FOOTBALL SHOT CLASSIFIER

ANTONIO INVENINATO
ANGELO COCUZZA

APLICACIONES IOT 23/24

INDEX

- Football shot classification problem
- Hardware
- Dataset
- Neural Network 1D
- Contact

PROBLEM

Football shot classification problem

Target:

We want to use angular velocity and acceleration data to classify the different types of shots in football

HARDWARE

We use a Raspberry pi 4 + Sense Hat (includes acceleration and gyro sensors).
Power is supplied via PowerBank.

DATASET

Dataset:

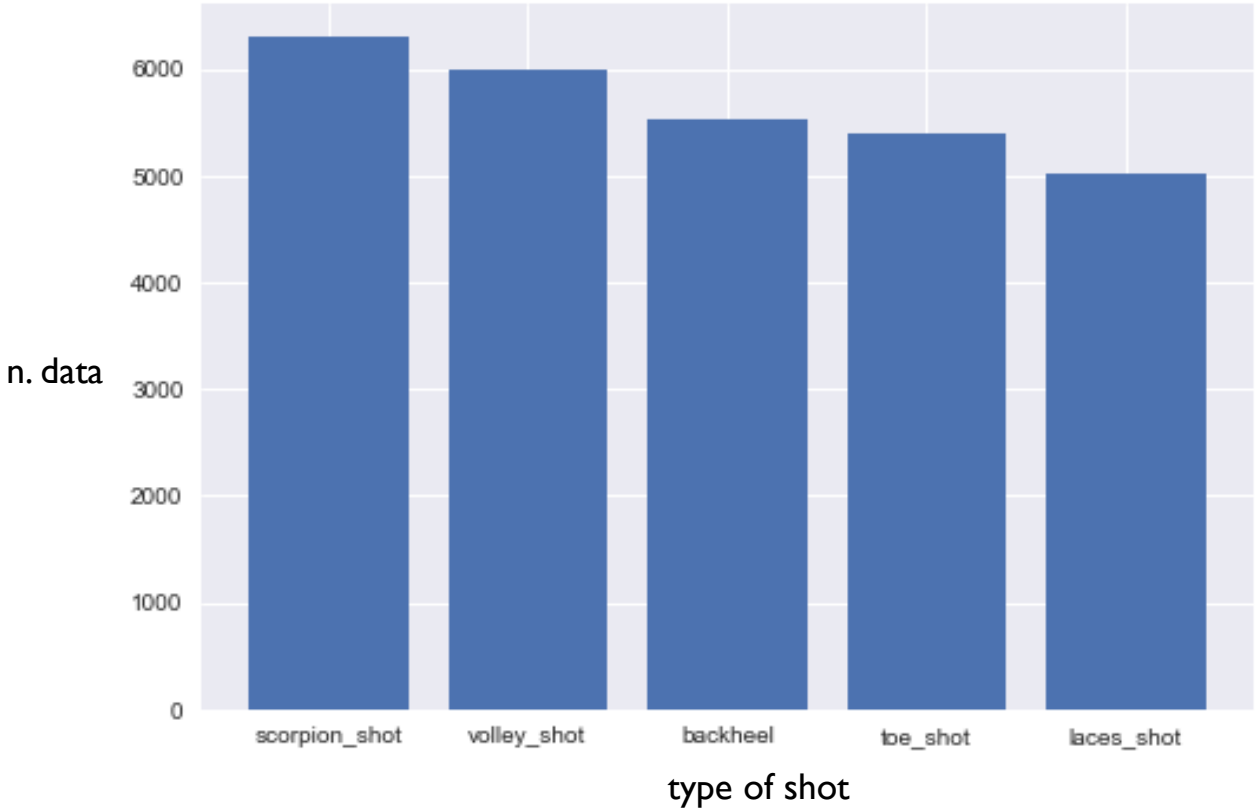
User ID	Sex	Foot	Age
1	M	Dx	23
2	M	Dx	24
3	F	Dx	21
4	M	Dx	22
5	M	Dx	23

Types of football shot:

1. Backheel
2. Laces shot
3. Toe shot
4. Volley shot
5. Scorpion shot

DATASET

Amount of data for each shot:



DATASET

Dataset file:

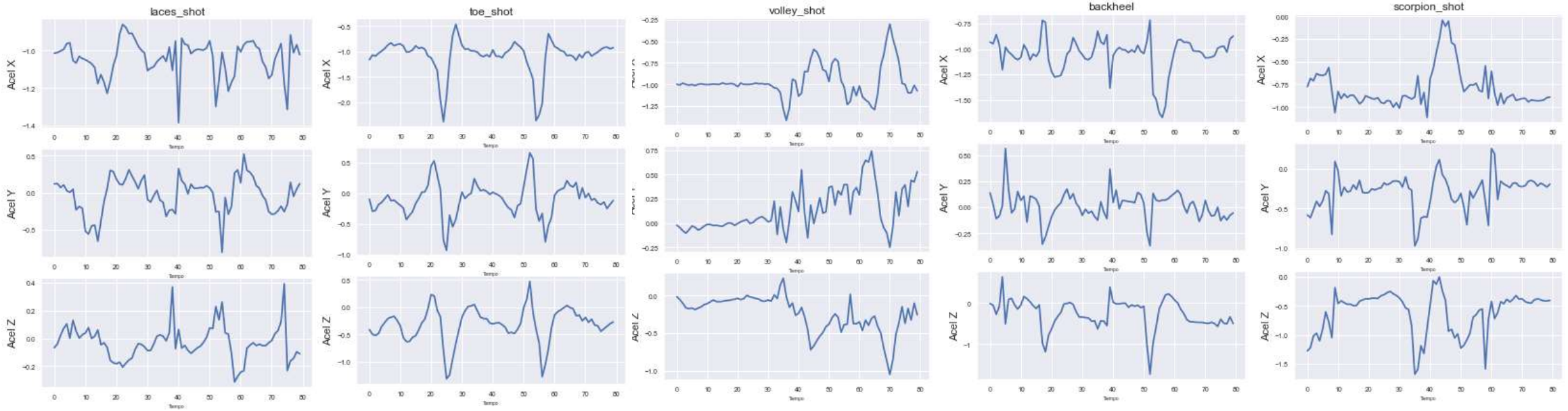
- football_dat.txt
- We have 28218 rows and 12 columns.
- 1200 rows correspond to a time window in which the shooting activity took place.
- 12 columns: 4 columns with players' data, 1 column with activity, 1 column with timestamp, 6 columns with acceleration and angular velocity data.

NEURAL NETWORK 1D

- Data Loading: Data from a CSV file is loaded into a Pandas DataFrame.
- Data Preprocessing: Data cleaning and preparation operations are performed.
- Data visualization: displays the distribution of activities in the dataset and creates graphs to show the acceleration and gyroscope data for each activity.

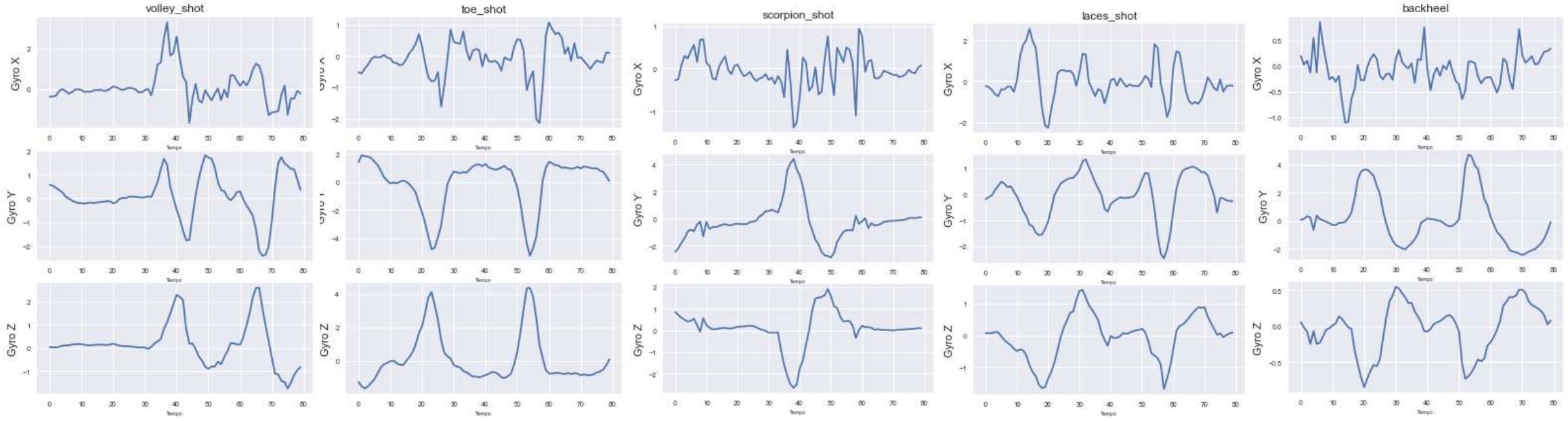
NEURAL NETWORK 1D

Acceleration:



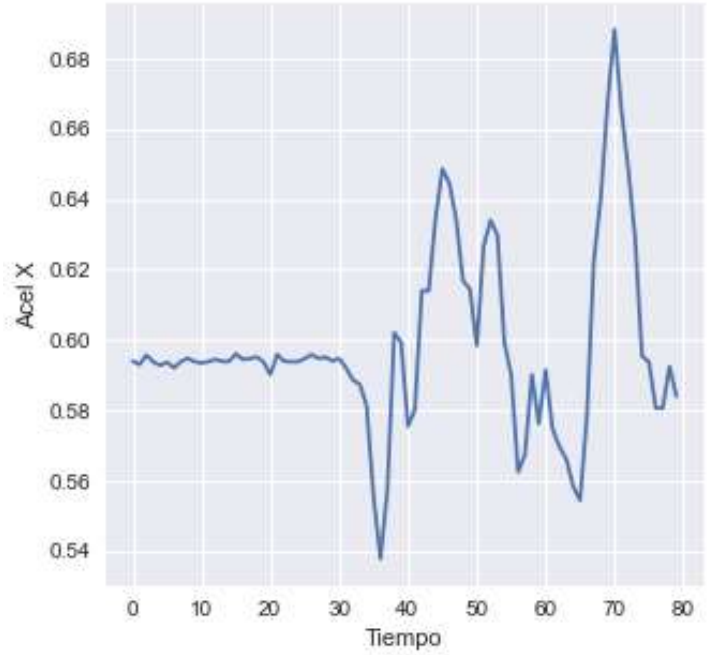
NEURAL NETWORK 1D

Angular velocity:



NEURAL NETWORK 1D

- Data normalization: for each column of interest (e.g. 'x-axis'), the minimum column value is subtracted from each value, and the result is divided by the difference between the maximum and minimum column value. This normalization process ensures that all values fall within the range [0, 1].



Estandarización	Normalización
$x_{stand} = \frac{x - mean(x)}{Standard\ Dev(x)}$	$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)}$

NEURAL NETWORK 1D

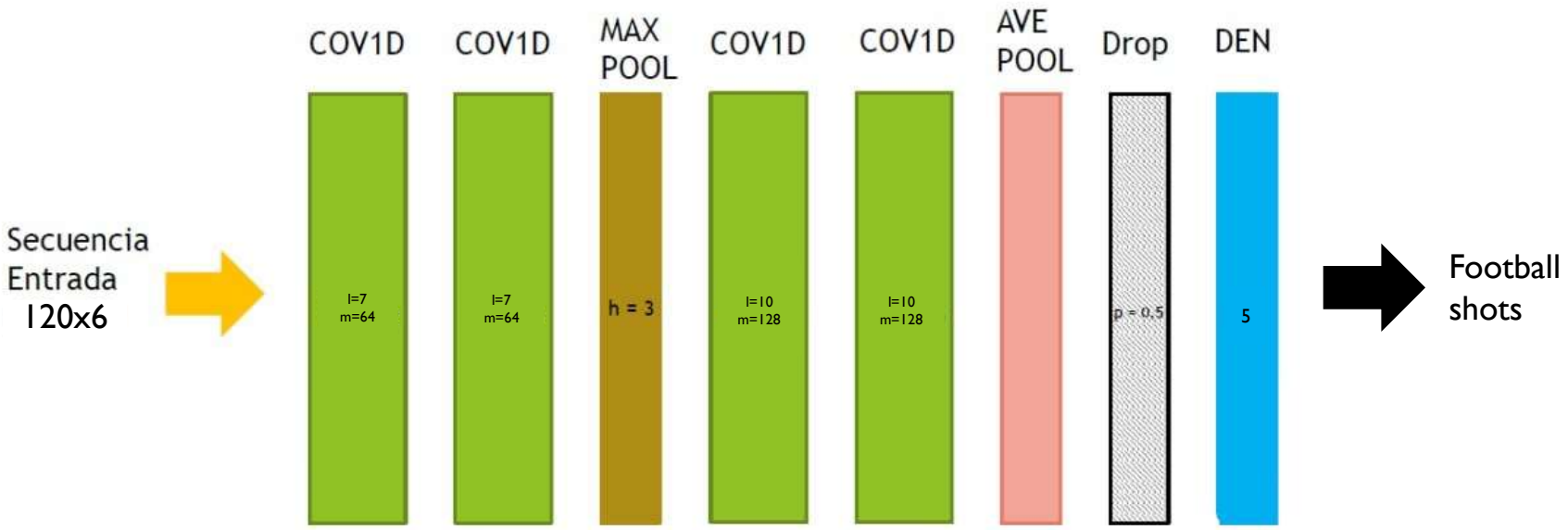
- Data division: data sequences and their labels are created for training and testing. The labels are also converted into one-hot encoding format.

```
memory usage: 2.6+ MB
None
(28218, 12)
```

```
Entrenamiento (22252, 13)
Test (5965, 13)
Entrenamiento 0.7886026154445901
Test 0.2113973845554098
```

NEURAL NETWORK 1D

- Neural network definition: A convolutional neural network (CNN) with Conv1D and Dense layers is used. The network is designed to learn relevant features from time sequences of acceleration and gyroscope data.



NEURAL NETWORK 1D

Layer (type)	Output Shape	Param #
conv1d_117 (Conv1D)	(None, 114, 64)	2752
conv1d_118 (Conv1D)	(None, 108, 64)	28736
max_pooling1d_29 (MaxPooling1D)	(None, 36, 64)	0
conv1d_119 (Conv1D)	(None, 27, 128)	82048
conv1d_120 (Conv1D)	(None, 18, 128)	163968
global_average_pooling1d_30 (GlobalAveragePooling1D)	(None, 128)	0
dropout_30 (Dropout)	(None, 128)	0
dense_30 (Dense)	(None, 5)	645

Total params: 278149 (1.06 MB)
Trainable params: 278149 (1.06 MB)
Non-trainable params: 0 (0.00 Byte)

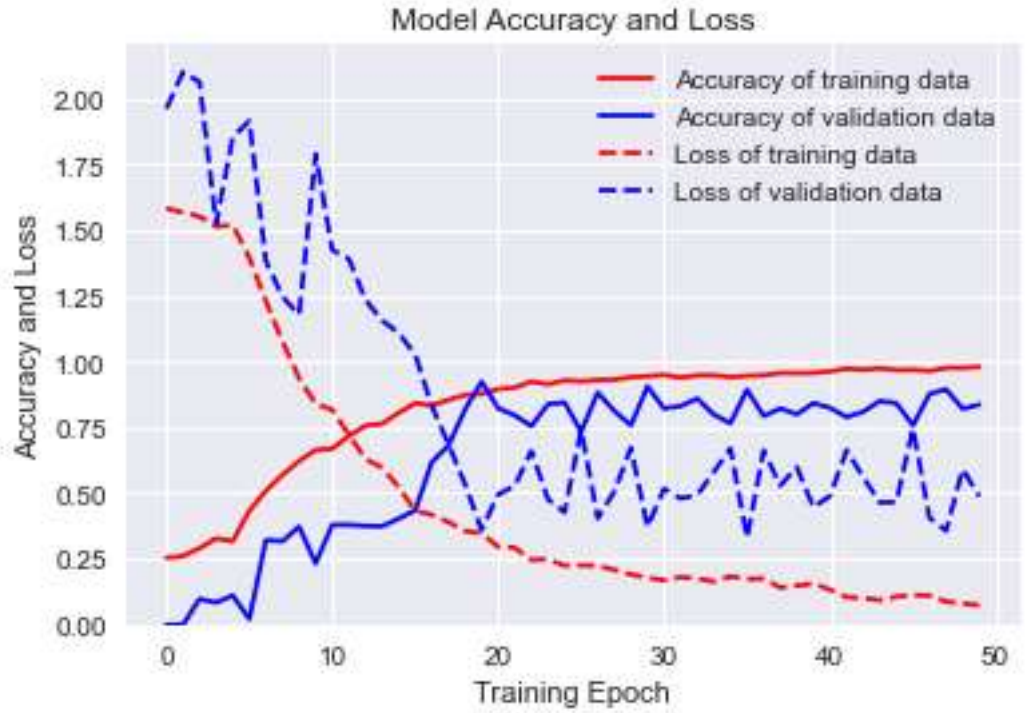
NEURAL NETWORK 1D

- Training the model: The model is trained using the training data. The optimiser 'adam' and the loss function 'categorical_crossentropy' are used. ModelCheckpoint is also used to save the best model during training.
- Evaluation of the model on test data: The trained model is evaluated using test data. Accuracy and loss metrics are displayed, and a confusion matrix is created to evaluate the model's performance in classifying football shots.

NEURAL NETWORK 1D

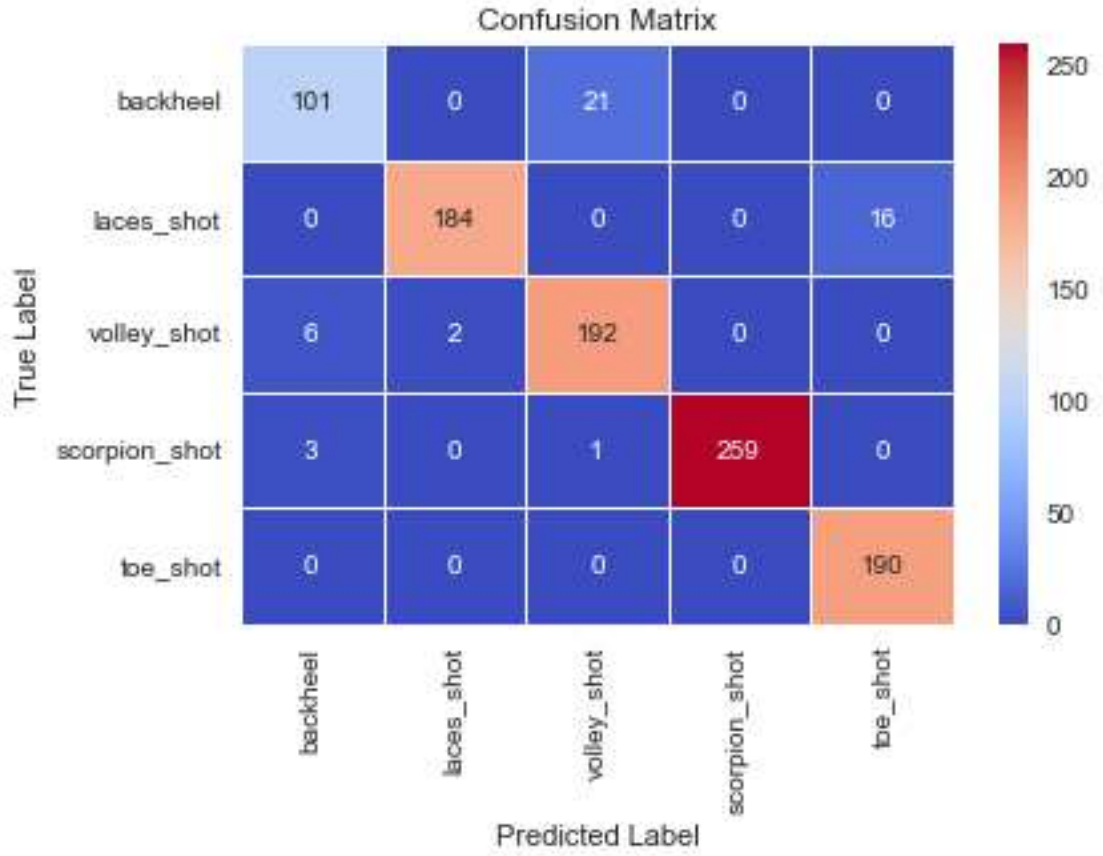
Test results:

```
Test accuracy 0.9497435688972473
Test loss 0.1447836011648178
```



NEURAL NETWORK 1D

Confusion Matrix:



CONTACT

Antonio Inveninato: antonioinv1@hotmail.it

Angelo Cocuzza: cczngl00a03c342s@studium.unict.it

GitHub:
https://github.com/Antonio3005/FootballShotClassifier

THANK YOU
ANY DOUBTS?