Group 3


- Yuting He
- Seungwook Han
- Shengwei Huang
- Mengran Xia
- Hongye Jiang



Summary: we finally chose the XGB model with gblinear as our booster. The parameters we chose for the gblinear boosters are: alpha =3 and lambda =0. We searched some open sources images in github, and we test our model with 50 images. The results are as follows:
MSE: 0.002511576 PSNR: 27.05951
Training time: 47.453 s. Testing time (for 50 random images): 11.602 s
Here's sample test result:


Low Resolution:

XGB:

High Resolution

Step by Step of our main.Rmd

First, load the packages

```{r}
if(!require("EBImage")){
  source("https://bioconductor.org/biocLite.R")
  biocLite("EBImage")
}
if(!require("gbm")){
  install.packages("gbm")
}
if(!require("xgboost")){
  install.packages("xgboost")
}
library("EBImage")
library("gbm")
library("xgboost")
```

We first load and install all required packages for this project and then we are going to specify the directories and set the working directory.

```
### Step 0: specify directories.

Set the working directory to the image folder. Specify the training and the testing set. For data without an independent test/validation set, you need to create your own testing data by random subsampling. In order to obain reproducible results, set.seed() whenever randomization is used.

```{r wkdir, eval=FALSE}
set.seed(2018)
#setwd("/Users/hyt/Documents/GitHub/Spring2019-Proj3-spring2019-proj3-grp3/doc")
#setwd("/Users/seungwookhan/Documents/Columbia/Senior/Applied Data
Science/Spring2019-Proj3-spring2019-proj3-grp3/")

# here replace it with your own path or manually set it in RStudio to where this rmd file is located.
# use relative path for reproducibility
getwd()
```

```
[1] "/Users/shengweihuang/Documents/GitHub/Spring2019-Proj3-spring2019-proj3-grp3/doc"
```

Provide directories for training images. Low-resolution (LR) image set and High-resolution (HR) image set will be in different subfolders.

```{r}
getwd()
train_dir <- "../data/train_set/" # This will be modified for different data sets.
#train_dir <- "../data/train_set_demo/" # This will be modified for different data sets.
(train_LR_dir <- paste(train_dir, "LR/", sep=""))
(train_HR_dir <- paste(train_dir, "HR/", sep=""))
(train_label_path <- paste(train_dir, "label.csv", sep=""))
```
```

Then, we are going to load the training images and the corresponding class label:

```
### Step 2: import training images class labels.

We provide extra information of image label: car (0), flower (1), market (2). These labels are not necessary for your model.

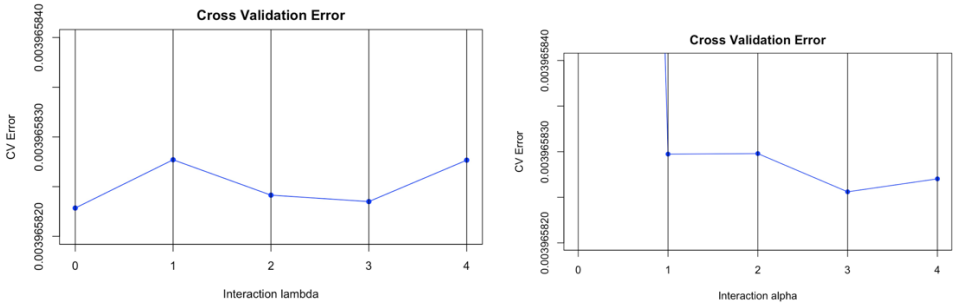
```{r train_label}
extra_label <- read.csv(train_label_path, colClasses=c("NULL", NA, NA))
```
```

Then we are going to construct the features and loading our training models:

```
```{r feature}
source("../lib/feature.R")
tm_feature_train <- NA
if(run.feature.train){
  tm_feature_train <- system.time(dat_train <- feature(train_LR_dir, train_HR_dir))
  feat_train <- dat_train$feature
  label_train <- dat_train$label
}
save(dat_train, file="../output/feature_train.RData")
```
```

```{r loadlib}
source("../lib/XGB/train.xgb.R")
source("../lib/XGB/test.xgb.R")
```

After that, we are doing the cross validation for our training models. Since we have tried different boosters for our XGBoost Model, we have performed the cross validation for both parameters when trying gbTree booster and gblinear booster:




Above are the cross validation results for the gblinear booster and we got our best parameters for the gblinear booster are: alpha=3 and lambda=0
Similarly we performed the cross validation for the parameters in gbTree booster and found that the best parameters are: max_depth = 5, n_rounds= 10, eta=0.5.

When we determined the best parameters, we changed our parameters to those optimal value and train our model.

```{r final_train}
tm_train=NA
par_best <- NULL
tm_train <- system.time(fit_train_xgboost <- train_xgboost(feat_train, label_train, par_best))
save(fit_train_xgboost, file="../output/fit_train_xgboost.RData")
```

We compared our model performance by testing some images we found online randomly and the results and codes are as follows:

```{r superresolution}
source("../lib/XGB/superResolution.xgb.R")
test_dir <- "../data/test_set/" # This will be modified for different data sets.
test_LR_dir <- paste(test_dir, "LR/", sep="")
test_HR_dir <- paste(test_dir, "HR/", sep="")
test_pred_dir <- paste(test_dir, "Pred/", sep="")
tm_test=NA
run.test <- T
if(run.test){
  load(file="../output/fit_train_xgboost.RData")
  tm_test <- system.time(superResolution(test_LR_dir, test_HR_dir, test_pred_dir, fit_train_xgboost))
}
```

| Booster | Mean Squared Error | PSNR |
|---|---|---|
| gbtree | 0.002593127 | 27.03771 |
| gblinear | 0.002511576 | 27.05951 |
| dart | 0.011715 | 19.42583 |

Thus from the results above, we choose our final model as the XGB booster with gblinear.