

Image compression using various kinds of dimension reduction techniques

2023-02-10

Contents

RGB image compression	1
Image compression using principle component analysis (PCA)	2
Image compression using Singular value decomposition	2
Image compression using Factor analysis	2
Image compression using ICA	3
Image compression using Kmeans	3
RGB image comparison	4
REAL VS PCA	4
REAL VS SVD	4
REAL VS FACTOR	5
REAL VS ICA	5
REAL VS Kmeans	5
Raster image and it's compression	5
Raster image compression for PCA	6
Raster Image compression using Factor analysis	6
Raster image compression using Singular value decomposition	6
Image compression using ICA	6
Image compression using Kmeans	8
Comparison Rastering image	8
Real vs PCA	8
Real vs Factor analysis	8
Real vs SVD	9
Real vs ICA	9
Real vs Kmeans	10
Extra	10
library(jpeg)	
library(fastICA)	
library(psych)	
library(knitr)	

RGB image compression

```
tiger <- readJPEG("C:/Users/W.C/Downloads/programming/PCA image/tiger.jpg")
```

Image compression using principle component analysis (PCA).

```
Image_pca <- function(Image, components ){
  r <- Image[, , 1]
  g <- Image[, , 2]
  b <- Image[, , 3]
  pcr <- prcomp(r)
  pcg <- prcomp(g)
  pcb <- prcomp(b)
  pr_list <- list(pcr, pcg, pcb)
  pc_im <- sapply(
    pr_list,  function(i){
      im_tr <- i$x[, 1:components] %*% t(i$rotation[, 1:components])
    },
    simplify = "array"
  )
  return( writeJPEG(pc_im,paste0("PCAIimage",components,".jpg")))
}

Image_pca(Image = tiger, 300)
```

Image compression using Singular value decomposition

```
Image_svd <- function(Image, components ){
  r <- Image[, , 1]
  g <- Image[, , 2]
  b <- Image[, , 3]
  pcr <- svd(r)
  pcg <- svd(g)
  pcb <- svd(b)
  pr_list <- list(pcr, pcg, pcb)
  pc_im <- sapply(
    pr_list,  function(i){
      im_tr <- i$u[, 1:components] %*% diag(i$d[1:components]) %*% t(i$v[, 1:components])
    },
    simplify = "array"
  )
  return( writeJPEG(pc_im,paste0("SVDimage",components,".jpg")))
}

Image_svd(Image = tiger, 300)
```

Image compression using Factor analysis

```
Image_factor <- function(Image, factors ){
  r <- Image[, , 1]
  g <- Image[, , 2]
  b <- Image[, , 3]
  pcr <- fa(r,nfactors = factors, scores = "Anderson")
  pcg <- fa(g,nfactors = factors, scores = "Anderson")
  pcb <- fa(b,nfactors = factors, scores = "Anderson")
```

```

pr_list <- list(pcr, pcg, pcb)
pc_im <- sapply(
  pr_list, function(i){
    img <- i$scores %*% t(i$loadings)
  },
  simplify = "array"
)
return( writeJPEG(pc_im,paste0("FACTORimage",factors,".jpg")))
}

Image_factor(Image = tiger, 300)

```

Image compression using ICA

```

Image_ica <- function(Image, components ){
  r <- Image[, , 1]
  g <- Image[, , 2]
  b <- Image[, , 3]
  pcr <- fastICA(r, n = components )
  pcg <- fastICA(g, n = components)
  pcb <- fastICA(b, n = components)
  pr_list <- list(pcr, pcg, pcb)
  pc_im <- sapply(
    pr_list, function(i){
      img <- i$S %*% i$A
    },
    simplify = "array"
  )
  return( writeJPEG(pc_im,paste0("ICAimage",components,".jpg")))
}

Image_ica(Image = tiger, 300)

```

Image compression using Kmeans

```

Image_kmeans <- function(Image, components ){
  r <- Image[, , 1]
  g <- Image[, , 2]
  b <- Image[, , 3]
  pcr <- kmeans(r,centers = components)
  pcg <- kmeans(g,centers = components)
  pcb <- kmeans(b,centers = components)
  pr_list <- list(pcr, pcg, pcb)
  pc_im <- sapply(
    pr_list, function(i){
      img <- i$centers[i$cluster,]
    },
    simplify = "array"
  )

```

```

    return( writeJPEG(pc_im,paste0("Kmeansimage",components,".jpg")))
}

Image_kmeans(Image = tiger, 300)

```

RGB image comparison

REAL VS PCA

```

components = 300
include_graphics("tiger.jpg")
include_graphics(paste0("PCAIimage",components,".jpg"))

```



Figure 1: REAL vs PCA

REAL VS SVD

```

include_graphics("tiger.jpg")
include_graphics(paste0("SVDimage",components,".jpg"))

```



Figure 2: REAL vs SVD

REAL VS FACTOR

```
include_graphics("tiger.jpg")
include_graphics(paste0("FACTORimage",components,".jpg"))
```



Figure 3: REAL vs FACTOR

REAL VS ICA

```
include_graphics("tiger.jpg")
include_graphics(paste0("ICAimage",components,".jpg"))
```



Figure 4: REAL vs ICA

REAL VS Kmeans

```
include_graphics("tiger.jpg")
include_graphics(paste0("Kmeansimage",components,".jpg"))
```

Raster image and it's compression

```
library(raster)
im <- raster("C:/Users/W.C/Downloads/programming/PCA image/tiger.jpg")
```



Figure 5: REAL vs KMNENAS

```
img.flip<-flip(im, direction = "y")
rasterimg<-t(as.matrix(img.flip))
```

Raster image compression for PCA

```
pca_func <- function(Rimage, components){
  image_pca <- prcomp(Rimage)
  compressed_image_matrix <- image_pca$x[, 1:components] %*% t(image_pca$rotation[, 1:components])
  return(image(compressed_image_matrix))
}

pca_func(rasterimg, 100)
```

Raster Image compression using Factor analysis

```
fa_func <- function(Rimage, components){
  image_factors <- fa(Rimage, nfactors = components)
  compressed_image_matrix <- image_factors$scores %*% t(image_factors$loadings)
  return(image(compressed_image_matrix))
}
# fa_func(rasterimg, 100)
```

Raster image compression using Singular value decomposition

```
svd_func <- function(Rimage, components){
  imsvd <- svd(Rimage)
  compressed_image_matrix <- imsvd$u[,1:components] %*% diag(imsvd$d[1:components]) %*% t(imsvd$v[,1:components])
  return(image(compressed_image_matrix))
}
# svd_func(rasterimg, 100)
```

Image compression using ICA

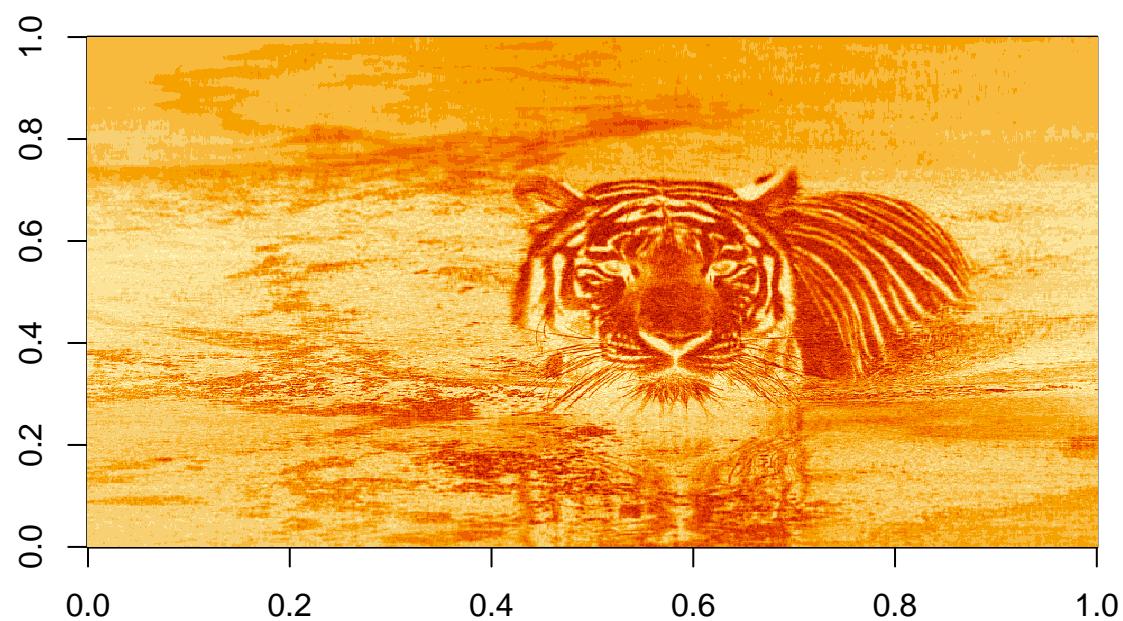


Figure 6: PCA image

```

ica_func <- function(Rimage, components){
  imica <- fastICA(Rimage, n = components)
  compressed_image_matrix <- imica$S %*% imica$A
  return(image(compressed_image_matrix))
}
# ica_func(rasterimg, 100)

```

Image compression using Kmeans

```

kmeans_func <- function(Rimage, center){
  imk <- kmeans(Rimage, centers = center)
  compressed_image_matrix <- imk$centers[imk$cluster,]
  return(image(compressed_image_matrix))
}
# kmeans_func(rasterimg, 100)

```

Comparison Rastering image

Real vs PCA

```

include_graphics(image(im))

## Error in Encoding(x): a character vector argument expected
include_graphics(pca_func(rasterimg, 100))

## Error in Encoding(x): a character vector argument expected

```

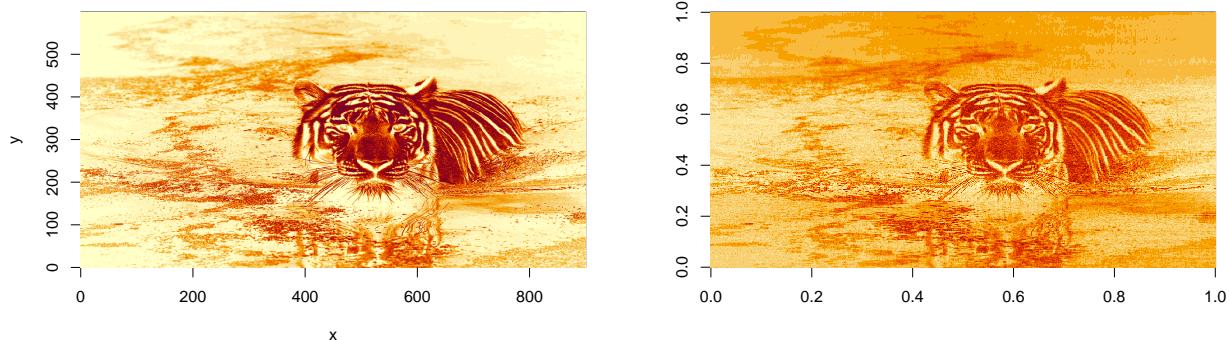


Figure 7: REAL vs PCA

Real vs Factor analysis

```

include_graphics(image(im))

## Error in Encoding(x): a character vector argument expected
include_graphics(fa_func(rasterimg, 100))

## Error in Encoding(x): a character vector argument expected

```

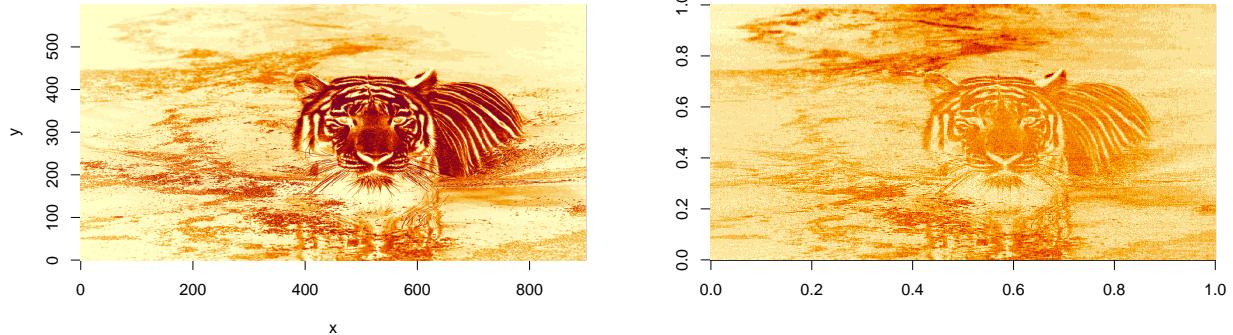


Figure 8: REAL vs Factor Analysis

Real vs SVD

```
include_graphics(image(im))

## Error in Encoding(x): a character vector argument expected
include_graphics(svd_func(rasterimg, 100))

## Error in Encoding(x): a character vector argument expected
```

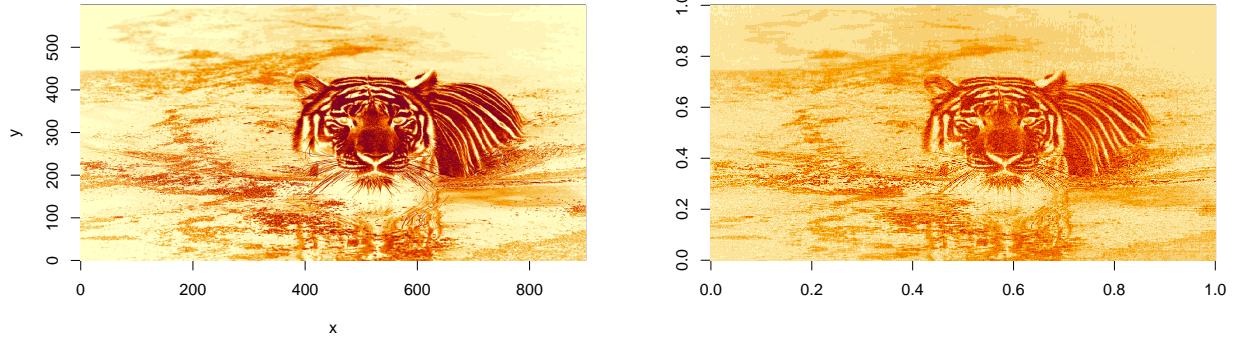


Figure 9: REAL vs SVD

Real vs ICA

```
include_graphics(image(im))

## Error in Encoding(x): a character vector argument expected
include_graphics(ica_func(rasterimg, 100))

## Error in Encoding(x): a character vector argument expected
```

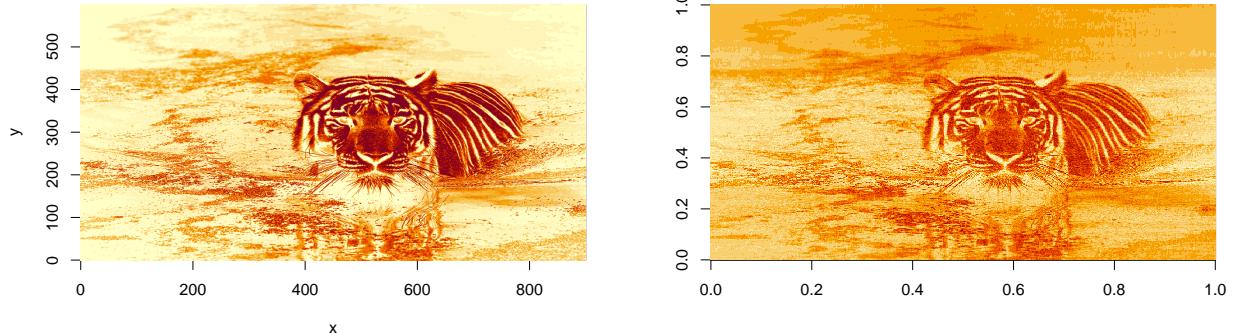


Figure 10: REAL vs ICA

Real vs Kmeans

```
include_graphics(image(im))

## Error in Encoding(x): a character vector argument expected
include_graphics(kmeans_func(rasterimg, 100))

## Error in Encoding(x): a character vector argument expected
```

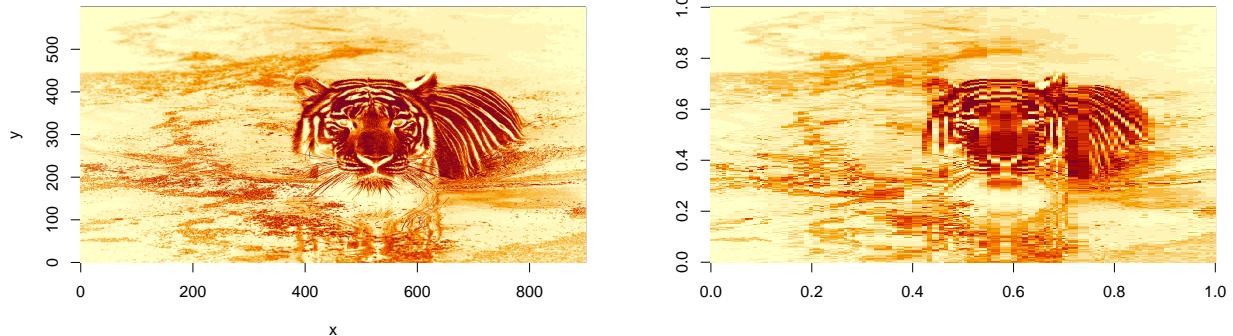


Figure 11: REAL vs KMENAS

Extra

```
library(EBImage)
files <- list.files(path="C:\\\\Users\\\\W.C\\\\Downloads\\\\programming\\\\PCA image", pattern=".jpg", all.files=TRUE)
#This imports the files as: files - chr[1:2] "n04197391_11_0" "n04197391_74_0"
mypic1 <- list()
for (i in 1:length(files))
{
  mypic1[[i]] <- readImage(files[i])
  display(mypic1[[i]])
```

