Code to analyze tree structure from an image

Kwanghun Choi

Install and load essential packages for analysis

```r
library(pacman) # package manager
p_load("imager", "tidyverse", "foreach", "segmented", "rmarkdown", "pander")
```

Assign folders for analysis

```r
data_path <- c("/home/kwanghun/Dropbox/Project/TreeInventory/Data/SemanticSegmentation/")
data_name <- c("Sample_data_3")
```

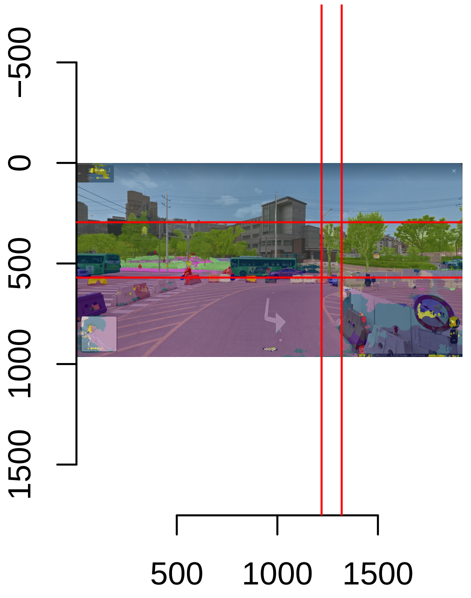
Load data

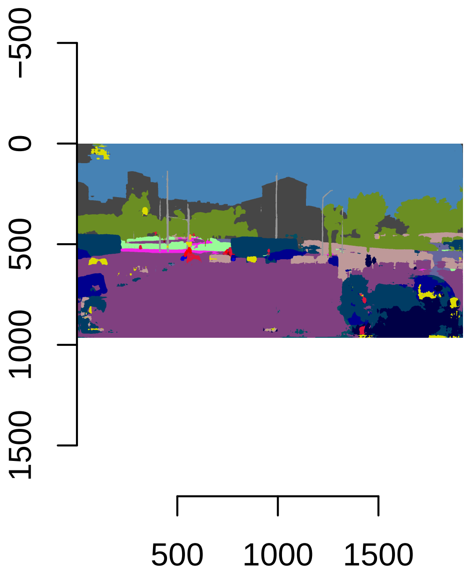
```r
# Image With Background pictures
iwb <- load.image(paste0(data_path, data_name, "/merged/semantic/semantic_pred_0.png"))
# Image withOut (Ohne) Background pictures
iob <- load.image(paste0(data_path, data_name, "/Semantic/semantic/semantic_pred_0.png"))
```

Plot and check cropping regions

```r
# Show two pictures in a row
par(mfrow=c(1,2))
# 1. Image With Background
plot(iwb)
# Create boundary for a tree manually
abline(v=1220, col="red")
abline(v=1320, col="red")
abline(295,0, col="red")
abline(570,0, col="red")

# 2. Image withOut Background
plot(iob)
```

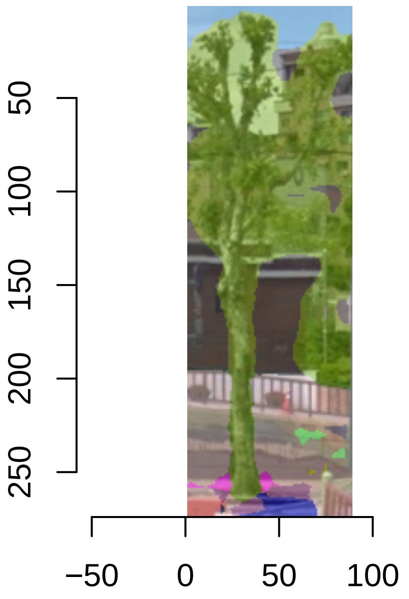




Split images

```
# Split image with imsub
IW <- iwb %>% imsub(x>1230&x<1320) %>% imsub(y>295&y<570)
IO <- iob %>% imsub(x>1230&x<1320) %>% imsub(y>295&y<570)

# Show two pictures in a row
par(mfrow=c(1,2))
# Plot results
plot(IW)
plot(IO)
```



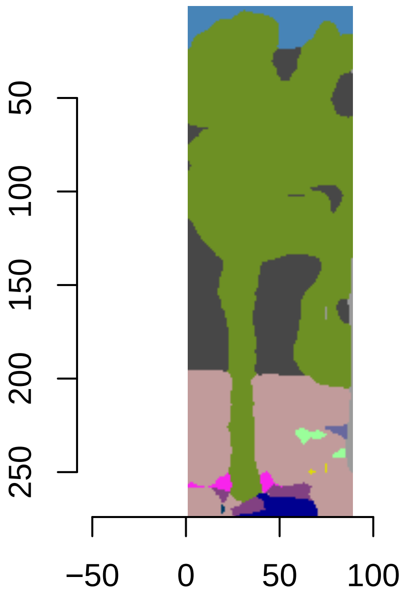


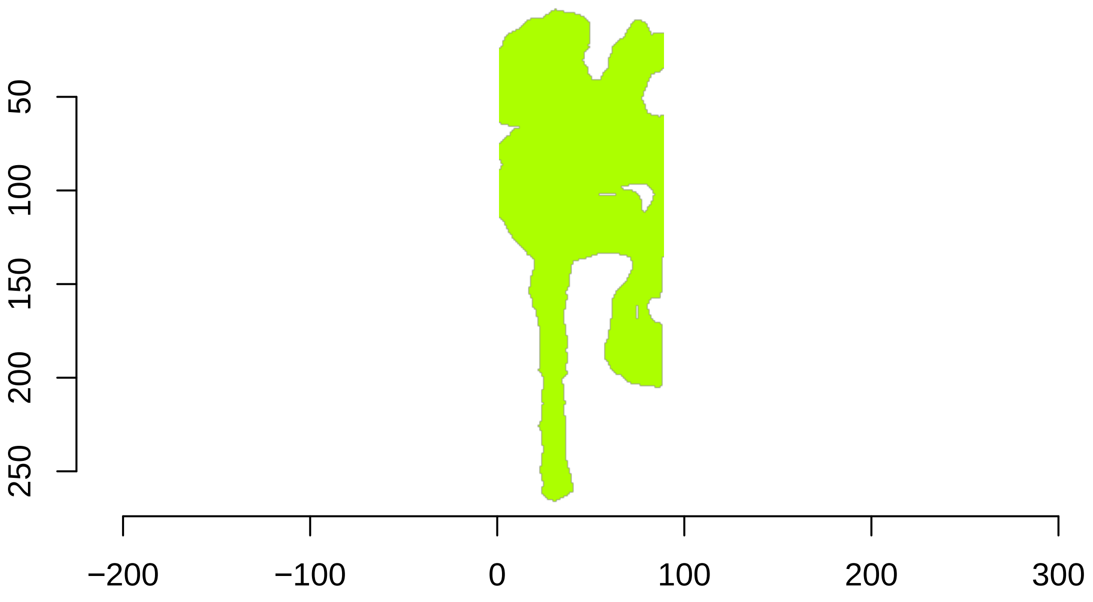
Image analysis

1. Extract colors of vegetation (107, 142, 35) from panoptic-deeplab hompasge (https://github.com/bowenc0221/panoptic-deeplab/issues/42)

0 : "road" 1 : "sidewalk" 2 : "building" 3 : "wall" 4 : "fence" 5 : "pole" (153, 153, 153) 6 : "traffic light" 7 : "traffic sign" 8 : "vegetation" (107, 142, 35) 9 : "terrain" 10 : "sky" 11 : "person" 12 : "rider" 13 : "car" 14 : "truck" 15 : "bus" (0, 60, 100) 16 : "train" 17 : "motorcycle" 18 : "bicycle"

```r
# Remove all other colors except those of vegetation pixels from the semented image
R(IO)[which(!R(IO)[] * 255 == 107)] <- NA
G(IO)[which(!G(IO)[] * 255 == 142)] <- NA
B(IO)[which(!B(IO)[] * 255 == 35)] <- NA

# Check the result map
plot(IO)
```

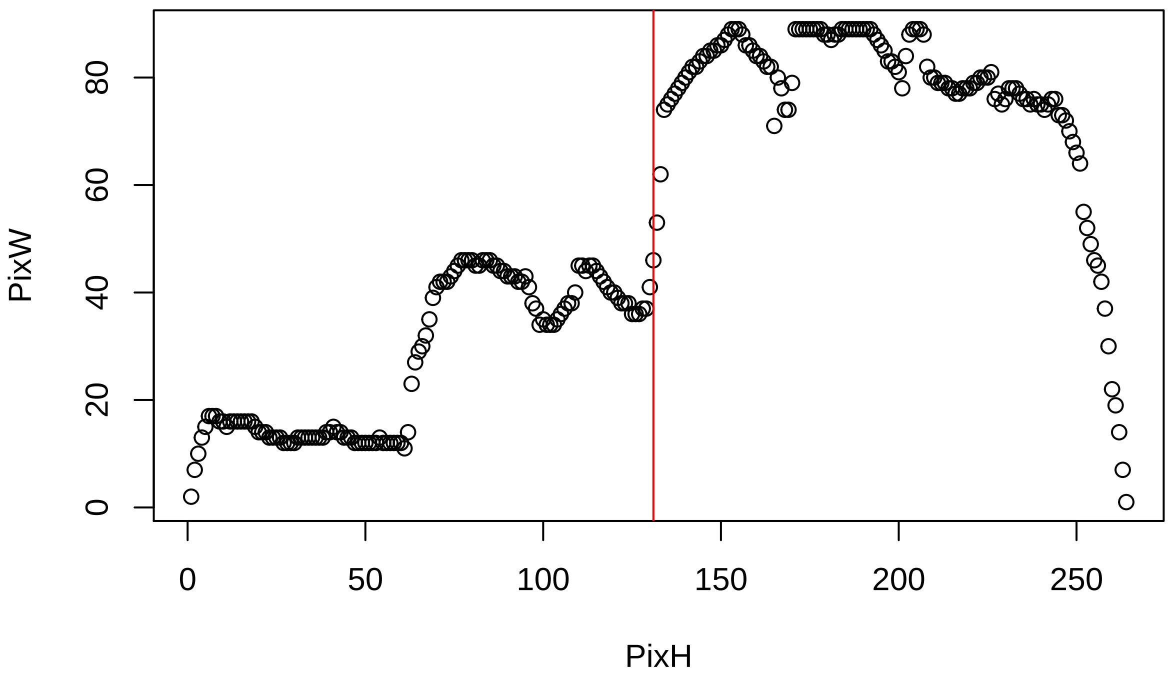


2. count values from bottom to top

```r
# In this part, I calculated the vegetation pixel by each row
out <- foreach(i = seq(ncol(IO),1, -1), .combine=c) %do% {
    length(which(!is.na(R(IO)[,i]+G(IO)[,i]+B(IO)[,i])))
}

# Remove the row with zero vegetation pixels
out.nonzero <- out[out[]>0]
out.df <- data.frame(PixW = out.nonzero) %>%
            mutate(PixH = c(1:nrow(.)))
plot(PixW ~ PixH, data=out.df)

# Check the first change point of the plot
out.cp <- strucchange::Fstats(PixW ~ 1, data=out.df)
abline(v=out.cp$breakpoint, col="red")
```



3. Create Tree inventory table

```
# Create Tree inventory in pixels
Tree_W   <- max( out.df$PixW, na.rm=T)             # Crown width
Tree_H   <- nrow(out.df)                           # Tree Height
Tree_BH  <- out.cp$breakpoint                      # Height below crown.
Tree_DBH <- quantile(out.df$PixW[1:Tree_BH], 0.1) # DBH which is assumed to be a median width below cro

# Create Table for the metric.
M <- data.frame(Tree_H, Tree_W, Tree_BH, Tree_DBH)

# Convert Pixels to real units
#FocLmm <- 3.94                               # Focal length of Pocophone F1
#PixSizeCCD <- 1.4 * 10^-6                       # Pixel width of the camera ccd of Pocophone F1
#Dist <- 100                                   # Distance from the camera to the object

# Calculate width of a pixel in the image
#PixSizeIMG <- Dist * PixSizeCCD / FocLmm
PixSizeIMG <- 0.025 # Assumption of PixSizeIMG of GSV
# Final table
pander(M *  PixSizeIMG)
```

	Tree_H	Tree_W	Tree_BH	Tree_DBH
10%	6.6	2.225	3.275	0.3