Reproducibility Companion Paper: Focusing on Persons: Colorizing Old Images Learning from Modern Historical Movies

Xin Jin, Ke Liu
Department of Cyber Security, Beijing Electronic Science and Technology Institute
Fengtai District, Beijing 100070, China

Dongqing Zou*
zoudongqing@sensetime.com
SenseTime Research and Tetras.AI Beijing
Beijing 100080, Beijing, China
Qing Yuan Research Institute, Shanghai Jiao Tong University
Shanghai 200240, China

Zhonglan Li, Heng Huang
Department of Cyber Security, Beijing Electronic Science and Technology Institute
Fengtai District, Beijing 100070, China

Vajira Thambawita
SimulaMet
Oslo, Norway

ABSTRACT

In this paper we reproduce experimental results presented in our earlier work titled "Focusing on Persons: Colorizing Old Images Learning from Modern Historical Movies" that was presented in the course of the 29th ACM International Conference on Multimedia. The paper aims at verifying the soundness of our prior results and helping others understand our software framework. We present artifacts that help reproduce results that were included in our earlier work. Specifically, this paper contains the technical details of the package, including dataset preparation, source code structure and experimental environment. Using the artifacts we show that our results are reproducible. We invite everyone to use our software framework going beyond reproducibility efforts.

CCS CONCEPTS

• **Computing methodologies → Image processing**.

KEYWORDS

Fine grained semantic parsing, Colorization, HistoryNet, MHMD, Reproducibility

ACM Reference Format:

Xin Jin, Ke Liu, Dongqing Zou, Zhonglan Li, Heng Huang, and Vajira Thambawita. 2022. Reproducibility Companion Paper: Focusing on Persons: Colorizing Old Images Learning from Modern Historical Movies. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Portugal.*

*Corresponding author

1 CONTRIBUTION SUMMARY

In order to achieve fine grained colorization of historical photos, we propose a new network architecture called HistoryNet and a new dataset called MHMD[4]. MHMD full name is Modern Historical Movies Dataset, which contains about 1.2 million images and 42 labels of eras, nationalities and garment types for automatic colorization from 147 historical movies or TV series made in modern time. Based on the proposed dataset, we design a colorization network. We use generative adversarial network as the main colorization network. In order to match the classification labels in MHMD, we have designed classifier subnetwork. The information of classifier subnetwork and classification labels jointly achieve HistoryNet colorization accuracy and classification labels more precise. We also propose semantic parsing subnetwork that can accurately obtain the semantic information of various parts of persons, which makes human semantic information colorization and image colorization boundary more accurate.

2 ARTIFACTS DESCRIPTION

The artifacts mainly include dataset, source codes and experimental environment settings. For reproducibility, we will introduce these parts in details one by one.

2.1 Dataset Preparation

MHMD obtains about 1.2 million images, including 1,147,517 images in training dataset and 100,000 images in testing dataset. The classification label sign.txt is also included in each dataset folder.

Besides, we also take Deeplab-v3[2] as the groundtruth of parsing subnetwork in HistoryNet. The datasets are available from Baidu Netdisk and Google Driver:

```
https://pan.baidu.com/s/19SIR9vMeYI9M0lhgZd5kGQ
(code: lejn)

https://drive.google.com/drive/folders/
1NDbPzooJS3Sv0FjYzKRW7LAkE5-kB-aT.
```

We have compressed them in .tar format. The detailed description of the dataset is as follows. The dataset file should be placed in the "DATASET" folder under the root directory, with at least 50GB disk space.

- **dataset_train**: The training dataset.
- **dataset_train_seg**: The parsing ground truth in the course of training.
- **dataset_second**: The testing dataset.
- **Quantitative_comparison**: Quantitative comparison experimental testing datasets, including original images and colorization effect images of six methods: HistoryNet, Iizuka et al.[3], Larsson et al.[5], Deoldify[1], ChromaGAN[7] and Su et al.[6].
- **Ablation_Experiments**: Ablation comparison experiments testing datasets.

Note there are two RAR packages which are dataset_train.part1.rar and dataset_train.part2.rar in dataset_train folder. They need to be decompressed at the same time.

2.2 Source Code Structure

Our source codes include 5 folders under the root directory. Github repository: https://github.com/BestiVictory/HistoryNet/.

- **Source_HistoryNet**: including HistoryNet.py, which trains HistoryNet network model.
- **Baseline**: including Baseline.py, which trains baseline model in ablation experiments.
- **Source_Parsing**: including HistoryNet_Parsing.py, which trains a model without classifier subnetwork in HistoryNet structure.
- **Source_Classifier**: including HistoryNet_Classifier.py, which trains a model without segmentation subnetwork in HistoryNet structure.
- **Util**: working to calculate PSNR, SSIM and LPIPS indices.
- **Visio**: We provide the Visio format of the comparison diagrams in our original paper [4].

Except for the main file, each folder (excluding Util and visio) also contains three python scripts: dataclass.py, colorization.py, and config.py.

- **dataclass.py**: setting the input and output of the network structure.
- **config.py**: configuration information during model training process.
- **colorization.py**: coloring the images with the pre-trained models.

The parameters in config.py are defined as follows:

- **BATCH_SIZE**: Training batch size.
- **NUM_EPOCHS**: Total training epochs.
- **PRETRAINED**: If you want to color the images with colorization.py, specify the colorization model name here.
- **OUT_DIR**: The storage path of the colorization effect of the testing dataset.
- **MODEL_DIR**: The storage path of the generated models during the training process.
- **LOG_DIR**: The storage path of the logs during the training process.
- **TRAIN_DATA**: The storage path of the training dataset.
- **TEST_DATA**: The storage path of the testing dataset.
- **TEST_DIR**: If you want to color the images with colorization.py, you need to place the images here.
- **RESULT_DIR**: The colorization results of the images under TEST_DIR.
- **GPU_ID**: Set the graphics card number to be used in training and testing.

The codes in util folder are used to calculate metrics.

- **compute_ssim_psnr.py**: working to calculate PSNR and SSIM indices.
- **lpips_2dirs.py**: working to calculate LPIPS indice.

2.3 Experimental Environment

Our source codes are tested in the following environment.

- **System and Hardware**: We recommend using Ubuntu 16.04.7 system with Intel Core i7-7800X CPU @ 3.50GHz and the graphics processing unit (GPU) NVIDIA TITAN X.
- **CUDA Toolkit and CuDNN**: Tested with CUDA==10.0.130 and CuDNN==7.6.
- **Version and Dependencies**: The tested python version is 3.6.0, and the dependencies of python packages are listed below. To easily install the dependencies, it is recommended to run "pip install -r requirements.txt" in the root folder of the package. We recommend to create a separate python virtual environment.

```
keras==2.2.4
numpy==1.15.4
opencv-python==4.1.0.25
tensorflow==1.14.0
tensorflow-gpu==1.14.0
h5py==2.10.0
```

We use Docker to encapsulate the local running environment in HistoryNet.tar.gz. Open HistoryNet.tar.gz to get our code runtime environment. To facilitate the activation by using the conda command, decompress the package to the envs directory in the Anaconda installation directory. Then execute the following scripts:

```
mkdir env_name
tar -xzf HistoryNet.tar.gz -C env_name
conda activate (full path)/env_name
```

You can obtain HistoryNet.tar.gz by Google Drive:

```
https://drive.google.com/drive/folders/
1C5ptVHaTDOVWWud_nXcpI5UAV-c-iM7W
```

3 TRAINING DETAILS

In this part, we will train HistoryNet in the original paper[4]. All the parameters have been set and the codes don't need to be modified. The training program is in the SOURCE_HistoryNet folder and execute the scripts:

```
cd  SOURCE_HistoryNet
python  HistoryNet.py
```

In the recommended operating system and graphics card environment, each epoch is approximately 28 hours.

We also provide pre-trained weight in "HistoryNet" folder in MODEL.tar:

```
https://drive.google.com/drive/folders/
1C5ptVHaTDOVWWud_nXcpI5UAV-c-iM7W
```

4 EVALUATION EXPERIMENTS

In this section, we will complete two evaluation experiments with three indices SSIM, PSNR and LPIPS. We provide the calculation codes in the Util folder, quantitative comparison experimental testing datasets, and ablation experimental testing datasets. We only need to change the paths of the source images and the colorization images in computing_PSNR_SSIM.py to calculate SSIM and PSNR. The method of computing SSIM and PSNR is as follows:

```
python ./Util/computing_PSNR_SSIM.py
```

In order to compute LPIPS, we need to use codes in Github respository: https://github.com/richzhang/PerceptualSimilarity. Clone this respository. We recommend using lpips_2dirs.py changed by us in Util folder to replace original one.

In order to speed up the process, we provide the running environment of LPIPS in LPIPS.tar.gz. You also install related dependencies by "pip install -r requirements.txt".

```
https://drive.google.com/drive/folders/
1C5ptVHaTDOVWWud_nXcpI5UAV-c-iM7W
```

We place the original images and the colored images by various colorization methods into folder imgs/ex_dir0 and imgs/ex_dir1 respectively. Execute the following codes and you will get the final result.

```
cd PerceptualSimilarity
python lpips_2dirs.py -d0 imgs/ex_dir0 -d1 imgs/ex_dir1
```

4.1 Evaluation 1: Quantitative Comparison

We provide quantitative comparison testing datasets with Iizuka et al.[3], Larsson et al.[5], Deoldify[1], ChromaGAN[7] and Su et al.[6] in "Quantitative_comparison" folder in the section of "Dataset Preparation". You can directly compute SSIM, PSNR and LPIPS of HistoryNet and other colorization methods to compare. The comparison results are shown in Table 1.

Table 1: Quantitative comparison of experimental

Method	LPIPS	PSNR	SSIM
Iizuka et al. [3]	0.134	25.779	0.956
Larsson et al. [5]	0.147	24.506	0.946
Deoldify [1]	0.127	26.321	0.957
ChromaGAN [7]	0.076	29.748	0.955
Su et al. [6]	0.150	24.660	0.936
HistoryNet	**0.074**	**29.958**	**0.963**

4.2 Evaluation 2: Ablation Experiments

In this part, we also provide ablation experiments codes in Baseline, SOURCE_Parsing, SOURCE_Classifier folders. SOURCE_Parsing only includes segmentation submodule and SOURCE_Classifier only includes classifier subnetwork. We excute the following scripts:

```
cd Baseline/SOURCE_Parsing/SOURCE_Classifier
python Baseline.py/HistoryNet_Parsing.py/
       HistoryNet_Classifier.py
```

We can get the corresponding models. We can use the testing datasets provided to calculate SSIM, PSNR and LPIPS to contrast. The comparison results are shown in Table 2.

Table 2: Ablation experiments

Method	LPIPS	PSNR	SSIM
Baseline	0.0652	31.2108	0.9587
Baseline+Classifier	0.0665	31.1887	0.9607
Baseline+Parsing	0.0662	31.4211	0.9615
HistoryNet	**0.0632**	**31.6704**	**0.9618**

Training these models from scratch is laborious and time consuming. Therefore, we also provide pre-trained weights in MODEL.tar in Google Drive:

```
https://drive.google.com/drive/folders/
1C5ptVHaTDOVWWud_nXcpI5UAV-c-iM7W
```

Decompress MODEL.tar into a directory MODEL in the root directory (the same directory as "SOURCE_HistoryNet"). For example, you can use pre-trained weight in "./MODEL/HistoryNet_Parsing", and only need to change "TEST_NAME" to "HistoryNet_Parsing" in config.py in "./SOURCE_Parsing". Then you use "python colorization.py" in the same folder to obtain colorization images. We also provide the ablation experiments testing dataset in "Ablation_experiments" folder in the section of "Dataset Preparation". You can experiment directly with these images.

5 REVIEWING PROCESS

First, the experimental setup to train and run HistoryNet was initiated as described in the paper and the GitHub repository. Then, all the steps, from downloading data to training the model, were followed one by one. When faults and miss communications were found, authors were informed to correct them and continuously monitored until problems were solved.

Overall, the paper gives clear instructions and links to the data and the source codes to reproduce the results in the paper. However, in the review process, a few minor problems were found, such as miss-matches between folder naming, missing steps, and difficulties in the training process to reproduce the results from scratch. Then, new recommendations were applied, and modified the code and data repositories and the content of the paper. In this review process, pre-trained weights were introduced into the dataset and the source code.

In conclusion, review process were performed by reviewers and authors together. The final version of the paper has all the information and data to reproduce the results in the original paper. Therefore, other researchers can use this manuscript as a guideline to reproduce results and continue there research on top of this solution.

6 CONCLUSION

In this paper, we provide the details of the artifacts of the paper "Focusing on Persons: Colorizing Old Images Learning from Modern Historical Movies" for replication. The artifacts contain the dataset and the source codes for experiments in the paper. Taking advantage of the source codes, the experiments can be operated and customized.

REFERENCES

[1] Jason Antic. 2019. DeOldify. https://github.com/jantic/DeOldify.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* 40, 4 (2017), 834–848.

[3] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2016. Let there be color! Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (ToG)* 35, 4 (2016), 1–11.

[4] Xin Jin, Zhonglan Li, Ke Liu, Dongqing Zou, Xiaodong Li, Xingfan Zhu, Ziyin Zhou, Qilong Sun, and Qingyu Liu. 2021. Focusing on Persons: Colorizing Old Images Learning from Modern Historical Movies. In *Proceedings of the 29th ACM International Conference on Multimedia*. 1176–1184.

[5] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. 2016. Learning representations for automatic colorization. In *European Conference on Computer Vision*. Springer, 577–593.

[6] Jheng-Wei Su, Hung-Kuo Chu, and Jia-Bin Huang. 2020. Instance-aware Image Colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7968–7977.

[7] Patricia Vitoria, Lara Raad, and Coloma Ballester. 2020. ChromaGAN: Adversarial Picture Colorization with Semantic Class Distribution. In *The IEEE Winter Conference on Applications of Computer Vision*. 2445–2454.