Segunda Prova de Processamento de Imagens

Gabriel D'assumpção de Carvalho

2025-07-31

Bibliotecas

```r
# install.packages("imager")
# install.packages("emmeans")

library(magick)
library(emmeans)

set.seed(42)
```

4º Questão

Introdução

Nesta atividade, trabalharemos com duas imagens convertidas para escala de cinza, nas quais será adicionado um ruído aditivo com distribuição normal $N(0, 0.01)$. O objetivo é aplicar o modelo quadrado greco-latino para investigar estatisticamente os fatores que influenciam a intensidade dos pixels, utilizando um janelamento de $5x5$ pixels.

Carregando a imagem

A imagem é carregada com a biblioteca `png`. Como possui três canais (RGB), foi convertida para tons de cinza pela média dos canais e padronizada para o intervalo [0, 1].

```r
url_books <-
"https://raw.githubusercontent.com/gabrieldadcarvalho/image_processing/main/exam/2_exam/livros.jpeg"
url_cobogo =
"https://raw.githubusercontent.com/gabrieldadcarvalho/image_processing/main/exam/2_exam/cobogo.jpeg"

books <- image_read(url_books)
cobogo <- image_read(url_cobogo)

# Diminuindo a resolução das imagens
books <- image_scale(books, "10%")
cobogo <- image_scale(cobogo, "5%")

books <- as.numeric(image_data(books))
cobogo <- as.numeric(image_data(cobogo))

dim_books <- dim(books)[1:2]
dim_cobogo <- dim(cobogo)[1:2]

print("Dimensões das imagens:")
```

[1] "Dimensões das imagens:"

```r
print(paste("Livros:", paste(dim(books), collapse = " x ")))
```

[1] "Livros: 79 x 74 x 3"

```r
print(paste("Cobogó:", paste(dim(cobogo), collapse = " x ")))
```

[1] "Cobogó: 80 x 79 x 3"

Podemos notar que as imagens possuem dimensões diferentes, sendo `books` com dimensões 79x74x3 e `cobogo` com dimensões 80x79x3. Com isso, vamos padronizar as imagens converter a imagem em tons de cinza utilizadno a média entre os três canais, aplicar o ruído e padronizar no intervalo [0, 1].

Processamento das imagens

```r
# Convertendo para tons de cinza
books <- (books[, , 1] + books[, , 2] + books[, , 3]) / 3
cobogo <- (cobogo[, , 1] + cobogo[, , 2] + cobogo[, , 3]) / 3
```

```r
# Padronizando as imagens para [0, 1]
books <- (books - min(books)) / (max(books) - min(books))
cobogo <- (cobogo - min(cobogo)) / (max(cobogo) - min(cobogo))
```

```r
# Aplicando o ruído
books <- books + rnorm(length(books), mean = 0, sd = sqrt(0.01))
cobogo <- cobogo + rnorm(length(cobogo), mean = 0, sd = sqrt(0.01))
```

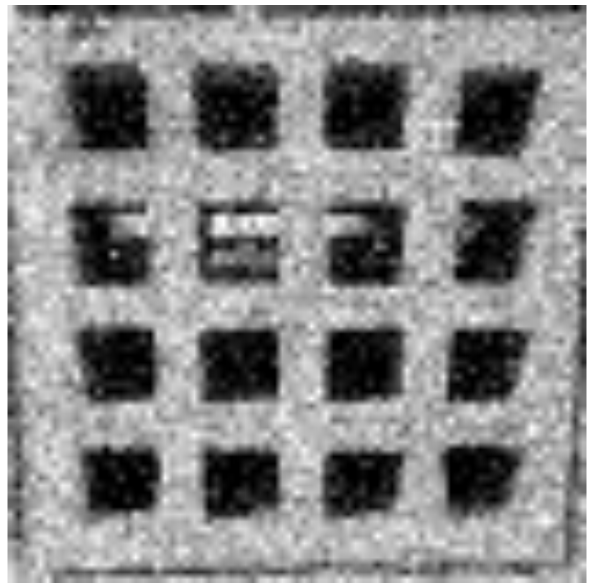
```r
# Truncando os valores para [0, 1]
books[books < 0] <- 0
books[books > 1] <- 1
cobogo[cobogo < 0] <- 0
cobogo[cobogo > 1] <- 1
```

```r
# Plotando as imagens
par(mfrow = c(1, 2), mar = c(2, 0, 3, 0))
plot(1:2,
  type = "n", xlab = "", ylab = "", xaxt = "n", yaxt = "n", bty = "n", asp = 1,
  main = "Livros"
)
rasterImage(books, 1, 1, 2, 2)

plot(1:2,
  type = "n", xlab = "", ylab = "", xaxt = "n", yaxt = "n", bty = "n", asp = 1,
  main = "Cobogó"
)
rasterImage(cobogo, 1, 1, 2, 2)
```

Livros

Cobogó



As figuras apresentadas ilustram dois cenários distintos para a análise de detecção de bordas. À esquerda, uma prateleira com oito livros (*Drácula, Frankenstein, Pinóquio, A Revolução dos Bichos, A Máquina do Tempo, Sandman, Alice no País das Maravilhas e Fundação*). À direita, um cobogó localizado no Centro de Ciências Sociais Aplicadas (CCSA) da UFPE. A resolução de ambas foi intencionalmente reduzida para viabilizar a análise por meio dos modelos ANOVA e Quadrado Greco-Latino, dada a sua elevada exigência computacional.

Modelo Quadrado Greco-Latino

O modelo quadrado greco-latino é dado por

$$y_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + \delta_l + \epsilon_{ijkl}$$

sendo:

- y_{ijkl}: intensidade do pixel na posição (i, j) da imagem;
- μ: média geral;
- α_i: efeito do fator i (linha);
- β_j: efeito do fator j (coluna);
- γ_k: efeito latino k (diagonal 135 graus);
- δ_l: efeito grego l (diagonal 45 graus);
- ϵ_{ijkl}: erro aleatório.

Para a aplicação do modelo, vamos considerar um janelamento de $5x5$ para estimação dos fatores, afim de discutir posteriormente duas janelas e seus ajustes.

```
m <- 5
n <- 5
```

```r
############## Fator Coluna ##############
# Fatores para a imagem de livros
a <- gl(n = n, k = m, labels = c(1, 2, 3, 4, 5))
contrasts(a) <- contr.sum
a
```

[1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5 attr(,"contrasts") [,1] [,2] [,3] [,4] 1 1 0 0 0 2 0 1 0 0 3 0 0 1 0 4 0 0 0 1 5 -1 -1 -1 -1 Levels: 1 2 3 4 5

```r
b <- rep(c(1, 2, 3, 4, 5), 5)
b <- as.factor(b)
contrasts(b) <- contr.sum
b
```

[1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 attr(,"contrasts") [,1] [,2] [,3] [,4] 1 1 0 0 0 2 0 1 0 0 3 0 0 1 0 4 0 0 0 1 5 -1 -1 -1 -1 Levels: 1 2 3 4 5

```r
# Fator com diagonal 135 graus
t <- c(
  1, 2, 3, 4, 5,
  5, 1, 2, 3, 4,
  4, 5, 1, 2, 3,
  3, 4, 5, 1, 2,
  2, 3, 4, 5, 1
)
t <- as.factor(t)
contrasts(t) <- contr.sum
t
```

[1] 1 2 3 4 5 5 1 2 3 4 4 5 1 2 3 3 4 5 1 2 2 3 4 5 1 attr(,"contrasts") [,1] [,2] [,3] [,4] 1 1 0 0 0 2 0 1 0 0 3 0 0 1 0 4 0 0 0 1 5 -1 -1 -1 -1 Levels: 1 2 3 4 5

```r
# Fator com diagonal 45 graus
d <- c(
  1, 2, 3, 4, 5,
  2, 3, 4, 5, 1,
  3, 4, 5, 1, 2,
  4, 5, 1, 2, 3,
  5, 1, 2, 3, 4
)
d <- as.factor(d)
contrasts(d) <- contr.sum
d
```

[1] 1 2 3 4 5 2 3 4 5 1 3 4 5 1 2 4 5 1 2 3 5 1 2 3 4 attr(,"contrasts") [,1] [,2] [,3] [,4] 1 1 0 0 0 2 0 1 0 0 3 0 0 1 0 4 0 0 0 1 5 -1 -1 -1 -1 Levels: 1 2 3 4 5

```r
contraste_a <- list(
  "A" = c(-3, -3, 2, 2, 2)
)
contraste_b <- list(
  "B" = c(-3, -3, 2, 2, 2)
)
contraste_t <- list(
  "T" = c(-3, -3, 2, 2, 2)
```

```
)
contraste_d <- list(
  "D" = c(-3, -3, 2, 2, 2)
)
```

Para o modelo Quadrado Greco-Latino, foram definidos os quatro fatores (linha, coluna e duas diagonais). Para cada um desses fatores, será utilizado um contraste específico, definido pelos coeficiente $\phi = \begin{pmatrix} -3 & -3 & 2 & 2 & 2 \end{pmatrix}$ com o objetivo de comparar a média dos níveis 1 e 2 com a média dos níveis 3, 4 e 5.

A análise seguirá um procedimento em duas etapas: primeiramente, um teste F-global será aplicado a cada janela de 5×5. Somente se este teste for significativo (indicando que os fatores influenciam a intensidade dos pixels), um segundo teste F (teste de forma) será realizado usando o contraste definido. Este segundo passo permite verificar especificamente se existe uma borda na interface entre os níveis 2 e 3 de cada fator.

```
i <- 1
j <- 1
alpha <- 0.05
# Acatando a sugestão do professor, a matriz de resultados foi reduzida em m vezes
result_books <- p_valor_f_books <- matrix(data = 0, nrow = dim_books[1] / m, ncol =
dim_books[2] / n)

result_cobogo <- p_valor_f_cobogo <- matrix(data = 0, nrow = dim_cobogo[1] / m, ncol =
dim_cobogo[2] / n)
```

Para armazenar os p-valores do teste F, que avalia a hipótese nula de que os fatores não influenciam a intensidade dos pixels, criaremos uma matriz de resultados, em que cada elemento (i, j) contém o p-valor associado à janela 5×5 posicionada em (i, j).

Como as imagens `books` e `cobogo` possuem dimensões (79×74) e (80×79), respectivamente, o janelamento 5×5 será aplicado em blocos lado a lado e sem sobreposição (*tiling*) sobre cada imagem. Isso resultará em matrizes de p-valores com dimensões (15×14) para `books` e (16×15) para `cobogo`, representando as regiões analisadas estatisticamente.

Imagem de livros

A presente seção detalha a aplicação do modelo Quadrado Greco-Latino para a detecção de bordas na imagem de livros. A análise será conduzida por meio do procedimento de teste em duas etapas para garantir a validade estatística dos resultados.

```
set.seed(42)
for (i in (1:as.integer(dim_books[1] / m)))
{
  print(i)
  for (j in (1:as.integer(dim_books[2] / n)))
  {
    y1 <- books[(1 + 5 * (i - 1)):(5 * i), (1 + 5 * (j - 1)):(5 * j)]
    y1 <- c(t(y1))
    df <- data.frame(y1, a, b, t, d)
    fit0 <- lm(y1 ~ a + b + t + d, data = df)
    p_valor_f_books[i, j] <- pf(summary(fit0)$fstatistic[1], summary(fit0)$fstatistic[2],
summary(fit0)$fstatistic[3], lower.tail = FALSE)
    if (p_valor_f_books[i, j] < alpha) {
      fit <- aov(y1 ~ a + b + t + d, data = df)
```

```
      emm_a <- emmeans(fit, ~a)
      emm_b <- emmeans(fit, ~b)
      emm_t <- emmeans(fit, ~t)
      emm_d <- emmeans(fit, ~d)
      resultado_contraste_1 <- contrast(emm_a, method = contraste_a)
      resultado_contraste_2 <- contrast(emm_b, method = contraste_b)
      resultado_contraste_3 <- contrast(emm_t, method = contraste_t)
      resultado_contraste_4 <- contrast(emm_d, method = contraste_d)
      p_valor_a <- summary(resultado_contraste_1)$p.value
      p_valor_b <- summary(resultado_contraste_2)$p.value
      p_valor_t <- summary(resultado_contraste_3)$p.value
      p_valor_d <- summary(resultado_contraste_4)$p.value
      p_valor <- min(p_valor_a, p_valor_b, p_valor_t, p_valor_d)
      if (p_valor < alpha) {
        result_books[i, j] <- 1
      }
    }
  }
}
```

Detectação de bordas [1] 1 [1] 2 [1] 3 [1] 4 [1] 5 [1] 6 [1] 7 [1] 8 [1] 9 [1] 10 [1] 11 [1] 12 [1] 13 [1] 14 [1] 15

```
par(mfrow = c(1, 2), mar = c(1, 1, 2, 1))

plot(1:2,
  type = "n", xlab = "", ylab = "", xaxt = "n", yaxt = "n", bty = "n", asp = 1,
  main = "Livros (Original com Ruído)"
)
rasterImage(books, 1, 1, 2, 2)

plot(1:2,
  type = "n", xlab = "", ylab = "", xaxt = "n", yaxt = "n", bty = "n", asp = 1,
  main = "Teste F Forma - Livros"
)
rasterImage(result_books, 1, 1, 2, 2)
```


```r
par(mfrow = c(1, 1))
```

A figura à direita exibe o mapa de bordas resultante da aplicação do procedimento de teste em duas etapas. As regiões brancas representam as janelas onde o modelo Quadrado Greco-Latino foi globalmente significativo e, subsequentemente, o contraste específico também detectou um padrão de borda.

O resultado demonstra a eficácia do método em capturar as principais estruturas da imagem, como as bordas verticais e as linhas horizontais das lombadas dos livros. A ocorrência de falsos positivos, pontos brancos em regiões presumivelmente homogêneas, evidencia a sensibilidade do modelo a duas fontes de variação: o ruído gaussiano adicionado e as texturas e grafismos nas lombadas, que constituem bordas em menor escala.

Imagem de cobogó

Em seguida, o mesmo método de detecção de bordas será avaliado na imagem do cobogó. O delineamento em Quadrado Greco-Latino será novamente utilizado, com a inferência baseada no procedimento de teste em duas etapas, para identificar as estruturas da imagem.

```r
set.seed(42)
for (i in (1:as.integer(dim_cobogo[1] / m)))
{
  print(i)
  for (j in (1:as.integer(dim_cobogo[2] / n)))
  {
    y1 <- cobogo[(1 + 5 * (i - 1)):(5 * i), (1 + 5 * (j - 1)):(5 * j)]
    y1 <- c(t(y1))
    df <- data.frame(y1, a, b, t, d)
    fit0 <- lm(y1 ~ a + b + t + d, data = df)
    p_valor_f_cobogo[i, j] <- pf(summary(fit0)$fstatistic[1],
summary(fit0)$fstatistic[2], summary(fit0)$fstatistic[3], lower.tail = FALSE)
    if (p_valor_f_cobogo[i, j] < alpha) {
      fit <- aov(y1 ~ a + b + t + d, data = df)
```

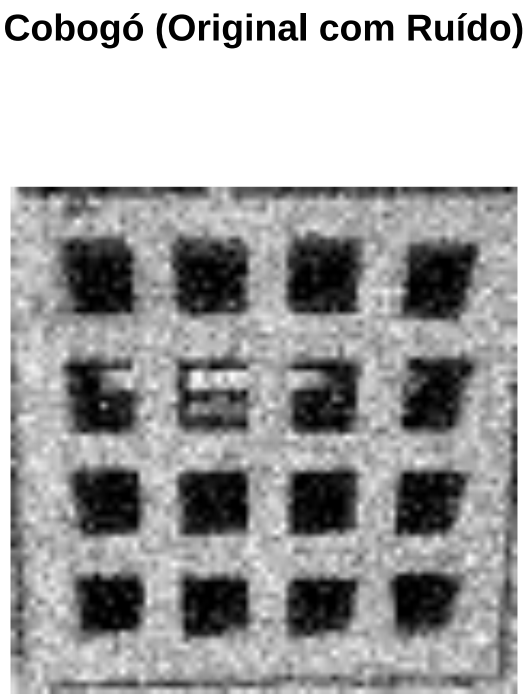
```
      emm_a <- emmeans(fit, ~a)
      emm_b <- emmeans(fit, ~b)
      emm_t <- emmeans(fit, ~t)
      emm_d <- emmeans(fit, ~d)
      resultado_contraste_1 <- contrast(emm_a, method = contraste_a)
      resultado_contraste_2 <- contrast(emm_b, method = contraste_b)
      resultado_contraste_3 <- contrast(emm_t, method = contraste_t)
      resultado_contraste_4 <- contrast(emm_d, method = contraste_d)
      p_valor_a <- summary(resultado_contraste_1)$p.value
      p_valor_b <- summary(resultado_contraste_2)$p.value
      p_valor_t <- summary(resultado_contraste_3)$p.value
      p_valor_d <- summary(resultado_contraste_4)$p.value
      p_valor <- min(p_valor_a, p_valor_b, p_valor_t, p_valor_d)
      if (p_valor < alpha) {
        result_cobogo[i, j] <- 1
      }
    }
  }
}
```

Teste F global [1] 1 [1] 2 [1] 3 [1] 4 [1] 5 [1] 6 [1] 7 [1] 8 [1] 9 [1] 10 [1] 11 [1] 12 [1] 13 [1] 14 [1] 15 [1] 16

```
par(mfrow = c(1, 2), mar = c(1, 1, 2, 1))

plot(1:2,
  type = "n", xlab = "", ylab = "", xaxt = "n", yaxt = "n", bty = "n", asp = 1,
  main = "Cobogó (Original com Ruído)"
)
rasterImage(cobogo, 1, 1, 2, 2)

plot(1:2,
  type = "n", xlab = "", ylab = "", xaxt = "n", yaxt = "n", bty = "n", asp = 1,
  main = "Teste F Forma - Cobogó"
)
rasterImage(result_cobogo, 1, 1, 2, 2)
```




```r
par(mfrow = c(1, 1))
```

A aplicação do modelo à imagem do cobogó, cujo resultado é apresentado à direita, oferece uma avaliação de sua performance em um cenário geométrica poligonal. O mapa de bordas evidencia a tentativa do método em delinear a estrutura quadradas do cobogó, capturando seus contornos com precisão.

Contudo, a sensibilidade do algoritmo também leva à detecção de falsos positivos. Além do ruído gaussiano, uma fonte primária de erro advém dos gradientes de intensidade gerados pelo fundo visível através dos vãos da estrutura. Esses padrões, contrários ao cobogó em si, são interpretados pelo modelo como bordas locais, resultando em detecções espúrias.

Dois Janelamento Aleatórios

Para uma melhor analise dos fatores, vamos aplicar o modelo Quadrado Greco-Latino em duas janelas aleatórias de 5×5 pixels, tanto na imagem de livros quanto na imagem de cobogó. Essas janelas serão selecionadas aleatoriamente dentro das dimensões da imagem, garantindo que não excedam os limites da matriz.

```r
set.seed(42)
# Coordenadas para a imagem de LIVROS
h1_books <- sample(1:(dim_books[1] - 4), 1)
w1_books <- sample(1:(dim_books[2] - 4), 1)
h2_books <- sample(1:(dim_books[1] - 4), 1)
w2_books <- sample(1:(dim_books[2] - 4), 1)

# Cordenadas para a imagem de livros
xleft1 <- 1 + (w1_books - 0.5) / dim_books[1]
xright1 <- 1 + (w1_books + 4 + 0.5) / dim_books[1]
ytop1 <- 2 - (h1_books - 0.5) / dim_books[2]
ybottom1 <- 2 - (h1_books + 4 + 0.5) / dim_books[2]
```

```r
xleft2 <- 1 + (w2_books - 0.5) / dim_books[1]
xright2 <- 1 + (w2_books + 4 + 0.5) / dim_books[1]
ytop2 <- 2 - (h2_books - 0.5) / dim_books[2]
ybottom2 <- 2 - (h2_books + 4 + 0.5) / dim_books[2]

# Janelas da imagem de LIVROS
books1 <- books[h1_books:(h1_books + 4), w1_books:(w1_books + 4)]
books2 <- books[h2_books:(h2_books + 4), w2_books:(w2_books + 4)]


books1_vec <- c(t(books1))
books2_vec <- c(t(books2))

df1 <- data.frame(y = books1_vec, a, b, t, d)
df2 <- data.frame(y = books2_vec, a, b, t, d)

# Configure a grade para 2 linhas e 2 colunas
par(mfrow = c(2, 2), mar = c(1, 1, 2, 1))

# --- Célula 1: Janela de Zoom 1 ---
plot(1:2, type = "n", xlab = "", ylab = "", xaxt = "n", yaxt = "n", bty = "n", asp = 1,
main = "Janela 5x5 - Livros 1")
rasterImage(books1, 1, 1, 2, 2)

# --- Célula 2: Janela de Zoom 2 ---
plot(1:2, type = "n", xlab = "", ylab = "", xaxt = "n", yaxt = "n", bty = "n", asp = 1,
main = "Janela 5x5 - Livros 2")
rasterImage(books2, 1, 1, 2, 2)

# --- Célula 3: Janela de Localização ---
plot(1:2, type = "n", xlab = "", ylab = "", xaxt = "n", yaxt = "n", bty = "n", asp = 1,
main = "Localização das Janelas")
rasterImage(books, 1, 1, 2, 2)
rect(
  xleft1, ybottom1, xright1, ytop1,
  border = "red",
  lwd = 3,
  col = NA
)
rect(
  xleft2, ybottom2, xright2, ytop2,
  border = "cyan",
  lwd = 3,
  col = NA
)

plot.new()

legend(
  "center", # Posição da legenda
  legend = c(
    "Janela 5x5 - Livros 1", # Texto da legenda
    "Janela 5x5 - Livros 2"
```

```r
  ),
  col = c("red", "cyan"), # Cores correspondentes
  pch = 15, # Usa um quadrado sólido (pch=15) para mostrar a cor
  bty = "n", # Remove a caixa ao redor da legenda
  cex = 1.2, # Aumenta o tamanho do texto e do símbolo
  title = "Legenda", # Título opcional para a legenda
  pt.cex = 1.5 # Aumenta o tamanho apenas do símbolo quadrado
)
```

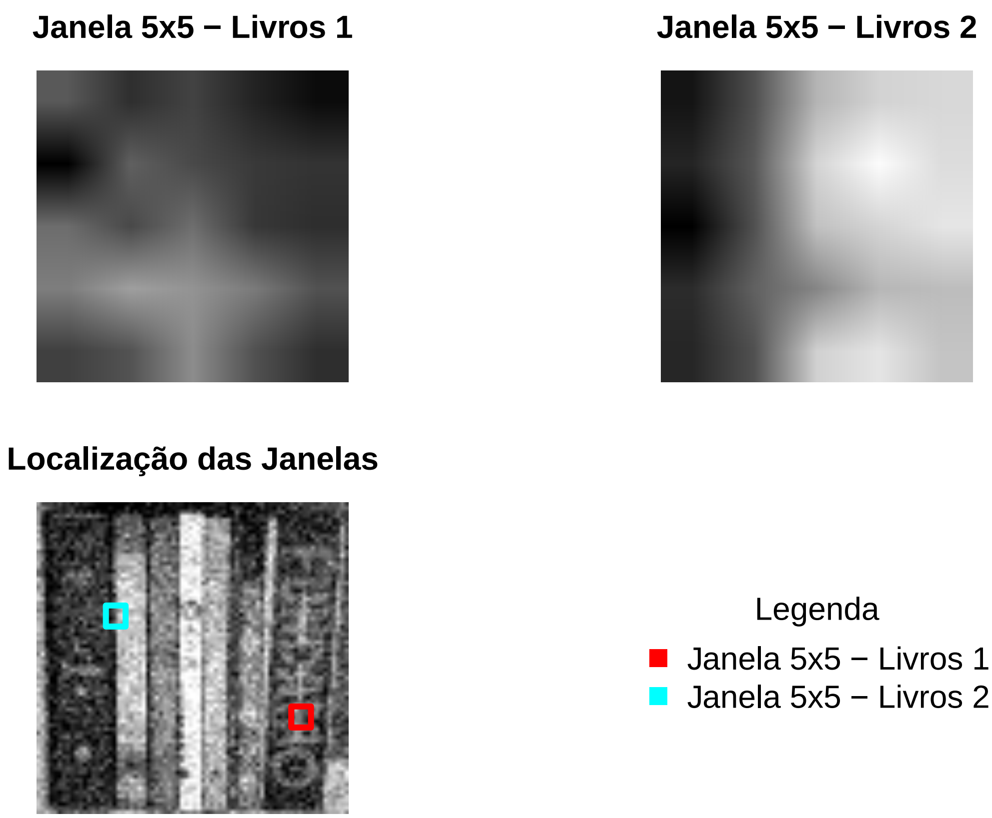


Imagem de livros

```r
# Resetar as configurações gráficas
par(mfrow = c(1, 1))
```

Na imagem acima podemos observar as duas janelas de 5×5 escolhidas aleatoriamente na imagem de livros. Sendo a primeira imagem localizada entre o primeiro (Drácula) e o segundo livro (Frankenstein) da esquerda para a direita, e a segunda imagem localizada na lombada do último livro (Fundação). Com isso, vamos aplicar o modelo Quadrado Greco-Latino para cada uma dessas janelas, a fim de verificar se os fatores influenciam a intensidade dos pixels.

```r
set.seed(42)
# Livros
fit_books1 <- lm(y ~ a + b + t + d, data = df1)
fit_books2 <- lm(y ~ a + b + t + d, data = df2)
```

```r
print("Livros - Janela 1")
```

[1] "Livros - Janela 1"

```
cat(capture.output(summary(fit_books1)), sep = "\n")
```

Call: lm(formula = y ~ a + b + t + d, data = df1)

Residuals: Min 1Q Median 3Q Max -0.117799 -0.029546 0.005066 0.038128 0.078406

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.3105465 0.0168951 18.381 7.9e-08 ***a1 -0.1144093 0.0337902 -3.386 0.009558*** *a2 -0.0904479 0.0337902 -2.677 0.028067*
a3 -0.0006274 0.0337902 -0.019 0.985641
a4 0.1902856 0.0337902 5.631 0.000492 ***b1 -0.0063752 0.0337902 -0.189 0.855050***
b2 0.0503822 0.0337902 1.491 0.174292
b3 0.1135048 0.0337902 3.359 0.009945 *b4 -0.0339414 0.0337902 -1.004 0.344562*
t1 0.0552305 0.0337902 1.635 0.140792
t2 -0.0570345 0.0337902 -1.688 0.129909
t3 -0.0118197 0.0337902 -0.350 0.735523
t4 0.0792123 0.0337902 2.344 0.047103
d1 0.0254991 0.0337902 0.755 0.472099
d2 -0.0303356 0.0337902 -0.898 0.395527
d3 0.0314754 0.0337902 0.931 0.378851
d4 -0.0325579 0.0337902 -0.964 0.363498
— Signif. codes: 0 '' ***0.001*** '' *0.01* '' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08448 on 8 degrees of freedom Multiple R-squared: 0.9062, Adjusted R-squared: 0.7185 F-statistic: 4.829 on 16 and 8 DF, p-value: 0.0149

A análise da primeira janela aleatória revela um resultado estatístico digno de nota. Inicialmente, ao se inspecionar os coeficientes individuais do modelo, os fatores de linha (`a`) e coluna (`b`) parecem ser os que mais contribuem para a variabilidade da imagem. Especificamente, os níveis a_2, a_4 (referentes às linhas 1 e 4) e b_3 (coluna 3) foram os que apresentaram os p-valores mais baixos.

Essa observação é consistente com o conteúdo visual da janela: a coluna 3 pode estar capturando a transição da capa escura de *Drácula* para a mais clara de *Frankenstein*, ou possivelmente um detalhe escuro no design da lombada deste último. As linhas 1 e 4, por sua vez, podem corresponder às variações de intensidade causadas pelos elementos gráficos nas lombadas.

Entretanto, a análise decisiva vem do teste F-global, que avalia a significância do modelo como um todo. Este teste resultou em um p-valor de aproximadamente 0,15 (15%). Como este valor está acima do nível de significância predefinido ($\alpha \approx 0.05$), a hipótese nula não pode ser rejeitada. Conclui-se, portanto, que não há evidências estatísticas suficientes para afirmar que os fatores, em conjunto, influenciam a intensidade dos pixels nesta janela específica.

A hipótese mais provável para essa aparente contradição é a localização da janela 5×5. Como pode ser visualizado na figura da Janela 1, ela cobre uma área predominantemente homogênea na capa do livro *Frankenstein*, o que provavelmente dilui o efeito das poucas bordas estruturadas presentes, levando à não significância do modelo como um todo.

```
print("Livros - Janela 2")
```

[1] "Livros - Janela 2"

```
cat(capture.output(summary(fit_books2)), sep = "\n")
```

Call: lm(formula = y ~ a + b + t + d, data = df2)

Residuals: Min 1Q Median 3Q Max -0.077556 -0.033317 0.005992 0.027300 0.079084

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.570974 0.014993 38.082 2.48e-10 ***a1 -0.015574 0.029987 -0.519 0.617566***

a2 0.067079 0.029987 2.237 0.055688 .
a3 -0.008625 0.029987 -0.288 0.780934
a4 -0.062083 0.029987 -2.070 0.072188 .
b1 -0.462936 0.029987 -15.438 3.08e-07 b2 -0.234313 0.029987 -7.814 5.17e-05 ***b3 0.159290 0.029987 5.312 0.000718*** b4 0.283107 0.029987 9.441 1.30e-05 *** t1 -0.036035 0.029987 -1.202 0.263848
t2 0.003755 0.029987 0.125 0.903448
t3 0.048027 0.029987 1.602 0.147908
t4 0.008908 0.029987 0.297 0.773988
d1 -0.046382 0.029987 -1.547 0.160507
d2 0.007783 0.029987 0.260 0.801762
d3 -0.030765 0.029987 -1.026 0.334930
d4 0.013882 0.029987 0.463 0.655747
— Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ” 0.05 ‘.’ 0.1 ’ ’ 1

Residual standard error: 0.07497 on 8 degrees of freedom Multiple R-squared: 0.9808, Adjusted R-squared: 0.9424 F-statistic: 25.55 on 16 and 8 DF, p-value: 4.021e-05

Em contrapartida, a análise da segunda janela aleatória apresenta um resultado marcadamente diferente e conclusivo. Neste caso, observou-se que todos os níveis do fator coluna (`b`) são estatisticamente significativos, com p-valores individuais abaixo de 0.01 (1%). Isso indica uma forte variação na intensidade dos pixels ao longo das colunas da janela.

Confirmando essa forte evidência, o teste F-global do modelo resultou em um p-valor extremamente baixo, de aproximadamente 0.00004. Estando bem abaixo do nível de significância ($\alpha = 0.05$), a hipótese nula é rejeitada com segurança. Conclui-se, portanto, que os fatores, em conjunto, exercem uma influência significativa na intensidade dos pixels desta janela.

A localização da Janela 2 explica claramente esses resultados. Ela está posicionada sobre a lombada do livro *Fundação*, abrangendo tanto o design gráfico do título quanto uma porção do fundo preto à esquerda. A presença dessas texturas contrastantes, os detalhes do design e a transição para o fundo escuro, cria fortes gradientes verticais que são eficientemente capturados pelo modelo como efeitos de coluna altamente significativos.

```r
set.seed(32)
# Coordenadas para a imagem do COBOGÓ
h1_cobogo <- sample(1:(dim_cobogo[1] - 4), 1)
w1_cobogo <- sample(1:(dim_cobogo[2] - 4), 1)
h2_cobogo <- sample(1:(dim_cobogo[1] - 4), 1)
w2_cobogo <- sample(1:(dim_cobogo[2] - 4), 1)

# Cordenadas para a imagem de livros
xleft1 <- 1 + (w1_cobogo - 0.5) / dim_cobogo[1]
xright1 <- 1 + (w1_cobogo + 4 + 0.5) / dim_cobogo[1]
ytop1 <- 2 - (h1_cobogo - 0.5) / dim_cobogo[2]
ybottom1 <- 2 - (h1_cobogo + 4 + 0.5) / dim_cobogo[2]

xleft2 <- 1 + (w2_cobogo - 0.5) / dim_cobogo[1]
xright2 <- 1 + (w2_cobogo + 4 + 0.5) / dim_cobogo[1]
ytop2 <- 2 - (h2_cobogo - 0.5) / dim_cobogo[2]
ybottom2 <- 2 - (h2_cobogo + 4 + 0.5) / dim_cobogo[2]

# Janelas da imagem do COBOGÓ
cobogo1 <- cobogo[h1_cobogo:(h1_cobogo + 4), w1_cobogo:(w1_cobogo + 4)]
cobogo2 <- cobogo[h2_cobogo:(h2_cobogo + 4), w2_cobogo:(w2_cobogo + 4)]
```

```r
cobogo1_vec <- c(t(cobogo1))
cobogo2_vec <- c(t(cobogo2))

df3 <- data.frame(y = cobogo1_vec, a, b, t, d)
df4 <- data.frame(y = cobogo2_vec, a, b, t, d)

par(mfrow = c(2, 2), mar = c(1, 1, 2, 1))

# --- Célula 1: Janela de Zoom 1 ---
plot(1:2, type = "n", xlab = "", ylab = "", xaxt = "n", yaxt = "n", bty = "n", asp = 1,
main = "Janela 5x5 - Cobogó 1")
rasterImage(cobogo1, 1, 1, 2, 2)

# --- Célula 2: Janela de Zoom 2 ---
plot(1:2, type = "n", xlab = "", ylab = "", xaxt = "n", yaxt = "n", bty = "n", asp = 1,
main = "Janela 5x5 - Cobogó 2")
rasterImage(cobogo2, 1, 1, 2, 2)

# --- Célula 3: Janela de Localização ---
plot(1:2, type = "n", xlab = "", ylab = "", xaxt = "n", yaxt = "n", bty = "n", asp = 1,
main = "Localização das Janelas")
rasterImage(cobogo, 1, 1, 2, 2)
rect(
  xleft1, ybottom1, xright1, ytop1,
  border = "red",
  lwd = 3,
  col = NA
)
rect(
  xleft2, ybottom2, xright2, ytop2,
  border = "cyan",
  lwd = 3,
  col = NA
)

plot.new()

legend(
  "center", # Posição da legenda
  legend = c(
    "Janela 5x5 - Cobogó 1", # Texto da legenda
    "Janela 5x5 - Cobogó 2"
  ),
  col = c("red", "cyan"), # Cores correspondentes
  pch = 15, # Usa um quadrado sólido (pch=15) para mostrar a cor
  bty = "n", # Remove a caixa ao redor da legenda
  cex = 1.2, # Aumenta o tamanho do texto e do símbolo
  title = "Legenda", # Título opcional para a legenda
  pt.cex = 1.5 # Aumenta o tamanho apenas do símbolo quadrado
)
```

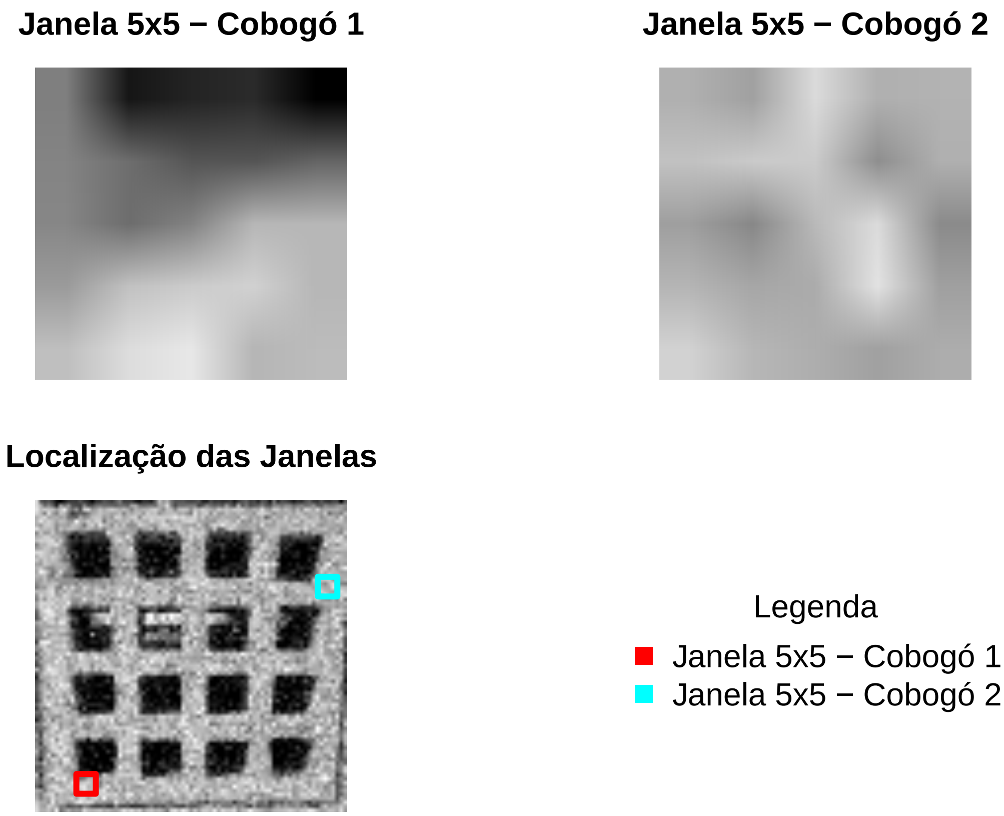


Imagem de cobogó

```
par(mfrow = c(1, 1))
```

Para a análise da imagem do cobogó, foi adotada a mesma metodologia de amostragem aleatória. As duas janelas 5×5 selecionadas, visualizadas na figura acima, representam dois cenários de grande interesse analítico.

A primeira janela incide sobre uma intersecção dos eixos ortogonais do cobogó. Para esta região de alta complexidade geométrica, a hipótese é que, além dos efeitos de linha e coluna, os fatores diagonais também se mostrem estatisticamente significativos.

Em forte contraste, a segunda janela foi posicionada sobre uma área de concreto com textura visivelmente homogênea. Este segundo caso serve como um importante controle, onde a expectativa é que o modelo não encontre mudanças significativas na intensidade dos pixels, demonstrando assim sua capacidade de distinguir regiões uniformes de regiões com bordas estruturadas.

```
# Cobogó
set.seed(42)
fit_cobogo1 <- lm(y ~ a + b + t + d, data = df3)
fit_cobogo2 <- lm(y ~ a + b + t + d, data = df4)
```

```
print("Cobogó - Janela 1")
```

[1] "Cobogó - Janela 1"

```
cat(capture.output(summary(fit_cobogo1)), sep = "\n")
```

Call: lm(formula = y ~ a + b + t + d, data = df3)

Residuals: Min 1Q Median 3Q Max -0.12020 -0.03604 0.00277 0.03580 0.10229

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.5401660 0.0210715 25.635 5.75e-09 ***a1 -0.3617793 0.0421430 -8.585 2.62e-05*** a2 -

0.1373839 0.0421430 -3.260 0.01153 *
a3 0.0421146 0.0421430 0.999 0.34690
a4 0.2025947 0.0421430 4.807 0.00134 ** b1 0.0397575 0.0421430 0.943 0.37309
b2 -0.0248151 0.0421430 -0.589 0.57222
b3 0.0004084 0.0421430 0.010 0.99250
b4 0.0108722 0.0421430 0.258 0.80293
t1 0.0605754 0.0421430 1.437 0.18855
t2 -0.0186586 0.0421430 -0.443 0.66968
t3 -0.0065871 0.0421430 -0.156 0.87967
t4 0.0123271 0.0421430 0.293 0.77734
d1 0.1152430 0.0421430 2.735 0.02566 *
d2 0.0705070 0.0421430 1.673 0.13286
d3 -0.0338954 0.0421430 -0.804 0.44447
d4 -0.0833856 0.0421430 -1.979 0.08322 .
— Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ” 0.05 ‘.’ 0.1 ’ ’ 1

Residual standard error: 0.1054 on 8 degrees of freedom Multiple R-squared: 0.9437, Adjusted R-squared: 0.8312 F-statistic: 8.385 on 16 and 8 DF, p-value: 0.002402

Os resultados desta janela validam a hipótese inicial de que os efeitos diagonais seriam proeminentes. O teste F-global do modelo é evidentemente significativo ($p \approx 0.0024$), um resultado impulsionado principalmente pelos fatores de linhas e diagonais. Especificamente, o nível d_1 apresentou forte significância ($p \approx 0.026$), enquanto d_4 se mostrou marginalmente significativo ($p \approx 0.083$).

Essa combinação de resultados evidencia a capacidade do modelo em capturar a complexa geometria da intersecção do cobogó. A significância geral é atribuída à transição de alta frequência entre a borda sólida da estrutura e seu vão, um padrão que o modelo corretamente identificou como sendo influenciado por múltiplos fatores direcionais.

```r
print("Cobogó - Janela 2")
```

[1] “Cobogó - Janela 2”

```r
cat(capture.output(summary(fit_cobogo2)), sep = "\n")
```

Call: lm(formula = y ~ a + b + t + d, data = df4)

Residuals: Min 1Q Median 3Q Max -0.152274 -0.035318 0.005526 0.050949 0.117227

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.7024811 0.0249342 28.173 2.72e-09 *** a1 0.0114829 0.0498683 0.230 0.824
a2 0.0191317 0.0498683 0.384 0.711
a3 -0.0444710 0.0498683 -0.892 0.399
a4 0.0094835 0.0498683 0.190 0.854
b1 0.0194185 0.0498683 0.389 0.707
b2 -0.0360221 0.0498683 -0.722 0.491
b3 0.0464045 0.0498683 0.931 0.379
b4 0.0240042 0.0498683 0.481 0.643
t1 0.0538216 0.0498683 1.079 0.312
t2 0.0462331 0.0498683 0.927 0.381
t3 -0.0236284 0.0498683 -0.474 0.648
t4 -0.0344410 0.0498683 -0.691 0.509
d1 0.0251721 0.0498683 0.505 0.627
d2 -0.0005267 0.0498683 -0.011 0.992
d3 0.0052176 0.0498683 0.105 0.919
d4 -0.0209060 0.0498683 -0.419 0.686
— Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ” 0.05 ‘.’ 0.1 ’ ’ 1

Residual standard error: 0.1247 on 8 degrees of freedom Multiple R-squared: 0.4408, Adjusted R-squared: -0.6776 F-statistic: 0.3941 on 16 and 8 DF, p-value: 0.9462

Confirmando o papel desta janela como um caso de controle, os resultados corroboram integralmente a hipótese inicial. A análise da região homogênea de concreto produziu um teste F-global com p-valor de aproximadamente 0.95, um valor que indica ausência total de significância estatística.

Portanto, para esta janela, não há evidências que justifiquem rejeitar a hipótese nula. Este resultado nulo é crucial, pois demonstra a especificidade do método: sua capacidade de corretamente identificar a ausência de estruturas direcionais em áreas de textura uniforme, evitando falsos positivos.