1

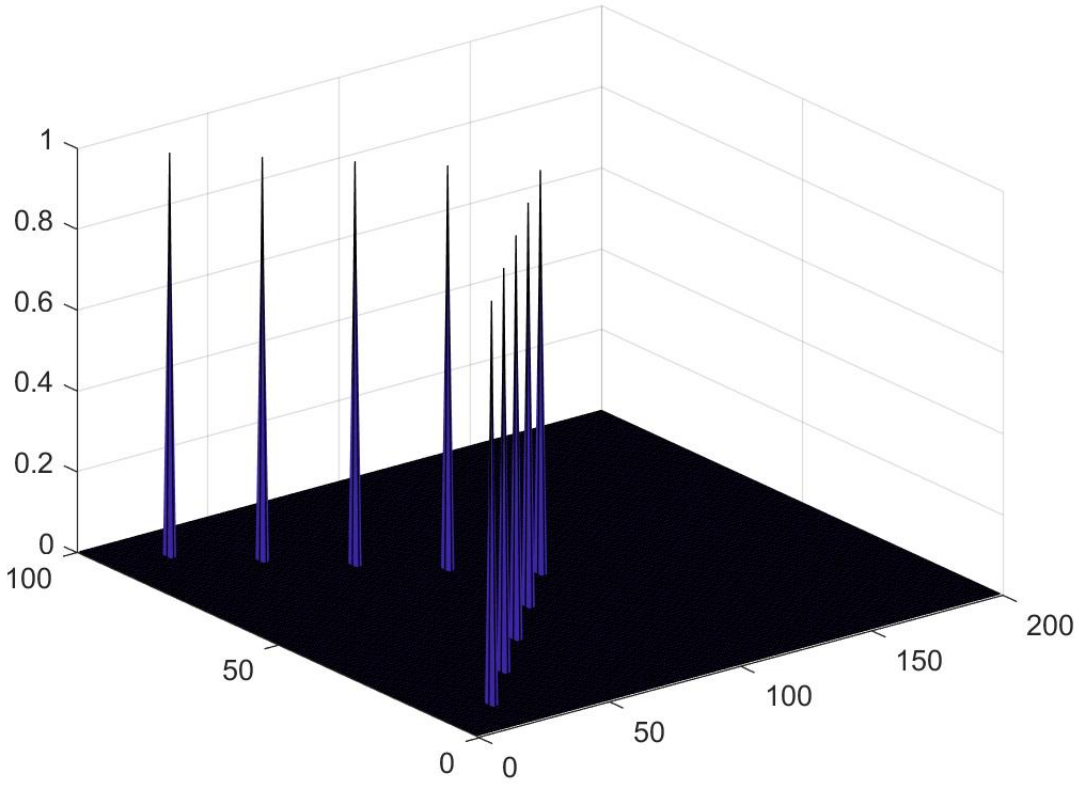
a.)

$$\delta(u,v) = \begin{cases} 1 & u = 0, v = 0 \\ 0 & u \neq 0, v \neq 0 \end{cases}$$

b.)

$$\delta(u,v) = \begin{cases} 1 & u = m, v = n \\ 0 & u \neq m, v \neq n \end{cases}$$

c.)



***Matlab Code* :**

```
image = zeros(100,200);
image(10,20) = 1;
image(20,40) = 1;
image(30,60) = 1;
image(40,80) = 1;
```

```
image(50,100) =1;
image(60,80) = 1;
image(70,60) =1;
image(80,40) = 1;
image(90,20) =1;
surf(image);
```

d.)

$f[m,n] = \sum_{i=0}^{M} \sum_{j=0}^{N} f(i,j) \cdot \delta(m-i, n-j)$

2

a.)

For a $m * m$ filter, the cost of computing a convolution of single pixel is m^2. Therefore, it needs $m^2 n^2$ operations to compute convolution for a $n * n$ image.

b.)

For a separable filter, it cost $2m$ operations to compute convolution for a single pixel. Therefore, it total cost $2mn^2$ operations to compute convolution for $n * n$ image.

c.)

For F_1, it is not separable.

For F_2, it is separable. The horizontal filter is $\begin{bmatrix}\text{-2.4108} & \text{-1.2054} & \text{-2.4108}\end{bmatrix}$ The vertical filter is $\begin{bmatrix}\text{-2.4888} \\ \text{-0.8296} \\ \text{-2.4888}\end{bmatrix}$

3

For a.) - f.)v. Please see the attached Matlab code and the solution figure.

f.)v.

The threshold T that I chose at the end is 0.68. The following figure show the result. The number of correctly labeled digits is 13.

f.)vi.

Cropping your own templates can improve performance, because the own cropping template is more similar with digits. Therefore, the result of correlation will be better.

As the figure shows above, I crop digit number 1 into scale_3 folder. Compared to the figure that in the Attached solution figure. Both of them are using threshold 6.8. It is easy to see that the performance is been improved, because the digit number 1 is able to been detected now.

Qestion3 Matlab code

The following pages are the solutions from a.) - f.)v. The first page is about function isLocalMaximum(x, y, thisCorr), which is in f.)iii. The rest of page is the main Matlab code and final solution figure of Qestion 3.

```
function isMax = isLocalMaximum(x, y, thisCorr)

temp_matrix = [thisCorr(x-1,y-1) thisCorr(x-1,y) thisCorr(x-1,y+1);
               thisCorr(x,y-1) thisCorr(x,y) thisCorr(x,y+1);
               thisCorr(x+1,y-1) thisCorr(x+1,y) thisCorr(x+1,y+1)];

max_col = max(temp_matrix);
isMax = (thisCorr(x,y) == max(max_col));

end
```

```
Not enough input arguments.

Error in isLocalMaximum (line 3)
temp_matrix = [thisCorr(x-1,y-1) thisCorr(x-1,y) thisCorr(x-1,y+1);
```

Published with MATLAB® R2017b

```
%(b)Read in the templates.
[templates, dimensions] = readInTemplates();

%(c)Compute normalized correlation
image = rgb2gray(imread('thermometer.png'));
iamge = double(image);
[m, n] = size(image);
corrArray = zeros(m,n,30);
for s = 1:30
    templates{s} = rgb2gray(templates{s});
    temp_output = normxcorr2(templates{s},image);
    offSetX = round(dimensions(s).width/2);
    offSetY = round(dimensions(s).height/2);
    corrArray(:,:,s) = temp_output((offSetY + 1):...
   (end- dimensions(s).height+ offSetY + 1), (offSetX + 1):...
   (end- dimensions(s).width + offSetX + 1));
end

%(d)find the maximum along the third dimension
[maxCorr, maxIdx] = max(corrArray,[],3);

%(f)i.
candX = [];
candY = [];
for x =1:m
    for y = 1:n
        %(e)Find the pixels for which maxCorr exceeds threshold T=0.68
        if maxCorr(x,y) < 0.68
            maxCorr(x,y) = 0;
        else
            %(f)i.Let's call the coordinates of candidate pixels
 (output
            % of Step-e)as candX, candY.
            candX(length(candX)+1) = x;
            candY(length(candY)+1) = y;
        end
    end
end
[h,w] = size(candX);
imshow('thermometer.png');
for t = 1:w
    %(f)ii.Extract the correlation matrix for the above templateIndex
 i.e.
    % thisCorr = corrArray
    thisCorr = corrArray(:,:,maxIdx(candX(t),candY(t)));
    %(f)iii. write a function isLocalMaximum
    isMax = isLocalMaximum(candX(t),candY(t),thisCorr);
    if isMax == 1
        drawAndLabelBox( candY(t),candX(t),...
        maxIdx(candX(t),candY(t)) ,dimensions);
        drawnow;
    end
```

```
end
```

Warning: Image is too big to fit on screen; displaying at 67%

Published with MATLAB® R2017b