CSE 515

Multimedia and Web Databases Group 6

Team

1. Aryan Patel (apate294@asu.edu)

2. Tejas Ajay Parse (tparse@asu.edu)

3. Om Patel (opatel14@asu.edu)

4. Tanishque Zaware (tzaware@asu.edu)

Abstract

This project involves extraction of features from human motion dataset and identifying similar motions for a given input. We have used Histogram of Optical Flow (HoF) and Histogram of Oriented Gradients (HoG) to create feature vectors and saved the representatives for further use. Different models have been generated based on different problem statements. Each of these models extracts feature descriptors for a video in the dataset and stores it. These features include a 512 dimensional tensor, Histogram of Oriented Gradients(HOG) bag of vectors, Histogram of Optical Flow(HOF) bag of vectors and Color Histogram. The feature descriptors are then used to find m closest videos to the given video using distance metrics like cosine and Euclidean distance

Keywords

Histogram of Optical Flow (HoF), Feature Extraction, Histogram of Oriented Gradients (HoG), OpenCV, R3D18

Introduction

This report is for the phase 1 of our project in CSE 515 - Multimedia and Web Databases. This phase involves working with the HMDB51 dataset[2], which contains 51 distinct action categories, to extract feature descriptors for the video files and finding 'm' most similar videos , given a video file query. For this phase, we move the action folders of 'cartwheel', 'drink', 'ride_bike', 'sword', 'sword_exercise', 'wave' into a separate folder named 'Target_videos' and the rest of the 45 folders go to 'Non_target_videos'. There are a total of 5 tasks in this phase, where the first three tasks deal with different ways to extract feature descriptors from the given video file. Task 4 stores all these descriptors which are then used by task 5 to compare a given video file query with all the videos in Target_videos to find 'm' most similar videos to the given video query.

More Specifically, Task 1 involves working with R3D18 model [3] present in python's pytorch framework. Forward hooks are attached to three layers of R3D18 model, namely "layer3", "layer4", and "avgpool" and get the outputs of these 3 layers. The outputs are then converted into a 512 dimensional tensor.

In Task 2, we aim to extract meaningful spatiotemporal features from videos using Spatiotemporal Interest Points (STIPs) and generate compact representations for each video based on HoG (appearance) and HoF (motion) features. These representations are then used to compare and find similar videos.

Task 3 is based on working with the color models of the videos in the dataset. Three specific frames from each video are extracted and color histograms representing red, green and blue channel intensities of frame are created. The histograms are stored to be used as a feature for the video.

In Task 4, the feature descriptors obtained from tasks 1, 2 and 3 are stored for all of the videos present inside Target_videos folder. Task 5 then uses the stored features to find similar videos to the given video and any of the visual models.

Proposed Solution

In Task 1, the video files are read frame by frame using the OpenCV library. Each frame is resized to height and width of 112. The color channels are also converted from BGR, which is the default reading format of OpenCV, to RGB. Based on the input model_name, a forward hook is attached to one of the three layers of R3D18. For layer 3 , the output tensor is of shape 256 x n x 14 x 14, where n is the depth of the input or simply speaking, the number of frames in the video. This tensor is averaged on its spatial features first (14 x 14) and is then flattened to obtain a tensor with 256*n features. A linear transformation is then applied to convert it into a 512 dimensional tensor. The linear layer projects the 256*n features into a 512 dimensional space. For layer 4, which gives an output of shape 512 x n x 7 x 7 , the tensor is directly averaged on both its spatial (7 x 7) as well as temporal features (n) to obtain a 512 dimensional tensor. The output of the avgpool layer is already a 512 dimensional tensor so no further transformation is needed for it. Given a video file and a model name (either layer3, layer4 or avgpool) task 1 returns a 512 dimensional tensor of the corresponding layer.

Task 2 begins by selecting the top 400 highest-confidence STIPs from each non-target video. From these, we sample 10,000 STIPs for each combination of spatial scale (sigma2) and temporal scale (tau2) pairs. Using k-means clustering, we generate 40 cluster representatives for both HoG and HoF features for each of the 12 ⟨sigma2, tau2⟩ combinations. For a given video, the 400 highest-confidence STIPs are assigned to the closest cluster representatives. This forms 12 histograms, one for each ⟨sigma2, tau2⟩ pair, capturing how often specific patterns appear in the video. These histograms are concatenated to create 480-dimensional HoG and HoF bag-of-features (BoF) vectors, which serve as feature descriptors. These descriptors allow for comparison between videos based on their visual and motion characteristics.

Task 3 deals with extracting features of the video dataset in the form of color histograms. OpenCV is used to capture the video file and count the total frames in the video. Hereafter the first, middle, and last frame of the video are stored as images in jpg format. The dimensions of the frames are calculated, and the frame is divided into r x r cells where r is equal to 4 and it represents the resolution input. For each cell, a histogram of 12 bins is calculated for red, green, and blue color channels indicating the intensity of pixels in terms of the 3 color channels. Histograms of each cell are then concatenated to get a total of 576 values per frame and are stored as a frame feature vector. Finally, the frame feature vectors of all three frames are concatenated into 1 video feature vector having 1728 values and hence represents the feature value of color histogram for that particular video. The frames and concatenated histograms of each are visualized.

In Task 4, we first define functions of the previous tasks (1,2,3). We then walk through each video file in the Target_videos folder and call the task functions on the video file . The output of the function is then stored at "Outputs/Task[No.]". For example, since task 1 has 3 different models, the function is called for all 3 models for the same video file and the outputs are stored under "Outputs/Task1". The same is applied for tasks 2 and 3 as well.

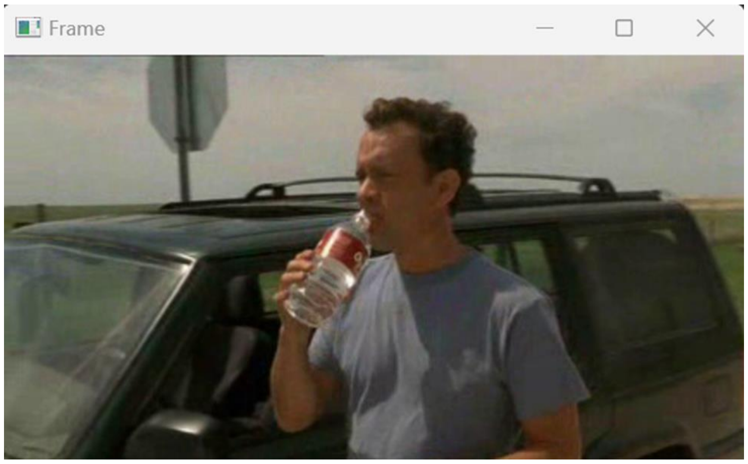
The final task is then to use the features of different models stored through task 4 to perform a near neighbor search to find m most similar videos from the target_videos folder. The comparison for the tasks 1, 2 and 3 is done using cosine distance since for higher dimensions, cosine distance tends to work well. Also, since task1 encodes relational information where direction is more important than magnitude, cosine distance is better.

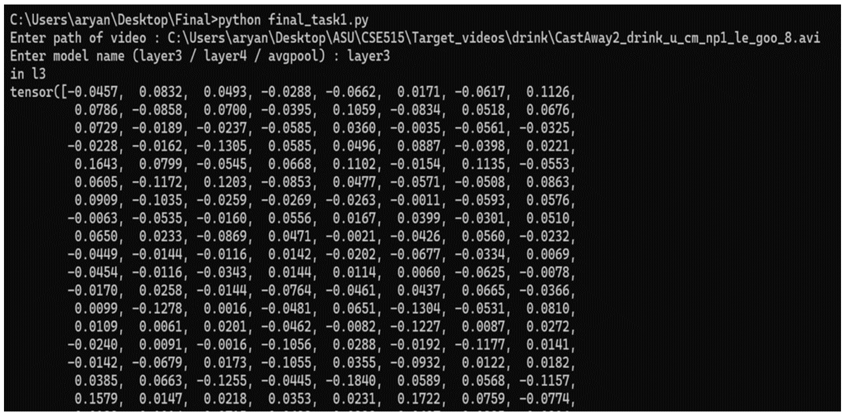
Outputs

Task 1

Task 1 has a simple execution structure. When task 1 is executed, it will first ask for the path of the video. You can provide relative or absolute path but relative path will only work if you are in the correct working directory. For the following outputs of task 1, I have used absolute paths. It will next ask for which model to use. You can input either layer3, layer4, or avgpool. Based on these inputs, first the video will be played and then the output tensor of 512 dimensions will be printed. Note : The tensors printed below are not complete tensors since their screenshot takes up the whole page.

For : "drink\CastAway2_drink_u_cm_np1_le_goo_8.avi"

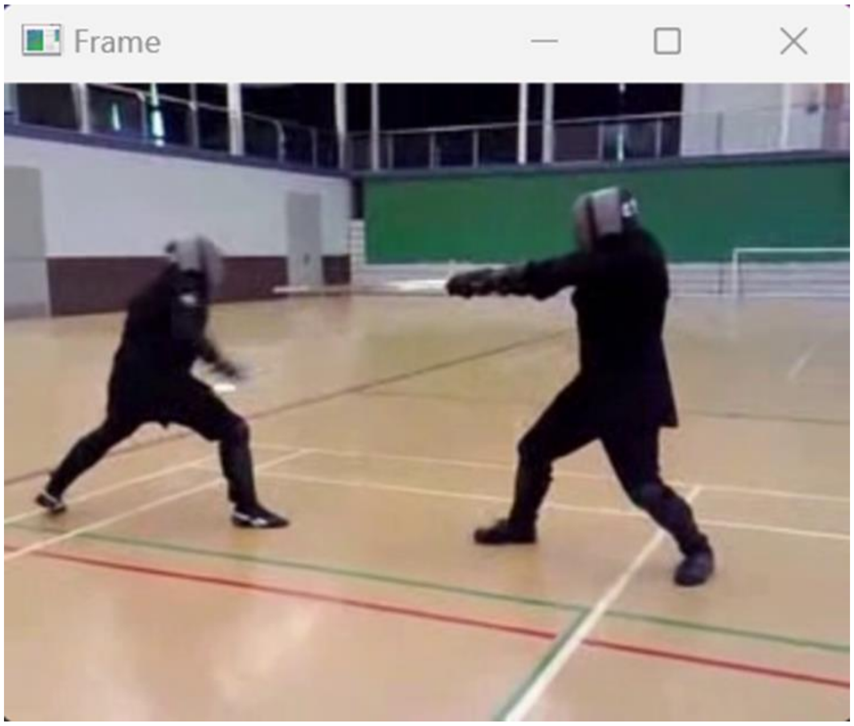


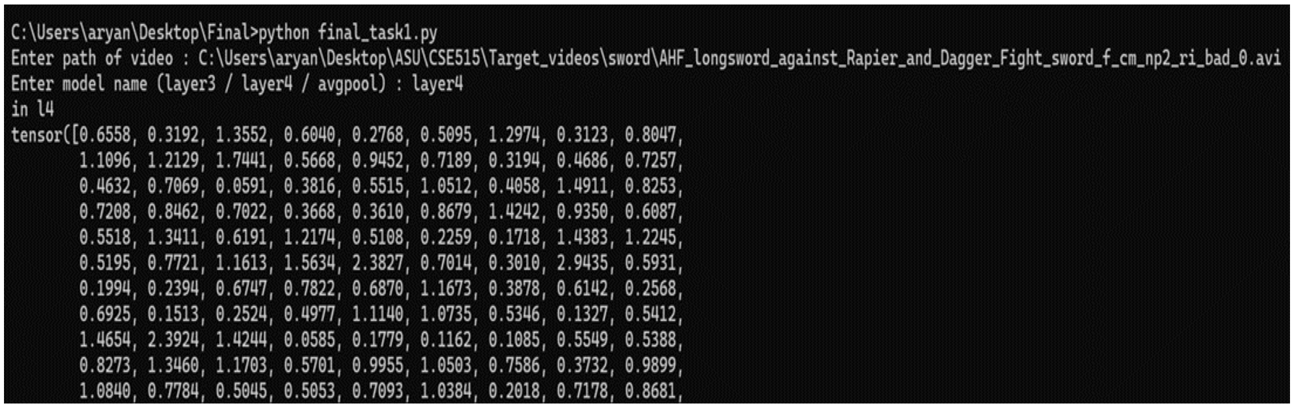


```
C:\Users\aryan\Desktop\Final>python final_task1.py
Enter path of video : C:\Users\aryan\Desktop\ASU\CSE515\Target_videos\drink\CastAway2_drink_u_cm_np1_le_goo_8.avi
Enter model name (layer3 / layer4 / avgpool) : layer3
in l3
tensor([-0.0457,  0.0832,  0.0493, -0.0288, -0.0662,  0.0171, -0.0617,  0.1126,
         0.0786, -0.0858,  0.0700, -0.0395,  0.1059, -0.0834,  0.0518,  0.0676,
         0.0729, -0.0189, -0.0237, -0.0585,  0.0360, -0.0035, -0.0561, -0.0325,
        -0.0228, -0.0162, -0.1305,  0.0585,  0.0496,  0.0887, -0.0398,  0.0221,
         0.1643,  0.0799, -0.0545,  0.0668,  0.1102, -0.0154,  0.1135, -0.0553,
         0.0605, -0.1172,  0.1203, -0.0853,  0.0477, -0.0571, -0.0508,  0.0863,
         0.0909, -0.1035, -0.0259, -0.0269, -0.0263, -0.0011, -0.0593,  0.0576,
        -0.0063, -0.0535, -0.0160,  0.0556,  0.0167,  0.0399, -0.0301,  0.0510,
         0.0650,  0.0233, -0.0869,  0.0471, -0.0021, -0.0426,  0.0560, -0.0232,
        -0.0449, -0.0144, -0.0116,  0.0142, -0.0202, -0.0677, -0.0334,  0.0069,
        -0.0454, -0.0116, -0.0343,  0.0144,  0.0114,  0.0060, -0.0625, -0.0078,
        -0.0170,  0.0258, -0.0144, -0.0764, -0.0461,  0.0437,  0.0665, -0.0366,
         0.0099, -0.1278,  0.0016, -0.0481,  0.0651, -0.1304, -0.0531,  0.0810,
         0.0109,  0.0061,  0.0201, -0.0462, -0.0082, -0.1227,  0.0087,  0.0272,
        -0.0240,  0.0091, -0.0016, -0.1056,  0.0288, -0.0192, -0.1177,  0.0141,
        -0.0142, -0.0679,  0.0173, -0.1055,  0.0355, -0.0932,  0.0122,  0.0182,
         0.0385,  0.0663, -0.1255, -0.0445, -0.1840,  0.0589,  0.0568, -0.1157,
         0.1579,  0.0147,  0.0218,  0.0353,  0.0231,  0.1722,  0.0759, -0.0774,
```

For :

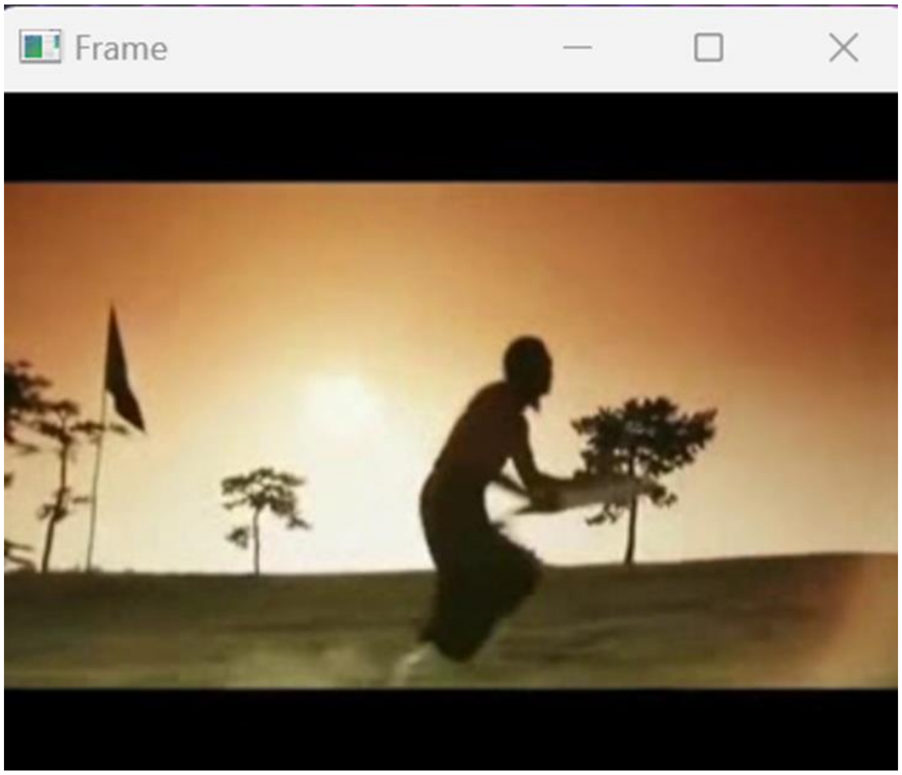
sword\AHF_longsword_against_Rapier_and_Dagger_Fight_sword_f_cm_np2_ri_bad_0.avi

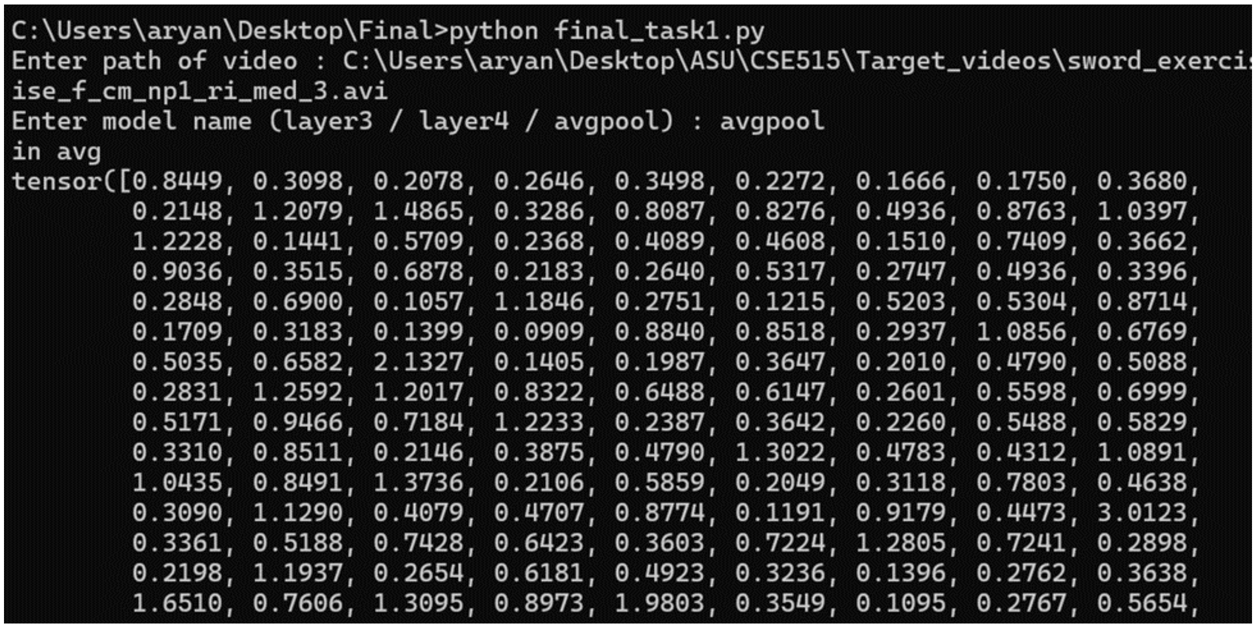




```
C:\Users\aryan\Desktop\Final>python final_task1.py
Enter path of video : C:\Users\aryan\Desktop\ASU\CSE515\Target_videos\sword\AHF_longsword_against_Rapier_and_Dagger_Fight_sword_f_cm_np2_ri_bad_0.avi
Enter model name (layer3 / layer4 / avgpool) : layer4
in l4
tensor([0.6558, 0.3192, 1.3552, 0.6040, 0.2768, 0.5095, 1.2974, 0.3123, 0.8047,
        1.1096, 1.2129, 1.7441, 0.5668, 0.9452, 0.7189, 0.3194, 0.4686, 0.7257,
        0.4632, 0.7069, 0.0591, 0.3816, 0.5515, 1.0512, 0.4058, 1.4911, 0.8253,
        0.7208, 0.8462, 0.7022, 0.3668, 0.3610, 0.8679, 1.4242, 0.9350, 0.6087,
        0.5518, 1.3411, 0.6191, 1.2174, 0.5108, 0.2259, 0.1718, 1.4383, 1.2245,
        0.5195, 0.7721, 1.1613, 1.5634, 2.3827, 0.7014, 0.3010, 2.9435, 0.5931,
        0.1994, 0.2394, 0.6747, 0.7822, 0.6870, 1.1673, 0.3878, 0.6142, 0.2568,
        0.6925, 0.1513, 0.2524, 0.4977, 1.1140, 1.0735, 0.5346, 0.1327, 0.5412,
        1.4654, 2.3924, 1.4244, 0.0585, 0.1779, 0.1162, 0.1085, 0.5549, 0.5388,
        0.8273, 1.3460, 1.1703, 0.5701, 0.9955, 1.0503, 0.7586, 0.3732, 0.9899,
        1.0840, 0.7784, 0.5045, 0.5053, 0.7093, 1.0384, 0.2018, 0.7178, 0.8681,
```

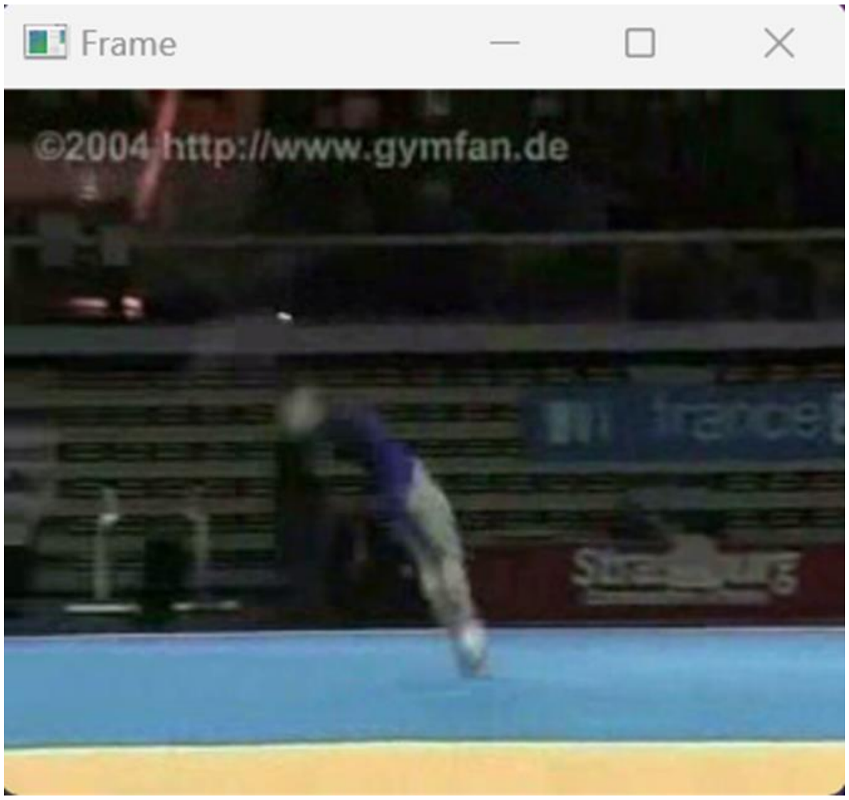
For : sword_exercise\Blade_Of_Fury_-_Scene_1_sword_exercise_f_cm_np1_ri_med_3.avi

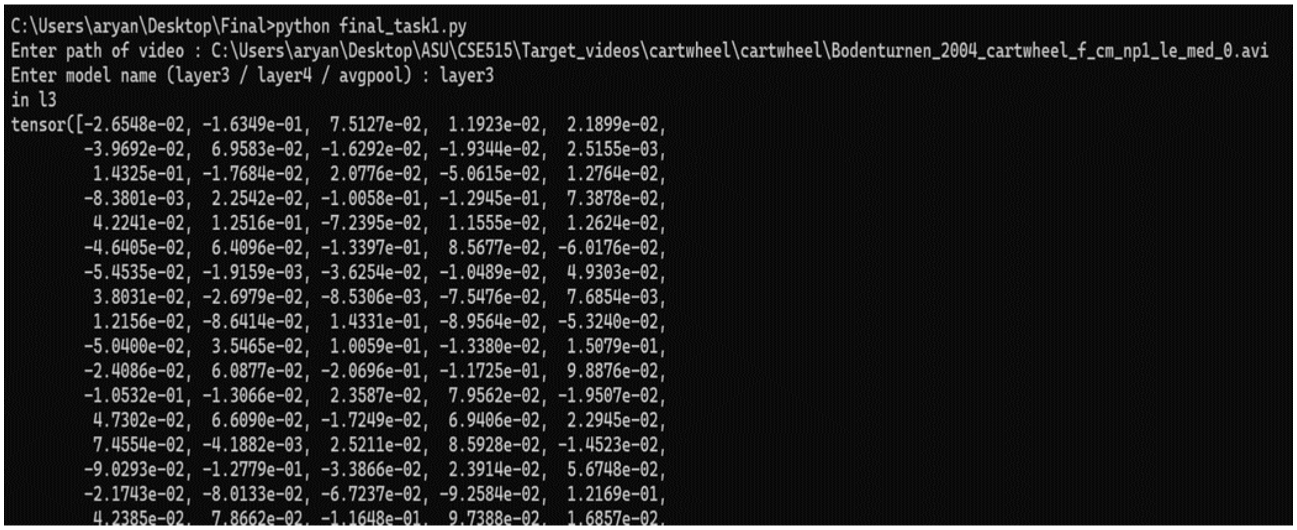




```
C:\Users\aryan\Desktop\Final>python final_task1.py
Enter path of video : C:\Users\aryan\Desktop\ASU\CSE515\Target_videos\sword_exerci
ise_f_cm_np1_ri_med_3.avi
Enter model name (layer3 / layer4 / avgpool) : avgpool
in avg
tensor([0.8449, 0.3098, 0.2078, 0.2646, 0.3498, 0.2272, 0.1666, 0.1750, 0.3680,
        0.2148, 1.2079, 1.4865, 0.3286, 0.8087, 0.8276, 0.4936, 0.8763, 1.0397,
        1.2228, 0.1441, 0.5709, 0.2368, 0.4089, 0.4608, 0.1510, 0.7409, 0.3662,
        0.9036, 0.3515, 0.6878, 0.2183, 0.2640, 0.5317, 0.2747, 0.4936, 0.3396,
        0.2848, 0.6900, 0.1057, 1.1846, 0.2751, 0.1215, 0.5203, 0.5304, 0.8714,
        0.1709, 0.3183, 0.1399, 0.0909, 0.8840, 0.8518, 0.2937, 1.0856, 0.6769,
        0.5035, 0.6582, 2.1327, 0.1405, 0.1987, 0.3647, 0.2010, 0.4790, 0.5088,
        0.2831, 1.2592, 1.2017, 0.8322, 0.6488, 0.6147, 0.2601, 0.5598, 0.6999,
        0.5171, 0.9466, 0.7184, 1.2233, 0.2387, 0.3642, 0.2260, 0.5488, 0.5829,
        0.3310, 0.8511, 0.2146, 0.3875, 0.4790, 1.3022, 0.4783, 0.4312, 1.0891,
        1.0435, 0.8491, 1.3736, 0.2106, 0.5859, 0.2049, 0.3118, 0.7803, 0.4638,
        0.3090, 1.1290, 0.4079, 0.4707, 0.8774, 0.1191, 0.9179, 0.4473, 3.0123,
        0.3361, 0.5188, 0.7428, 0.6423, 0.3603, 0.7224, 1.2805, 0.7241, 0.2898,
        0.2198, 1.1937, 0.2654, 0.6181, 0.4923, 0.3236, 0.1396, 0.2762, 0.3638,
        1.6510, 0.7606, 1.3095, 0.8973, 1.9803, 0.3549, 0.1095, 0.2767, 0.5654,
```

For : cartwheel\Bodenturnen_2004_cartwheel_f_cm_np1_le_med_0.avi





```
C:\Users\aryan\Desktop\Final>python final_task1.py
Enter path of video : C:\Users\aryan\Desktop\ASU\CSE515\Target_videos\cartwheel\cartwheel\Bodenturnen_2004_cartwheel_f_cm_np1_le_med_0.avi
Enter model name (layer3 / layer4 / avgpool) : layer3
in l3
tensor([-2.6548e-02, -1.6349e-01,  7.5127e-02,  1.1923e-02,  2.1899e-02,
        -3.9692e-02,  6.9583e-02, -1.6292e-02, -1.9344e-02,  2.5155e-03,
         1.4325e-01, -1.7684e-02,  2.0776e-02, -5.0615e-02,  1.2764e-02,
        -8.3801e-03,  2.2542e-02, -1.0058e-01, -1.2945e-01,  7.3878e-02,
         4.2241e-02,  1.2516e-01, -7.2395e-02,  1.1555e-02,  1.2624e-02,
        -4.6405e-02,  6.4096e-02, -1.3397e-01,  8.5677e-02, -6.0176e-02,
        -5.4535e-02, -1.9159e-03, -3.6254e-02, -1.0489e-02,  4.9303e-02,
         3.8031e-02, -2.6979e-02, -8.5306e-03, -7.5476e-02,  7.6854e-03,
         1.2156e-02, -8.6414e-02,  1.4331e-01, -8.9564e-02, -5.3240e-02,
        -5.0400e-02,  3.5465e-02,  1.0059e-01, -1.3380e-02,  1.5079e-01,
        -2.4086e-02,  6.0877e-02, -2.0696e-01, -1.1725e-01,  9.8876e-02,
        -1.0532e-01, -1.3066e-02,  2.3587e-02,  7.9562e-02, -1.9507e-02,
         4.7302e-02,  6.6090e-02, -1.7249e-02,  6.9406e-02,  2.2945e-02,
         7.4554e-02, -4.1882e-03,  2.5211e-02,  8.5928e-02, -1.4523e-02,
        -9.0293e-02, -1.2779e-01, -3.3866e-02,  2.3914e-02,  5.6748e-02,
        -2.1743e-02, -8.0133e-02, -6.7237e-02, -9.2584e-02,  1.2169e-01,
         4.2385e-02,  7.8662e-02, -1.1648e-01,  9.7388e-02,  1.6857e-02,
```

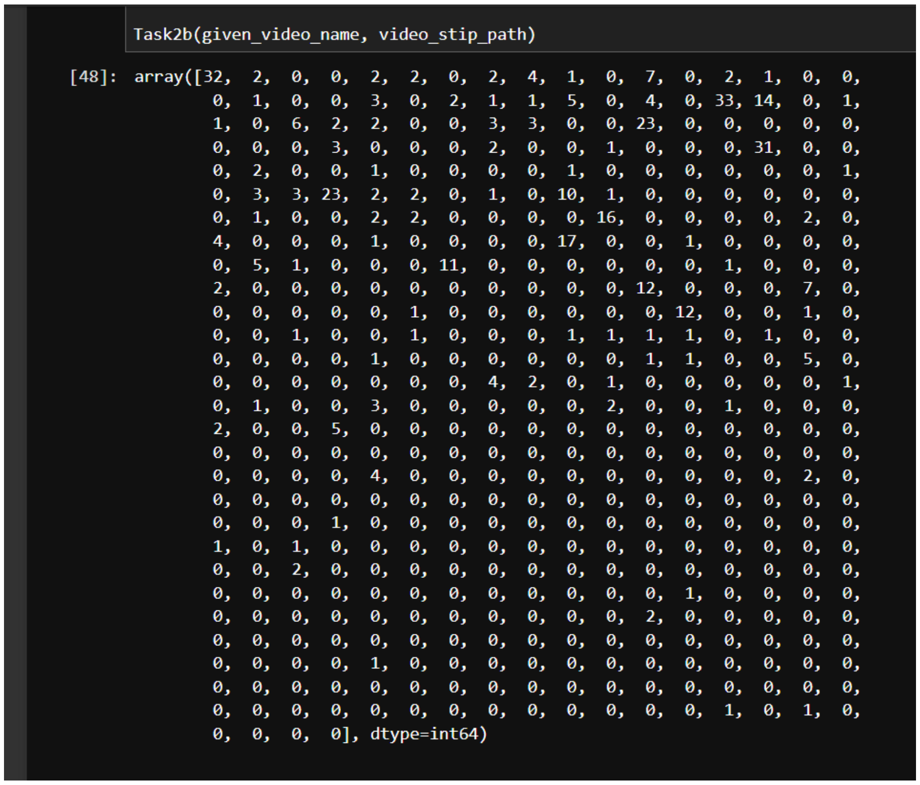
Task 2

In Task 2, after providing the path to the video, the system extracts the top 400 highest-confidence STIPs and processes them to generate spatiotemporal feature descriptors. For each video, 10,000 STIPs are sampled across 12 combinations of spatial (sigma2) and temporal (tau2) scales, and k-means clustering is applied to obtain 40 HoG and HoF cluster representatives per pair. Each STIP is then assigned to the nearest representative, forming 12 histograms for HoG and 12 for HoF, which are concatenated into a 480-dimensional Bag-of-Features (BoF) vector. This vector serves as the feature descriptor for comparing and identifying similar videos.
Later in 2b, we get the BOG-HOF-480 for given video and in 2c, we get the BOG-HOG-480

Following are the Outputs for the 4 sample videos for task 2

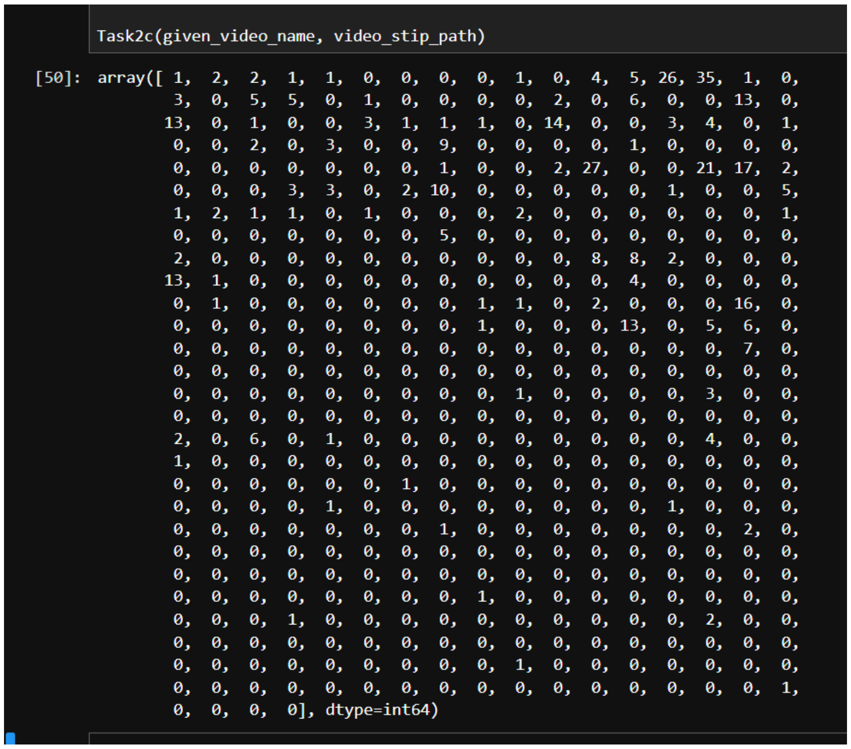
For : "cartwheel/Bodenturnen_2004_cartwheel_f_cm_np1_le_med_0.avi"

2b Output



```
Task2b(given_video_name, video_stip_path)

[48]: array([32,  2,  0,  0,  2,  2,  0,  2,  4,  1,  0,  7,  0,  2,  1,  0,  0,
            0,  1,  0,  0,  3,  0,  2,  1,  1,  5,  0,  4,  0, 33, 14,  0,  1,
            1,  0,  6,  2,  2,  0,  0,  3,  3,  0,  0, 23,  0,  0,  0,  0,  0,
            0,  0,  0,  3,  0,  0,  0,  2,  0,  0,  1,  0,  0,  0, 31,  0,  0,
            0,  2,  0,  0,  1,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  1,
            0,  3,  3, 23,  2,  2,  0,  1,  0, 10,  1,  0,  0,  0,  0,  0,  0,
            0,  1,  0,  0,  2,  2,  0,  0,  0,  0, 16,  0,  0,  0,  0,  2,  0,
            4,  0,  0,  0,  1,  0,  0,  0,  0, 17,  0,  0,  1,  0,  0,  0,  0,
            0,  5,  1,  0,  0,  0, 11,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0,
            2,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, 12,  0,  0,  0,  7,  0,
            0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0, 12,  0,  0,  1,  0,
            0,  0,  1,  0,  0,  1,  0,  0,  0,  1,  1,  1,  1,  0,  1,  0,  0,
            0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  1,  1,  0,  0,  5,  0,
            0,  0,  0,  0,  0,  0,  0,  4,  2,  0,  1,  0,  0,  0,  0,  0,  1,
            0,  1,  0,  0,  3,  0,  0,  0,  0,  0,  2,  0,  0,  1,  0,  0,  0,
            2,  0,  0,  5,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  4,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  2,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            1,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  2,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  2,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  1,  0,
            0,  0,  0,  0], dtype=int64)
```

2c Output

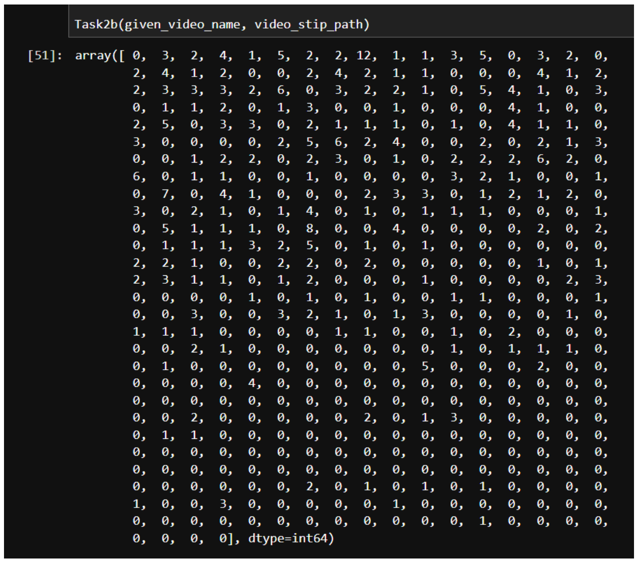


```
Task2c(given_video_name, video_stip_path)

[50]: array([ 1,  2,  2,  1,  1,  0,  0,  0,  0,  1,  0,  4,  5, 26, 35,  1,  0,
              3,  0,  5,  5,  0,  1,  0,  0,  0,  0,  2,  0,  6,  0,  0, 13,  0,
             13,  0,  1,  0,  0,  3,  1,  1,  1,  0, 14,  0,  0,  3,  4,  0,  1,
              0,  0,  2,  0,  3,  0,  0,  9,  0,  0,  0,  0,  1,  0,  0,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  2, 27,  0,  0, 21, 17,  2,
              0,  0,  0,  3,  3,  0,  2, 10,  0,  0,  0,  0,  0,  1,  0,  0,  5,
              1,  2,  1,  1,  0,  1,  0,  0,  0,  2,  0,  0,  0,  0,  0,  0,  1,
              0,  0,  0,  0,  0,  0,  0,  5,  0,  0,  0,  0,  0,  0,  0,  0,  0,
              2,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  8,  8,  2,  0,  0,  0,
             13,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  4,  0,  0,  0,  0,
              0,  1,  0,  0,  0,  0,  0,  0,  1,  1,  0,  2,  0,  0,  0, 16,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0, 13,  0,  5,  6,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  7,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  3,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
              2,  0,  6,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  4,  0,  0,
              1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
              0,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
              0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  2,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,
              0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  2,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,
              0,  0,  0,  0], dtype=int64)
```

For : "sword_exercise/Blade_Of_Fury_-

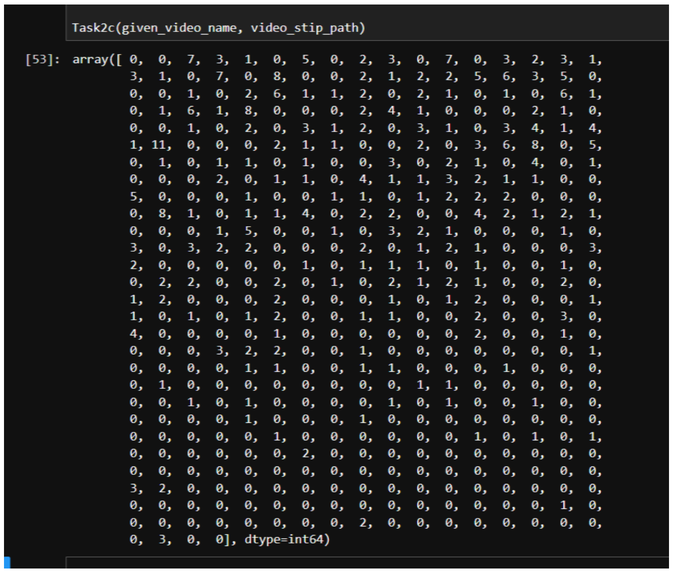
_Scene_1_sword_exercise_f_cm_np1_ri_med_3.avi"

2b Output



```
Task2b(given_video_name, video_stip_path)

[51]: array([ 0,  3,  2,  4,  1,  5,  2,  2, 12,  1,  1,  3,  5,  0,  3,  2,  0,
                2,  4,  1,  2,  0,  0,  2,  4,  2,  1,  1,  0,  0,  0,  4,  1,  2,
                2,  3,  3,  3,  2,  6,  0,  3,  2,  2,  1,  0,  5,  4,  1,  0,  3,
                0,  1,  1,  2,  0,  1,  3,  0,  0,  1,  0,  0,  0,  4,  1,  0,  0,
                2,  5,  0,  3,  3,  0,  2,  1,  1,  1,  0,  1,  0,  4,  1,  1,  0,
                3,  0,  0,  0,  0,  2,  5,  6,  2,  4,  0,  0,  2,  0,  2,  1,  3,
                0,  0,  1,  2,  2,  0,  2,  3,  0,  1,  0,  2,  2,  2,  6,  2,  0,
                6,  0,  1,  1,  0,  0,  1,  0,  0,  0,  0,  3,  2,  1,  0,  0,  1,
                0,  7,  0,  4,  1,  0,  0,  0,  2,  3,  3,  0,  1,  2,  1,  2,  0,
                3,  0,  2,  1,  0,  1,  4,  0,  1,  0,  1,  1,  1,  0,  0,  0,  1,
                0,  5,  1,  1,  1,  0,  8,  0,  0,  4,  0,  0,  0,  0,  2,  0,  2,
                0,  1,  1,  1,  3,  2,  5,  0,  1,  0,  1,  0,  0,  0,  0,  0,  0,
                2,  2,  1,  0,  0,  2,  2,  0,  2,  0,  0,  0,  0,  0,  1,  0,  1,
                2,  3,  1,  1,  0,  1,  2,  0,  0,  0,  1,  0,  0,  0,  0,  2,  3,
                0,  0,  0,  0,  1,  0,  1,  0,  1,  0,  0,  1,  1,  0,  0,  0,  1,
                0,  0,  3,  0,  0,  3,  2,  1,  0,  1,  3,  0,  0,  0,  0,  1,  0,
                1,  1,  1,  0,  0,  0,  0,  1,  1,  0,  0,  1,  0,  2,  0,  0,  0,
                0,  0,  2,  1,  0,  0,  0,  0,  0,  0,  0,  1,  0,  1,  1,  1,  0,
                0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  5,  0,  0,  0,  2,  0,  0,
                0,  0,  0,  0,  4,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                0,  0,  2,  0,  0,  0,  0,  0,  2,  0,  1,  3,  0,  0,  0,  0,  0,
                0,  1,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                0,  0,  0,  0,  0,  0,  2,  0,  1,  0,  1,  0,  1,  0,  0,  0,  0,
                1,  0,  0,  3,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,
                0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,
                0,  0,  0,  0], dtype=int64)
```

2c output

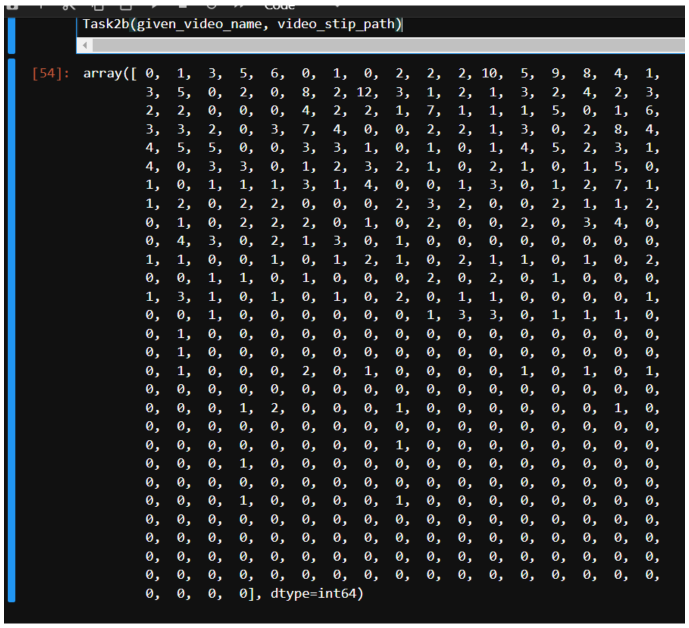


```
Task2c(given_video_name, video_stip_path)

[53]: array([ 0,  0,  7,  3,  1,  0,  5,  0,  2,  3,  0,  7,  0,  3,  2,  3,  1,
                3,  1,  0,  7,  0,  8,  0,  0,  2,  1,  2,  2,  5,  6,  3,  5,  0,
                0,  0,  1,  0,  2,  6,  1,  1,  2,  0,  2,  1,  0,  1,  0,  6,  1,
                0,  1,  6,  1,  8,  0,  0,  0,  2,  4,  1,  0,  0,  0,  2,  1,  0,
                0,  0,  1,  0,  2,  0,  3,  1,  2,  0,  3,  1,  0,  3,  4,  1,  4,
                1, 11,  0,  0,  0,  2,  1,  1,  0,  0,  2,  0,  3,  6,  8,  0,  5,
                0,  1,  0,  1,  1,  0,  1,  0,  0,  3,  0,  2,  1,  0,  4,  0,  1,
                0,  0,  0,  2,  0,  1,  1,  0,  4,  1,  1,  3,  2,  1,  1,  0,  0,
                5,  0,  0,  0,  1,  0,  0,  1,  1,  0,  1,  2,  2,  2,  0,  0,  0,
                0,  8,  1,  0,  1,  1,  4,  0,  2,  2,  0,  0,  4,  2,  1,  2,  1,
                0,  0,  0,  1,  5,  0,  0,  1,  0,  3,  2,  1,  0,  0,  0,  1,  0,
                3,  0,  3,  2,  2,  0,  0,  0,  2,  0,  1,  2,  1,  0,  0,  0,  3,
                2,  0,  0,  0,  0,  0,  1,  0,  1,  1,  1,  0,  1,  0,  0,  1,  0,
                0,  2,  2,  0,  0,  2,  0,  1,  0,  2,  1,  2,  1,  0,  0,  2,  0,
                1,  2,  0,  0,  0,  2,  0,  0,  0,  1,  0,  1,  2,  0,  0,  0,  1,
                1,  0,  1,  0,  1,  2,  0,  0,  1,  1,  0,  0,  2,  0,  0,  3,  0,
                4,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  2,  0,  0,  1,  0,
                0,  0,  0,  3,  2,  2,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  1,
                0,  0,  0,  0,  1,  1,  0,  0,  1,  1,  0,  0,  0,  1,  0,  0,  0,
                0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  1,  1,  0,  0,  0,  0,  0,
                0,  0,  1,  0,  1,  0,  0,  0,  0,  1,  0,  1,  0,  0,  1,  0,  0,
                0,  0,  0,  0,  1,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,
                0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  1,  0,  1,  0,  1,
                0,  0,  0,  0,  0,  0,  2,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                3,  2,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,
                0,  0,  0,  0,  0,  0,  0,  0,  2,  0,  0,  0,  0,  0,  0,  0,  0,
                0,  3,  0,  0], dtype=int64)
```

For :

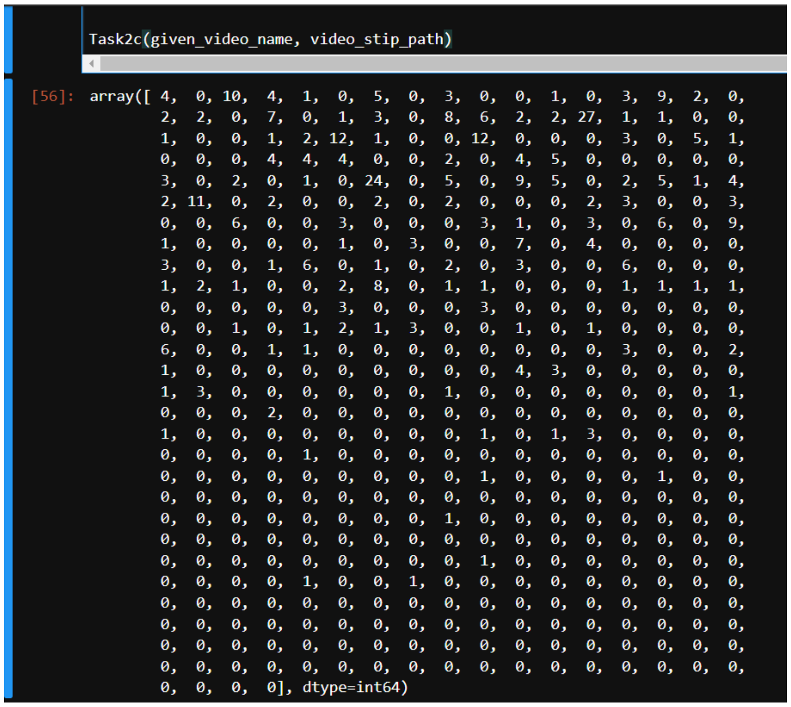
"sword/AHF_longsword_against_Rapier_and_Dagger_Fight_sword_f_cm_np2_ri_bad_0.avi"

2b Output



```
Task2b(given_video_name, video_stip_path)

[54]: array([ 0,  1,  3,  5,  6,  0,  1,  0,  2,  2,  2, 10,  5,  9,  8,  4,  1,
             3,  5,  0,  2,  0,  8,  2, 12,  3,  1,  2,  1,  3,  2,  4,  2,  3,
             2,  2,  0,  0,  0,  4,  2,  2,  1,  7,  1,  1,  1,  5,  0,  1,  6,
             3,  3,  2,  0,  3,  7,  4,  0,  0,  2,  2,  1,  3,  0,  2,  8,  4,
             4,  5,  5,  0,  0,  3,  3,  1,  0,  1,  0,  1,  4,  5,  2,  3,  1,
             4,  0,  3,  3,  0,  1,  2,  3,  2,  1,  0,  2,  1,  0,  1,  5,  0,
             1,  0,  1,  1,  1,  3,  1,  4,  0,  0,  1,  3,  0,  1,  2,  7,  1,
             1,  2,  0,  2,  2,  0,  0,  0,  2,  3,  2,  0,  0,  2,  1,  1,  2,
             0,  1,  0,  2,  2,  2,  0,  1,  0,  2,  0,  0,  2,  0,  3,  4,  0,
             0,  4,  3,  0,  2,  1,  3,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,
             1,  1,  0,  0,  1,  0,  1,  2,  1,  0,  2,  1,  1,  0,  1,  0,  2,
             0,  0,  1,  1,  0,  1,  0,  0,  0,  2,  0,  2,  0,  1,  0,  0,  0,
             1,  3,  1,  0,  1,  0,  1,  0,  2,  0,  1,  1,  0,  0,  0,  0,  1,
             0,  0,  1,  0,  0,  0,  0,  0,  0,  1,  3,  3,  0,  1,  1,  1,  0,
             0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  1,  0,  0,  0,  2,  0,  1,  0,  0,  0,  0,  1,  0,  1,  0,  1,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  1,  2,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  1,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  1,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0], dtype=int64)
```

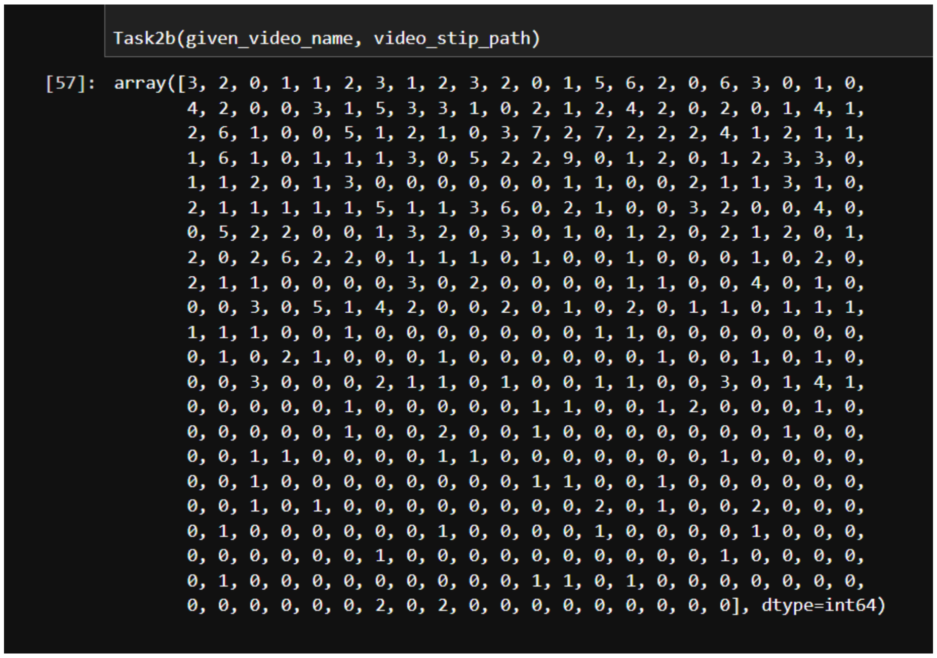
2c Output



```
Task2c(given_video_name, video_stip_path)

[56]: array([ 4,  0, 10,  4,  1,  0,  5,  0,  3,  0,  0,  1,  0,  3,  9,  2,  0,
             2,  2,  0,  7,  0,  1,  3,  0,  8,  6,  2,  2, 27,  1,  1,  0,  0,
             1,  0,  0,  1,  2, 12,  1,  0,  0, 12,  0,  0,  0,  3,  0,  5,  1,
             0,  0,  0,  4,  4,  4,  0,  0,  2,  0,  4,  5,  0,  0,  0,  0,  0,
             3,  0,  2,  0,  1,  0, 24,  0,  5,  0,  9,  5,  0,  2,  5,  1,  4,
             2, 11,  0,  2,  0,  0,  2,  0,  2,  0,  0,  0,  2,  3,  0,  0,  3,
             0,  0,  6,  0,  0,  3,  0,  0,  0,  3,  1,  0,  3,  0,  6,  0,  9,
             1,  0,  0,  0,  0,  1,  0,  3,  0,  0,  7,  0,  4,  0,  0,  0,  0,
             3,  0,  0,  1,  6,  0,  1,  0,  2,  0,  3,  0,  0,  6,  0,  0,  0,
             1,  2,  1,  0,  0,  2,  8,  0,  1,  1,  0,  0,  0,  1,  1,  1,  1,
             0,  0,  0,  0,  0,  3,  0,  0,  0,  3,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  1,  0,  1,  2,  1,  3,  0,  0,  1,  0,  1,  0,  0,  0,  0,
             6,  0,  0,  1,  1,  0,  0,  0,  0,  0,  0,  0,  0,  3,  0,  0,  2,
             1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  4,  3,  0,  0,  0,  0,  0,
             1,  3,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  1,
             0,  0,  0,  2,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             1,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  1,  3,  0,  0,  0,  0,
             0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  1,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  1,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             0,  0,  0,  0], dtype=int64)
```

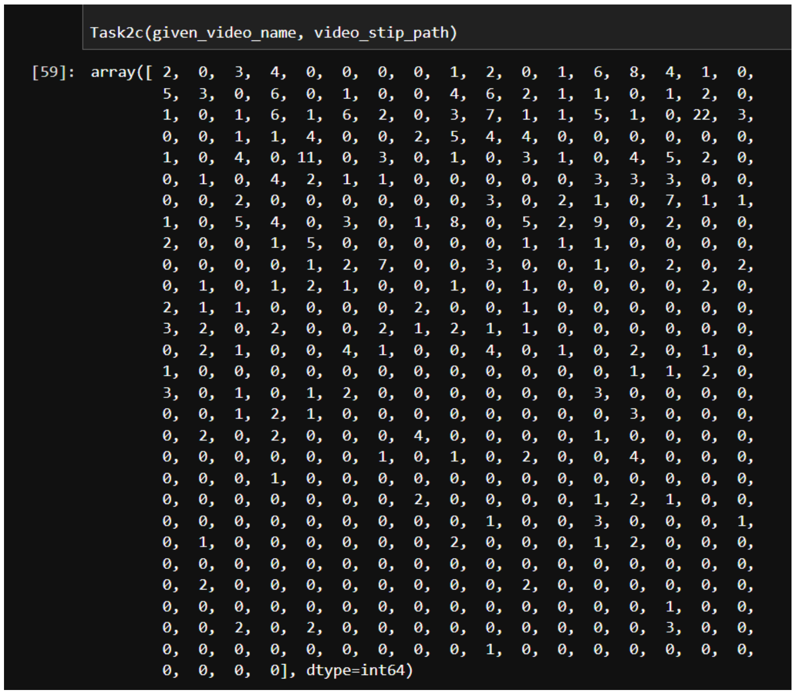
For : "drink/CastAway2_drink_u_cm_np1_le_goo_8.avi"

2b Output



```
Task2b(given_video_name, video_stip_path)

[57]: array([3, 2, 0, 1, 1, 2, 3, 1, 2, 3, 2, 0, 1, 5, 6, 2, 0, 6, 3, 0, 1, 0,
             4, 2, 0, 0, 3, 1, 5, 3, 3, 1, 0, 2, 1, 2, 4, 2, 0, 2, 0, 1, 4, 1,
             2, 6, 1, 0, 0, 5, 1, 2, 1, 0, 3, 7, 2, 7, 2, 2, 2, 4, 1, 2, 1, 1,
             1, 6, 1, 0, 1, 1, 1, 3, 0, 5, 2, 2, 9, 0, 1, 2, 0, 1, 2, 3, 3, 0,
             1, 1, 2, 0, 1, 3, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 2, 1, 1, 3, 1, 0,
             2, 1, 1, 1, 1, 1, 5, 1, 1, 3, 6, 0, 2, 1, 0, 0, 3, 2, 0, 0, 4, 0,
             0, 5, 2, 2, 0, 0, 1, 3, 2, 0, 3, 0, 1, 0, 1, 2, 0, 2, 1, 2, 0, 1,
             2, 0, 2, 6, 2, 2, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 2, 0,
             2, 1, 1, 0, 0, 0, 0, 3, 0, 2, 0, 0, 0, 0, 1, 1, 0, 0, 4, 0, 1, 0,
             0, 0, 3, 0, 5, 1, 4, 2, 0, 0, 2, 0, 1, 0, 2, 0, 1, 1, 0, 1, 1, 1,
             1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0,
             0, 1, 0, 2, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0,
             0, 0, 3, 0, 0, 0, 2, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 3, 0, 1, 4, 1,
             0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 2, 0, 0, 0, 1, 0,
             0, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
             0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
             0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
             0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 1, 0, 0, 2, 0, 0, 0,
             0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
             0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

2c Output



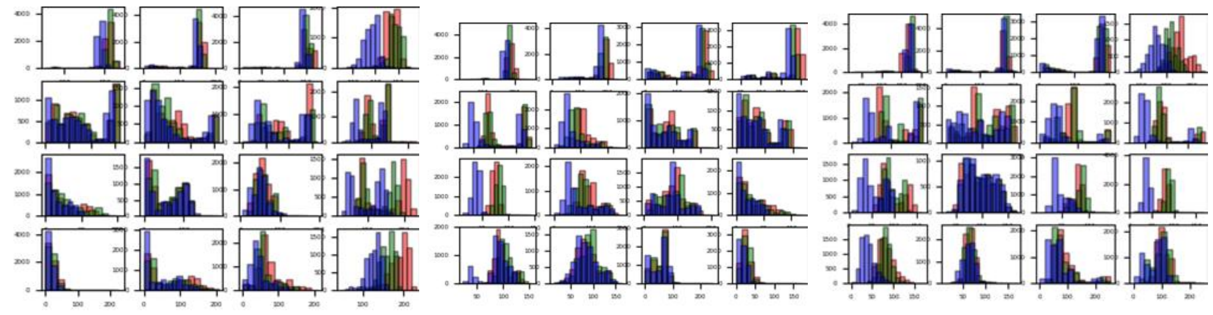
```
Task2c(given_video_name, video_stip_path)

[59]: array([ 2,  0,  3,  4,  0,  0,  0,  0,  1,  2,  0,  1,  6,  8,  4,  1,  0,
              5,  3,  0,  6,  0,  1,  0,  0,  4,  6,  2,  1,  1,  0,  1,  2,  0,
              1,  0,  1,  6,  1,  6,  2,  0,  3,  7,  1,  1,  5,  1,  0, 22,  3,
              0,  0,  1,  1,  4,  0,  0,  2,  5,  4,  4,  0,  0,  0,  0,  0,  0,
              1,  0,  4,  0, 11,  0,  3,  0,  1,  0,  3,  1,  0,  4,  5,  2,  0,
              0,  1,  0,  4,  2,  1,  1,  0,  0,  0,  0,  0,  3,  3,  3,  0,  0,
              0,  0,  2,  0,  0,  0,  0,  0,  0,  3,  0,  2,  1,  0,  7,  1,  1,
              1,  0,  5,  4,  0,  3,  0,  1,  8,  0,  5,  2,  9,  0,  2,  0,  0,
              2,  0,  0,  1,  5,  0,  0,  0,  0,  0,  1,  1,  1,  0,  0,  0,  0,
              0,  0,  0,  0,  1,  2,  7,  0,  0,  3,  0,  0,  1,  0,  2,  0,  2,
              0,  1,  0,  1,  2,  1,  0,  0,  1,  0,  1,  0,  0,  0,  0,  2,  0,
              2,  1,  1,  0,  0,  0,  0,  2,  0,  0,  1,  0,  0,  0,  0,  0,  0,
              3,  2,  0,  2,  0,  0,  2,  1,  2,  1,  1,  0,  0,  0,  0,  0,  0,
              0,  2,  1,  0,  0,  4,  1,  0,  0,  4,  0,  1,  0,  2,  0,  1,  0,
              1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  1,  2,  0,
              3,  0,  1,  0,  1,  2,  0,  0,  0,  0,  0,  0,  3,  0,  0,  0,  0,
              0,  0,  1,  2,  1,  0,  0,  0,  0,  0,  0,  0,  0,  3,  0,  0,  0,
              0,  2,  0,  2,  0,  0,  0,  4,  0,  0,  0,  0,  1,  0,  0,  0,  0,
              0,  0,  0,  0,  0,  0,  1,  0,  1,  0,  2,  0,  0,  4,  0,  0,  0,
              0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  2,  0,  0,  0,  0,  1,  2,  1,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  3,  0,  0,  0,  1,
              0,  1,  0,  0,  0,  0,  0,  0,  2,  0,  0,  0,  1,  2,  0,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
              0,  2,  0,  0,  0,  0,  0,  0,  0,  0,  2,  0,  0,  0,  0,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  0,
              0,  0,  2,  0,  2,  0,  0,  0,  0,  0,  0,  0,  0,  0,  3,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,
              0,  0,  0,  0], dtype=int64)
```

Task 3

Task 3 takes a the path of selected video as input. Provide the absolute or relative path of the video. Next it will store the feature vector, visualization of histogram and the three frames. The frames and features will be stored in the folders “Frames” and “Features” respectively and histogram will be saved in the task3_save folder.

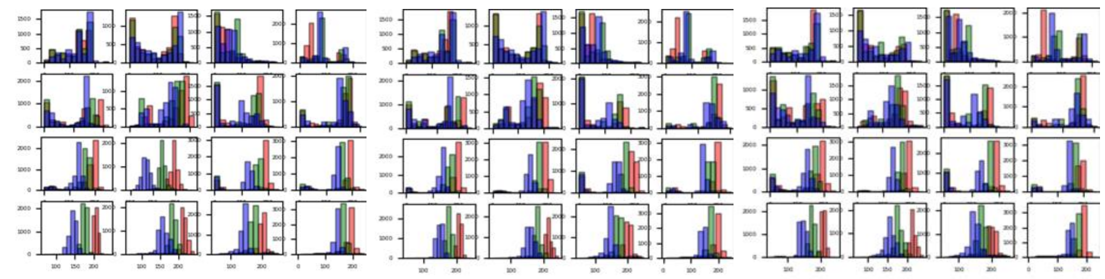
Following are the Outputs for the 4 sample videos for task 3

For : "drink\CastAway2_drink_u_cm_np1_le_goo_8.avi"

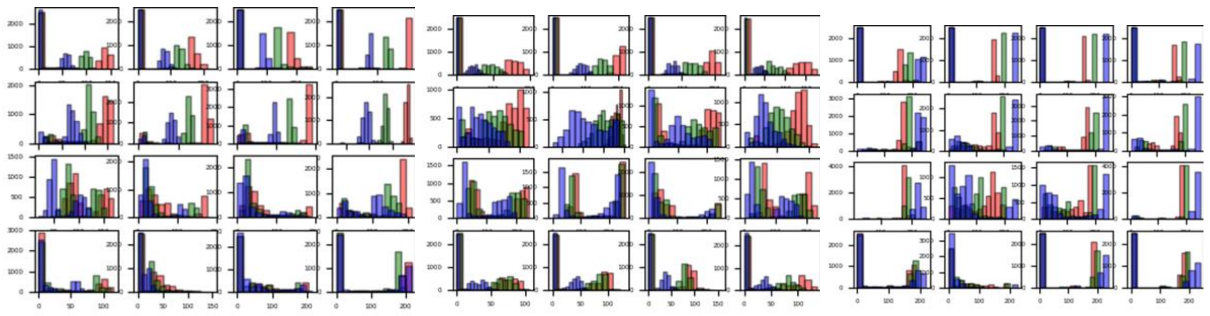


For :

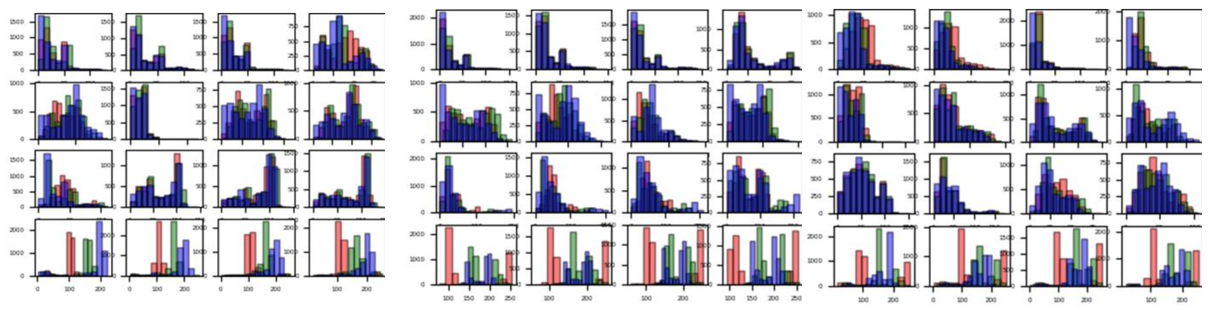
sword\AHF_longsword_against_Rapier_and_Dagger_Fight_sword_f_cm_np2_ri_bad_0.avi



For : sword_exercise\Blade_Of_Fury_-_Scene_1_sword_exercise_f_cm_np1_ri_med_3.avi



For : cartwheel\Bodenturnen_2004_cartwheel_f_cm_np1_le_med_0.avi

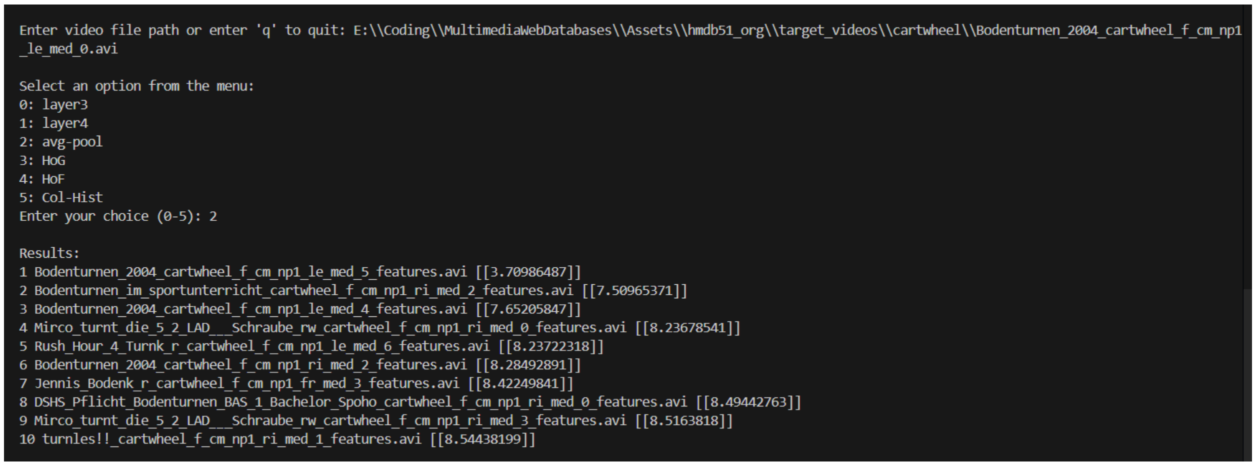


Task 5

Task 5 takes the path of selected video as input. For the given model, it performs a feature comparison with the target_videos to give 10 most similar videos.

Following are the Outputs for the 4 sample videos for task 5

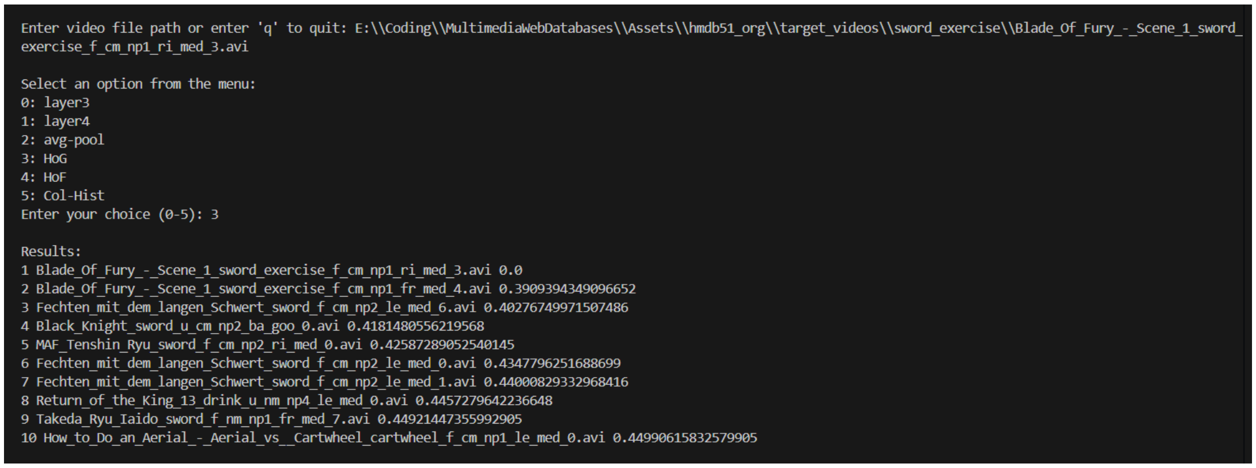
For : "cartwheel/Bodenturnen_2004_cartwheel_f_cm_np1_le_med_0.avi"



```
Enter video file path or enter 'q' to quit: E:\\Coding\\MultimediaWebDatabases\\Assets\\hmdb51_org\\target_videos\\cartwheel\\Bodenturnen_2004_cartwheel_f_cm_np1
_le_med_0.avi

Select an option from the menu:
0: layer3
1: layer4
2: avg-pool
3: HoG
4: HoF
5: Col-Hist
Enter your choice (0-5): 2

Results:
1 Bodenturnen_2004_cartwheel_f_cm_np1_le_med_5_features.avi [[3.70986487]]
2 Bodenturnen_im_sportunterricht_cartwheel_f_cm_np1_ri_med_2_features.avi [[7.50965371]]
3 Bodenturnen_2004_cartwheel_f_cm_np1_le_med_4_features.avi [[7.65205847]]
4 Mirco_turnt_die_5_2_LAD___Schraube_rw_cartwheel_f_cm_np1_ri_med_0_features.avi [[8.23678541]]
5 Rush_Hour_4_Turnk_r_cartwheel_f_cm_np1_le_med_6_features.avi [[8.23722318]]
6 Bodenturnen_2004_cartwheel_f_cm_np1_ri_med_2_features.avi [[8.28492891]]
7 Jennis_Bodenk_r_cartwheel_f_cm_np1_fr_med_3_features.avi [[8.42249841]]
8 DSHS_Pflicht_Bodenturnen_BAS_1_Bachelor_Spoho_cartwheel_f_cm_np1_ri_med_0_features.avi [[8.49442763]]
9 Mirco_turnt_die_5_2_LAD___Schraube_rw_cartwheel_f_cm_np1_ri_med_3_features.avi [[8.5163818]]
10 turnles!!_cartwheel_f_cm_np1_ri_med_1_features.avi [[8.54438199]]
```

For : "sword_exercise/Blade_Of_Fury_-

_Scene_1_sword_exercise_f_cm_np1_ri_med_3.avi"

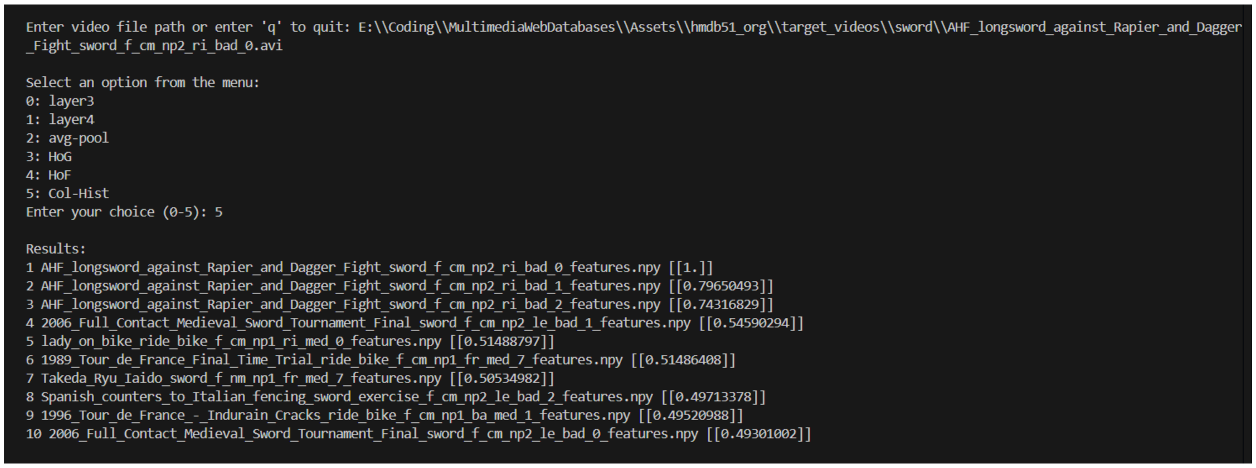


```
Enter video file path or enter 'q' to quit: E:\\Coding\\MultimediaWebDatabases\\Assets\\hmdb51_org\\target_videos\\sword_exercise\\Blade_Of_Fury_-_Scene_1_sword_
exercise_f_cm_np1_ri_med_3.avi

Select an option from the menu:
0: layer3
1: layer4
2: avg-pool
3: HoG
4: HoF
5: Col-Hist
Enter your choice (0-5): 3

Results:
1 Blade_Of_Fury_-_Scene_1_sword_exercise_f_cm_np1_ri_med_3.avi 0.0
2 Blade_Of_Fury_-_Scene_1_sword_exercise_f_cm_np1_fr_med_4.avi 0.3909394349096652
3 Fechten_mit_dem_langen_Schwert_sword_f_cm_np2_le_med_6.avi 0.40276749971507486
4 Black_Knight_sword_u_cm_np2_ba_goo_0.avi 0.4181480556219568
5 MAF_Tenshin_Ryu_sword_f_cm_np2_ri_med_0.avi 0.42587289052540145
6 Fechten_mit_dem_langen_Schwert_sword_f_cm_np2_le_med_0.avi 0.4347796251688699
7 Fechten_mit_dem_langen_Schwert_sword_f_cm_np2_le_med_1.avi 0.44000829332968416
8 Return_of_the_King_13_drink_u_nm_np4_le_med_0.avi 0.4457279642236648
9 Takeda_Ryu_Iaido_sword_f_nm_np1_fr_med_7.avi 0.44921447355992905
10 How_to_Do_an_Aerial_-_Aerial_vs__Cartwheel_cartwheel_f_cm_np1_le_med_0.avi 0.44990615832579905
```

For :

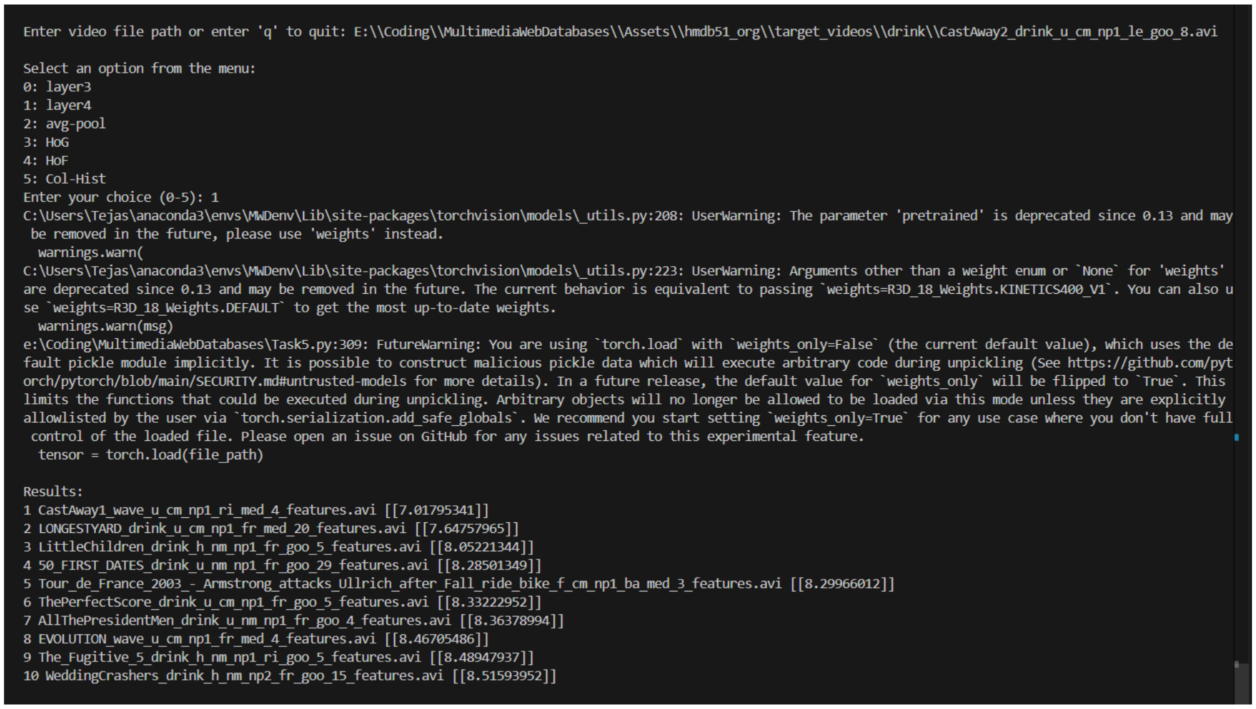
"sword/AHF_longsword_against_Rapier_and_Dagger_Fight_sword_f_cm_np2_ri_bad_0.a

vi"



```
Enter video file path or enter 'q' to quit: E:\\Coding\\MultimediaWebDatabases\\Assets\\hmdb51_org\\target_videos\\sword\\AHF_longsword_against_Rapier_and_Dagger
_Fight_sword_f_cm_np2_ri_bad_0.avi

Select an option from the menu:
0: layer3
1: layer4
2: avg-pool
3: HoG
4: HoF
5: Col-Hist
Enter your choice (0-5): 5

Results:
1 AHF_longsword_against_Rapier_and_Dagger_Fight_sword_f_cm_np2_ri_bad_0_features.npy [[1.]]
2 AHF_longsword_against_Rapier_and_Dagger_Fight_sword_f_cm_np2_ri_bad_1_features.npy [[0.79650493]]
3 AHF_longsword_against_Rapier_and_Dagger_Fight_sword_f_cm_np2_ri_bad_2_features.npy [[0.74316829]]
4 2006_Full_Contact_Medieval_Sword_Tournament_Final_sword_f_cm_np2_le_bad_1_features.npy [[0.54590294]]
5 lady_on_bike_ride_bike_f_cm_np1_ri_med_0_features.npy [[0.51488797]]
6 1989_Tour_de_France_Final_Time_Trial_ride_bike_f_cm_np1_fr_med_7_features.npy [[0.51486408]]
7 Takeda_Ryu_Iaido_sword_f_nm_np1_fr_med_7_features.npy [[0.50534982]]
8 Spanish_counters_to_Italian_fencing_sword_exercise_f_cm_np2_le_bad_2_features.npy [[0.49713378]]
9 1996_Tour_de_France_-_Indurain_Cracks_ride_bike_f_cm_np1_ba_med_1_features.npy [[0.49520988]]
10 2006_Full_Contact_Medieval_Sword_Tournament_Final_sword_f_cm_np2_le_bad_0_features.npy [[0.49301002]]
```

For : "drink\CastAway2_drink_u_cm_np1_le_goo_8.avi"



```
Enter video file path or enter 'q' to quit: E:\\Coding\\MultimediaWebDatabases\\Assets\\hmdb51_org\\target_videos\\drink\\CastAway2_drink_u_cm_np1_le_goo_8.avi

Select an option from the menu:
0: layer3
1: layer4
2: avg-pool
3: HoG
4: HoF
5: Col-Hist
Enter your choice (0-5): 1
C:\Users\Tejas\anaconda3\envs\MWDenv\Lib\site-packages\torchvision\models\_utils.py:208: UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may
 be removed in the future, please use 'weights' instead.
  warnings.warn(
C:\Users\Tejas\anaconda3\envs\MWDenv\Lib\site-packages\torchvision\models\_utils.py:223: UserWarning: Arguments other than a weight enum or `None` for 'weights'
are deprecated since 0.13 and may be removed in the future. The current behavior is equivalent to passing `weights=R3D_18_Weights.KINETICS400_V1`. You can also u
se `weights=R3D_18_Weights.DEFAULT` to get the most up-to-date weights.
  warnings.warn(msg)
e:\Coding\MultimediaWebDatabases\Task5.py:309: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default value), which uses the de
fault pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See https://github.com/pyt
orch/pytorch/blob/main/SECURITY.md#untrusted-models for more details). In a future release, the default value for `weights_only` will be flipped to `True`. This
limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly
allowlisted by the user via `torch.serialization.add_safe_globals`. We recommend you start setting `weights_only=True` for any use case where you don't have full
 control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.
  tensor = torch.load(file_path)

Results:
1 CastAway1_wave_u_cm_np1_ri_med_4_features.avi [[7.01795341]]
2 LONGESTYARD_drink_u_cm_np1_fr_med_20_features.avi [[7.64757965]]
3 LittleChildren_drink_h_nm_np1_fr_goo_5_features.avi [[8.05221344]]
4 50_FIRST_DATES_drink_u_nm_np1_fr_goo_29_features.avi [[8.28501349]]
5 Tour_de_France_2003_-_Armstrong_attacks_Ullrich_after_Fall_ride_bike_f_cm_np1_ba_med_3_features.avi [[8.29966012]]
6 ThePerfectScore_drink_u_cm_np1_fr_goo_5_features.avi [[8.33222952]]
7 AllThePresidentMen_drink_u_nm_np1_fr_goo_4_features.avi [[8.36378994]]
8 EVOLUTION_wave_u_cm_np1_fr_med_4_features.avi [[8.46705486]]
9 The_Fugitive_5_drink_h_nm_np1_ri_goo_5_features.avi [[8.48947937]]
10 WeddingCrashers_drink_h_nm_np2_fr_goo_15_features.avi [[8.51593952]]
```

Related Work

There are several proposed approaches for video action recognition. Some of these include Spatio Temporal Interest Points(STIPS), optical flow based techniques as well as deep convolutional neural networks. Spatio Temporal Interest Points (STIPS) are the points in a video that show some significant change in either appearance or motion[1]. These points are then used to extract features such as Histogram of Oriented Gradient (HOG) [4] and Histogram of Optical Flow (HOF). These can then be used to extract features and represent actions using techniques such as bag of words. Dhulekar et al. [1] used a combination of HOG and HOF features from STIPS to represent different actions. This demonstrated the effectiveness of combining spatial and temporal information for action recognition.

Conclusion

This phase of the project focused on extracting features from the HMDB51 human motion dataset and using them to identify similar videos from the dataset. The feature extraction methods include Histogram of Optical Flow (HoF), Histogram of Oriented Gradients (HoG), and color histograms. The project is divided into five tasks, each involving different approaches to feature extraction, storage, and similarity comparison of the videos in the dataset. Each of the first three tasks involve unique ways of extracting features from the given video. The extracted features are stored in task 4 which are then used to compare videos using distance metrics like cosine distance in the final task of the project.

Bibliography

1. A. Dhulekar, P., and S. T. Gandhe. ‘Action Recognition Based on Histogram of Oriented Gradients and Spatio-Temporal Interest Points’. *International Journal of Engineering & Technology*, vol. 7, no. 4, Sept. 2018, p. 2153. *DOI.org (Crossref)*, https://doi.org/10.14419/ijet.v7i4.17274.
2. Kuehne, H., et al. ‘HMDB: A Large Video Database for Human Motion Recognition’. *2011 International Conference on Computer Vision,* 2011, pp. 2556–63. IEEE Xplore, https://doi.org/10.1109/ICCV.2011.6126543.
3. Tran, Du, et al. *A Closer Look at Spatiotemporal Convolutions for Action Recognition.* arXiv, 2017. DOI.org (Datacite), https://doi.org/10.48550/ARXIV.1711.11248.
4. Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 886-893). IEEE.

Appendix

Specific roles of the group members

Aryan Patel - Tasks 1 and 4

Tejas Parse - Task 2

Om Patel - Task 3

Tanishque Zaware - Task 5