第3章

理論

ニューラルネットワークを用いたStyle Transferの研究では, 畳み込みニューラルネットワークやGenerative Adversarial Network を応用したモデルや手法がよく用いられている．そこで，本研究の前提知識として 3.1節では畳み込みニューラルネットワークについて説明し，3.2節ではGenerative Adversarial Networkとその改良について説明する．

また本章の最後の3.3節では，ある文字列の形状の歪みを別の文字列へと転写するために本研究で用いた射影変換について説明する．

3.1 畳み込みニューラルネットワーク

畳み込みニューラルネットワーク（Convolutional Neural Network: CNN）とは，畳み込み（Convolution）演算をはじめとする画像処理の技法をニューラルネットワークに応用したものである．そのため，CNNは他のニューラルネットワークに比べて画像データを扱うことに長けており，主に画像認識の分野で広く用いられている．

先述したように，CNNには画像処理の技法が応用されているため， CNNを十分に理解するためには画像処理の技法の知識が必要不可欠だと考える．ゆえに，コンピュータ上での画像データの扱いと，画像処理の技法について先に説明する．

初めに，コンピュータ上での画像データの扱いについて説明する．

一般に，コンピュータ上における画像データのフォーマットは「ラスタ形式」のものと「ベクタ形式」のものの二つに大別される．これらのうち，ラスタ形式では画像を構成する各ピクセルの色に対応する数値を全て格納した配列[*1]として画像データを表現する．なお，この「ピクセルの色に対応する数値」は「画素値」や「ピクセル値」などと呼ばれている．一方，ベクタ形式では画像に描かれているものを表現するための座標や数式などのあつまりとして画像データを表現する．よって，写真のように色や形状が複雑な画像の場合，ベクタ形式よりもラスタ形式が適していると言える．また，ラスタ形式の画像データはピクセル値の配列であるため，コンピュータ上で画像データのピクセル値を参照した処理を行う際にも，ベクタ形式よりもラスタ形式が適していると言える．

以下では，絵や写真を含む様々な画像のコンピュータを用いた処理について考えるため，単に画像と言った場合はラスタ形式の画像を指すものとする．

ここで，後の説明のために画像の「チャネル（channel）」について説明する．

その前提として，白黒でない画像のピクセル値が複数の成分をもつことを確認する．白黒画像では，輝度を表すスカラをピクセル値とすれば良く，ピクセル値の成分は一つでよい．しかし，カラー画像では色の指定に三原色[*2]の混合比率が必要なので，ピクセル値には三つの成分が必要になる．さらに，透過画像では三原色に対応した三つの成分に加え，ピクセルの不透明度[*3]が必要なので，ピクセル値には計四つの成分が必要になる．以上から，白黒でない画像のピクセル値が複数の成分をもつことが確認できた．

このことから，白黒でない画像は複数の白黒画像で構成されているものだとみなせるが，このように「ある画像中の特定の成分を抽出してできる白黒画像」はチャネルと呼ばれる．例えば，通常のカラー画像は赤のチャネル，緑のチャネル，青のチャネルの三つの白黒画像から構成されているとみなせる．

なお，CNNによって処理されている途中の画像は，通常の画像より遥かにチャネル数が多い．これは各チャネルが元画像の何らかの特徴を表すためである．（詳しくは後述する．）

[*1] 但し，圧縮については考慮しないとする．

[*2] 光の三原色である赤（R）・緑（G）・青（B）

[*3] アルファ値とも呼ばれる

次に，画像処理の技法のうち畳み込み演算を用いたフィルタリングについて説明する．

畳み込み演算を用いたフィルタリングとは，対象の画像と「カーネル*4」と呼ばれる画像との畳み込み演算である．そのため，フィルタリングに用いるカーネルがどのようなものかによってフィルタリング後の結果が大きく左右される．（詳しくは後述する．）

ここからは，図 3.1aに示す5×5ピクセルの白黒画像を対象に， 図 3.1bに示すカーネルを用いてフィルタリングする場合を例にして動作を説明する．

1	6	11	16	21
12	15	2	17	22
13	18	20	18	23
4	9	14	10	24
5	10	15	20	5

(a) フィルタリング対象の画像

0	-1	0
0	1	0
0	0	0

(b) フィルタリングに用いるカーネル

図3.1　例として用いる画像とカーネル
（文献[1]の<u>p157</u>図8.2をもとに作成）

対象の画像とカーネルとの畳み込み演算では次の図 3.2に示すような処理が基本となる．

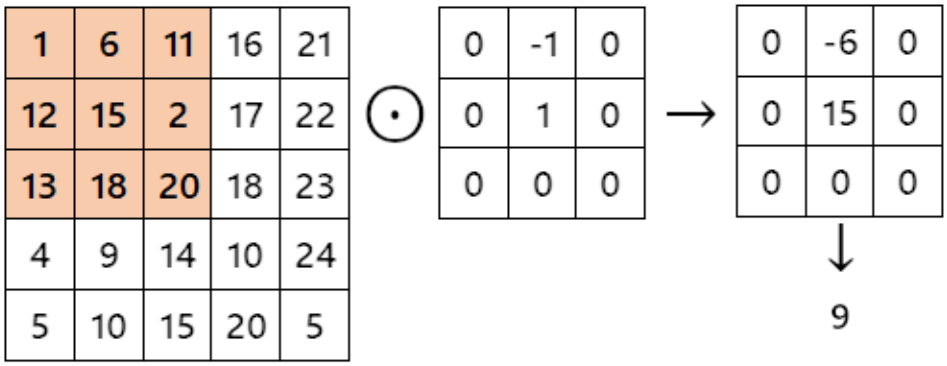

図3.2　畳み込み演算の過程
（文献[1]の図<u>p158</u>図8.2をもとに作成）

*4 画像フィルタなどとも呼ばれる

図 3.2に示した処理は次の2つのステップに分けられる．

Step1. フィルタリング対象の画像からカーネルと同じサイズの小さな画像を切り出す．
また，以下ではここで切り出した小さな画像のことを「パッチ」と呼ぶとする．
なお，図 3.2では切り出すパッチ内のピクセルに色を付けて示した．

Step2. Step1で切り出したパッチとカーネルの標準内積をとる．
なお，図 3.2では標準内積を要素積と総和の二段階に分けて表記した．

対象の画像とカーネルとの畳み込み演算とは，パッチを切り出す範囲を一定幅で移動させつつ上記の処理を繰り返し実行し，得た数値を順に並べて新たな画像を作成する処理である．なお，この「パッチを切り出す範囲をずらす幅」はストライド[*5]と呼ばれている．

例えば，次の図 3.3は図 3.1の組をストライドを1として畳み込み演算した結果である．

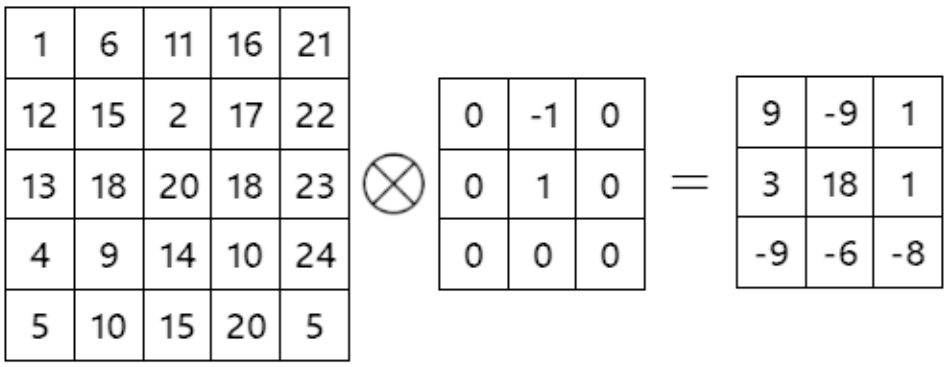

図3.3　畳み込み演算の結果
（文献[1]の図p158図8.2をもとに作成）

図 3.3を見ると分かるように，カーネルとの畳み込み演算を行うと画像のサイズは元画像より小さくなる．そこで，出力画像サイズの担保のため，畳み込み演算の実行前に元画像を拡張する手法が用いられることがある．この手法は「パディング」と呼ばれている．

また，カーネルとの畳み込み演算の過程で標準内積が用いられていることから分かるが，畳み込み演算を用いたフィルタリングはカーネルとの相関 [*6] の計算を意味している．

[*5] スライド幅とも呼ばれる

[*6] CNNなどの分野で畳み込み演算と呼ばれている演算は，厳密に言えば数学の分野では相互相関にあたる．本来の畳み込みは系列の積をとる際，逆順で積をとるように定義されている．

続いて，画像処理の技法のうちプーリング（Pooling）について説明する．

プーリングとは，入力画像の特徴を最大限保持しつつ画像サイズを縮小する処理である．プーリングにおいても出力画像のピクセル値は入力画像のパッチに基づいて決定されるが，その際にパッチ内のピクセル値の最大値を採用する手法は「最大値プーリング」と呼ばれ，平均値を採用する手法は「平均値プーリング」と呼ばれている．

本節の最後に，以上の説明を踏まえてCNNについて説明する．

TODO:まとめとCNN どのネットワークを例にするか？畳み込み演算によって入力画像の特徴量抽出を行う層を畳み込み層（Convolution layer）プーリングによって画像サイズの縮小を行う層をプーリング層（Pooling layer）と呼ぶ

3.2 Generative Adversarial Networkとその改良

3.2.1 Generative Adversarial Network

Generative Adversarial Network（GAN）とは，「乱数などの入力を元に新たなデータを生成するネットワーク」であるGeneratorと「入力されたデータがGeneratorが生成したデータか，訓練データかを判断するネットワーク」であるDiscriminatorの二つのネットワークからなる生成モデルである． GANは2014年に Goodfellowら[2]によって提案されて以降，幅広い分野で応用されてきたが，本研究で扱うStyle Transferもそうした応用先の一つである．

TODO：

3.2.2 Wasserstein GAN

3.2.1節で説明したGANには次のような問題点があることが知られている．

TODO：

以上のようなGANの問題点を解決するため， Wasserstein GAN(WGAN)は Arjovskyら[3]によって提案された． GANの損失関数をWasserstein距離を用いたものに変更した

TODO： WGANでは，

3.2.3 WGAN with a gradient penalty

3.2.2節で説明したWGANはGANの問題点を改善したものの完璧ではない

TODO：

Gulrajaniら[4]は，WGANにgradient penalty項を追加することを提案した．これはgradient penaltyをgpと略してWGAN-gpと呼ばれている．

3.3 射影変換

TODO：

参考文献

[1] 藤田毅, *C++*で学ぶディープラーニング. エキサイト株式会社, 2017.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014.

[3] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, PMLR, Jun. 2017, pp. 214–223.

[4] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.