10. Image Inpainting tutorial: how to digitally restore damaged images

Author(s)

Name	Institution	Mail Address	Social Contacts
Alessandro Bombini	Cultural Heritage Network (CHNet), INFN, Firenze on behalf of European Science Cloud (EOSC) - Pillar	bombini@fi.infn.it	Skype: ; Linkedin: https://it.linkedin.com/in/alessandro-bombini-7929a2133; Twitter: ; Hangouts:
			Skype: ; Linkedin: ; Twitter: ; Hangouts:

How to Obtain Support

Mail	bombini@fi.infn.it
Social	Skype: ; Linkedin: ; Twitter: ; Hangouts:
Jira	

General Information

ML/DL Technologies	CNN U-Net
Science Fields	Applied Physics
Difficulty	Entry level
Language	Eng
Type	fully annoted, runnable

Software and Tools

Programming Language	Python
ML Toolset	Keras
Additional libraries	Sci-kit image, PIL, OpenCV, matplotlib
Suggested Environments	

Needed datasets

Data Creator	Toronto University
Data Type	RGB Images
Data Size	175 MB
Data Source	https://www.cs.toronto.edu/~kriz/cifar.html

Short Description of the Use Case

Inpainting is a conservation process where damaged, deteriorating, or missing parts of an artwork are filled in to present a complete image. This process can be applied to both physical and digital art mediums such as oil or acrylic paintings, chemical photographic prints, 3-dimensional sculptures, or digital images and video.

Here we show a dummy example of Image Inpainting by means of Convolutional Neural Network.

As a first example, we apply a digital damage to a set of 32x32 RGB images; the damage consists in adding a random number of black segments on top of the RGB image (mimicking cuts on the surface).

After that, we train a 2D Convolutional Neural Network of the U-Net type. U-Nets are 2D CNN of the Encoder/Decoder type which are completely symmetrical, i.e. if it has N layers, the k and the Nk layer have the same dimensions, and each layer of the encoding part is forwardly connected either with the subsequent encoding layer and with the symmetrical decoding layer; this means that it can be graphically arranged in a U-shape, hence the name.

We will build a network this form:

0. Input Layer

1. Conv2D x2 + MaxPool2D
2. Conv2D x2 + MaxPool2D
3. Conv2D x2 + MaxPool2D
4. Conv2D x2 + MaxPool2D
5. Conv2D x2 + Upsampling + Concatenate (4,5)
6. Conv2D x2 + Upsampling + Concatenate (3,6)
7. Conv2D x2 + Upsampling + Concatenate (2,7)
8. Conv2D x2 + Upsampling + Concatenate (1,8)
9. OutPut Layer to RGB

For the training of the 32x32 case we will use the CIFAR dataset (https://www.cs.toronto.edu/~kriz/cifar.html). It consists in 50.000 RGB 32 x 32 images.

We offer also the same model trained over 32x32 images and over 192x192 images, and we use transfer learning to apply the trained NN to different damages.

Since INFN-CHNet (Cultural Heritage Network) is the network of the National Institute of Nuclear Physics (INFN) devoted to cultural heritage, we apply the NN to paintings.

This is a dummy case of Digital Restoration, showing a glimpse of how NN can be applied to Physical Imaging of Cultural Heritages.

How to execute it

The example is available in the GitLab repository:

https://gitlab.com/alessandro.bombini.fi/image-inpainting-tutorial/

You need Jupyter notebook as well as all the Python packages necessary to run it.

Annotated Description

The repository is arranged as follows:

Folders:

- **pics/**: contains the useful pics to be shown in the notebooks to describe it
- **images/numbered_images/**: the images on which we apply the trained Neural Network
- **Model_data/**: contains the .h5, .json files of the trained models, as well as pics of the history of the training.

Notebooks:

1. **Image_Inpaiting_Tutorial.ipynb:** annotated notebook containing the construction and the training of the model

2. **open_model.ipynb**: annoted notebook where the model trained over damaged 32x32 images is applied;
3. **open_model_192x192.ipynb**: annoted notebook where the model trained over damaged 192x192 images is applied;

Other:

1. **Tutorial Image Inpainting.pdf:** pdf version of the slides used while presenting this notebook;
2. **Tutorial Image Inpainting.pptx:** powerpoint version of the slides used while presenting this notebook;

References

1. Cifar dataset: https://www.cs.toronto.edu/~kriz/cifar.html
2. Original UNet paper: https://arxiv.org/abs/1505.04597
3. Original ADAM optimazer paper: https://arxiv.org/abs/1412.6980

Attachments

https://gitlab.com/alessandro.bombini.fi/image-inpainting-tutorial/