差分がある方向に線が描画されているのが確認できる。これらの画像を比較すると、1 セルあたりのピクセル数が小さいほど、より細かく輝度勾配を表現できていることがわかる。

2.2.4　Co-HOG 特徴量を用いた手法

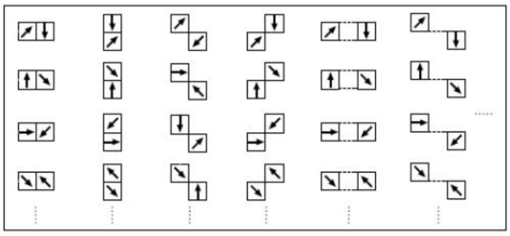
Co-HOG 特徴量（Co-occurrence HOG）[22]は東芝の渡辺友樹氏らが 2009 年に提案した HOG 特徴量をベースとして派生した特徴量である。ピクセルごとの勾配情報だけでなく、周囲のピクセルを考慮した勾配共起も記述した特徴量である。

Co-HOG 特徴量は、輝度値の勾配情報をピクセルごとに算出したのちに勾配方向を 8 方向に分類し、勾配方向を周りのピクセルの勾配方向と組み合わせる[23]。このようにして勾

配方向の共起を扱うことにより、より詳細な情報を持つ特徴量となる。8 方向に分類した勾配方向および共起ペアについて図 17 に示す。



(a)勾配方向（8 方向）



(b)共起ペア

図 17　CoHOG 特徴量における勾配量子［22］

Co-HOG 特徴量では勾配方向の共起を行列として扱う。勾配方向は 8 方向あるため、共起行列は 8×8 となる。図 18 に Co-HOG 特徴量の共起行列を示す。

また、Co-HOG 特徴量は画像を複数のブロックに分割し、各ブロックの共起行列を計算して並べてベクトル化したものである。図 19 に Co-HOG 特徴量のイメージ図を示す。

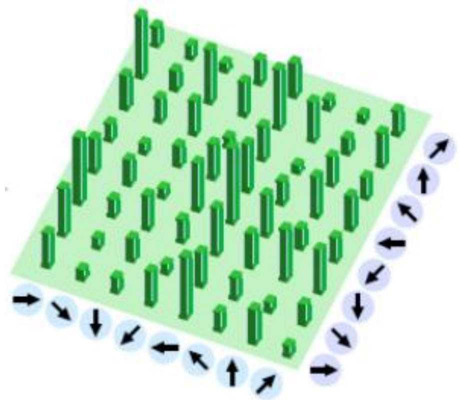
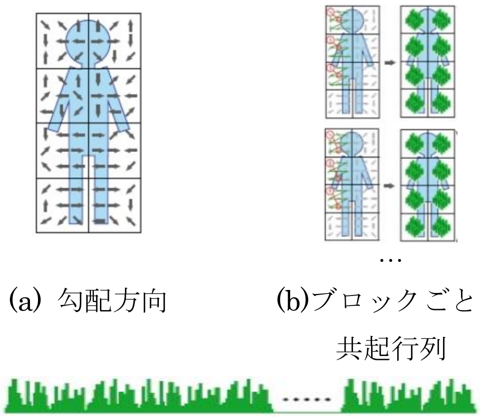


図 18　Co-HOG 特徴量の共起行列［22］



(a) 勾配方向　(b)ブロックごとの共起行列

(c)画像全体の共起行列

図 19　Co-HOG 特徴量のイメージ図[22]

2.2.5　Joint HOG 特徴量を用いた手法

Joint HOG 特徴量は、HOG 特徴量および Real AdaBoost を組み合わせた特徴量である[24]。Joint 特徴量の最大の特徴は、1 つの弱識別器で 2 つの領域の HOG 特徴量を用いて識別を行う点である。HOG 特徴量では 1 つの弱識別器で HOG 特徴量は 1 つまでしか用い

ることができなかった。Joint HOG 特徴量ではこれにより、物体が対称な物体かそうでないかなど、従来の HOG 特徴量では捉えられなかった特徴を把握することができる［25]。

図 20 に Joint HOG 特徴量を用いた物体検出の概念図を示す。また、参考として図 21 に HOG 特徴量を用いた物体検出の概念図を示す。

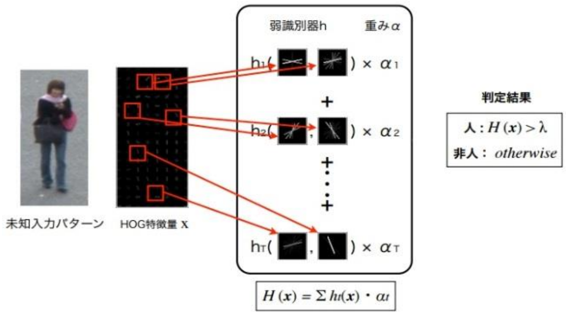


図 20　Joint HOG 特徴量を用いた物体検出の概念図［26]

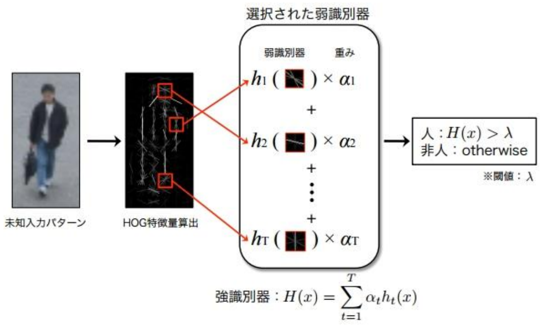


図 21　HOG 特徴量を用いた物体検出の概念図［26]

2.2.6　R-CNN を用いた手法

R-CNN を用いた物体検出では、事前に Selective search で物体候補領域を検出し、検出した候補領域を AlexNet 及び VGGNet に入力して物体検出を行う[3]。ここで、Selective search とは、セグメンテーション情報を繰り返しグループ化して物体候補を大まかに細分化し、物体候補領域を特定する手法である。R-CNN は Selective search で検出した領域に対して、リサイズ処理を加えてから CNN に渡すため、候補領域が多ければ多いほど計算コストが高くなるというデメリットがある。この R-CNN の最も計算コストが高い畳み込み処理の回数を減らして高速化を図った CNN が Fast R-CNN である。

Fast R-CNN は、入力画像に対して Selective search で物体候補領域を検出し、画像全体に対して重みフィルタを畳み込み、得られた特徴マップ上に先ほどの物体候補領域の位置を反映して該当する領域の特徴マップのみを切り出して全結合層に入力する[3]。このように、Fast R-CNN は物体候補領域の検出処理と領域認識処理の 2 つに分かれるため、R-CNN よりは高速なものの依然として計算コストが高いという欠点がある。

Faster R-CNN は Fast R-CNN を基に物体候補領域を RPN（Region Proposal Network）により行うモデルである。RPN により、Fast R-CNN ではできなかった物体候補領域検出と領域認識の同時処理が可能になり、より高速な物体検出を行うことができる。Faster R-CNN では Fast R-CNN のように入力画像に対して Selective search で物体候補領域を検出し、画像全体に対して重みフィルタを畳み込んだ後に、得られた特徴マップからアンカ

ーと呼ばれる検出手法を用いて物体検出を行う。アンカーとは、ある領域を中心として複数の形の検出ウィンドウを当てはめて、ラスタキャンする検出手法である[3]。

以下の図 22 に、Fast R-CNN および Faster R-CNN の概念図を示す。

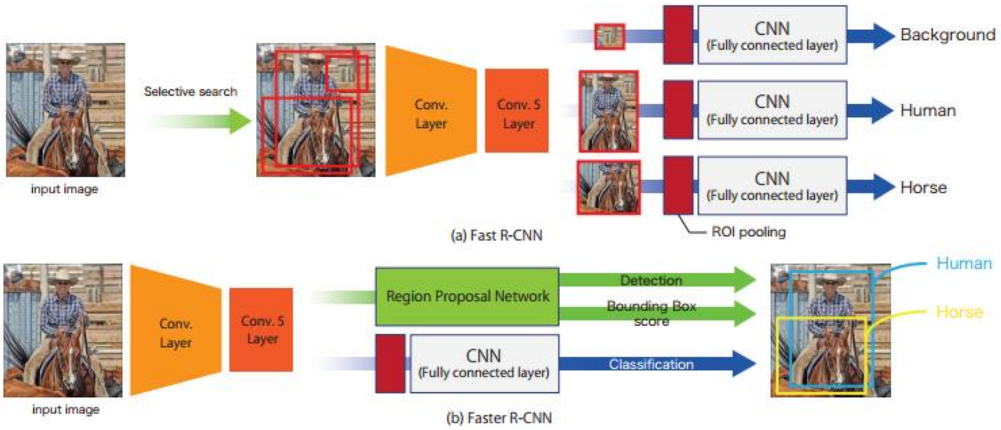


図 22　Fast R-CNN および Faster R-CNN の概念図 [3]

2.2.7　SSD を用いた手法

SSD は Single Shot MultiBox Detector の略である。SSD を用いた手法では、浅い層ではスケールの小さな物体候補を、深い層ではスケールの大きい物体候補を検出している。これは、浅い層であるほどプーリングによる特徴マップの縮小の度合いが小さいことを利用しているためである。また、SSD はエンドトゥエンドで学習可能なネットワークであり、高い検出精度を実現することができる[3]。以下の図 23 に SSD の概念図を示す。

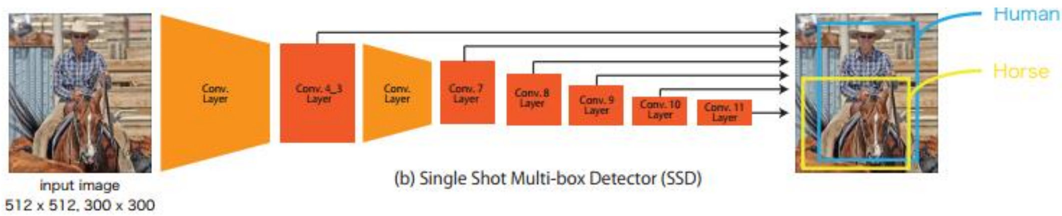


図 23　SSD の概念図 [3]

2.2.8　YOLOv2 を用いた手法

YOLOv2 はリアルタイム人物検出システムであり、R-CNN の 1000 倍、Fast R-CNN の 100 倍高速な物体検出手法である[27]。YOLOv2 による物体検出はグリッドベースで行われ、

画像を領域分割して、各領域におけるバウンディングボックスおよび確率を予測する。YOLOv2 では画像を入力すると、物体カテゴリと Bounding Box の位置とスコアが出力される。カテゴリ数の多さや検出速度は優れているものの、グリッド内で識別できるクラスが 1 つ且つ検出できる物体数が 2 つまで等といった欠点も見られる。そのため、1 つのグリッド内にオブジェクトが 3 つ以上映っている画像には不向きである[3]。図 24 に YOLOv2 の概念図を示す。

図 24　YOLOv2 の概念図[3]

YOLOv2 のプログラムは GitHub 上で公開されている[28]。実際に YOLOv2 を動かして物体検出を行った結果を図 25 に示す。また、参考として HOG 特徴量を用いて物体検出を行った結果を図 26 に示す。

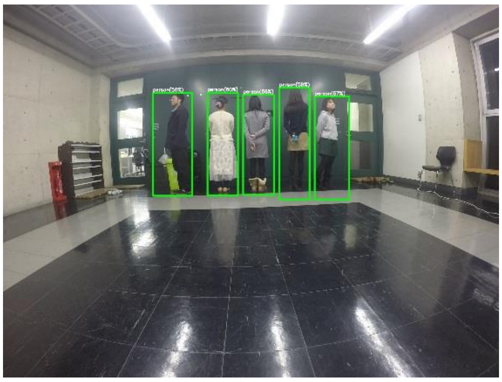


図 25　YOLOv2 を用いた場合の
物体検出結果

図 26（参考）HOG 特徴量を用いた場合の
物体検出結果

第3章　　一般的な画像鮮明化手法

第 2 章に述べたように、Deep Learning を用いた物体検出手法が提案されて以降、画像に写る物体の検出精度は飛躍的に向上している。しかし夜間や悪天候時に撮影された不鮮明画像の場合には、物体検出処理を行う前に一度画像の鮮明化を行う必要がある。

本章では、不鮮明画像の中でも特に HAZE 発生時に撮影された画像および夜間に撮影された画像を扱う。HAZE とは気象用語であり、主に霧や靄を指す用語である。

以下に、不鮮明画像の一般的な鮮明化手法の一例として、線形濃度変換を用いた手法、非線形濃度変換（ガンマ補正）を用いた手法、ヒストグラム平均化を用いた手法、鮮明化機器を用いた手法、Local Contrast 強調を用いた手法について述べる。

3.1　線形濃度変換を用いた鮮明化手法

この手法は、線形濃度変換を利用して不鮮明画像の鮮明化する手法である。線形濃度変換では、以下の式(16)に従って画像を変換する[29]。

$$z' = az + b \qquad 式(16)$$

z:入力画像の輝度値

z': 出力画像の輝度値

a: $gain$

b:　$bias$

式(16)における a, b 値を指定することにより、元の画像の輝度値が変化するように画像を変換することができる。出力画像は a=1 b=0 では変化せず、b の値により輝度値が底上げされる。図 27 に a を 1 から 3、b を 0 から 100 まで変化させた場合のトーンカーブを示す。また、線形濃度変換の変換画像例およびそのヒストグラムを以下の図 28 に示す。

本章は文献文献[高橋沙季 et.: ”CNN を用いた画風変換による物体検出精度の改善"]に基づいている

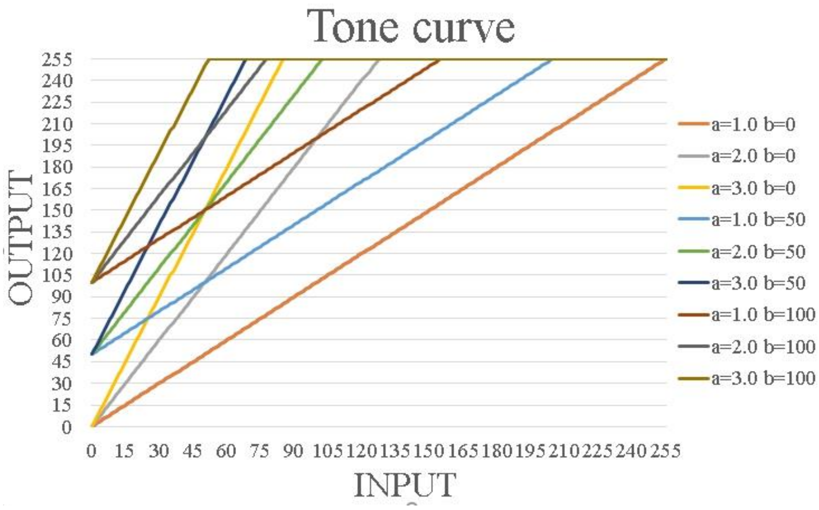
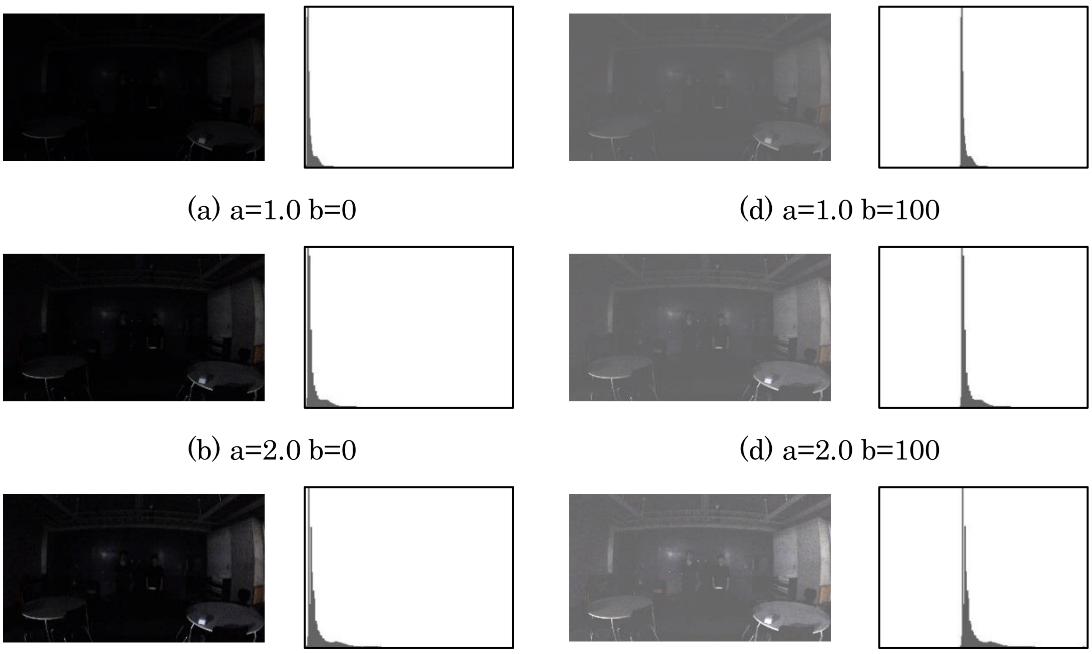


図 27　線形濃度変換のトーンカーブ例



(a) a=1.0 b=0　(d) a=1.0 b=100

(b) a=2.0 b=0　(d) a=2.0 b=100

(c) a=3.0 b=0　(e) a=3.0 b=100

図 28　線形濃度変換の変換結果画像およびヒストグラム

3.2　非線形濃度変換（ガンマ補正）を用いた鮮明化手法

この手法は、非線形濃度変換（γ補正）を利用して不鮮明画像の鮮明化を行う[30]。非線形濃度変換では、以下の式(17)に従って画像を変換する。

$$z' = \left(\frac{z}{255}\right)^{\frac{1}{\gamma}} \times 255 \qquad \text{式(17)}$$

z:入力画像の輝度値

z': 出力画像の輝度値

γ: ガンマ値

式(17)におけるγ値を指定することにより、元の画像の輝度値が変化するように画像を変換することができる。出力画像はγ=1では変化せず、γ<1では暗く、γ>1では明るくなる。図29にγを変化させた場合のトーンカーブを示す。また、非線形濃度変換の変換画像例およびそのヒストグラムを以下の図30に示す。

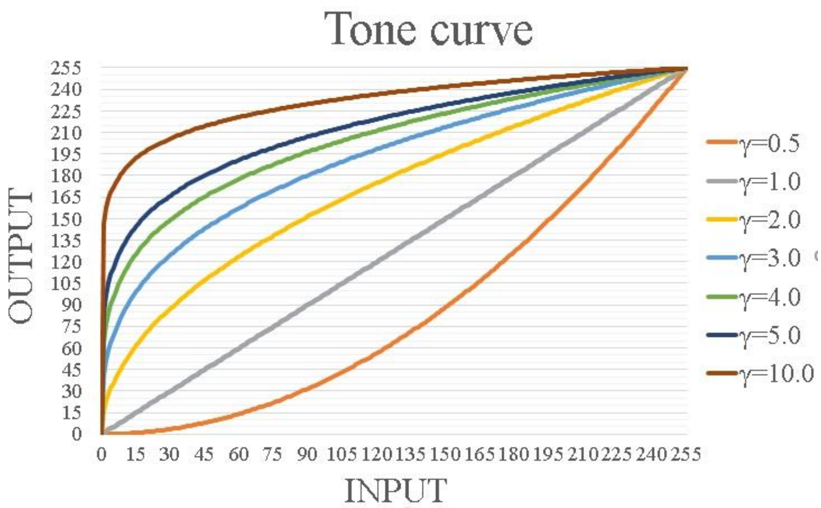
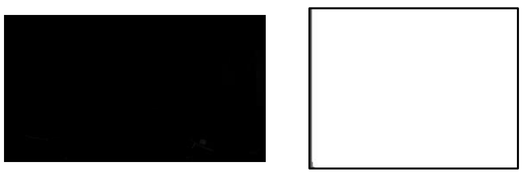


図29　非線形濃度変換（γ補正）のトーンカーブ例

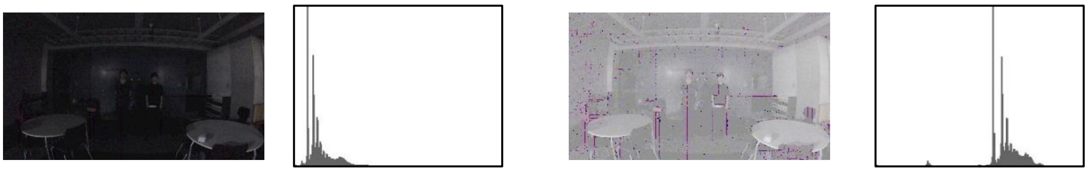


(a)　γ=0.5

(d)　γ=3.0

(b)　γ=1.0

(d)　γ=5.0



(c) γ =2.0　　　　　　　　(e) γ =10.0

図 30　非線形濃度変換の変換結果画像およびヒストグラム

3.3　ヒストグラム平均化を用いた鮮明化手法

この手法は、ヒストグラム均一化を利用して不鮮明画像の鮮明化を行う手法である。ヒストグラム均一化では、以下の式(18)に従って画像を変換する[31]。

式(18)

$$g(a,b) = INT\left(\frac{\frac{S(f(a,b))}{A\times B} - min\left(\frac{S(f(a,b))}{A\times B}\right)}{1 - min\left(\frac{S(f(a,b))}{A\times B}\right)}(C-1)\right)$$

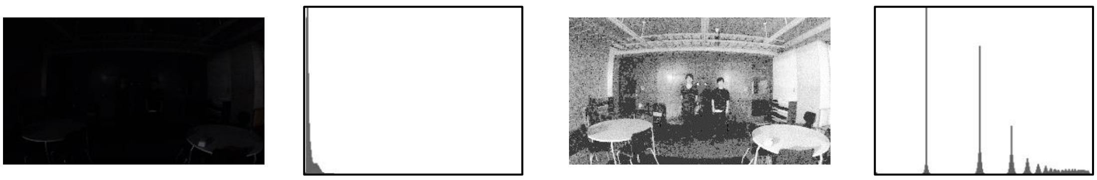
$f(a,b), g(a,b)$：処理前後の輝度値

$S(f(a,b))$：$f(a,b)$までの累積度数

$A\times B$：総画素数

C：輝度値の階調数(256)

ヒストグラム均一化は、画像のヒストグラムの累積度数のグラフを取った時に、傾きが一定になるように画像を変換する処理である[31]。この処理は輝度値が偏っている画像に特に有効であるとされている。以下の図 31 に、ヒストグラム均一化の変換画像例およびそのヒストグラムを示す。



(a) 原画像　　　　　　　　(d) ヒストグラム均一化後

図 31　ヒストグラム均一化の変換結果画像およびヒストグラム

3.4 ダークチャネルを用いた鮮明化手法

この手法は、CVPR2009 にて He らによって発表された論文 “Single Image Haze Removal Using Dark Channel Prior”[7]で提案された手法である。画像から HAZE を除去する手法の中で最も代表的な手法の一つである。

本手法では HAZE 画像モデルを利用して、HAZE 除去を行う。HAZE 画像モデルにより、HAZE 画像は以下の式(19)で表される。

$$I(x) = J(x)t(x) + A(1 - t(x)) \qquad \text{式(19)}$$

x: 画素位置

$I(x)$: HAZE画像

$J(x)$:鮮明画像（HAZE 除去画像）

$t(x)$:透過マップ

A:環境光

この HAZE 画像モデルを利用して HAZE 画像から鮮明画像を得るためには、式(19)で鮮明画像$J(x)$を明らかにする必要がある。しかし、式(19)では未知数が多数存在するため、式を解くために仮定や前提条件を設ける必要がある。そこで、He らはダークチャネル処理を利用して HAZE 画像モデルから鮮明画像を得る手法を提案している。

He らの手法では、屋外の HAZE 無し画像の統計に基づいてダークチャネル処理を利用し、HAZE 画像の空領域から環境光を推定する[7]。次に、HAZE 状況下で光がどの程度散乱なくカメラに到達するのかを示す透過マップを推定し、最後に求めた環境光および透過マップの値を利用して式(19)より鮮明画像を復元する。

以下の 3.4.1～3.4.3 に、HAZE 画像モデル、ダークチャネル処理、本手法の流れについて詳細に述べる。

3.4.1 HAZE 画像モデル

HAZE 画像モデルは、以下の式(20)で表される。$t(x)$は透過マップであり、光が霧や靄の粒子に散乱されずにカメラまで到達する度合いを示す。また、Aは環境光である。式(20)の$J(x)t(x)$は直接減衰と呼ばれ、情景情報が散乱なくカメラに到達する度合いを意味する。また$A(1 - t(x))$は環境光が HAZE 画像に及ぼす影響の度合いを意味する。

$$I(x) = J(x)t(x) + A(1 - t(x)) \qquad \text{式(20)}$$

x: 画素位置

$I(x)$: HAZE画像

$J(x)$:鮮明画像（HAZE 除去画像）

$t(x)$:透過マップ

A:環境光

透過マップは上述の通り、光が霧や靄の粒子に散乱されずにカメラまで到達する度合いを意味しており、次式(21)で表される。

$$t(x) = e^{-\beta \mathrm{d(x)}} \qquad \text{式(21)}$$

β: 環境光の散乱係数

$d(x)$:物体からカメラまでの距離

光が通過する距離が長いと環境光による減衰が大きくなるため、透過マップは物体が近くに写っている場合には値が大きく、物体が遠くに写っている場合には値が小さくなる。HAZE 画像において透過マップは HAZE 濃度を表し、透過マップが小さい場合は HAZE 濃度が低いことから近景、透過マップが大きい場合には HAZE 濃度が高いことから遠景であると推測することができる[32]。

3.4.2　ダークチャネル処理

ダークチャネルとは、画素位置xを中心とする局所領域におけるIの色チャネルの最小値のことである。He らはダークチャネルを次式(22)のように定義している[32]。

$$I^{dark}(x) = min_{\mathrm{c}\in\{\mathrm{r,g,b}\}}(min_{y\in\Omega(x)} I^{c}(y)) \qquad \text{式(22)}$$

I^{c}: Iの色チャネル

$\Omega(x)$: xを中心とする局所領域

ダークチャネルは「屋外の HAZE 無し画像において、画像の空領域以外の領域のほとんどでは、少なくとも一つの色チャネルの強度が非常に低い」という仮定に基づいている。He らは、5000 枚の日中の HAZE 無し画像を用いた実験を行った際に、全画素の約 75%はダークチャネル強度が 0、90%はダークチャネル強度が 25 未満となった結果を踏まえてこの仮定を立てている。一方 HAZE 画像の場合は環境光が加わるため、鮮明画像よりも明るくなり、ダークチャネル強度は高くなる[32]。ゆえにダークチャネル強度は HAZE 濃度を意

味し、ダークチャネル強度が強い場合には HAZE 濃度が高く、ダークチャネル強度が低い場合には HAZE 濃度が低いと推測することができる。

3.4.3　本手法の流れ

本手法の流れを、①環境光の推定、②透過マップの推定、③鮮明画像の復元の三ステップに分けて述べる。

＜＜ステップ１：環境光の推定＞＞

He らの手法では、環境光を推定するために空領域の色情報を利用している。このように、環境光推定に空領域の情報が必要になる理由を以下に述べる。
3.4.1 では HAZE 画像モデルについて述べた。この HAZE 画像モデルにダークチャネルを適用した場合、次式(23)で表現される。

$$I^{dark}(x) = J^{dark}(x)t(x) + A^c(1 - t(x)) \qquad \text{式(23)}$$

x:画素位置
$I^{dark}(x)$: HAZE画像のダークチャネル
$J^{dark}(x)$:鮮明画像のダークチャネル
A^c:環境光の色チャネル

鮮明画像のダークチャネルはほとんどが輝度値 0 であることから、式(23)は次式(24)のように書き換えることができる。

$$I^{dark}(x) \approx A^c(1 - t(x)) \qquad \text{式(24)}$$

これを基に、次式(25)の透過マップを推定する式が得られる。

$$t(x) \approx 1 - I^{dark}(x)/A^c \qquad \text{式(25)}$$

また HAZE 画像の空領域は距離が無限遠であるとすると、空領域の透過マップ$t(x)$はほぼ 0 になり、次式(26)のように環境光Aの値は空領域の色と等しいということができる。これが環境光を空領域の情報から推定することができる理由である。

なお空領域の特定にあたっては、I^{dark}の上位 0.1%の高強度画素を抽出し、HAZE 画像の輝度値が最も高い画素を選出する必要がある[32]。

$$A^c = I^{dark}(x) \qquad \text{式(26)}$$

＜＜ステップ２：透過マップの推定＞＞

透過マップを推定するためには、まず入力 HAZE 画像から、局所領域の HAZE 濃度を示す粗マップを得る。次に、詳細なエッジ情報を示す詳細マップを得て、得られた二つのマップから各画素の HAZE 濃度を表す透過マップを合成する。

以下にその詳細を述べる。粗マップとは局所領域に基づくダークチャネルのことであり、式(27)で表される。一方 HAZE 画像のエッジ詳細情報を表す詳細マップは式(28)で表される。

$$M^{coarse}(x) = min_{c\in\{r,g,b\}}(min_{y\in\Omega(x)} I^c(y)) \qquad \text{式(27)}$$

$$M^{fine}(x) = min_{c\in\{r,g,b\}}\, I^c(x) \qquad \text{式(28)}$$

透過マップを得るためには、これらの粗マップと詳細マップを合成する必要がある。合成マップは式(29)に示すとおり、詳細マップのブロックの最大値と、詳細マップの位置 x にある画素値とを比べて、小さいほうを選び出すことによって算出される[32]。

また、合成マップから透過マップを導き出すことができ、式(30)で表される。なお、ω は画像の HAZE 除去の度合いを表すパラメータであり、0 より大きく 1 未満である。一般的にこの値は 0.9 が適切とされている。

$$M^t(x) = \min(\max_{y\in\Omega(x)} M^{coarse}(y), M^{fine}(x)) \qquad \text{式(29)}$$

$$t(x) = 1 - \omega M^t(x)/A \qquad \text{式(30)}$$

＜＜ステップ３：鮮明画像の復元＞＞

環境光と透過マップを推定し終えたら、最後に次式(31)により鮮明画像を復元する。

$$J(x) = \frac{I(x) - A}{\max(t(x), t_0)} + A \qquad \text{式(31)}$$

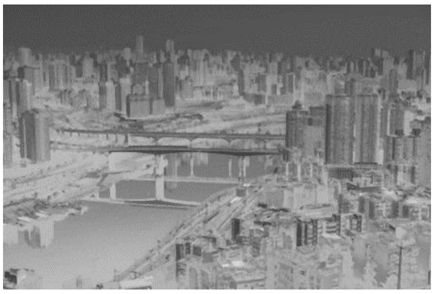
t_0:0 による除算を避けるための定数値

このように、空領域の色情報から環境光を推定し透過マップを推定した上で、求めた値を HAZE 画像モデルに代入することにより、HAZE 画像から鮮明画像を復元できる。

以下の図 32 に、HAZE 画像、HAZE 画像のダークチャネル画像、HAZE 画像の透過マップ画像、鮮明画像の例を示す。

HAZE 画像

HAZE 画像のダークチャネル画像



HAZE 画像の透過マップ画像

鮮明画像

図 32　ダークチャネルを用いた HAZE 除去に関する画像例

3.5 鮮明化機器を用いた鮮明化手法

この手法は、市販の鮮明化機器を用いて不鮮明画像を鮮明化する手法である。代表的な鮮明化機器の例として、INFOTECH 社の Red Super Eye G2 [33][34]を挙げる。図 33 の Red Super Eye G2 は、濃霧や雨などの悪天候映像、夜間などの光量不足映像、西日等の逆光映像など、悪条件下で撮影された映像の鮮明化が可能である。映像の状態を自動で把握し濃度ヒストグラムの平均化処理や色情報処理を行うため、複雑なパラメータ設定が不要な点が大きな特徴である[35]。鮮明化機器を用いて鮮明化を行った例を図 34 に示す。

図 33　鮮明化機器 Red Super Eye G2 [34]

原画像　　　　鮮明化後画像

図 34　鮮明化機器を用いた鮮明化例[10]

3.6 Local Contrast 強調処理を用いた鮮明化手法

この手法は、Local Contrast 強調処理を用いて画像を鮮明化する手法である。Local Contrast 強調処理は、強いエッジを保持しながら細かな部分を増加させて局所的なコントラストを強調する処理である[36]。

パラメータ値は、そのまま保持する強いエッジの最小強度の振れ幅を示す Edge threshold、必要な強調量を示す Amount の 2 種類である。以下の図 35 に Edge threshold および Amount を変化させて Local Contrast 強調を行った結果を示す。




Edge threshold=0 Amount=0	Edge threshold=0.33 Amount=0	Edge threshold=0.66 Amount=0	Edge threshold=1 Amount=0




Edge threshold=0 Amount=1	Edge threshold=0.33 Amount=1	Edge threshold=0.66 Amount=1	Edge threshold=1 Amount=1

図 35　Local Contrast 強調の実行結果例

図 35 上段の画像を見ると、Amount の値が 0 であるため Edge threshold の値を変化さ

せても画像に変化が見られない。一方、下段の画像では、Amount が 0 以外の値を取っているため Edge threshold の値を大きくするにつれて画像の強調量も増えている。また、各列を見てみると、上段と下段では Edge threshold の値に変化は無いが、下段のほうが Amount の値が大きいため画像の強調量が増えているのが分かる。

第4章　　画風変換を用いた画像鮮明化の提案

4.1 概要

本研究では、画風変換を用いた不鮮明画像の画像鮮明化を提案する。本手法では、画風変換の特性を活かし、入力コンテンツ画像を不鮮明画像、入力スタイル画像を晴れの日の昼間に撮影した画像として画風変換を行うことにより、不鮮明画像を晴れの日の昼間に撮影したような画像に変換して、鮮明化を図る。また、鮮明化だけでなく、不鮮明画像に写っている物体の検出精度向上も試みる。

4.2 画風変換を用いた画像鮮明化の提案

本研究では、画風変換による不鮮明画像の鮮明化および物体検出精度向上を図る。本来、画風変換は変換された画像の美しさを楽しむためや、どのような画像が生成されるのかを楽しむために使用されており、画像の鮮明化に使用されることはない。しかし、本研究では図 36 のように画風変換を利用して、入力コンテンツ画像を不鮮明画像、入力スタイル画像を晴れの日の昼間に撮影した画像とすることにより、任意の不鮮明画像を晴れの日の昼間に撮影した画像風に変換する。さらには、不鮮明画像に写っている物体の検出精度向上も試みる。画風変換を行うにあたっては図 37 に示す Johnson 氏らのモデル[9]を用いる。

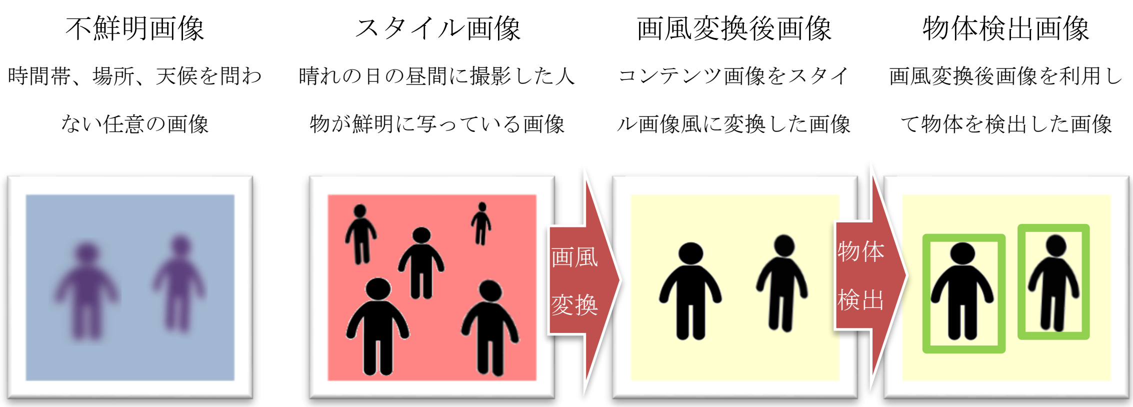


図 36　画風変換を用いた不鮮明画像鮮明化および物体検出のイメージ

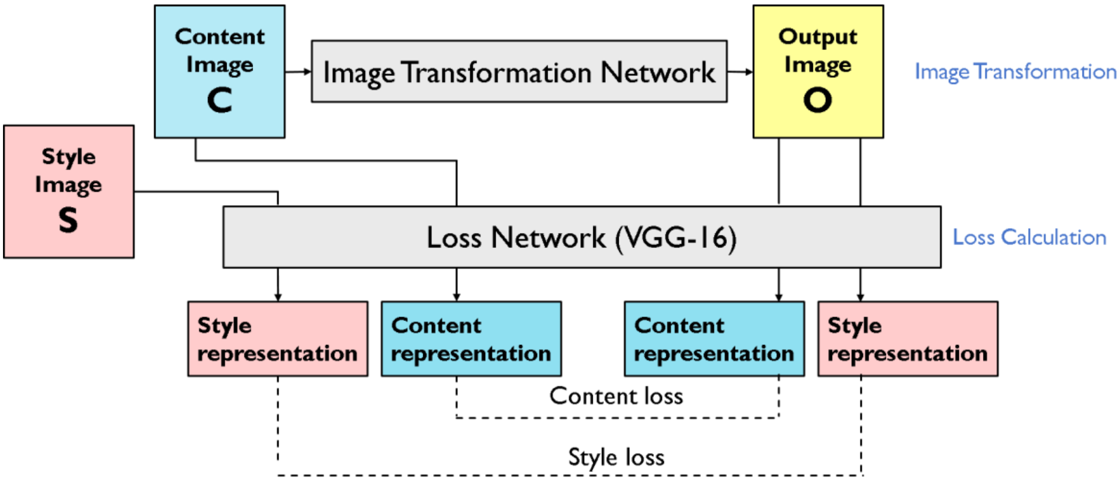


図 37　本研究で用いる画風変換モデル（Johnson 氏らのモデル）[9]

5 章の 5.3〜5.5 における実験では、以下の図 38 の画像をスタイル画像とする。図 38 のスタイル画像は、晴れの日の昼間に大学のキャンパス内で撮影した画像である。

不鮮明画像の鮮明化を画風変換を用いて行う場合、晴れの日の昼間風に変換するわけではなく、あくまでも図 38 の画像風に変換する点に注意が必要である。画風変換はスタイル画像とコンテンツ画像を損失計算ネットワークに入力し、その画素値から得られる情報を基に計算を行うため、入力するスタイル画像が異なれば画風変換結果にも大きな違いが生じる。したがって、たとえスタイル画像に晴れの日の昼間に撮影した画像を用いたとしても図 38 と別画像であれば同一の結果は得られず、その時に用いたスタイル画像風に変換されることとなる。

図 38　5.3〜5.5 節の実験で用いるスタイル画像

第5章　実験

5.1 概要

本提案手法の有効性を確認するため、画風変換を用いた不鮮明画像鮮明化に関する実験を行った。

5.2 には実験環境および実験使用画像について述べる。5.3 では本提案手法を用いて不鮮明画像に対し画風変換処理を用いて鮮明化を行った結果を示す。ここで、不鮮明画像は HAZE 画像と夜間画像の 2 パターンを用意する。5.4 では、5.3 の結果画像に対して物体検出処理を行った結果を示す。さらに、5.5 では鮮明化後画像の画質評価結果を示し、最後に 5.6 でスタイル画像の違いが及ぼす実験結果への影響について比較結果を示す。

5.2 実験環境と実験使用画像

5.2.1　実験環境

本実験は、OS が Ubuntu 16.04LTS、CPU が Intel(R) Xeon(R)E5-2630 @2.30GHz、GPU が NVIDIA Corporation Device 1c82(rev al)のマシンを用いて行った。また、ソフトウェアは Chainer1.24.0 および Python3.5.2 を使用している。

5.2.2　実験使用画像

本実験に使用する不鮮明画像は、HAZE 画像および夜間画像である。HAZE 画像は、インターネット上で公開されている Haze Level Estimation Dataset[37]、FRIDA[38]、FRIDA2[38]からダウンロードした画像を使用する。一方、夜間画像には、自分で撮影した画像の他、Microsoft COCO dataset[10]に含まれる夜間画像を使用する。

5.3 実験 1：画風変換による不鮮明画像の鮮明化

4.2 節で示した、画風変換を用いた不鮮明画像鮮明化手法に対する有効性を確認するため、実験を行った。以下に、3 章で述べた他の鮮明化手法と比較する形で、実験結果をまとめる。また、5.3.1 には HAZE 画像の鮮明化実験結果を、5.3.2 には夜間画像の鮮明化実験結果を

述べる。

5.3.1　HAZE 画像の鮮明化実験結果

HAZE 発生時に撮影された画像に対して画風変換を行った際の実験結果を以下の図 39 に示す。また、一般的な鮮明化手法の例である線形濃度変換手法、非線形濃度変換手法（γ補正）、ヒストグラム平均化手法、ダークチャネルを用いた手法、鮮明化機器を用いた手法、Local Contrast 強調を用いた手法による各実験結果も併せて示す。

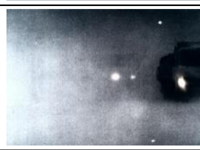
	鮮明化結果例 1	鮮明化結果例 2	鮮明化結果例 3
原画像			
(1) 画風変換			
(2) 線形濃度変換			
(3) 非線形濃度変換（γ変換）			
(4) ヒストグラム平均化			
(5) ダークチャネル手法			
(6) 鮮明化機器			
(7) Local Contrast 強調			

図 39　各鮮明化手法による HAZE 画像の鮮明化結果[39][40]

5.3.2　　夜間画像の鮮明化実験結果

夜間時に撮影された画像に対して画風変換を行った際の実験結果を以下の図 40 に示す。また、一般的な鮮明化手法の例である線形濃度変換手法、非線形濃度変換手法（γ 補正）、ヒストグラム平均化手法、ダークチャネルを用いた手法、鮮明化機器を用いた手法、Local Contrast 強調を用いた手法による各実験結果も併せて載せている。

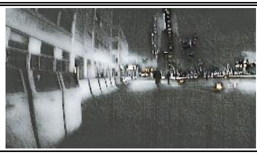
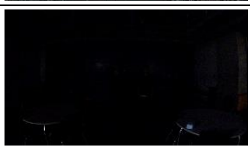
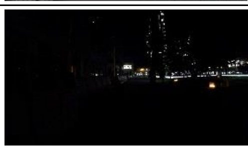
	鮮明化結果例 1	鮮明化結果例 2	鮮明化結果例 3
原画像			
(1) 画風変換			
(2) 線形濃度変換			
(3) 非線形濃度変換（γ変換）			
(4) ヒストグラム平均化			
(5) ダークチャネル手法			
(6) 鮮明化機器			
(7) Local Contrast 強調			

図 40　各鮮明化手法による夜間画像の鮮明化結果

5.3.3　実験結果のまとめ

5.3.1 および 5.3.2 の結果より推測できることを表 2 に示す。

表 2　5.3.1 及び 5.3.2 の鮮明化結果のまとめ

	HAZE 発生時	夜間時
画風変換	○事前パラメータ値設定が不要 ○物体がくっきりと見える ×太陽光が差し込む方向がぼやける	○事前パラメータ値設定が不要 ○コントラストの強い視認性の良い画像が得られる ×変換後は原画像の色味が失われる
線形濃度変換	×原画像とほぼ変化無し ×画像ごとに最適なパラメータ値を設定するのが困難	○原画像の色味が失われない ×画像ごとに最適なパラメータ値を設定するのが困難
非線形濃度変換（γ補正）	×原画像とほぼ変化無し ×画像ごとに最適なγ値を設定するのが困難	○原画像の色味が失われない ○照明の周りの白飛びが少ない ×画像ごとに最適なγ値を設定するのが困難
ヒストグラム平均化	○事前パラメータ値設定が不要 ×コントラストが強く、視認性が悪い ×太陽光が差し込む方向がぼやける	○事前パラメータ値設定が不要 ○コントラストの強い視認性の良い画像が得られる ×照明の周りが白飛びする
ダークチャネルを用いた手法	○事前のパラメータ設定が不要 ×空領域の色から環境光を推定するため、空領域が見えていない画像では鮮明化効果が低い	○事前のパラメータ設定が不要 ×夜間画像は鮮明化できない（もともと HAZE 発生時等の白っぽい不鮮明画像を想定しているため）
鮮明化機器	○事前のパラメータ設定が不要 ○検出したい物体が遠くに写っている場合に特に有効 ×コントラスト強度があまり強くない	○事前のパラメータ設定が不要 ×コントラスト強度が画風変換・ヒストグラム平均化と比べて弱い
Local Contrast 強調	○物体がくっきりと見える ×太陽光が差し込む方向がぼやける ×画像ごとに最適なパラメータ値を設定する必要がある	×夜間画像は鮮明化できない ×画像ごとに最適なパラメータ値を設定する必要がある

表 2 より、画風変換による画像鮮明化の優位性は画像ごとの事前パラメータ値設定が不要である点、不鮮明画像の撮影場所や撮影時間帯に捉われる必要がない点、出力画像のコントラストが高く視認性が良い点であると考えられる。

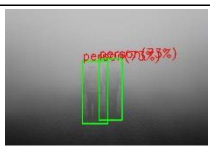
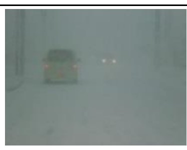
一方で、画風変換による画像鮮明化の問題点として、出力画像では原画像の色味が失われる点や、光が差し込む方向がぼやける点が挙げられる。原画像の色味が失われる理由は、画風変換では出力画像が原画像だけでなくスタイル画像の色味にも影響されるためであると考える。

5.4 実験 2：鮮明化後画像を用いた物体検出

5.3 では、画風変換による不鮮明画像の鮮明化実験について述べた。本節では、鮮明化後画像を用いた物体検出について、3 章で述べた他の鮮明化手法と比較する形で、実験結果をまとめる。5.4.1 では HAZE 画像の鮮明化画像の物体検出結果について述べ、5.4.2 では夜間画像の鮮明化画像の物体検出結果について述べる。

5.4.1 HAZE 画像の鮮明化画像の物体検出結果

HAZE 発生時に撮影された画像に対して画風変換を行ったのちに物体検出処理を行った結果を以下の図 41 に示す。また、一般的な鮮明化手法の例である線形濃度変換手法、非線形濃度変換手法（γ 補正）、ヒストグラム平均化手法、ダークチャネルを用いた手法、鮮明化機器を用いた手法、Local Contrast 強調を用いた手法による各実験結果も示す。

	鮮明化画像の 物体検出結果例 1	鮮明化画像の 物体検出結果例 2	鮮明化画像の 物体検出結果例 3
原画像			
(1) 画風変換	 77%,82%	 0%	 50%
(2) 線形濃度変換	 73%,73%	 0%	 0%

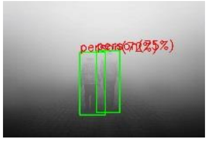
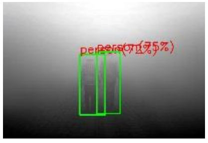
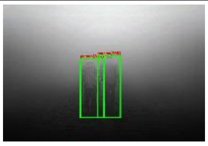
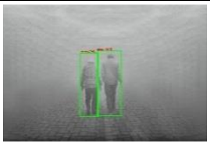
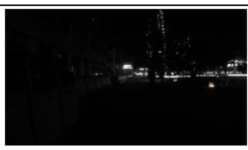
(3) 非線形濃度変換（γ変換）	 72%,75%	 0%	 0%
(4) ヒストグラム平均化	 72%,75%	 0%	 0%
(5) ダークチャネル手法	 64% 76%	 0%	 0%
(6) 鮮明化機器	 75%,81%	 0%	 0%
(7) Local Contrast 強調	 77%,81%	 68%	 58%

図 41　各手法による HAZE 鮮明化画像の物体検出結果

5.4.2　　夜間画像の鮮明化画像の物体検出結果

夜間時に撮影された画像に対して画風変換を行ったのちに物体検出処理を行った結果を以下の図 42 に示す。また、一般的な鮮明化手法の例である線形濃度変換手法、非線形濃度変換手法（γ補正）、ヒストグラム平均化手法、ダークチャネルを用いた手法、鮮明化機器による手法、Local Contrast 強調を用いた手法による各実験結果も併せて載せている。

	鮮明化画像の 物体検出結果例 1	鮮明化画像の 物体検出結果例 2	鮮明化画像の 物体検出結果例 3
原画像			

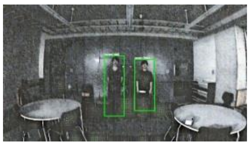
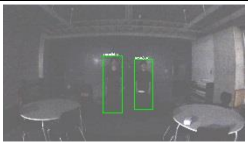
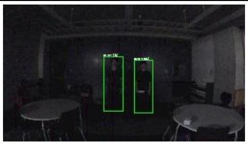
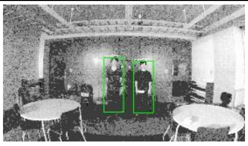
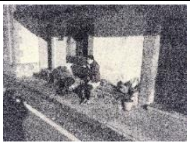
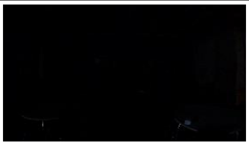
(1) 画風変換	 59%,66%	 0%	 0%
(2) 線形濃度変換	 58%,62%	 0%	 0%
(3) 非線形濃度変換（γ変換）	 54%,59%	 0%	 0%
(4) ヒストグラム平均化	 51%,53%	 0%	 0%
(5) ダークチャネル手法	 0%	 0%	 0%
(6) 鮮明化機器	 68%,59%	 0%	 0%
(7) Local Contrast 強調	 0%	 0%	 0%

図 42　各鮮明化手法による夜間鮮明化画像の物体検出結果

5.4.3　　実験結果のまとめ

5.4.1 および 5.4.2 の結果を見ると、HAZE 画像に関しては画風変換手法および Local Contrast 強調処理による鮮明化後画像の物体検出精度が、他手法と比べて高い結果となった

。画風変換およびLocal Contrast強調処理による鮮明化では共に強調効果が見られるため、物体が背景と分離され物体検出精度が高くなったと考えられる。また、画風変換手法は画像ごとに最適なパラメータ値を設定する必要がない。この点が、Local Contrast強調処理と比べた時のアドバンテージの一つである。

一方、夜間画像に関しては画風変換手法および鮮明化機器による鮮明化後画像の物体検出精度が、他手法と比べて高い結果となった。HAZE画像ではLocal Contrast強調処理が優れていたが夜間画像に対しては同様の結果とならなかった。

5.5 実験3：鮮明化後画像の画質評価結果

本節では、鮮明化後画像の画質評価実験について述べる。5.5.1では、不鮮明画像の画質を基に画質評価を行う。5.5.2では鮮明画像（HAZE無し画像）を基に画質評価を行う。

5.5.1　不鮮明画像を基にした鮮明化後画像の画質評価

①画風変換手法②Heらが提案したダークチャネルを用いた手法③鮮明化機器を用いた手法④Local Contrast強調処理を用いた手法により鮮明化した画像を、不鮮明画像（HAZE画像）と比較する形で画質評価を行った。画質評価にはPSNRを用いている。表3に画質評価結果および物体検出結果を示す。

表3　各手法による鮮明化後画像の画質評価結果及び物体検出結果の比較
（PSNRの比較対象：不鮮明画像）

不鮮明画像	画風変換手法	ダークチャネルを用いた手法	鮮明化機器	Local Contrast強調
	 PSNR=17.3927 Car: 79%	 13.4090 Car: 62%	 18.7642 Car: 76%	 15.5681 Car: 78%
	 15.7413	 12.5078	 16.4618	 11.6497

	Car: 79%	Car: 76%	Car: 75%	Car: 78%
	 17.9031 Car: 81%	 13.9094 未検出	 15.5960 Car: 78%	 17.5691 Car: 81%
	 21.5935 Car: 74%	 14.5120 Car: 81% 56%	 14.8782 Car:83%58%69%	 19.1082 Car: 79% 51%

5.5.2　本来の鮮明画像（HAZE 無し画像）を基にした鮮明化後画像の画質評価

次に、①不鮮明画像（HAZE 画像）②画風変換手法③He らが提案したダークチャネルを用いた手法④鮮明化機器を用いた手法⑤Local Contrast 強調処理を用いた手法により鮮明化した画像を、本来の鮮明画像（HAZE 無し画像）と比較する形で画質評価を行った。画質評価には PSNR を用いている。表 4 に画質評価結果および物体検出結果を示す。

表 4　各手法による鮮明化後画像の画質評価結果及び物体検出結果の比較
（PSNR の比較対象：本来の鮮明画像）

本来の鮮明画像（HAZE 無し画像）	HAZE 画像	画風変換手法	ダークチャネルを用いた手法	鮮明化機器	Local Contrast 強調
	 PSNR= 11.8804	 10.6276 Car: 79%:	 12.9562 Car: 62%	 12.5770 Car: 76%	 10.3746 Car: 78%
					

	10.2356	10.2747 Car: 79%	13.4500 Car: 76%	12.4193 Car: 75%	13.1286 Car: 78%
	 9.4421	 11.1222 Car: 81%	 11.3842 未検出	 12.1796 Car: 78%	 9.7211 Car: 81%
	 13.4625	 14.1002 Car: 74%	 15.1079 Car:81%56%	 12.2042 Car:83%58% 69%	 12.3810 Car:79%51%

表 3 および表 4 の結果から、PSNR 値が高いほど物体検出精度が高いという結果ではないことが分かる。また、各表 4 段目の画像(車が遠くに 3 台写っている画像)では、1 台しか検出できていない画風変換手法よりも、3 台全て検出できている鮮明化機器のほうが PSNR は低い。この結果例からも、鮮明化後画像の物体検出精度と PSNR 値には相関が見られないことがわかる。

5.6 実験 4：スタイル画像の違いが及ぼす実験結果への影響

実験 1 から実験 3 では、画風変換する際のスタイル画像に一貫して同一の画像を用いてきた。本実験では、様々な画像を用いて画風変換を行い、スタイル画像の違いが及ぼす実験結果への影響について比較実験を行う。

実験にあたっては、画素位置の違いによる影響、カラーバリエーションの違いによる影響、ヒストグラム分布の違いによる影響の 3 つの観点から比較を行う。

5.6.1　画素位置の違いによる影響

本節では、ある画像を反転処理等して画素位置を変え、それぞれスタイル画像として画風変換した際に結果にどのような違いが生じるかを確認するための実験を行う。5.6.1.1 には画風変換結果の違いを、5.6.1.2 には物体検出結果の違いをまとめる。

5.6.1.1　画素位置の違いによる画風変換結果への影響

はじめに、あるスタイル画像を反転処理等して画素位置を変えた画像を用意し、それぞれをスタイル画像として画風変換した際に、画風変換結果にどのような影響を及ぼすのかを確認する。図 43 に、スタイル画像の画素位置の違いによる画風変換結果への影響を示す。

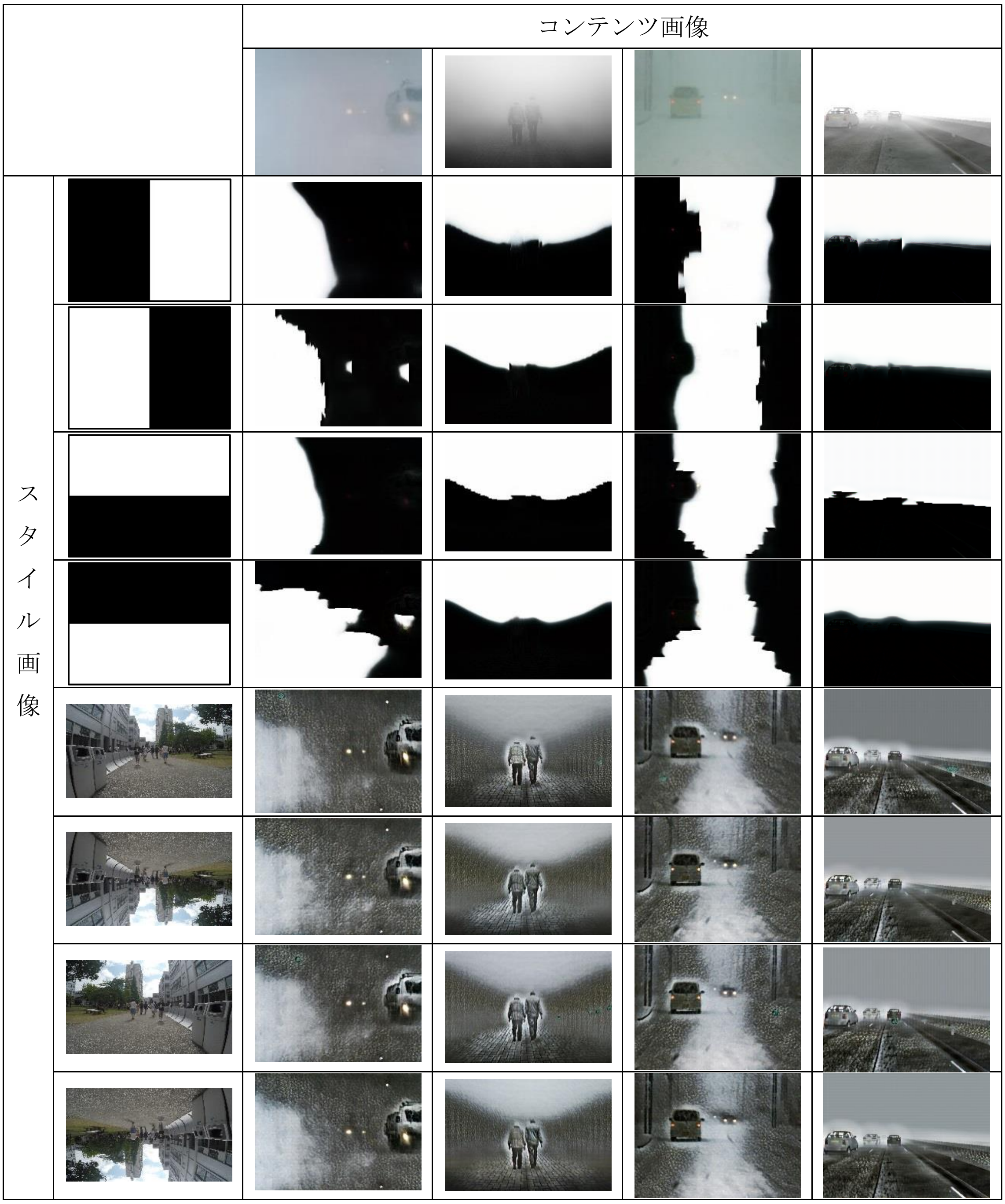
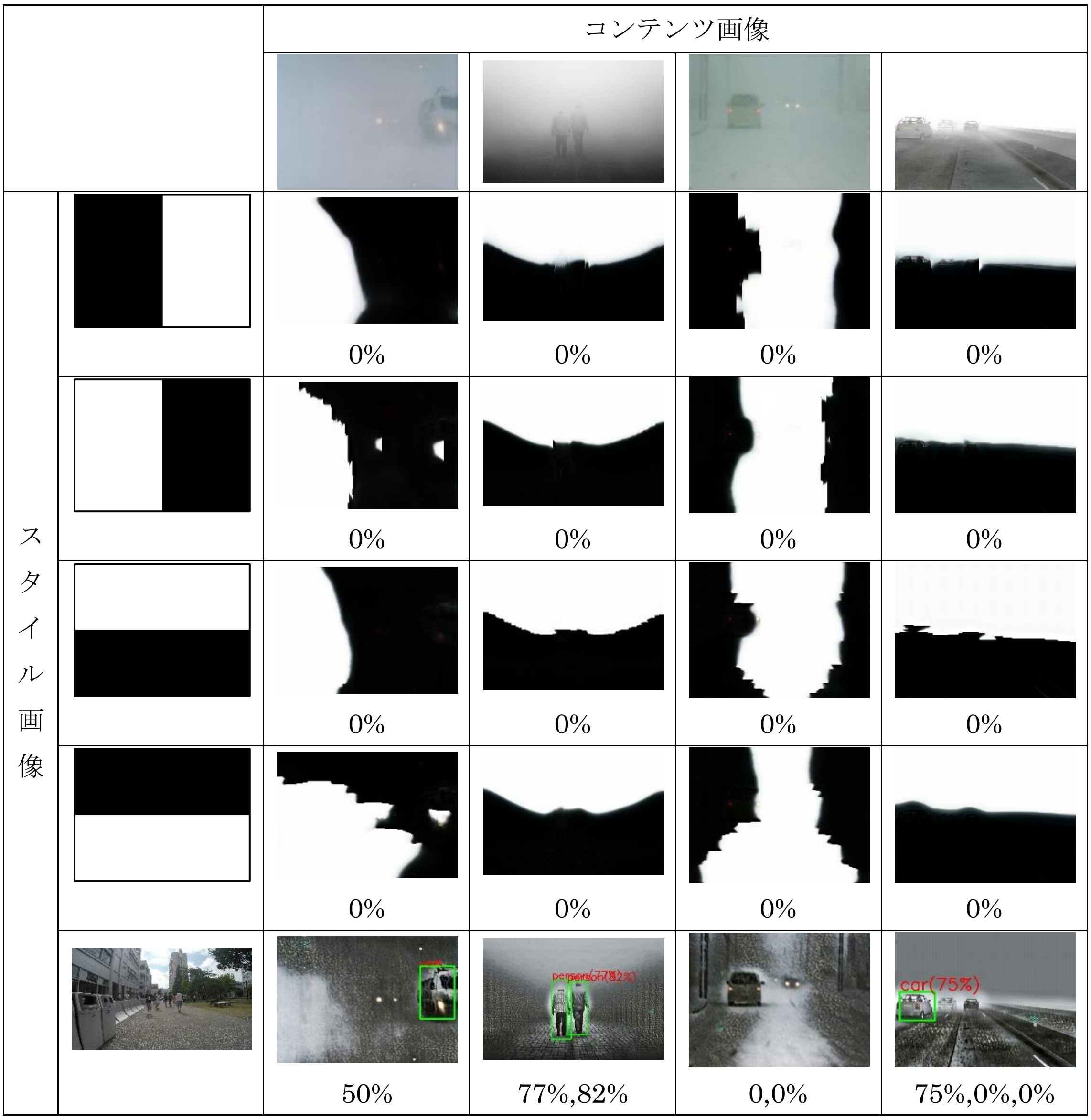


図 43　スタイル画像の画素位置の違いによる画風変換結果への影響

図 43 より、スタイル画像を反転等して画素位置を変えて画風変換を行ったところ、出力画像にも目視で若干の差が見られた。この結果から、スタイル画像の画素位置は出力画像に影響することを確認した。このような結果になった原因は、スタイル画像を損失計算ネットワークに入力する際に、画素位置順に値が入力されるためであると考える。

5.6.1.2　画素位置の違いによる物体検出精度への影響

次に、5.6.1.1 の画風変換後画像を用いて物体検出を行った結果を図 44 に示す。



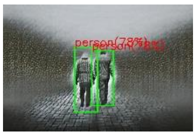
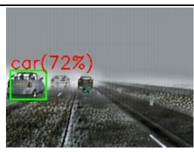
		 0%	 78%,78%	 0%,0%	 77%,0%,0%
		 66%	 73%,76%	 0%,0%	 72%,0%,0%
		 0%	 78%,80%	 0%,0%	 78%,0%,54%

図 44　スタイル画像の画素位置の違いによる物体検出結果への影響

図 44 より、スタイル画像を反転等して画素位置を変えて画風変換した画像に物体検出処理を行ったところ、物体検出精度に違いが生じることが分かった。この結果から、スタイル画像の画素位置は出力画像の物体検出精度に影響することを確認した。目視ではわずかな違いしか見られないが、スタイル画像は入力先の損失計算ネットワークだけでなく画風変換ネットワークにも影響を及ぼすため、ピクセルレベルでは全く別の画像を生成していることが推測できる。

5.6.2　　カラーバリエーションの違いによる影響

本節では、単色画像とグラデーション画像とデジタルカメラで撮影した画像を用意し、カラーバリエーションの違いによって実験結果にどのような違いが生じるかを確認するための実験を行う。5.6.2.1 には画風変換結果の違いを、5.6.2.2 には物体検出結果の違いをまとめる。

5.6.2.1　カラーバリエーションの違いによる画風変換結果への影響

単色画像とグラデーション画像とデジタルカメラで撮影した画像を用意し、それぞれをスタイル画像として画風変換した際に、画風変換結果にどのような影響を及ぼすかを確認する。図 45 にカラーバリエーションの画素位置の違いによる画風変換結果への影響を示す。

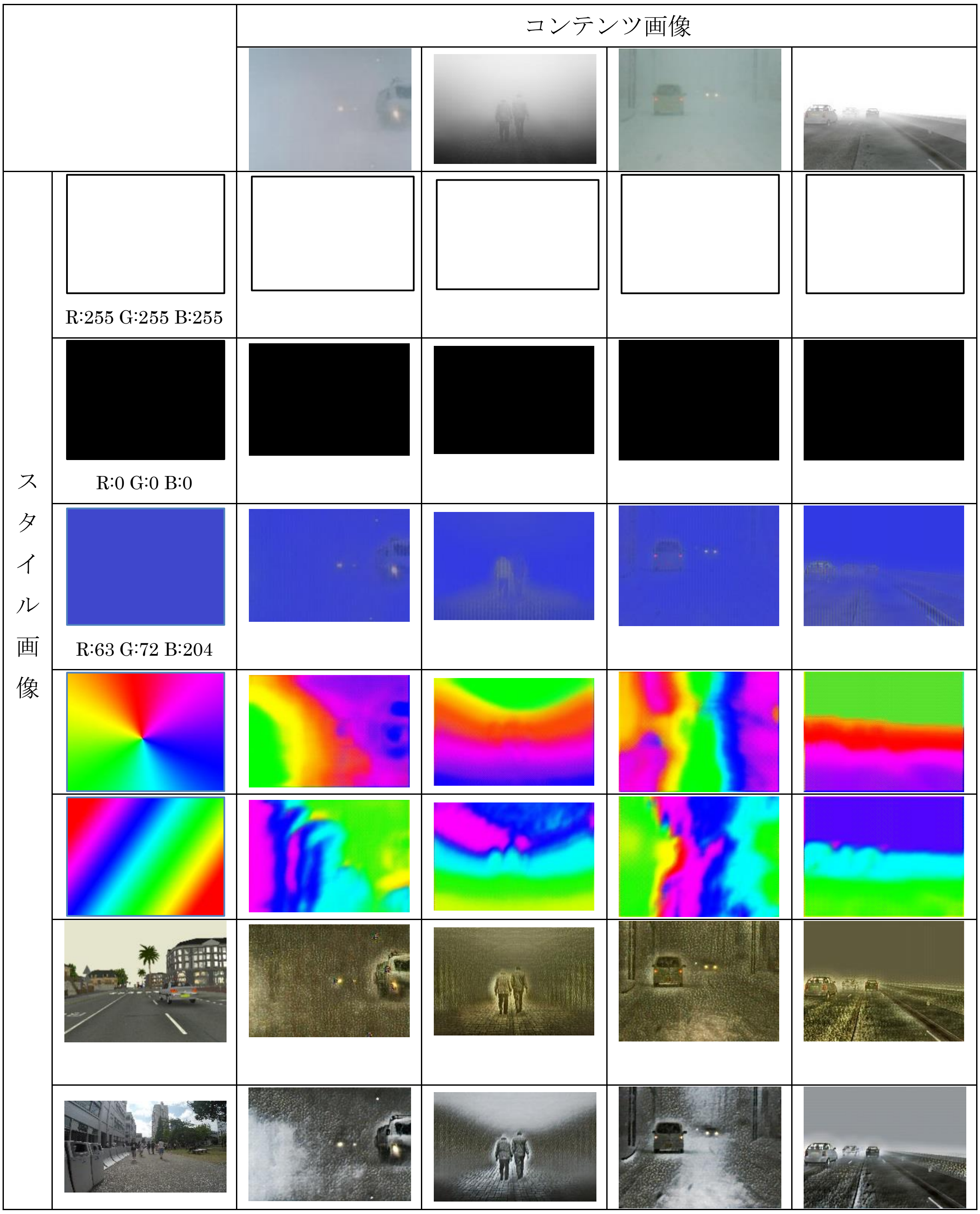
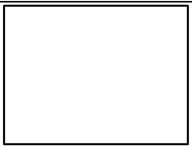
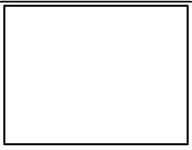
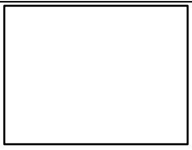
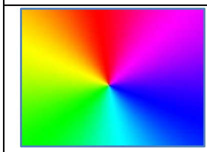
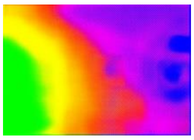
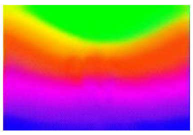
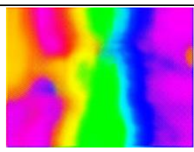
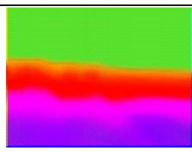
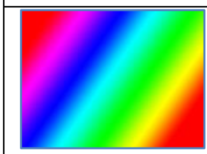
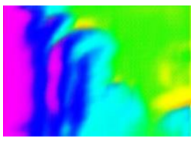
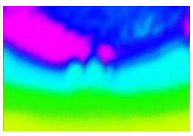
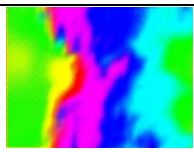
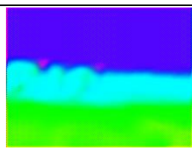
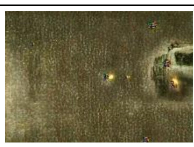


図 45　スタイル画像のカラーバリエーションの違いによる画風変換結果への影響

図 45 より、スタイル画像が単色画像の場合、RGB 値が全て同じ場合には画風変換効果が見られないが、RGB 値が異なる場合の単色画像であれば画風変換効果が得られることが分かった。また、単にスタイル画像の色を増やしたからといって出力画像の視認性が良くなるわけではないことを確認した。

5.6.2.2　カラーバリエーションの違いによる物体検出精度への影響

次に、5.6.2.1 の画風変換後画像を用いて物体検出を行った結果を図 46 に示す。

		コンテンツ画像			
					
スタイル画像		 0%	 0%	 0%	 0%
		 0%	 0%	 0%,0%	 0%,0%,0%
		 0%	 0%,0%	 0%,0%	 69%,0%,0%
		 0%	 0%,0%	 0%,0%	 0%
		 0%	 0%,0%	 0%,0%	 0%,0%,0%
		 0%	 73%,80%	 58%,0%	 77%,0%,53%

		 50%	 77,82%	 0%,0%	 74%,0%,0%

図 46　スタイル画像のカラーバリエーションの違いによる物体検出結果への影響

図 46 より、スタイル画像が単色画像であっても物体検出ができる例を確認した。これは、RGB 値が全て同じ単色画像ではなく、RGB 値が全て異なる単色画像をスタイル画像として用いた場合に見られた結果である。また、単にスタイル画像の色を増やすだけでは物体検出精度は向上しなかった。画風変換時の Style Loss の計算では中間層のチャネル間の相関を利用するため、チャネルの値にバラつきがあるほうがより効果的に画風変換ができるのではないかと推測する。

5.6.3　　ヒストグラム分布の違いによる影響

最後に、ある画像とその画像にヒストグラム平均化処理をした画像を用意し、それぞれスタイル画像として画風変換した際に結果にどのような違いが生じるかを確認するための実験を行う。5.6.3.1 には画風変換結果の違いを、5.6.3.2 には物体検出結果の違いをまとめる。また、図 47 に今回用いるスタイル画像とそのヒストグラム分布を示す。

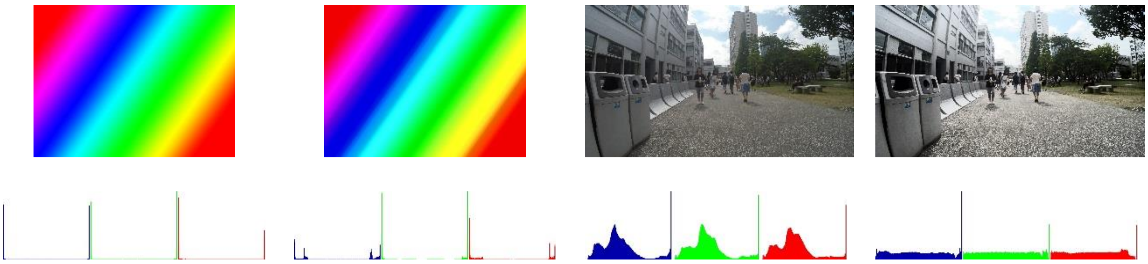


図 47　使用スタイル画像とそのヒストグラム分布

5.6.3.1　ヒストグラム分布の違いによる画風変換結果への影響

はじめに、ある画像とその画像にヒストグラム平均化処理をした画像を用いてそれぞれ画風変換した際に、画風変換結果にどのような影響を及ぼすのかを確認する。図 48 にヒストグラム分布の違いによる画風変換結果への影響を示す。

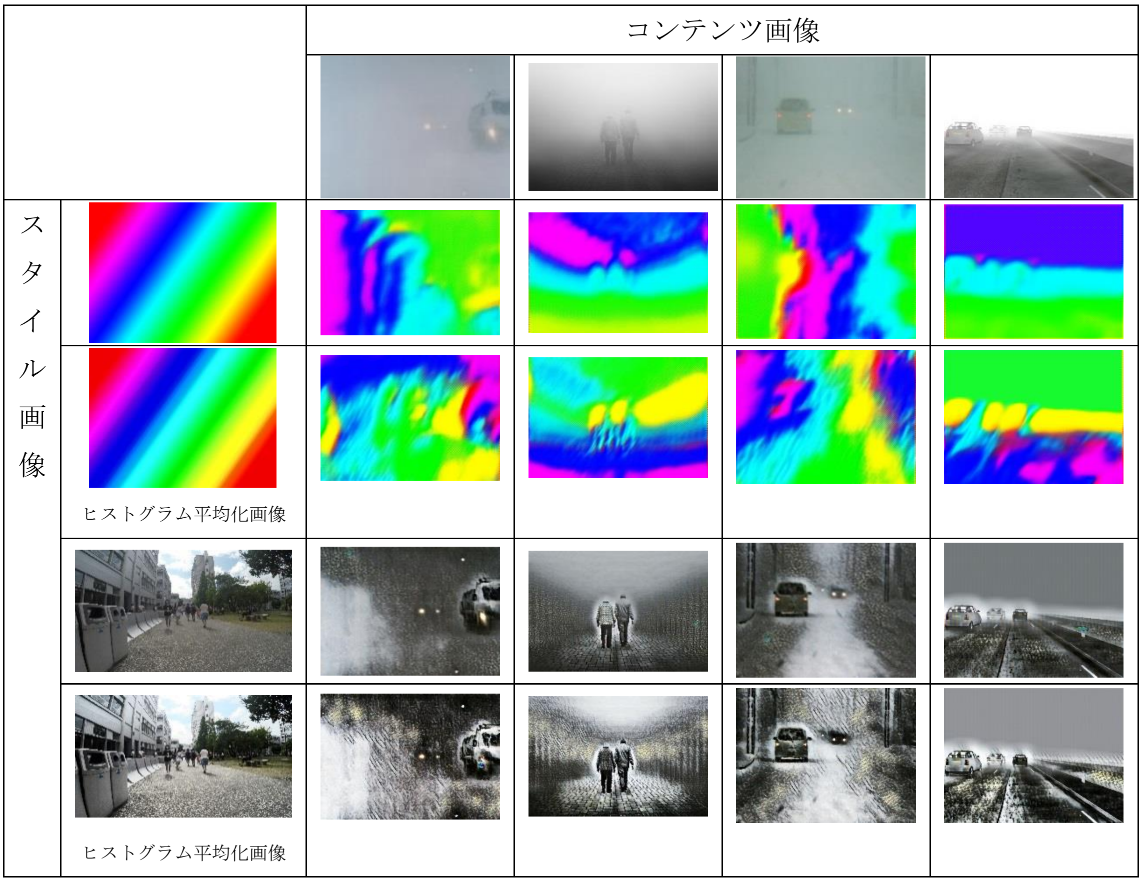


図 48　スタイル画像のヒストグラム分布の違いによる画風変換結果への影響

図 48 より、ヒストグラム分布の前処理をスタイル画像に行ってから画風変換を行うと、出力画像のコントラストも強くなる影響が見られた。また、画像に写る物体を背景として捉えられている度合いが、原画像を用いた場合よりもヒストグラム平均化画像を用いた場合のほうが強く見える。

5.6.3.2　ヒストグラム分布の違いによる物体検出結果への影響

次に、5.6.3.1 の画風変換後画像を用いて物体検出を行った結果を示す。

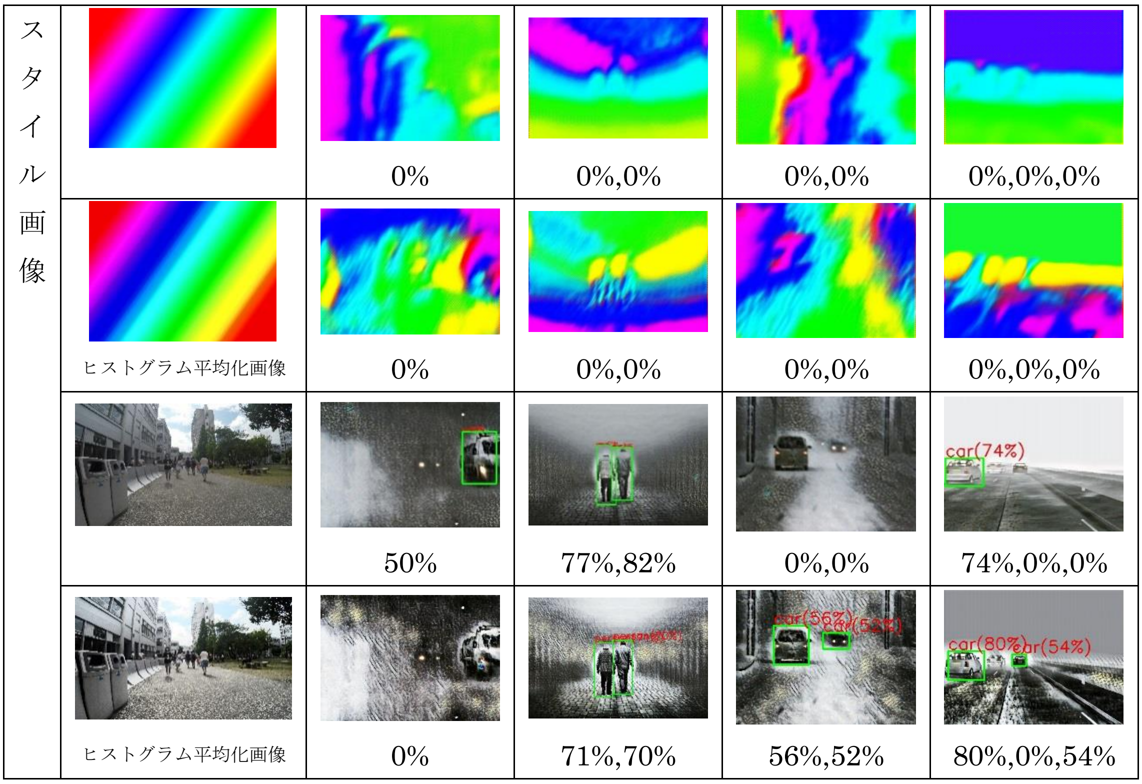


図 49　スタイル画像のヒストグラム分布の違いによる物体検出結果への影響

図 49 より、ヒストグラム平均化画像をスタイル画像として使用した場合に物体検出精度が上がる例と下がる例をそれぞれ確認した。図 49 を見ると、検出したい物体が遠くに写っている場合にはヒストグラム平均化処理を前処理として加えたほうが物体検出精度が高くなる傾向があると推測できる。

第6章　　総括

6.1 まとめ

本論文では、不鮮明画像の鮮明化手法として、従来手法とは全く異なるアプローチである画風変換を用いる手法を提案した。

実験により、画風変換を用いた不鮮明画像の鮮明化は、HAZE 画像及び夜間画像に対して有効であることを確認した。また、鮮明化後画像に写る物体の検出精度向上も試みた。特に、検出したい物体が遠くに写っている場合には、スタイル画像にヒストグラム平均化処理等の前処理を行うことにより、検出精度が向上する例を確認した。

6.2 今後の課題

本手法では、画風変換を用いた不鮮明画像の鮮明化手法を提案している。しかし、画風変換処理により得られる出力画像の予測は非常に困難である。画風変換は入力コンテンツ画像と入力スタイル画像を基に、CNN を用いて誤差逆伝播を繰り返しながら少しずつ画像を修正するため、スタイル画像の画素情報をわずかに変えただけでも出力画像には大きな影響を与える。本手法のさらなる改善点として、スタイル画像が出力画像に与える影響の具体的な傾向を掴むことや、スタイル画像の適切さを定量的に評価する評価関数の導入すること等が挙げられる。

謝辞

本研究を行うにあたり、日頃よりゼミや発表の場でご指導をいただき大変貴重な助言を頂いた甲藤二郎教授に心より感謝申し上げます。

また、研究を進める上で多くの有益なアドバイスを頂いた金井謙治様、竹内健様をはじめとする先輩方に深く御礼申し上げます。

最後に、研究だけでなく様々な面でお世話になりました甲藤研究室の皆様に深く感謝いたします。

2018 年 1 月 30 日

高橋 沙季

参考文献

[1] “AlphaGo”, https://deepmind.com/research/alphago/.

[2] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, “A Neural Algorithm of Artistic Style,” CVPR2015.

[3] 福井宏, “Deep Learningを用いた歩行者検出の研究動向”, 信学技報, vol. 116, no. 366, pp. 37-46, 2016年12月.

[4] Navneet Dalal, Bill Triggs, “Histograms of Oriented Gradients for Human Detection,” CVPR 2005.

[5] A. Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ”ImageNet classification with deep convolutional neural networks,” NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems,vol.1 ,pp.1097-1105, 2012.

[6] 柳井 啓司, “一般物体認識の現状と今後,” 情報処理学会論文誌, vo.48, no. SIG 16, Nov.2007.

[7] Kaiming He, Jian Sun, Xiaoou Tang, ”Single Image Haze Removal Using Dark Channel Prior,” CVPR2009.

[8] yusuketomoto, “chainer-fast-neuralstyle”, https://github.com/yusuketomoto/chainer-fast-neuralstyle.

[9] Justin Johnson, Alexandre Alahi, Li Fei-Fei, ”Perceptual Losses for Real-Time Style Transfer and Super-Resolution,” ECCV2016.

[10] “COCO–Common Objects in Context”, http://mscoco.org/home/.

[11] Leon A. Gatys, Matthias Bethge, Aaron Hertzmann, Eli Shechtman, “Preserving Color in Neural Artistic Style Transfer,” https://arxiv.org/abs/1606.05897.

[12] Manuel Ruder, Alexey Dosovitskiy, Thomas Brox, “Artistic style transfer for videos,” GCPR2016.

[13] Fujun Luan, Sylvain Paris, Eli Shechtman, Kavita Bala, “Deep Photo Style Transfer,” CVPR2017.

[14] “色を保ったまま画像のスタイル変換を行う手法”, http://qiita.com/dsanno/items/08ad2af32ad834c757dc.

[15] “画風のニューラルネットワークのアルゴリズム（人工知能が美術や音楽を作るMagenta 翻訳②)”, http://qiita.com/YoshikawaMasashi/items/d45861e4fbb99f1675dd.

[16] “ディープラーニングを用いて「写真の見た目の特徴」を別の写真に転送してしまう「Deep Photo Style Transfer」”, http://gigazine.net/news/20170329-deep-photo-style-transfer/.

[17] “Artistic Style Transfer with Deep Neural Networks”,
https://shafeentejani.github.io/2016-12-27/style-transfer/.
[18] “Keras で VGG16 を使う”,
http://aidiary.hatenablog.com/entry/20170104/1483535144.
[19] mattya,“chainer-gogh”, https://github.com/mattya/chainer-gogh.
[20] “Style Transfer in Real-Time”,
https://shafeentejani.github.io/2017-01-03/fast-style-transfer/.
[21] 藤吉弘亘, “Gradient ベースの特徴抽出-SIFT と HOG-”,
http://www.hci.iis.u-tokyo.ac.jp/~ysato/class14/supplements/sift_tutorial-Fujiyoshi.pdf.
[22] Tomoki Watanabe, Satoshi Ito, Kentaro Yokoi, “Co-occurrence Histograms of Oriented Gradients for Pedestrian Detection”,
https://www.jstage.jst.go.jp/article/ipsjtcva/2/0/2_0_39/_article.
[23] 矢野達也, “車載単眼カメラによる歩行者向き推定に関する研究”, https://repository.dl.itc.u-tokyo.ac.jp/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=5852&item_no=1&page_id=28&block_id=31.
[24] 藤吉弘亘,”Joint 特徴量を用いた人検出”, http://www.vision.cs.chubu.ac.jp/joint_hog/pdf/Joint+RAB.pdf.
[25] 尾崎貴洋, “Joint HOG 特徴を用いた 2 段階 AdaBoost による車両検出”,
http://vision.cs.chubu.ac.jp/04/pdf/ITS07.pdf.
[26] MPRG 中部大学機械知覚&ロボティクス研究グループ,”局所特徴量と統計学習手法による物体検出”, https://www.slideshare.net/MPRG_Chubu_University/ss-32258845.
[27] YOLO, “YOLO”, https://pjreddie.com/darknet/yolo/.
[28] leetenki,”YOLOv2”, https://github.com/leetenki/YOLOv2.
[29] “線形濃度変換”, http://www.hamanoweb.com/blog/?p=3323.
[30] “ガンマ補正”, http://www.hamanoweb.com/blog/?p=3408.
[31] “階調イコライゼーション（ヒストグラム均一化）”, http://www.mis.med.akita-u.ac.jp/~kata/image/equalize.html.
[32] 東 明浩,“「霧」や「もや」などをクリアにする高速画像処理技術”, http://www.fujitsu.com/downloads/JP/archive/imgjp/jmag/vol64-5/paper10.pdf.
[33] “INFOTECH-Red Super Eye G2”,
http://www.infotech-japan.co.jp/rse2_3ghd_s.html.
[34] “Red Super Eye G2”, http://www.infotech-japan.co.jp/red_super_eye.html.
[35] ”Red Super Eye G2 製品資料”, http://www.infotech-japan.co.jp/_src/sc761/RedSuperEyeG2_JPN2017.pdf.

[36] “local contrast”, https://jp.mathworks.com/help/images/ref/localcontrast.html?requestedDomain=true.
[37] “Haze Level Estimation Dataset”, https://www.cs.rochester.edu/u/yli/haze-level.html.
[38] “FRIDA and FRIDA2”, http://perso.lcpc.fr/tarel.jean-philippe/bdd/frida.html.
[39] “地吹雪時の運転のお願い”, http://www.melmac.co.jp/blog/archives/6644.
[40] ”年を越す前にやっておきたい事”, https://ameblo.jp/decto/entry-12335763105.html.

発表文献リスト

[1] 高橋沙季, 竹内健, 折橋翔太, 甲藤二郎, "赤外線熱画像を用いた可視光画像の人物検出精度改善の検討," 映像情報メディア学会冬季大会, Dec. 2015.

[2] 高橋沙季, 竹内健, 折橋翔太, 甲藤二郎, "HOG 特徴量と温度情報を用いた人物検出精度改善," 電子情報通信学会総合大会, Mar. 2016.

[3] 折橋翔太, 雑賀新太郎, 高橋沙季, 竹内健, 甲藤二郎, "エッジ抽出を用いた列車車載カメラ映像におけるレール検出," 電子情報通信学会総合大会, Mar. 2016.

[4] S. Saika, S. Takahashi, M. Takeuchi and J. Katto, "Accuracy Improvement in Human Detection Using HOG Features on Train-Mounted Camera", IEEE Global Conference on Consumer Electronics, Oct. 2016.

[5] 高橋沙季, 雑賀新太郎, 竹内健, 金井謙治, 甲藤二郎, "可視光画像と熱画像を併用した人物検出精度改善の一検討," 映像メディア処理シンポジウム, Nov. 2016.

[6] 雑賀新太郎, 高橋沙季, 竹内健,甲藤二郎, " HOG 特徴量を用いた列車車載カメラ映像における人物検出," 映像メディア処理シンポジウム, Nov. 2016.

[7] 高橋沙季, 竹内健, 金井謙治, 甲藤二郎, "CNN を用いた画風変換による不鮮明画像の改善," 映像メディア処理シンポジウム, Nov. 2017.

[8] 高橋沙季, 竹内健, 金井謙治, 甲藤二郎, "CNN を用いた画風変換による物体検出精度の改善," 情報処理学会オーディオビジュアル複合情報処理研究会, Dec. 2017.