

선화 자동채색을 위한 효율적인 힌트입력 기법

김태훈^o, 이성진^{*}

^o경상대학교 기계항공정보융합공학부 항공우주및소프트웨어공학전공,

^{*}경상대학교 기계항공정보융합공학부 항공우주및소프트웨어공학전공

e-mail: xogns8121@gmail.com^o, insight@gnu.ac.kr^{*}

An Efficient Hint Input Method for Automatic Color Painting

Taehoon Kim^o, Seongjin Lee^{*}

^oDepartment of Aerospace and Software Engineering, Gyeongsang National University,

^{*}Department of Aerospace and Software Engineering, Gyeongsang National University

● 요약 ●

본 논문에서는 자동채색 머신러닝 모델을 모바일에서 사용하기 위해 효율적으로 힌트 입력하기 위한 방법을 제시한다. 기존의 자동채색 기법은 서버-클라이언트 채색 방식으로 인터넷에 연결되어 있어야만 채색이 된다는 문제가 있다. 이를 보완하여 네트워크 연결 없이 모바일에서 사용할 수 있는 머신러닝과 이를 사용하기 위해 입력으로 넣어야 하는 힌트 및 이미지를 효율적으로 나눌 수 있는 방법을 고안한다.

키워드: 모바일, 어플리케이션, 머신러닝, 힌트입력

I. Introduction

하드웨어의 발전으로 딥러닝에 관련된 많은 연구, 개발들이 활발하게 이루어지고 있다. 데스크톱 PC뿐만 아니라 이제는 딥러닝 모델의 경량화를 통해 모바일 기기에서도 딥러닝을 통한 학습 및 구동을 할 수 있게 되었다[1]. 경량화된 모델을 모바일에서 사용하기 위해서는 그에 따른 어플리케이션 플랫폼이 필요하고 딥러닝 모델을 사용하기 위한 최적화된 데이터가 필요하다.

대부분의 딥러닝을 사용한 선화 자동채색 기법(paintschainer[2])은 서버-클라이언트 방식으로 인터넷에 연결되어 있어야만 채색이 된다는 문제가 있다. 이를 보완하여 네트워크 연결 없이 모바일에서 사용할 수 있는 딥러닝 모델의 입력으로 필요한 힌트 이미지와 선화 이미지에 대해 bitmap pixel을 사용하여 효율적으로 구분할 수 있는 기법을 제시한다.

II. Preliminaries

1. Related works

1.1 선화 자동채색 방식

기존의 선화 자동채색 방식은 크게 두 가지로 나뉜다. 첫 번째로 Ci et al.[3]에서는 사용자 입력을 힌트로 사용자가 원하는 색으로 채색하는 방식이다. 이는 사용자가 원하는 색으로 채색을 할 수 있다는

장점이 있다. 이 방식을 사용하기 위해서는 사용자의 입력과 선화 이미지가 있어야 한다. 두 번째로 Zhang et al.[4]의 선화와 컬러 이미지를 사용하여 입력한 이미지 특징으로 채색하는 스타일 변환을 통한 채색방식이다. 사용자가 원하는 색이 아닌 입력한 이미지를 참고해 채색한다. 이는 첫 번째 방식과는 다르게 제한적으로 사용자가 원하는 색을 채색한다.

1.2 힌트입력 방식

힌트 입력 방식의 경우, 기존 연구의 힌트 입력 및 채색방식[3]은 Local Feature Net의 선화 특징추출 및 ResNextBlock을 통한 채색성능을 향상하는 방식으로 진행되었다. 여기에서 쓰인 선화 이미지와 힌트 이미지는 겹치는 부분이 있더라도 해당 선화 이미지와 힌트 이미지를 그대로 사용하여 힌트를 입력하고 자동채색을 진행한다.

본 연구에서는 더 나아가 학습 성능에 영향을 끼칠 수 있는(겹치는) 부분을 제거함으로써 더 정확하고 효율적으로 힌트를 입력하는 방법을 제시한다.

III. The Proposed Scheme

본 연구에서는 Ci et al.[3]에서 사용한 컬러 스트로크방식을 채택하여 사용자가 원하는 색으로 자동 채색할 시 효율적인 힌트를

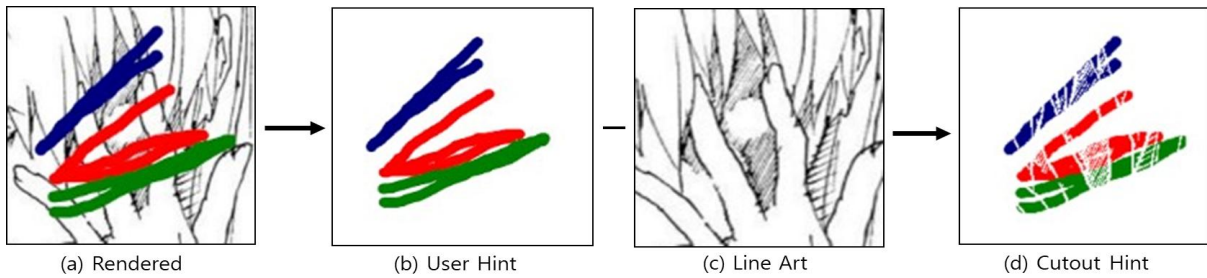


Fig. 1. 사용자 힌트로부터 스케치가 제거된 힌트 생성 과정

입력하는 기법을 제시한다. 이 방식을 사용할 때 입력값으로 선화 이미지와 그 위에 그려진 힌트 이미지를 사용하는데 이때 선화 이미지의 선과 힌트 이미지의 선이 겹치는 부분은 학습 성능에 영향을 줄 수 있다. 이 방식에 착안하여 Fig. 1. 과 같은 방식을 고안하였다.

Fig. 1.(c) 사용자가 선화 이미지를 불러올 때 불러온 Line Art(선화 이미지)를 가장 먼저 bitmap에 저장한다. 이 Line Art bitmap은 후에 입력값으로 사용될 수 있다. Fig. 1.(a), Fig. 1.(b) 사용자가 Line Art 위에 힌트를 준다면 이것이 User Hint로 bitmap에 따로 저장된다. 이것 또한 후에 입력값으로 사용될 수 있다. 이렇게 따로 bitmap에 저장되어서 각각을 변경, 수정, 관리하기에 효율적이다.

Fig. 1.(d) 본 연구에서는 여기서 그치지 않고 더 효율적인 방법을 고안한다. Line Art와 User Hint에는 선화와 힌트가 저장되어 있는데 User Hint에서 Line Art와 겹치는 부분을 pixel 단위로 확인하여 제거함으로써 User Hint와 겹치는 Line Art가 제거된 Cutout Hint를 얻을 수 있다.

마지막으로 앞서 말했던 Line Art bitmap 이미지와 Cutout Hint를 선화 자동채색을 위한 입력값으로 사용될 수 있다. 정교한 작업이 요구된다면 본 논문에서 고안한 방법이 더욱 효과적으로 작용할 수 있다.

Table 1. 은 고안한 방법에 대해 FID를 사용하여 평가하였다. FID값은 낮을수록 높은 품질을 나타낸다. 학습 성능에 영향을 끼칠 수 있는 부분을 제거함으로써 더 나은 FID결과를 얻을 수 있다는 것을 알 수 있다. 본 논문의 방법을 사용한다면 정교한 작업이 요구되는 학습에 효율적으로 힌트 이미지와 선화 이미지를 입력값으로 사용할 수 있다.

Table 1. 비교결과

(FID)	Min	1Q	Mean	Median	3Q	Max
Original	134.1	176.6	218.1	205.3	263.8	346.2
Proposed	132.9	169.6	216.2	198.4	260.2	348.5

IV. Conclusions

본 논문에서는 모바일에서 자동채색 모델을 사용하기 위해 입력으로 넣어야 하는 힌트 이미지 및 선화 이미지를 효율적으로 나누는 방법에 대해 고안하였다. User Hint와 Line Art를 각각 bitmap으로 관리하고 User Hint와 겹치는 Line Art를 제거함으로써 기존의 연구보다 더 정확하고 정교한 입력이 가능하다.

ACKNOWLEDGEMENT

이 논문은 2018년도 정부(교육부)의 재원으로 한국과학창의재단(2019년도 학부생 연구프로그램)의 지원을 받아 수행된 연구이며 또한, 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2019R1G1A1100455).

REFERENCES

- [1] Hoseung Kim, Gyeonghwan Hong, Dongkun Shin. "Acceleration and Performance Analysis of Deep Learning Models with Low-bit Precision on Mobile Devices." KCC2019, pp. 1418-1420, 2019
- [2] PaintsChainer, <https://paintschainer.preferred.tech>
- [3] Ci, Yuanzheng, Xinzhu Ma, Zhihui Wang, Haojie Li, and Zhongxuan Luo. "User-guided deep anime line art colorization with conditional adversarial networks." In 2018 ACM Multimedia Conference on Multimedia Conference, pp. 1536-1544. ACM, 2018.
- [4] Zhang, Lvmin, Yi Ji, Xin Lin, and Chunping Liu. "Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan." In 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), pp. 506-511. IEEE, 2017.