

Image Boarder Detection

Kathy. Chenying. Gao

October 13, 2016

This is an R Markdown Image Boarder Detection Example

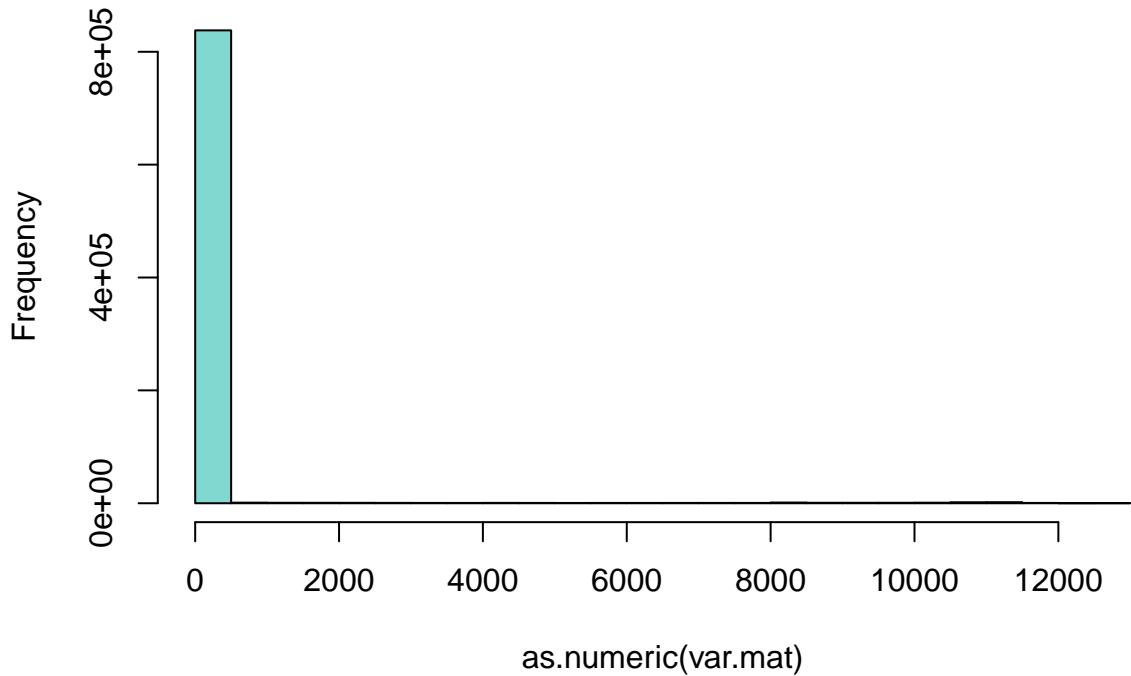
Github Logo

```
# Load image
matfile <- readMat("/Users/gaochenying1/Desktop/My Projects Demo/github.mat")
testimage = matfile$maskedRGBImage
dim(testimage)

## [1] 701 1234      3

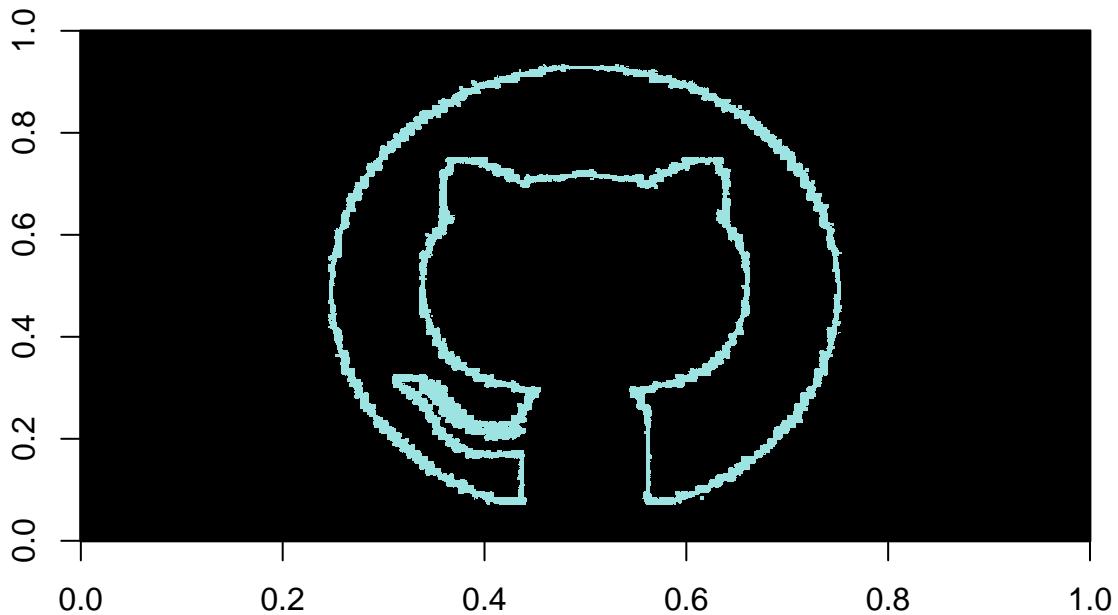
# Let's see how it goes for a single image
nr=dim(testimage)[1]
nc=dim(testimage)[2]
band=5
N=nr*nc
flag.mat=matrix(0,nrow = (nr-(band-1)),ncol = (nc-(band-1)))
var.mat=matrix(0,nrow = (nr-(band-1)),ncol = (nc-(band-1)))
# Calculate Variance of Neighbours
for (i in 3:(nr-(band-1)/2))
{
  for (j in 3:(nc-(band-1)/2))
  {
    tempmat1=testimage[(i-2):(i+2),(j-2):(j+2),1]
    tempvar1=sum((tempmat1-mean(tempmat1))^2)/(band^2)
    # tempflag=ifelse(tempvar>
    var.mat[i-(band-1)/2,j-(band-1)/2]=tempvar1
  }
}
hist(as.numeric(var.mat), col = "#81D8D0") # Adjust Threshold according to this
```

Histogram of as.numeric(var.mat)



```
# Flag the large variance element and call them borders
thershold=quantile(var.mat,0.95)
for (i in 1:(nr-(band-1)))
{
  for (j in 1:(nc-(band-1)))
  {
    tempflag=ifelse(var.mat[i,j]>thershold,1,0)
    flag.mat[i,j]=tempflag
  }
}
flag.rt = t(apply(flag.mat, 2, rev))
image(flag.rt,col=c("black", "#9de3e1"), main = "Github Logo")
```

Github Logo



```
remove(list=c("testimage", "matfile", "flag.mat", "var.mat"))
```

LinkedIn Logo

```
# Load image
matfile <- readMat("/Users/gaochenying1/Desktop/My Projects Demo/linkedin.mat")
testimage = matfile$maskedRGBImage
dim(testimage)
```

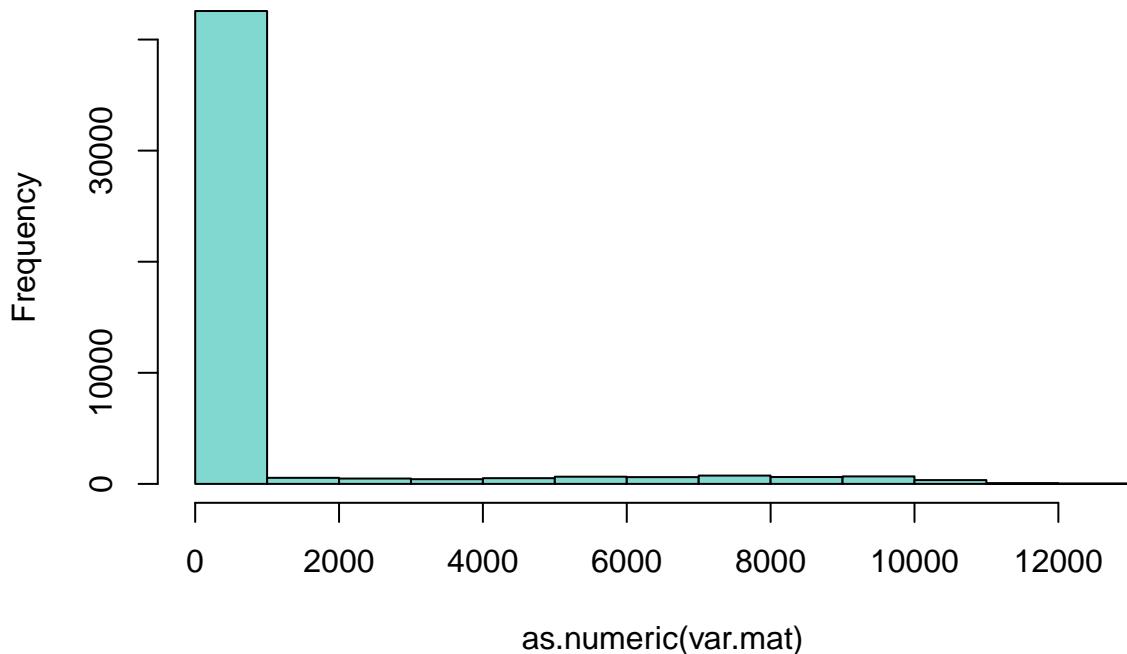
```
## [1] 137 367 3
```

```
# Let's see how it goes for a single image
nr=dim(testimage)[1]
nc=dim(testimage)[2]
band=5
N=nr*nc
flag.mat=matrix(0,nrow = (nr-(band-1)),ncol = (nc-(band-1)))
var.mat=matrix(0,nrow = (nr-(band-1)),ncol = (nc-(band-1)))
# Calculate Variance of Neighbours
for (i in 3:(nr-(band-1)/2))
{
  for (j in 3:(nc-(band-1)/2))
  {
    tempmat1=testimage[(i-2):(i+2),(j-2):(j+2),1]
    tempvar1=sum((tempmat1-mean(tempmat1))^2)/(band^2)
    # tempflag=ifelse(tempvar>
    var.mat[i-(band-1)/2,j-(band-1)/2]=tempvar1
  }
}
```

```
}
```

```
hist(as.numeric(var.mat), col = "#81D8D0") # Adjust Threshold according to this
```

Histogram of as.numeric(var.mat)



```
# Flag the large variance element and call them borders
```

```
thershold=quantile(var.mat,0.92)
```

```
for (i in 1:(nr-(band-1)))
```

```
{
```

```
  for (j in 1:(nc-(band-1)))
```

```
  {
```

```
    tempflag=ifelse(var.mat[i,j]>thershold,1,0)
```

```
    flag.mat[i,j]=tempflag
```

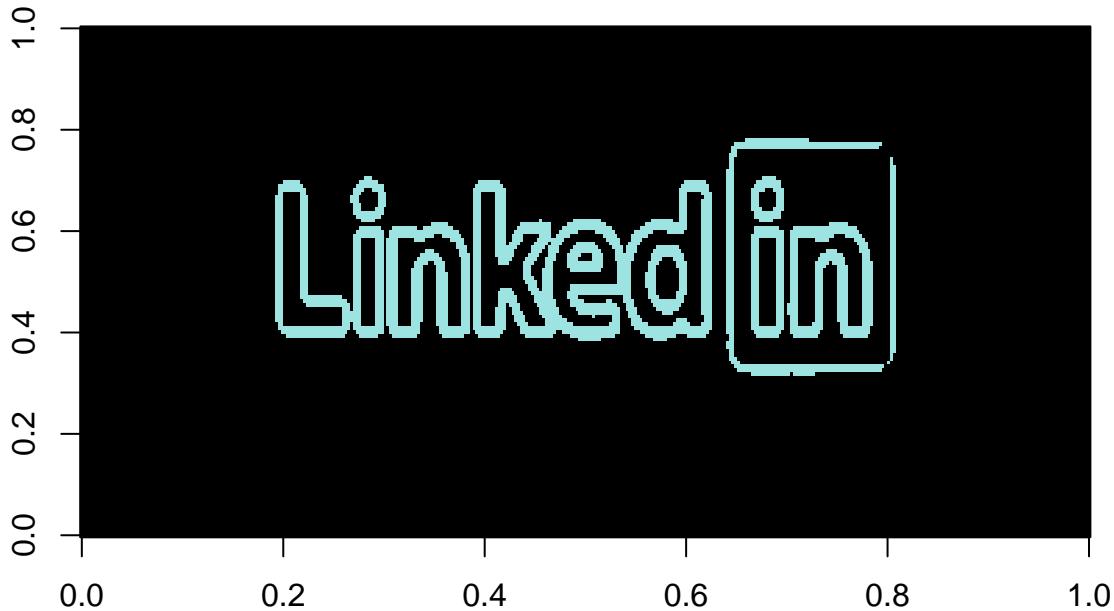
```
  }
```

```
}
```

```
flag.rt = t(apply(flag.mat, 2, rev))
```

```
image(flag.rt,col=c("black", "#9de3e1"), main = "LinkedIn Logo")
```

LinkedIn Logo



```
remove(list=c("testimage", "matfile", "flag.mat", "var.mat"))
```

Pepe The Frog

```
# Load image
matfile <- readMat("/Users/gaochenying1/Desktop/My Projects Demo/frog.mat")
testimage = matfile$maskedRGBImage
dim(testimage)

## [1] 675 1200     3

# Let's see how it goes for a single image
nr=dim(testimage)[1]
nc=dim(testimage)[2]
band=5
N=nr*nc
flag.mat=matrix(0,nrow = (nr-(band-1)),ncol = (nc-(band-1)))
var.mat=matrix(0,nrow = (nr-(band-1)),ncol = (nc-(band-1)))
# Calculate Variance of Neighbours
for (i in 3:(nr-(band-1)/2))
{
  for (j in 3:(nc-(band-1)/2))
  {
    tempmat1=testimage[(i-2):(i+2),(j-2):(j+2),1]
    tempvar1=sum((tempmat1-mean(tempmat1))^2)/(band^2)
    # tempflag=ifelse(tempvar>
    var.mat[i-(band-1)/2,j-(band-1)/2]=tempvar1
  }
}
```

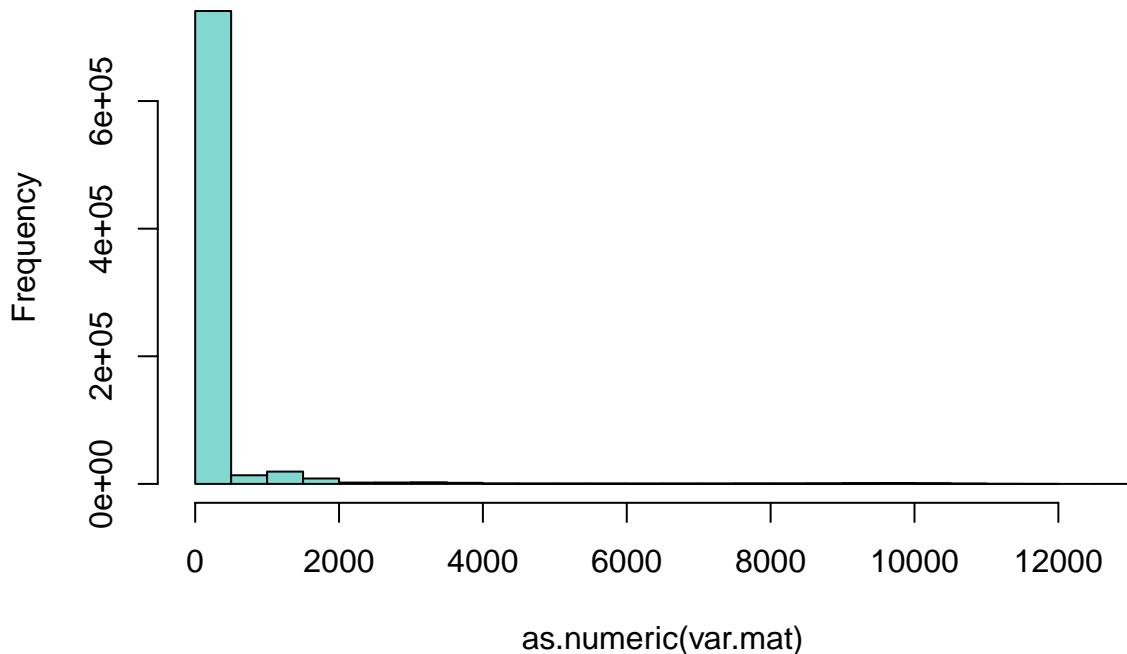
```

}

hist(as.numeric(var.mat), col = "#81D8D0") # Adjust Threshold according to this

```

Histogram of as.numeric(var.mat)

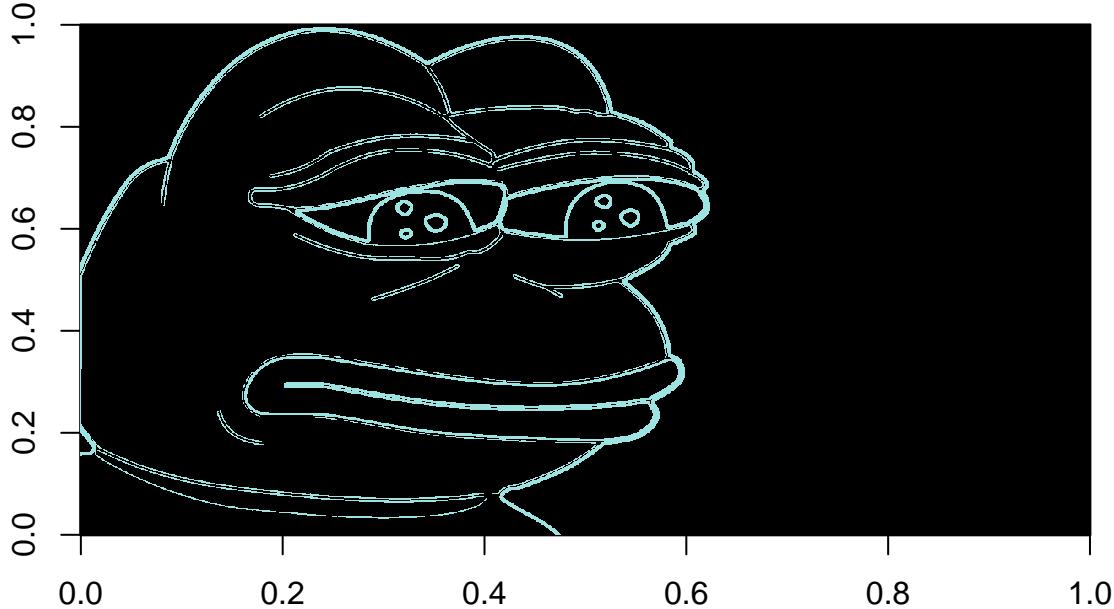


```

# Flag the large variance element and call them borders
thershold=quantile(var.mat,0.95)
for (i in 1:(nr-(band-1)))
{
  for (j in 1:(nc-(band-1)))
  {
    tempflag=ifelse(var.mat[i,j]>thershold,1,0)
    flag.mat[i,j]=tempflag
  }
}
flag.rt = t(apply(flag.mat, 2, rev))
image(flag.rt,col=c("black", "#9de3e1"), main = "Pepe the Frog")

```

Pepe the Frog



```
remove(list=c("testimage", "matfile", "flag.mat", "var.mat"))
```

Rose

```
# Load image
matfile <- readMat("/Users/gaochenying1/Desktop/My Projects Demo/rose.mat")
testimage = matfile$maskedRGBImage
dim(testimage)

## [1] 278 181 3

# Let's see how it goes for a single image
nr=dim(testimage)[1]
nc=dim(testimage)[2]
band=5
N=nr*nc
flag.mat=matrix(0,nrow = (nr-(band-1)),ncol = (nc-(band-1)))
var.mat=matrix(0,nrow = (nr-(band-1)),ncol = (nc-(band-1)))
# Calculate Variance of Neighbours
for (i in 3:(nr-(band-1)/2))
{
  for (j in 3:(nc-(band-1)/2))
  {
    tempmat1=testimage[(i-2):(i+2),(j-2):(j+2),1]
    tempvar1=sum((tempmat1-mean(tempmat1))^2)/(band^2)
    # tempflag=ifelse(tempvar>
    var.mat[i-(band-1)/2,j-(band-1)/2]=tempvar1
  }
}
```

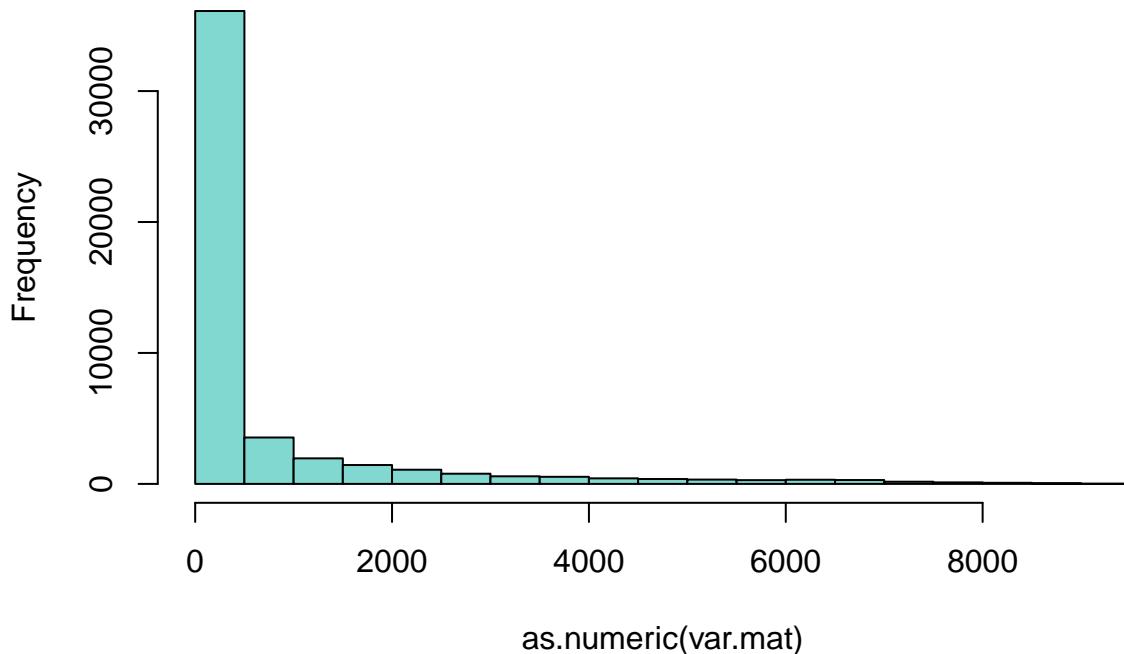
```

}

hist(as.numeric(var.mat), col = "#81D8D0") # Adjust Threshold according to this

```

Histogram of as.numeric(var.mat)

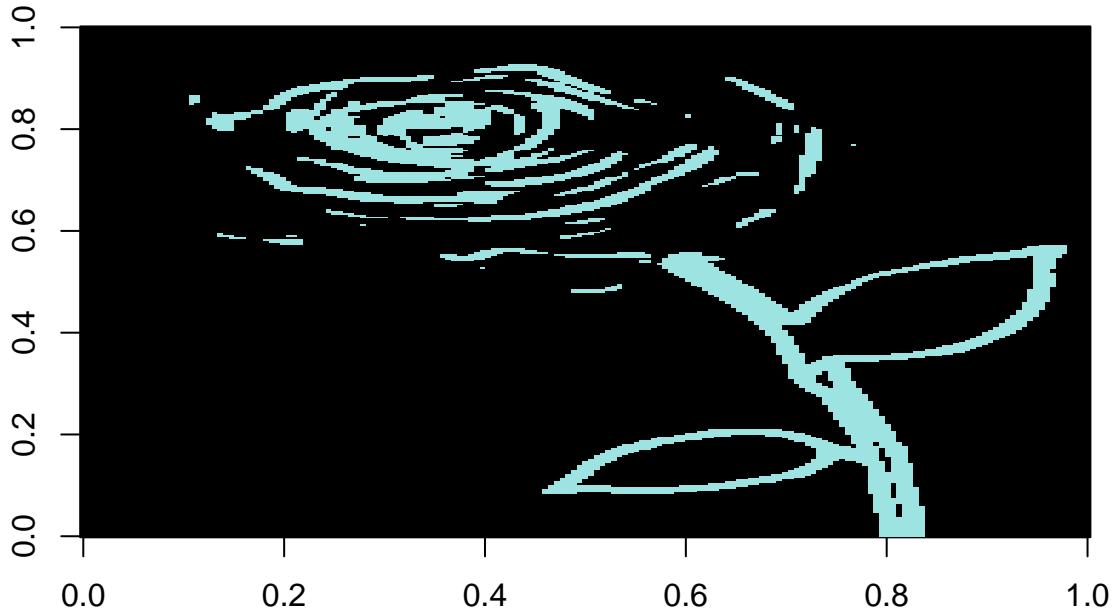


```

# Flag the large variance element and call them borders
thershold=quantile(var.mat,0.9)
for (i in 1:(nr-(band-1)))
{
  for (j in 1:(nc-(band-1)))
  {
    tempflag=ifelse(var.mat[i,j]>thershold,1,0)
    flag.mat[i,j]=tempflag
  }
}
flag.rt = t(apply(flag.mat, 2, rev))
image(flag.rt,col=c("black", "#9de3e1"), main = "Rose")

```

Rose



```
remove(list=c("testimage", "matfile", "flag.mat", "var.mat"))
```

My Desktop

```
# Load image
matfile <- readMat("/Users/gaochenying1/Desktop/My Projects Demo/mydesktop.mat")
testimage = matfile$maskedRGBImage
dim(testimage)

## [1] 727 983    3

# Let's see how it goes for a single image
nr=dim(testimage)[1]
nc=dim(testimage)[2]
band=5
N=nr*nc
flag.mat=matrix(0,nrow = (nr-(band-1)),ncol = (nc-(band-1)))
var.mat=matrix(0,nrow = (nr-(band-1)),ncol = (nc-(band-1)))
# Calculate Variance of Neighbours
for (i in 3:(nr-(band-1)/2))
{
  for (j in 3:(nc-(band-1)/2))
  {
    tempmat1=testimage[(i-2):(i+2),(j-2):(j+2),1]
    tempvar1=sum((tempmat1-mean(tempmat1))^2)/(band^2)
    # tempflag=ifelse(tempvar>
    var.mat[i-(band-1)/2,j-(band-1)/2]=tempvar1
  }
}
```

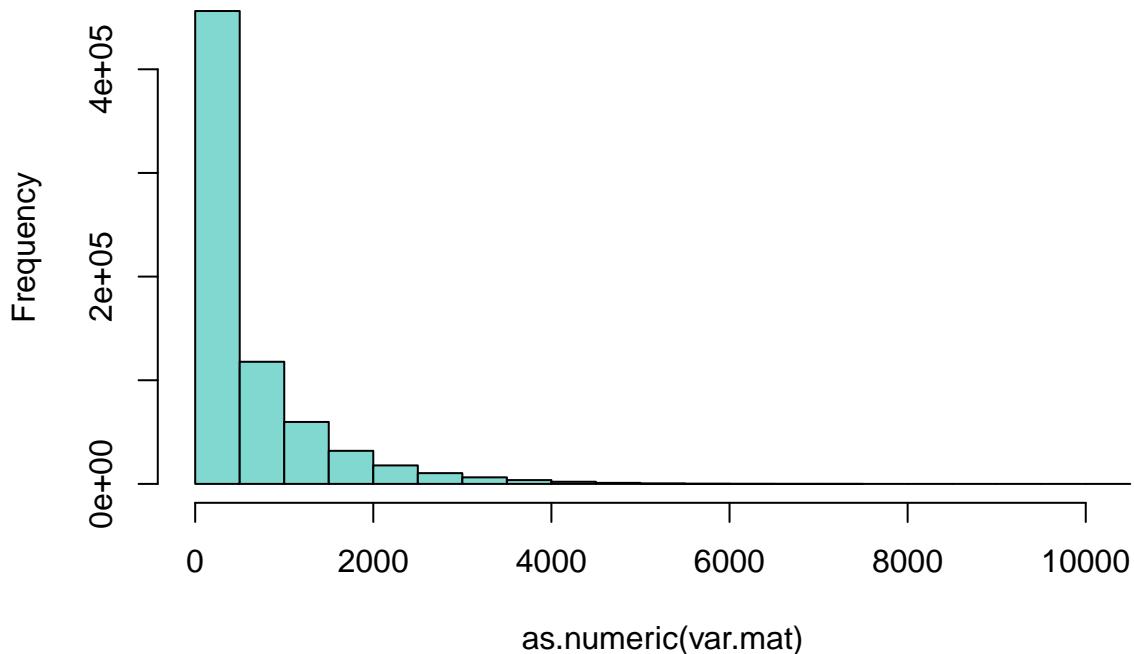
```

}

hist(as.numeric(var.mat), col = "#81D8D0") # Adjust Threshold according to this

```

Histogram of as.numeric(var.mat)

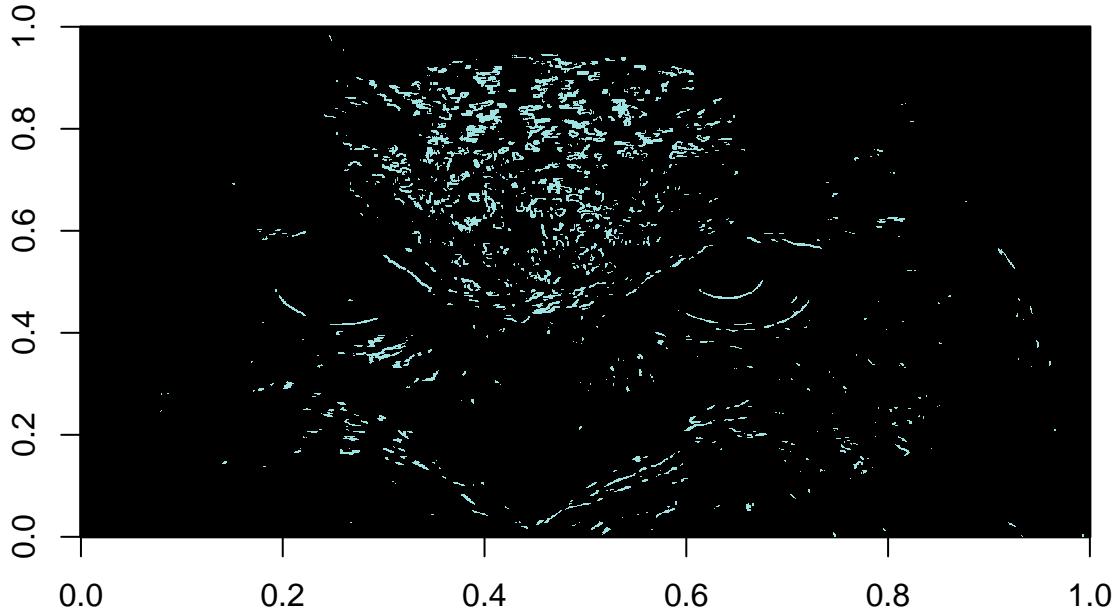


```

# Flag the large variance element and call them borders
thershold=quantile(var.mat,0.97)
for (i in 1:(nr-(band-1)))
{
  for (j in 1:(nc-(band-1)))
  {
    tempflag=ifelse(var.mat[i,j]>thershold,1,0)
    flag.mat[i,j]=tempflag
  }
}
flag.rt = t(apply(flag.mat, 2, rev))
image(flag.rt,col=c("black", "#9de3e1"), main = "My Desktop")

```

My Desktop



```
remove(list=c("testimage", "matfile", "flag.mat", "var.mat"))
```

Why So Serious

```
# Load image
matfile <- readMat("/Users/gaochenying1/Desktop/My Projects Demo/whysoserious.mat")
testimage = matfile$maskedRGBImage
dim(testimage)

## [1] 900 1600 3

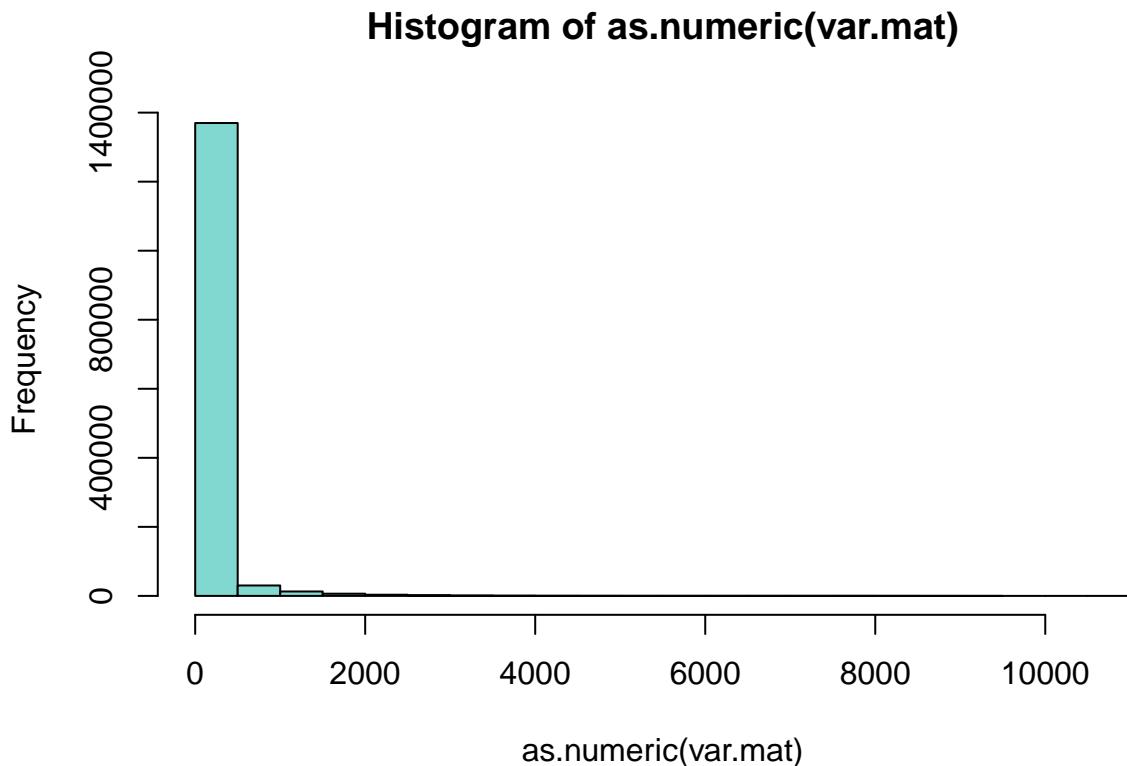
# Let's see how it goes for a single image
nr=dim(testimage)[1]
nc=dim(testimage)[2]
band=5
N=nr*nc
flag.mat=matrix(0,nrow = (nr-(band-1)),ncol = (nc-(band-1)))
var.mat=matrix(0,nrow = (nr-(band-1)),ncol = (nc-(band-1)))
# Calculate Variance of Neighbours
for (i in 3:(nr-(band-1)/2))
{
  for (j in 3:(nc-(band-1)/2))
  {
    tempmat1=testimage[(i-2):(i+2),(j-2):(j+2),1]
    tempvar1=sum((tempmat1-mean(tempmat1))^2)/(band^2)
    # tempflag=ifelse(tempvar>
    var.mat[i-(band-1)/2,j-(band-1)/2]=tempvar1
  }
}
```

```

}

hist(as.numeric(var.mat), col = "#81D8D0") # Adjust Threshold according to this

```

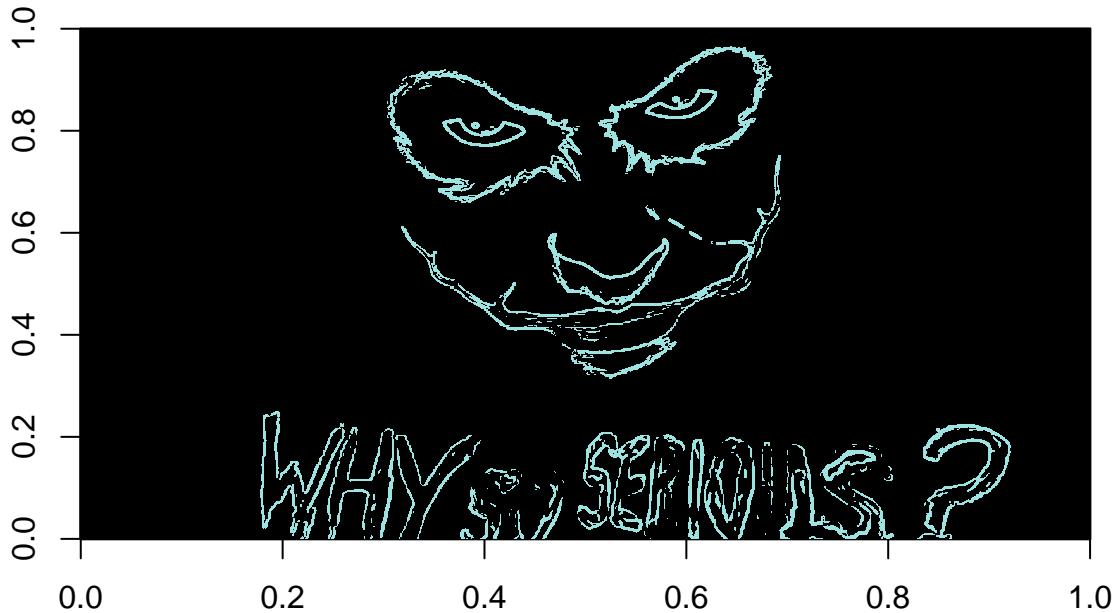


```

# Flag the large variance element and call them borders
thershold=quantile(var.mat,0.95)
for (i in 1:(nr-(band-1)))
{
  for (j in 1:(nc-(band-1)))
  {
    tempflag=ifelse(var.mat[i,j]>thershold,1,0)
    flag.mat[i,j]=tempflag
  }
}
flag.rt = t(apply(flag.mat, 2, rev))
image(flag.rt,col=c("black", "#9de3e1"), main = "Why So Serious")

```

Why So Serious



```
remove(list=c("testimage", "matfile", "flag.mat", "var.mat"))
```

Red Panda

```
# Load image
matfile <- readMat("/Users/gaochenying1/Desktop/My Projects Demo/redpanda.mat")
testimage = matfile$maskedRGBImage
dim(testimage)

## [1] 768 1024 3

# Let's see how it goes for a single image
nr=dim(testimage)[1]
nc=dim(testimage)[2]
band=5
N=nr*nc
flag.mat=matrix(0,nrow = (nr-(band-1)),ncol = (nc-(band-1)))
var.mat=matrix(0,nrow = (nr-(band-1)),ncol = (nc-(band-1)))
# Calculate Variance of Neighbours
for (i in 3:(nr-(band-1)/2))
{
  for (j in 3:(nc-(band-1)/2))
  {
    tempmat1=testimage[(i-2):(i+2),(j-2):(j+2),1]
    tempvar1=sum((tempmat1-mean(tempmat1))^2)/(band^2)
    # tempflag=ifelse(tempvar>
    var.mat[i-(band-1)/2,j-(band-1)/2]=tempvar1
  }
}
```

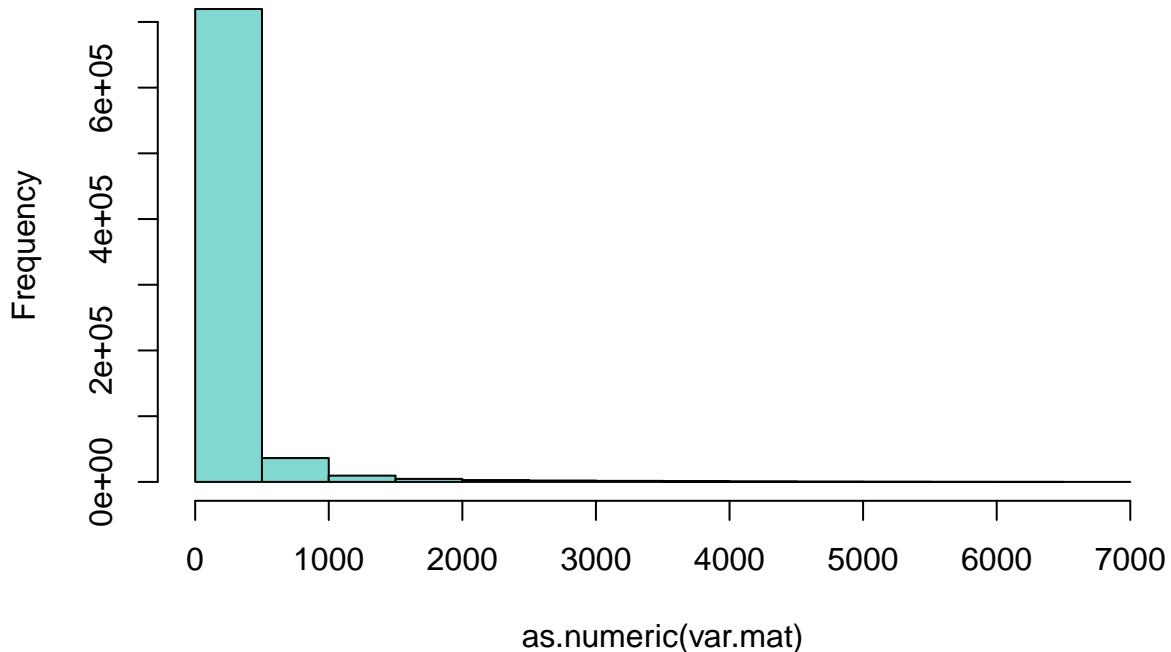
```

}

hist(as.numeric(var.mat), col = "#81D8D0") # Adjust Threshold according to this

```

Histogram of as.numeric(var.mat)

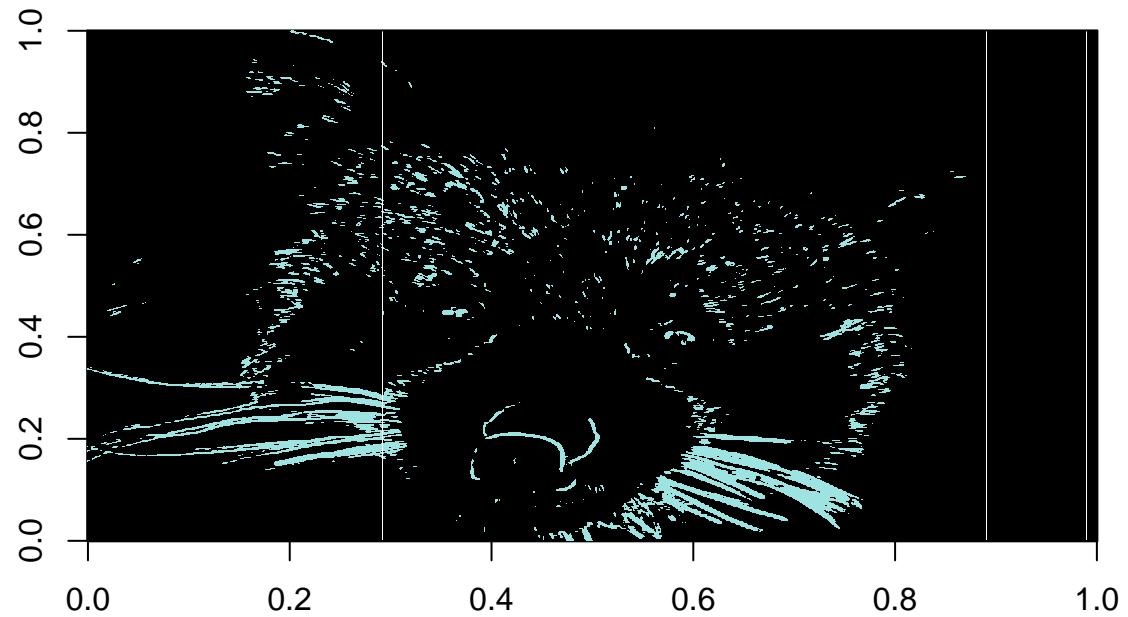


```

# Flag the large variance element and call them borders
thershold=quantile(var.mat,0.95)
for (i in 1:(nr-(band-1)))
{
  for (j in 1:(nc-(band-1)))
  {
    tempflag=ifelse(var.mat[i,j]>thershold,1,0)
    flag.mat[i,j]=tempflag
  }
}
flag.rt = t(apply(flag.mat, 2, rev))
image(flag.rt,col=c("black", "#9de3e1"), main = "Red Panda")

```

Red Panda



```
remove(list=c("testimage", "matfile", "flag.mat", "var.mat"))
```