HW 4 - ME 6406 Machine Vision

Table of Contents

Prob 1a: Pose Estimation [1] 1
Prob 1b: Pose Estimation [1] 2
Problem 2a: Artificial Neural Network 4
Problem 2b: Artificial Neural Network 4
Problem 2c: Artificial Neural Network 4
Problem 2c: Artificial Neural Network 4
Prob 3 part 1a: Artificial Color Contrast (ACC) 5
Prob 3 part 1b: Artificial Color Contrast (ACC) 5
Prob 3 part 2a: Color-based Image Segmentation 6
Prob 3 part 2b: Color-based Image Segmentation 8
Prob 3 part 2c: Color-based Image Segmentation 8
Prob 3 part 3ab: Principle component analysis (PCA): 9
Prob 3 part 3c: Principle component analysis (PCA): 10
functions 12

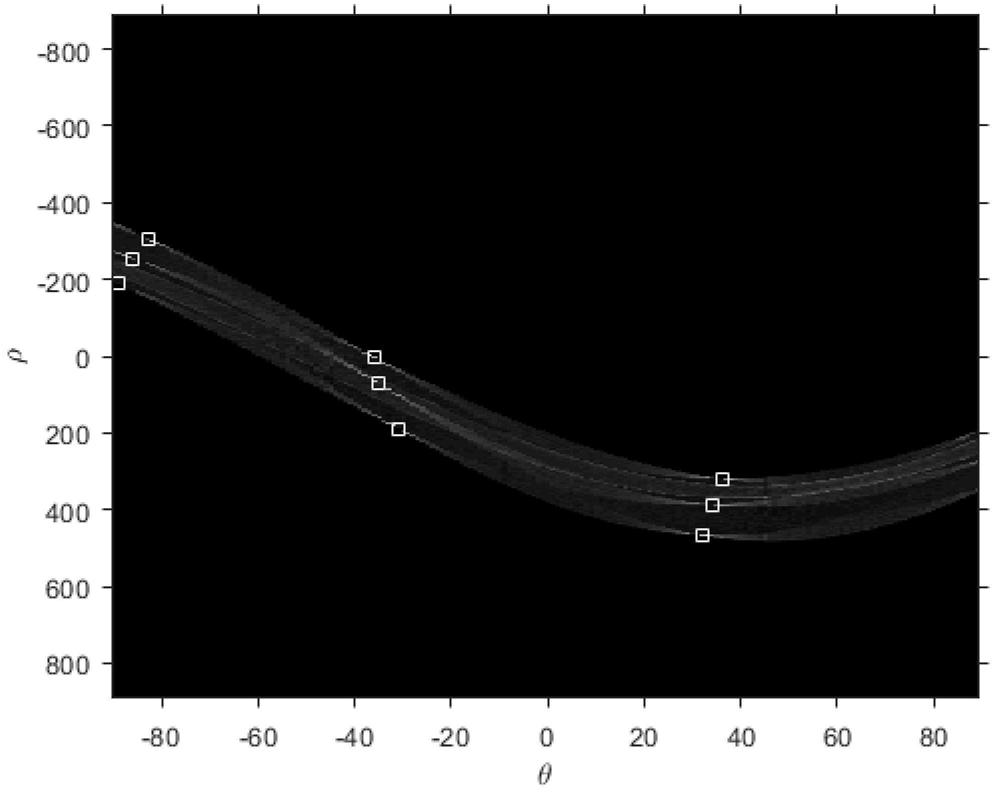
by Cody Houff 11/17/21

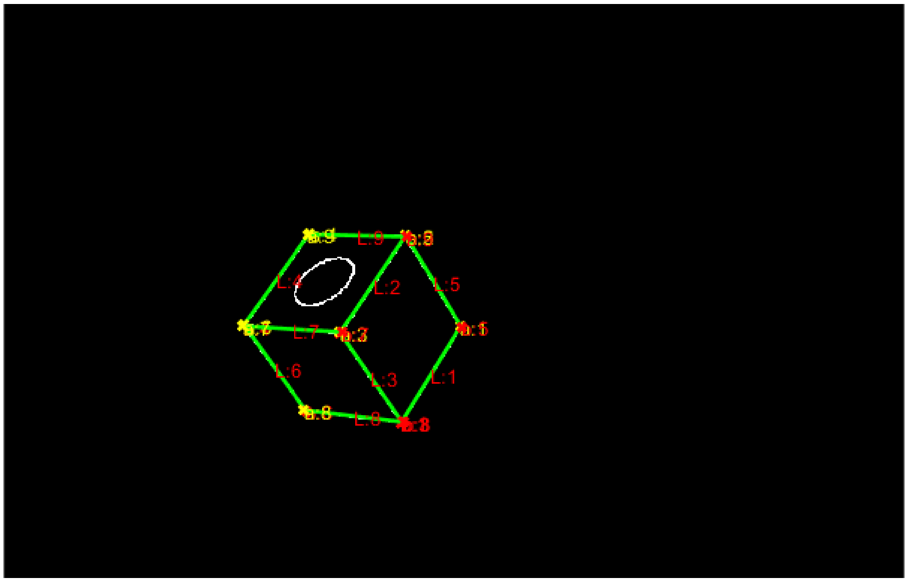
Prob 1a: Pose Estimation [1]

Hough Transform 9 straight edges in ‘block.png’. Find the 9 normal vectors n=(af,bf,c)^T/|(af,bf,c)^T|

```
n_array_abc =

    0.8508    0.5252   -0.0158
    0.8297    0.5565    0.0446
   -0.8195    0.5682   -0.0742
    0.8087    0.5806    0.0949
   -0.8535    0.5210   -0.0077
   -0.8048    0.5813   -0.1198
   -0.0601    0.9977   -0.0308
   -0.1192    0.9891   -0.0864
   -0.0241    0.9992    0.0334
```





Prob 1b: Pose Estimation [1]

[R] and T

```
V =

    0.4515
   -0.7845
   -0.4245
   -0.7246
   -0.0368
   -0.6886
    0.4992
    0.6580
   -0.5844
   -3.8929
   10.0000
  119.7974


r_matrix =

    0.4515   -0.7845   -0.4245
   -0.7246   -0.0368   -0.6886
    0.4992    0.6580   -0.5844


T_array =

   -3.8929
   10.0000
  119.7974


ans =

    'check Tz ~ 120'


ans =

  119.7974


ans =

    'check inv(T) ~ transpose(T)'


inverse_r =

    0.4695   -0.7300    0.5190
   -0.7590   -0.0514    0.6119
   -0.4536   -0.6814   -0.5788


transpose_r =
```

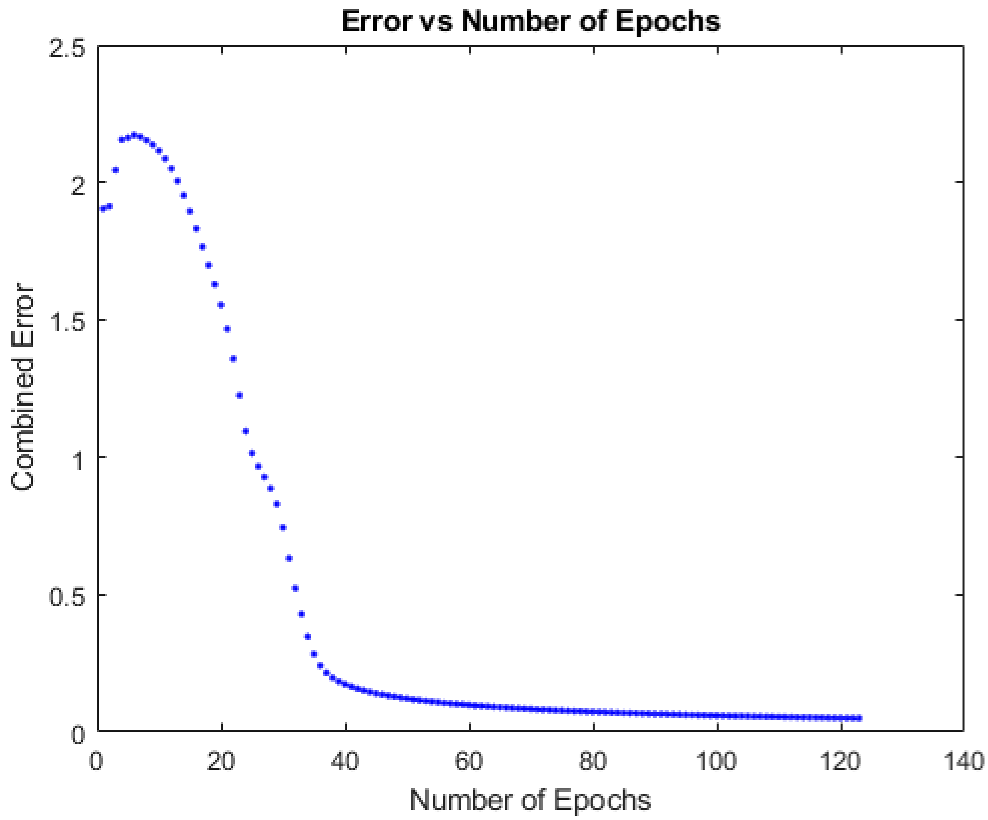
```
 0.4515   -0.7246    0.4992
-0.7845   -0.0368    0.6580
-0.4245   -0.6886   -0.5844
```

Problem 2a: Artificial Neural Network

Design an ANN to recognize O and T on a binary 4x4 square grid.

Problem 2b: Artificial Neural Network

Problem 2c: Artificial Neural Network



Problem 2c: Artificial Neural Network

Test the data in Fig 2(b) by reading the weights of nodes in your “.mat” file. Show the output values and results.

```
test # 1
O 0.93742 confidence
T 0.061495 confidence

test # 2
```

O 0.046533 confidence
T 0.95433 confidence

test # 3
O 0.7214 confidence
T 0.27313 confidence

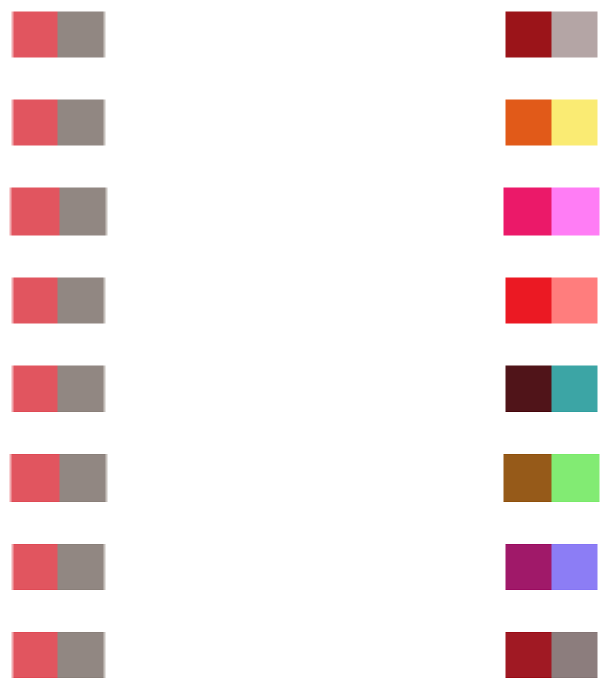
Prob 3 part 1a: Artificial Color Contrast (ACC)

Derive the following equations with fk(x,y) = +(R-G) and +(R+G-B)

Prob 3 part 1b: Artificial Color Contrast (ACC)

Perform the ACC transformation (#c=1, #s=10) on sample

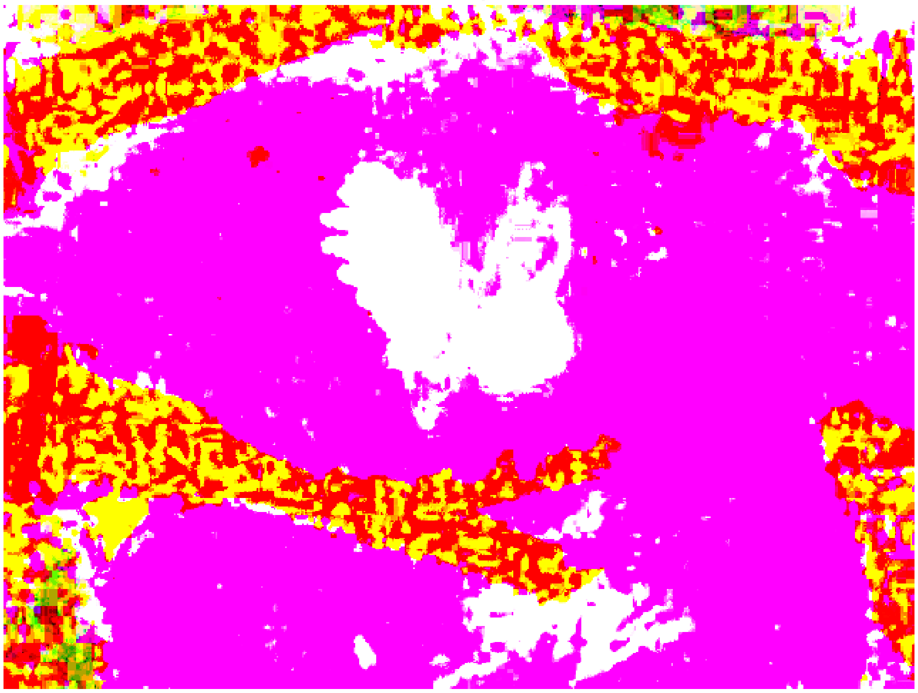
color patterns (100×100 each) with the following combinations:
1-2-3, 1-2-c, 1-b-3, 1-b-c, a-2-3, a-2-c, a-b-3, a-b-c



Prob 3 part 2a: Color-based Image Segmentation

Step 1. Transfer pixels from RGB to Lab color system original

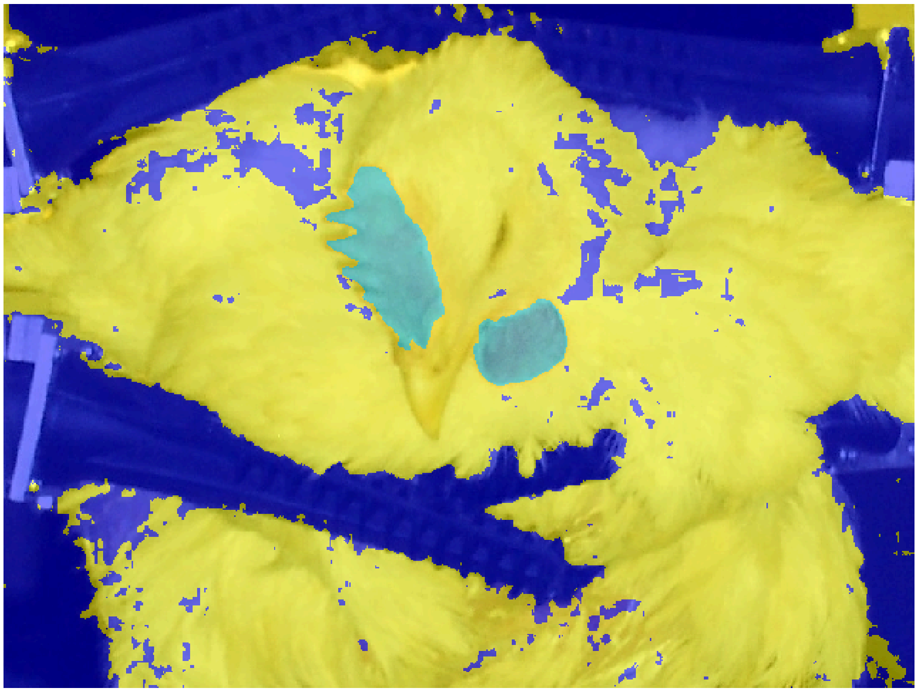
lab color system



Prob 3 part 2b: Color-based Image Segmentation

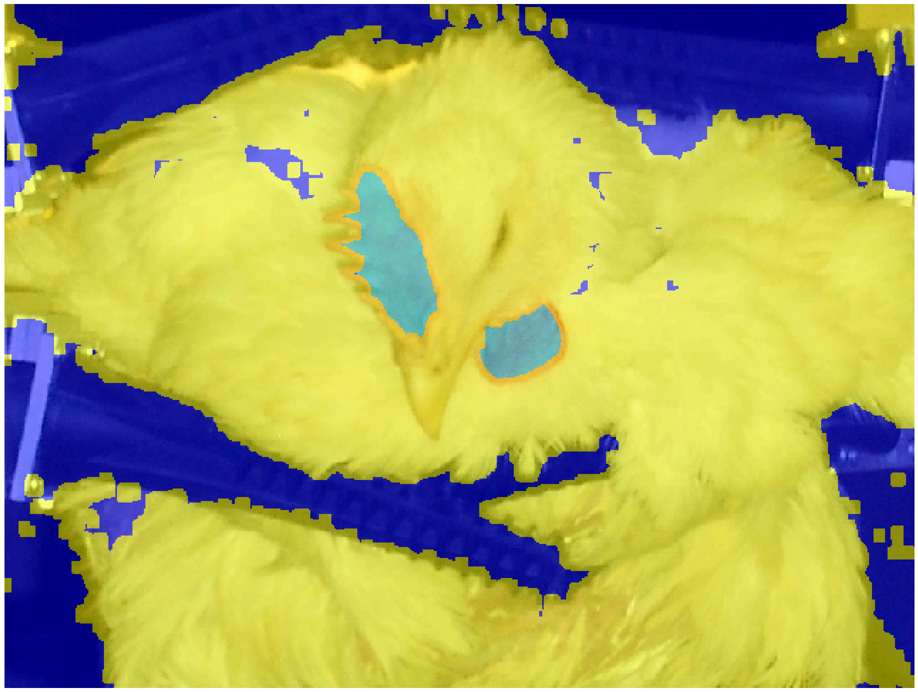
Step 2. Apply k-means clustering on data in a-b domain with cluster number (k=2)

Clustered image with k=3



Prob 3 part 2c: Color-based Image Segmentation

Step 3. Erode the segment image to filter out small fragments.



Prob 3 part 3ab: Principle component analysis (PCA):

```
a. Determine the covariance matrix of data.
b. Derive the components (eigenvectors) with eigenvalues arranged
in a descending order.
```

```
eigen_vectors2 =

    0.6109    0.7720    0.7423
    0.5973   -0.3038   -0.1756
    0.5197   -0.5584   -0.6467


ans =

    'check conv_matrix & eigen_vectors & eigen_values1'


conv_matrix =

   1.0e+03 *

    5.3589    5.0279    4.3272
    5.0279    5.0238    4.3544
```

```
    4.3272    4.3544    3.8641


eigen_vectors1 =

    0.7423    0.7720    0.6109
   -0.1756   -0.3038    0.5973
   -0.6467   -0.5584    0.5197


eigen_values1 =

   1.0e+04 *

    1.3955
    0.0251
    0.0041
```

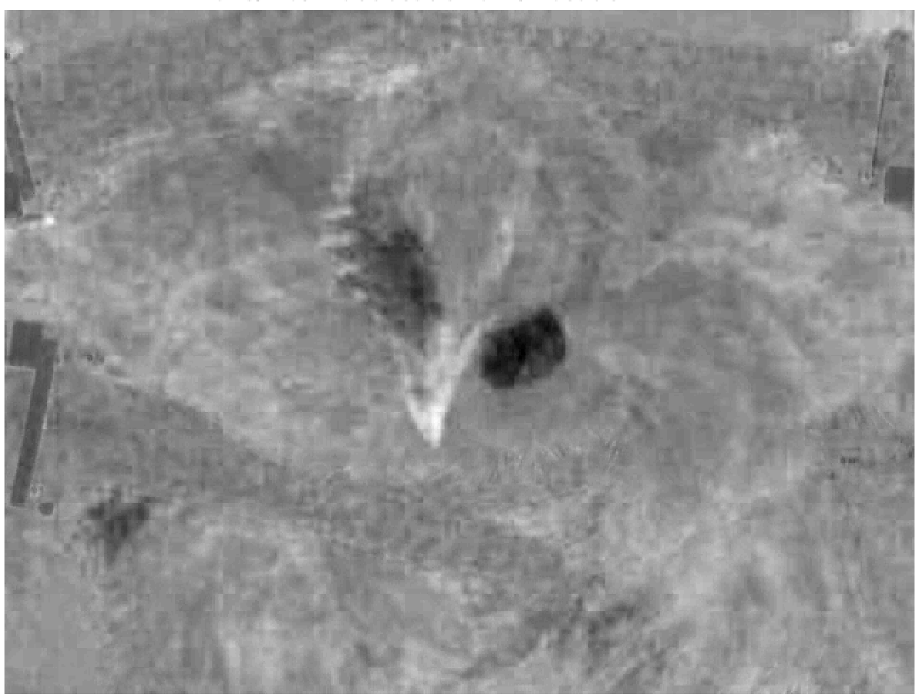
Prob 3 part 3c: Principle component analysis (PCA):

Obtain the maximum and minimum values of three component matrices. Show these three matrices(images) with linear mapping from the minimum and maximum values to the range of (0-255).

uint8 matrix red double max: 441 double min: 0

uint8 matrix green double max: 106 double min: -53

uint8 matrix blue double max: 31 double min: -47



functions

Published with MATLAB® R2021a

2. a) (also in code)

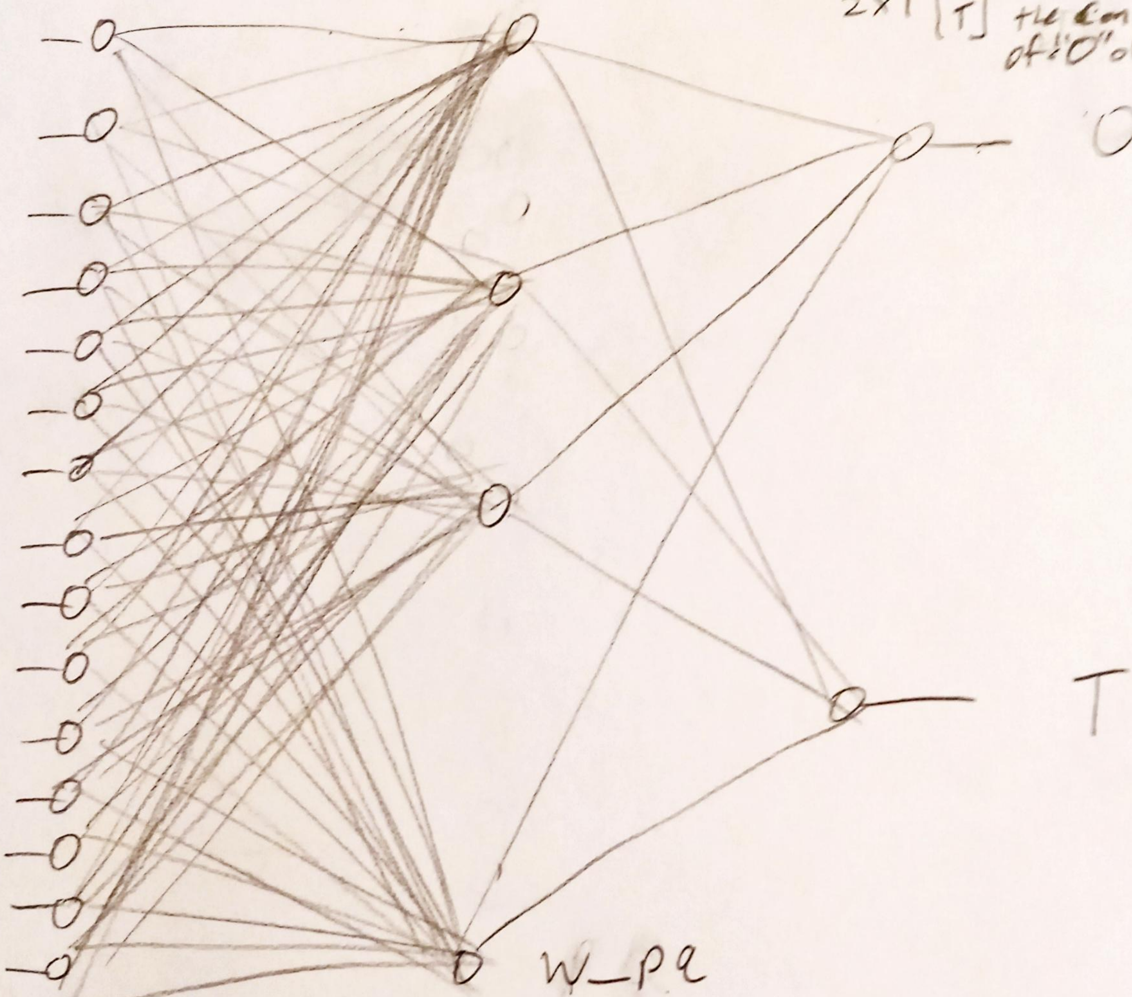
16 inputs 0 or 1

K Layer
Input Layer
16 nodes

P layer
Hidden Layer
4 nodes

Q Layer
output Layer
2 nodes

$2\times1 \begin{bmatrix} O \\ T \end{bmatrix}$ output is matrix showing the confidence of "O" or "T"



W_pk
K-P weights
16x4 matrix

P-Q weights
4x2 matrix

$DOG = G_{\sigma c} - G_{\sigma s}$

machine vision

Prob 3a)

h_1 : $h_1(x,y) = G_{\sigma c} \cdot f_1(x,y) - G_{\sigma s} \cdot (R-G)$

$h_1(x,y) = G_{\sigma c} \cdot R - G_{\sigma s} R + G_{\sigma s} G$

$h_1(x,y) = [G_{\sigma c} - G_{\sigma s}] R + G_{\sigma s} \cdot G$

$\boxed{h_1(x,y) = DOG \cdot R + G_{\sigma s} \cdot G}$

170-170+110

h_2 : $h_2(x,y) = G_{\sigma c} \cdot f_2(x,y) - G_{\sigma s}(R-G)$

$h_2(x,y) = G_{\sigma c} \cdot G - G_{\sigma s} R + G_{\sigma s} \cdot G + (G_{\sigma s} \cdot G - G_{\sigma s} \cdot G)$

110-170+110

$\boxed{h_2(x,y)\ \underbrace{(G_{\sigma c} - G_{\sigma s})}_{DoG} G + G_{\sigma s}[2G - R]}$

h_3 : $h_3(x,y) = G_{\sigma c} \cdot f_3(x,y) - G_{\sigma s}(R-G)$

+B -B

$h_3(x,y) = G_{\sigma c} \cdot B - G_{\sigma s} R + G_{\sigma s} G + (G_{\sigma s} \cdot B - G_{\sigma s} \cdot B)$

$\boxed{h_3(x,y) = \underbrace{(G_{\sigma c} - G_{\sigma s})}_{DoG} B + G_{\sigma s}[B - (R-G)]}$

---

h_a : $h_a(x,y) = G_{\sigma c} \cdot f_1(x,y) - G_{\sigma s}(R+G-B)$

$h_a(x,y) = G_{\sigma c} \cdot R - G_{\sigma s} R - G_{\sigma s} G + G_{\sigma s} B$

$\boxed{h_a(x,y) = \underbrace{(G_{\sigma c} - G_{\sigma s})}_{DoG} R + G_{\sigma s}[B-G]}$

h_b : $h_b(x,y) = G_{\sigma c} \cdot f_2(x,y) - G_{\sigma s}(R+G-B)$

$h_b(x,y) = G_{\sigma c} \cdot G - G_{\sigma s} R - G_{\sigma s} G + G_{\sigma s} B$

$\boxed{h_b(x,y) = \underbrace{[G_{\sigma c} - G_{\sigma s}]}_{DoG} G + G_{\sigma s}[B-R]}$

+B -B

h_c : $h_c(x,y) = G_{\sigma c} \cdot f_3(x,y) - G_{\sigma s}(R+G-B)$

$h_c(x,y) = G_{\sigma c} \cdot B - G_{\sigma s} R - G_{\sigma s} G + G_{\sigma s} B + (G_{\sigma s} B - G_{\sigma s} B)$

$\boxed{h_c(x,y) = \underbrace{[G_{\sigma c} - G_{\sigma s}]}_{DoG} B + G_{\sigma s}[2B - (R+G)]}$

4b) $n(x) = \frac{1}{1+e^{-x}}$

$w_{pq}^{(k+1)} = w_{qp}^{(k)} + \Delta w_{qp} \quad \Delta w_{qp} = -\eta \cdot \frac{\delta E_{qp}}{\delta w_{qp}}$

$\frac{\delta E_q}{\delta w_{qp}} = \frac{\delta E_q}{\delta I_q} \cdot \frac{\delta I_q}{\delta w_{qp}} \quad \frac{\delta I}{\delta w_{qp}} = \frac{\delta}{\delta w_{qp}} \sum_{p=1}^{N_p} (w_{qp} O_p) = O_p$

$\frac{\delta E_q}{\delta I_q} = -\delta_q \quad -\frac{\delta E_q}{\delta I_q} = \delta_q = -\frac{\delta E_q}{\delta O_q} \cdot \frac{\delta O_q}{\delta I_q}$

$\frac{\delta E_q}{\delta O_q} = \frac{\delta}{\delta O_q} \left[\frac{1}{2} \sum_{p=1}^{N_q} (r_q - O_q)^2 \right] = -(r_q - O_q)$

$\frac{\delta O_q}{\delta I_q} = \frac{\delta}{\delta I_q} h_q(I_q) = O_q(1-O_q) \quad \delta_q (r_q - O_q)(O_q - O_q^2) = -\frac{\delta E_q}{\delta I_q}$

$\Delta w_{qp} = -\eta \cdot \frac{\delta E}{\delta w_{qp}} = \eta \cdot \delta_q O_p = \eta (r_q - O_p)(O_q - O_q^2) O_p$

$$\boxed{w_{qp}^{(k+1)} = w_{qp}^{(k)} + \eta \cdot (r_q - O_q) \cdot (O_q - O_q)^2}$$