(1)

a) $y[n] = x[n] * h[n] \Rightarrow \begin{cases} y[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k] \\ Y(e^{jw}) = X(e^{jw})H(e^{jw}) \\ Y(z) = X(z)H(z) \end{cases}$

1 روش:

$$x[n-1] * h[n-1] = \sum_{k=-\infty}^{+\infty} x[k-1]h[n-k-1] \underbrace{=}_{k-1=s} \sum_{s=-\infty}^{+\infty} x[s]h[n-s-2] = y[n-2] \neq y[n-1] \Rightarrow$$ پاسخ نیاز است.

2 روش:

$$x[n-1] * h[n-1] = X(z)z^{-1}H(z)z^{-1} = X(z)H(z)z^{-2} = Y(z)z^{-2} = y[n-2] \neq y[n-1]$$

b) $y(t) = x(t) * h(t) \rightarrow \begin{cases} y(t) = \int_{-\infty}^{+\infty} x(\tau)h(t-\tau)d\tau \\ Y(jw) = X(jw)H(jw) \\ Y(s) = X(s)H(s) \end{cases}$

1 روش

$$x(-t) * h(-t) = \int_{-\infty}^{+\infty} x(-\tau)h(-t+\tau)d\tau \underbrace{=}_{s} -\int_{+\infty}^{-\infty} x(s)h(-t-s)ds = \int_{-\infty}^{+\infty} x(s)h(-t-s)ds = y(-t) \checkmark$$

2 روش:

$$x(-t) * h(-t) : X(-s) \cdot H(-s) = Y(-s) \Rightarrow y(-t) \checkmark$$

(2)

$$S_1 : w[n] = \frac{1}{2}w[n-1] + x[n]$$

$$S_2 : y[n] = \alpha y[n-1] + \beta w[n-1]$$

$$H_1(z) = \frac{1}{1-\frac{1}{2}z^{-1}} \overset{\text{ROC}}{\rightarrow} |z| > 1/2$$

$$H_2(z) = \frac{\beta}{1-\alpha z^{-1}} \overset{\text{ROC}}{\rightarrow} |z| > \alpha$$

$$\Rightarrow \alpha = 1/4, \quad \beta = 1$$

$$H(z) = \frac{1}{1-\frac{3}{4}z^{-1}+\frac{1}{8}z^{-2}} = \frac{1}{(1-\frac{1}{2}z^{-1})(1-\frac{1}{4}z^{-1})} \quad \text{A}$$

b)

$$h_1[n] * h_2[n] = H_1(z) \cdot H_2(z) = H(z) = \frac{1}{(1-\frac{1}{2}z^{-1})(1-\frac{1}{4}z^{-1})} = \frac{A}{1-\frac{1}{2}z^{-1}} + \frac{B}{1-\frac{1}{4}z^{-1}}$$

$$A = 2$$

$$B = -1$$

ROC : $|z| > 1/2$

ROC $\Rightarrow h[n] = 2(\frac{1}{2})^n u[n] - (\frac{1}{4})^n u[n]$

(3)

a) $x_1[n] = 1 + \sin\left(\frac{3\pi}{8}n + \frac{\pi}{4}\right) = 1 + \frac{1}{2j}e^{j\left(\frac{3\pi}{8}n+\frac{\pi}{4}\right)} - \frac{1}{2j}e^{j\left(\frac{3\pi}{8}n+\frac{\pi}{4}\right)}$

$X_1(e^{jw}) = 2\pi\,\tilde{\delta}(w) + \frac{e^{j\frac{\pi}{4}}}{2j}\tilde{\delta}\left(w+\frac{3\pi}{8}\right) - \frac{e^{-j\frac{\pi}{4}}}{2j}\tilde{\delta}\left(w-\frac{3\pi}{8}\right)$

$X_1(e^{jw}) \cdot H(e^{jw}) \Longrightarrow Y(e^{jw}) = \frac{e^{j\frac{\pi}{4}}}{2j}\tilde{\delta}\left(w+\frac{3\pi}{8}\right) - \frac{e^{-j\frac{\pi}{4}}}{2j}\delta\left(w-\frac{3\pi}{8}\right)$

$$\boxed{y[n] = \sin\left(\frac{3\pi}{8}n + \frac{\pi}{4}\right)}$$

b) $x_2[n] = \sum_{k=-\infty}^{+\infty}\left(\frac{1}{2}\right)^{n-4k}u[n-4k] = \left(\frac{1}{2}\right)^n u[n] * \sum_{k=-\infty}^{+\infty}\delta[n-4k]$

$X_2(e^{jw}) = \frac{1}{1-\frac{1}{2}e^{-jw}} \cdot \sum_{k=-\infty}^{+\infty}\frac{\pi}{2}\delta\left(w-k\frac{\pi}{2}\right)$

$X_2(e^{jw})H(e^{jw}) = 0 \Longrightarrow \boxed{y[n]=0}$

هیچ بازه‌ای با شتاب‌ای نداریم.

(4)

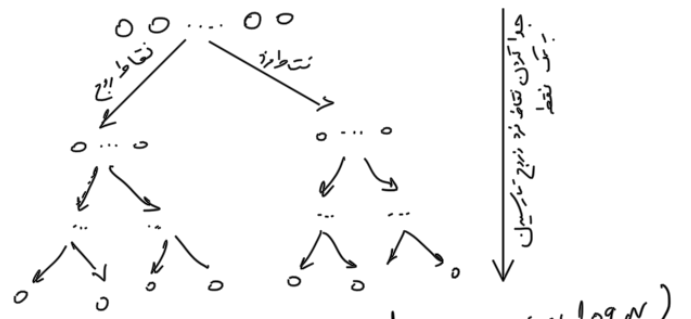
a) محاسبه کانولوشن دوبعدی به صورت مستقیم حاصل N^2 مرتبه ضرب N^2 داریم که برابر $O(N^4)$ خواهد بود.

b) استفاده از FFT برای محاسبه $(x*y)[m,n]$:

$$x * y = F^{-1}\left\{F\{x\} \cdot F\{y\}\right\}$$

برای محاسبه عبارت فوق به صورت efficient، از الگوریتم FFT (تبدیل) استفاده کنید که از مرتبه $O(N^2 \log N^2)$ خواهد بود که در زیر به شرح آن می‌پردازیم:

برای سادگی FFT یک بعدی توضیح داده خواهد شود سپس به ابعاد بالاتر تعمیم خواهد یافت.



$$h = \log N \Rightarrow \text{complexity} = O(\underbrace{N}_{\text{تعداد نقاط}}\underbrace{\log N}_{h})$$

2D $\Longrightarrow$ $h = \log N^2$

$\text{complexity} = O(\underbrace{N^2}_{\text{تعداد نقاط}}\underbrace{\log N^2}_{h})$

محاسبه FT هر نقطه $= O(1)$

ترکیب FT در هر نقطه فرد و زوج $= O(1)$

در نتیجه محاسبه $F\{x\}$ و $F\{y\}$ و $F^{-1}\{z\}$ هرکدام برابر با $O(N^2\log N^2)$ قابل محاسبه خواهد، پیچیدگی نهایی از مرتبه $O(N^2\log N^2)$ خواهد بود.

(5) Store (Imaging) data: تمام انواع تصاویر اعم از CT، MR، X-Ray، Ultrasound، PET و... و هم چنین single یا multiframe را DICOM ذخیره می‌کند.

Manage (Imaging) data: نه تنها pixel ها بلکه meta-data ها را هم دارد مثل Patient IDentification، demographics، content و workflow را مدیریت می‌کند.

Display Images: screen calibration، annotation، layout، key img flagging

Distribute Images: Network Push/Pull، media transfer، web Protocols، Email attachment.

secure: audit Trails، De-identification schemes، Encryption.

store analysis Results: Registration، segmentation، ...

این اطلاعات باعث می‌شود که موقعیت وضع و دستگاه تصویربرداری را بدانیم، همچنین اطلاعات مربوط به تصویر و شخص هم‌خوانی داشته باشند. پردازش و تحلیل‌های مورد نیاز دقیق‌تر و بدون خطای اطلاعاتی باشد.

Rescale slope: این تگ بیانگر یک مقدار عددی است که میزان ضریب scaling برای اعمال به مقادیر pixel ها را تعریف می‌کند. فرمول مورد استفاده:

Rescale slope * Pixel value + Rescale Intercept

Rescale Intercept: این تگ بیانگر یک مقدار عددی است که میزان value جمع‌شونده به مقادیر pixel ها را مشخص می‌کند.

b) DICOM anonymization بر روی حذف اطلاعات حساس بیمار برای حفظ Privacy تمرکز دارد همچنین از DICOM encryption شامل رمزگذاری داده‌های DICOM برای محافظت از محرمانه بودن و یکپارچگی داده در طول انتقال یا ذخیره‌سازی است.

anonymization در DICOM به طرق مختلفی انجام می‌شود:

De-identification: bluring، Pixelating و blocking

masking: عوض شناسه‌های file، و mask کردن آنها.

Pseudonymization.

```
The custom deidentifier class
from deid.config import DeidRecipe
from deid.dicom.parser import DicomParser
import pydicom
from Crypto.Hash import SHA512
from datetime import datetime

class DeidDataset:
    """This class allows to pseudonymize an instance of
    pydicom.Dataset with our custom recipe and functions.
    """
    def __init__(self, secret_salt: str, recipe_path: str):
        """New instance of our pseudonymizer class.

        :param secret_salt: a random string that makes the
         hashing harder to break.
        :param recipe_path: path to our deid recipe.
        """
        self.secret_salt = secret_salt
        self.recipe = DeidRecipe(recipe_path)
```

```
def pseudonymize(self, dataset:pydicom.Dataset) -> pydicom.Dataset:
    """Pseudonymize a single dicom dataset

    :param dataset: dataset that will be pseudonymized
    :returns: pseudonymized dataset
    """
    parser = DicomParser(dataset, self.recipe)
    # register functions that are specified in the recipe
    parser.define('replace_name', self.replace_name)
    parser.define('hash_func', self.deid_hash_func)
    parser.define('remove_day', self.remove_day)
    # parse the dataset and apply the deidentification
    parser.parse(strip_sequences=True, remove_private=True)
    return parser.dicom

# All registered functions that are used in the recipe must
# receive the arguments: `item`, `value`, `field`, `dicom`

def deid_hash_func(self, item, value, field, dicom) -> str:
    """Performs self.hash to field.element.value"""
    val = field.element.value
    return self.hash(str(val))

@staticmethod
def remove_day(item, value, field, dicom) -> str:
    """Removes the day from a DT field in the deid framework"""
    dt = datetime.strptime(field.element.value, '%Y%m%d')
    return dt.strftime("%Y%m01")

@staticmethod
def replace_name(item, value, field, dicom) -> str:
    """Replace PatientName with PatientSex and coarse PatientAge"""
    sex = dicom.get('PatientSex')
    sex = {"F":'Female', "M": 'Male', 'O':'Other'}[sex]
    age = DeidDataset.round_to_nearest(int(dicom.get('PatientAge')[:-1]), 5)
    return f"{sex} {age:03d}Y {dicom.get('Modality')}"

# Helper methods for our registered ones
@staticmethod
def round_to_nearest(value, interval):
    """Rounds value to closest multiple of interval"""
    return interval * round(value/interval)

def hash(self, msg: str) -> str:
    """
    :param msg: message that we want to encrypt,
     normally the PatientID or the StudyID.
    :return: the encrypted message as hexdigest
     (in characters from '0' to '9' and 'a' to 'f')
    """
    assert type(msg) == str, f"value is not of type str, {type(msg)}"
    h = SHA512.new(truncate="256")
    bytes_str = bytes(f"{self.secret_salt}{msg}", "utf-8")
    h.update(bytes_str)
    return str(h.hexdigest())
```