SVD and Image Comparison in R

Eric Hirsch

9/18/2021

We have been given 17 images of shoes, all presumably the same model with subtle differences in color and decoration. Each image is about 1 mb in size and the matrices that describe them will be very large. If we want images that allow us to distinguish the shoes, how can we reduce dimensionality so that we can compare only the most important aspects of each shoe? For this exercise, we use Singular Value Decomposition (SVD).

First we load our images. We will load 6 representative images, which comprise 3 pairs of similar looking shoes.

```
library(imager)
```

```
## Warning: package 'imager' was built under R version 4.0.5
```

```
## Loading required package: magrittr
```

```
##
## Attaching package: 'imager'
```

```
## The following object is masked from 'package:magrittr':
##
##     add
```

```
## The following objects are masked from 'package:stats':
##
##     convolve, spectrum
```

```
## The following object is masked from 'package:graphics':
##
##     frame
```

```
## The following object is masked from 'package:base':
##
##     save.image
```

```
WhiteOrange1 <- load.image("D:\\RStudio\\CUNY_605\\4\\Pictures\\RC_2500x1200_2014_us_53446.jpg")
WhiteOrange2 <- load.image("D:\\RStudio\\CUNY_605\\4\\Pictures\\RC_2500x1200_2014_us_53649.jpg")
BrownStripe1 <- load.image("D:\\RStudio\\CUNY_605\\4\\Pictures\\RC_2500x1200_2014_us_54130.jpg")
BrownStripe2 <- load.image("D:\\RStudio\\CUNY_605\\4\\Pictures\\RC_2500x1200_2014_us_54148.jpg")
PlainBlack1 <- load.image("D:\\RStudio\\CUNY_605\\4\\Pictures\\RC_2500x1200_2014_us_54018.jpg")
PlainBlack2 <- load.image("D:\\RStudio\\CUNY_605\\4\\Pictures\\RC_2500x1200_2014_us_54067.jpg")
```

Here we can see the pairs and examine their differences

```
par(mfrow=c(1, 2))
plot(WhiteOrange1, main="WhiteOrange1", axes=FALSE)
plot(WhiteOrange2, main="WhiteOrange2", axes=FALSE)
```

WhiteOrange1

WhiteOrange2

```
plot(BrownStripe1, main="BrownStripe1", axes=FALSE)
plot(BrownStripe2, main="BrownStripe2", axes=FALSE)
```

BrownStripe1

BrownStripe2

```
plot(PlainBlack1, main="PlainBlack1", axes=FALSE)
plot(PlainBlack2, main="PlainBlack2", axes=FALSE)
```

Now we create a function to perform SVD on the images. SVD will factor the matrix into $U\Sigma V^T$. The columns of U are the left singular vectors, Σ is a diagonal matrix with singular values, and VT has rows that are the right singular vectors. When we perform SVD the singular values are ordered so that they run from largest to smallest. When squared, they are proportional to the amount of variance explained by a the associated singular vectors. In order to perform this exercise we will then reconstruct the original image with only a subset of the vectors, and these singular values will be useful in deciding which vectors to keep,

We first extract the color matrices from the image and then we perform SVD on each separately.

```
Perform_SVD <- function(image)
{

R = image[,,1]
G = image[,,2]
B = image[,,3]

R_svd<-svd(R)
G_svd<-svd(G)
B_svd<-svd(B)

RGB_svd <-list(R_svd,G_svd,B_svd)
return (RGB_svd)
}
```

```
All_svd_WO1 <- Perform_SVD(WhiteOrange1)
All_svd_WO2 <- Perform_SVD(WhiteOrange2)
```

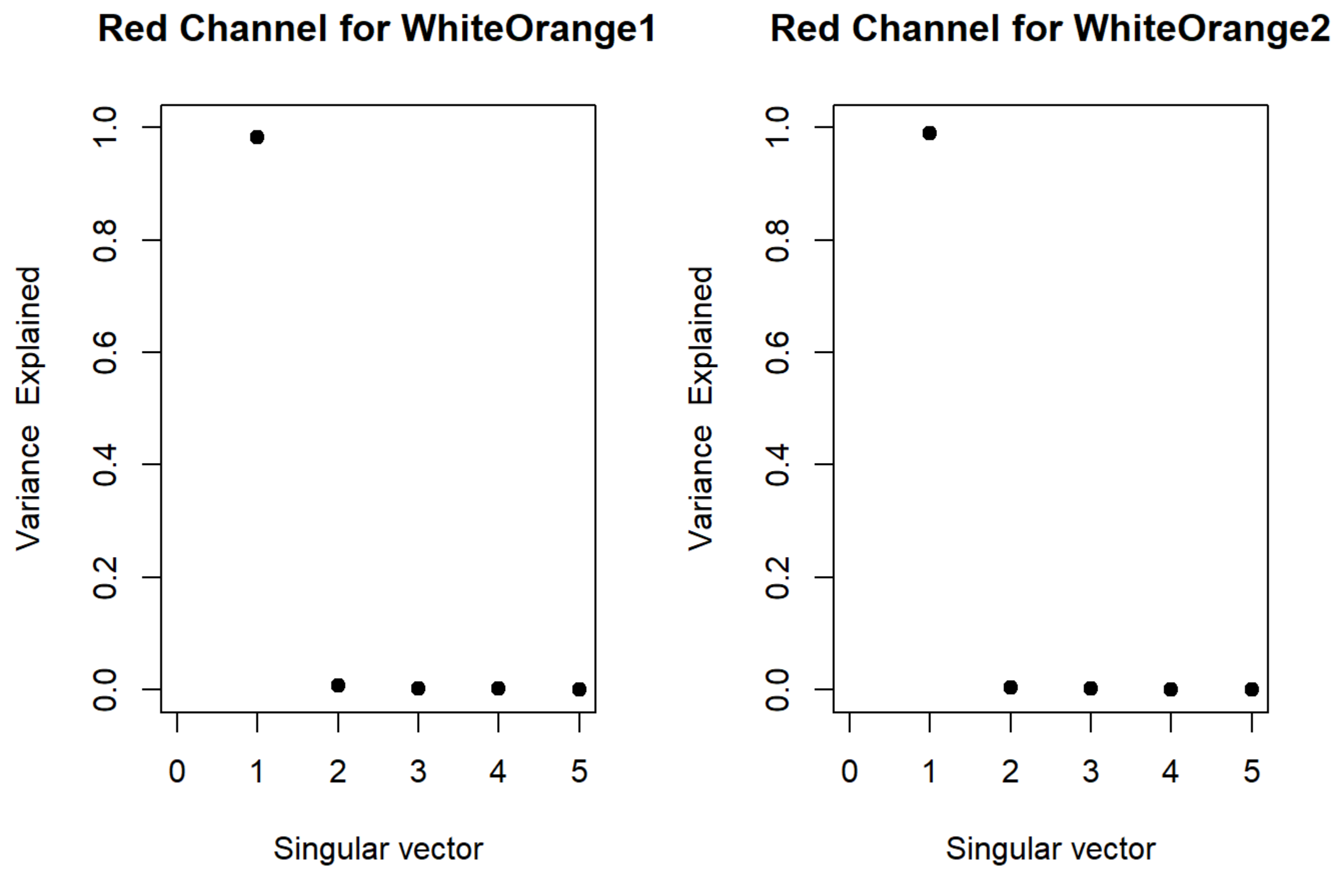
As stated above, SVD will produce singular values that, when squared, are proportional to the amount of variance explained by the associated singular vectors. This comes in handy, because features which have a lot of variance are obviously more critical in identifying an object than those that vary little. We can calculate the percent of variance, therefore, by dividing the squared elements of the diagonal matrix by the sum the of all squared elements of that matrix.

We do that here with the red channels of the first two shoes as an example. We can see that an extremely high degree of variance is already captured by the first singular vector (the two plots are nearly, though not completely, identical). This is not too surprising as the images are in fact quite simple with vast areas of little detail.

```
par(mfrow=c(1, 2))
svd_WO1 <- All_svd_WO1[[1]]
svd_WO2 <- All_svd_WO2[[1]]

plot(svd_WO1$d^2/sum(svd_WO1$d^2), pch=19, ylim=c(0,1), xlim=c(0,5), xlab="Singular vector",ylab="Variance  Explained", main
="Red Channel for WhiteOrange1")

plot(svd_WO2$d^2/sum(svd_WO2$d^2), pch=19, ylim=c(0,1), xlim=c(0,5), xlab="Singular vector",ylab="Variance  Explained", main
="Red Channel for WhiteOrange2")
```



Now comes the powerful part of SVD. As evident above, we can select a small subset of singular vectors and still capture the majority of the variance. This gives us an image that takes up a fraction of the space but is nonetheless good enough to distinguish images.

While the analysis suggests that even one vector captures the lion's share of variance, I'm going to use 5 and 10 to capture any subtleties missed with 1 or 2.

We do this with a function:

```
ReconstructImage <- function(svdx, NumOfVectors)
{

comp <- sapply(svdx, function(i){
        compressed = i$u[,1:NumOfVectors] %*% diag(i$d[1:NumOfVectors]) %*% t(i$v[,1:NumOfVectors])
}, simplify = 'array')

comp<-as.cimg(comp)

return (comp)
}
```

We do the analysis with the White Orange shoes. Even with only 5 vectors we can figure out which shoe is which.

```
par(mfrow=c(2, 3))

NewImage <- ReconstructImage(All_svd_WO1, 5)
```

```
## Warning in as.cimg.array(comp): Assuming third dimension corresponds to colour
```

```
plot(NewImage, axes=FALSE,main="Number Of vectors=5")

NewImage <- ReconstructImage(All_svd_WO1, 10)
```

```
## Warning in as.cimg.array(comp): Assuming third dimension corresponds to colour
```

```
plot(NewImage,axes=FALSE, main="Number Of vectors=10")

plot(WhiteOrange1, main="Original", axes=FALSE)

NewImage <- ReconstructImage(All_svd_WO2, 5)
```

```
## Warning in as.cimg.array(comp): Assuming third dimension corresponds to colour
```

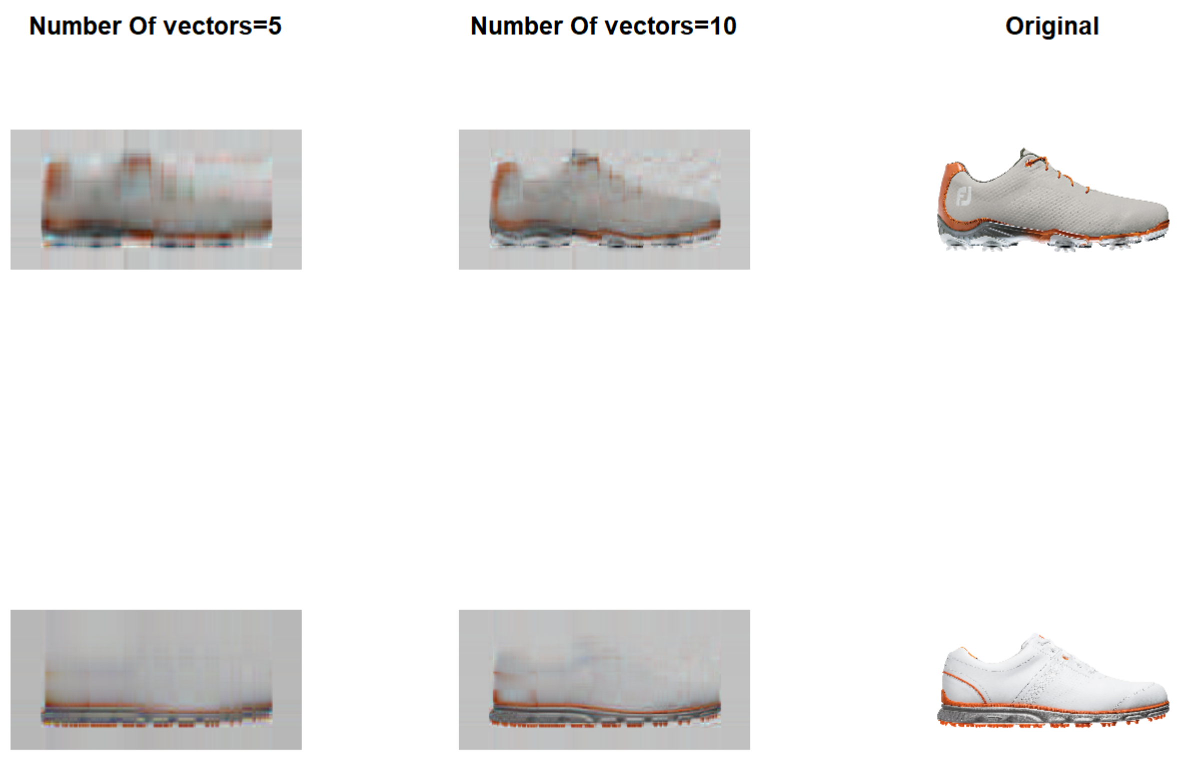
```
plot(NewImage, axes=FALSE)

NewImage <- ReconstructImage(All_svd_WO2, 10)
```

```
## Warning in as.cimg.array(comp): Assuming third dimension corresponds to colour
```

```
plot(NewImage,axes=FALSE)

plot(WhiteOrange2, axes=FALSE)
```



The Brown Stripe shoes show a similar pattern. Even with only 5 vectors we can figure out which shoe is which.

```
par(mfrow=c(2, 3))

All_svd_BS1 <- Perform_SVD(BrownStripe1)
All_svd_BS2 <- Perform_SVD(BrownStripe2)

NewImage <- ReconstructImage(All_svd_BS1, 5)
```

```
## Warning in as.cimg.array(comp): Assuming third dimension corresponds to colour
```

```
plot(NewImage, axes=FALSE,main="Number Of vectors=5")

NewImage <- ReconstructImage(All_svd_BS1, 10)
```

```
## Warning in as.cimg.array(comp): Assuming third dimension corresponds to colour
```

```
plot(NewImage,axes=FALSE, main="Number Of vectors=10")

plot(BrownStripe1, main="Original", axes=FALSE)

NewImage <- ReconstructImage(All_svd_BS2, 5)
```

```
## Warning in as.cimg.array(comp): Assuming third dimension corresponds to colour
```

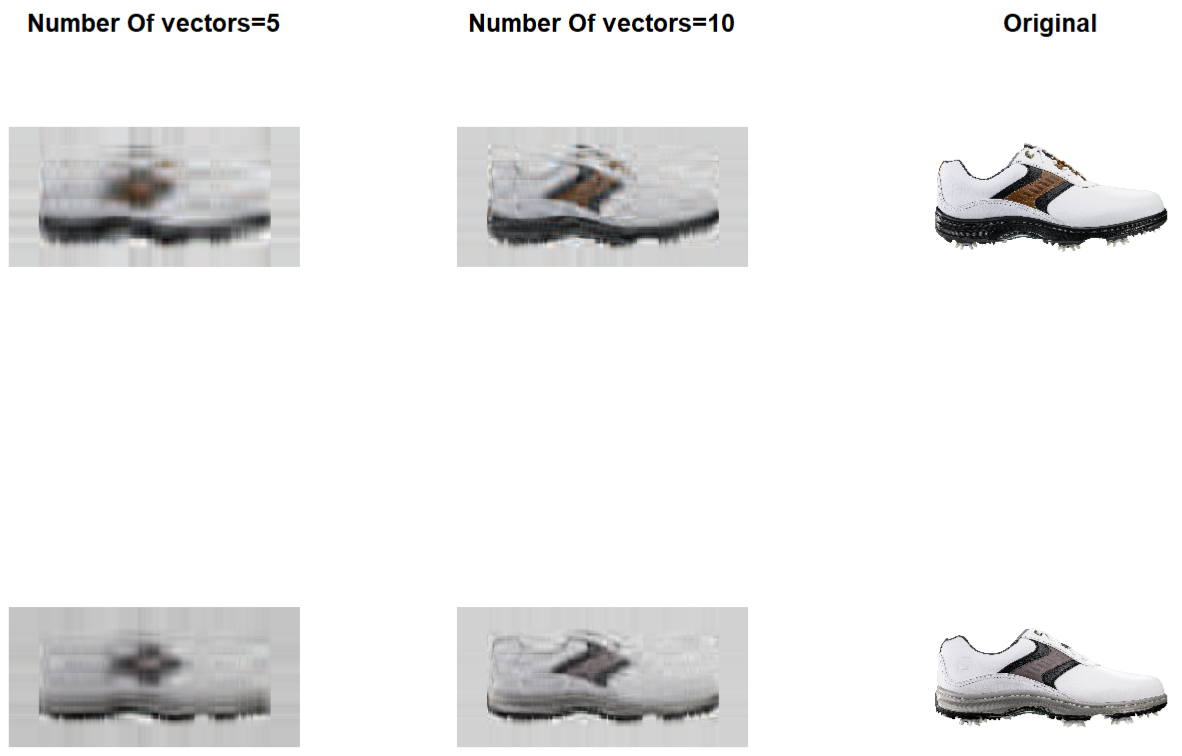
```
plot(NewImage, axes=FALSE)

NewImage <- ReconstructImage(All_svd_BS2, 10)
```

```
## Warning in as.cimg.array(comp): Assuming third dimension corresponds to colour
```

```
plot(NewImage,axes=FALSE)

plot(BrownStripe2, axes=FALSE)
```



The Plain Black shoes show a similar pattern. They are hard to distinguish, even without SVD, as they mainly differ in a subtle shade of color and a small knob on the back of one shoe. However, these differences are apparent in the reduced images.

```
par(mfrow=c(2, 3))

All_svd_PB1 <- Perform_SVD(PlainBlack1)
All_svd_PB2 <- Perform_SVD(PlainBlack2)

NewImage <- ReconstructImage(All_svd_PB1, 5)
```

```
## Warning in as.cimg.array(comp): Assuming third dimension corresponds to colour
```

```
plot(NewImage, axes=FALSE,main="Number Of vectors=5")

NewImage <- ReconstructImage(All_svd_PB1, 10)
```

```
## Warning in as.cimg.array(comp): Assuming third dimension corresponds to colour
```

```
plot(NewImage,axes=FALSE, main="Number Of vectors=10")

plot(PlainBlack1, main="Original", axes=FALSE)

NewImage <- ReconstructImage(All_svd_PB2, 5)
```

```
## Warning in as.cimg.array(comp): Assuming third dimension corresponds to colour
```

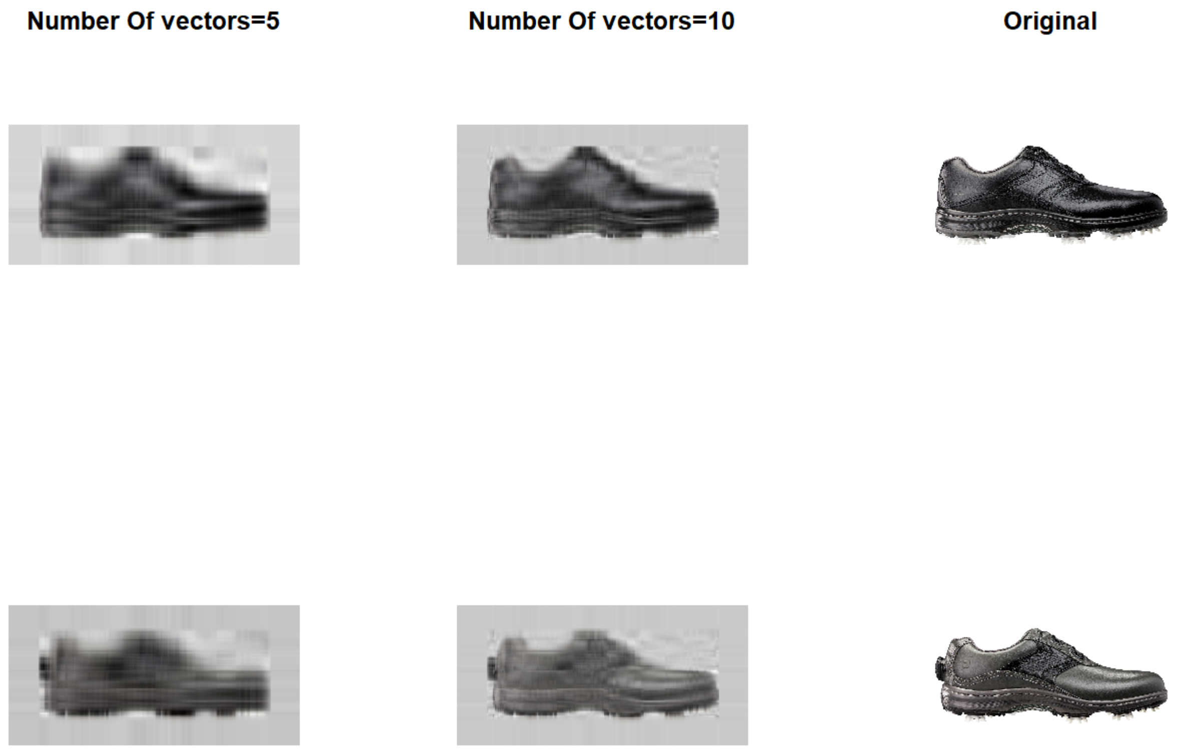
```
plot(NewImage, axes=FALSE)

NewImage <- ReconstructImage(All_svd_PB2, 10)
```

```
## Warning in as.cimg.array(comp): Assuming third dimension corresponds to colour
```

```
plot(NewImage,axes=FALSE)

plot(PlainBlack2, axes=FALSE)
```



This exercise shows us how SVD is useful in dimensionality reduction