Question 1:

Movement analysis & prediction based on user action : from image and/or video captured on a mobile. The purpose is to track key points in the body as they move through space and analyse and predict which action was attempted by the user. For example, a tennis shot from a video. Suggested tools : OpenCV, Mediapipe, YOLO, etc.

Solution:

Dataset

For the above problem I went for selecting the cricket as sport. The dataset I got from the ascuet/CricShot10 which is a video action recognition dataset consisting of 10 cricket batting shots. This data was collected through the authorization of the author. For this assignment I have used 4 classes. The dataset was split for train, val and test split as 70%, 15% and 15% respectively.

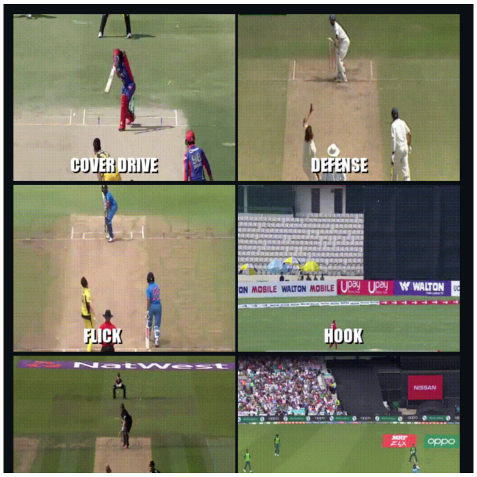


Fig. 1: Sample dataset

Preparation:

For each video all the frames were extracted and the further basic preprocessing is applied for the training, validation and test dataset.

For all these frames features are being extracted using CLIP-based (Fig-2) model and then they are stored. These significantly reduces the computational cost as the features are not being calculated multiple times.

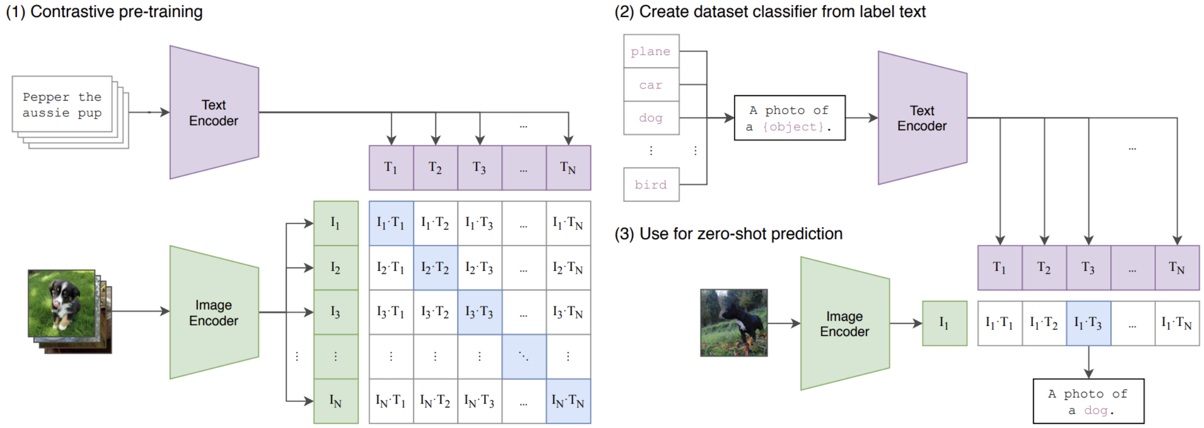

(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ⋮ I_N
$I_1{\cdot}T_1$ $I_1{\cdot}T_2$ $I_1{\cdot}T_3$ ... $I_1{\cdot}T_N$
$I_2{\cdot}T_1$ $I_2{\cdot}T_2$ $I_2{\cdot}T_3$ ... $I_2{\cdot}T_N$
$I_3{\cdot}T_1$ $I_3{\cdot}T_2$ $I_3{\cdot}T_3$ ... $I_3{\cdot}T_N$
$I_N{\cdot}T_1$ $I_N{\cdot}T_2$ $I_N{\cdot}T_3$ ... $I_N{\cdot}T_N$
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
I_1
$I_1{\cdot}T_1$ $I_1{\cdot}T_2$ $I_1{\cdot}T_3$ ... $I_1{\cdot}T_N$
A photo of a dog.

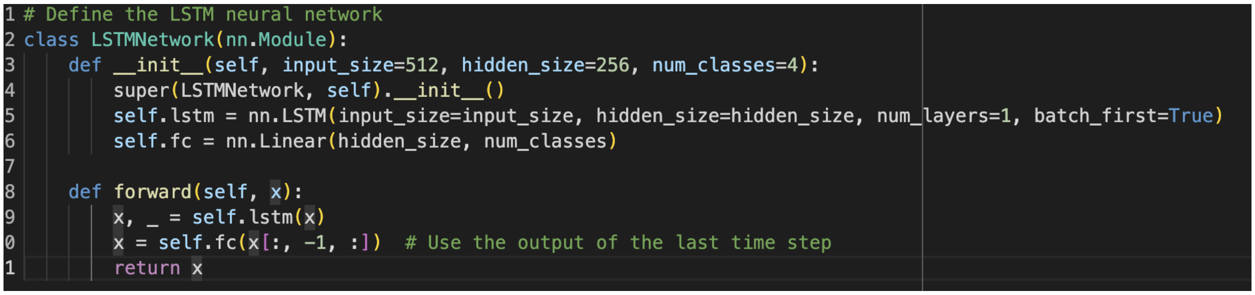
Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

Fig. 1: CLIP Model

Training:

Model

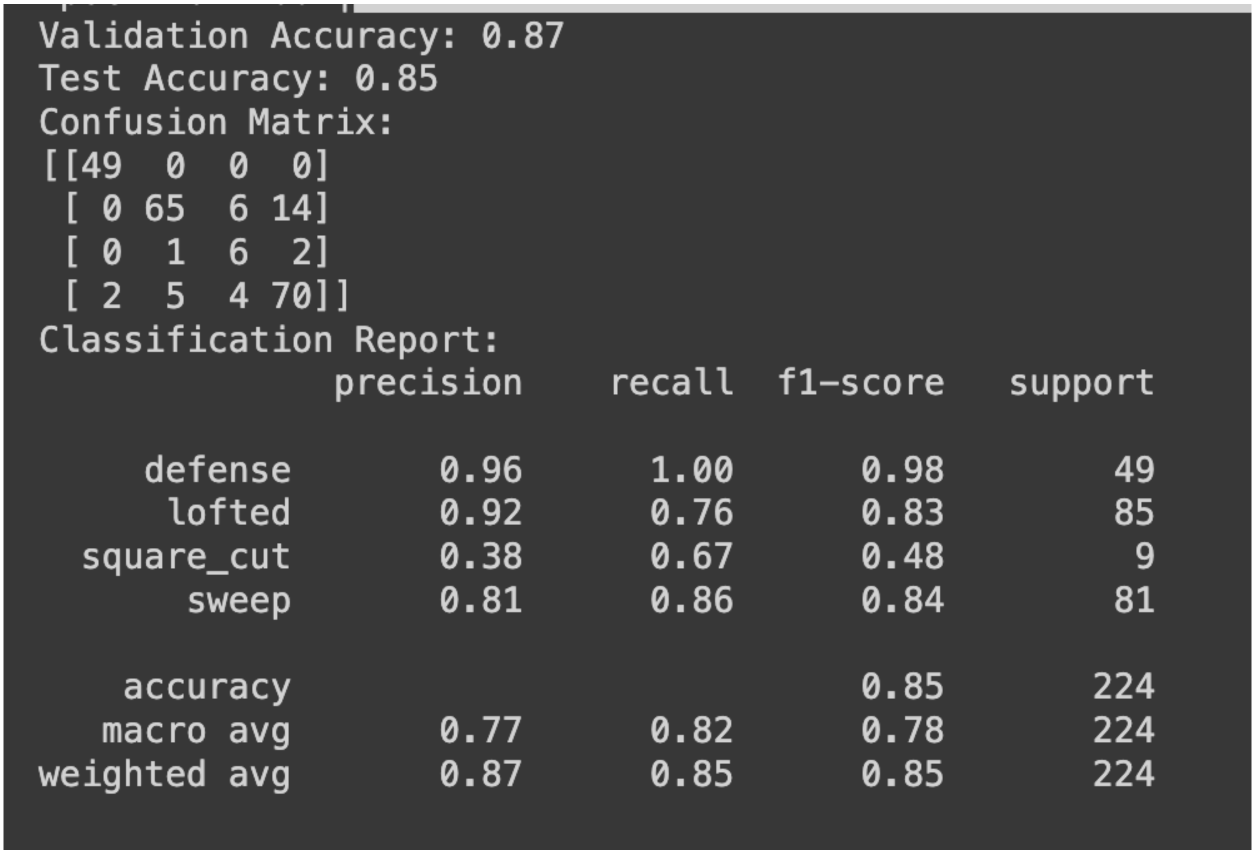
As the video are sequential data we are using LSTM for our purpose. This helps to get the temporal dependencies which helps for better decision making. For this problem only the last hidden is being taken into account for making the decision using classification layer. There could be other method taking all the hidden state but I took it as there is almost all the information is present in this state which help for prediction.



```
# Define the LSTM neural network
class LSTMNetwork(nn.Module):
    def __init__(self, input_size=512, hidden_size=256, num_classes=4):
        super(LSTMNetwork, self).__init__()
        self.lstm = nn.LSTM(input_size=input_size, hidden_size=hidden_size, num_layers=1, batch_first=True)
        self.fc = nn.Linear(hidden_size, num_classes)

    def forward(self, x):
        x, _ = self.lstm(x)
        x = self.fc(x[:, -1, :])  # Use the output of the last time step
        return x
```

I have trained the model for 20 epoch and learning rate of 0.001 using adam optimizer. For the above I got the **test accuracy** of the **85%.**



```
Validation Accuracy: 0.87
Test Accuracy: 0.85
Confusion Matrix:
[[49  0  0  0]
 [ 0 65  6 14]
 [ 0  1  6  2]
 [ 2  5  4 70]]
Classification Report:
              precision    recall  f1-score   support

     defense       0.96      1.00      0.98        49
      lofted       0.92      0.76      0.83        85
  square_cut       0.38      0.67      0.48         9
       sweep       0.81      0.86      0.84        81

    accuracy                           0.85       224
   macro avg       0.77      0.82      0.78       224
weighted avg       0.87      0.85      0.85       224
```