Exercício 10 – Reconhecimento de Padrões

Aluno: Giovanni Martins de Sá Júnior – 2017001850

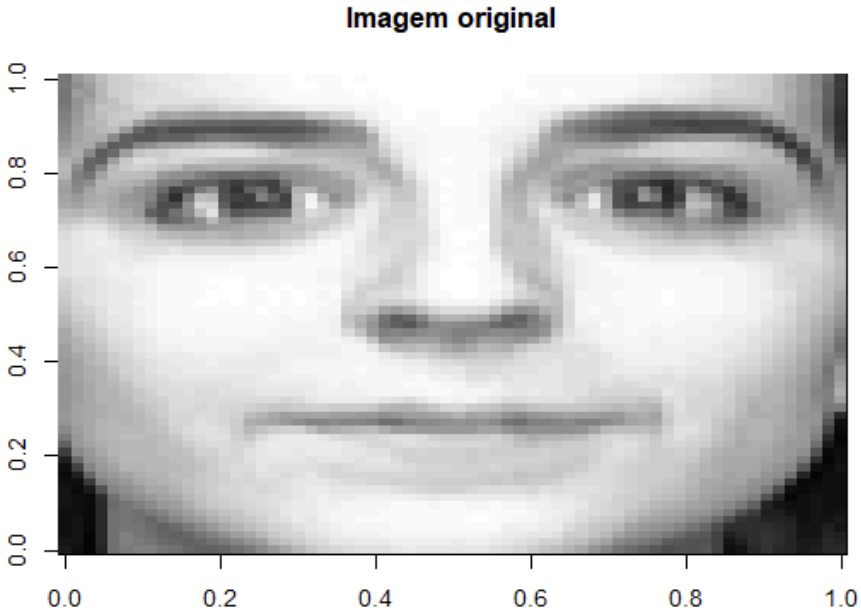
Neste exercício, serão aplicados os conceitos de CNNs, no qual foram aplicados Filtros de Convolução em cima da base de dados Olivetti. Nesse sentido, foi realizada a importação da base de dados e logo em seguida, implementada a função na qual mostra uma das imagens presentes no dataset. Com isso, após a escolha, a imagem passará por diferentes aplicações de filtros, utilizando a operação de convolução:

1. Filtro de bordas:
   $\mathrm{f} = \mathrm{matrix}(\mathrm{c}(-1,-1,-1,-1,8,-1,-1,-1,-1), \mathrm{nrow} = 3, \mathrm{ncol} = 3)$
2. Filtro de linhas verticais:
   $\mathrm{f} = \mathrm{matrix}(\mathrm{c}(1,2,1,0,0,0,-1,-2,-1), \mathrm{nrow} = 3, \mathrm{ncol} = 3)$
3. Filtro de linhas horizontais:
   $\mathrm{f} = \mathrm{matrix}(\mathrm{c}(1,0,-1,2,0,-2,1,0,-1), \mathrm{nrow} = 3, \mathrm{ncol} = 3)$
4. Filtro Sharpen:
   $\mathrm{f} = \mathrm{matrix}(\mathrm{c}(0,-1,0,-1,5,-1,0,-1,0), \mathrm{nrow} = 3, \mathrm{ncol} = 3)$

A seguir, é apresentada a implementação do exercício:

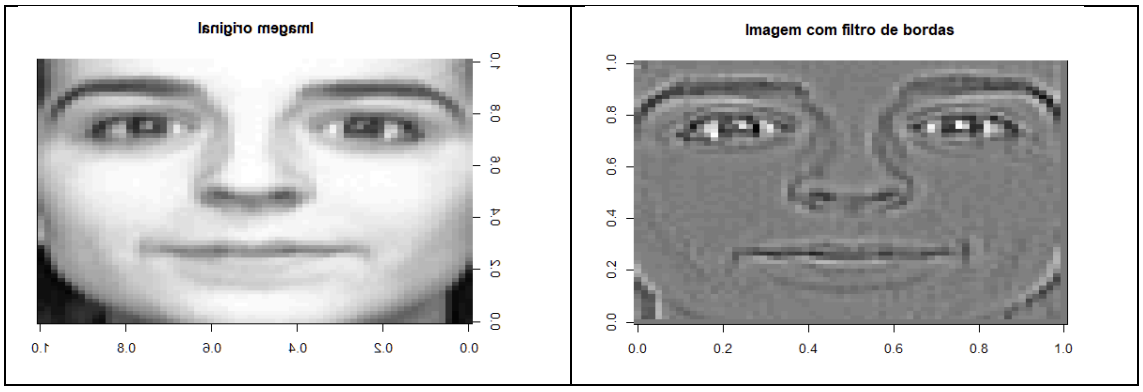
Importação da Base de Dados e Escolha da Imagem:

```
rm(list = ls())
require(RnavGraphImageData)

# Carregando a Base de dados
data( faces )
faces <- t( faces )
MostraImagem <- function( x, k, titulo )
{
  rotate <- function(x) t( apply(x, 2, rev) )
  imagem <- matrix( x, nrow=k )
  cor <- rev( gray(50:1/50) )
  image( rotate( imagem ), col=cor, main=titulo)
  return(imagem)
}

tituloImagemOriginal <- "Imagem original"
numeroImgOriginal <- 100
img = MostraImagem(faces[numeroImgOriginal,], 64, tituloImagemOriginal)

dimx = dim(img)[1]
dimy = dim(img)[2]

tf = 3
```



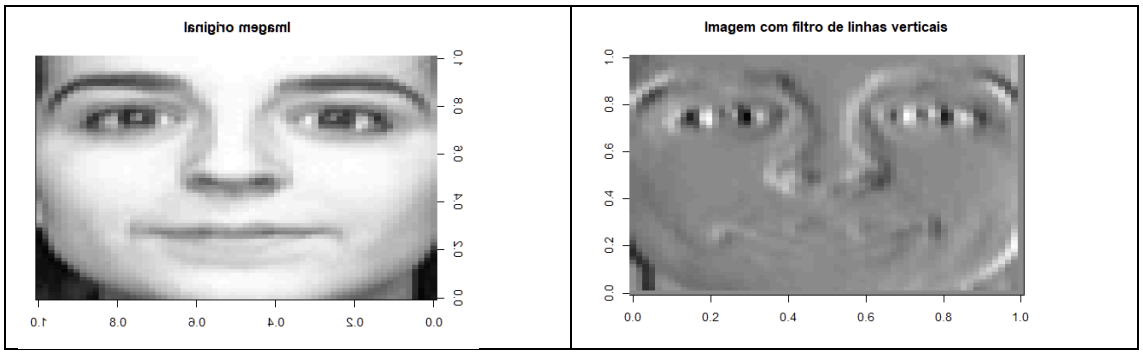
Filtro de Bordas

```
# Filtro de bordas
f = matrix(c(-1,-1,-1,-1,8,-1,-1,-1,-1), nrow = tf, ncol = tf)
M = matrix(0, nrow=(dimx-2), ncol=(dimy-2))

for (l in 1: (dimx-tf)){
  for (c in 1: (dimy-tf)){
    M[l,c]=sum(img[l:(l+2), c:(c+2)] * f)
  }
}

MostraImagem(M, 62, "Imagem com filtro de bordas")
```



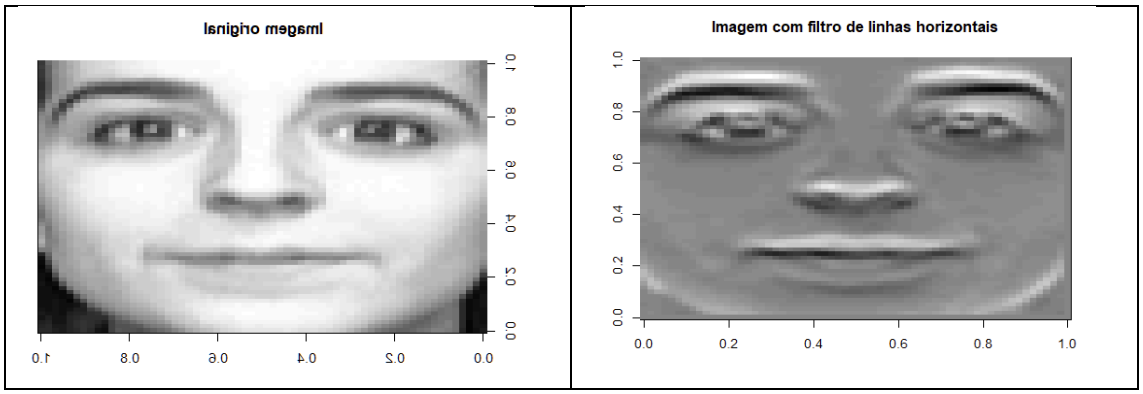
Filho de Linhas Verticais

```
# Filtro de linhas verticais
f = matrix(c(1,2,1,0,0,0,-1,-2,-1), nrow = tf, ncol = tf)
M = matrix(0, nrow=(dimx-2), ncol=(dimy-2))

for (l in 1: (dimx-tf)){
  for (c in 1: (dimy-tf)){
    M[l,c]=sum(img[l:(l+2), c:(c+2)] * f)
  }
}
MostraImagem(faces[numeroImgOriginal,], 64, tituloImagemOriginal)
MostraImagem(M, 62, "Imagem com filtro de linhas verticais")
```



Filtro de Linhas Horizontais

```
# Filtro de linhas horizontais
f = matrix(c(1,0,-1,2,0,-2,1,0,-1), nrow = tf, ncol = tf)
M = matrix(0, nrow=(dimx-2), ncol=(dimy-2))

for (l in 1: (dimx-tf)){
  for (c in 1: (dimy-tf)){
    M[l,c]=sum(img[l:(l+2), c:(c+2)] * f)
  }
}
MostraImagem(faces[numeroImgOriginal, ], 64, tituloImagemOriginal)
MostraImagem(M, 62, "Imagem com filtro de linhas horizontais")
```



Filtro Sharpen

```
# Filtro Sharpen
f = matrix(c(0,-1,0,-1,5,-1,0,-1,0), nrow = tf, ncol = tf)
M = matrix(0, nrow=(dimx-2), ncol=(dimy-2))

for (l in 1: (dimx-tf)){
  for (c in 1: (dimy-tf)){
    M[l,c]=sum(img[l:(l+2), c:(c+2)] * f)
  }
}
MostraImagem(faces[numeroImgOriginal,], 64, tituloImagemOriginal)
MostraImagem(M, 62, "Imagem com filtro Sharpen")
```