4Q-02

pix2pix を用いたデジタルイラスト制作における レイヤ分け作業の自動化

渡邉　優†　　阿部　清彦†　　阿倍　博信†

東京電機大学†

１．はじめに

デジタルでイラストを描くときには，ラフ→線画→色塗りの順番で進める.色塗りの工程においてレイヤ分けと呼ばれる作業をしてから，影を塗る．レイヤ分けとは，肌，髪などのパーツごとにレイヤを分けることである.

このレイヤ分け作業は，ペイントツールに存在する塗りつぶしツールや選択範囲ツールを使うことが多い.しかし,細かいところ上手く選択できないことが多い．これは人が手を加えて線を閉じ，塗りつぶされなかった部分は手動で修正する必要があり時間がかかる.

手間がかかるが,レイヤに分けることで後から色を変えることやクリッピングを行うことで,ほかのパーツに影響せずに,簡単に影を塗ることができるという大きなメリットが存在する.そのため,この作業を行う絵描きは多い.

このレイヤ分け作業を自動化することで絵を描く作業の負担を減らすことができるのではないかと考えた.本研究の目標はレイヤ分け作業を自動化し,作業効率の向上を目指すことである.また,そのためには線画からパーツの画像を認識する必要がある.本研究では conditional GAN の一種である pix2pix[1]を使用することで線画からパーツのセグメンテーションを行った.

２．関連研究

類似の研究に，ドワンゴの参照画像を用いて線画を塗る研究が存在する．また，paints Chainer[2]という Web サービスは，参照画像なしでユーザが色を指定することで線画を着色することができる．ペイントソフトに内蔵されている場合もあり,PC,ipad 向けのソフトウェアである CLIPStudio[3]に関しても同様の機能が存在する．これらの研究の課題は出力結果が背景を含め一枚の画像ファイルであるという点が挙げられる．　そのため,パーツごとに色を変えたりすることができないという問題点が存在する.

An Automatic layer splitting method for creating digital illustration using pix2pix

†Yuu Watanabe, Kiyohiko Abe and Hironobu Abe, Tokyo Denki University

３．モデルの作成

３．１ 学習フェーズ

今回使用したモデルは pix2pix である.これは Phillip らによって考案された conditional GAN を用いて Image to Image のタスクを汎用的に行うアルゴリズムで G 側に U-net を用いている．この pix2pix はセマンティックセグメンテーションのタスクも行うことができる.この研究では線画とレイヤ分けした画像の変換タスクを学習させる.

イラストコミュニケーションサービス pixiv[4]に上がっている線画データから下記の条件に該当するデータを 110 枚収集した.

- 人である
- 背景を含んでいない
- 上半身のみ

収集したデータを手作業で塗り分け，左右反転と 10 度程度の回転を行い，1656 枚に水増ししたものを学習データとして使用した．塗り分けの際の領域は肌，髪，目，服，帽子の 5 つのパーツに分けた．図 2 に色とパーツの対応について示す.

図 2. 色とパーツの対応表

学習方法は batchnormalize を 4 と設定した.また,patchSize を 256px とした.

Generator 側の入力チャンネル，出力チャンネルともに 3 とした．入力は線画とし,教師ラベルを塗り分け画像とした.また画像サイズはすべて 256px まで縮小して使用した．損失関数はラベルと出力の平均絶対値誤差 x 100 + adversarial loss とした.

Discriminator 側は入力チャンネルを 3,出力チャンネルを 1 とした.また,損失関数は adversarial loss とした.

また,最適化手法は Adam とし,パラメータは $\alpha = 0.0002, \beta 1 = 0.5, \beta 2 = 0.999$とした.

学習時の損失関数の推移について図 3 に示す.

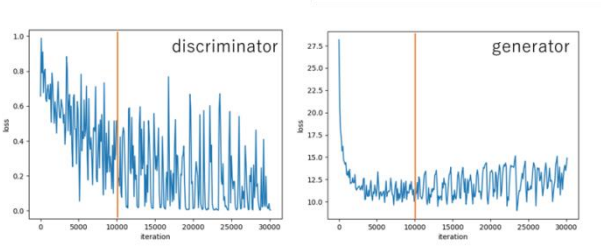

図 3. 学習結果

図 3 より，10000iterator 付近が損失関数が一番低いため，10000iterator の時のモデルを使用した．

3．2 レイヤ分け処理の自動化

pix2pix の出力（上段）に対して図 4 に示した方式で後処理を行った．

Step0:pix2pix に線画を入力し，セグメンテーションをした後，線画に合わせてリサイズを行う．

Step1:線画を Nagendraprasad-Wang-Gupta のアルゴリズム[5]を用いて細線化する．線の跡切れである端点を検出し，端点から周辺の線を参照して線を延長し，線の跡切れを軽減する．ここでの端点はある点に着目したときの 8 近傍のピクセルのうち 1 つが黒，そのほかが透明である点と定義した．延長は端点から隣接する 5px 分の線を抜き出し，端点に対して貼り付けている．

Step2:線画を延長した絵を python の画像処理ライブラリである pillow の fill 関数を用いてすべての領域を塗りつぶす．

Step3: pix2pix の出力結果と Step2 の出力結果を参照し，領域に含まれている色を調べ，一番多く含まれている色でその領域を塗りつぶす．これをすべての領域に対して行う．

Step4:線画の下には色がついていないため，線画の下と延長した線の領域は近隣のピクセルを参照して塗りつぶす．最後に色ごとに分解し，レイヤに分ける．

上記のような後処理を行い，領域の内部を完全に単色で塗りつぶした．

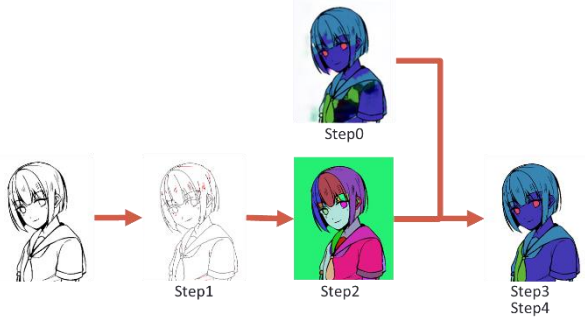

図 4. レイヤ分け処理の手順

4．性能評価

先ほど作成したモデルに対して学習データに含まれていない線画を入力し，出力結果と正解ラベルと比較して表 1 にまとめた．

表 1. 性能評価結果

項目	Sample1	Sample2	Sample3	Sample4
画像サイズ	447x564	429x612	487x550	487x550
領域数	43	47	37	42
正答率	86%	87%	86%	83%

5．考察

正答率が 80%を超える結果が得られた．肌と目の位置はどのイラストでも正確に認識することができた．髪も全体的には認識していたが，前髪や服につくほど長い髪などは誤認する傾向が見られた．服に縦線が多く含まれている場合ではその部分を髪と誤認したりするなどの現象が見られた．また，上に挙げた手などは帽子であると誤認される現象もみられた．これらの問題は学習データを増やすことやネットワークの構造を見直すことで改善すると思われる．

6．おわりに

pix2pix を利用してデジタルイラストのレイヤ分け作業を自動化するということについて有用性を持つことを示した．学習データを修正し，さらに精度を上げることが大きな課題である．また，どのパーツを分けるかはユーザによって異なると考えられる．そういった場合の時に，間違えた部分を修正するような UI を作成し，ユーザが対話方式で自由に修正が可能になるようなシステムを作成することが重要だと考えられる．また，近年絵を描くデバイスは多様化しており，Web アプリケーションとすることで様々なユーザのニーズにこたえることができると考えられる．

参考文献

[1] “Image-to-Image Translation with Conditional Adversarial Nets”, https://phillipi.github.io/pix2pix/.
[2] “PaintsChainer”, https://paintschainer.preferred.tech/index_ja.html.
[3] “CLIPStudio”, https://www.clip-studio.com/clip_site/.
[4] “pixiv”, http://www.pixiv.net/.
[5] P. S. P. Wang , Y. Y. Zhang,” A Fast and Flexible Thinning Algorithm”, IEEE Transactions on Computers, v.38 n.5, p.741-745, May 1989.