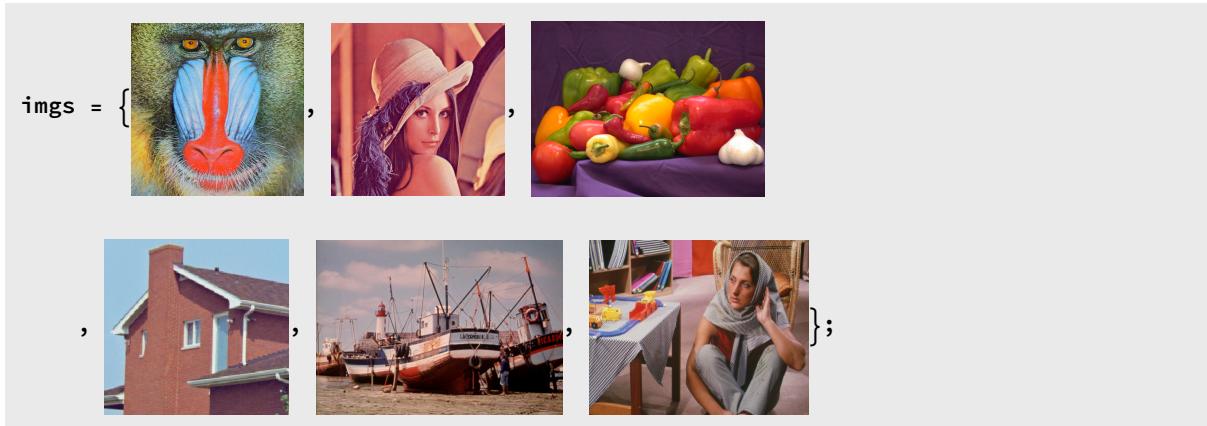


Kittipong Tapyou 65070501003

Homework 02 : Thresholding

Load Image

In[356]:=



Convert RGB image to grayscale

In[357]:=

```
grayImg = Table[ColorConvert[img, "Grayscale"], {img, imgs}];
```

In[358]:=

```
GraphicsGrid[
  ArrayReshape[
    Table[
      Image[grayImg, "Byte"]
    , {grayImg, grayImg}
    , {2, 3}
  ]
]
```

Out[358]=



Maximum Normal Line Method

```
GetPeakVal[grayImg_] := Module[{i, j, histVal, diffVal, peak, nPeak},
  For[i = 1, i < 256/5, i = i + 2,
    histVal = HistogramList[Flatten[ImageData[grayImg, "Byte"], 1], {i}];
    peak = Select[Range[Length[histVal[[2]]]],
      ((# == 1 && histVal[[2,1]] > histVal[[2,2]]) ||
       (# == Length[histVal[[2]]] && histVal[[2,-1]] > histVal[[2,-2]]) ||
       (1 < # < Length[histVal[[2]]] && histVal[[2,#]] > histVal[[2,#-1]] &&
        histVal[[2,#]] > histVal[[2,# + 1]])) &];
    nPeak = Length[peak];
    If[nPeak < 5,
      Return[
        {i,
         TakeLargestBy[
           Thread[{histVal[[1]][[peak]], histVal[[2]][[peak]]}], Last, 2
         ]}]
    ];
  ]
]
```

จาก function GetPeakVal จะมีการหา modal ของ การกระจายตัวของ gray intensity โดยที่จะทำการเลือกก่อนว่าจะใช้ขนาดของ bin เป็นเท่าไร โดยจะทำการใช้ตั้งแต่ bin ขนาด 1 ถึง 256/5 เนื่องจากหากต้องการให้มี 2 modal ในการกระจายตัว ควรจะมีอย่างน้อย 5 จุด และเพื่อนให้เริ่วขึ้นจะมีการเพิ่มขนาดของ bin ไปครั้งละ 2

จากนั้นจะทำการคำนวณหาค่าของ histogram ว่าแต่ละแท่งจะมีค่าเท่าไรบ้าง โดยกำหนดจากขนาด bin ในขั้นตอนก่อนหน้า จากนั้นทำการหา peak โดยการตรวจสอบการเปลี่ยนแปลงของ bin ที่ติดกันว่ามีการเปลี่ยนแปลงจากเพิ่มเป็นลด หรือลดเป็นเพิ่มหรือไม่ หากมีการเปลี่ยนแปลงดังเงื่อนไขที่กล่าวมา ก็จะดำเนินการนับว่าทั้งหมดแล้วมีกี่ modal โดยในแต่ละรอบนั้นจะมีการตรวจสอบว่า มี modal < 4 หรือไม่ หากใช่ ก็จะทำการหยุดการทำงานแล้วเลือกเอาแค่ 2 modal ที่มีค่ามากที่สุด

****หมายเหตุ**

- สาเหตุที่ เริ่มจาก bin มีขนาด 1 ก่อนนั้น เพื่อที่จะให้ histogram มีความละเอียดสูงสุดเท่าที่จะเป็นไปได้ก่อน ทำให้การหาค่า threshold มีความละเอียดเพิ่มขึ้น
- สาเหตุที่ เลือกให้หาให้ครบ 4 modal เพราะว่างานนี้ หากทำงานไปเรื่อยๆจนเหลือแค่ 2 modal จริงๆ ความละเอียดของ histogram จะน้อยทำให้ได้ค่า threshold ที่ไม่ละเอียด เลยตัดสินใจที่จะ เอาให้ครบ 4 เพราะว่า จะได้เลือกที่จะเอา 2 ตัวได้ หากมากกว่านั้น peak ที่ได้อาจอยู่ติดกันมากเกินไป

In[473]:=

```
peakVal = Table[GetPeakVal[grayImg], {grayImg, grayImgS}];
```

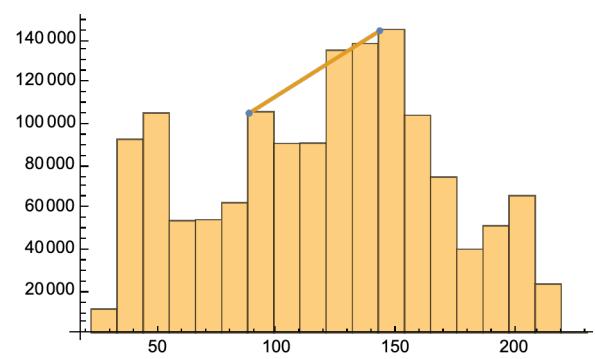
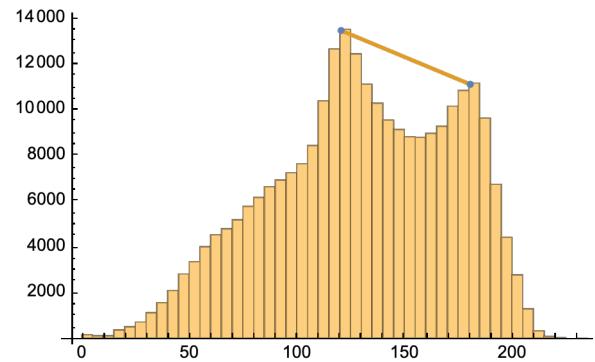
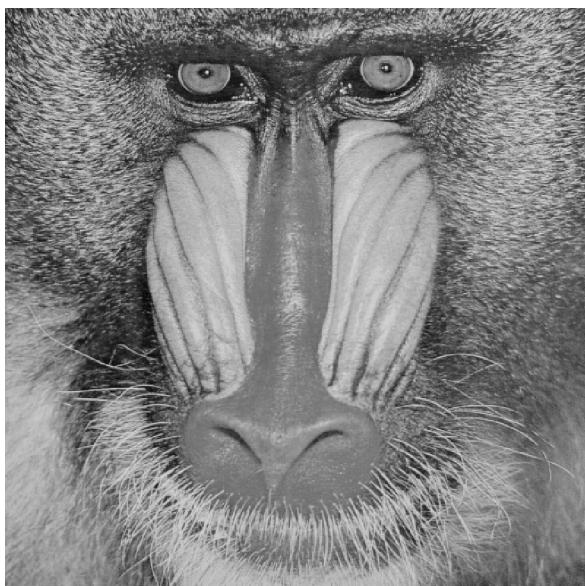
In[477]:=

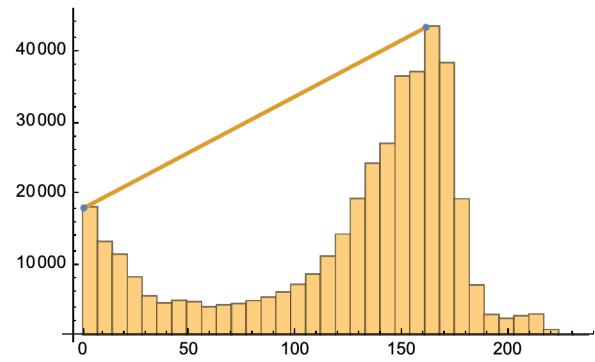
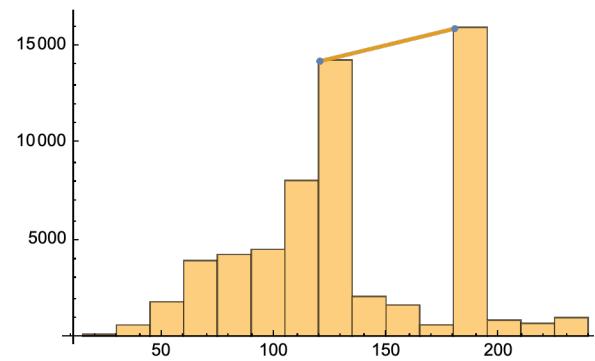
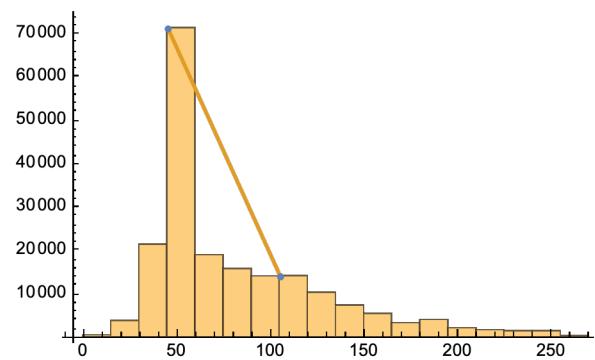
```

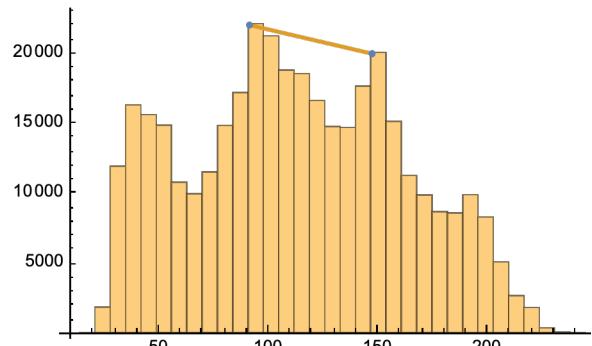
GraphicsGrid[
  ArrayReshape[
    Table[
      {
        grayImgs[[i]],
        Show[
          Histogram[Flatten[ImageData[grayImgs[[i]], "Byte"], 1], {peakVal[[i]]},
          ListPlot[{peakVal[[i]][2], peakVal[[i]][2]}], Joined→{False, True}],
          PlotRange→All]
        }
      , {i, Length[grayImgs]}]
    , {6, 2}]
  ]
]

```

Out[477]=







จากตัวอย่างด้านบน เป็นการแสดงถึงภาพแต่ละภาพ ว่ามีการกระจายตัวของ gray intensity เป็นอย่างไร และ histogram ที่แสดง เกิดจากการเลือก ขนาดของ bin ที่เหมาะสมที่สุด(จากที่กล่าวมาด้านบน) โดยใน histogram จะมีเลี้นตรงที่ลากจาก สองสุดสูงสุด ที่มีการเปลี่ยนแปลงจากเงื่อนไขก่อนหน้า

In[462]:=

```
PerpendicularDist[p1_, p2_, p3_] := Module[{x1, y1, x2, y2, x0, y0},
  {x1, y1} = p1;
  {x2, y2} = p2;
  {x0, y0} = p3;
  Return[
    Abs[(x2 - x1) (y1 - y0) - (x1 - x0) (y2 - y1)]/
    Sqrt[(x2 - x1)^2 + (y2 - y1)^2]
  ];
]
```

Function ด้านบน คือ function สำหรับการหา perpendicular distance โดยที่ p1 และ p2 เป็นจุดเริ่มและสิ้นสุดของเลี้นตรงและ p3 คือจุดใดๆ

In[568]:=

```
GetThreshold[grayImg_, peakVal_] := Module[
    {histVal, bound, ibIdx, max, i, dist, allPoints, ans = {}},
    histVal = HistogramList[Flatten[ImageData[grayImg, "Byte"], 1], {peakVal[[1]]}];
    histVal[[1]] = histVal[[1]][;;-2];
    bound = Transpose[SortBy[peakVal[[2]], First]][[1]];
    ibIdx = Flatten[Position[histVal[[1]], x_ /; bound[[1]] < x < bound[[2]]], 1];
    allPoints = Transpose[histVal];
    max = 0;
    Do[
        dist = PerpendicularDist[peakVal[[2, 1]], peakVal[[2, 2]], allPoints[[i]]];
        If[max < dist,
            ans = allPoints[[i]];
            max = dist;
        ]
    ,{i, ibIdx}];
    Return[ans];
]
```

การหา threshold จะเริ่มจากการหา bin ที่อยู่ระหว่างเลนต์รงนั้นเท่านั้น (ไม่รวมจุดบนเลนต์รง)
โดยที่จะทำการหา distance ของทุกจุดระหว่างนั้น และเลือกจุดที่มีระยะห่างจากเลนต์รงมากที่สุดเป็นค่า thresold

In[569]:=

```
GrayToBin[image_, thres_] := Module[{dim, img},
    img = ImageData[image, "Byte"];
    dim = Dimensions[img];
    Return[
        Table[If[img[[i, j]] > thres, 1, 0], {i, dim[[1]]}, {j, dim[[2]]}]
    ];
]
```

In[570]:=

```
thres = Table[
    GetThreshold[grayImg[i], peakVal[[i]]]
    ,{i, Length[grayImg]}
] [[All, 1]]
binImg = Table[Image[GrayToBin[grayImg[i], thres[[i]]]], {i, Length[grayImg]}];
```

Out[570]=

```
{150, 110, 60, 165, 98, 126}
```

จาก list ด้านบน คือค่า threshold ที่หาได้แต่ละรูป

In[572]:=

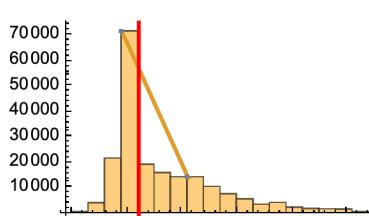
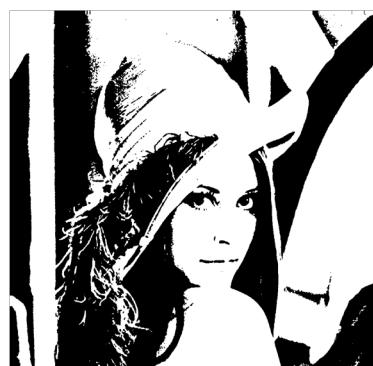
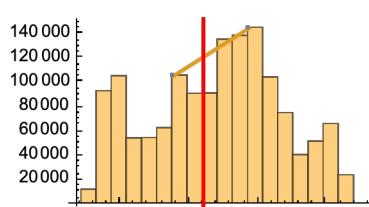
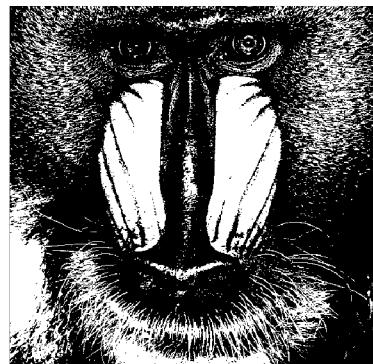
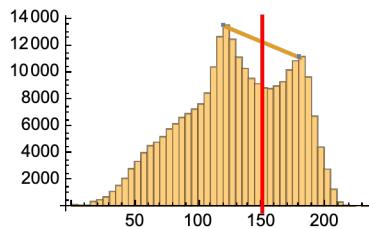
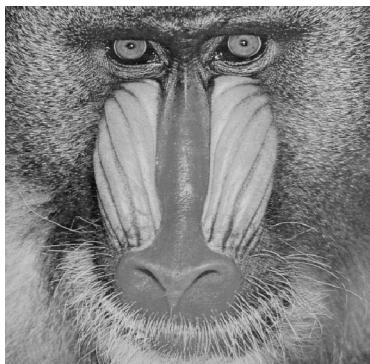
```
GrayToBin[image_, thres_] := Module[{dim, img},
    img = ImageData[image, "Byte"];
    dim = Dimensions[img];
    Return[
        Table[If[img[[i, j]] > thres, 1, 0], {i, dim[[1]]}, {j, dim[[2]]}]
    ];
]
```

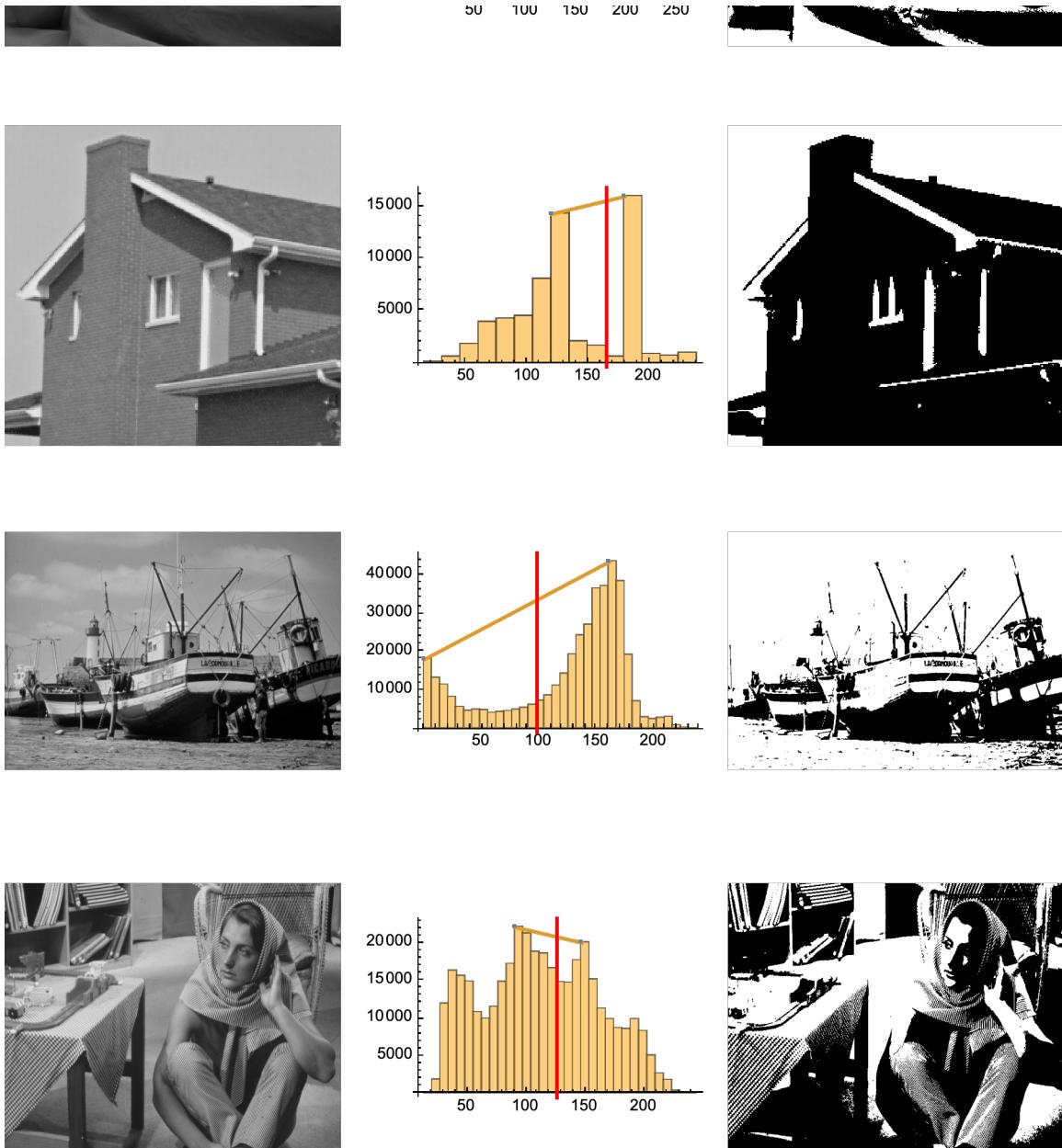
```

GraphicsGrid[
  ArrayReshape[
    Table[
      {
        grayImgs[[i]],
        Show[
          Histogram[
            Flatten[ImageData[grayImgs[[i]], "Byte"], 1], {peakVal[[i]][1]},
            Epilog -> {Red, Thick, InfiniteLine[
              {{thres[[i]], 0}, {thres[[i]], 1}}]
            }],
          ListPlot[{peakVal[[i]][2], peakVal[[i]][2]}, Joined -> {False, True}],
          PlotRange -> All
        ],
        binImgs[[i]]
      }
    , {i, Length[binImgs]}]
  , {6, 3}]
]

```

Out[573]=





จากรูปที่แสดงด้านบนจะเป็นการเปรียบเทียบรูปก่อน binarize และ หลัง จะเห็นได้ว่า histogram ของบางรูปนั้นจำเป็นต้องใช้ขนาดของ bin ที่ใหญ่ขึ้น เพื่อทำการหาค่า peak (รูปที่ 3, 4) โดยหากภาพมี spike เ酵ะเกินทำให้ต้องปรับขนาดของ bin ให้ใหญ่ขึ้น

โดยเส้นสีแดงแสดงถึงค่า threshold ที่ได้รูปภาพที่ได้รับนั้น
บางภาพก็สามารถที่จะแบ่งการกระจายตัวของ intensity ได้อย่างชัดเจน
แต่ไม่ได้หมายความว่าภาพที่ได้จากการทำ binarize จะสามารถจัดองค์ประกอบของภาพได้ครบ
 เช่นภาพเรือ จะเห็นได้ว่าการกระจายตัวนั้น หากมองด้วยตาเปล่าสามารถแบ่งได้ชัดเจน แต่ภาพที่ได้นั้น
 องค์ประกอบบางอย่างขาดหายไป

Average Intensities Method

In[586]:=

```
GetThresholdAveInt[img_, Δ_, w_] := Module[{pixel, δ = 255, g1, g2, thres, newThres},
  pixel = Flatten[ImageData[img, "Byte"], 1];
  thres = RandomInteger[{10, 255}];
  While[δ > Δ,
    g1 = Mean[Select[pixel, # < thres &]];
    g2 = Mean[Select[pixel, # ≥ thres &]];
    newThres = Dot[w, {g1, g2}] / Total[w];
    δ = Abs[thres - newThres];
    thres = newThres;
  ];
  Return[thres];
]
```

function นี้คือการหาค่า threshold โดยการรับค่า Δ คือผลต่างที่ยอมรับ, และ w คือ weight ของแต่ละกลุ่มจากนั้นสุ่ม threshold ตั้งแต่ 10 ถึง 255 จากนั้นแบ่งกลุ่มของข้อมูลเป็นสองกลุ่ม และหาค่าเฉลี่ย โดยหลังจากนั้นจะทำค่าเฉลี่ยที่ได้ในแต่ละกลุ่มมาคูณกับ weight (ใช้ dot product) โดยที่จะทำการเก็บผลต่างของ threshold ครั้งก่อนหน้าไว้ที่ δ นำไปเรื่อยๆจนกว่า $\delta > \Delta$

```
thresAveInt = Table[GetThresholdAveInt[grayImg, 0.01, {0.5, 0.5}], {grayImg, grayImgs}]
binImgsAveInt = Table[
  Image[GrayToBin[grayImgs[[i]], thresAveInt[[i]]]],
  {i, Length[grayImgs]}
];
```

Out[589]=

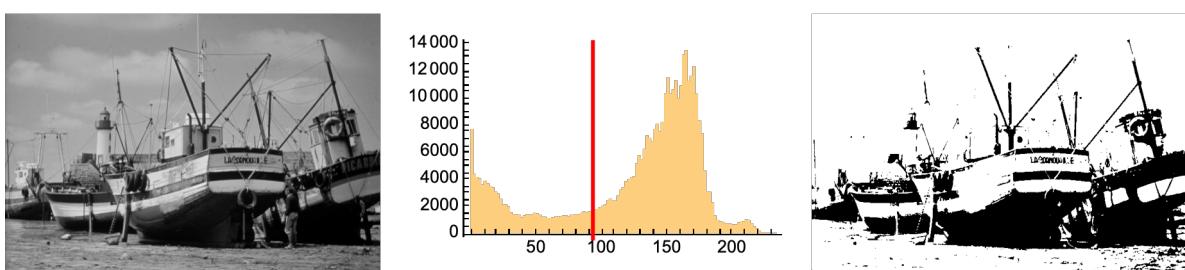
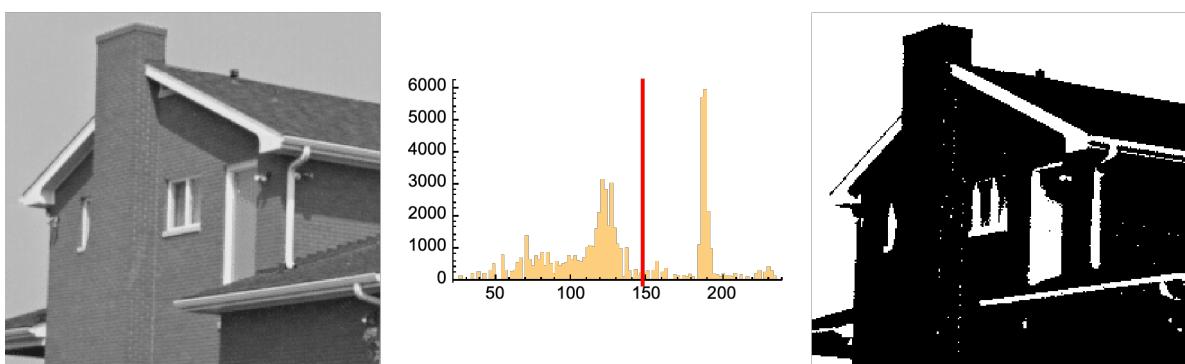
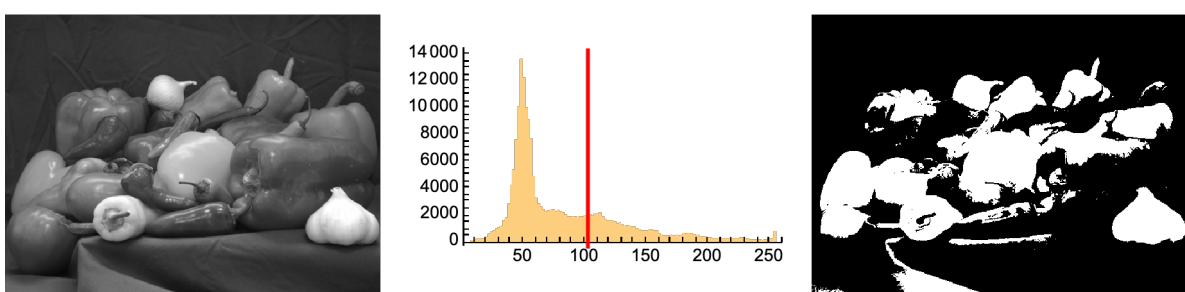
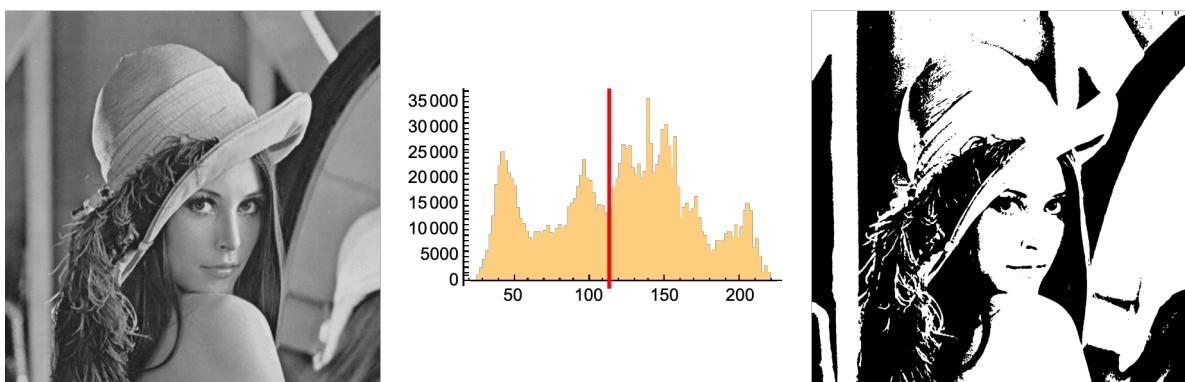
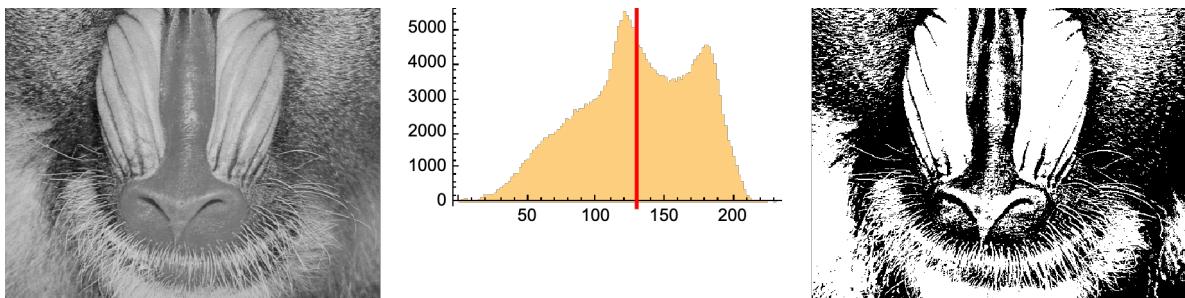
```
{127.77, 113.057, 102.089, 146.764, 92.5564, 116.119}
```

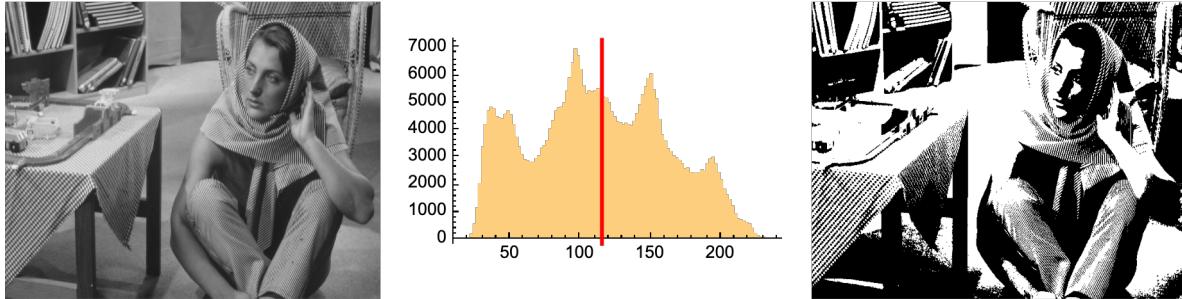
ค่าใน list ด้านบนคือค่า threshold ที่หาได้ของภาพแต่ละภาพ

```
GraphicsGrid[
  ArrayReshape[
    Table[
      {
        grayImgs[[i]],
        Histogram[
          Flatten[ImageData[grayImgs[[i]], "Byte"], 1], {2},
          Epilog → {Red, Thick, InfiniteLine[
            {{thresAveInt[[i]], 0}, {thresAveInt[[i]], 1}}
          ]}]
        ],
        binImgsAveInt[[i]]
      }
    , {i, Length[binImgs]}]
  , {6, 3}
]
```

Out[575]=







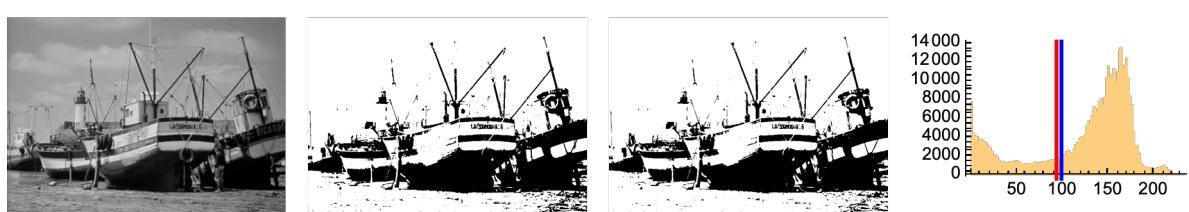
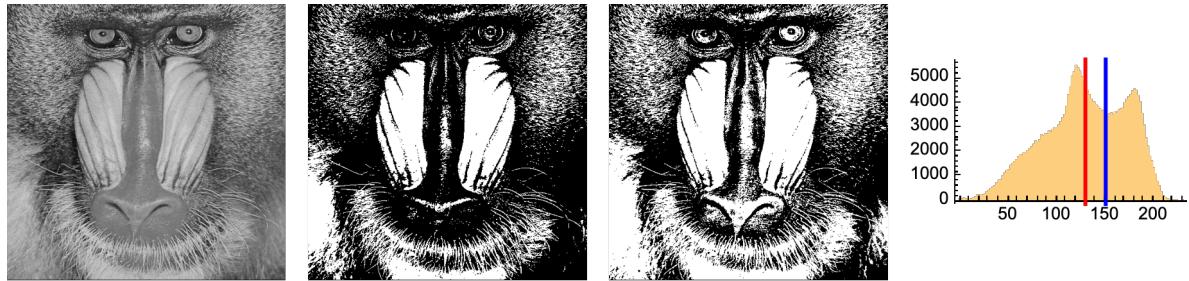
จากรูปด้านบนจะเป็นการเปรียบเทียบรูปก่อนทำ binarize กับหลังทำ binarize โดยใช้วิธีการ Average Intensities Method จะเห็นได้ว่าวิธีการนี้ไม่ได้แบ่งข้อมูลจาก modal แต่จะเน้นที่การดูปริมาณของ intensity สะสมในแต่ละกลุ่ม(พื้นที่ ได้กราฟ) โดยพยายามแบ่งข้อมูลให้มีพื้นที่เท่าๆ กัน จากรูปแรกจะเห็นได้ว่าค่า threshold ที่ได้นั้นไม่ได้แบ่งกึ่งกลางระหว่าง 2 modal เมื่ออนวิธีการก่อนหน้า รูปภาพที่ได้นั้นสามารถเก็บรายละเอียดได้ดี ยกเว้นรูปบ้าน

```

GraphicsGrid[
  ArrayReshape[
    Table[
      {
        grayImg[ i ],
        binImg[ i ],
        binImgAveInt[ i ],
        Histogram[
          Flatten[ImageData[grayImg[ i ], "Byte"], 1], {2},
          Epilog -> {
            Red, Thick, InfiniteLine[
              {{thresAveInt[ i ], 0}, {thresAveInt[ i ], 1}}
            ],
            Blue, Thick, InfiniteLine[
              {{thres[ i ], 0}, {thres[ i ], 1}}
            ],
          }
        ]
      }
      , {i, Length[binImg]}
      , {6, 4}
    ]
  ]
]

```

Out[579]=



จากการเปรียบเทียบรูปภาพที่ได้จากการทำ binarize ของทั้งสองวิธี โดยที่ column ที่ 2 คือวิธีการ Maximum Normal Line Method, column ที่ 3 คือ Average Intensities Method และ column สุดท้ายคือ histogram ของภาพ grayscale โดยที่เส้นสีแดงคือ Average Intensities Method และเส้นสีฟ้าคือ Maximum Normal Line Method

การเปรียบเทียบ ในมุกการประมวลผล(อ้างอิงจาก algorithm ที่เขียน) Maximum Normal Line Method มีการประมวลผลที่ช้ากว่าวิธีที่สองเนื่องจากต้องทำการหาค่า bin ที่เหมาะสมในการหา peak ของการกระจายตัว จากนั้นจึงต้องทำการหาระยะห่างของข้อมูลระหว่าง 2 modal

การเปรียบเทียบเชิงรายละเอียดของภาพ Average Intensities Method
สามารถเก็บรายละเอียดของภาพได้ดีกว่าวิธีการแรกมากๆ สังเกตได้จาก ภาพแรก
รายละเอียดของลิงนั้นมีความละเอียดมากกว่า
หรือภายในดวงตาเห็นการแยกออกจากกันของตาด้านขวาอย่างชัดเจน ภาพถัดมาคือ pepper
หากเราต้องการที่จะนัดหรือดูจำนวนชนิดของวัตถุ ในภาพนั้น
วิธีที่สองจะทำให้เราเห็นถึงขอบของวัตถุแต่ละชิ้นได้ดีกว่าวิธีการแรก