**Faculty of Computer Science and Artificial Intelligence Helwan university‏**

**Selected 1**

Team number: **22**

**الاسم:** رحيم احمد موسي

**ID:** 202000318

**الاسم:** ايمن شوقي محمد

**ID:** 202000181

**الاسم:** حامد سامي حميدة

**ID:** 202000253

**الاسم:** يحيي طارق محمد

**ID:** 202001058

**الاسم:** بكر ابو حسيبة ذكي

**ID:** 202000206

**الاسم:** حازم احمد سعد حامد

**ID:** 202000243

**Project Description Document**

**A) General information on dataset**

**Name of dataset (Image):** Caltech-UCSD Birds-200 2011

**Number of classes and their labels:**

We used 20 classes from 200 class (+1000 image ) and the labels "Black\_Footed\_Albatross","Laysan\_Albatross","Sooty\_Albatross","Groove\_Billed\_Ani","Crested\_Auklet","Least\_Auklet","Parakeet\_Auklet","Rhinoceros\_Auklet","Brewer\_Blackbird","Red\_Winged\_Blackbird","Rusty\_Blackbird","Yellow\_Headed\_Blackbird","Bobolink","Indigo\_Bunting","Lazuli\_Bunting","Painted\_Bunting","Cardinal","Spotted\_Catbird","Gray\_Catbird","Yellow\_Breasted\_Chat “

**The total number of samples in the datasets and the size of each :**

The total number of samples is (+11000) and the size of each is 100x100

**The total number of samples used in training, validating and testing:**

Number of samples in training is 80% and validating and testing is 20%

**Name of dataset (Numerical):** Titanic

<https://www.kaggle.com/code/startupsci/titanic-data-science-solutions/data>

**Number of classes and their labels:**

We used 10 classes and the 1 label “survived’

**The total number of samples in the datasets and the size of each :**

The total number of samples is (1300+).

**The total number of samples used in training, validating and testing: 418**

Number of samples in training is 800+

**B) Implementing details**

**Caltech-UCSD Birds-200 2011 :**

We have several kind of birds and our system working on recognizing which bird’s Image is it.

Our Dataset has 2 files the first one was for The Raw Birds Image without any Feature Extraction while The second one was for The segmentation and a little bit of masking .

OUR WORK :

1. Import the Dataset (raw & segmentation) of image
2. Data preprocessing:

* Segmentation
* GrayScale
* Resize of image

1. Data splitting (using 2 pickle files the first one for images and the second one for labels)
2. Building and Training of Model using ANN Algorithm
3. Checking accuracy
4. Optimization of model
5. Plotting (loss curve, accuracy, confusion matrix, ROC curve)

* There is No cross validation used

FOR ANN MODEL :

loss='sparse\_categorical\_crossentropy', optimizer='adam', metrics=['accuracy']

,batch\_size = 20, epochs=10, verbose=1, validation\_split = .2, callbacks=[neptune\_cbk])

model = keras.Sequential()

model.add(Flatten(input\_shape=(10000, 1)))

model.add(Dense(1024))

model.add(Activation('relu'))

model.add(Dropout(0.3))

model.add(Dense(512))

model.add(Activation('relu'))

model.add(Dropout(0.3))

model.add(Dense(20))

model.add(Activation('softmax'))

FOR SVM:

lsvc = LinearSVC(C=1.0, class\_weight=None, dual=True, fit\_intercept=True,

          intercept\_scaling=1, loss='squared\_hinge', max\_iter=1000,

          multi\_class='ovr', penalty='l2', random\_state=10, tol=0.0001,

          verbose=1)

matrix = plot\_confusion\_matrix(lsvc, X\_test, y\_test,cmap=plt.cm.Blues,normalize='true')

**C) RESULT DETAILS:**

ANN:

VALIDATION EPOCH LOSS:

Chart, line chart

Description automatically generated

VALIDATION EPOCH ACCURACY :

Chart, line chart

Description automatically generated

TRAIN EPOCH LOSS:

Chart, line chart

Description automatically generated

TRAIN EPOCH ACCURACY:

Chart, line chart

Description automatically generated

**SVM:**

A picture containing table

Description automatically generated

**Titanic:**

Our Dataset has 3 csv files the first one was for the

1. gender\_submission which 2 columns (passengerID and survived)
2. training (11 CLASS)
3. testing and validation. (10 CLASS)
4. The Features extractions numbers: 7 (Pclass – Sex – Age – SibSp – Parch - Ticket - Cabin)
5. There is no cross validation used

FOR LOGESTIC REGRESSION:

logreg = LogisticRegression(max\_iter = 150, solver = 'liblinear', penalty = 'l1', C = 0.11)

FOR SVM:

clf= SVC(kernel='rbf' ,random\_state = 42, C=1.0, gamma='auto')

RESULT DETAILS (Numerical)

**Logistic Regression:**

**Accuracy:**



**classification report:**

Table

Description automatically generated with medium confidence

**confusion matrix:**

Text

Description automatically generated

**Specificity:**

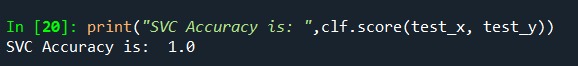


Chart, line chart

Description automatically generated**ROC Curve:**

**SVM:**

**Accuracy:**



**classification report:**

A picture containing calendar

Description automatically generated

**confusion matrix:**

Text

Description automatically generated

**Specificity:**



**Optimization of model:**

**Logistic Regression:**

Before optimization🡪 **96.65%**

**Text

Description automatically generated**

After optimization🡪 **97.13%**

**A picture containing calendar

Description automatically generated**