# Image Compression using PCA

Aishwarya

2023-02-28

**Reading the image**

```
library("jpeg")
img <- readJPEG("tiger.jpeg")
```

**Dividing the components into red, green and blue**

```
red<-img[,,1]
green<-img[,,2]
blue<-img[,,3]
```

**Checking number of rows and columns of original image**

```
nrow(img)
```

```
## [1] 802
```

```
ncol(img)
```

```
## [1] 1200
```

**Applying principal component analysis on each color component**

```
red.pca <- prcomp(red, center = F)
green.pca <- prcomp(green, center = F)
blue.pca <- prcomp(blue, center = F)
```

**Calculating principal component variance, principal component percent variance and principal component cumulative variance percentage for red**

```
red_var<- (red.pca$sd)^2

variance_red_percent<-(red_var/sum(red_var)*100)

cumulative_variance_red<-cumsum(red_var/sum(red_var)*100)
```

**Defining an empty list to save the principal components that explain a cumulative variance of less than 90 percentage**

```
pc.red=list()
```

**Looping through the cumulative variance to get the principal components with cumulative variance less than 90 percent**

```
for (i in cumulative_variance_red)
{
 if (i<=90)
{
  pc.red<-append(pc.red,i)
}
}
```

**Converting the list into a matrix**

```
as.matrix(pc.red)
```

```
##       [,1]
## [1,] 83.22453
## [2,] 86.65945
## [3,] 88.496
## [4,] 89.78184
```

**Calculating principal component variance, principal component percent variance and principal component cumulative variance percentage for green**

```
green_var<-green.pca$sd^2

variance_green_percent<-(green_var/sum(green_var)*100)

cumulative_variance_green<-cumsum(green_var/sum(green_var)*100)
```

**Defining an empty list to save the principal components that explain a cumulative variance of less than 90 percentage**

```
pc.green=list()
```

**Looping through the cumulative variance to get the principal components with cumulative variance less than 90 percent**

```
for (i in cumulative_variance_green)
{
 if (i<=90)
{
   pc.green<-append(pc.green,i)
}
}

as.matrix(pc.green)
```

```
##      [,1]
## [1,] 85.98807
## [2,] 88.02078
## [3,] 89.53239
```

**Calculating principal component variance, principal component percent variance and principal component cumulative variance percentage for blue**

```
blue_var<-(blue.pca$sd)^2

variance_blue_percent<-(blue_var/sum(blue_var)*100)

cumulative_variance_blue<-cumsum(blue_var/sum(blue_var)*100)
```

**Defining an empty list to save the principal components that explain a cumulative variance of less than 90 percentage**

```
pc.blue=list()
```

**Looping through the cumulative variance to get the principal components with cumulative variance less than 90 percent**

```
for (i in cumulative_variance_blue)
{
 if (i<=90)
{
   pc.blue<-append(pc.blue,i)
}
}

as.matrix(pc.blue)
```

```
##      [,1]
## [1,] 79.80665
## [2,] 82.61546
## [3,] 84.69891
## [4,] 85.948
## [5,] 87.16583
## [6,] 88.18191
## [7,] 88.95714
## [8,] 89.56759
```

**Reducing the image size by using the principal components that explain more than 90 percentage of the total variation**

```
R<-red.pca$x[,1:length(pc.red)]%*%t(red.pca$rotation[,1:length(pc.red)])
B<-green.pca$x[,1:length(pc.green)]%*%t(red.pca$rotation[,1:length(pc.green)])
G<-green.pca$x[,1:length(pc.blue)]%*%t(red.pca$rotation[,1:length(pc.blue)])
```

**Combining the individual components from red, green and blue**

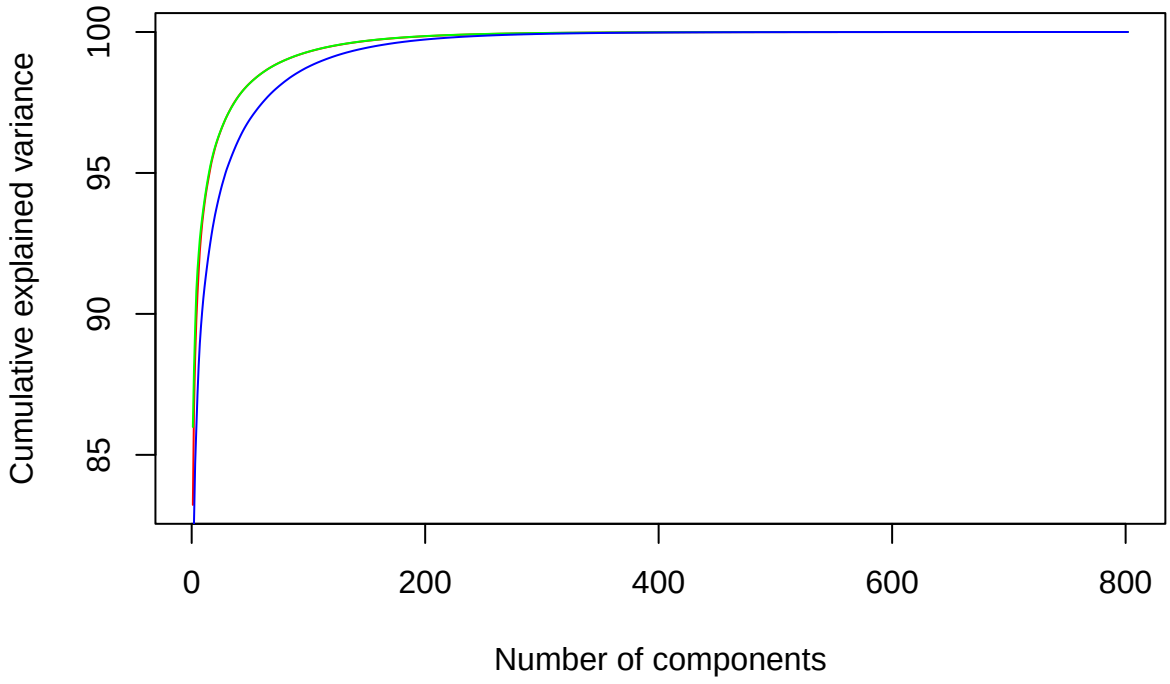
```
new_img=array(c(R,B,G),dim=c(dim(img)[1:2],3))
```

**Writing the compressed image**

```
writeJPEG(new_img,paste('img_compressed_.jpeg',sep=''))
```

**Plot the percentage of variance as k increases using total principal components**

```
plot(cumulative_variance_red,type='l',col='red',xlab="Number of components",
     ylab="Cumulative explained variance")
lines(cumulative_variance_green,col='green')
lines(cumulative_variance_blue,col='blue')
```




The graph shows that for a few initial principal components the explained variance increases. But after a few initial components, the explained variance is stable and graph is flat. This means that the initial principal components explain the most variation in the data. Variance is minimal for components that have higher index.

**Calculating the compression ratio of compressed image to the original image**

```
original_size<-file.info('tiger.jpeg')$size
compressed_size<-file.info('img_compressed_.jpeg')$size
compressed_size/original_size
```

```
## [1] 0.1766477
```

**Creating a vector with all values of cumulative principal components to be taken into consideration**

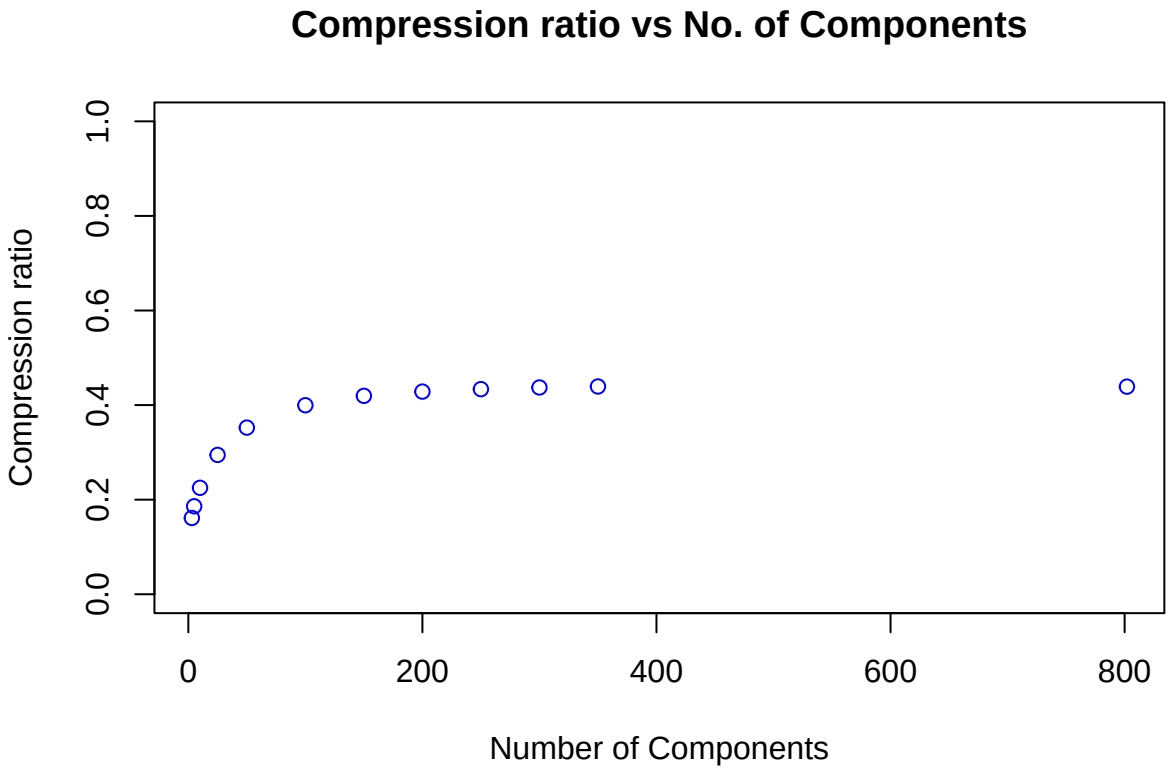
```
k=c(3, 5, 10, 25, 50, 100, 150, 200, 250, 300, 350, nrow(img))
compress_ratio=list()
```

**Looping through the values of k creating image using specified principal components**

```
for(i in k)
{
  R<-red.pca$x[,1:i]%*%t(red.pca$rotation[,1:i])+matrix(red.pca$center,
                                                        nrow(red),
                                                        ncol(red),
                                                        byrow=TRUE)
  B<-blue.pca$x[,1:i]%*%t(blue.pca$rotation[,1:i])+matrix(blue.pca$center,
                                                        nrow(blue),
                                                        ncol(blue),
                                                        byrow=TRUE)
  G<-green.pca$x[,1:i]%*%t(green.pca$rotation[,1:i])+matrix(green.pca$center,
                                                          nrow(green),
                                                          ncol(green),
                                                          byrow=TRUE)

  new_img=array(c(R,B,G),dim=dim(img))

  #Defining name of the newly created images
  full.path <- paste('image_compressed', i, sep='','.jpeg')

  #Writing image to disk
  writeJPEG(new_img,paste(full.path))

  #Calculating compression ratio
  original_size<-file.info('tiger.jpeg')$size
  compressed_size<-file.info(full.path)$size
  rat<-compressed_size/original_size
  compress_ratio<-append(compress_ratio,rat)
}
```

**Plot compression ratio vs number of components used while constructing the image**

```
plot(k,compress_ratio,xlab="Number of Components",ylab="Compression ratio",
     main="Compression ratio vs No. of Components",
     ylim=c(0,1),
     col=c("#0000FF"))
```




The graph shows that the compression ratio is increasing as number of components increase. After a few initial components, the ratio has stabilized and doesn't show an increase in the ratio.