



IPOL Journal - Image Processing On Line
www.ipol.im

IPOL MLBriefs 3

31/05 — 01/06

Day 3

Today

- Morning:
 - How to submit your articles to IPOL/MLBriefs ?
 - Examples of good articles
 - What now?
 - Round table of issues encountered
 - Authors start working in the workshop rooms
- Lunch: 13:00, 1E29
- Afternoon
 - Authors work on their demo in the workshop rooms
 - Coffee break at 15:30, delivered directly to the working spaces

What now?

- Finish and submit your works
 - Submission deadline: June 30th, 2023
 - Best paper awards will be given at the MLBriefs 3 presentation
- Announcing MLBriefs 3 presentation
 - Tuesday, December 12th, 2023
 - Here, at the ENS Paris-Saclay
 - More information to come soon
- MLBriefs 4
 - MLBriefs will now happen on a yearly basis
 - Next session: May or June 2024

What is an IPOL publication?

LOW RESOLUTION PDF: Images may show compression artifacts. A full resolution PDF is available at www.ipol.im.



Published in Image Processing On Line on 2022-08-05.
Submitted on 2022-04-29, accepted on 2022-08-09.
1026.2020-1232 © 2022 IPOL & the authors CC-BY-NC-SA.
This article is available online with supplementary materials,
software, datasets and online demo at
<https://doi.org/10.5201/ipol.2022.403>

DeOldify: A Review and Implementation of an Automatic Colorization Method

Antoine Salmona¹, Lucia Bouza¹, Julie Delon^{1,2}

¹ Université Paris Cité, CNRS, MAPS, France
² Institut Universitaire de France (IUF)
(antoine.salmona@paris.fr, lucia.bouza@paris.fr, julie.delon@paris.fr)

Communicated by Gregory Randall Demo edited by Lucia Bouza

Abstract

DeOldify is a recent automatic colorization method based on Convolutional Neural Networks which yields impressive results. The method was initially created by Jason Antic with the support of the Californian start-up Fast.ai and thus does not come from the academic research world. The goal of this paper is twofold. First we propose a rigorous mathematical presentation of the method along with a critical analysis of its different steps. Second, we provide an open-source implementation of a simplified but effective version of the approach, based on Pytorch and without dependence on the Fast.ai framework.

Source Code

The reviewed source code and documentation for this implementation are available from the web page of this article¹. Compilation and usage instruction are included in the README file of the archive.

Supplementary Material

A video colorized by the network is provided in the web page of the article.

Keywords: colorization; transfer learning; neural networks; CNN

¹<https://doi.org/10.5201/ipol.2022.403>

Antoine Salmona, Lucia Bouza, Julie Delon - DeOldify: A Review and Implementation of an Automatic Colorization Method, Image Processing On Line, 12 (2022), pp. 347-368. <https://doi.org/10.5201/ipol.2022.403>

```
#!/usr/bin/env python3
import sys
import argparse
import numpy as np
from analysis import detect_faces

args = argparse.ArgumentParser()
args.add_argument('--input', type=str, help='Input image path')
args.add_argument('--output', type=str, help='Output image path')
args.add_argument('--render_factor', type=float, help='Render factor')
args.add_argument('--saturation', type=float, help='Saturation')
args.add_argument('--model', type=str, help='Model name')
args.add_argument('--device', type=str, help='Device name')
args.add_argument('--verbose', type=bool, help='Verbose mode')
args.add_argument('--help', type=bool, help='Help message')

args = args.parse_args()

if args.help:
    print('DeOldify: A Review and Implementation of an Automatic Colorization Method')
    sys.exit(0)

if args.verbose:
    print('Input image: %s' % args.input)
    print('Output image: %s' % args.output)
    print('Render factor: %s' % args.render_factor)
    print('Saturation: %s' % args.saturation)
    print('Model: %s' % args.model)
    print('Device: %s' % args.device)

# Load the model
model = load_model(args.model, args.device)

# Detect faces
faces = detect_faces(args.input)

# Colorize the image
colorized_image = colorize_image(args.input, model, args.render_factor, args.saturation, faces)

# Save the colorized image
save_image(colorized_image, args.output)
```

Parameters Reset

render_factor

16

Max: 24
Min: 8

saturation_value

2

Max: 2.5
Min: 1.5

Run

Execution successful

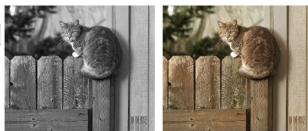
Results

Colorized image

Gray-scale image

Original image

☒ Compare



Colorized image

Gray-scale image

Original image

IPOL Journal · Image Processing On Line

HOME · ABOUT · ARTICLES · PREPRINTS · WORKSHOPS · NEWS · SEARCH

DeOldify: A Review and Implementation of an Automatic Colorization Method

Article Demo Archive

Please cite the reference article if you publish results obtained with this online demo.

69 public experiments since 2022-09-05

This archive is not moderated. In case you uploaded images that you don't want that appear in the archive, please contact the editor in charge. In case of copyright infringement or similar problems, please contact us to request the removal of some images. Some archived content may be deleted by the editorial board for size matters, inadequate content, user requests, or other reasons.

First Previous 1 2 3 4 5 6 7 Next Last Go to page: (1...7) Go

Experiment #525561.
2022-09-21 02:58:58 UTC

Parameters
render_factor: 16
saturation_value: 2

Reconstruct

Experiment #525562.
2022-09-21 02:59:34 UTC

Parameters
render_factor: 12
saturation_value: 1.5

Reconstruct

Experiment #525565.
2022-09-21 03:19:11 UTC

IPOL or MLBriefs tracks

MLBriefs track	IPOL track
Typically short paper (about 5 single-column pages)	More extensive paper
Relatively brief description of the method (Mainly qualitative) experiments to highlight the strengths and flaws of the method and the impact of the different parameters.	The method is fully described in pseudo-code, which corresponds to the code (same functions). If any learning, it is fully described, ideally retrained by the authors. Experiments remain, they can be more expansive.
Same method as the original.	Improvements on the method can be proposed
The code can be in any language.	Python, C, C++, Matlab/octave (possible exceptions)
The article and demo are peer-reviewed. The code is not peer-reviewed.	The article, demo and code are peer-reviewed. Guarantee of reproducibility: The code corresponds exactly to the pseudocode.
Ideal on methods that are not yours: easy to prepare without delving into the code	Ideal for your own works or methods you studied and understand very well: the pseudo-code is then easy to write

Examples of good papers

- Use them for inspiration, keep in mind there is not one single way and style to write a good paper
- MLBriefs track
 - [Image Forgery Detection via Forensic Similarity Graphs](#)
 - [A Brief Analysis of the SwinIR Image Super-Resolution](#)
 - [Disparity Estimation Networks for Aerial and High-Resolution Satellite Images: A Review](#)
 - [Monocular Depth Estimation: a Review of the 2022 State of the Art](#)
- IPOL track
 - [LSD: a Line Segment Detector](#)
 - [Image Forgeries Detection through Mosaic Analysis: the Intermediate Values Algorithm](#)
 - [ZERO: a Local JPEG Grid Origin Detector Based on the Number of DCT Zeros and its Applications in Image Forensics](#)
 - [Association Rules Discovery of Deviant Events in Multivariate Time Series: An Analysis and Implementation of the SAX-ARM Algorithm](#)

Paper template

<https://www.overleaf.com/read/kvwrsvwmmgnb>

Paper size

- No set min or max page limit, but remain reasonable
- Typical sizes: 5-10 pages for MLBriefs track, 20-30 pages for IPOL track
- No need to use dozens of examples to prove one single point
- Papers that are too long will take longer to review, and will be more easily rejected for this reason
- If you see your paper will be much longer than the limit, contact us first so we can discuss whether the length is appropriate for your work or where it could be cut down
- No hard limit, it is about the time it takes to read, not the bare length
 - A paper with a lot of figures and images can be longer than a paper with few figures

Method description

- IPOL track: Each function in the code should be described in pseudo-code in the article. Keep the functions names and separations matching between the article and the code
- IPOL track: Fully describe the training process (if applicable). Ideally, retrain it yourself to make sure the model was trained fairly (If reasonable in terms of resources)
- MLBriefs track: The above is not required (but always appreciated), the description can be simpler. Still give as much information as possible about the training (if this information is available)
- For the pseudo-code, use algorithm2e (for consistency between articles)
- An example configuration of the package with keywords and our custom namedinputs package are provided in the template: feel free to use them or to make your own

Experiments

- Mainly **qualitative** experiments
- Quantitative results on datasets are welcome too (especially if they prove something, like a robustness or weakness to false positives) but not at the core of the article
- The experiments should highlight the strengths, limitations of the method. If your method have parameters, it should show how to set them best and the impact of each parameter
- All the inputs in the qualitative experiments should be available as blobs in the demo
- For each experiment, write the name of the image in the blobs, as well as the parameters used (if applicable)

Figures and tables

- Use as many figures, tables and images (within figures) as needed (as long as each figure serves a point)
- Graphics should be kept at a high resolution (a low-resolution version will be automatically generated)
- One figure, one idea: If you present different ideas, they should be in different figures (unless the figure for both is really the same)
- Keep your figures large and legible: This is not a page-limited conference paper
- Save graphs, charts, etc as .eps files (or at worst as .png at a high enough resolution), **never** as .jpeg (lines and text will be blocky)
- Save photos as .png or .jpeg (without too much compression)
- Figures and tables preferably at the top of pages (`\begin{figure}[t]`)
- Each figure or table should have a caption which completely describes the figure and its interpretation (even if it repeats the main body) : **a reader will usually look at the figures before reading the paper, so he should understand the main ideas from those.**
- Tables and tabular: Use the booktabs package, usually with a `\toprule` at the top of the table, a `\midrule` to separate the headers/footers from the results, `\cmidrules` to make categories, a `\bottomrule` at the bottom. Avoid using `\hline` which does not render as well. **Never** put vertical line separators
- All images should be credited at the end of the paper (see the template)

Submission process

- MLBriefs track: easily on https://docs.google.com/forms/d/e/1FAIpQLSfMvyD55Api9CeKEHufL6qo1E3e7ikM_iKhHZsMRkhNv5DQ0A/viewform?usp=sf_link
- IPOL track:
 - IPOL-track submissions should be submitted via [OJS](#).
 - Register in the system (please don't use a gmail email address).
 - Once registered, create a new article submission.
 - Make sure your submission includes:
 - Your article in PDF.
 - A link to the github repository of your code, with the branch and rev to be reviewed, or an archive in the ZIP format containing your code without large files (the archive should not exceed 20 Mb).
 - If applicable, a link to the large files in the code, that were not included above. (e.g. model weights)
 - An URL to the original code (e.g. github link including a commit hash). If possible, include the license of the code.
- Submission deadline: **30th June 2023**
- Papers submitted by this deadline will be reviewed for presentation in the MLBriefs 3 presentation, and will be considered for best paper awards
- Late papers will still be reviewed, but may not be reviewed in time for the MLBriefs 3 presentation, and may not be considered for best papers awards

Feedback

You will shortly receive a feedback form.

Thank you for your time and participation.

Questions
Issues encountered