DESIGN DOCUMENTATION

for

Supervised Colorisation of Grayscale Images Using Deep Neural Networks

Prepared by
CSU13202 12140804 Abhiraj P R
CSU13205 12140808 Akheel K M
CSU13223 12140824 Gokul A V
CSU13230 12140838 Manu Antony
CSU13248 12140851 Sarath M Joy

November 25, 2016

Contents

1 Introduction

Problem Statement

The proposed system convert the input which is given as grayscale image to a more realistic colourful image without user intervention. The Existing system need constant user guidance to identify the true colour of objects in image, which is largely time consuming and staggered.The technology has advanced significantly, but the existing system fails to implement it. Thus leads to the wastage of resources.

Proposed Solution

A perfect patch matching technique is used by employing an extremely large-scale reference database. The process of colorization can be solved by employing deep learning techniques. A joint bilateral filtering based post-processing step is proposed (for increased quality). The model is trained using Google's machine learning framework named Tensorflow.

1.1 Purpose

The main objective of this document is to present the requirements of conversion of grayscale images to colorful ones . Our project aims to completely fully-automate the process of conversion of a grayscale image to a coloured one with high quality by making use of deep learning. The document gives a detailed description of the design and implementation of the system and the interfaces of the system

1.2 Overview

Section 1 provides an introduction to the system. Section 2 provides the module wise system architecture of the coverting of grayscale images to colourful image. Section 3 deals with the data description and provides us with a diagrammatic description of the system. Section 4 gives relevant algorithms in pseudocode format.

2 System Architecture

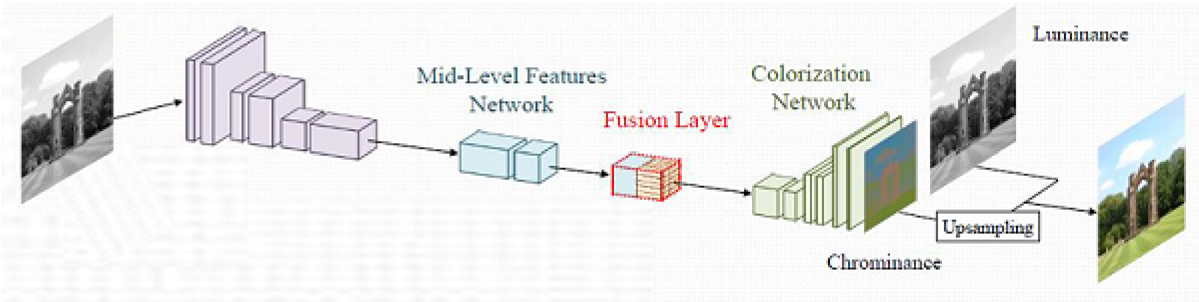

Figure 2.1: Model for Image Conversion

The various modules of the proposed system are as follows:

1. Low-level patch feature
2. Mid-level DAISY feature
3. High-level Semantic feature
4. Color-Space Discretization
5. Chrominance Refinement

2.1 Low-level patch feature

Intuitively, there exists too many pixels with same luminance but fairly different chrominance in a color image, thus it's far from being enough to use only the luminance value to represent a pixel. In practice, different pixels typically have different neighbors, using a patch centered at a pixel p tends to be more robust to distinguish pixel p from other pixels in a grayscale image. Let x_p denote the array containing the sequential grayscale values in a 7 x 7 patch center at p, x_p is used as the low-level feature descriptor in our framework. This feature performs better than traditional features like SIFT and DAISY at low-texture regions when used for image colorization.

2.2 Mid-level DAISY feature

DAISY is a fast local descriptor for dense matching. Unlike the low-level patch feature, DAISY can achieve a more accurate discriminative description of a local patch and thus can improve the colorization quality on complex scenarios. A DAISY descriptor is computed at a pixel location p in a grayscale image and is denote as x_p^M. The adoption of DAISY feature in our model leads to a more detailed and accurate colorization result on complex regions. However, DAISY feature is not suitable for matching low-texture regions/objects and thus will reduce the performance around these regions.

2.3 High-level Semantic feature

Patch and DAISY feature are low-level and mid-level features indicating the geometric structure of the neighbors of a pixel. The existing state-of-art methods typically employ such features to match pixels between the reference and target images. Recently, high-level properties of a image have demonstrated its importance and virtues in some fields (*e.g.* image enhancement, edge detection). Consider that the image colorization is typically a semantic-aware process, we extract a semantic feature at each pixel to express its category (*e.g.* sky, sea, animal) in our model.

We utilize the state-of-art scene parsing algorithm to annotate each pixel with its category label, and obtain a semantic map for the input image. The semantic map is not accurate around region boundaries. As a result, it is smoothed using an efficient edge-preserving filter with the guidance of the original gray scale image. An Ndimension probability vector will be computed at each pixel location, where N is the total number of object categories and each element is the probability that the current pixel belongs to the corresponding category. This probability vector is used as the high-level descriptor denoted as x^H.

2.4 Color-Space Discretization

The proposed method adopts the patch feature and DAISY feature, and we hope to use patch feature to describe lowtexture simple regions and DAISY to describe fine-structure regions. However, we simply concatenate the two features instead of digging out a better combination. This will result in potential artifacts especially around the low-texture objects (*e.g.,* sky, sea). This is because DAISY is vulnerable to these objects and presents a negative contribution.

The artifacts around low-texture regions can be significantly reduced using joint bilateral filtering technique . It was first introduced to remove image noise of a no-flash image with the help of a noise-free flash image. Our problem is similar, the chrominance values obtained from the trained neural network is noisy (and thus results in visible artifacts) while the target grayscale image is noise-free. As a result, to ensure artifact-free quality, we apply joint bilateral filtering to smooth/refine the chrominance values (computed by the trained neural network) with the target grayscale image as the guidance.

2.5 Chrominance Refinement

Each image in the set of training files is converted to the Lab (Luminance, a, b) color space. This is used over RGB because euclidean distance in Lab is more similar to how humans perceive color differences. e Luminance channel becomes the grayscale image from which features are extracted. Color information is represented via the two 8-bit values (a,b), allowing for 256^2 possible hues. We then reduce the color space via k-means to a manageable subset of colors – typically ranging between 8 and 32 – and store this color mapping for use by the final output step. Once the reduced colormap is selected, we quantize the training image or images, and randomly select a subset of training pixels – typically 5000 pixels from each 640x480 pixel image, or about 1.6%.

3 Data Description

3.1 Data Flow Diagram

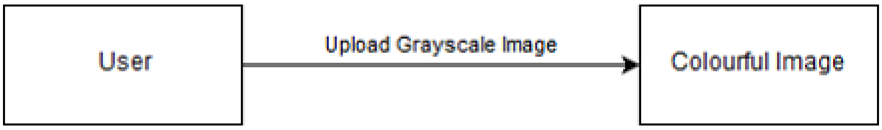

Figure 3.1: Level 0 DFD

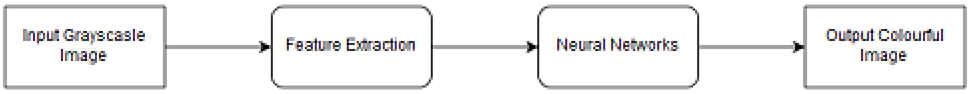

Figure 3.2: Level 1 DFD

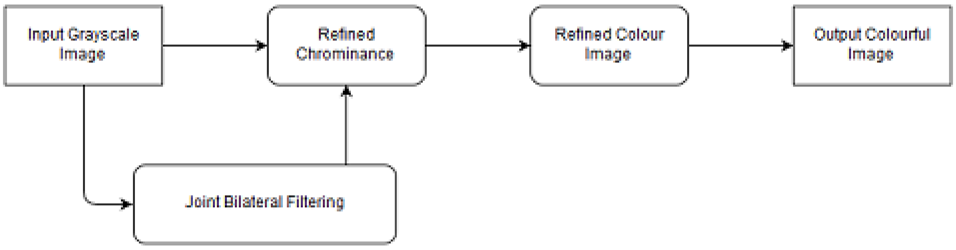

Figure 3.3: Level 2 DFD

3.2 Use case diagram

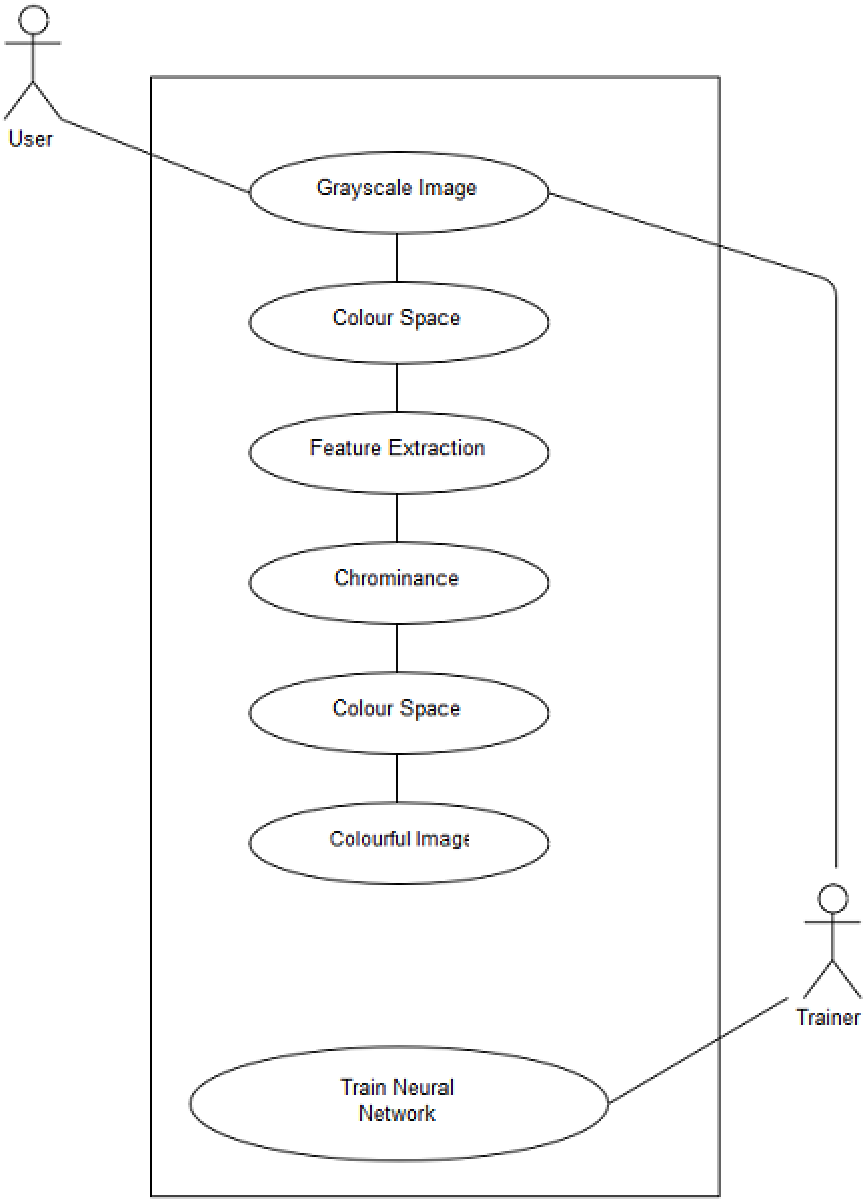

Figure 3.4: Use Case Diagram

3.3 Activity diagram

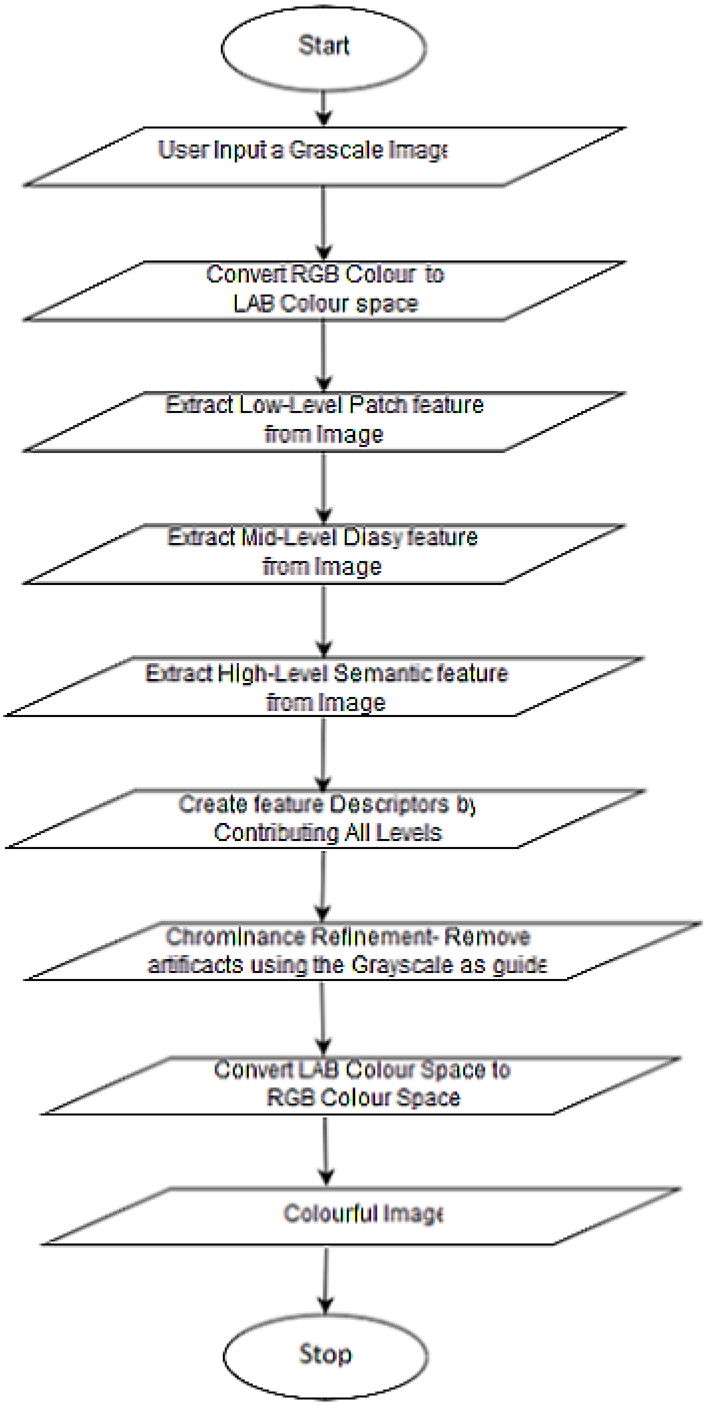

Figure 3.5: Activity Diagram

4 Algorithms

The proposed method has two major steps: (1) training a neural network using a large set of example reference images; (2) using the learned neural network to colorize a target grayscale image. These two steps are summarized in Algorithm 1 and 2, respectively.

4.1 Image Colourization - Training Step

Algorithm 1

Input: Pairs of reference images: $\Lambda = \{\vec{G}, \vec{C}\}$.
Output: A trained neural network.

---

1. Compute feature descriptors $\vec{x}$ at sampled pixels in $\vec{G}$ and the corresponding chrominance values $\vec{y}$ in $\vec{C}$;
2. Construct a deep neural network;
3. Train the deep neural network using the training set $\Psi = \{\vec{x}, \vec{y}\}$.

4.2 Image Colourization - Testing Step

Algorithm 2

Input: A target grayscale image I and the trained neural network.
Output: A corresponding color image: $\hat{I}$.

---

1. Extract a feature descriptor at each pixel location in I;
2. Send feature descriptors extracted from I to the trained neural network to obtain the corresponding chrominance values;
3. Refine the chrominance values to remove potential artifacts;
4. Combine the refined chrominance values and I to obtain the color image $\hat{I}$.

Bibliography

[1] Richard Zhang, Phillip Isola, Alexei A. Efros-'Colorful Image Colorization'-University of California, Berkeley

[2] Zezhou Cheng, Qingxiong Yang - 'Deep Colorization'-2015 IEEE International Conference on Computer Vision

[3] Austin Sousa Rasoul Kabirzadeh Patrick Blaes-Automatic Colorization of Grayscale Images

[4] Charpiat, G., Hofmann, M., Scholkopf, B.: Automatic image colorization via multimodal predictions. In: Computer VisionECCV 2008.

[5] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[6] ImageNet Classification with Deep Convolutional Neural Networks Alex Krizhevsky University of Toronto kriz@cs.utoronto.ca

[7] Colorization by Example R. Irony, D. Cohen-Or, D. Lischinski Tel-Aviv University The Hebrew University of Jerusalem Eurgraphics Symposium on Rendering.

[8] S. Belongie J. Malik J. Puzicha - Shape matching and object recognition using shape contexts- IEEE Transactions on Pattern Analysis and Machine Intelligence archive

[9] Object recognition from local scale-invariant features- David G. Low - ICCV '99 Proceedings of the International Conference on Computer Vision-Volume 2Volume 24 Issue 4, April 2002

[10] Aude Oliva and Antonio Torralba - Building the gist of a scene: the role of global image features in recognition.