Assignment 3
Manipulating Images in Frequency Domain

Please note:

1. What you must hand in includes the assignment report (.pdf) and – if necessary – source codes (.py or .m). Please zip them all together into an archive file named according to the following template: XXXXXXXX.zip
   Where XXXXXXXX must be replaced with your student ID.
2. Some problems are required to be solved *by hand* (shown by the icon), some need to be implemented (shown by the icon), and some need to be dockerized (shown by the icon).
3. As for the first type of the problems, you are free to solve them on a paper and include the picture of it in your report. Here, cleanness and readability are of high importance.
4. Your work will be evaluated mostly by the quality of your report. Don't forget to explain what you have done, and provide enough discussions when it's needed.
5. 5 points of each homework belongs to compactness, expressiveness and neatness of your report and codes.
6. By default, we assume you implement your codes in Python. If you're using Matlab, you have to use equivalent functions when it is asked to use specific Python functions.
7. Your codes must be separated for each question, and for each part. For example, you have to create a separate .py file for part b. of question 3. Please name it like p3b.py.
8. Problems with bonus points are marked by the icon.
9. If there is ***any*** question, please don't hesitate to contact us through the following email addresses:
   - Atghaei@aut.ac.ir
   - fardin.aiar@gmail.com
   - minoo.dolatabadi75@gmail.com
   - aidin.khalili@ymail.com
   - b.roshanfekr@aut.ac.com
10. Unfortunately, it is quite easy to detect copy-pasted or even structurally similar works, no matter being copied from another student or internet sources. Try to send us your own work, without being worried about the grade! ;)

1. Decoding Identities – A Frequency Domain Approach to Face Recognition (20 Pts.)

Owners of modern smartphones are often surprised by the perceived complexity of face recognition systems such as Apple Face ID. In reality, even a system based on fundamental knowledge of images in the frequency domain can be quite capable. In this assignment, you will use template matching to design a specific filter, namely the Minimum Average Correlation Energy (MACE) filter, and perform cross correlation between a test image and the designed filter.

The intuition behind the MACE filter design is to create a filter from a set of training images that minimizes the average correlation energy, thus producing a correlation plane with a sharp peak at the origin for genuine matches while maintaining low responses elsewhere. For a training set with N images of size $n \times m$ (with $d = n \times m$ pixels), the MACE filter H is given by:

$$H = D^{-1}X(X^{*}D^{-1}X)^{-1}u$$

where:

- D is a $d \times d$ diagonal matrix that contains the average correlation energies of the training images on its diagonal elements.
- X is a $N \times d$ matrix whose columns are the Discrete Fourier Transform (DFT) coefficients of the training images.
- X^{*} denotes the complex conjugate of X.
- u is an $N \times 1$ vector containing the desired correlation peak values (usually set to 1 for all training images of the authentic class).

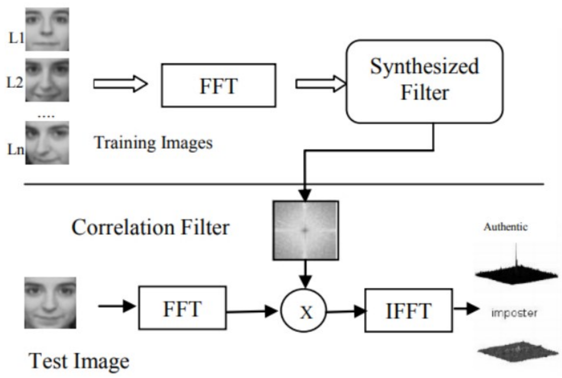


Figure 1: Block diagram of the algorithm. Training images are used to obtain a synthesized filter in the frequency domain. In the testing phase, the filtering output will give high peak value at the origin of the correlation plane if the test face belongs to true class.

The filter is optimized to produce a desired correlation output plane. A sharp peak in the correlation plane indicates that the test image corresponds to the genuine class, while lower responses or multiple peaks indicate an imposter. Please refer to [this paper](#) for more details about how the algorithm works.

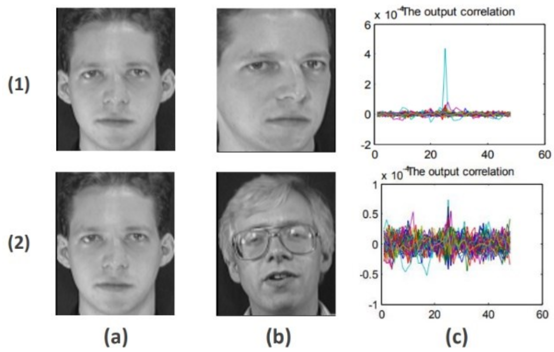


Figure 2: Two test cases, one a 'matched' and the other a 'non matched' state. As can be seen, correlation coefficients shows peak when the classes are the same (a) Test image (b) training a images (c) The correlation coefficients

Download the provided cropped version of a face database (a minimal subset of the AT\&T (ORL) Face Database). This subset consists of a few images per subject to match limited computational resources. All images must be converted to grayscale before further processing.

a) **Basic Implementation**

   i. **Data Selection:** Consider subjects s1, s2, s3, s4, and s5. Use images 1 to 7 of each subject as the training set, and use the remaining images of each subject as the test set.

ii. **Algorithm Implementation:** Pick a random test image and perform face recognition using the MACE filter algorithm described above. Your implementation should:
   - Compute and display the MACE filter (both magnitude and phase).
   - Compute the cross correlation between the test image and the MACE filter, marking the peak clearly.
   - Output the final recognition decision (matched/unmatched).

b) Recognition Rate Calculation

Calculate the recognition rate by dividing the number of true identifications by the total number of test images.

Submission Guidelines

Report: Submit a complete report that includes your theoretical derivations, code explanations, experimental results, and annotated visual outputs (including the MACE filter's magnitude/phase and correlation maps with detected peaks). Discuss the strengths and limitations of the frequency-domain approach to face recognition.

Source Code: Attach your MATLAB or Python code as an appendix in your report. Your code must be well-commented and modular to facilitate testing and parameter adjustments.

Visual Results: Provide all relevant images and plots generated during your experiments.

2. Image Inpainting: Interpolation vs. PDE-Based Inpainting (35 Pts.)

In this assignment, you will implement an inpainting system using *Partial Differential Equations (PDE)* and *Interpolation* methods to restore missing parts of images. You will also analyze and compare the performance of these methods in terms of visual quality, computational efficiency, and parameter sensitivity. Through this process, you will develop an understanding of how mathematical models can be used to solve image restoration problems.

Task 1: Data Preparation and Preprocessing

a) Image Setup:
- Download the provided images, which consist of both *grayscale* and *color (RGB)* images. The images contain missing or destroyed regions, and your goal is to restore these missing areas using different inpainting methods.
- Convert the *grayscale image* to a numeric format with pixel values scaled to [0,1].
- For the *color image*, ensure each channel (Red, Green, Blue) is handled independently, and normalize the pixel values to the range [0,1].

Question: Explain why converting the images to the normalized range is important for the subsequent inpainting algorithms.

b) Destroyed Region Definition:

- Define the destroyed regions in each image. You can either manually select the destroyed regions or use a binary mask if provided or create your own.

Question: How do you store and utilize the binary mask to identify the destroyed regions? Describe the necessary data structures you will use.

Task 2: Interpolation-Based Inpainting

c) Coordinate Extraction:

- Extract the coordinates and pixel intensities of the *undamaged pixels* in both grayscale and color images. These will serve as the basis for interpolation.

Question: Write a code snippet (or describe the approach) to extract the coordinates of all valid pixels and their corresponding intensities from the images.

d) Implement Interpolation Method:

- Implement an interpolation-based inpainting method (e.g., *nearest-neighbor*, *linear*, or *cubic interpolation*) to fill the destroyed regions.

Question: How does each interpolation method behave differently in terms of edge preservation and smoothness? Compare two methods (e.g., nearest vs. linear interpolation) for both *small destroyed areas* and *large missing regions*.

e) Inpainting for Color Images:

- Apply the interpolation method on the *RGB channels* independently, or propose a method to handle all channels together.

Question: How would you handle inpainting for color images? What potential issues might arise when interpolating each color channel independently?

f) Visual and Quantitative Evaluation:

- Display the restored images for both grayscale and color images. Provide a visual comparison between the original, destroyed, and inpainted images.

Question: Calculate the *PSNR (Peak Signal-to-Noise Ratio)* for the inpainted images compared to the original ones. How does the interpolation method affect the PSNR value?

Task 3: PDE-based Inpainting

g) PDE Formulation and Initialization:

- Implement the *PDE-based inpainting* method using the heat-diffusion equation:

$$u_t = \lambda \, \Delta u + \chi_{\Omega \setminus D}(f - u)$$

where u is the evolving image, Δu is the Laplacian, and λ is the diffusion coefficient.

Question: Explain the roles of λ, Δu, and the fidelity term $\chi_{\Omega \setminus D}(f - u)$ in this equation. What effect does changing λ have on the diffusion process?

h) Boundary Conditions and Implementation:

- Implement *Neumann boundary conditions* to prevent the boundary from diffusing into the image. Iterate through the image pixels and solve the PDE using an explicit finite difference scheme.

Question: How do you handle the boundary condition numerically? Describe how ghost points or direct matrix modifications work to implement Neumann boundary conditions.

i) Parameter Sensitivity:

- Experiment with two sets of λ and Δt values. Compare the inpainting results, including smoothness and edge preservation.

Question: How does varying λ affect the smoothing of edges in the inpainted images? What happens if Δt is too large or too small? Provide visual examples and a stability analysis.

j) Inpainting for Color Images:

- Implement the PDE-based inpainting for *color images*. Either apply it to each channel independently or suggest an alternative approach to account for all three channels together.

Question: If you are applying the PDE to each color channel independently, how might this affect color consistency? Would a coupled approach provide better results?

Task 4: Performance Evaluation and Comparison

k) Visual Comparison:

- After performing inpainting using both the *interpolation-based* and *PDE-based* methods, visually compare the results for both grayscale and color images. Discuss:
    - Which method is better at preserving *edges*?
    - Which method performs better for *large missing regions*?

Question: Based on your visual observations, which method would you choose for large missing areas? Why?

l) Quantitative Metrics:

- Calculate the *MSE (Mean Squared Error)* and *PSNR (Peak Signal-to-Noise Ratio)* for both methods, using the original image as the reference.

Question: How do the *MSE* and *PSNR* values compare between the interpolation-based and PDE-based methods? What can you infer from these metrics about the quality of the restored images?

m) Computational Efficiency:

- Compare the computational time for both inpainting methods (interpolation vs. PDE) on images with different sizes of destroyed regions.

Question: Which method is more computationally efficient for small regions of missing data? Which method might become slower as the destroyed area increases?

Submission Guidelines

Report: Address the questions from each task, integrating visual results, code snippets, and metrics. Discuss the strengths and weaknesses of each inpainting method and their performance in different scenarios.

Code Implementation: Provide well-documented code for the *interpolation-based* and *PDE-based* inpainting methods. Ensure that the code handles both *grayscale* and *color* images, and includes the ability to apply Neumann boundary conditions for PDE-based inpainting.

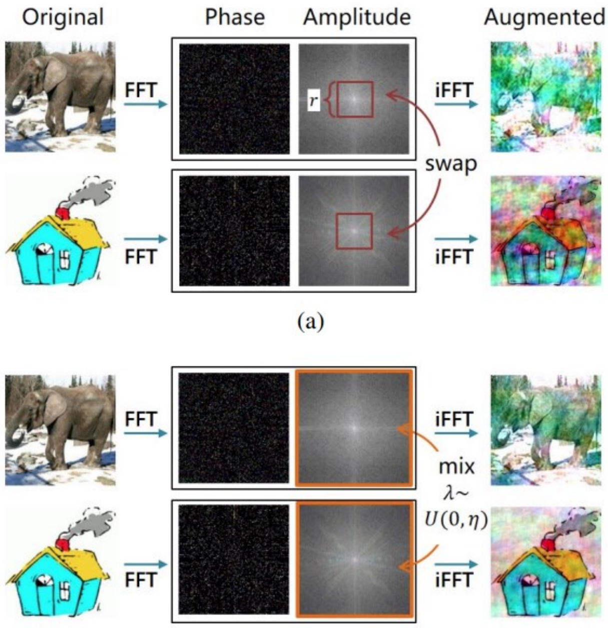
Results and Comparison: Include *before-and-after* images showing the results of inpainting for both methods. Provide a *table of metrics* (MSE, PSNR) if the ground-truth is available, or otherwise provide qualitative analysis of artifacts.

3. Innovative Augmentation with FFT (25 Pts.)

In 2021, Xu et al. (Xu, Qinwei, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. "A Fourier-based framework for domain generalization." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14383-14392, 2021) introduced an augmentation method designed to address the domain generalization problem. The main concept is that when images are transformed using FFT, the semantic information of an image is primarily encoded in its phase spectrum, while the amplitude spectrum plays a less significant role, particularly in semantic of the image. This proposed approach is particularly beneficial for addressing the domain generalization problem.

The proposed augmentation method involves selecting two images, one from a source domain (image 1) and another from a different domain (image 2). The paper introduces two augmentation techniques (illustrated in figure below) including:

- **AmpSwap (AS) :** In this method, the phase of image 1 is retained, and the amplitude is swapped with the amplitude of image 2.
- **AmpMix(AM) :** Here, the phase of image 1 is retained, and the amplitudes of both images are mixed to create a new amplitude for the augmented image. The weight of each image's amplitude in the mix should be adjustable.



(a) (b)

After applying these modifications, you are required to perform an inverse FFT to reconstruct the augmented image.

- All images are in BGR format.
- You can use images from the PACS dataset across different domains for this task.

What we want?

First : Implement this augmentation code by your own with definitions that gets each weights for AmpMix. (def Ampmix(image1,image2,w1,w2))
Second step : Provide a brief explanation of the domain generalization problem and discuss how this augmentation technique helps improve performance in addressing this issue (at least 5 lines description).

Third step(bonus): After experimenting with this algorithm, can you suggest alternative augmentation methods inspired by the transformations discussed in the course or beyond? These methods should preserve the semantic content of the original image while modifying its visual details based on another image. We are looking for novel and logically sound techniques, excluding previously mentioned augmentations like CutMix or others. This idea could serve as the foundation for writing a highly-cited article by your own.

4. Practicing the Basics of 1-D Fourier Transform (15 Pts.)

The Fourier Transform is a powerful tool that allows us to represent time-domain signals in the frequency domain. Understanding the Fourier Transform of different signals can give us deeper insight into their frequency components. In this exercise, you will compute the Fourier Transforms of more complex functions and apply your knowledge of Fourier properties.

The objective of this problem is to strengthen your understanding of the theoretical foundations of the 1-D Fourier transform and to prepare you for applications in signal processing.

a) To begin, use the properties of the Fourier transform to compute and determine the 1-D and 2-D Fourier transform of the following functions:

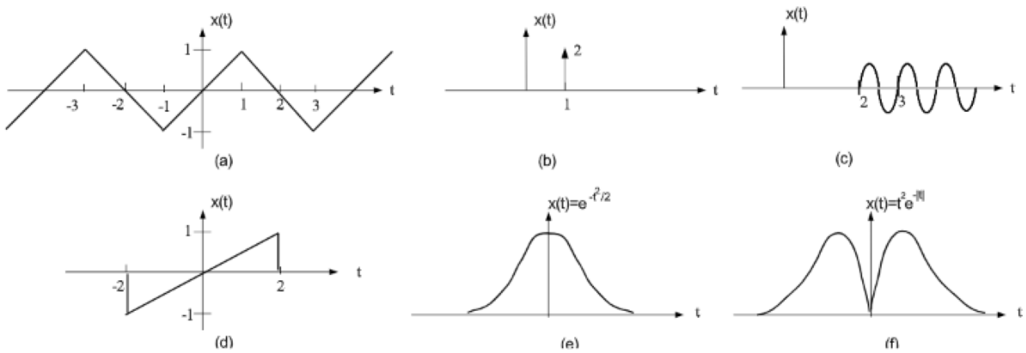
1. $x(t) = (e^{-\alpha t})\, u(t)$, for $\alpha > 0$. Compute the Fourier transform for this function where $u(t)$ is the Heaviside step function.
2. $x(t) = \delta(t-3) + \delta(t+3) - 2\delta(t)$.
3. $x(t) = \left(\frac{1}{1+t^2}\right) \cdot e^{i\pi t}$.
4. $x(t) = \left(\frac{\sin(\pi t)}{\pi t}\right) \cdot e^{-\alpha t}$, for $\alpha > 0$.
5. $\Pi(x,y) = \begin{cases} 1 & \text{if } |x| < 0.5 \text{ and } |y| < 0.5 \\ 0 & \text{otherwise} \end{cases}$

   $\Pi(x,y) = \Pi(x)\Pi(y)$
6. $f(x,y) = \Pi(x)\delta(x-y)$

b) Next, determine which of the following conditions are satisfied by any of the signals in the figure below:

1. $\text{Re}\{X(j\omega)\} = 0$
2. $\text{Im}\{X(j\omega)\} = 0$
3. There exists a real α such that $e^{j\omega\alpha}X(j\omega)$ is real
4. $\int_{-\infty}^{\infty} X(j\omega)\, d\omega = 0$
5. $\int_{-\infty}^{\infty} \omega X(j\omega)\, d\omega = 0$
6. $X(j\omega)$ is periodic

Below are the real signals plotted in the time domain:



a) Finally, let $f = [1, -1, 2, 3]$ and $g = [1, -2, -1]$.

1. Pad both f and g with zeros to a length of 8 and find the 1-D convolution $f * g$.
2. Find the convolution of f and g using the Discrete Fourier Transform (DFT) approach. Show all steps of your calculation.
3. Plot the Fourier Transforms of f and g in the complex plane for $0 \leq \omega \leq 9$.
4. Calculate and interpret the following:
   (a) $\sum_x |f(x)|^2$
   (b) $\sum_x |g(x)|^2$

   Where $f(x)$ and $g(x)$ are your original sequences.

5. Some Explanatory Questions (15 Pts.)

Please answer the following questions as clearly as possible:

a) Provide a mathematical explanation of how a *rotation* of an image in the spatial domain affects its *Fourier domain*. Specifically, discuss how the phase and magnitude of the Fourier Transform of an image are impacted when the image is rotated by a certain angle.

b) *Image compression* techniques, such as JPEG, typically use a *block-wise discrete cosine transform (DCT)*. Discuss the advantages of using the DCT for compression, and explain why the DCT is preferred over other transforms, such as the Discrete Fourier Transform (DFT), for this purpose.

c) Explain why image blurring in the spatial domain corresponds to multiplication by a low-pass filter in the frequency domain. How does the blurring process affect the magnitude and phase of the Fourier Transform of an image?

d) Band-pass filtering in image processing is used to retain specific frequency components of an image while suppressing others. Explain how a band-pass filter works in the frequency domain and describe the resulting effects on an image's appearance in the spatial domain after filtering.

e) Aliasing can occur during sampling, and it is generally avoided by ensuring the sampling rate is above the Nyquist rate. However, in real-world scenarios, aliasing may still occur due to imperfections in sampling devices or noise. Explain why aliasing can still occur even when the Nyquist criterion is theoretically satisfied.

f) The shift theorem of the Fourier Transform states that a shift in the spatial domain corresponds to a linear phase shift in the Fourier domain. Explain how this property is useful in practical applications like image registration or image alignment.

g) Image deblurring is a common technique used in image restoration. Discuss the typical steps involved in deblurring an image using Fourier transforms, and explain how the frequency domain is utilized in this process.

Good Luck