Digital Image Processing (261453)

Computer Assignment 3

วรธรรม ฆังตระกูล 570610597

**บทนำ**

ในการตรวจข้อสอบกากบาท นักศึกษาได้พัฒนาโปรแกรมขึ้นมาเพื่อวิเคราะห์รูปภาพ โดยมีข้อมูลเบื้องต้นคือรูปแบบกระดาษและจุดบอกตำแหน่งสองจุด วิธีการที่นักศึกษาใช้คือเมื่อป้อนรูปภาพข้อสอบเข้ามา โปรแกรมจะตรวจหาจุดสองจุด แล้วคำนวณระยะห่างและมุม จากนั้นนำไปเปรียบเทียบกับต้นฉบับเพื่อปรับให้ตรงกัน เมื่อตรงกันแล้วนักศึกษานำภาพต้นฉบับมาลบออกจากภาพข้อสอบเพื่อให้เหลือแค่รอยกากบาท จากนั้นตรวจหาจุดกลางของกากบาท เมื่อทราบแล้วก็ตรวจสอบว่าอยู่คอลัมน์ไหน เมื่อได้แล้วตรวจสอบว่าอยู่แถวไหนและข้อไหน จากนั้นรวบรวมผลลัพธ์

**วิธีการ**

1. ตรวจหาจุด

นักศึกษาได้ทำการ crop ภาพจุดดำจากรูปต้นฉบับ จากนั้นนำไปคิด cross-correlation กับรูปที่ป้อนมา ค่า max ของ correlation คือจุดดำหนึ่งในสองจุด นำมาย้อนหาตำแหน่งจุดก็จะได้มา จากนั้นนักศึกษาลบจุดนั้นออกจาก correlation จะได้ค่า max ที่สองที่บอกตำแหน่งจุดที่สอง เมื่อได้ทั้งสองจุดแล้วนักศึกษาคำนวณขนาดส่วนของเส้นตรงและมุมระหว่างมัน ได้แล้วนำมาเปรียบเทียบกับต้นฉบับ จากนั้นหมุน ปรับขนาด และเลื่อนรูปที่ป้อนเข้ามาเพื่อให้จุดดำทับของรูปต้นฉบับพอดี

1. ลบกรอบ

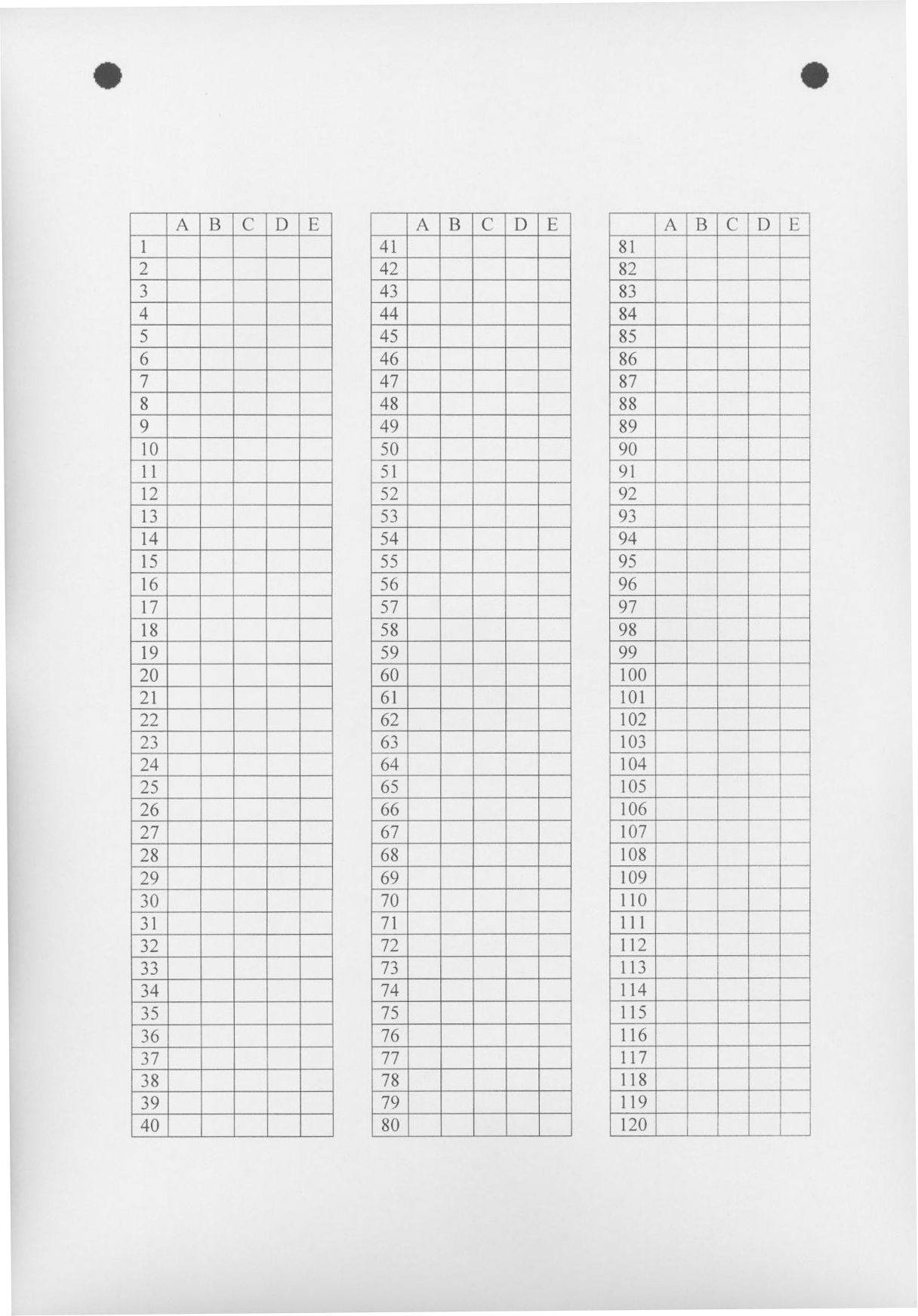
นักศึกษา dilate รูปต้นฉบับเพื่อให้เส้นหนาขึ้น จากนั้นลบออกจากรูปที่ได้มาหลังจากปรับขนาด เท่านี้จะได้รูปที่เหลือแต่กากบาท ที่นักศึกษาทำตรงนี้เพราะในรูปต้นฉบับมีตัวอักษรอยู่และอาจรบกวนขั้นตอนต่อไปได้

1. ตรวจหากากบาท

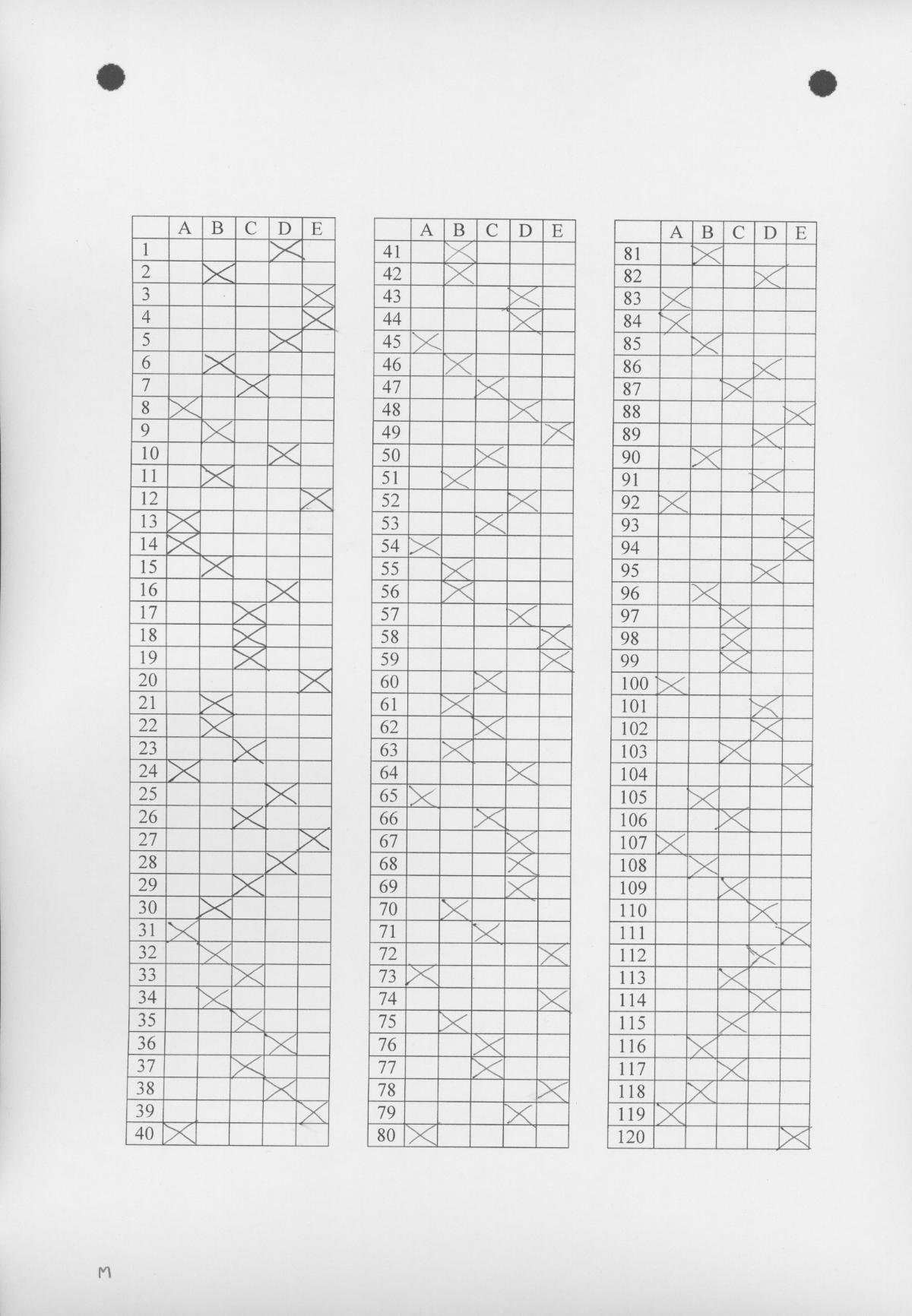
นักศึกษา crop ภาพกากบาทจากรูปตัวอย่าง โดย crop มาแค่ส่วนกลางเพราะเป็นส่วนที่คงที่มากที่สุด จากนั้นนักศึกษาได้ลองสองวิธีการด้วยกัน วิธีแรกคือเปลี่ยนภาพกากบาทนี้ให้เป็นแบบ binary แล้วใช้เป็น neighborhood กับ median filter วิธีที่สองคือนำไปคำนวณ cross-correlation กับรูปที่ป้อนมา พบว่าวิธีที่สองดีกว่า เพราะรูปที่ได้นำไปแยกออกได้ง่ายกว่า

1. นับข้อ

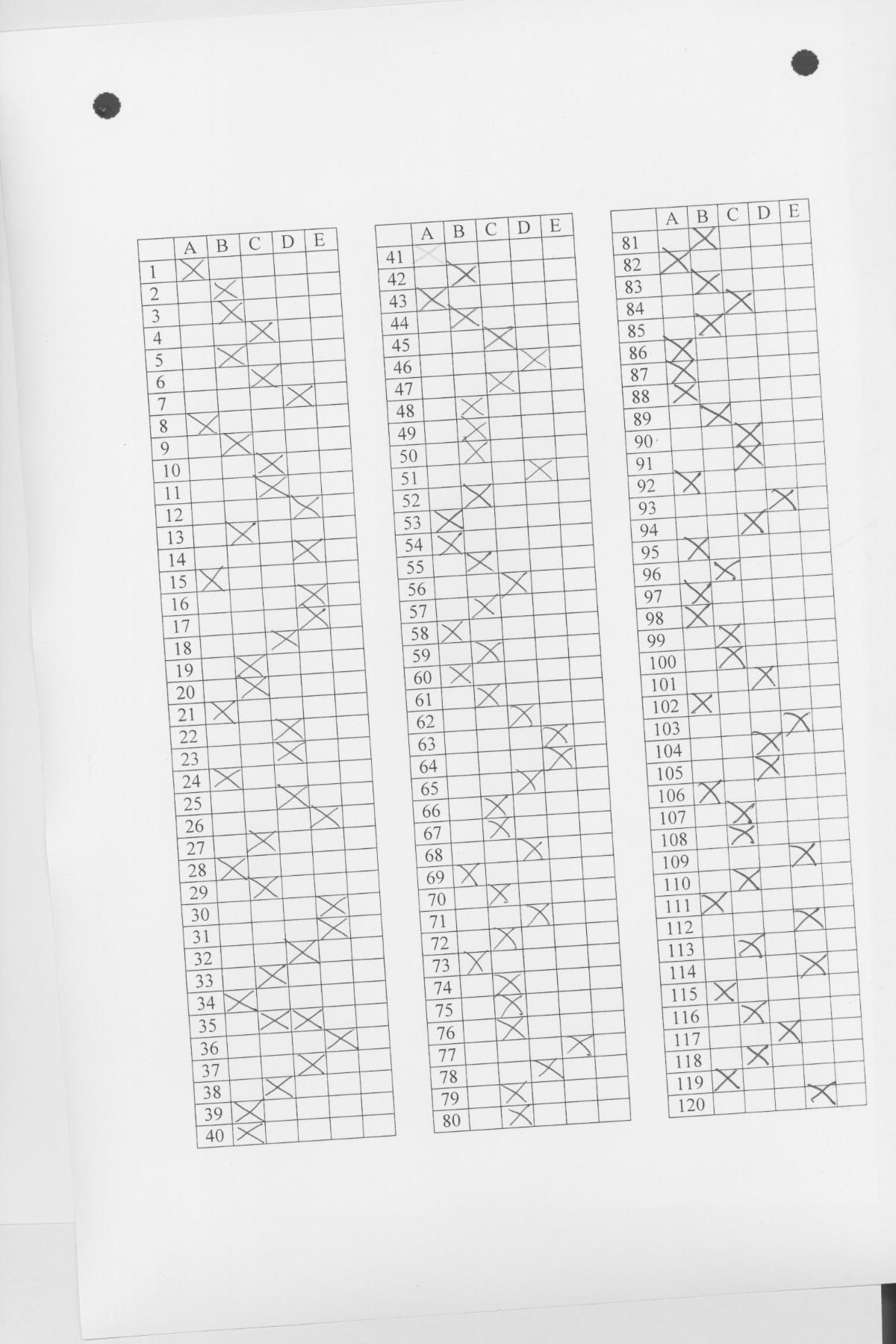
จากการที่นักศึกษามีรูปต้นฉบับ จึงไม่เป็นเรื่องยากที่จะเขียนโปรแกรมวนตามคอลัมน์ วนตามแถว แล้วนับข้อ อย่างไรก็ตามหากมีการเปลี่ยนรูปแบบกระดาษคำตอบ โปรแกรมส่วนนี้ต้องแก้ไขเอง ซึ่งเป็นสิ่งที่ไม่พึงประสงค์นัก



รูปต้นฉบับ

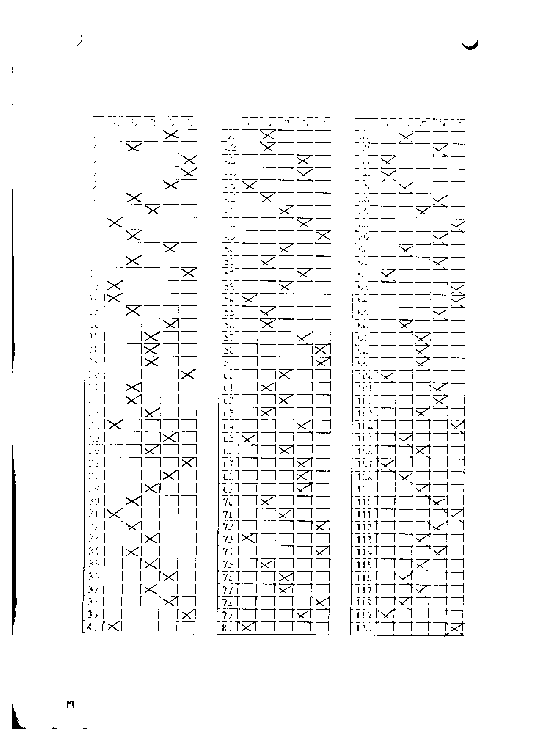


รูปทดสอบ 1



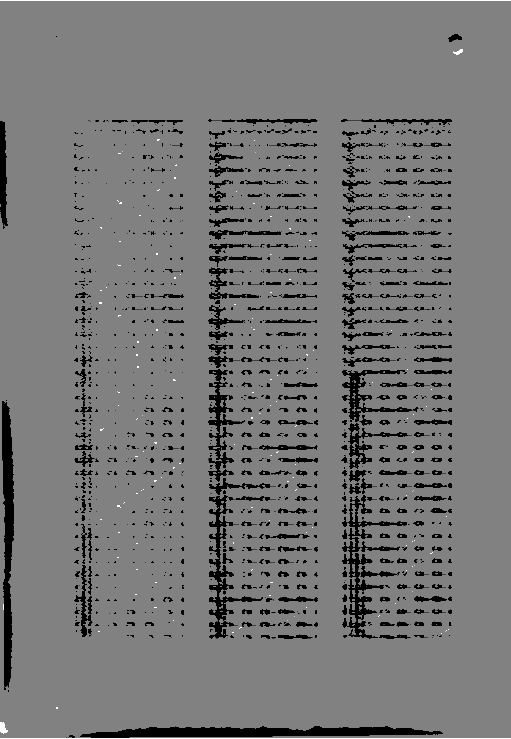
รูปทดสอบ 2**ผลการทดลอง**

ในขั้นตอนการลบกรอบ ผลการทดลองไม่เป็นไปตามที่คาดหวัง เพราะโปรแกรมตรวจหาจุดดำทั้งสองจุดคลาดเคลื่อน ทำให้ลบกรอบออกไปไม่หมด มิหนำซ้ำยังไปลบกากบาทออกบางส่วน อย่างไรก็ตามหากไม่ถูกลบตรงกลางโปรแกรมยังคงทำงานต่อไปตามที่หวังไว้



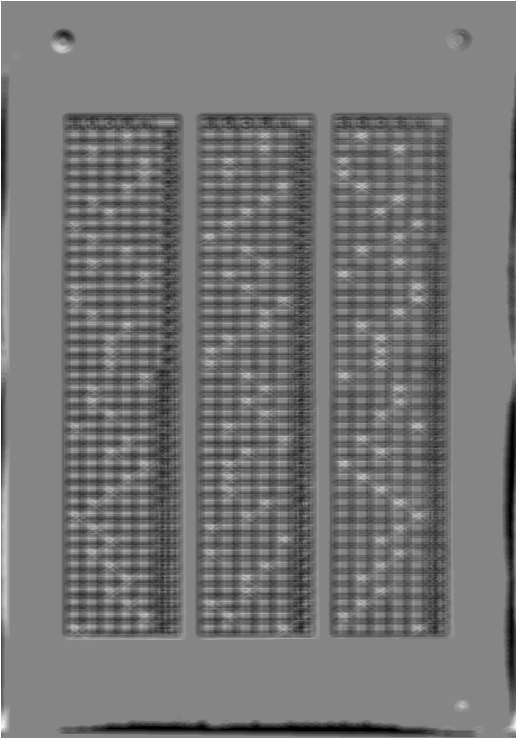
รูปที่ 1. ลบกรอบออกไม่หมด

ในส่วนการตรวจหากากบาท นักศึกษาได้ทดลองแบบ median filter ก่อน ได้ผลลัพธ์ดังรูปที่ 2 สังเกตได้ว่าในบางช่องกากบาทจะเหลือเป็นรอยขาดๆ ซึ่งไม่เป็นผลดีกับการนำไปตรวจสอบต่อ เพราะยากและโอกาสผิดพลาดสูง



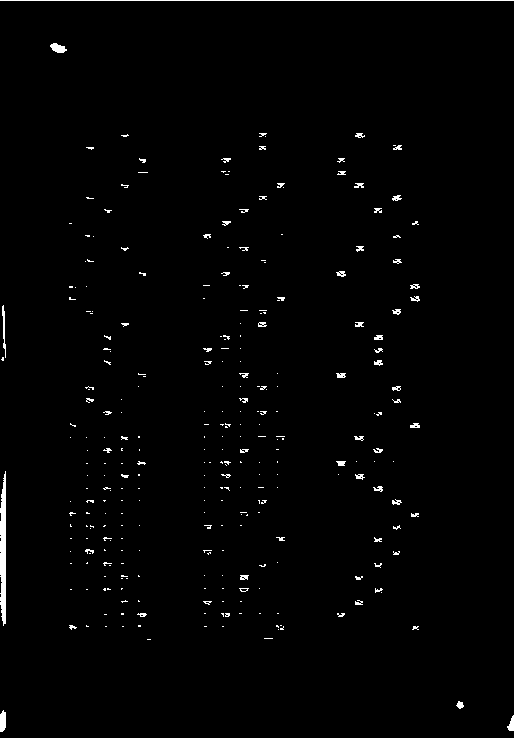
รูปที่ 2. ตรวจหากากบาทโดยใช้ median filter

วิธีทีที่สองที่ใช้คือ cross-correlation กับรูปกากบาท โดยได้ผลลัพธ์ดังรูปที่ 3 สังเกตได้ว่าจุดที่มีกากบาทจะเป็นจุดขาวที่มีความหนา ซึ่งเหมาะสมแก่การนำไปทำขั้นตอนต่อไป



รูปที่ 3. ตรวจหากากบาทโดยใช้ cross-correlation

ในการนับข้อ เพื่อให้ง่ายนักศึกษาได้แปลงรูปให้เป็น binary จากนั้นก็นับข้อตามรูปแบบกระดาษ คือไล่ตามคอลัมน์ ไล่ตามแถว แล้วดูว่าตำแหน่งแกน x ของจุดสีขาวควรจัดเป็นข้ออะไร



รูปที่ 4. รูป binary สำหรับนับข้อ

**สนทนา**

วิธีการหนึ่งที่นำมาช่วยในการนับข้อกากบาทได้คือการเก็บรูปแบบกากบาท บางค้นอาจกาตวัด บางคนกาแข็ง โดยนักศึกษาสามารถเก็บข้อมูล นำมาจัดแบ่งให้เป็นหมวดเหมือนแบ่งลายมือ แล้วจัดเก็บเป็น kernel bank เพื่อนำไปใช้ วิธีนี้จำช่วยให้รองรับรูปแบบกากบาทได้มากขึ้นและลดหย่อนว่าผู้สอบต้องกาให้สมบูรณ์

จากการทำการบ้านครั้งนี้ทำให้นักศึกษาได้เห็นว่าข้อสอบที่ฝนเป็นวงกลมนั้นสร้างความสะดวกสบายให้แก่ผู้ตรวจ ผู้ดูแลระบบเป็นอย่างมาก อีกทั้งลดโอกาสความผิดพลาด ลดภาระที่ต้องใช้คนมาตรวจสอบ

**สรุป**

การนำ image processing มาใช้ตรวจข้อสอบกากบาทมีข้อดีคือผู้ตรวจไม่ต้องคอยตรวจเองทีละข้อ ทุกๆคน แต่ใช้โปรแกรมมาช่วยตรวจแทน อย่างไรก็ตามต้องระวังถึงโอกาสผิดพลาด เพราะกากบาทเหมือนกับลายมือ แต่ละคนมีไม่เหมือนกัน การที่จะเขียนโปรแกรมเพื่อรองรับลายมือให้ครบถ้วนนั้นลำบาก วิธีการหนึ่งคือใช้ machine learning เข้ามาช่วย ทำให้ผู้พัฒนาไม่จำเป็นต้องมาคอยปรับแต่งโปรแกรมทุกๆครั้ง

ข้อจำกัดของโปรแกรมที่นักศึกษาเขียนคือจำเป็นต้องรู้รูปแบบกระดาษล่วงหน้า โดยจะเปลี่ยนแปลงไปนิดเดียวไม่ได้ และกากบาทควรมีความสมบูรณ์ระดับหนึ่ง แต่หากนักศึกษาแก้ไขให้โปรแกรมสามารถลบกรอบออกไปได้สมบูรณ์ ถึงกากบาทจะบิดเบี้ยวแค่ไหน ขอให้ยังเป็นกากบาทคือมีการขีดตัดกันตรงกลาง โปรแกรมนักศึกษาจะนับได้

**โปรแกรม**

|  |
| --- |
| blackcirc = im2bw(mat2gray(imread('circle.png')(:,:,1)), 0.8);  train = im2bw(mat2gray(imread('Choice\_train01.jpg')), 0.8);  test1 = im2bw(mat2gray(imread('Choice\_test01.jpg')), 0.8);  test2 = im2bw(mat2gray(imread('Choice\_Rtest01.jpg')), 0.8);  xcross = im2bw(mat2gray(imread('xcross.png')(:,:,1)), 0.8);  function [C] = myxcorr2(A, B)  s1 = max(size(A)(1), size(B)(1));  s2 = max(size(A)(2), size(B)(2));  C = real(ifft2(  conj(fft2(imcomplement(A), s1, s2))  .\*  fft2(imcomplement(B), s1, s2)));  C = flipud(C);  end  function [x1,y1,x2,y2] = getpivot(C, blackcirc)  [ssr,snd] = max(C(:));  [y1,x1] = ind2sub(size(C),snd);  p1 = padarray(blackcirc,[size(C)(1)-size(blackcirc)(1),size(C)(2)-size(blackcirc)(2)],255,'post');  p1 = circshift(p1, [y1-floor(size(blackcirc)(1)/2),x1-floor(size(blackcirc)(2)/2)]);  C\_ = imsubtract(mat2gray(C), imcomplement(mat2gray(p1)));  [ssr,snd] = max(C\_(:));  [y2,x2] = ind2sub(size(C\_),snd);  if x1 > x2  tmp1 = x1;  tmp2 = y1;  x1 = x2;  y1 = y2;  x2 = tmp1;  y2 = tmp2;  end  #y1 = size(C)(1) - y1;  #y2 = size(C)(1) - y2;  end  function [dis] = getdis(x1,y1,x2,y2)  dis = sqrt((x1-x2)^2+(y1-y2)^2);  end  function [ang] = getang(x1,y1,x2,y2)  ang = atan2(y2-y1,x2-x1) \* 180 / pi;  end  function [dis,ang,x1,y1,x2,y2] = getpivot2(A, B)  C = myxcorr2(A, B);  [x1,y1,x2,y2] = getpivot(C, B);  dis = getdis(x1,y1,x2,y2);  ang = getang(x1,y1,x2,y2);  end  function [A, B] = padeq(A, B)  d1 = size(A)(1) - size(B)(1);  d2 = size(A)(2) - size(B)(2);  if d1 < 0  A = padarray(A,[-d1,0],255,'post');  else  B = padarray(B,[d1,0],255,'post');  end  if d2 < 0  A = padarray(A,[0,-d2],255,'post');  else  B = padarray(B,[0,d2],255,'post');  end  end  function [E,F] = test(A, B, C, xcross)  [dis0,ang0,x01,y01,x02,y02] = getpivot2(A, C);  [dis2,ang2,x21,y21,x22,y22] = getpivot2(B, C);  [test2size1 test2size2] = size(B);  test2 = imrotate(B,ang0-ang2,'bilinear','crop');  test2 = imresize(B,dis0/dis2);  ang2\_ = (ang0-ang2) \* pi / 180;  x21\_ = x21 - test2size2/2;  y21\_ = y21 - test2size1/2;  x21\_ = x21\_ \* cos(ang2\_) - y21\_ \* sin(ang2\_);  y21\_ = x21\_ \* sin(ang2\_) + y21\_ \* cos(ang2\_);  x21\_ = x21\_ + test2size2/2;  y21\_ = y21\_ + test2size1/2;  x21\_ \*= dis0/dis2;  y21\_ \*= dis0/dis2;  x21\_ += size(C)(2)/2;  #y21\_ -= size(blackcirc)(1)/2;  x21\_ = floor(x21\_);  y21\_ = floor(y21\_);  test2 = imtranslate(B,x21\_-x01,y21\_-y01,'crop');  [A, B] = padeq(A, B);  se = strel('disk', 2, 0);  A = imdilate(imcomplement(A), se);  E = imsubtract(imcomplement(B), (A));  imshow(imcomplement(E));  F = myxcorr2(E, imcomplement(xcross));  end  [E1,1F] = test(train, test1, blackcirc, xcross);  [E2,F2] = test(train, test2, blackcirc, xcross); |