

硕 士 研 究 生 读 书 报 告



题目 可逆灰度图像

作者姓名 郭寒风

作者学号 21851152

指导教师 李启雷

学科专业 软件工程

所在学院 软件学院

提交日期 二○ 18年12月

Invertible Grayscale

A Report Submitted to

Zhejiang University

in partial fulfillment of the requirements for

the degree of

Master of Engineering

Major Subject: Software Engineering

Advisor: Li Qilei

By

Guo Hanfeng

Zhejiang University, P.R. China

2018

摘要

本文为阅读并分析《Invertible Grayscale》文献后而编写的读书报告。

文章中将介绍一种全新的方法，不仅能够将彩色图像转换为灰度图像，并且转换后的灰度图像也能恢复为原有的彩色图像。该转换方法的基本思路为：将原彩色图像的色彩信息编码至灰度图像中，此后便可使用该信息将灰度图像恢复为彩色图像。

系统主要包含两大卷积神经网络：编码神经网络，将原图像转换为灰度图像；解码神经网络，将灰度图像恢复为彩色图像。并使用三大损失函数来实现神经网络的准确性、高效性：可逆性损失（invertibility loss）、灰度准确性损失（grayscale conformity loss）以及量化损失（quantization loss）。

**关键词：颜色编码；可逆灰度图像；图像重构；神经网络；**

Abstract

This reading report is written based on the *Invertible Grayscale*.

The paper proposed an innovative method to convert a color image to grayscale, that can be later inverted back to its color version. The converted grayscale is called invertible grayscale. The key idea is to encode the original color information in the generated grayscale, so that the encoded color information can be restored during the later color restoration.

The system consists of an encoding neural network which converts any color image in RGB color space to a grayscale image, and a decoding neural network which converts any grayscale image to a color image in RGB space. The feasibility of the network is guaranteed by three loss function: invertibility loss, grayscale conformity loss and quantization loss.

**Key Words: color encoding; invertible grayscale; image reconstruction; neural network;**

目录

[摘要 I](#_Toc533450924)

[Abstract II](#_Toc533450925)

[1. 引言 1](#_Toc533450926)

[1.1 文章基本信息 1](#_Toc533450927)

[1.2 文章内容概述 1](#_Toc533450928)

[1.3 文章贡献 2](#_Toc533450929)

[2. 相关研究现状 3](#_Toc533450930)

[2.1 去色 3](#_Toc533450931)

[2.2 着色 3](#_Toc533450932)

[3. 系统详细设计 4](#_Toc533450933)

[3.1 系统概览 4](#_Toc533450934)

[3.2 损失函数 5](#_Toc533450935)

[3.3 数据训练 6](#_Toc533450936)

[4. 实验结果与分析 8](#_Toc533450937)

[4.1 定性评估 8](#_Toc533450938)

[4.2 定量评估 9](#_Toc533450939)

[4.3 性能测试 10](#_Toc533450940)

[5. 总结 11](#_Toc533450941)

[参考文献 12](#_Toc533450942)

# 1. 引言

## 1.1 文章基本信息

标题：Invertible Grayscale

作者：

Menghan Xia（SIAT, China and The Chinese University of Hong Kong）；

Xueting Liu（The Chinese University of Hong Kong）；

Tien-Tsin Wong（The Chinese University of Hong Kong and Shenzhen Key Laboratory of Virtual Reality and Human Interaction Technology, SIAT, China）

发表期刊：

ACM Trans. Graph. 37, 6, Article 246 (November 2018), 10 pages.

<https://doi.org/10.1145/3272127.3275080>

## 1.2 文章内容概述

为了满足多方面的需求——例如进行黑白打印、兼容旧版显示器配置等——我们常需要将彩色图像转换为灰度图像，但在此转换过程中，原图像的彩色信息便丢失了，无法再从灰度图像转换回原有的彩色图像。

文章中将介绍一种全新的方法，不仅能够将彩色图像转换为灰度图像，并且转换后的灰度图像也能恢复为原有的彩色图像。我们将使用这种方法转换而来的灰度图像称为“可逆灰度图像”。

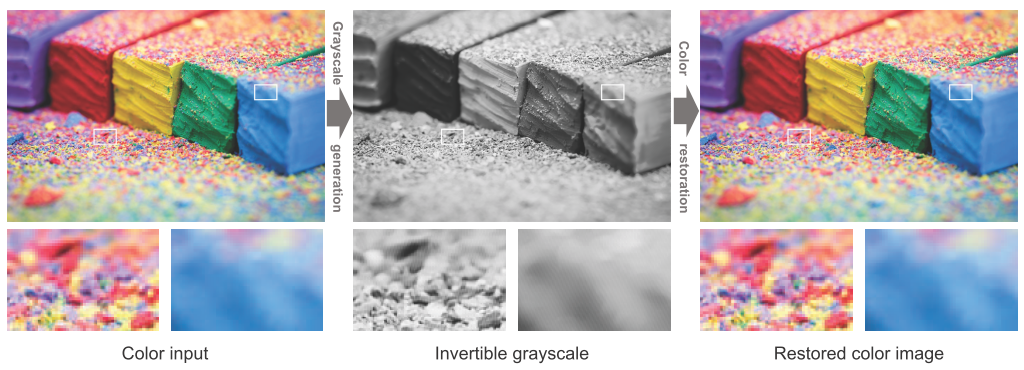


图1.1 将原彩色图像（左）转换为可逆灰度图像（中），再恢复原图（右）的过程。主要使用卷积神经网络将色彩信息以一种不明显的模式编码至生成的灰度图像中（第二排放大图）

该转换方法的基本思路为：将原彩色图像的色彩信息编码至灰度图像中，此后便可使用该信息将灰度图像恢复为彩色图像。根据色彩编码模式的复杂性，以及所需转换的原彩色图像的多样性，该方法使用卷积神经网络（Convolutional Neural Network，CNN）来完成色彩信息的编码及解码。

如图1.1所示，左侧为原有的彩色图像，中间为使用该方法生成的可逆灰度图像，右侧则是其所恢复而来的彩色图像；第二排的放大图中显示了将色彩信息编码至灰度图像后对其所产生的影响。

系统主要包含两大部分：

（1）编码神经网络：将原彩色图像转换为可逆灰度图像；

（2）解码神经网络：将转换而来的可逆灰度图像恢复为彩色图像；

而使得这两大神经网络能够进行高效、准确的转换，在于损失函数（loss function）的设计。系统中损失函数共包含：可逆性损失（invertibility loss）、灰度准确性损失（grayscale conformity loss）以及量化损失（quantization loss）。损失函数中每一项的设计都从不同角度对转换的效率、准确度进行一定量的提升。

## 1.3 文章贡献

（1）提出一种灰度图像的转换方法，不仅能够将彩色图像转换为灰度图像，并且转换后的灰度图像也能恢复为原有的彩色图像；

（2）结合CNN完成色彩信息编码\解码的工作，从而能够有效、快速地完成灰度图像的转换。

# 2. 相关研究现状

## 2.1 去色

去色（Decolorization）主要将彩色图像转换为灰度图像。对于生成灰度图像的方法而言，没有绝对统一的标准，不同的程序面对不同的需求也使用不同的标准来进行灰度图像的生成。最简单且常用的方式，既直接将原图像的亮度值当成灰度值从而生成灰度图像，但这种方式将经常造成颜色、对比度、结构化等信息的丢失，为了解决该问题，当今也已有多种方法被提出，这些方法主要可分为两大类：

局部去色（local decolorization）：根据局部色彩边缘线加强灰度图像中的对比度。现有的主要方案包括：

（1）Bala等人所提出的通过给亮度频道添加高频率色彩组件来保留色彩边缘信息；

（2）Neumann等人所提出的基于局部颜色和亮度对比度的连续渐变字段标记；

（3）Lu等人所提出的通过优化双峰分布找到相对视觉合适的颜色顺序；

全局去色（global decolorization）：通过全局颜色映射保留局部和非局部的对比度。现有的主要方案包括：

（1）Gooch等人所提出的基于使用线性模型的对比度参照图的映射规则优化；

（2）Kuk通过局部和全局对比度信息对Gooch所提出的方法进行扩展；

（3）Liu所提出的计算输入颜色和目标灰度图像之间的渐变相关度；

## 2.2 着色

着色（Colorization）主要将灰度图像进行着色，从而转换为彩色图像。在此过程中，由于色彩信息的丢失，常常需要用户手动提供某些图像参考才可完成图像的转换。现有的相关研究包含：

（1）Huang（2005）、Levin（2004）：着色过程可表达为基于低级特征进行用户引导颜色传播的优化；

（2）Luan（2007）、Qu（2006）、An（2008）、Xu（2009）：对增强低级特征可信度的进一步相关研究；

（3）Wang（2010）、Chang（2015）：色彩的全局统一性可通过加强全局色彩主题来达到；

（4）Charpiat（2008）、Chia（2011）、Irony（2005）、Liu（2008）、Welsh（2002）：用户所提供的参考图像可用于指引图像着色；

近年来，已有相关研究针对颜色和纹理、物体之间的相关性进行深度学习，使得能够高效、快速的为灰度图像进行着色[1]，但在此过程中需要有足够的纹理或对象来满足深度学习的要求。同时为了实现用户交互性，更多基于CNN的着色方法也被提出[2]。

# 3. 系统详细设计

## 3.1 系统概览

如图3.1所示，系统主要包含两大神经网络：

（1）编码神经网络 ：将任何RGB彩色图像转换为灰度图像。其中包含2项下卷积层（down-convolution block）、2项上卷积层（up-convolution block）、8项残差层（residual block）以及2项扁平卷积层（flat convolution layer），其中的上/下卷积结构用于特征提取中的接受域（receptive field）；

（2）解码神经网络 ：将灰度图像恢复为RGB彩色图像。其中包含8项残差层和1项扁平卷积层；



图3.1 系统概览

由于在将颜色信息编码至灰度图像时需要相对更强的neighborhood context，因此只有在编码神经网络中使用到了上/下卷积结构。

在训练过程中，对于训练数据集中的每一张彩色图片 ，使用编码神经网络 将其转换为灰度图像 ，并且使用解码神经网络 再将 恢复为彩色图像 :

在测试阶段，对于给定的任何图像输入，首先将其转为灰度图像，此时中所有像素值均为32位浮点数，因此需将其量化（quantize）成8位灰度图像，使得中所有像素值均为0到255的整数。恢复的彩色图则可通过来计算。

## 3.2 损失函数

系统总体损失函数为三项部分损失函数的权重和：

**（1） 可逆性损失（invertibility loss）**

对于给定的图像及将其灰度图像还原后所得的图像，可逆性损失用于确保和的相似性。该函数使用图像中每一像素的均方误差（mean square error，MSE）进行计算：

其中代表MSE的范数，则代表训练数据集中所有图像的平均运算符。

**（2） 灰度准确性损失（grayscale conformity loss）**

对于给定的图像及其所生成的灰度图像,该损失函数用于确保从视觉角度观察是由所生成。该损失函数同样为三部分的权重和，包含亮度损失（lightness loss）、对比度损失（contrast loss）及局部结构损失（local structure loss）：

其中与分别为对应权重值，在实验中设定以达到更好的效果。图3.2显示了该损失函数的重要性。

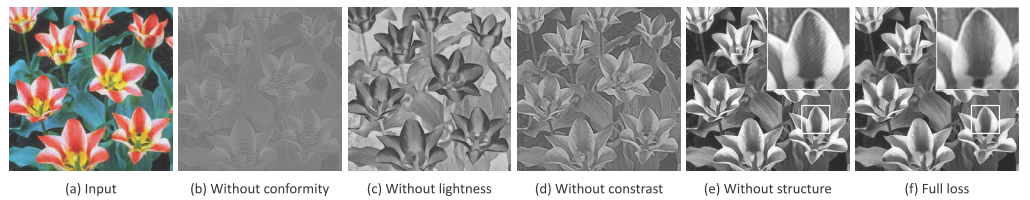


图3.2 灰度准确性损失函数的重要性

（a）亮度损失（lightness loss）

该损失函数用于确保所生成的灰度图像与原图的亮度在视觉上呈现一致性，既表明较亮的部分依然保留高亮度，较暗的部分依然保留较低的亮度。该函数计算公式如下：

其中代表MSE的范数，为按各元素计算的绝对值；表示图像的光照频道；表示2个相同行列数矩阵中按各元素比较的最大值计算；以及均为与相同行列数的矩阵，其元素均分别等于和0。实验中将的值设定为70。

（b）对比度损失（contrast loss）

该损失函数用于确保所生成的灰度图像与原图全局对比度的一致性。计算公式为：

其中表示在某图像中所提取出的第项层[3]；为图像的颜色频道。

（c）局部结构损失（local structure loss）

该损失函数用于确保所生成的灰度图像与原图局部结构的一致性，既表明在原图像中局部平滑的部分在所生成的灰度图像中同样平滑，从而降低所生成灰度图像中颜色信息编码所带来副作用的可见性。该函数计算公式如下：

其中为计算某图像局部变动平均值（mean of local variation）的函数。

**（3） 量化损失（quantization loss）**

该损失函数用于确保的所有像素值均为整数。计算公式如下：

其中表示对多个相同大小的矩阵进行各元素比较所得的最小值；为所有元素值均等于的矩阵，其大小与相同。通过该损失函数，可减少在量化过程中所产生的瑕疵，从而提高转换结果的质量，如图3.3所示。



图3.3 量化损失函数的重要性。三列分别为使用不同规模的量化损失函数所产生的恢复色彩图像及其像素颜色对比图

## 3.3 数据训练

在进行训练及测试时，所使用的数据均来自V0C2012（Visual Object Classes Challenge 2012）数据集[4]。该数据集中共包含17125张彩色图像，其中13758张用于数据训练，剩余图像则用于测试。尽管所使用的数据量较少，但实验结果表明该方法可以达到较高的准确率（SSIM平均0.9681，PSNR平均36.02dB）。在训练过程中所有图像均裁剪为256x256的大小，但在测试过程中图像的分辨率为任意值。

对于数据的训练，可从无到有直接训练（既1阶段训练），但由于量化损失函数具有部分独立性，对其训练的难度较大，且将占用较大的存储空间，因此我们提出使用2阶段训练的模式，具有加快训练速度、降低存储空间占用的优势。如图3.4所示，2阶段训练模式在速度上较1阶段训练模式更快。

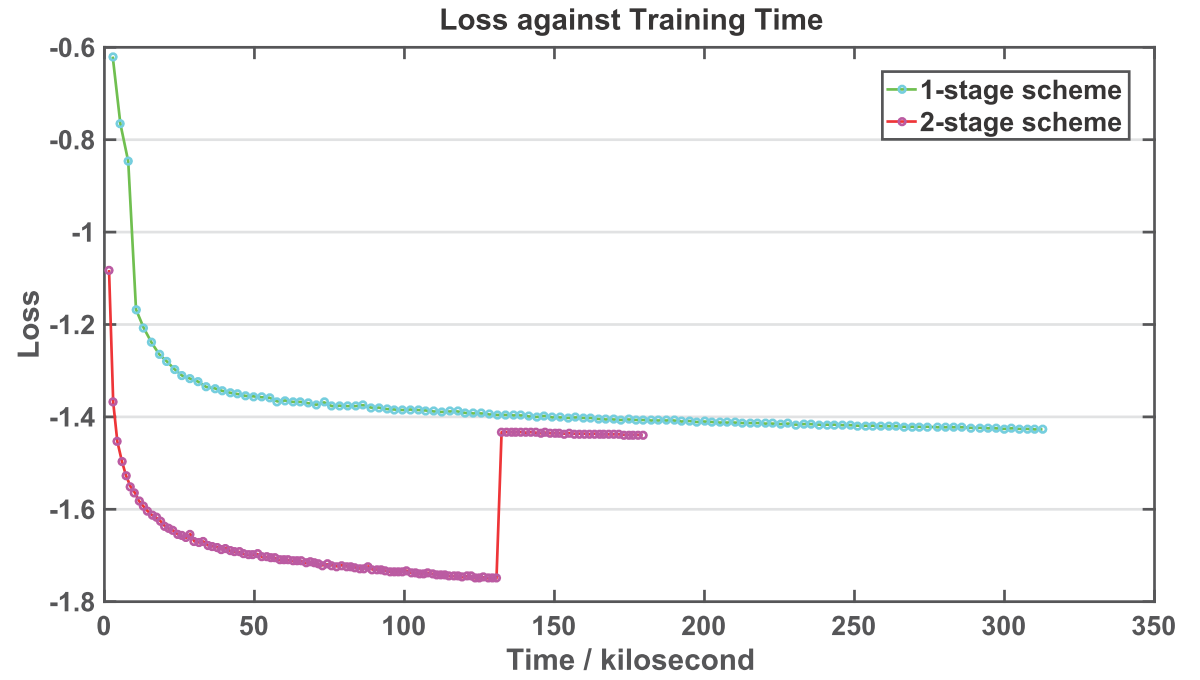


图3.4 不同训练模式的训练时间-损失对比曲线

在2阶段训练模式中，第1阶段仅考虑可逆性损失及灰度准确性损失，既将整体损失函数计算公式中的设为0，而第2阶段则将该值设为一相对较大的值，使得该阶段将重点放在量化损失中。量化损失函数计算公式中及的数值在2阶段中均不变。详细的参数值如表3.1所示。该训练模型使用ADAM solver[5]进行优化。

表3.1 不同阶段中的各参数值

|  |  |  |  |
| --- | --- | --- | --- |
| 阶段 |  |  | Epochs |
| 1 | 1.0 | 0.0 | 90 |
| 2 | 0.5 | 10.0 | 30 |

# 4. 实验结果与分析

## 4.1 定性评估

（1）与现有的着色方法进行比较

从与Zhang等人所提出的基于CNN的着色方法[6]比较结果来看，如图4.1所示，文章所提出的方法相较准确度更高。

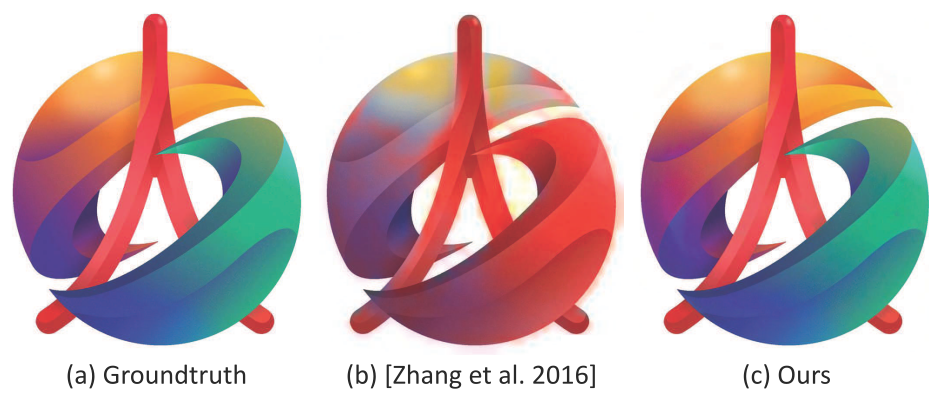


图4.1 与现有着色方法比较结果，原图(a)，着色方法生成图(b)，文章所提方法生成图(c)

进一步地与可交互的着色方法[7]进行比较，在此比较过程中，按照其文章中的意见，对测试数据集中的每一图像随机选择其500个像素点作为着色参照。

比较结果如图4.2所示，其中列a为原图像，列b为使用着色参照生成的灰度图，列c为使用Zhang的方法进行着色的结果，列d为原图与列c所示图的颜色对比图，列e为文章所提方法所生成的灰度图，列f为所还原的色彩图像，列g为原图与列f所示图的颜色对比图；各图右下角即为其对应的MAE值。



图4.2 与Zhang等人2017年所提方法比较的图像结果

（2）颜色信息编码副作用

将颜色信息编码至所生成的灰度图像中，将带来一些不明显的瑕疵，但只有在将图片放大到一定程度后才可发现。同时，这一副作用的大小与原图像的分辨率没有直接关系，如图4.3所示，分辨率越高的图片，所产生的瑕疵越不明显，需要进一步将图片放大才容易发现这些瑕疵。

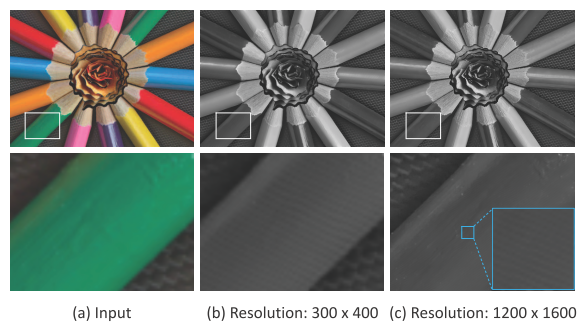


图4.3 颜色信息编码所产生的副作用与图像分辨率大小的关系

而若输入图像（原图像）已经为一灰度图像，由于不需要再进行颜色信息编码，因此将不会带来类似的副作用，如图4.4所示，行二中输入图像已为一灰度图像，再将其进行转换便不再产生副作用。

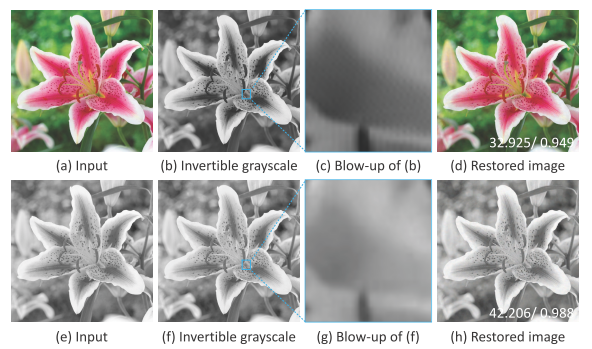


图4.4 若原图已经为灰度图像，则不会带来副作用

## 4.2 定量评估

使用V0C2012数据集中剩余的3367张图像进行定量评估。为了对比原图像以及所恢复图像的相似度，使用MAE（绝对平均误差）、PSNR以及SSIM[8]度量标准进行测量，同样也与Zhang等人在2017年所提出的方法进行比较。比较结果如表4.1所示，其中MAE的数值越低则代表相似度越高，PSNR及SSIM的数值则越高越好。

表4.1 与Zhang等人2017年所提方法比较的MAE\PSNR\SSIM数值结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 模式 | MAE | | PSNR | | SSIM | |
| 中值 | 标准差 | 中值 | 标准差 | 中值 | 标准差 |
| Zhang et al. 2017 | 6.682 | 1.584 | 29.72 | 2.183 | 0.9375 | 0.03970 |
| 1阶段 | 3.345 | 1.089 | 35.44 | 2.499 | 0.9641 | 0.02271 |
| 2阶段 | **3.083** | 1.055 | **36.02** | 2.666 | **0.9681** | 0.02007 |

与Zhang等人的方法所比，文章提出的方法具有更低的MAE值以及更高的PSNR、SSIM值，表明文章所提方法能够更加接近原图地将图像恢复。

## 4.3 性能测试

使用Python语言以及TensorFlow[9]对该模型进行实现，所有实验均在PC端进行，机器配置如下：

CPU：Intel Xeon E5-1630 v4 3.70GHz

GPU：GeForce GTX 980Ti

根据多类不同分辨率的图像以及是否开启GPU，分别对编码网络和解码网络的运行时间进行测试。在对不同分辨率图像进行测试时，进行100次测试，最后结果取其平均值。测试结果如表4.2所示，编码所花时间普遍比解码较多（主要因为编码网络中包含较多的卷积层），并且开启GPU时性能也有较大提升。

表4.2 性能测试结果（单位：秒）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 图像大小 | CPU | | CPU + GPU | |
| 编码 | 解码 | 编码 | 解码 |
| 256 x 256 | 0.939 | 0.825 | 0.034 | 0.029 |
| 512 x 512 | 3.717 | 3.284 | 0.113 | 0.086 |
| 1024 x 1024 | 15.117 | 13.133 | 0.384 | 0.321 |

# 5. 总结

文章介绍了一种全新的灰度图像转换方法，不仅能够将彩色图像转换为灰度图像，并且转换后的灰度图像也能恢复为原有的彩色图像。将采用这种方法所生成的灰度图称为“可逆灰度图像”。

通过使用CNN来对系统进行设计，其主要包含两大神经网络：

（1）编码神经网络：将原彩色图像转换为可逆灰度图像；

（2）解码神经网络：将转换而来的可逆灰度图像恢复为彩色图像；

神经网络的高效率工作得益于损失函数的设计。总体损失函数（loss function）为三部分损失函数的加权和：可逆性损失（invertibility loss）、灰度准确性损失（grayscale conformity loss）以及量化损失（quantization loss）。其中量化损失函数同样为三个子损失函数的权重和，包含亮度损失（lightness loss）、对比度损失（contrast loss）以及局部结构损失（local structure loss）。各损失函数中参数的设定对转换效果起重要的作用。

文章所介绍的方法只能够用于对静态图像进行转换，若要对视频进行灰度图像的转换，则可按逐帧转换的方式来进行，但该方法不能够保证时序一致性（Temporal Consistency）。同时，该方法所生成的灰度图像不支持图像操作（例如图像旋转、缩放，以及JPEG压缩等），因为对图像进行操作将可能破坏编码至灰度图像中的颜色信息。

进一步研究工作：

（1）达到时序一致性（Temporal Consistency），从而实现对视频的转换；

（2）实现所生成的灰度图像能够进行图像操作；

参考文献

[1] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. 2015. Deep colorization. In IEEE International Conference on Computer Vision (ICCV).

[2] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. 2017. Scribbler: Controlling deep image synthesis with sketch and color. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[3] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2012. The PAS-CAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. (2012).

[5] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1511.06349 (2014).

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learningfor Image Recognition. In IEEE Conference Computer Vision and Pattern Recognition(CVPR).

[7] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. 2017. Real-time user-guided image colorization with learned deep priors. ACM Transactions on Graphics (TOG) 36, 4 (2017), 119.

[8] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing (TIP) 13, 4 (2004), 600–612.

[9] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeff rey Dean, Matthieu Devin, Sanjay Ghemawat, Geoff rey Irving, Michael Isard, et al. 2016. TensorFlow: A system for large-scale machine learning. In OSDI, Vol. 16. 265–283.