AARHUS UNIVERSITY

Stylized paintings and classification (self-assigned)

Emil Trenckner Jessen

20th May 2021

Contents

1 Introduction

1.1 GitHub repository

Access to the GitHub repository can be found via the link here:

https://github.com/emiltj/cds-visual-exam

The repository contains the individual assignments (including this assignment) a long with READMEs. The READMEs specify cloning the repository, setup (virtual environment installation and data collection). You may also read the information here.

1.2 Getting started

For running my script I'd recommend following the below steps in your bash-terminal (notice the bash scripts are different, depending on your OS). This functions as a setup of the virtual environment, as well as an execution of a bash script that downloads all the data to the data folders respective to the assignments.

Cloning repository and creating virtual environment

Listing 1: bash terminal - MAC/LINUX/WORKER02

```
git clone https://github.com/emiltj/cds-visual-exam.git
cd cds-visual-exam
bash ./create_vis_venv.sh
```

Listing 2: bash terminal - WINDOWS

```
git clone https://github.com/emiltj/cds-visual-exam.git
cd cds-visual-exam
bash ./create_vis_venv_win.sh
```

Retrieving the data

The data is not contained within this repository, considering the sheer size of the data. Using the provided bash script data_download.sh that I have created, the data will be downloaded from a Google Drive folder and automatically placed within the respective assignment directories.

Listing 3: bash terminal

```
bash data_download.sh
```

After cloning the repo, creating the virutal environment and retrieving the data you should be ready to go. Move to the assignment folder and read the README for further instructions (or read a long here).

2 Stylized paintings and classification (self-assigned)

2.1 Assignment description

This self-assigned assignment has two main questions it seeks to investigate. The investigation is not hypothesis driven nor research oriented. Rather, it is meant to be a fun and atypical way of exploring some methods used in visual analytics.

Question 1 - Generating stylized paintings

Is it possible to use the method of style transfer to stylize paintings of one artist with the style of another? This assignment seeks to investigate the possibility of not just transferring style from a painting to an actual image as is so often done, but to explore the transferral of the style of one painting to another painting which already is painted in a particular style. Using paintings from the artists Cezanne and Monet from the impressionist paintings dataset, create a script which generates new stylized images. The stylized images should contain the contents of Cezanne with styles of Monet and vice versa.

- Save the stylized images in unique folders with names specifying the which artist the content and style came from.
- Save examples that clearly show the before and after result; save an image which includes content image, style image and the resulting stylized image.

Question 2 - Classification of stylized images

When CNN's classify paintings from artists, do they rely on the style of a given image? Or rather more on the content of the image? This part of the assignment seeks to train and test a classifier on the original paintings, and subsequently use the same trained model to classify the stylized paintings. Are Monet paintings with Cezanne style classified as Monet due to their content? Or rather classified as Cezanne due to their style? In other words, investigate the importance of content vs. style when classifying paintings.

- Use a pre-trained CNN classifier to distinguish between Cezanne paintings and Monet paintings.
- Use the same model to classify between the newly generated stylized images.
- Discuss the findings and consider whether the results tell us something general about either the classifier **or** about the stylization process

2.2 Methods

Generating stylized paintings

This script utilizes neural style transfer. Style transfer refers to the act of minimizing the distance between two sets of embedded images. One image (the style image) is embedded using the first few layers of a neural network. The other image (the content image) is embedded using the same neural network, but using the first many layers of the network. Likewise, the embedded image from this layer is also extracted. Using these two embeddings, style transfer then seeks to synthesize the two with regards to a loss function that minimizes the distance between the content embedding and the style embedding. Sometimes - as in the specific model use here - a Lagrange multiplier is introduced, which determines the weight of importance of the style embedding (level of stylization).

The original paintings were loaded and their order shuffled. They were then paired by their new shuffled index and each pair was then use to generate two new stylized paintings using the "magenta/arbitrary-image-stylization-v1-256" style transfer procedure. Content from painting A and style from painting B (and vice versa) were synthesized into stylized images. The stylized images were then preprocessed and a few examples of content + style + stylized images were saved to the examples directory in the directory "out". The entire script is designed to generalize to any other image corpora of any size. The script is furthermore designed so that you may specify both inpaths and outpaths.

Classifying paintings and stylized paintings

Both the original paintings and the stylized paintings were loaded using a self-define function and labels were automatically assigned. The paintings were then resized and formatted, as to match the input expectations of the subsequently defined pre-trained CNN model (MobileNetV2). The model was then trained to classify the artists of the original paintings, and later tested on an unseen subset of the same data. The same model was then set to classify the stylized images. The training history as well as the classification reports for both test sets were printed to the terminal and saved to the out directory.

On a more general level (this applies to all assignments)

I have tried to as accessible and user-friendly as possible. This has been attempted by the use of:

- Smaller functions. These are intended to solve the sub-tasks of the assignment. This is meant to improve readability of the script, as well as simplifying the use of the script.
- Information prints. Information is printed to the terminal to allow the user to know what is being processed in the background.
- Argparsing. Arguments that let the user determine the behaviour and paths of the script.

2.3 Results and discussion

Generating stylized paintings

Figure 1: Example of the stylization process. Left: Monet content (top) and Cezanne style (middle) synthesization of the two (bottom). Right: Cezanne content (top) and Monet style (middle) synthesization of the two (bottom).

When looking at the above two images and (the rest of the examples), it does indeed seem possible to transfer style from a painting, to another painting. However, from a brief glance at the 20 examples it seems that when using portraits of people as style image, the process seems to generate paintings that are hard to interpret (e.g. image 12).

Alternatively to the random pairings of style/content images, one could have considered extracting the styles of all images of one artist and then subsequently found the weights resulting in the least information loss across all these images. This way we would have the general style of an artist to use as the style image when stylizing images. However, a caveat to this method would be the artist we have here, tend to not have the same style of painting over time. Moreover, the fact that the style embedding inevitably also extracts bits of content, would mean that one would have to acquire a very large number of paintings from an artist to model out the noise.

	monet	cezanne	accuracy	macro avg	weighted avg
precision	0.92	0.97	0.94	0.95	0.95
recall	0.97	0.92	0.94	0.94	0.94
f1-score	0.95	0.94	0.94	0.94	0.94
support	101.0	98.0	0.94	199.0	199.0

Table 1: Original paintings classification report

Classification of stylized images

From inspecting the classification report when predicting the stylized images, we can see that the classifier predicted roughly 25% of the Monet paintings with Cezanne style, as Monet (and vice versa).

	monet	cezanne	accuracy	macro avg	weighted avg
precision	0.21	0.28	0.25	0.24	0.24
recall	0.18	0.32	0.25	0.25	0.25
f1-score	0.19	0.3	0.25	0.24	0.24
support	497.0	497.0	0.25	994.0	994.0

Table 2: Stylized paintings classification report. **NOTE:** The Monet paintings with style from Cezanne had - for this classification - their True label set as "Monet".

Can we use this report for shedding light on the question of "*Are Monet paintings with Cezanne style classified as Monet due to their content? Or rather classified as Cezanne due to their style? Discuss the findings and consider whether the results tell us something general about either the classifier or about the stylization process.*"

There are two ways of interpreting these results. The first being that the model bases its predictions not so much on the content of the painting, but rather on the style. However, a confound comes in the way of going to this conclusion. Another way of interpreting the results takes the way in which the stylized paintings were generated into account. The classification results are likely the product of the fact that the stylized paintings that were synthesized were not 50% content and 50% style in the first place. Neural style transfer in its most basic form seeks to minimize the distance between the content of one image and the style of another - resulting in roughly half of each. However, the "magenta/arbitrary-image-stylization-v1-256" model uses a more sophisticated approach.

In summary, the classification report of the stylized paintings cannot lead to any direct conclusions. If instead a simple neural transfer model that stylized images with equal weighting to both content and style had been used, then we would have been able to make inferences about the relative importance that content and style had on the classifier.

2.4 Usage

Make sure to follow the instructions in the README.md located at the parent level of the repository, for the required installation of the virtual environment as well as the data download.

Subsequently, use the following code (when within the cds-visual-exam folder):

Listing 4: bash terminal

```
cd assignment_6
source ../cv101/bin/activate # If not already activated
python generate_stylized.py
python cnn-stylized.py
```

2.4.1 Optional arguments

- "-f" "–filepath", type = str, default = os.path.join("data", "*.jpg"), required = False, help= "str - path to image corpus")
- "-t" "–targetpath", type = str, default = os.path.join("data", "image_0002.jpg"), required = False, help = "str - path to target file from which to calculate distance to the other images")