SHEEP ACTIVITY RECOGNITION

CENTRE FOR TECHNOLOGY INFUSION

TABLE OF CONTENT:

1. Aim----------------------------------------------------------------------------------------------- 3
2. Data Available---------------------------------------------------------------------------------- 3
3. Process of Training----------------------------------------------------------------------------3

## Adding Feature---------------------------------------------------------------------------- 3

## Separating the features with grouping the activities and rolling them------- 4

## Concatenating the separated features----------------------------------------------- 4

## Joining the seven arrays and reshaping them-------------------------------------- 4

## Label Encoding----------------------------------------------------------------------------- 4

## Building and fitting the model---------------------------------------------------------- 4

1. Testing the Model----------------------------------------------------------------------------- 5
   1. Adding features--------------------------------------------------------------------------- 5

## Label Encoding---------------------------------------------------------------------------- 5

## Test data pre-processing---------------------------------------------------------------- 5

4.4 Prediction---------------------------------------------------------------------------------- 5

# Aim :

The ultimate aim is to build a 1D Convolutional Neural Network model for recognising the sheep activity with the help of the input data.

Two different python files are created. The first one is made for pre-processing, extracting new features with the available features of the training data, building a 1D Convolutional Neural Network and training them. The next one is for pre-processing, feature extracting and reshaping the test data to predict the activities with the help of trained model.

# Data available:

The data which is available for training and test have four columns:

**Axis1**- Which is the x axis reading of the accelerometer

**Axis2**- Which is the x axis reading of the accelerometer

**Axis3**- Which is the x axis reading of the accelerometer

**label\_name**- It has the class labels (Idle, Walk, rumination, grazing, Licking). Label name is the target variable in the model.

# Process of Training:

## Adding Feature:

The process of training is to make the machine learn the features for the prediction of the class labels.

So far the available features are the general ones Axis1, Axis2, Axis3. Only with these general features the possibility of developing a model with greater accuracy is low. So, it was decided to add few more features which denotes the relationship between the Axis1,Axis2 and Axis3. They are:

**Magnitude Vector** : Square root of Axis1\*Axis2\*Axis3

**X\_Y** : Axis1\*Axis2(Axis1 and Axis2 relation)

**Y\_Z** : Axis2\*Axis3(Axis2 and Axis3 relation)

**X\_Z** : Aixs1\*Axis3 (Axis1 and Axis3 relation)

To do this function add feature **add\_feature(data)** function is used.

**add\_feature(data):**

This function can be used to add more features for the data. So far four features have been added which are Magnitude\_vector, relation between Axis1 and Axis2, relation between Axis2 and Axis3, relation between Axis1 and Axis3.

It has only one input parameter which is the whole raw data.

## Separating the features with grouping the activities and rolling them:

After the extraction of necessary features , each feature column is filtered based upon each activity and copied as different data series like Idle\_x, Idle\_y , Walk\_y,etc.

Example-Idle\_x is the Axis1 data of Idle activity.

This is done for rolling them with a preferred window size in order to roll the information.

This activity is done by the function **seperating\_features(data,activity,roll\_size).**This function can be used to separate each features as series based upon each activity and to roll them with a preferred window size.It has three parameters which are the data obtained from add\_feature() function, the activity type and the window size.

**roll\_ts(series,window)** function is used to roll the data series with preferred window size.

## Concatenating the separated features:

After the rolling is done the rolled feature arrays are concatenated with grouping the features and seven array are obtained which are x1,x2,x3,x4,x5,x6,x7. (Each for each feature)

For example x1 is the rolled Axis1 data for all the activities This work is done by the function **concat(Idle,Walk,grazing,licking,rumination).**This function is used for concatenating the rolling results obtained from separating feature() function. The four input parameters are the rolled data of each activity of same feature.

## Joining the seven arrays and reshaping them:

All the seven arrays are joined and reshaped as final data in such a way that the machine could learn them as different features. These are done with reshape() pre-defined function.

## Label Encoding:

The class labels which are categorical are encoded as numerical data with the help of LabelEncoder() predefined function. The encoding pattern is saved as a pickle file(encoder.pkl) which can be used to encode the label while testing.

## Building and fitting the model:

After the above steps are done the Convolutional Neural Network is built according to the input\_shape.The inpust\_shape here is (30,7).

This development of model is done by **Convolution\_Model(time\_steps,features)** function. This function is used for designing the Convolutional Neural Network required for the activity recognition. The input parameters are the time steps and the no of feature which is the input shape.

After developing the model, it is fitted and trained with the final data.

# Testing the model:

A separate dataset which is similar but not same to the Training data is kept for testing.

The columns are similar to that of the training data.

* 1. Adding features:

The features which are extracted for training the model are extracted for testing the model as well. The process is same as the feature extraction in training data.

## Label Encoding:

Label encoding is done for test data by loading the encoder.pkl file in order to encode the test dataset same as the train dataset.

## Test data pre-processing:

After extracting the features the test data is pre-processed in such a way that the machine accept them as input. Here unlike the training process the data is not rolled.

Here the process carried out is, as the data is of the frequency 30Hz, each 30 rows of data are taken as a single input in each feature. By doing this the input shape will be same as the shape which is used to build the model.

This work is done with the help of function **test\_preprocess(time\_steps,step).**This function is used to pre-process the test data in a way which the CNN model receive. The two input parameters time\_steps and step both denotes the frequency of the data. For example: In the Sheep project the data is recorded with the frequency of 30Hz. So the value of time\_steps and step will be 30.

## Prediction:

After pre-processing the data the test data is given as input to the model. The result obtained after testing the model is compared with the actual result with the help of Confusion matrix. Confusion matrix is built with the help of function **show\_confusion\_matrix(validations, predictions ,LABELS).** This function is for plotting the confusion matrix after testing the data with the model. The input parameter validation is the actual class label, predictions is the class label predicted by the model and the LABELS is the equivalent categorical value of each numerical label value.