Vignette Title

Vignette Author

2017-09-04

The *wuepix* package counts visitor numbers using computer vision. Therefore three approaches were wrapped here. Addiotional management tools, as a Ground-Thruth-Data sampler, are also packed here.

```
library(wuepix)
library(tidyverse)
```

Site configuration

Paths & Filenames

Here the workflow has to be configured for each site. This means defining the directory paths and filename patterns implying tha data code.

```
# Where to find Images?
## Raw data
img.folder_raw <- "../Schorfheide/Kloster/IMG_raw/"
# Preprocessed (croped, scaled, enhanced,...)
img.folder  <- "IMG/"
# Remove corrupted images by filesize (in byte)
threshold <- 10000
# How to grep date?
gsub.Date <- function(Filename){gsub("picam-", "", gsub(".jpg", "", Filename))}
# Date code
date.code <- "%Y%m%d-%H%M"



# Hubland
# threshold <- 1000
# gsub.Date <- function(Filename){gsub("Camera1_M_", "", gsub(".jpg", "", Filename))}
# date.code <- "%Y-%m-%d-%H%M%S"
```

Extent of interest

To speed up processing an extend of interest (EOI) should be selected. Using the linux comandline tool *ImageMagick*, this can also include rotations aswell as other image operations. However identifying the correct comand involves visual interpreation of the results. To do so I proceeded as follows.

1. Using Gimp / Photoshop

Initially use *GIMP* to identify the preprocess routine (boundingbox, optional rotation).

> Tipp: Overlay several images to cover different scenarios.

Figure 1: Operational EOI

2. Test comandline

After identifying the preprocess routine try to put the parameters into *ImageMagick* and test comand on a single image using `convert`.

This results in the following extend of interest. Only this part of the image will be further analysed, so please only proceed if satisfied with the result.

3. Preprocess image archive

Next all images will be preprocessed according to the routine developed above using `mogrify`. Please pay attention to the slightly different syntax of the following command `mogrify -crop 2850x1000+0+980 -path IMG/ IMG_raw/*.jpg`. This will preprocess all images from `IMG_raw/` and save them in `IMG/`. (Less than 5 minutes for 506 images).

List images

First all images will be listed. The following chunk does so, plus enhances the data frame according to *Site configuration*: (1) due to external effects (eg. transmission) images can be corrupted. Here files with a file size smaller than the `threshold` will be exluded. (2) The Timestamp gets interpreted, therefore first the filenames are cropped with help of `gsub.Date`. Because filenames can be very different and the corresponding regular expression can very complex, it seemed easiest to do with a function. This also makes developing it more simple due better testing option. After cropping the timestamp it will be converted to a *POSIXlt* time object using `date.code`. (3) Last but not least the relative filepaths are reconstructed. Note, that this should also work with `list.files(..., fullnames=TRUE)` but I remeber then struggeling with grepping the datecode.

```
Files <- data.frame(Filename=list.files(img.folder, pattern = "*.jpg"),
                    stringsAsFactors = FALSE)
```

```
# Remove corrupted images
Files$Size <- file.size(paste0(img.folder, Files$Filename)) > threshold
Files <- Files[which(Files$Size),]
Files <- select(Files, -Size)

# Add Timestamp
Files$Timestamp <- strptime(gsub.Date(Files$Filename), date.code)
# Order by Timestamp
Files <- Files[order(Files$Timestamp),]

# Full Filename
Files$Filename <- paste0(img.folder, Files$Filename)
```

To get an overview about the data beeing processed, here some metadata summarys are promted.

```
## 21 files to analize
```

```
## Dates from 24.06.2017 12:30 to 24.06.2017 12:50
```

```
## Time difference of 20 mins
```

Ground-Truth-Data

Manually count pedestrians. This Ground-Truth-Data (GTD) will be utilized to access accuracies. Here all images (100%) got evaled.

```
start <- Sys.time()  # Get start time
#GTD <- GTD_list(sample(x = Files$Filename, size = 10))
#the.sample <- sample(c(1:nrow(Files)), size = 100)
#Files <- Files[the.sample,]
Files$GTD <- GTD_list(Files$Filename)
Files$GTD <- as.numeric(Files$GTD)
(Sys.time() - start)  # Print runtime

save(Files, file = "Results/GTD.RData")
write.csv(Files, file = "Results/GTD.csv")
```

```
load("Results/GTD.RData")

print(paste("Visitor number:", sum(Files$GTD)))
hist(Files$GTD)
```

Processing

Now the (preprocessed) image archive get processed.