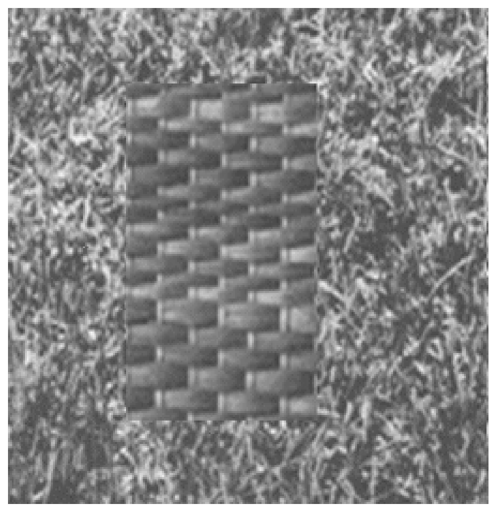
ECGR 6118
Computer Project: Segmentation
Student Name:___________

(copyright T. Weldon, 2006)

**For this project, you may use mathcad or NetBeans**
**However, MathCad will provide the most direct solution**

Complete the tasks below and turn in a project report answering all questions.

infile := READBMP("grass1weave55t1.gif")



$rr := 0..rows(infile) - 1 \quad cc := 0..cols(infile) - 1$

$rows(infile) = 256 \quad cols(infile) = 256$

$max(infile) = 251 \quad min(infile) = 61$

**Define 2 Gabor filters in frequency domain**

$ri := rows(infile) \quad ci := cols(infile)$

$r0 := 50 \quad c0 := 50 \quad \sigma := 20$

$$gab1_{rr,cc} := e^{-\left[\frac{\left(if\left(rr<\frac{ri}{2}, rr, rr-ri\right)-r0\right)^2+\left(if\left(cc<\frac{ci}{2}, cc, cc-ci\right)-c0\right)^2}{\sigma^2}\right]}$$

$r02 := 15 \quad c02 := 0 \quad \sigma 2 := 10$

$$gab2_{rr,cc} := e^{-\left[\frac{\left(if\left(rr<\frac{ri}{2}, rr, rr-ri\right)-r02\right)^2+\left(if\left(cc<\frac{ci}{2}, cc, cc-ci\right)-c02\right)^2}{\sigma 2^2}\right]}$$

**1.** Why is the "if(rr<ri/2,rr,rr-ri)" needed in the definition of the above Gabor filters in the frequency domain? (Note: gab1 and gab2 are frequency responses of a Gabor filter) Hint: try setting r0=0 and plot gab1 with and without the "if"

**Filter the image (AC component only) using both filters**

$X := \mathrm{cfft}(\mathrm{infile})$ $\quad\quad$ $xac := \mathrm{infile} - \mathrm{mean}(\mathrm{infile})$ $\;$ $Xac := \mathrm{cfft}(xac)$

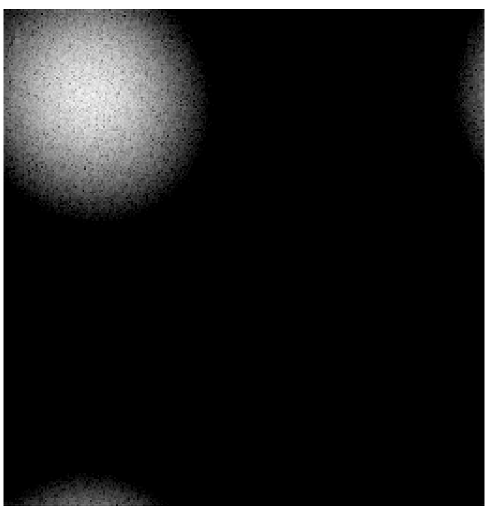
$Y1 := \overrightarrow{(gab1 \cdot Xac)}$ $\quad\quad$ $Y2 := \overrightarrow{(gab2 \cdot Xac)}$

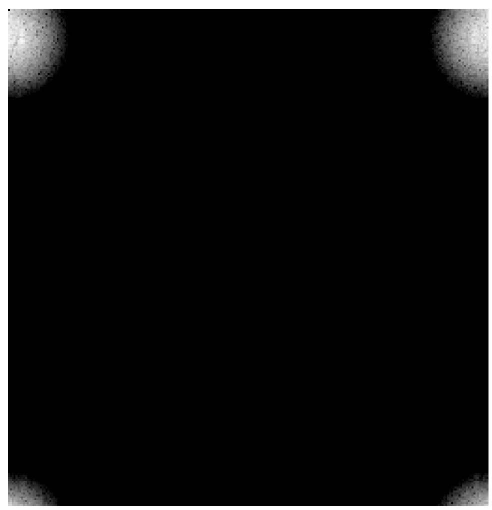
$logY1 := \overrightarrow{50\log(|Y1| + 0.00000001)}$ $\quad\quad$ $logY2 := \overrightarrow{50\log(|Y2| + 0.00000001)}$

$maxY1 := \max(logY1)$ $\quad$ $maxY1 = 89.64$ $\quad\quad$ $maxY2 := \max(logY2)$ $naxY2 = 123.31$

$logY1 := logY1 - maxY1 + 250$ $\quad\quad$ $logY2 := logY2 - maxY2 + 250$

$logY1_{rr,cc} := \mathrm{if}\left(logY1_{rr,cc} < 0, 0, \mathrm{floor}\left(logY1_{rr,cc}\right)\right)$ $\quad$ $logY2_{rr,cc} := \mathrm{if}\left(logY2_{rr,cc} < 0, 0, \mathrm{floor}\left(logY2_{rr,cc}\right)\right)$





**2.** Why does the Gabor filter frequency response appear in all four corners of the right-hand plot?

### Take inverse transform to recover filtered images

$ygab1 := icfft(Y1)$

$ygab1 := \overrightarrow{|ygab1|}$

$miny1 := min(ygab1) \quad maxy1 := max(ygab1)$

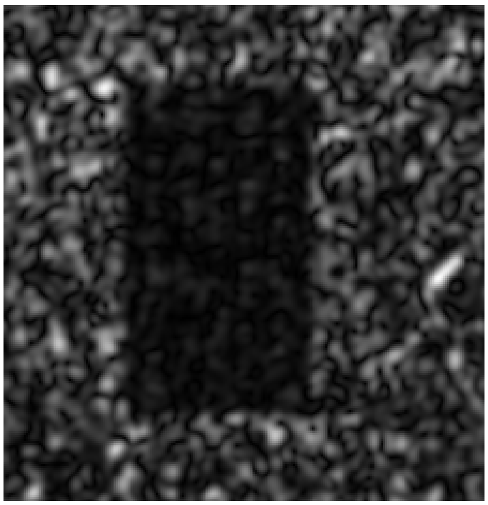
$miny1 = 5.902 \times 10^{-}$[illegible] $maxy1 = 9.499$

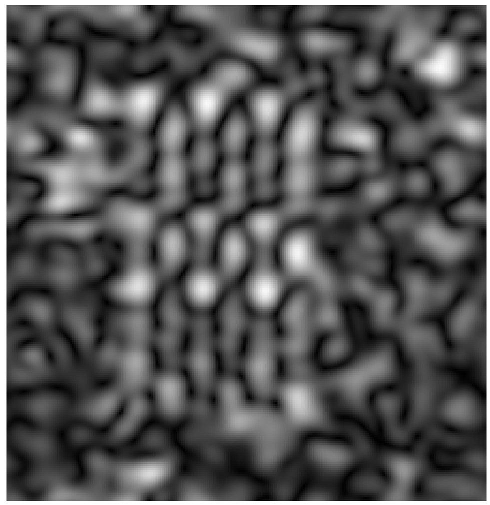
$ygab2 := icfft(Y2)$

$ygab2 := \overrightarrow{|ygab2|}$

$miny2 := min(ygab2) \quad maxy2 := max(ygab2)$

$miny2 = 0.012 \qquad maxy2 = 19.507$





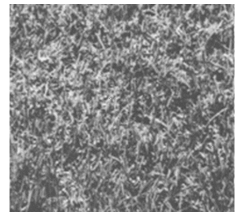
**3.** Explain which of the two filter-output features above appears to be the better featue?

**To train classifier, obtain sample data for first texture and both filters**

$infile1 := READBMP("grass1t1.gif")$

$rr := 0..rows(infile1) - 1 \quad cc := 0..cols(infile1) - 1 \quad rows(infile1) = 256 \quad cols(infile1) = 256$

$max(infile1) = 251 \quad min(infile1) = 67$



$rr := 0..rows(infile1) - 1 \quad cc := 0..cols(infile1) - 1$

$X1 := cfft(infile1)$

$xac1 := infile1 - mean(infile1) \quad Xac1 := cfft(xac1)$

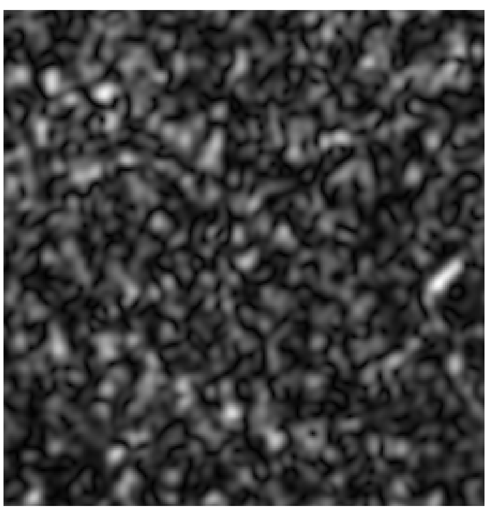
$Ya1 := \overrightarrow{(gab1 \cdot Xac1)} \quad Ya2 := \overrightarrow{(gab2 \cdot Xac1)}$

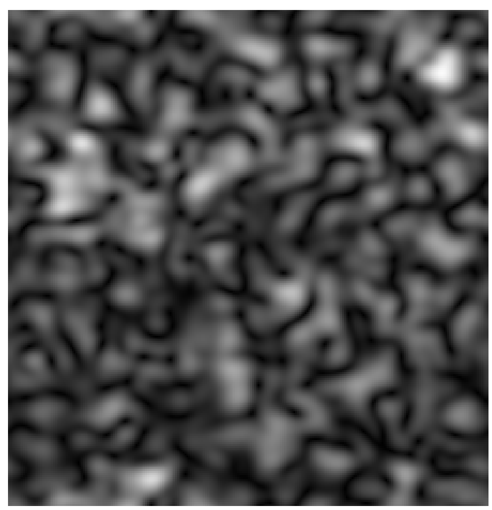
$yagab1 := icfft(Ya1) \quad yagab2 := icfft(Ya2)$

$yagab1 := \overrightarrow{|yagab1|} \quad yagab2 := \overrightarrow{|yagab2|}$

$minya1 := min(yagab1) \quad maxya1 := max(yagab1) \quad minya2 := min(yagab2) \quad maxya2 := max(yagab2)$

$minya1 = 0.012 \quad maxya1 = 9.501 \quad minya2 = 0.026 \quad maxya2 = 19.159$





**4.** Why does the left filter-output feature above appear to be less coarse than the one on the right (the left has finer resolution features)?

**Measure statistics of filtered samples for Bayesian Multivariate Gaussian classifier**

$oo_0 := yagab1 \qquad oo_1 := yagab2$

$$uu1 := \begin{pmatrix} mean(oo_0) \\ mean(oo_1) \end{pmatrix} \qquad uu1 = \begin{pmatrix} 2.181 \\ 5.454 \end{pmatrix}$$

$ssr := 0 .. rows(uu1) - 1 \qquad ssc := ssr$

$$RR1_{ssr, ssc} := mean\left(\overrightarrow{oo_{ssr} oo_{ssc}}\right) \qquad \Sigma\Sigma1 := RR1 - uu1 \cdot uu1^T$$

$$RR1 = \begin{pmatrix} 6.462 & 12.084 \\ 12.084 & 38.278 \end{pmatrix} \qquad \Sigma\Sigma1 = \begin{pmatrix} 1.705 & 0.188 \\ 0.188 & 8.528 \end{pmatrix}$$

**Gather exemplars of filtered samples for use in nearest-neighbor classifier**

$knn := 10 \qquad nn := 0 .. knn - 1$

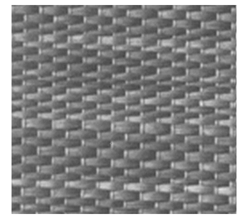
$$neighborsA_{nn} := \begin{pmatrix} yagab1_{10 \cdot nn+10,\, 10 \cdot nn+10} \\ yagab2_{10 \cdot nn+10,\, 10 \cdot nn+10} \end{pmatrix} \qquad neighborsA_0 = \begin{pmatrix} 1.607 \\ 6.227 \end{pmatrix}$$

### Obtain sample data for second texture and both filters

infile2 := READBMP("weave55t1.gif")

$rr := 0..rows(infile2) - 1$ $cc := 0..cols(infile2) - 1$ $rows(infile2) = 256$ $cols(infile2) = 256$

$max(infile2) = 228$ $min(infile2) = 66$



$rr := 0..rows(infile2) - 1$ $cc := 0..cols(infile2) - 1$

$X1 := cfft(infile2)$

$xac2 := infile2 - mean(infile2)$ $Xac2 := cfft(xac2)$

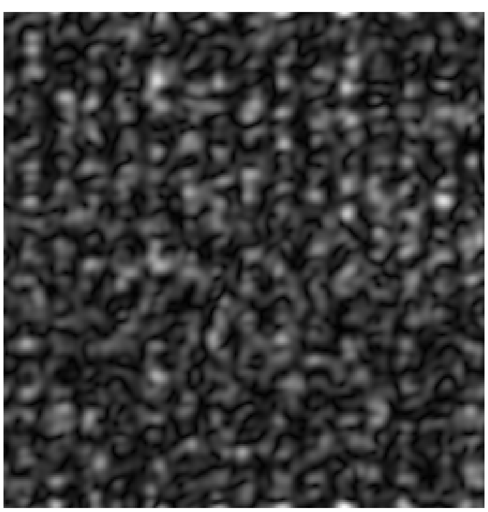
$Yb1 := \overrightarrow{(gab1 \cdot Xac2)}$ $Yb2 := \overrightarrow{(gab2 \cdot Xac2)}$

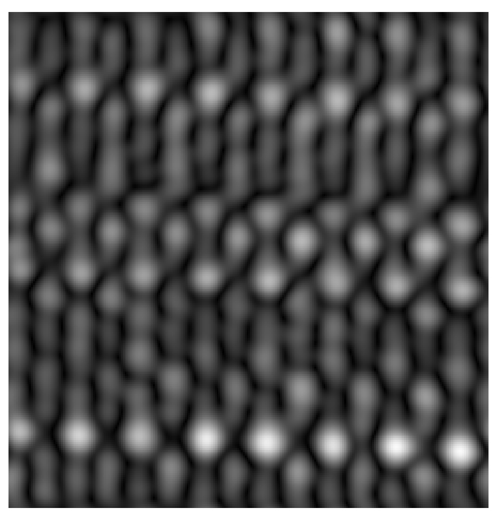
$ybgab1 := icfft(Yb1)$ $ybgab2 := icfft(Yb2)$

$ybgab1 := \overrightarrow{|ybgab1|}$ $ybgab2 := \overrightarrow{|ybgab2|}$

$minyb1 := min(ybgab1)$ $maxyb1 := max(ybgab1)$ $minyb2 := min(ybgab2)$ $maxyb2 := max(ybgab2)$

$minyb1 = 2.677 \times 10^{-3}$ $maxyb1 = 3.047$ $minyb2 = 0.017$ $maxyb2 = 27.146$





**Measure statistics of filtered samples for Bayesian Multivariate Gaussian classifier**

$oo_0 := ybgab1 \qquad oo_1 := ybgab2$

$$uu2 := \begin{pmatrix} mean(oo_0) \\ mean(oo_1) \end{pmatrix} \qquad uu2 = \begin{pmatrix} 0.642 \\ 7.464 \end{pmatrix}$$

$ssr := 0 .. rows(uu1) - 1 \qquad ssc := ssr$

$$RR2_{ssr, ssc} := mean\left(\overrightarrow{oo_{ssr} oo_{ssc}}\right) \qquad \Sigma\Sigma2 := RR2 - uu2 \cdot uu2^T$$

$$RR1 = \begin{pmatrix} 6.462 & 12.084 \\ 12.084 & 38.278 \end{pmatrix} \qquad \Sigma\Sigma1 = \begin{pmatrix} 1.705 & 0.188 \\ 0.188 & 8.528 \end{pmatrix}$$

**Gather exemplars of filtered samples for use in nearest-neighbor classifier**

$knn := 10 \qquad nn := 0 .. knn - 1$

$$neighborsB_{nn} := \begin{pmatrix} ybgab1_{10 \cdot nn+10,\, 10 \cdot nn+10} \\ ybgab2_{10 \cdot nn+10,\, 10 \cdot nn+10} \end{pmatrix} \qquad neighborsB_0 = \begin{pmatrix} 0.335 \\ 7.462 \end{pmatrix}$$

B-distance

$$B(u1, u2, s1, s2) := \left| \frac{1}{8} \cdot ((u1 - u2))^T \cdot \left(\frac{s1 + s2}{2}\right)^{-1} \cdot (u1 - u2) + \frac{1}{2} \cdot \ln\left[\frac{\left|\frac{(s1 + s2)}{2}\right|}{\sqrt{|s1|} \cdot \sqrt{|s2|}}\right] \right|$$

$$B(uu1, uu2, \Sigma\Sigma1, \Sigma\Sigma2) = 0.706 \qquad uu1 = \begin{pmatrix} 2.181 \\ 5.454 \end{pmatrix} \qquad \Sigma\Sigma1 = \begin{pmatrix} 1.705 & 0.188 \\ 0.188 & 8.528 \end{pmatrix}$$

$$\frac{1}{2} e^{-B(uu1, uu2, \Sigma\Sigma1, \Sigma\Sigma2)} = 0.247 \qquad uu2 = \begin{pmatrix} 0.642 \\ 7.464 \end{pmatrix} \qquad \Sigma\Sigma2 = \begin{pmatrix} 0.141 & -0.108 \\ -0.108 & 15.974 \end{pmatrix}$$

**Compute probabilities based on multivariate Gaussian**

mvgauss computes Gaussian pdf for vector x=[i0, i1 ...]$^T$ is comprised of feature images i0, i1, etc,
and given a Gaussin pdf with mean vector = u, and covariance matrix s

$$
\begin{aligned}
\mathrm{mvgauss}(x,u,s) := \; & N \leftarrow \mathrm{rows}(x) \\
& nn \leftarrow 0..N-1 \\
& dets \leftarrow |s| \\
& c1 \leftarrow \ln\left[\frac{1}{(2\cdot\pi)^{\frac{N}{2}}\cdot\sqrt{dets}}\right] \\
& sinv \leftarrow s^{-1} \\
& \text{for } rr \in 0..\mathrm{rows}(x_0)-1 \\
& \quad \text{for } cc \in 0..\mathrm{cols}(x_0)-1 \\
& \qquad \text{for } nn \in 0..\mathrm{rows}(x)-1 \\
& \qquad\quad u1_{nn} \leftarrow (x_{nn})_{rr,cc} \\
& \qquad temp \leftarrow c1 - 0.5\cdot\left[(u1-u)^T\cdot sinv\cdot(u1-u)\right] \\
& \qquad y_{rr,cc} \leftarrow temp_0 \\
& \qquad y_{rr,cc} \leftarrow e^{y_{rr,cc}} \\
& y
\end{aligned}
$$

generate the vector of feature images, fv

$fv_0 := ygab1 \qquad fv_1 := ygab2$

**Create probability images based on multivariate Gaussian**

$p1img := mvgauss(fv, uu1, \Sigma\Sigma1)$

$maxp1img := max(p1img)$

$maxp1img = 0.042$

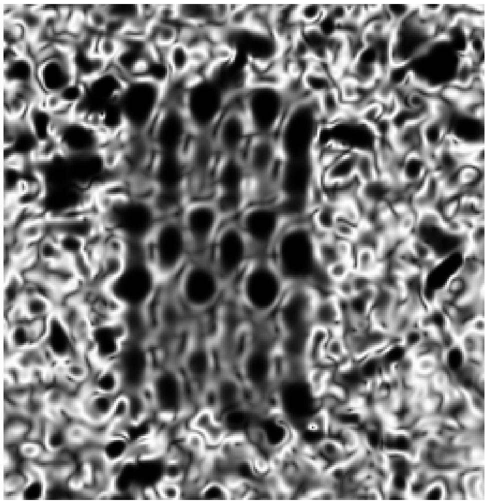
$min(p1img) = 4.677 \times 10^{-9}$

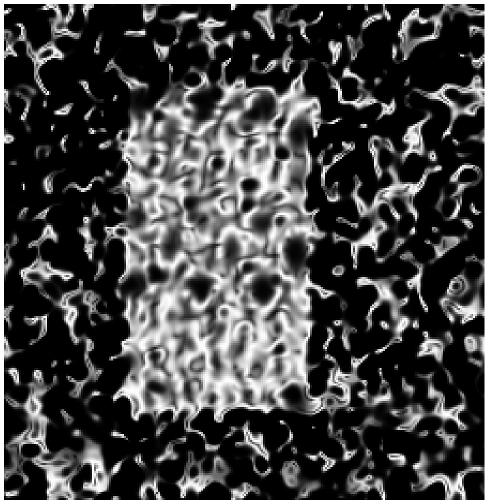
$p2img := mvgauss(fv, uu2, \Sigma\Sigma2)$

$maxp2img := max(p2img)$

$maxp2img = 0.106$

$min(p2img) = 0$

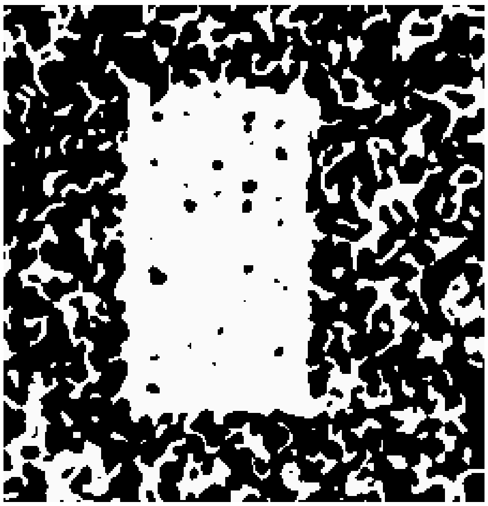


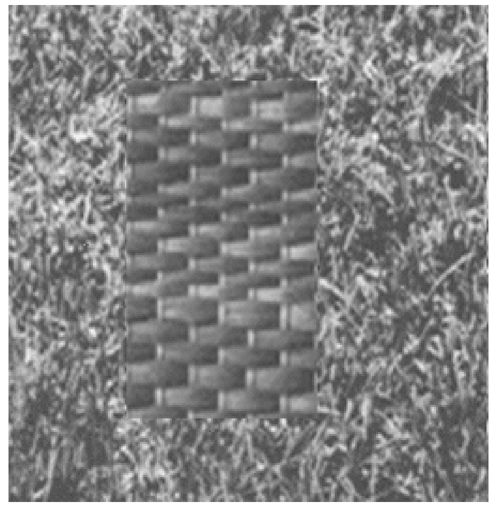


**5.** Explain what the lighter and darker regions mean in the right-hand image above?

**Create segmented image based on multivariate Gaussian**

$$segimg_{rr,cc} := \text{if}\left(p1img_{rr,cc} > p2img_{rr,cc}, 0, 250\right)$$





**6.** Find a way to measure the error on the segmented images, and report the measured error for the above Bayesian multivariate Gaussian segmentation.

**Nearest-neighbor classifier**

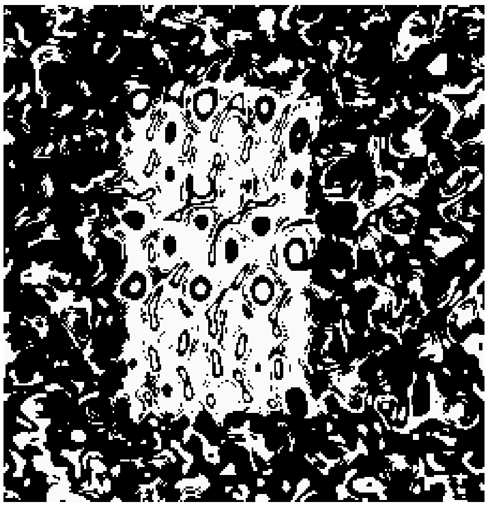
nearestneighbor computes the nearest neighbor,
where vector fv=[i0, i1 ...]$^T$ is comprised of feature images i0, i1, etc,
where vector ndat=[na, nb ...]$^T$ is comprised of vectors na, nb, etc,
where vectors na, nb, nc, correspond to classes a, b, c, etc
where na=[a1, a2 ...]$^T$ where each element a1, a2 is a sample feature vector for class a,
where nb=[b1, b2 ...]$^T$ where each element b1, b2 is a sample feature vector for class b,

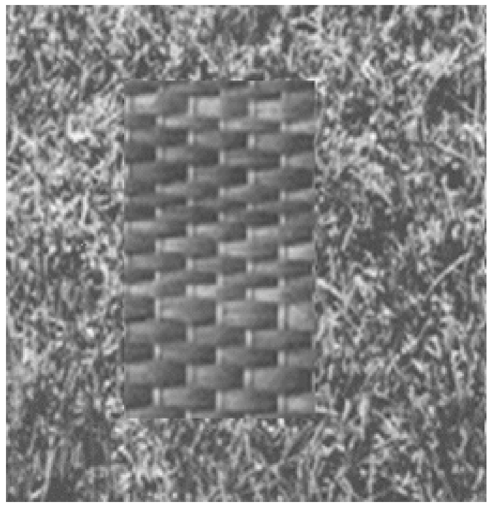
```
nearestneighbor(ndat, fv) := | nclass ← rows(ndat)
                             | nfeat ← rows(fv)
                             | nnei ← rows(ndat_0)
                             | for rr ∈ 0 .. rows(fv_0) − 1
                             |   for cc ∈ 0 .. cols(fv_0) − 1
                             |     | for nf ∈ 0 .. nfeat − 1
                             |     |   vv_nf ← (fv_nf)_rr,cc
                             |     | class ← 0
                             |     | dmin ← | vv − [(ndat_0)_0] |
                             |     | for nc ∈ 0 .. nclass − 1
                             |     |   for nn ∈ 0 .. nnei − 1
                             |     |     | d ← | vv − [(ndat_nc)_nn] |
                             |     |     | class ← nc  if d < dmin
                             |     |     | dmin ← d  if d < dmin
                             |     | y_rr,cc ← class
                             | y
```

**Create segmented image based on nearest-neighbor classifier**

$ndat_0 := neighborsA$ $\qquad$ $ndat_1 := neighborsB$

$nearneigh := 250 \cdot nearestneighbor(ndat, fv)$





**7.** Find a way to measure the error on the segmented images, and report the measured error for the above nearest-neighbor segmentation.