Vol.2020-CG-180 No.3
Vol.2020-DCC-26 No.3
Vol.2020-CVIM-223 No.3
2020/11/5

カラーイラストからの微分可能評価指標を用いた線画抽出

金森 由博[1,a)]　松田 祐紫[1,b)]　遠藤 結城[1,c)]

概要：カラーイラストの線画は一般に濃淡を持つ線で描かれており，この線を抽出できればアーティストの筆致の解析などに有益である．しかし我々の知る限りでは，線画の濃淡まで考慮して線画抽出を行う既存手法はない．本研究では，グレースケール線画のための微分可能な適合率，再現率，F 値などの評価指標を定義し，これらから損失関数を導出して，畳み込みニューラルネットワークに基づく教師あり学習を行う．実験により，従来の二値画像向けの損失関数と比べて精度が向上することを確認した．さらに注意機構を導入し，誤って黒くあるいは白く出力された画素を補正できるようにすることで，線画抽出の精度向上を図る．

Line Extraction from Color Illustrations with Differentiable Metrics

1. はじめに

カラーイラストの線は一般に，白黒の二値ではなく濃淡を持つ．もし濃淡を含む線画をカラーイラストから抽出できれば，アーティストの筆致の解析，印刷されたカラーイラストのデジタルリマスタリング，さらに線画の自動着色のための教師データ作成などに有用である．しかし，下絵の線画の段階ではグレースケール画像として描かれていても，イラストによっては仕上げる途中で線にもところどころ色付けされる場合があるため，例えば単に色だけを見て線画を抽出することは難しい．

自然画像からの線画の抽出はコンピュータビジョン分野で盛んに研究が行われてきた．最近では畳み込みニューラルネットワーク (CNN) を用いた教師あり学習の枠組みにおいて，線画抽出に関する評価指標である dice 係数 (F1 値と等価) を微分可能な形で定義し，これを損失関数として学習する手法 [1] が提案されている．しかし文献 [1] で定義された dice 係数は，線画が二値画像として定義されていることが前提となっている．我々の知る限りでは，濃淡を持つ線画を対象とした dice 係数はこれまで提案されていない．

[1] 筑波大学
University of Tsukuba, Tennoudai 1-1-1, Tsukuba, Ibaraki, 305–8573, Japan
[a)] kanamori@cs.tsukuba.ac.jp
[b)] matsuda@cgg.cs.tsukuba.ac.jp
[c)] endo@cs.tsukuba.ac.jp

そこで本研究では，濃淡を持つ線画を対象として，適合率，再現率，F1 値を微分可能な形で定義し，これらに基づいて損失関数を導出して学習することを提案する．まず，二値画像に対する適合率および再現率をグレースケール画像に拡張し，その定式化において現れる微分不能な関数を近似的に微分可能な関数に置き換える．F1 値はこれらの適合率および再現率から導出し，学習には近似を含む評価指標を損失関数として用いる．実験により，二値画像のための dice 係数を用いた場合よりも，我々の損失関数を用いた方が，近似のない評価指標で評価した際に性能が向上することを確認した．さらにネットワークに注意機構 (attention) を導入し，ネットワークの出力した線画において誤って黒くあるいは白くなった部分を補正できるようにすることで，精度向上を図る．

2. 関連研究

画像からの線画抽出は，コンピュータビジョンの分野で古くから研究されてきた．輝度勾配などに着目した古典的な手法は，本研究が対象とするカラーイラストの線画に対しては，線の色が必ずしも単色ではなく，線の濃淡まで必要となることから，良好な結果が得られない．Mao ら [2] はグラフカットに基づいた，カラーイラストからの線画手法を提案している．我々がグレースケール線画の抽出を目的としているのと異なり，彼らの手法で得られるのは白黒の線画である．

近年では CNN を用いた線画抽出の手法が多数提案され

Vol.2020-CG-180 No.3
Vol.2020-DCC-26 No.3
Vol.2020-CVIM-223 No.3
2020/11/5

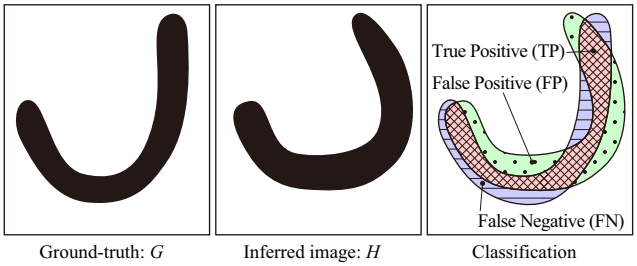

図 1 二値線画における正解と不正解の分類.

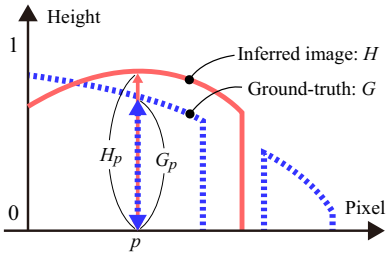

図 2 線画の濃淡を反転し，線の濃さを高さマップの高さ (図中の赤線と青点線) とみなす.

ている．CNN に基づく従来の研究 [3], [4] では，線が太めに抽出されるという不具合があった．これに対し Deng [1] らの研究では，損失関数としてバイナリクロスエントロピーと線画に対する dice 係数の逆数を用いることで，細い線を抽出できるようにした．しかし Deng [1] らの dice 係数の定式化では，正解および抽出される線がともに白黒の二値であることを仮定している．

松田ら [5] は，我々と同様に，CNN を用いた教師あり学習によってカラーイラストから線画を抽出する手法を提案した．彼らの手法では，周囲とのコントラストが低い線を強調するため，グレースケール化した入力イラストにラプラシアンフィルタを適用して得られるマップを線の重み付けに用いた．また再現率を高めるため，薄い線まで抽出されやすいモルフォロジ演算による線画抽出法 [6] で得られた線画を，ネットワークへの入力として追加した．しかし十分な精度は得られなかった．

本研究では，グレースケール画像のための微分可能な F1 値を損失関数として学習に用い，さらに attention module による補正処理を加え，精度を向上させる．

3. 提案手法

提案手法は，カラーイラストを入力とし，CNN を用いた教師あり学習によってグレースケールの線画を抽出する．以下，グレースケール線画に対する微分可能評価指標を説明する前に，基礎として白黒線画に対する評価指標 (3.1 節) を説明する．続いてそれをグレースケール線画に拡張した後 (3.2 節)，微分可能にする方法 (3.3 節) を述べる．さらに本研究での注意機構の導入方法 (3.4 節) を説明する．

3.1 白黒線画に対する評価指標

まず基本として，白黒の二値の線画に対する評価指標を考える．**図 1** に正解線画 G，推定した線画 H と，それらによって定義される True Positive (TP), False Positive (FP), False Negative (FN) の分類を示す．簡単のため，線画 I に対して線 (黒地) の領域を I，白地の領域を $\bar{I}$ と表記することにし，それらの領域に含まれる画素数を数える演算子を $\#(\cdot)$ とする．すると適合率 (Precision)P および再現率 (Recall)R は，定義よりそれぞれ次のように表せる．

$$P = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} = \frac{\#(G\cap H)}{\#(G\cap H)+\#\left(\bar{G}\cap H\right)} = \frac{\#(G\cap H)}{\#(H)} \quad (1)$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} = \frac{\#(G\cap H)}{\#(G\cap H)+\#\left(G\cap \bar{H}\right)} = \frac{\#(G\cap H)}{\#(G)} \quad (2)$$

また F1 値は定義より，次のように表せる．

$$\mathrm{F1} = \frac{2\,R\cdot P}{R+P} = \frac{2\,\#(G\cap H)}{\#(H)+\#(G)} \quad (3)$$

3.2 グレースケール線画に対する評価指標

上記の二値線画に対する評価指標を，グレースケール線画に拡張する．便宜上，濃淡を反転させ，1 が黒，0 が白を表すものとする．式 (1) と (2) はそれぞれ，出力線画 H または正解線画 G に対する，正解と推定結果における線領域の共通部分の割合を表している．以降，これらの式を分子と分母に分けて，グレースケール線画への拡張を試みる．

分子に現れる $\#(G\cap H)$ は，二値線画の場合，正解と推定結果における線領域の共通部分の画素数を表す．これをグレースケール線画に拡張するにあたり，まず**図 2** に示すように，グレースケール線画を画素値を高さとする高さマップと見なす．そして各画素 p での正解および推定結果の高さ $G_p, H_p \in [0,1]$ を用いて「共通部分」の値を定義する．まず，正解の高さ G_p と推定結果の高さ H_p に対し，共通の高さは $\min\{G_p, H_p\}$ と書ける．この値が，正解と推定結果が一致する場合に 1 となるよう正規化したい．しかし単純に G_p(または H_p) で除算すると，他方の H_p(または G_p) の値がより大きい場合，除算結果が常に 1 となってしまう．そこで代わりに $\max\{G_p, H_p\}$ で除算する[*1]．

一方，分母に現れる $\#(H)$ と $\#(G)$ は，二値線画の場合，それぞれ推定結果と正解における線領域の画素数を表す．グレースケール線画の場合，推定結果と正解が少しでも線を含む領域，すなわち (濃淡を反転した結果)，$H_p > 0$ および $G_p > 0$ となる画素数を数えることにする．

以上をまとめると，グレースケール線画の場合の適合率，再現率，F1 値は次のように書ける．

[*1] 実際の数値計算ではゼロ除算を防ぐため，微小な正の数 ϵ を用いて $\max\{G_p, H_p, \epsilon\}$ で除算する．

Vol.2020-CG-180 No.3
Vol.2020-DCC-26 No.3
Vol.2020-CVIM-223 No.3
2020/11/5

$$P = \frac{\sum_p \frac{\min\{G_p, H_p\}}{\max\{G_p, H_p\}}}{\sum_p [H_p > 0]} \tag{4}$$

$$R = \frac{\sum_p \frac{\min\{G_p, H_p\}}{\max\{G_p, H_p\}}}{\sum_p [G_p > 0]} \tag{5}$$

$$\mathrm{F1} = \frac{2 \sum_p \frac{\min\{G_p, H_p\}}{\max\{G_p, H_p\}}}{\sum_p [H_p > 0] + [G_p > 0]} \tag{6}$$

ここで $[x]$ は指示関数を表し，x が真のとき 1，偽のとき 0 を返す．なおこれらの式は，線画が二値，すなわち $G_p, H_p \in \{0, 1\}$ の場合にそれぞれ式 (1), (2), (3) と一致することから，二値線画の場合の自然な拡張となっている．

3.3 微分可能な評価指標の導出

上述のグレースケール線画のための評価指標を損失関数として用いる場合，分母の指示関数が微分不能であることが問題となる．より具体的には，$[G_p > 0]$ は定数であるため問題ないが，$[H_p > 0]$ は推定値 H_p に基づいて計算され，$H_p > 0$ となるすべての画素 p に対して勾配がゼロとなってしまう．結果的に $[H_p > 0]$ もほぼ定数のように振る舞い，適合率 P も再現率 R と同様な振る舞いをすることになる．すると学習が進むにつれ，「再現率が上がる」すなわち「線領域が増える」ことから，線で塗りつぶされる領域ばかり増えていくことになる．

そこで指示関数を近似的に微分可能な関数に置き換え，$H_p > 0$ となる画素 p に対しても勾配が計算できるようにする．具体的な候補として，次のような区分線形関数 $f_{linear}(x)$ および指数関数 $f_{power}(x)$ を検討した (**図 3**)．

$$f_{linear}(x) = \begin{cases} 0 & \text{if} \quad x < 0, \\ \dfrac{y_0}{x_0} x & \text{if} \quad 0 \le x < x_0, \\ \dfrac{1 - y_0}{1 - x_0}(x - x_0) + y_0 & \text{if} \quad x_0 \le x < 1, \\ 1 & \text{otherwise} \end{cases} \tag{7}$$

$$f_{power}(x) = \begin{cases} 0 & \text{if} \quad x < 0, \\ x^{\frac{1}{\gamma}} & \text{if} \quad 0 \le x < 1, \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

ここで (x_0, y_0) および γ は定数である．予備実験において，$f_{linear}(x)$ における (x_0, y_0) の候補として $\{(0.1, 0.9), (0.05, 0.95)\}$，$f_{power}(x)$ における γ の候補として $\{100, 200\}$ を試した．これらの計 4 通りの候補のうち，$(x_0, y_0) = (0.1, 0.9)$ として f_{linear} を用いた場合が最良であったため，これ以降の結果ではこの条件を用いた．なお最良となった理由は恐らく，この条件のときに勾配の絶対値が最も大きくなり，学習が促進されたためではないかと思われる．

上記で検討した微分可能な関数 $f(x)$ を指示関数の代わりに用いると，特に x がゼロに近い場合に微小な値で除算することになり，真の値よりも過大になってしまう．そこで分母の $[H_p > 0]$ の代わりに $f(H_p)$ を用いる場合，分子に $f(H_p)$ を掛けて相殺する．まず適合率の式 (4) を次式で置き換える．

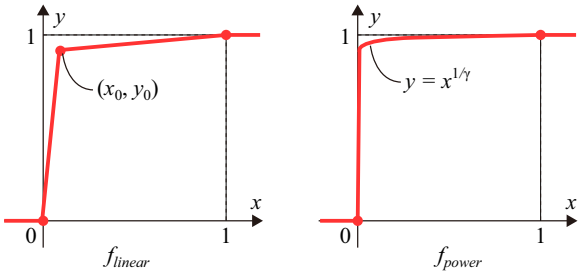

図 3 指示関数の近似に用いる微分可能な関数の候補．

$$\tilde{P} = \frac{\sum_p f(H_p) \frac{\min\{G_p, H_p\}}{\max\{G_p, H_p\}}}{\sum_p f(H_p)} \tag{9}$$

そして式 (9), (5) を式 (3) に代入することで，F1 値は以下のように計算できる．

$$\tilde{\mathrm{F1}} = \frac{2\, R \cdot \tilde{P}}{R + \tilde{P}} \tag{10}$$

損失関数としては，Deng らの手法が dice 係数の逆数を用いたように，本研究では式 (10) の逆数を損失関数として用いる．なお，上記の相殺の処理はヒューリスティックな方法であるが，これを行わない場合と比べ，行った場合の方が精度が向上することを確認した．

3.4 注意機構の導入

本研究では微分可能評価指標に加えて，線画抽出の精度を高めるため，注意機構 (attention) をネットワークに組み込む．本研究で用いるネットワークは，U-Net のようにスキップ接続で繋げられたエンコーダとデコーダからなるネットワークであり，エンコーダとデコーダの間に 2 つの残差ブロックを含む (詳細は付録 A.1 参照)．出力層は入力画像と同じ解像度を持つ，1 チャンネルの画像 H を線画として出力する．この出力線画 H に対し，何らかの方法で attention map を導出して要素積を計算する場合，もし attention map の各要素が $[0, 1]$ の値を持つなら，要素積の計算結果は暗くなる (ゼロに近づく) ことはあっても明るくなる (1 に近づく) ことはない．一方，出力線画 H の画素が誤って暗くなっている場合に明るく補正しようとすると，1 を超える係数を掛ける必要があり，その係数に上限はなく，attention map の出力値を制御しづらい．

そこで本研究では，出力線画 H とそれを反転した画像 $1 - H$ に対し，線領域を強調するための darkening attention module と，白地の背景領域を強調する brightening attention module をそれぞれ適用する (これらの module の詳細は付録 A.2 参照)．これらの attention module の出

Vol.2020-CG-180 No.3
Vol.2020-DCC-26 No.3
Vol.2020-CVIM-223 No.3
2020/11/5

力は，1 チャンネルの画像で各要素は $[0,1]$ の値を持てばよく，それぞれ darkening attention map M^d，brighting attention map M^b と呼ぶことにする．M^d および M^b を用いて要素積 $M^d * H$ と $M^b * (1-H)$ を計算し，これらの計算結果をチャンネル方向に連結した後，1×1 畳み込みおよびシグモイド関数を適用して，最終的な線画を出力する．

4. 実験結果

実装には Python ライブラリである PyTorch を使用し，NVIDIA GeForce GTX 1080 Ti を用いて学習を行った．実験に用いたデータセットおよび基本的な実験設定は，松田らの研究 [5] と同様であり，以下，再掲する．データセットは pixiv から収集したカラーイラストと線画のペアからなる．全てのカラーイラストは白背景に統一されており，訓練データは 60 組，テストデータは 15 組である．訓練時にはデータ拡張として，各エポックでランダムでスケーリング，回転（90 度単位），左右反転を適用したのち，256×256 画素になるようトリミングした．バッチサイズは 15，最大エポック数は 20,000 とした．学習時の最適化手法には Adam を使用し，学習率は 0.001，$\beta_1 = 0.9$，$\beta_2 = 0.999$ とした．計算時間について，小規模なデータセットのため最初にすべてのデータを GPU に読み込んでから学習するようにしたところ，約 5 時間半で学習が完了した．損失関数は，Deng の研究に倣い，バイナリクロスエントロピーと，式 (10) で定義される F1 値の逆数を，それぞれ重み 0.001 と 1 で足し合わせたものを用いた．以下の実験結果はテストデータに対するもので，訓練データの結果は含まない．

提案手法の評価のため，以下に示す手法について比較を行った．

Exp_5: 松田らの研究 [5] で RMSE が最小となった実験条件．

Pix2pix: pix2pix [7] による結果．

Dice: Deng [1] らの手法を再現した結果．損失関数として式 (10) で定義される F1 値の逆数の代わりに，Deng らが用いた dice 係数の逆数を使用．3.4 節で述べた attention は不使用．

W/o_attn: 我々が提案した損失関数を用いるが，3.4 節で述べた attention は不使用．

Ours: 我々が提案した損失関数および attention を使用．

評価指標には，RMSE および，式 (4), (5), (6) で定義される，近似を含まない適合率，再現率，F1 値を用いた．

4.1 定量評価

前述の 5 つの手法の定量評価結果を表 1 に示す．全体的に松田らの Exp_5 より改善が認められる．RMSE は Dice, W/o_attn, Ours のいずれもほぼ同等である．再現率は Dice の値が最も高い．この結果は正解の線をより多く抽出できていることを意味し，実際に定性評価でも Dice は薄い線を過剰に抽出する傾向が見られた．適合率および F1 に関しては W/o_attn および Ours が Dice を大きく上回っている．この理由は，提案した損失関数を用いることで線の濃さをより適切に考慮でき，余分な線の抽出を抑制できたために適合率が向上し，それによって F1 値も向上したためと考えられる．さらに W/o_attn と Ours の比較から，提案した attention module による補正が有効に機能していることがわかる．

表 1 定量評価の結果．それぞれの指標で最良の結果を赤字で示す．

比較手法	RMSE↓	適合率↑	再現率↑	F1↑
Exp_5	0.111	0.507	0.659	0.560
Pix2pix	0.113	0.472	0.634	0.535
Dice	**0.106**	0.392	**0.713**	0.494
W/o_attn	0.109	0.561	0.681	0.609
Ours	0.107	**0.618**	0.673	**0.638**

4.2 定性評価

図 4 に，Pix2pix, Dice, および Ours の出力結果を示す．1, 3, 5 段それぞれの図の赤枠で囲まれた部分の拡大図を 2, 4, 6 段に示している．1, 2 段の結果を見ると，入力イラストで色の付いている線が，Pix2pix や Dice の結果では薄く掠れ気味であるのに対し，Ours の結果では全体的にはっきりと抽出できている．3, 4 段の結果では，入力イラストがスクリーントーンのようなパターンを持っており，特に Dice の結果ではそのパターンまで抽出されてしまっているが，Ours ではパターンはほぼ無視できている．5, 6 段の結果を見ると，Dice では入力イラストの色の濃い部分を線として過剰に抽出しているが，Ours では余分な線は抑制されている．

4.3 失敗例

図 5 に提案手法の失敗例を示す．提案手法は線と周囲の色のコントラストが低い場合に線の抽出に失敗する場合がある．この問題については，入力カラーイラストの局所コントラストを高めてからネットワークに入力することで改善できるのではないかと考えている．また，太い線の輪郭のみ抽出され，その内側が欠けてしまう，という不具合が見られる．これは提案手法だけでなく他の手法でも同様の傾向があり，ネットワークの構造に由来する可能性がある．例えば多重スケールの特徴を捉えられるネットワーク構造を用いることで，太さの異なる線を抽出できないかと考えている．

5. おわりに

本研究では，カラーイラストからの線画抽出の精度向上のために，CNN に基づいた教師あり学習の枠組みの下，以

Vol.2020-CG-180 No.3
Vol.2020-DCC-26 No.3
Vol.2020-CVIM-223 No.3
2020/11/5

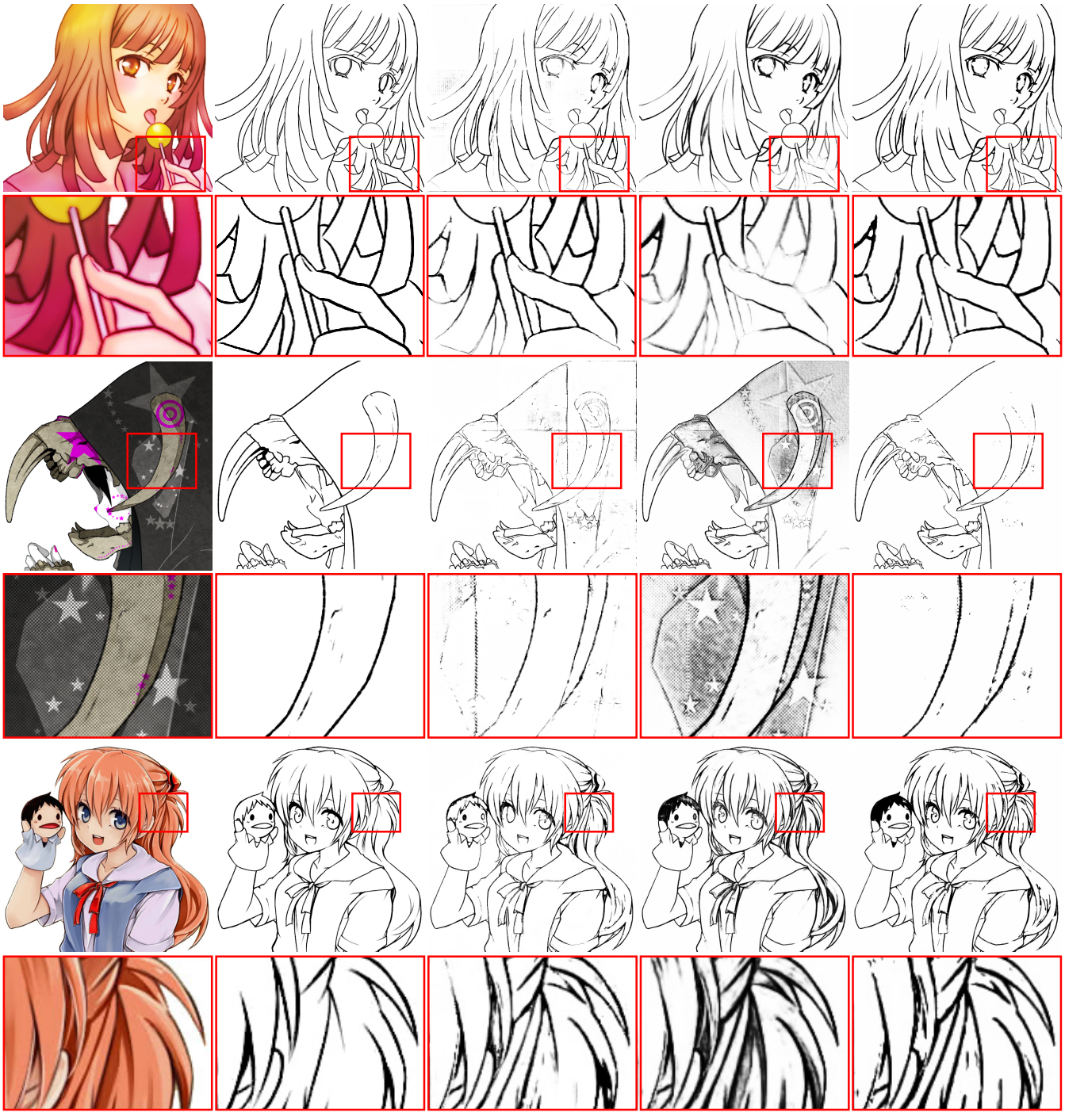

図 4　定性評価の結果．2, 4, 6 段の画像は 1, 3, 5 段の画像の赤枠部分の拡大図．

下を提案した．

- グレースケール線画のための適合率，再現率，F1 値を微分可能な評価値として定義し，損失関数に組み込んだ．
- ネットワークが出力する線画の暗い/明るい部分を補正するための attention module を導入した．

結果として，既存手法よりも定量的・定性的に優れた結果が得られた．

今後の課題として，細い線が掠れてしまったり，眉毛のように太く描かれる線の内側が欠けてしまったりすることがあるため，これらの改善が必要である．また，他の既存手法との比較や，カラーイラストに留まらず自然画像を含めた他の種類の画像での評価も行いたい．

参考文献

[1] Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. Learning to predict crisp boundaries. In *ECCV 2018*.
[2] Xueting Liu, Tien-Tsin Wong, and Xuemiao Xu. Region-

図 5 提案手法の失敗例．周囲とのコントラストが低い線や太い線の抽出に失敗する場合がある．

based structure line detection for cartoons. *Computational Visual Media*, Vol. 1, pp. 69–78, 2015.
[3] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Zhijiang Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *CVPR 2015*, pp. 3982–3991, 2015.
[4] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. *International Journal of Computer Vision*, Vol. 125, pp. 1–16, 2017.
[5] 松田祐紫, 金森由博, 遠藤結城. 非顕著な線を重視したカラーイラストからの線画抽出. 情報処理学会研究報告, コンピュータグラフィックスとビジュアル情報学 (CG), Sep 2020.
[6] カラー画像から線画を作る [OpenCV & python] - MATHGRAM. `https://www.mathgram.xyz/entry/cv/contour`. (最終アクセス日 2020-7-21).
[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR 2017*, pp. 5967–5976, 2017.

付　　録

A.1　ネットワーク構造

本研究で用いたネットワークは，畳み込み層のブロック 4 つからなるエンコーダ，2 つの残差ブロック，畳み込み層のブロック 6 つからなるデコーダから構成される．エンコーダとデコーダはスキップ接続で繋がれている．

エンコーダ部分の畳み込み層の出力チャンネル数はそれぞれ $\{32, 64, 128, 256\}$ である．いずれもカーネルサイズは 4，ストライドは 2 とした．本研究では入力イラストを 256×256 画素に切り出して処理するため，画像の境界部分で白地が保たれるよう，パディングには reflection padding を用いた．活性化関数には LeakyReLU を用いた．エンコーダの第 2, 3, 4 層には，活性化関数の適用直前にバッチ正規化を適用した．

残差ブロックはそれぞれ，256 チャンネルの畳み込み層 2 つとバッチ正規化からなり，第 1 層にのみ活性化関数として ReLU を用いた．2 層の畳み込み層の出力と入力を加算した結果を出力する．

デコーダ部分の畳み込み層の出力チャンネル数はそれぞれ $\{256, 128, 64, 32, 32, 1\}$ である．転置畳み込みを用いた場合のチェッカー模様のアーティファクトを避けるため，まず前のブロックからの入力をアップサンプリングしたのち，通常の畳み込み層で処理する．すべての層にバッチ正規化を適用する．活性化関数については，最後の層についてのみシグモイド関数を用い，それ以外の層には LeakyReLU を用いた．

ネットワークの重みは $[-0.005, 0.005]$ の範囲で一様乱数で初期化した．

A.2　Attention module の構造

本研究で用いる 2 つの attention module は同じ構造を採用した．まず学習の収束を早めるため，各画素が $[0, 1]$ の値を持つ入力を $[-1, 1]$ の範囲に線形変換する．その結果を，1×1 畳み込み，LeakyReLU，1×1 畳み込み，そしてシグモイド関数へ順に入力し，attention map を得る．畳み込み層のチャンネル数はいずれも 1 である．畳み込み層の重みは $[0, 1]$ の範囲で一様乱数で初期化した．