TECHNISCHE UNIVERSITÄT BERLIN
Fakultät IV – Elektrotechnik und Informatik
Fachgebiet Internet Network Architectures

IP Multimedia Lab WS 2018/19
Homework 1
Due: 20.11.2018, 11.55 PM

You can upload your submission to the ISIS online course (`https://isis.tu-berlin.de/mod/folder/view.php?id=628432`). Please submit your results as an archive, including your code and any scripts that you wrote to solve the assignments. Prepare a PDF file, in which you write down your findings and answers to the questions. For your submission, please consider the following points:

- Please name your archive *worksheet_⟨ws_number⟩_group_⟨group_number⟩*
- Add a cover page to the PDF including your group number and all group members
- Describe all relevant steps you performed
- Make your plots easily understandable, this is supported by
  - labelling your x-axis and y-axis
  - using informative captions
  - setting an appropriate font size
  - adding legends when needed
- Write down which libraries/tools you used
- Describe your results and denote to which (sub-)task they belong
- All code must be properly documented using inline comments
- All files that belong to a specific question must have the question/subquestion in their filenames

Question 1: (20 Points) *Traffic behaviour: Download versus video streaming* For this task, the packet analyzer Wireshark is required. With this tool, captured packets can be filtered based on different characteristics, like source or destination IP-address/port, used protocol, or packet type (e.g. SYN, ACK). For this task, we advise you to apply Wireshark's I/O Graph for data visualization. It is also sufficient to plot your data with this Wireshark built-in function, i.e. you do not need to plot anything with Python or R.

Further information on Wireshark can be found here: `www.wireshark.org`

(a) *Traffic behaviour of a file download.*
Capture the traffic between your client and a remote server when downloading a file from the web. Test files can, upon others, be found here: `www.thinkbroadband.com/download` and here: `www.speedtestx.de`. Please choose a file of at least 1GB. Visualize the throughput over time and shortly describe the results.

(b) *Traffic behaviour of a video stream.*
Capture the traffic when you stream a video from a Video-on-Demand (VoD) platform. As most of the platforms use HTTPS, and encrypted packets limit your possibilities to analyze them, we propose to request a video from `www.metacafe.com`, which does not apply HTTPS. Please choose a video of at least 2 minutes duration. Visualize the throughput over time and shortly describe the results.

(c) *Compare and interpret your results.*
What are the main differences between the file download and the video stream in terms of traffic behaviour? In this context, also have a look on the HTTP GET requests sent by these applications.

(d) *Filtering packets from a video stream using HTTPS.*
Now, analyze the traffic from a video stream transmitted via HTTPS, e.g. from YouTube. Can you still get information about the GET requests sent by the client? What are the challenges when analyzing HTTPS traffic, compared to HTTP traffic?

For the following tasks, you need Python, Matplotlib[1], and the Open Source Computer Vision Library (openCV)[2]. We also advise you to use numpy[3], scipy[4], and Image (from PIL)[5].

Question 2: (40 Points) *Image analysis*

(a) Download the two images *lena.png* and *northcap.png* from `https://service.inet.tu-berlin.de/owncloud/index.php/s/Gwbliu8XqH7yd8o`. Use the following commands to read the images from your filesystem and show them:
`cv2.imread()`
`plt.imshow()`
What do you notice when displaying the images using imshow() compared to opening the picture with any image viewer on your machine? What is the reason for this?

(b) Find out, how many color space conversions your OpenCV version provides.

(c) Transform the Lena image to the RGB color space, plot the single components (R,G,B), and plot the distributions of the component's intensities.

(d) Compute the entropy of R,G, and B component, for the North Cap and the Lena image! What can you see?

(e) Transform the Lena and North Cap images to the YCbCr color space and plot the single components as well as the distribution of their intensities!

(f) Compute the entropy of the Y, Cb, and Cr-component of both images. Compare the results to those obtained in (d), what can you see?

(g) What is the difference between subsampling and a mean filter?

(h) Perform subsampling (4:4:4 $\rightarrow$ 4:2:2/4:2:0) on the Cb and Cr components of Lena. Compute the entropies of YUV 4:4:4, 4:2:2, and 4:2:0. Afterwards, subjectively rate the quality of the resulting images.

Question 3: (15 Points) *Image and frequency space* Load the Lena image and transform it to the YUV space. For the following tasks, only consider the Y component.

(a) Transorm the image using *numpy.fft2* and plot the absolute value of the transformed Y-channel. What can you see? What happens when you additionally use *fftshift*?

(b) Implement a mean filter and perform a convolution with the image. Vary the filter size ($n = 3, 4, 5, ..$). At which values of n can you recognize differences in the resulting image?

(c) Add a Gaussian noise to the image. Consider 1) $\mu = 0, \sigma = 10$ and 2) $\mu = 0, \sigma = 50$. Transform the image to the frequency domain. How do you recognize the noise?

[1] https://matplotlib.org/
[2] https://pypi.org/project/opencv-python/
[3] http://www.numpy.org/
[4] https://www.scipy.org/
[5] https://pillow.readthedocs.io/en/3.1.x/reference/Image.html

Question 4: (25 Points) *Haar Transformation* Load the Lena image and transform it to the YCbCr space. For the following tasks, only consider the Y component.

(a) Compute the image's entropy, as well as the estimated file size in case of an optimal encoding.

(b) Perform a quantization with quantization steps $q = \{2, 5, 10, 20, 50\}$. How does the entropy change and how do the quantized images look like?

(c) Peform the Haar transformation with a block size of 1x2 blocks. Once more, compute the entropy and expected file size for the different quantization steps. (Keep in mind the reordering step!)

(d) Repeat subtask (c) with a block sizes $bs = \{2x2, 4x4, 8x8\}$ and compare the results.

(e) Repeat subtask (d), but use DCT instead of Haar transformation. Compare the results!

Question 5: (20 Points) *JPEG Compression and PSNR*

(a) Write your own function to compute the Peak Signal to Noise Ratio (PSNR) to measure the distortion of a compressed image (you only need to consider the images' Y-channels for this function). You can find the definition of PSNR at the end of this document.

(b) Take the Lena and Northcap image and save them as JPEG using the *save* function from *PIL Image*. Repeat this with different values for the quality flag, i.e. $quality = \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. Visualize the resulting file size and PSNR values!

(c) Re-do substask (b), but this time, set the flag *optimize* to true. Which values are, compared to the results from (b), affected? The resulting file size or the PSNR values? Visualize the results in a comparative manner.

(d) The python module *timeit* provides a simple way to measure execution time. Monitor the execution time when *optimize* is (1) true and (2) false for $quality = \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. Visualize the trade-off between compression time and compression result!

Question 6: (15 Points) *Image compression in the wild*

(a) Use the smartphone of one of your team members and take a picture. Send the image via a messenger of your choice (WhatsApp, Telegram, etc.) to another team member. What can you observe when comparing the original image to the one received by your colleague?

(b) Plot a part of the original and received picture, so that the difference is made visible.

(c) You are a software developer working on a messenger app and responsible for implementing the compression. Which aspects would you consider when deciding about the way images and videos are compressed?

Comments

$PSNR = 10 * log_{10} \frac{R^2}{MSE}$

$MSE = \frac{\sum_{M,N} [I_1(m,n)] - [I_2(m,n)]}{M * N}$

R denotes the maximum value, i.e. 255 when pixels are represented using 8 bits per sample