Project Proposal - Principal Components

(1) a list of members;

Shaurik Sudheer Deshpande, Johanna Niggemann, Mary Hunter Russell, Cameron Santos, and Henry Shugart

(2) precise description including location where I can find (unless this is copyright protected) the dataset;

The dataset we chose for our project is a collection of images of birds for which we plan to build a model to classify by species (and can be found here (https://www.kaggle.com/gpiosenka/100-bird-species?select=my_csv-2-17-2022-1-17-48.csv)). The data includes 58,388 observations across 400 species and was already partitioned into training and test data for the model. Additional files with basic information about the bird species and scientific name are included; however, those are not relevant to the classification process. Link: https://www.kaggle.com/gpiosenka/100-bird-species?select=my_csv-2-17-2022-1-17-48.csv (https://www.kaggle.com/gpiosenka/100-bird-species?select=my_csv-2-17-2022-1-17-48.csv)

(3) your motivations and goals;

While none of our group members are aviary aficionados, we chose this dataset because it allows us to dive into image classification, a fundamental component of machine learning. The concept of “computer vision” - developing programs that allow a computer to understand what is being presented to it in the form of pixels - is increasing in popularity across tech fields. Image classification is key to developing computer vision. It has broad applications today ranging from disease detection in medical imaging to programming autonomous vehicles. This dataset is robust enough to apply and experiment with several techniques, and we are excited to learn more about birds along the way.

Our main goal is classifying the images by species. In the process, we would like to work with neural networks and deep learning techniques. As cost reduction in ML applications is such a big topic in analytics, we would also like to explore how much of the relatively large dataset is needed to develop a model with sufficient accuracy.

(4) Exploratory Data Analysis

```
setwd("C:/Users/Henry Shugart/Desktop/STOR 565/archive/")
birds <- read_csv("birds.csv",show_col_types = FALSE)
birds$data_set = birds[,4]
n_birds = birds %>% filter(data_set == "train") %>% group_by(labels) %>% summarize(n = n()) %
>% arrange(desc(n))
birds_train = birds %>% filter(data_set == "train")
```

Here we get basic information about the dataset. We see the dimensions of the images are all $224 * 224$ and there are 58388 total images. We also see the distribution of the number of images per species in the data set. There are an average of 146 images per species which should be plenty to train the model.

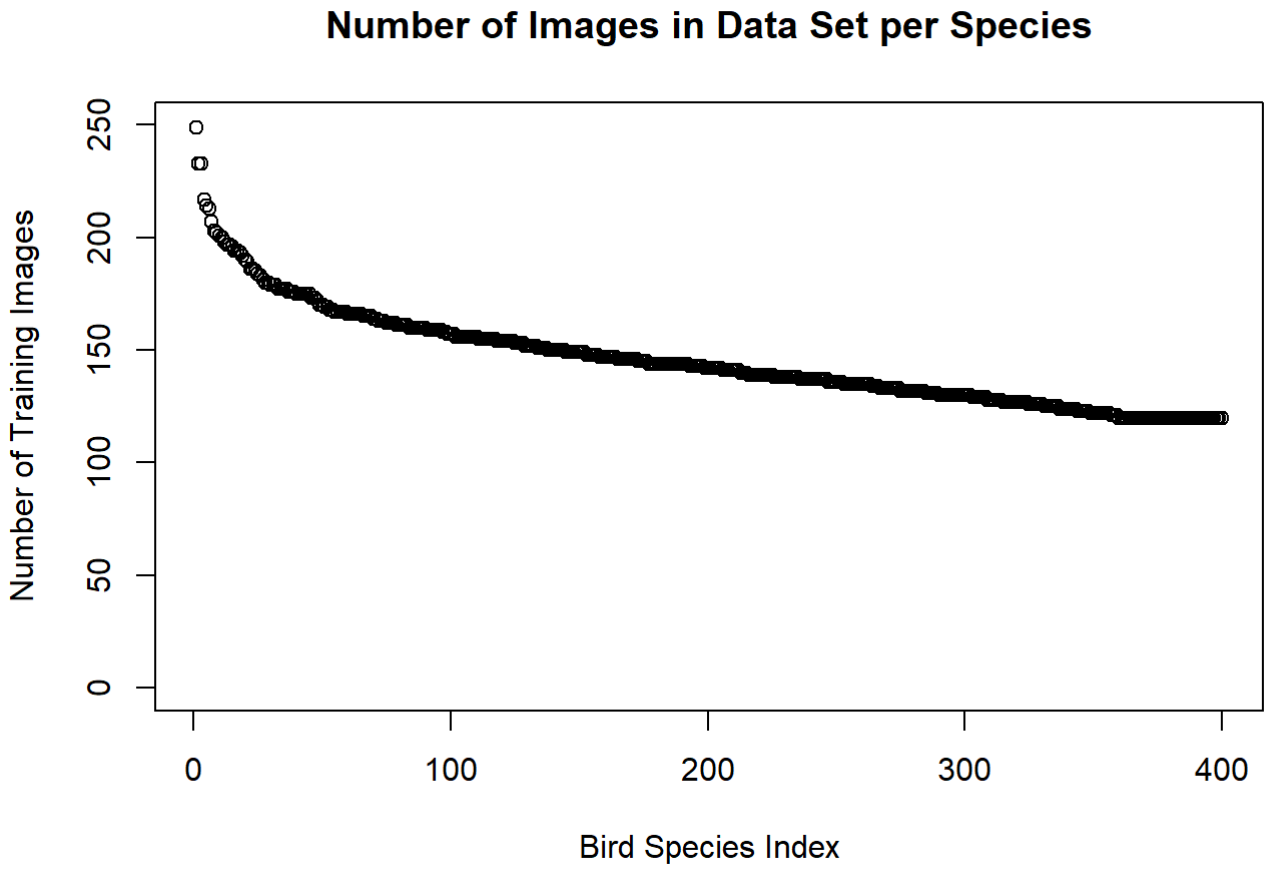
```
nrow(birds_train)
```

```
## [1] 58388
```

```
dim(as.raster(readImage(birds_train$filepaths[1])))
```

```
## [1] 224 224
```

```
plot( n_birds$n, ylim = c(0,250), xlab = "Bird Species Index", ylab = "Number of Training Ima
ges", main = "Number of Images in Data Set per Species")
```

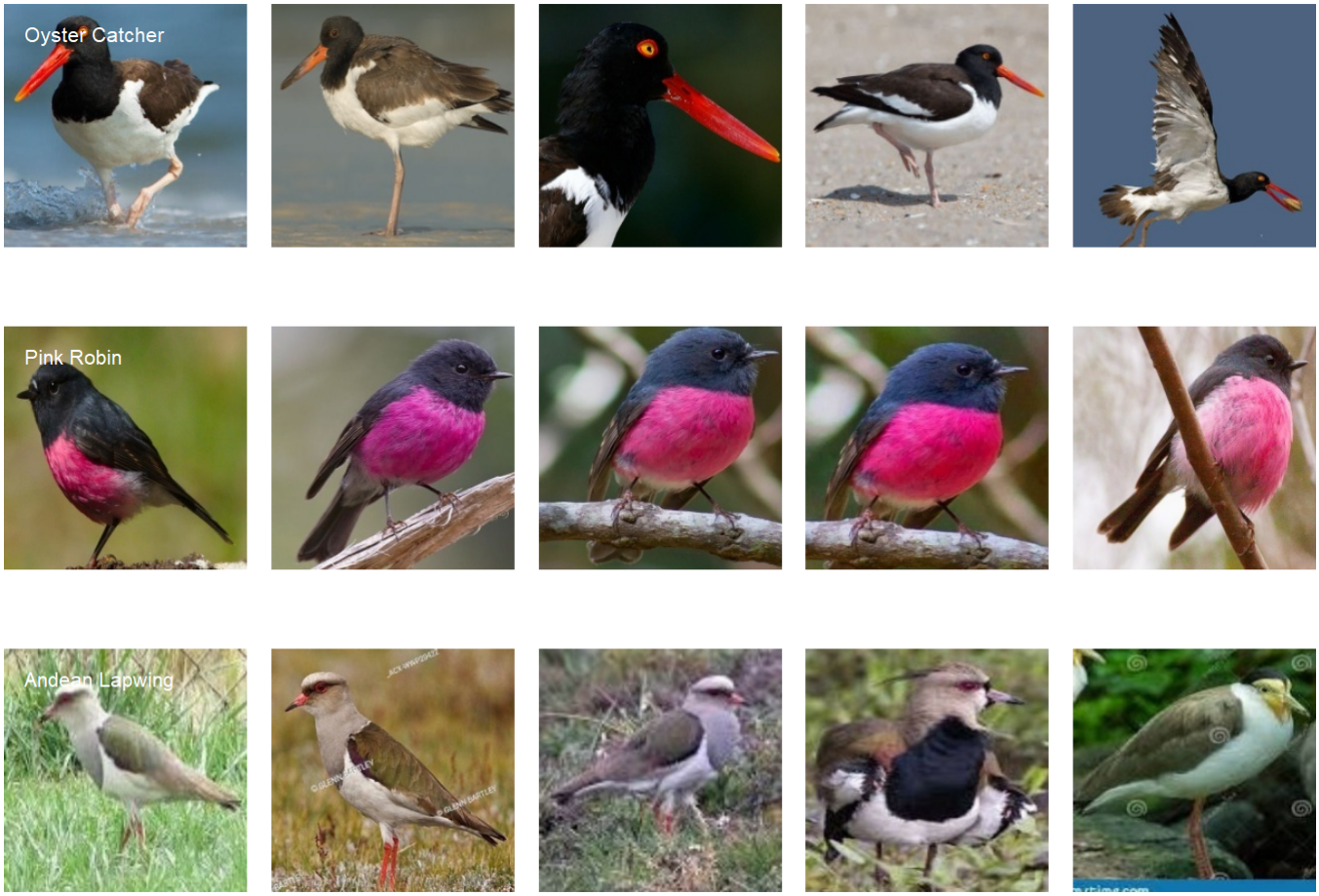



```
mean(n_birds$n)
```

```
## [1] 145.97
```

We now look at the compressibility of the images. We do this looking at 2 metrics, both the average percent variance explained by n principal components as well as the minimum variance explained. This allows us to see if we can potentially reduce the size of each image prior to training to make our model training more efficient. We can see that most images are very compressible and even where the data is not easily projected into lower dimensions, there is some ability to dimension reduce.
Here we sample 300 random images from the training data to do PCA on as it would be extremely slow to do PCA on all 58,000 images.

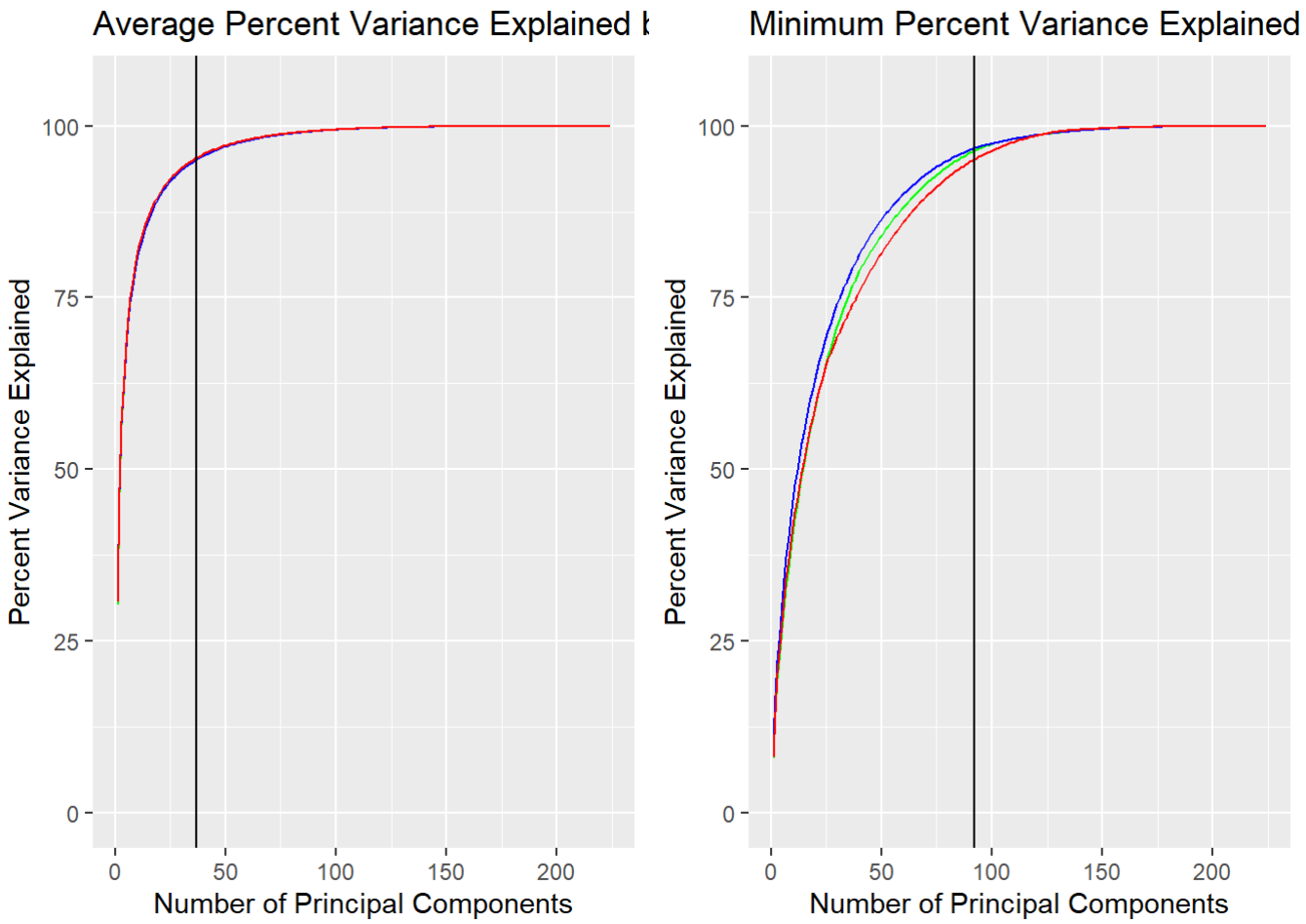
```
set.seed(99)
nx <- 5
ny <- 3
specieslist = sample(unique(birds_train$labels), ny)
images_vec = c()
for(species in specieslist){

species_only = birds_train %>% filter(labels == species)
images_vec = c(images_vec, sample(species_only$filepaths, nx))
}

img <- list()
j = 0
for(name in images_vec){
  j = j+1
img[[j]] <- readImage(name)}



cols <- 2*nx-1
rows <- 2*ny-1
m <- matrix(0, cols, rows)
m[2*(1:nx)-1, 2*(1:ny)-1] <- 1:(nx*ny)
m <- t(m)


pad <- .1

w <- rep(1, cols)
w[!(1:cols)%%2] <- pad
h <- rep(1, rows)
h[!(1:rows)%%2] <- pad * dim(img[[1]])[1L]/dim(img[[1]])[2L]

layout(m, widths = w, heights = h)


j = 0
k = 0
for (i in 1:(nx*ny)) {
  j = j+1
  display(img[[j]], method="raster")
  if(j %% nx == 1){
    k= k+1
  text(x = 20, y = 20, label = str_to_title(specieslist[k]), adj = c(0,1), col = "white", cex
= 1, )}
}
```



Here we look at a couple of the images in the data set and see that there are a variety of poses which each bird takes. This should help improve the model as a picture from any angle should be classified well.

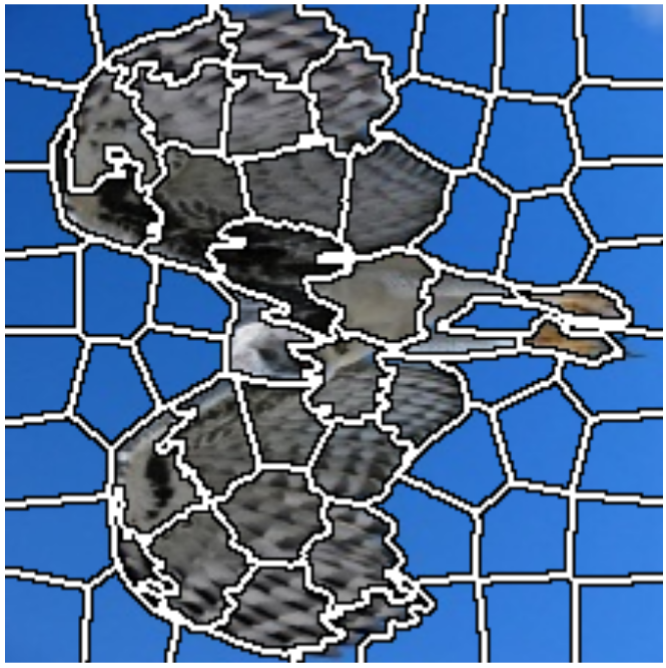
```
set.seed(99)
min_vec = function(vec_1, vec_2){
  mins = vec_1
  for(i in 1:length(vec_1)){
    mins[i] = min(vec_1[i], vec_2[i])}
  return(mins)
  }

cum.pve.red = cum.pve.green = cum.pve.blue = 0
minvar.red = minvar.green = minvar.blue = rep(1, 224)



n = 300

for(i in sample(1:nrow(birds_train), n)){
j = i
noise = rnorm(224^2, 0, .01)
bird = readJPEG(birds_train$filepaths[i])
bird_red_only <- bird[, , 1]+noise
pr.red <- prcomp(bird_red_only, scale = T)
pr.var.red <- pr.red$sdev^2
pve.red <- pr.var.red / sum(pr.var.red)
cum.pve.red = pve.red+ cum.pve.red
minvar.red = min_vec(cumsum(pve.red),minvar.red)


bird_blue_only <- bird[, , 3]+noise
pr.blue <- prcomp(bird_blue_only, scale = T)
pr.var.blue <- pr.blue$sdev^2
pve.blue <- pr.var.blue / sum(pr.var.blue)
cum.pve.blue = pve.blue+ cum.pve.blue
minvar.blue = min_vec(cumsum(pve.blue),minvar.blue)

bird_green_only <- bird[, , 2]+noise
pr.green <- prcomp(bird_green_only, scale = T)
pr.var.green <- pr.green$sdev^2
pve.green <- pr.var.green / sum(pr.var.green)
cum.pve.green = pve.green+ cum.pve.green
minvar.green = min_vec(cumsum(pve.green),minvar.green)
}




avg.plot = ggplot()+geom_line(aes(x = 1:224, y = cumsum(cum.pve.green)/n*100), color = "gree
n")+geom_line(aes(x = 1:224, y = cumsum(cum.pve.blue)/n*100), color = "blue")+geom_line(aes(x
= 1:224, y = cumsum(cum.pve.red)/n*100), color = "red")+ geom_vline(xintercept = max(min(whic
h(cumsum(cum.pve.green)/n > .95)),min(which(cumsum(cum.pve.red)/n > .95)),min(which(cumsum(cu
m.pve.blue)/n > .95))), color = "black")+xlab("Number of Principal Components")+ylab("Percent
Variance Explained")+ylim(0,105) + ggtitle("Average Percent Variance Explained by Principle C
omponents")
```

```
min.plot = ggplot()+geom_line(aes(x = 1:224, y = minvar.green*100), color = "green")+geom_lin
e(aes(x = 1:224, y = minvar.blue*100), color = "blue")+geom_line(aes(x = 1:224, y = minvar.re
d*100), color = "red")+xlab("Number of Principal Components")+ylab("Percent Variance Explaine
d")+ylim(0,105) + ggtitle("Minimum Percent Variance Explained by Principle Components") +  ge
om_vline(xintercept = max(min(which(minvar.green > .95)),min(which(minvar.red > .95)),min(whi
ch(minvar.blue> .95))), color = "black")

grid.arrange(avg.plot, min.plot, ncol=2)
```



Here we look at a couple of images in the data set and look at the ability of it to be segmented using the superpixels algorithm. We can see some data is very well segmented while other the algorithm struggles to identify the boundaries of the birds in others. This is something which we will consider while using image segmentation for analysis going forward.

```
images = list(length = n)
i = 0
imagepaths = c("train/HARPY EAGLE/128.jpg", "train/MYNA/045.jpg")
n = length(imagepaths)
for(name in imagepaths){
i = i+1
False.Color <- readImage(name)
Region.slic = suppressWarnings(superpixels(input_image = False.Color, method = "slic", superp
ixel = 80, compactness = 30, return_slic_data = TRUE, return_labels = TRUE, write_slic = "",
 verbose = FALSE))

images[[i]] = Region.slic$slic_data
}


for(i in 1:n){
#Code inspired by https://cran.r-project.org/web/packages/OpenImageR/vignettes/Image_segmenta
tion_superpixels_clustering.html
par(mfrow=c(1,2), mar = c(0.2, 0.2, 0.2, 0.2))
graphics::plot(as.raster(readImage(imagepaths[i])))
plot_slic = NormalizeObject(images[[i]])
plot_slic = grDevices::as.raster(plot_slic)
graphics::plot(plot_slic)
}
```



(5) preliminary ideas on what kinds of techniques you plan to apply to achieve your goals

In order to achieve our goal, we hope to use a number of different techniques to create a model that can accurately and efficiently classify bird species. We hope that certain methods utilized in this model will also logically extend to other image-classification problems.

As we approach this problem we will utilize a number of different techniques to preprocess our images prior to training and testing our model. Some methods we are interested in are compression and PCA for the purposes of dimensionality reduction. We hope that this will allow us to create a model that is not only accurate, but also efficient. We are also interested in exploring the trade-offs between compression and accuracy in our model.

In addition to compression, PCA, and similar methods, we hope to explore and utilize image segmentation algorithms as part of our preprocessing for our model. We are interested in whether providing a model with a pre-segmented image will result in a better prediction accuracy than a non pre-segmented image

Finally, as indicated earlier, we intend to use these images to create a model that can accurately predict each bird species. In order to do this we intend to utilize a number of image classification algorithms such as SVMs, random forests, and NNs.

Ultimately, through this process we hope to discover a model that can predict new images of birds with a high accuracy, and which can be extrapolated well to other images (especially to other animals).