**IMAGE CLASSIFICATION IN R**

Code –

library(EBImage)

library(keras)

library(prob)

library(entropy)

library(stringi)

setwd("C:\\Users\\mailt\\OneDrive\\Desktop\\R\\cps")

pics <- c('c1.jpg','c2.jpg','c3.jpg','c4.jpg','c5.jpg','c6.jpg','c7.jpg','c8.jpg','c9.jpg',

          'd1.jpg','d2.jpg','d3.jpg','d4.jpg','d5.jpg','d6.jpg','d7.jpg','d8.jpg','d9.jpg',

          'm1.jpg','m2.jpg','m3.jpg','m4.jpg','m5.jpg','m6.jpg','m7.jpg','m8.jpg','m9.jpg',

          'h1.jpg','h2.jpg','h3.jpg','h4.jpg','h5.jpg','h6.jpg','h7.jpg','h8.jpg','h9.jpg')

mypic<-list()

for (i in 1:36) {mypic[[i]] <- readImage(pics[i])}

print(mypic[[1]])

display(mypic[[8]])

# Resize

for (i in 1:36) {mypic[[i]] <- resize(mypic[[i]], 28, 28)}

# Reshape

for (i in 1:36) {mypic[[i]] <- array\_reshape(mypic[[i]],c(28,28,3))}

str(mypic)

# Row Bind

trainx <- NULL

for (i in 1:8)   {trainx <- rbind(trainx, mypic[[i]])}

for (i in 10:17) {trainx <- rbind(trainx, mypic[[i]])}

for (i in 19:26) {trainx <- rbind(trainx, mypic[[i]])}

for (i in 28:35) {trainx <- rbind(trainx, mypic[[i]])}

str(trainx)

#trainx

testx <- rbind(mypic[[9]],mypic[[18]],mypic[[27]],mypic[[36]])

trainy <- c(0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3)

testy <- c(0,1,2,3)

# One Hot Encoding

trainLabels <- to\_categorical(trainy)

testLabels <- to\_categorical(testy)

testLabels

# Model

model <- keras\_model\_sequential()

model %>%

  layer\_dense(units = 256, activation = 'relu', input\_shape = c(2352)) %>%

  layer\_dense(units = 128, activation = 'relu') %>%

  layer\_dense(units = 4, activation = 'softmax')

summary(model)

model %>%

  compile(loss = "binary\_crossentropy",

          optimizer = optimizer\_rmsprop(),

          metrics = c('accuracy'))

# Fit Model

history <- model %>%

  fit(trainx,

      trainLabels,

      epochs = 200,

      validation\_split = 0.2)

plot(history)

# Evaluation & Prediction - train data

k<-model %>% evaluate(trainx,trainLabels)

round(k$loss\*100)

round(k$acc\*100)

pred <- model %>% predict\_classes(trainx)

table(Predicted = pred, Actual = trainy)

prob <- model %>% predict\_proba(trainx)

p<-prob\*100

cbind(p, Predicted = pred, Actual= trainy,ifelse(pred==trainy,"T","F"))