

# Alabama Roads Project

## Background

This is a ‘proof of concept’ exercise that will attempt to predict road safety using the Google streetview api and a retrained (transfer learning based) version of Tensorflow to enable mass prediction of road safety on large datasets. The ultimate output would be a heat map/scoring of predicted road safety across Alabama.

## Take streetview photos

### Import roads

To do this we will use the road shapefile in the raw project data folder

~/Dropbox/pkg.data/alabama\_roads/raw/us82erik/us82\_dissolve. If you do not currently have access to the folder, you can download the folder by going to this Dropbox web link. Ideally, you should be able to dynamically link locally by using Dropbox desktop...

To build the streetview folders, the first step is to discretize the road shapefile and then use the google streetview metadata api to get locations of nearest google streetview picture (henceforth called pano\_id). In this case the picture will be manually tuned to have the car driving in the ‘correct’ direction. In the future, this can easily be done with tensorflow as well, by simply ensuring that the photo taken is on the right (not the left) side of the road. Again, this can be easily automated in tensorflow in the future for bigger data applications.

```
if(!(file.exists(roads.clean.location))){
  roads <- rgdal::readOGR(dsn = path.expand(roads.raw.location), layer = 'us82d', verbose = FALSE)
  roads <- sp::spTransform(roads, CRSobj =CRS(proj.env))

  saveRDS(roads, file = roads.clean.location)
} else {
  roads <- readRDS(roads.clean.location)
}
# Plot
map.center <- as.numeric(geosphere::centroid(rgeos::gBuffer(roads, width=0.1)))

## Warning in rgeos::gBuffer(roads, width = 0.1): Spatial object is not
## projected; GEOS expects planar coordinates

map <- ggmap::get_googlemap(center=c(lon=map.center[1], lat=map.center[2]), zoom=7)

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=32.890998,-87.106853&zoom=7&size=
roads_fortify <- ggplot2::fortify(roads)
p <- ggmap(map) +
  geom_line(data=roads_fortify, aes(x=long, y=lat, group=group))
print(p)
```



## Roads Discretize

This creates the points to find the panoids

```
sample.line <- function(x, sdist=100){
  if (!require(sp)) stop("sp PACKAGE MISSING")
  if (!inherits(x, "SpatialLinesDataFrame")) stop("MUST BE SP SpatialLinesDataFrame OBJECT")
  lgth <- SpatialLinesLengths(x)
  lsub <- x[1,]
  ns <- round( (lgth[1] / sdist), digits=0)
  lsamp <- spsample(lsub, n=ns, type="regular", offset=c(0.5,0.5))
  results <- SpatialPointsDataFrame(lsamp, data=data.frame(ID=rep(1:length(lsamp))))
  ( results )
}
road.lines <- sample.line(roads, sdist=0.001)
```

```
## Warning in .local(x, n, type, ...): working under the assumption of
## projected data!
```