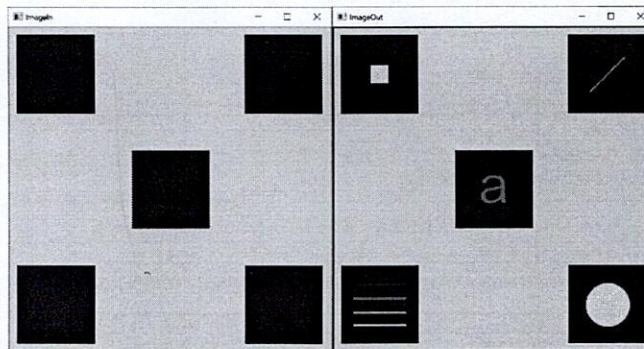


Câu 1: (2đ)

Xây dựng và cài đặt thuật toán làm rõ những đối tượng được nhúng trong năm hình vuông màu đen dùng thống kê histogram. (1.5đ)

Tại sao trong trường hợp này cân bằng histogram toàn cục lại không có kết quả? (0.5đ)



Đáp án:

```
def HistogramStatistics(imgin, imgout):
    M, N = imgin.shape
    h = np.zeros(L, dtype = np.int32)
    m = 3
    n = 3
    w = np.zeros((m,n), dtype = np.uint8)
    a = m // 2
    b = n // 2
    mG, sigmaG = cv2.meanStdDev(imgin)
    # mG = np.mean(imgin)
    # sigmaG = np.std(imgin)
    C = 22.8
    k0 = 0.0
    k1 = 0.1
    k2 = 0.0
    k3 = 0.1

    for x in range(a, M-a):
        for y in range(b, N-b):
            for s in range(-a, a+1):
                for t in range(-b, b+1):
                    w[s+a, t+b] = imgin[x+s, y+t]
            msxy, sigma_sxy = cv2.meanStdDev(w)
```



```

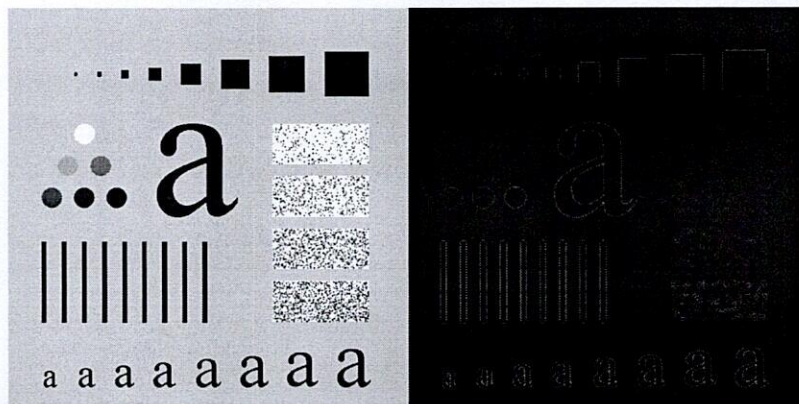
if (k0*mG <= msxy and msxy <= k1*mG) and (k2*sigmaG <= sigmasxy and
sigmasxy <= k3*sigmaG):
    imgout[x, y] = (C*imgin[x,y]).astype(np.uint8)
else:
    imgout[x, y] = imgin[x, y]

```

Do sự chênh lệch độ sáng của các điểm ảnh cần quan tâm so với phần còn lại là quá lớn nên ta không thể dùng cân bằng histogram toàn cục.

Câu 2: (3đ)

Xây dựng và cài đặt thuật toán lọc ảnh trong miền tần số dùng bộ lọc thông cao Butterworth có tần số cắt là $D_0 = 60$ và bậc của bộ lọc là $n = 2$ cho hình sau:



Đáp án:

```

def CreateHighPassFilter(P, Q):
    n = 2
    D0 = 60
    H = np.zeros((P,Q,2), np.float)
    for u in range(0, P):
        for v in range(0, Q):
            D = np.sqrt((u-P//2)**2 + (v-Q//2)**2)
            if D > 0:
                H[u,v,0] = 1/(1 + np.power(D0/D, 2*n))
    return H

def FrequencyFilter(imgin):
    # Bước 1:
    # Mở rộng ảnh có kích thước PxQ
    # Thêm số 0 vào phần mở rộng
    M, N = imgin.shape
    P = cv2.getOptimalDFTSize(M)
    Q = cv2.getOptimalDFTSize(N)
    f = np.zeros((P,Q), np.float)
    f[0:M,0:N] = imgin

    # Bước 2: DFT

```



```

F = cv2.dft(f, flags = cv2.DFT_COMPLEX_OUTPUT)

# Buoc 3: Shift vao image center
F = np.fft.fftshift(F)

# Buoc 4: Tao bo loc H
H = CreateHighPassFilter(P, Q)
# Buoc 5: G = FH
G = cv2.mulSpectrums(F, H, flags = cv2.DFT_ROWS)

# Buoc 6: Shift return
G = np.fft.ifftshift(G)

# Buoc 7: IDFT
g = cv2.idft(G, flags = cv2.DFT_SCALE)

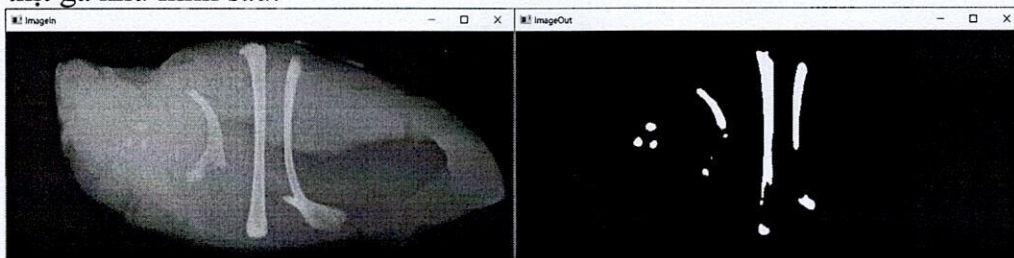
# Buoc 8: Lay phan thuc
g = g[0:M,0:N,0]

g = np.clip(g, 0, L-1)
return g.astype(np.uint8)

```

Câu 3: (2đ)

Xây dựng và cài đặt thuật toán đếm và tính các điểm ảnh của các mảnh xương trong miếng thịt gà như hình sau:



Đáp án:

```

def MyConnectedComponent(imgin):
    ret, temp = cv2.threshold(imgin, 200, L-1, cv2.THRESH_BINARY)
    temp = cv2.medianBlur(temp, 7)
    M, N = temp.shape
    dem = 0
    color = 150
    for x in range(0, M):
        for y in range(0, N):
            if temp[x,y] == L-1:
                mask = np.zeros((M+2,N+2),np.uint8)
                cv2.floodFill(temp, mask, (y,x), (color,color,color))
                dem = dem + 1
                color = color + 1
    print('Co %d thanh phan lien thong' % dem)

```



```

a = np.zeros(L, np.int)
for x in range(0, M):
    for y in range(0, N):
        r = temp[x,y]
        if r > 0:
            a[r] = a[r] + 1
dem = 1
for r in range(0, L):
    if a[r] > 0:
        print('%4d %5d' % (dem, a[r]))
        dem = dem + 1
return temp

```

```

def ConnectedComponent(imgin):
    ret, temp = cv2.threshold(imgin, 200, L-1, cv2.THRESH_BINARY)
    temp = cv2.medianBlur(temp, 7)
    dem, label = cv2.connectedComponents(temp)
    print('Co %d thanh phan lien thong' % (dem-1))

```

```

a = np.zeros(dem, np.int)
M, N = label.shape
color = 150
for x in range(0, M):
    for y in range(0, N):
        r = label[x, y]
        a[r] = a[r] + 1
        if r > 0:
            label[x,y] = label[x,y] + color

for r in range(1, dem):
    print('%4d %10d' % (r, a[r]))
return label.astype(np.uint8)

```

Câu 4: (3đ)

Implementing the LeNet-5 convolutional neural network architecture to classify MNIST Digits as following:

```

def LeNet_5():
    model = Sequential()
    model.add(Conv2D(filters = 6,
                    kernel_size = 5,
                    strides = 1,
                    activation = 'relu',
                    input_shape = (32,32,1)))
    model.add(MaxPooling2D(pool_size = 2, strides = 2))
    model.add(Conv2D(filters = 16,
                    kernel_size = 5,
                    strides = 1,
                    activation = 'relu',
                    input_shape = (14,14,6)))
    model.add(MaxPooling2D(pool_size = 2, strides = 2))
    model.add(Flatten())

```



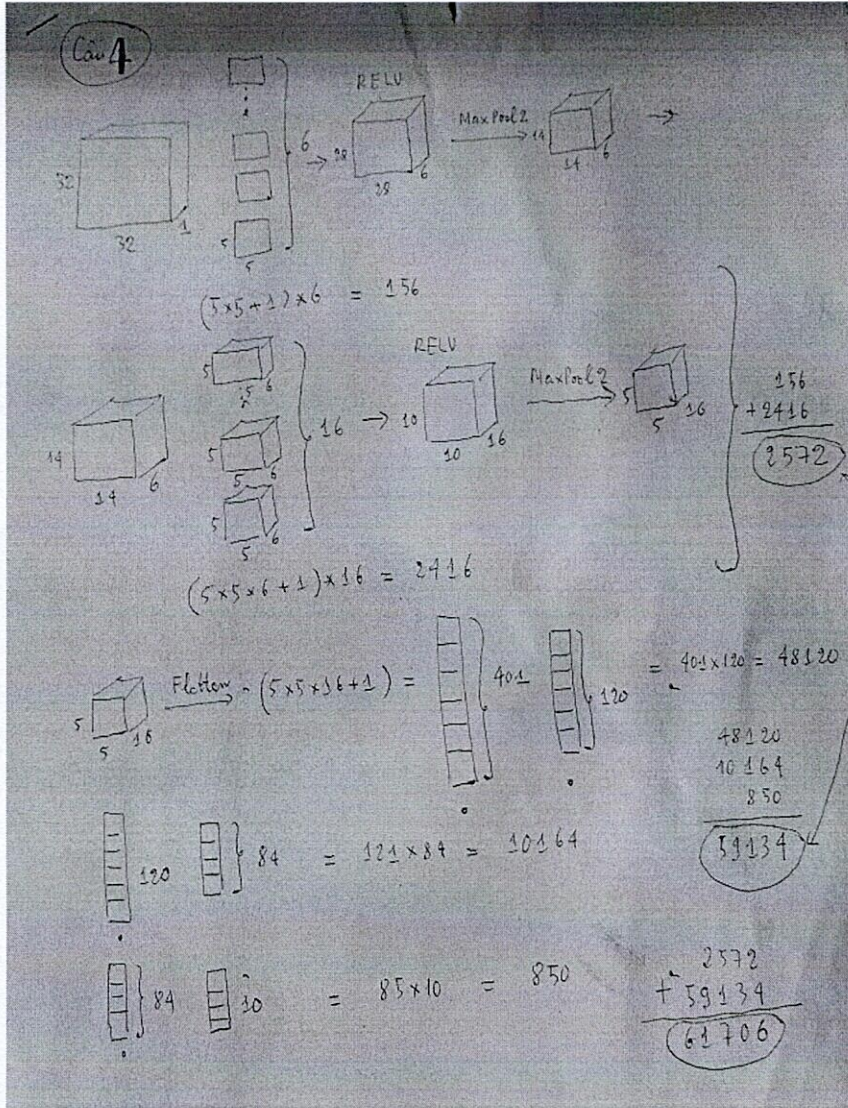
```

model.add(Dense(units = 120, activation = 'relu'))
model.add(Dense(units = 84, activation = 'relu'))
model.add(Dense(units = 10, activation = 'softmax'))

```

- Draw the network architecture diagram. (1 point)
- Determine the number of parameters of filters (include the bias node) in the convolutional layers. (1 point)
- Determine the number of weights of the fully connected layer. (1 point)

Đáp án:



Using TensorFlow backend.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 28, 28, 6)	156
max_pooling2d_1 (MaxPooling2)	(None, 14, 14, 6)	0


conv2d_2 (Conv2D)	(None, 10, 10, 16)	2416
max_pooling2d_2 (MaxPooling2)	(None, 5, 5, 16)	0
flatten_1 (Flatten)	(None, 400)	0
dense_1 (Dense)	(None, 120)	48120
dense_2 (Dense)	(None, 84)	10164
dense_3 (Dense)	(None, 10)	850

Total params: 61,706
Trainable params: 61,706
Non-trainable params: 0

-----HẾT-----

Ngày 3 tháng 2 năm 2021

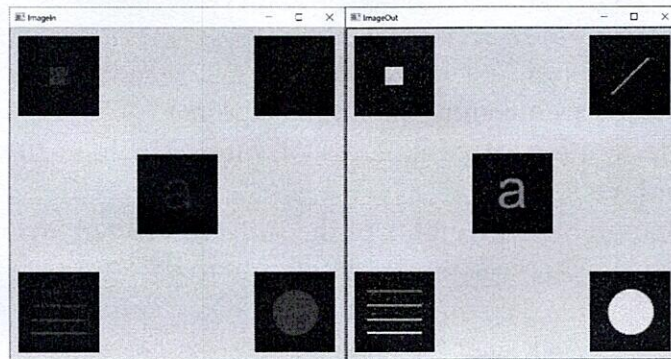
Thông qua bộ môn
(ký và ghi rõ họ tên)


Trần Tiến Đức

Câu 1: (2đ)

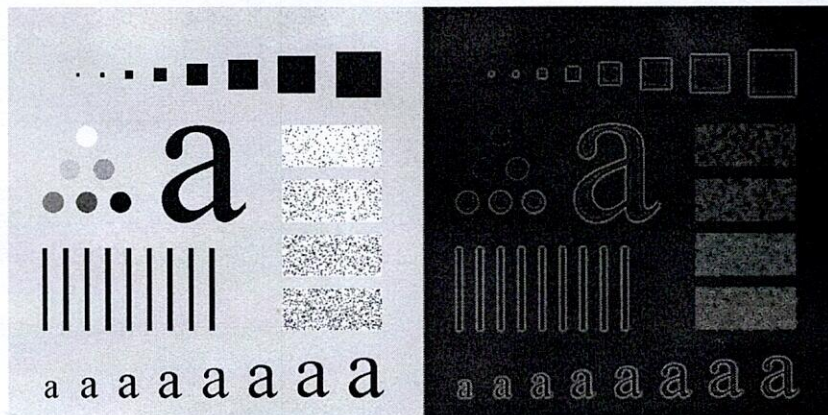
Xây dựng và cài đặt thuật toán làm rõ những đối tượng được nhúng trong năm hình vuông màu đen dùng thông kê histogram. (1.5đ)

Tại sao trong trường hợp này cân bằng histogram toàn cục lại không có kết quả? (0.5đ)



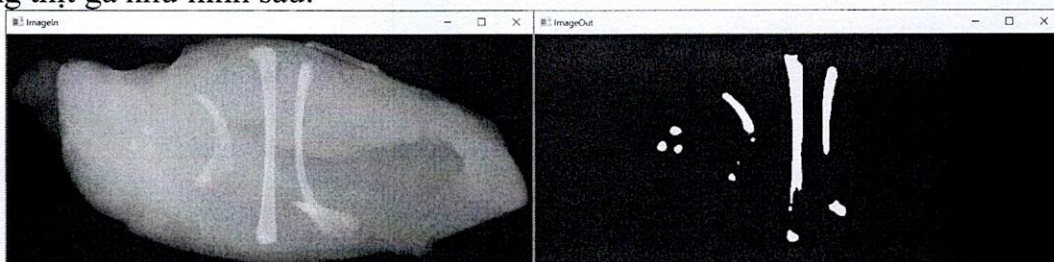
Câu 2: (3đ)

Xây dựng và cài đặt thuật toán lọc ảnh trong miền tần số dùng bộ lọc thông cao Butterworth có tần số cắt là $D_0 = 60$ và bậc của bộ lọc là $n = 2$ cho hình sau:



Câu 3: (2đ)

Xây dựng và cài đặt thuật toán đếm và tính các điểm ảnh của các mảnh xương trong miếng thịt gà như hình sau:



Câu 4: (3đ)

Implementing the LeNet-5 convolutional neural network architecture to classify MNIST Digits as following:

```
def LeNet_5():
    model = Sequential()
    model.add(Conv2D(filters = 6,
                    kernel_size = 5,
                    strides = 1,
                    activation = 'relu',
                    input_shape = (32,32,1)))
    model.add(MaxPooling2D(pool_size = 2, strides = 2))
    model.add(Conv2D(filters = 16,
                    kernel_size = 5,
                    strides = 1,
                    activation = 'relu',
                    input_shape = (14,14,6)))
    model.add(MaxPooling2D(pool_size = 2, strides = 2))
    model.add(Flatten())
    model.add(Dense(units = 120, activation = 'relu'))
    model.add(Dense(units = 84, activation = 'relu'))
    model.add(Dense(units = 10, activation = 'softmax'))
```

- Draw the network architecture diagram. (1 point)
- Determine the number of parameters of filters (include the bias node) in the convolutional layers. (1 point)
- Determine the number of weights of the fully connected layer. (1 point)

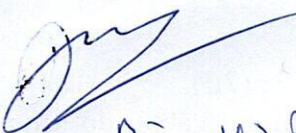
-----HẾT-----

Ghi chú: Cán bộ coi thi không giải thích đề thi.

Chuẩn đầu ra của học phần (về kiến thức)	Nội dung kiểm tra
[G 2.1]: <i>Nắm được các phương pháp xử lý ảnh cơ bản</i>	Câu 1, 2, 3, 4
[G 2.2]: <i>Tìm được phương pháp giải quyết một số bài toán xử lý ảnh bằng cách kết hợp một hoặc nhiều phương pháp</i>	Câu 3
[G 3]: <i>Hiểu được các thuật ngữ tiếng Anh cơ bản dùng trong xử lý ảnh</i>	Câu 4

Ngày 4 tháng 1 năm 2021

Thông qua bộ môn
(ký và ghi rõ họ tên)


Bùi Hà Đức