library(dplyr)

library(tidyverse)

library(dplyr)

library(splines)

library(survminer)

library(survival)

library(lubridate)

library(shinythemes)

library(shiny)

library(ggplot2)

library(lazyeval)

library(tibble)

library(qdapRegex)

# have to determine 15 to 19

setwd("")

combination\_number<-1

Scorer\_Name<-"Scorer1"

Image\_Set<-"practice"

Data<-Eval\_Data\_All

dm<-dim(Data)

RowNumber=dm[1]

PersonToCompare<-1

Data2<-data.frame(Data$Model, Data$Filename, Data$Folder,Data$Status)

Data2$Model<-as.character(Data2$Model)

Data2$Data.Status<-as.character(Data2$Data.Status)

Data2$Data.Folder<-as.character(Data2$Data.Folder)

Data2$Data.Filename<-as.character(Data2$Data.Filename)

a<-Data$Model

Model<-unique(a)

c<-length(Model)

d<-Data2$Data.Folder

Scorer<-unique(d)

f<-length(Scorer)

#getting name of image rm\_between(Data2[1,2],"/",".",extract=TRUE)

Data2[,5]<-as.character(Scorer[PersonToCompare])

Data2[,6]<-as.character(substr(Data2$Data.Filename,1,1))

Data2[,7]<-as.character(ifelse(Data2[,5]!= Data2$Data.Folder,NA,ifelse(Data2[,6]=="A",0,1)))

for (i in 1:c){

Data2[,i+7]<-as.character(Model[i])

}

for (i in 1:c){

Data2[,i+7+c]<-as.character(ifelse(Data2[,i+7]==Data2$Data.Model & Data2$Data.Status=="FALSE",1,ifelse(Data2[,i+7]==Data2$Data.Model & Data2$Data.Status=="TRUE",0,NA)))

}

for (i in 1:c){

Data2[,i+7+2\*c]<-as.numeric(ifelse(Data2[,i+7+c]==Data2[,7],1,0))

}

ImageID\_Scorer<-ifelse(Data2[,5]==Data2$Data.Folder,1,0)

Total\_Images\_Scored\_Per\_Model<-(sum(ImageID\_Scorer))/c

for (i in 1:c){

Data2[1,i+7+3\*c]<-sum(Data2[,i+7+2\*c],na.rm = TRUE)

}

for (i in 1:c){

Data2[1,i+7+4\*c]<-Data2[1,i+7+3\*c]/Total\_Images\_Scored\_Per\_Model\*100

}

#Alive by person as Alive by computer

#Alive by manual

Data2[,8+5\*c]<-ifelse(Data2[,7]==0,1,NA)

Data2[1,9+5\*c]<-sum(Data2[,8+5\*c],na.rm = TRUE)

Data2[1,10+5\*c]<-Data2[1,9+5\*c]/c

Manual\_Alive<-as.numeric(Data2[1,10+5\*c])

#agreement of those marked alive manually and marked alive by Neural Network

for (i in 1:c){

Data2[,i+10+5\*c]<-ifelse(Data2[,7]==0,Data2[,i+7+2\*c],NA)

}

for (i in 1:c){

Data2[1,i+10+6\*c]<-sum(Data2[,i+10+5\*c],na.rm = TRUE)

}

#those marked alive manually and dead by Neural Network

for (i in 1:c){

Data2[1,i+10+7\*c]<-Data2[1,10+5\*c]-Data2[1,i+10+6\*c]

}

#those marked dead manually

Data2[,11+8\*c]<-ifelse(Data2[,7]==1,1,NA)

Data2[1,12+8\*c]<-sum(Data2[,11+8\*c],na.rm = TRUE)

Data2[1,13+8\*c]<-Data2[1,12+8\*c]/c

Manual\_Dead<-as.numeric(Data2[1,13+8\*c])

#those marked dead manually and dead by neural network

for (i in 1:c){

Data2[,i+13+8\*c]<-ifelse(Data2[,7]==1,Data2[,i+7+2\*c],NA)

}

for (i in 1:c){

Data2[1,i+13+9\*c]<-sum(Data2[,i+13+8\*c],na.rm = TRUE)

}

#those marked dead manually and alive by neural network

for (i in 1:c){

Data2[1,i+13+10\*c]<-Data2[1,13+8\*c]-Data2[1,i+13+9\*c]

}

#percentages of alive manual marked alive by neural network

for (i in 1:c){

Data2[1,i+13+11\*c]<-Data2[1,i+10+6\*c]/Data2[1,10+5\*c]\*100

}

#percentages of alive manual marked dead by neural network

for (i in 1:c){

Data2[1,i+13+12\*c]<-Data2[1,i+10+7\*c]/Data2[1,10+5\*c]\*100

}

#percentages of dead manual marked dead by neural network

for (i in 1:c){

Data2[1,i+13+13\*c]<-Data2[1,i+13+9\*c]/Data2[1,13+8\*c]\*100

}

#percentages of dead manual marked dead by neural network

for (i in 1:c){

Data2[1,i+13+14\*c]<-Data2[1,i+13+10\*c]/Data2[1,13+8\*c]\*100

}

#write.csv(Data2, "Data2\_trialb.csv")

dmData2<-dim(Data2)

rowData2<-dmData2[1]

colData2<-dmData2[2]

accuracy<-matrix(0,ncol=c, nrow = 11)

Accuracy<-as.data.frame(accuracy)

for (i in 1:c){

colnames(Accuracy)[i]<-Model[i]

}

row.names(Accuracy)[1]<-paste0("Manual(",Scorer\_Name,") Scored Alive Scored Alive by Neural Network")

row.names(Accuracy)[2]<-"Manual Scored Alive Scored Dead by Neural Network"

row.names(Accuracy)[3]<-"Manual Alive Scored"

row.names(Accuracy)[4]<-"Manual Scored Dead Scored Dead by Neural Network"

row.names(Accuracy)[5]<-"Manual Scored Dead Scored Alive by Neural Network"

row.names(Accuracy)[6]<-"Manual Dead Scored"

row.names(Accuracy)[7]<-"Percent Accuracy of All Cells"

row.names(Accuracy)[8]<-"Percent Manual Scored Alive Scored Alive by Neural Network"

row.names(Accuracy)[9]<-"Percent Manual Scored Alive Scored Dead by Neural Network"

row.names(Accuracy)[10]<-"Percent Manual Scored Dead Scored Dead by Neural Network"

row.names(Accuracy)[11]<-"Percent Manual Scored Dead Scored Alive by Neural Network"

for (i in 1:c){

Accuracy[1,i]<-paste(Data2[1,i+10+6\*c])

}

for (i in 1:c){

Accuracy[2,i]<-paste(Data2[1,i+10+7\*c])

}

for (i in 1:c){

Accuracy[3,i]<-paste(Data2[1,10+5\*c])

}

for (i in 1:c){

Accuracy[4,i]<-paste(Data2[1,i+13+9\*c])

}

for (i in 1:c){

Accuracy[5,i]<-paste(Data2[1,i+13+10\*c])

}

for (i in 1:c){

Accuracy[6,i]<-paste(Data2[1,13+8\*c])

}

for (i in 1:c){

Accuracy[7,i]<-paste(Data2[1,i+7+4\*c])

}

for (i in 1:c){

Accuracy[8,i]<-paste(Data2[1,i+13+11\*c])

}

for (i in 1:c){

Accuracy[9,i]<-paste(Data2[1,i+13+12\*c])

}

for (i in 1:c){

Accuracy[10,i]<-paste(Data2[1,i+13+13\*c])

}

for (i in 1:c){

Accuracy[11,i]<-paste(Data2[1,i+13+14\*c])

}

#write.csv(Accuracy,paste("Accuracy\_for\_Individual\_Models\_Compared\_To\_",Scorer\_Name,"\_",Image\_Set,".csv"))

Data2matrix<-matrix(0,ncol=colData2, nrow = rowData2)

Data3<-as.data.frame(Data2matrix)

rrr<-sub("^","Cell\_Status",Model)

for (i in 1:c){

colnames(Data3)[i+7+c]<-rrr[i]

}

for (i in 1:c){

Data3[,i+7+c]<-paste(Data2[,i+7+c])

}

colnames(Data3)[1]<-"Model\_Used"

colnames(Data3)[2]<-"File\_Evaluated"

colnames(Data3)[3]<-"Folder"

colnames(Data3)[4]<-"Status\_Provided\_by\_Model"

colnames(Data3)[5]<-"Person\_Comparing\_To"

colnames(Data3)[6]<-"Status\_Provided\_by\_Person"

colnames(Data3)[7]<-"Status\_by\_Person\_Numerical"

for (i in 1:7){

Data3[,i]<-paste(Data2[,i])

}

for (i in 1:c){

Data3[,i+7]<-Data2[,i+7]

}

sss<-sub("^","Agreement(1)\_with\_Person\_",Model)

for (i in 1:c){

colnames(Data3)[i+7+2\*c]<-sss[i]

}

for (i in 1:c){

Data3[,i+7+2\*c]<-Data2[,i+7+2\*c]

}

tt<-sub("^","Sum\_ofAgreement\_withPerson",Model)

uu<-sub("^","Percent\_Agreement\_withPerson",Model)

for (i in 1:c){

colnames(Data3)[i+7+3\*c]<-tt[i]

}

for (i in 1:c){

colnames(Data3)[i+7+4\*c]<-uu[i]

}

for (i in 1:c){

Data3[1,i+7+3\*c]<-sum(Data3[,i+7+2\*c],na.rm = TRUE)

}

for (i in 1:c){

Data3[1,i+7+4\*c]<-Data3[1,i+7+3\*c]/Total\_Images\_Scored\_Per\_Model

}

#Individual Scoring

m<-matrix(0,ncol=c, nrow =1 )

Individual\_Scoring<-as.data.frame(m)

for (i in 1:c){

colnames(Individual\_Scoring)[i]<-Model[i]

}

for (i in 1:c){

Individual\_Scoring[1,i]<-paste(Data2[1,i+7+4\*c])

}

gg<-Scorer\_Name

ff<-paste("Accuracy\_Compared to ",gg, sep="")

rownames(Individual\_Scoring)[1]<-ff

#write.csv(Individual\_Scoring, paste0("Scoring\_Accuracy\_of\_Individual\_Models\_Compared\_To",gg,Image\_Set,".csv"))

# Combination Scoring for both positive and negative accuracy

mm<-matrix(0,ncol=c, nrow =dm )

Agreement\_with\_Manual<-as.data.frame(mm)

for (i in 1:c){

Agreement\_with\_Manual[,i]<-paste(Data2[,i+7+2\*c])

}

for (i in 1:c){

colnames(Agreement\_with\_Manual)[i]<-Model[i]

}

Agreement\_with\_Manual$Data.Filename<-Data2$Data.Filename

Agreement\_with\_Manual$Filename<-rm\_between(Agreement\_with\_Manual$Data.Filename,"/",".",extract=TRUE)

www<-RowNumber/(c\*f)

mmm<-matrix(0,ncol=c, nrow =www )

Agreement\_with\_Manual\_Sorted<-as.data.frame(mmm)

for (i in 1:c){

colnames(Agreement\_with\_Manual\_Sorted)[i]<-sss[i]

}

for (i in 1:c){

ifelse(i==1,Agreement\_with\_Manual\_Sorted[1:www,i]<-Agreement\_with\_Manual[1:www,i],Agreement\_with\_Manual\_Sorted[1:www,i]<-Agreement\_with\_Manual[(((i-1)\*www)+1):(www\*i),i])

}

#write.csv(Agreement\_with\_Manual\_Sorted, "Agreement\_with\_Manual\_Sorted.csv")

mmmmmm<-matrix(0,ncol=c, nrow =www )

Agreement\_with\_Manual\_Sorted2<-data.frame(mmmmmm)

for (i in 1:c){

Agreement\_with\_Manual\_Sorted2[,i]<-as.numeric(Agreement\_with\_Manual\_Sorted[,i])

}

k<- combn(Agreement\_with\_Manual\_Sorted2, m = combination\_number, FUN = rowSums, simplify = TRUE)

j<-combn(Model,m = combination\_number, FUN = NULL, simplify = TRUE )

jj<-as.data.frame(j)

xy<-dim(j)

xx<-xy[2]

zz<-xy[1]

zzz<-zz+1

possiblecombinations<-xx

Sum\_of\_Agreements\_And\_Disagreements<-as.data.frame(k)

#write.csv(Sum\_of\_Agreements\_And\_Disagreements,"Sum\_of\_Agreements\_And\_Disagreements.csv")

Average\_of\_Agreements\_And\_Disagreements<-as.data.frame(k/combination\_number)

#write.csv(Average\_of\_Agreements\_And\_Disagreements,"Average\_of\_Agreements\_and\_Disagreement.csv")

Combinations\_Agree\_orDisagree<-as.data.frame(ifelse(Average\_of\_Agreements\_And\_Disagreements>0.5,1,0))

#write.csv(Combinations\_Agree\_orDisagree, "Combinations\_Agree\_orDisagree.csv")

mmmmmmm<-matrix(0,ncol=possiblecombinations, nrow =1 )

Combinations\_Agree\_orDisagree\_Sum<-data.frame(mmmmmmm)

for (i in 1:possiblecombinations){

Combinations\_Agree\_orDisagree\_Sum[1,i]<-sum(Combinations\_Agree\_orDisagree[,i])

}

#write.csv(Combinations\_Agree\_orDisagree\_Sum, "Combinations\_Agree\_or\_Disagree\_Sum")

Combinations\_Agree\_orDisagree\_Mean\_Accuracy<-data.frame()

for (i in 1:possiblecombinations){

Combinations\_Agree\_orDisagree\_Mean\_Accuracy[2,i]<-Combinations\_Agree\_orDisagree\_Sum[,i]/Total\_Images\_Scored\_Per\_Model\*100

}

combomodel<-matrix(0,ncol=possiblecombinations, nrow =1 )

Names\_Model\_Combo<-data.frame(combomodel)

for (i in 1:possiblecombinations){

combomodel[1,i]<-paste(j[1:combination\_number,i],collapse = "")

}

for (i in 1:possiblecombinations){

Combinations\_Agree\_orDisagree\_Mean\_Accuracy[1,i]<-combomodel[,i]

}

#write.csv(Combinations\_Agree\_orDisagree\_Mean\_Accuracy,paste0("Scoring\_Accuracy\_Overall",combination\_number,"\_model\_combinations\_compared\_to",gg,Image\_Set,".csv"))

#combination scoring for alive cells

alive<-matrix(0,ncol=possiblecombinations, nrow =dm )

Agreement\_with\_Manual\_alive<-as.data.frame(alive)

for (i in 1:c){

Agreement\_with\_Manual\_alive[,i]<-paste( Data2[,i+10+5\*c])

}

for (i in 1:c){

colnames(Agreement\_with\_Manual\_alive)[i]<-Model[i]

}

Agreement\_with\_Manual\_alive$Data.Filename<-Data2$Data.Filename

Agreement\_with\_Manual\_alive$Filename<-rm\_between(Agreement\_with\_Manual\_alive$Data.Filename,"/",".",extract=TRUE)

www<-RowNumber/(c\*f)

alivesorted<-matrix(0,ncol=c, nrow = www )

Agreement\_with\_Manual\_alive\_Sorted<-as.data.frame(alivesorted)

for (i in 1:c){

colnames(Agreement\_with\_Manual\_alive\_Sorted)[i]<-sss[i]

}

for (i in 1:c){

ifelse(i==1,Agreement\_with\_Manual\_alive\_Sorted[1:www,i]<-Agreement\_with\_Manual\_alive[1:www,i],Agreement\_with\_Manual\_alive\_Sorted[1:www,i]<-Agreement\_with\_Manual\_alive[(((i-1)\*www)+1):(www\*i),i])

}

#write.csv(Agreement\_with\_Manual\_alive\_Sorted, "Agreement\_with\_Manual\_Alive\_Sorted.csv")

mmmmmm<-matrix(0,ncol=c, nrow =www )

Agreement\_with\_Manual\_Sorted\_aliveB<-data.frame(mmmmmm)

for (i in 1:c){

Agreement\_with\_Manual\_Sorted\_aliveB[,i]<-ifelse(Agreement\_with\_Manual\_alive\_Sorted[,i]=="NA","",Agreement\_with\_Manual\_alive\_Sorted[,i])

}

for (i in 1:c){

colnames(Agreement\_with\_Manual\_Sorted\_aliveB)[i]<-sss[i]

}

#write.csv(Agreement\_with\_Manual\_Sorted\_aliveB, "Agreement\_with\_Manual\_Alive\_SortedB.csv")

mmmmmm<-matrix(0,ncol=c, nrow =www )

Agreement\_with\_Manual\_Sorted\_aliveC<-data.frame(mmmmmm)

for (i in 1:c){

Agreement\_with\_Manual\_Sorted\_aliveC[,i]<-as.numeric(Agreement\_with\_Manual\_Sorted\_aliveB[,i])

}

k<- combn(Agreement\_with\_Manual\_Sorted\_aliveC, m = combination\_number, FUN = rowSums, simplify = TRUE)

j<-combn(Model,m = combination\_number, FUN = NULL, simplify = TRUE )

jj<-as.data.frame(j)

xy<-dim(j)

xx<-xy[2]

zz<-xy[1]

zzz<-zz+1

possiblecombinations<-xx

Sum\_of\_Agreements\_And\_Disagreements\_Alive<-as.data.frame(k)

#write.csv(Sum\_of\_Agreements\_And\_Disagreements\_Alive,"Sum\_of\_Agreements\_And\_Disagreements\_Alive.csv")

Average\_of\_Agreements\_And\_Disagreements\_Alive<-as.data.frame(k/combination\_number)

#write.csv(Average\_of\_Agreements\_And\_Disagreements\_Alive,"Average\_of\_Agreements\_and\_Disagreement\_Alive.csv")

AandDalive<-matrix(0,ncol=possiblecombinations, nrow =www)

Average\_of\_Agreements\_And\_DisagreementsB\_alive<-data.frame(AandDalive)

for (i in 1:possiblecombinations){

Average\_of\_Agreements\_And\_DisagreementsB\_alive[,i]<-Average\_of\_Agreements\_And\_Disagreements\_Alive[,i]

}

Average\_of\_Agreements\_And\_DisagreementsB\_alive[is.na(Average\_of\_Agreements\_And\_DisagreementsB\_alive)]<-""

#write.csv(Average\_of\_Agreements\_And\_DisagreementsB\_alive,"Average\_of\_Agreements\_and\_Disagreement\_AliveB.csv")

Combinations\_Agree\_orDisagree\_Alive<-as.data.frame(ifelse(Average\_of\_Agreements\_And\_DisagreementsB\_alive>0.5,1,0))

#write.csv(Combinations\_Agree\_orDisagree\_Alive, "Combinations\_Agree\_orDisagree\_Alive.csv")

mmmmmmm<-matrix(0,ncol=possiblecombinations, nrow =1 )

Combinations\_Agree\_orDisagree\_Sum\_Alive<-data.frame(mmmmmmm)

for (i in 1:possiblecombinations){

Combinations\_Agree\_orDisagree\_Sum\_Alive[1,i]<-sum(Combinations\_Agree\_orDisagree\_Alive[,i])

}

#write.csv(Combinations\_Agree\_orDisagree\_Sum\_Alive, "Combinations\_Agree\_or\_Disagree\_Sum\_Alive.csv")

Combinations\_Agree\_orDisagree\_Mean\_Accuracy\_Alive<-data.frame()

for (i in 1:possiblecombinations){

Combinations\_Agree\_orDisagree\_Mean\_Accuracy\_Alive[2,i]<-Combinations\_Agree\_orDisagree\_Sum\_Alive[,i]/Manual\_Alive

}

combomodel<-matrix(0,ncol=possiblecombinations, nrow =1 )

Names\_Model\_Combo<-data.frame(combomodel)

for (i in 1:possiblecombinations){

combomodel[1,i]<-paste(j[1:combination\_number,i],collapse = "")

}

for (i in 1:possiblecombinations){

Combinations\_Agree\_orDisagree\_Mean\_Accuracy\_Alive[1,i]<-combomodel[,i]

}

#write.csv(Combinations\_Agree\_orDisagree\_Mean\_Accuracy\_Alive,paste0("Scoring\_Accuracy\_Alive",combination\_number,"\_model\_combinations\_compared\_to",gg,Image\_Set,".csv"))

combomodel2<-matrix(0,ncol=possiblecombinations, nrow =1 )

Combinations\_Disagree\_ManualAlive<-data.frame(combomodel2)

for (i in 1:possiblecombinations){

Combinations\_Disagree\_ManualAlive[,i]<-Manual\_Alive- Combinations\_Agree\_orDisagree\_Sum\_Alive[,i]

}

combomodel3<-matrix(0,ncol=possiblecombinations, nrow =1 )

Combinations\_Disagree\_ManualAlivePercent<-data.frame(combomodel3)

for (i in 1:possiblecombinations){

Combinations\_Disagree\_ManualAlivePercent[,i]<-Combinations\_Disagree\_ManualAlive[,i]/Manual\_Alive\*100

}

#scoring accuracy combinations for dead

dead<-matrix(0,ncol=possiblecombinations, nrow =dm )

Agreement\_with\_Manual\_dead<-as.data.frame(alive)

for (i in 1:c){

Agreement\_with\_Manual\_dead[,i]<-paste(Data2[,i+13+8\*c])

}

for (i in 1:c){

colnames(Agreement\_with\_Manual\_dead)[i]<-Model[i]

}

Agreement\_with\_Manual\_dead$Data.Filename<-Data2$Data.Filename

Agreement\_with\_Manual\_dead$Filename<-rm\_between(Agreement\_with\_Manual\_dead$Data.Filename,"/",".",extract=TRUE)

www<-RowNumber/(c\*f)

deadsorted<-matrix(0,ncol=c, nrow = www )

Agreement\_with\_Manual\_dead\_sorted<-as.data.frame(deadsorted)

for (i in 1:c){

colnames(Agreement\_with\_Manual\_dead\_sorted)[i]<-sss[i]

}

for (i in 1:c){

ifelse(i==1,Agreement\_with\_Manual\_dead\_sorted[1:www,i]<-Agreement\_with\_Manual\_dead[1:www,i],Agreement\_with\_Manual\_dead\_sorted[1:www,i]<-Agreement\_with\_Manual\_dead[(((i-1)\*www)+1):(www\*i),i])

}

#write.csv(Agreement\_with\_Manual\_dead\_sorted, "Agreement\_with\_Manual\_Dead\_Sorted.csv")

mmmmmm<-matrix(0,ncol=c, nrow =www )

Agreement\_with\_Manual\_Sorted\_deadB<-data.frame(mmmmmm)

for (i in 1:c){

Agreement\_with\_Manual\_Sorted\_deadB[,i]<-ifelse(Agreement\_with\_Manual\_dead\_sorted[,i]=="NA","",Agreement\_with\_Manual\_dead\_sorted[,i])

}

for (i in 1:c){

colnames(Agreement\_with\_Manual\_Sorted\_deadB)[i]<-sss[i]

}

#write.csv(Agreement\_with\_Manual\_Sorted\_deadB, "Agreement\_with\_Manual\_dead\_SortedB.csv")

mmmmmm<-matrix(0,ncol=c, nrow =www )

Agreement\_with\_Manual\_Sorted\_deadC<-data.frame(mmmmmm)

for (i in 1:c){

Agreement\_with\_Manual\_Sorted\_deadC[,i]<-as.numeric(Agreement\_with\_Manual\_Sorted\_deadB[,i])

}

k<- combn(Agreement\_with\_Manual\_Sorted\_deadC, m = combination\_number, FUN = rowSums, simplify = TRUE)

j<-combn(Model,m = combination\_number, FUN = NULL, simplify = TRUE )

jj<-as.data.frame(j)

xy<-dim(j)

xx<-xy[2]

zz<-xy[1]

zzz<-zz+1

possiblecombinations<-xx

Sum\_of\_Agreements\_And\_Disagreements\_Dead<-as.data.frame(k)

#write.csv(Sum\_of\_Agreements\_And\_Disagreements\_Dead,"Sum\_of\_Agreements\_And\_Disagreements\_Dead.csv")

Average\_of\_Agreements\_And\_Disagreements\_Dead<-as.data.frame(k/combination\_number)

#write.csv(Average\_of\_Agreements\_And\_Disagreements\_Dead,"Average\_of\_Agreements\_and\_Disagreement\_Dead.csv")

AandD\_dead<-matrix(0,ncol=possiblecombinations, nrow =www)

Average\_of\_Agreements\_And\_DisagreementsB\_dead<-data.frame(AandD\_dead)

for (i in 1:possiblecombinations){

Average\_of\_Agreements\_And\_DisagreementsB\_dead[,i]<-Average\_of\_Agreements\_And\_Disagreements\_Dead[,i]

}

Average\_of\_Agreements\_And\_DisagreementsB\_dead[is.na(Average\_of\_Agreements\_And\_DisagreementsB\_dead)]<-""

#write.csv(Average\_of\_Agreements\_And\_DisagreementsB\_dead,"Average\_of\_Agreements\_and\_Disagreement\_Dead\_B.csv")

Combinations\_Agree\_orDisagree\_Dead<-as.data.frame(ifelse(Average\_of\_Agreements\_And\_DisagreementsB\_dead>0.5,1,0))

#write.csv(Combinations\_Agree\_orDisagree\_Dead, "Combinations\_Agree\_orDisagree\_Dead.csv")

mmmmmmm<-matrix(0,ncol=possiblecombinations, nrow =1 )

Combinations\_Agree\_orDisagree\_Sum\_Dead<-data.frame(mmmmmmm)

for (i in 1:possiblecombinations){

Combinations\_Agree\_orDisagree\_Sum\_Dead[1,i]<-sum(Combinations\_Agree\_orDisagree\_Dead[,i])

}

#write.csv(Combinations\_Agree\_orDisagree\_Sum\_Dead, "Combinations\_Agree\_or\_Disagree\_Sum\_Dead.csv")

Combinations\_Agree\_orDisagree\_Mean\_Accuracy\_Dead<-data.frame()

for (i in 1:possiblecombinations){

Combinations\_Agree\_orDisagree\_Mean\_Accuracy\_Dead[2,i]<-Combinations\_Agree\_orDisagree\_Sum\_Dead[,i]/Manual\_Dead

}

combomodel<-matrix(0,ncol=possiblecombinations, nrow =1 )

Names\_Model\_Combo<-data.frame(combomodel)

for (i in 1:possiblecombinations){

combomodel[1,i]<-paste(j[1:combination\_number,i],collapse = "")

}

for (i in 1:possiblecombinations){

Combinations\_Agree\_orDisagree\_Mean\_Accuracy\_Dead[1,i]<-combomodel[,i]

}

#write.csv(Combinations\_Agree\_orDisagree\_Mean\_Accuracy\_Dead,paste0("Scoring\_Accuracy\_Dead",combination\_number,"\_model\_combinations\_compared\_to",gg,Image\_Set,".csv"))

combomodel9<-matrix(0,ncol=possiblecombinations, nrow =1 )

Combinations\_Disagree\_ManualDead<-data.frame(combomodel9)

for (i in 1:possiblecombinations){

Combinations\_Disagree\_ManualDead[1,i]<-Manual\_Dead-Combinations\_Agree\_orDisagree\_Sum\_Dead[,i]

}

combomodel10<-matrix(0,ncol=possiblecombinations, nrow =1 )

Combinations\_Disagree\_ManualDead\_Percent<-data.frame(combomodel10)

for (i in 1:possiblecombinations){

Combinations\_Disagree\_ManualDead\_Percent[1,i]<-Combinations\_Disagree\_ManualDead[,i]/Manual\_Dead\*100

}

combomodel11<-matrix(0,ncol=possiblecombinations, nrow =1 )

Precision<-data.frame(combomodel11)

for (i in 1:possiblecombinations){

Precision[1,i]<-Manual\_Alive/(Manual\_Alive+Combinations\_Disagree\_ManualDead[,i])

}

combomodel12<-matrix(0,ncol=possiblecombinations, nrow =1 )

Recall\_<-data.frame(combomodel12)

for (i in 1:possiblecombinations){

Recall\_[1,i]<-Manual\_Alive/(Manual\_Alive+Manual\_Alive-Combinations\_Agree\_orDisagree\_Sum\_Alive[1,i])

}

combomodel13<-matrix(0,ncol=possiblecombinations, nrow =1 )

F1\_score<-data.frame(combomodel13)

for (i in 1:possiblecombinations){

F1\_score[1,i]<-2\*Precision[1,i]\*Recall\_[1,i]/(Precision[1,i]+Recall\_[1,i])

}

#output for combinations of correct and incorrect alive and dead

accuracycombo<-matrix(0,ncol=possiblecombinations, nrow = 15)

AccuracyCombination<-as.data.frame(accuracycombo)

for (i in 1:c){

colnames(Accuracy)[i]<-Model[i]

}

row.names(AccuracyCombination)[1]<-"Neural Networks Combined"

row.names(AccuracyCombination)[2]<-paste0("Manual(",Scorer\_Name,") Scored Alive Scored Alive by Neural Networks")

row.names(AccuracyCombination)[3]<-"Manual Scored Alive Scored Dead by Neural Networks"

row.names(AccuracyCombination)[4]<-"Manual Alive Scored"

row.names(AccuracyCombination)[5]<-"Manual Scored Dead Scored Dead by Neural Networks"

row.names(AccuracyCombination)[6]<-"Manual Scored Dead Scored Alive by Neural Networks"

row.names(AccuracyCombination)[7]<-"Manual Dead Scored"

row.names(AccuracyCombination)[8]<-"Percent Accuracy of All Cells"

row.names(AccuracyCombination)[9]<-"Percent Manual Scored Alive Scored Alive by Neural Networks"

row.names(AccuracyCombination)[10]<-"Percent Manual Scored Alive Scored Dead by Neural Networks"

row.names(AccuracyCombination)[11]<-"Percent Manual Scored Dead Scored Dead by Neural Networks"

row.names(AccuracyCombination)[12]<-"Percent Manual Scored Dead Scored Alive by Neural Networks"

row.names(AccuracyCombination)[13]<-"Precision"

row.names(AccuracyCombination)[14]<-"Recall"

row.names(AccuracyCombination)[15]<-"F1\_score"

for (i in 1:possiblecombinations){

AccuracyCombination[1,i]<-paste(j[1:combination\_number,i],collapse = "")

}

for (i in 1:possiblecombinations){

AccuracyCombination[2,i]<-paste(Combinations\_Agree\_orDisagree\_Sum\_Alive[1,i])

}

for (i in 1:possiblecombinations){

AccuracyCombination[3,i]<-as.numeric(Manual\_Alive-Combinations\_Agree\_orDisagree\_Sum\_Alive[1,i])

}

for (i in 1:possiblecombinations){

AccuracyCombination[4,i]<-paste(Manual\_Alive)

}

for (i in 1:possiblecombinations){

AccuracyCombination[5,i]<-paste(Combinations\_Agree\_orDisagree\_Sum\_Dead[1,i])

}

for (i in 1:possiblecombinations){

AccuracyCombination[6,i]<-paste(Manual\_Dead-Combinations\_Agree\_orDisagree\_Sum\_Dead[1,i])

}

for (i in 1:possiblecombinations){

AccuracyCombination[7,i]<-paste(Manual\_Dead)

}

for (i in 1:possiblecombinations){

AccuracyCombination[8,i]<-paste(Combinations\_Agree\_orDisagree\_Mean\_Accuracy[2,i])

}

for (i in 1:possiblecombinations){

AccuracyCombination[9,i]<-paste(Combinations\_Agree\_orDisagree\_Sum\_Alive[1,i]/Manual\_Alive\*100)

}

for (i in 1:possiblecombinations){

AccuracyCombination[10,i]<-paste(Combinations\_Disagree\_ManualAlivePercent[1,i])

}

for (i in 1:possiblecombinations){

AccuracyCombination[11,i]<-paste(Combinations\_Agree\_orDisagree\_Sum\_Dead[1,i]/Manual\_Dead\*100)

}

for (i in 1:possiblecombinations){

AccuracyCombination[12,i]<-paste(Combinations\_Disagree\_ManualDead\_Percent[1,i])

}

for (i in 1:possiblecombinations){

AccuracyCombination[13,i]<-paste(Precision[1,i])

}

for (i in 1:possiblecombinations){

AccuracyCombination[14,i]<-paste(Recall\_[1,i])

}

for (i in 1:possiblecombinations){

AccuracyCombination[15,i]<-paste(F1\_score[1,i])

}

write.csv(AccuracyCombination,paste("Accuracy\_for\_Combination\_of\_",combination\_number,"\_Models\_Compared\_To\_",Scorer\_Name,'\_',Image\_Set,".csv"))