remote sensing

MDPI

Supplementary material

Post-Hurricane Damage Severity Classification at the Individual Tree Level Using Terrestrial Laser Scanning and Deep Learning

Carine Klauberg[1], Jason Vogel[1], Ricardo Dalagnol[2,3], Matheus Pinheiro Ferreira[4], Caio Hamamura[5], Eben Broadbent[1], and Carlos Alberto Silva[1]

[1] School of Forest, Fisheries, and Geomatics Sciences, University of Florida Gainesville, FL 32611, USA; jvogel@ufl.edu (J.V.); eben@ufl.edu (E.B.); c.silva@ufl.edu (C.A.S.)
[2] Center for Tropical Research, Institute of the Environment and Sustainability, University of California Los Angeles (UCLA), Los Angeles, CA 90095, USA; dalagnol@ucla.edu
[3] NASA-Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA
[4] Cartographic Engineering Department, Military Institute of Engineering (IME), Praça Gen, Tibúrcio 80, Rio de Janeiro 22290-270, RJ, Brazil; matheus@ime.eb.br
[5] Federal Institute of Education, Science and Technology of São Paulo, Avenida Doutor Ênio Pires de Camargo, Capivari 13365-010, SP, Brazil; hamamura.caio@ifsp.edu.br
* Correspondence: carine.klaubergs@ufl.edu; Tel.: +1-(352)-294-6885

rTLSDeep Package

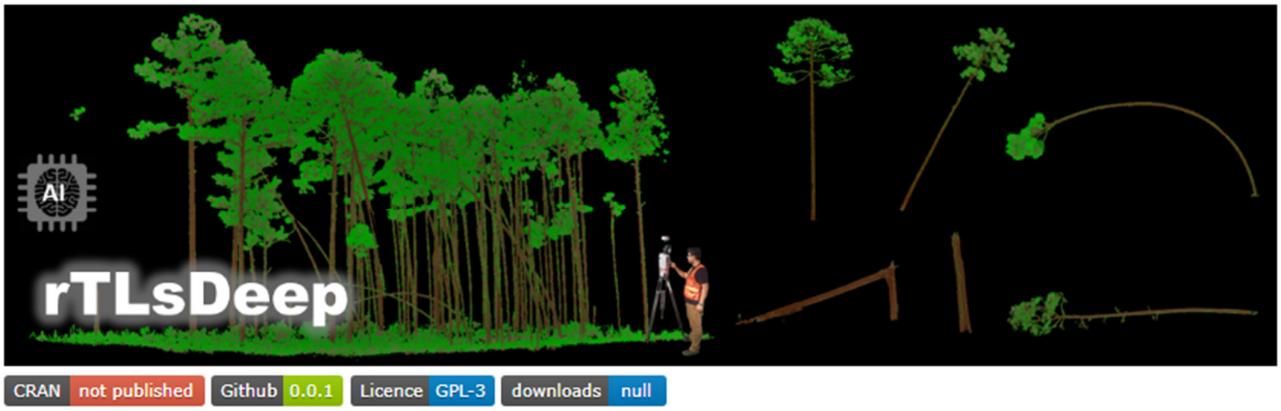

Figure S1. Cover page of the rTLSDeep package

rTLsDeep: An R Package for post-hurricane damage severity classification at the individual tree level using terrestrial laser scanning and deep learning.

Authors: Carine Klauberg, Carlos Alberto Silva, Ricardo Dalagnol, Matheus Ferreira, Danilo Romeu Farias de Souza, Luiz Guilherme Almeida Nogueira, Eben Broadbent, Caio Hamamura and Jason Vogel.

The rTLSDeep package provides options for i) rotating and deriving 2D images from TLS 3D point clouds, ii) calibrating and validating convolutional neural network (CNN) architectures and iii) predicting post-hurricane damage severity at the individual tree level

Getting Started

Install R, Git and Rtools40

i) *R (>= 4.0.0)*: https://www.r-project.org/

ii) *Git*: https://git-scm.com/

iii) *tensorflow (python environment)*: https://doi.org/10.5281/zenodo.3929709

rTLsDeep installation

```
# The CRAN version:
install.packages("rTLsDeep")

# The development version:
#install.packages("remotes")
library(remotes)
    install_github("https://github.com/carlos-alberto-silva/rTLsDeep", dependencies = TRUE)
```

Getting Started

Loading rTLsDeep and other required packages

```
# get pacman
install.packages("pacman")

#load pcaman and all packages
library(pacman)
    p_load(rTLsDeep,lidR,rgl,ggplot2,rgl,keras,reticulate,compiler)
```

TLS data processing

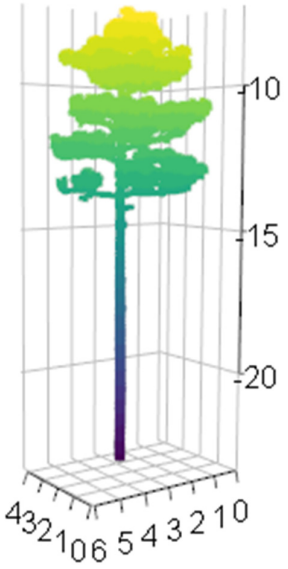

Figure S2. Example of TLS-derived 3d point cloud

Loading and visualizing TLS dataset

```
# Path to las file
lasfile <- system.file("extdata", "tree_c1.laz", package="rTLsDeep")

# Reading las file
las<-readLAS(lasfile)

# plotting las file in 3D
plot(las, bg="white")
rgl::axes3d(c("x+", "y-", "z-"), col="black")
      rgl::grid3d(side=c('x+', 'y-', 'z'), col="gray")
```

Figure S3. Illustration of the TLS-derived 3d point cloud

Rotating TLS-derived 3d point cloud

```
# Rotating around the x-axis
las<-tlsrotate3d(las,theta=120, by="x", scale=TRUE)

# Rotating around the y-axis
las<-tlsrotate3d(las,theta=120, by="y", scale=TRUE)

# Rotating around the z-axis
    las<-tlsrotate3d(las,theta=120, by="z", scale=TRUE)
```

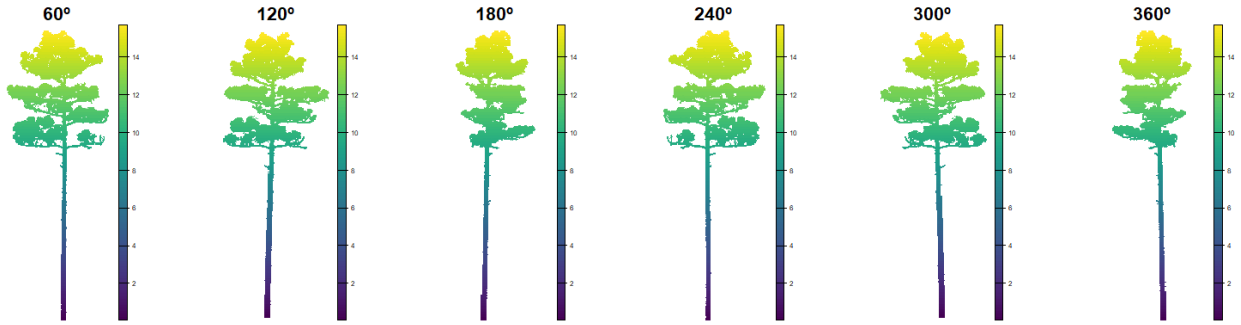

Figure S4. Illustration of 3d point cloud rotation

Capturing 2D grid snapshot

```
# Set output dir for downloading the example dataset files
outdir=getwd()

# downloading zip file
download.file("https://github.com/carlos-alberto-
    silva/rTLsDeep/tree/main/readme/laz_files.zip",destfile=file.path(outdir, "laz_files.zip"))

# unzip file
unzip(file.path(outdir,"laz_files.zip"))

# Reading las file for each post-hurricane individual tree-level damage classes
tree_c1<-readLAS(paste0(outdir,"//Tree_c1.laz"))
tree_c2<-readLAS(paste0(outdir,"//Tree_c2.laz"))
tree_c3<-readLAS(paste0(outdir,"//Tree_c3.laz"))
tree_c4<-readLAS(paste0(outdir,"//Tree_c4.laz"))
tree_c5<-readLAS(paste0(outdir,"//Tree_c5.laz"))
tree_c6<-readLAS(paste0(outdir,"//Tree_c6.laz"))

# Defining the func parameter
func = ~list(Z = max(Z)) # plot by height

# computing 2D grid snapshot
```

```
gtree_c1<-getTLS2D(tree_c1, res=0.05, by="xz", func = func, scale=TRUE)
gtree_c2<-getTLS2D(tree_c2, res=0.05, by="xz", func = func, scale=TRUE)
gtree_c3<-getTLS2D(tree_c3, res=0.05, by="xz", func = func, scale=TRUE)
gtree_c4<-getTLS2D(tree_c4, res=0.05, by="xz", func = func, scale=TRUE)
gtree_c5<-getTLS2D(tree_c5, res=0.05, by="xz", func = func, scale=TRUE)
gtree_c6<-getTLS2D(tree_c6, res=0.05, by="xz", func = func, scale=TRUE)

# Visualizing 2D grid snapshot
par(mfrow=c(2,3))
plot(gtree_c1, col=viridis::viridis(100),axes=FALSE, xlab="",ylab="", ylim=c(0,30), main="C1",cex=2)
plot(gtree_c2, col=viridis::viridis(100),axes=FALSE, xlab="",ylab="", ylim=c(0,30), main="C2",cex=2)
plot(gtree_c3, col=viridis::viridis(100),axes=FALSE, xlab="",ylab="", ylim=c(0,30), main="C3",cex=2)
plot(gtree_c4, col=viridis::viridis(100),axes=FALSE, xlab="",ylab="", ylim=c(0,30), main="C4",cex=2)
plot(gtree_c5, col=viridis::viridis(100),axes=FALSE, xlab="",ylab="", ylim=c(0,30), main="C5",cex=2)
plot(gtree_c6, col=viridis::viridis(100),axes=FALSE, xlab="",ylab="", ylim=c(0,30), main="C6",cex=2)

#Exporting 2D grid snapshot as tiff file
tiff("gtree_c1.tiff", units="in", width=5, height=5, res=300)
plot(gtree_c1, col=viridis::viridis(100),axes=FALSE, xlab="",ylab="", ylim=c(0,30), main="C1",cex=2)
    dev.off()
```

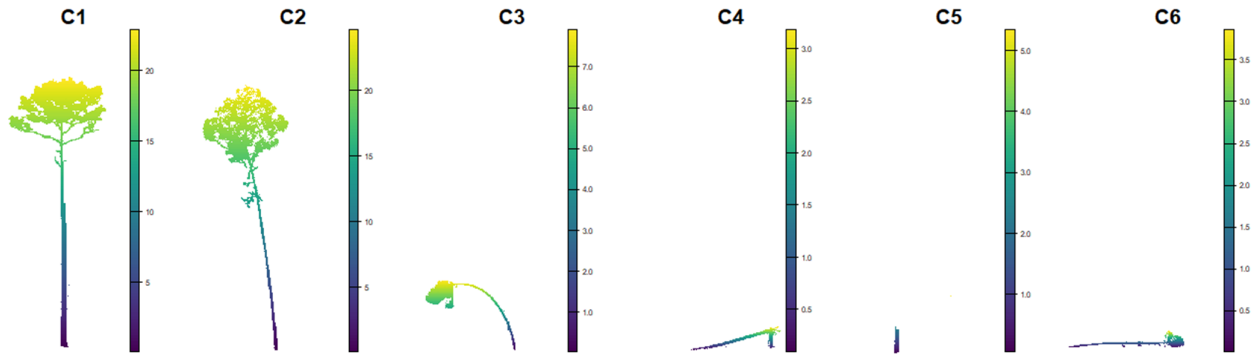

Figure S5. Illustration of the damage severity classes from TLS-derived 3d point cloud

Post-hurricane individual tree-level damage using deep learning

Selecting deep learning model properties

```
# Set directory to tensorflow (python environment)
# This is required if running deep learning local computer with GPU
# Guide to install here: https://doi.org/10.5281/zenodo.3929709
tensorflow_dir = '/apps/tensorflow/2.6.0'
```

```
# define model type
#model_type = "simple"
model_type = "vgg"
#model_type = "inception"
#model_type = "resnet"
#model_type = "densenet"
#model_type = "efficientnet"

# path to image folders - black
train_image_files_path <- getwd() # update the path for training datasets
test_image_files_path <- getwd() # update the path for testing datasets

# Image and model properties
img_width <- 256
img_height <- 256
class_list_train = unique(list.files(train_image_files_path))
class_list_test = unique(list.files(test_image_files_path))
lr_rate = 0.0001
target_size <- c(img_width, img_height)
channels <- 4
batch_size = 8L
epochs = 20L


# get model
model = get_dl_model(model_type=model_type,
                     img_width=img_width,
                     img_height=img_height,
                     lr_rate = lr_rate,
                     tensorflow_dir = tensorflow_dir,
                     class_list = class_list_train)
```

Model calibration

```
weights_fname = fit_dl_model(model = model,
                             train_input_path = train_image_files_path,
                             test_input_path = test_image_files_path,
                             target_size = target_size,
                             batch_size = batch_size,
                             class_list = class_list_train,
                             epochs = epochs,
                             lr_rate = lr_rate)
```

Predicting post-hurricane damage at the tree-level

```
tree_damage<-predict_treedamage(model = model,
                        input_file_path = test_image_files_path,
                        weights = weights,
                        target_size = c(256,256),
                        class_list=class_list_test,
                              batch_size = batch_size)
```

Confusion matrix

```
# Get damage classes for validation datasets
test_classes<-get_test_classes(file_path=test_image_files_path)

# Calculate confusion matrix
cm = confmatrix_treedamage(predict_class = tree_damage,
                        test_classes=test_classes,
                        class_list = class_list_test)
# Plot confusion matrix
gcmplot_vgg<-gcmplot(cm,
                  colors=c(low="white", high="#009194"),
                        title="densenet")
```

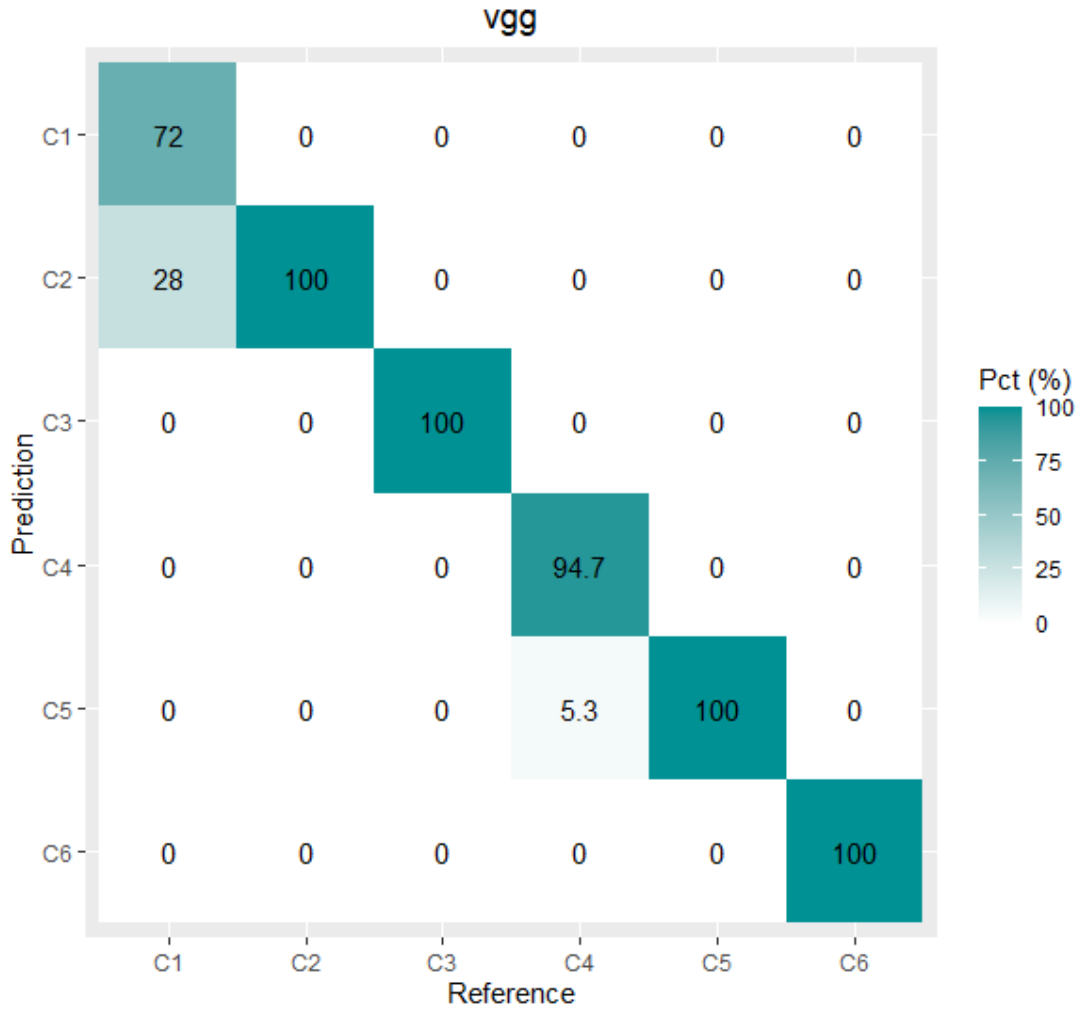

Figure S6. Confusion matrix for the post-hurricane damage severity classification