Machine Learning Project

Human Activity Recognition

Using kernelaized SVM

Group: 7

Harshit Giri (2021150)

Raghav Sakhuja(2021274)

Akshet Patial (MT23155)

Saurav (2020244)

Shivam (MT23004)

INDRAPRASTHA INSTITUTE *of* INFORMATION TECHNOLOGY

DELHI

Motivation & Problem Statement

- In the modern world, humans are evolving exponentially, giving rise to the dawn of a new era marked by unprecedented advancements in Artificial Intelligence and machine learning. Out of all the multiple domains, human activity recognition has emerged as a transformative force.
- From healthcare to security and beyond, HAR has contributed and placed itself in all areas, whether we talk about body mapping and posture detection in medical fields or security assistive technologies in Corporations.
- Our group believes diving deeper into human activity recognition and supporting technological advancements can help us develop a revolutionary result.

Project Assignment

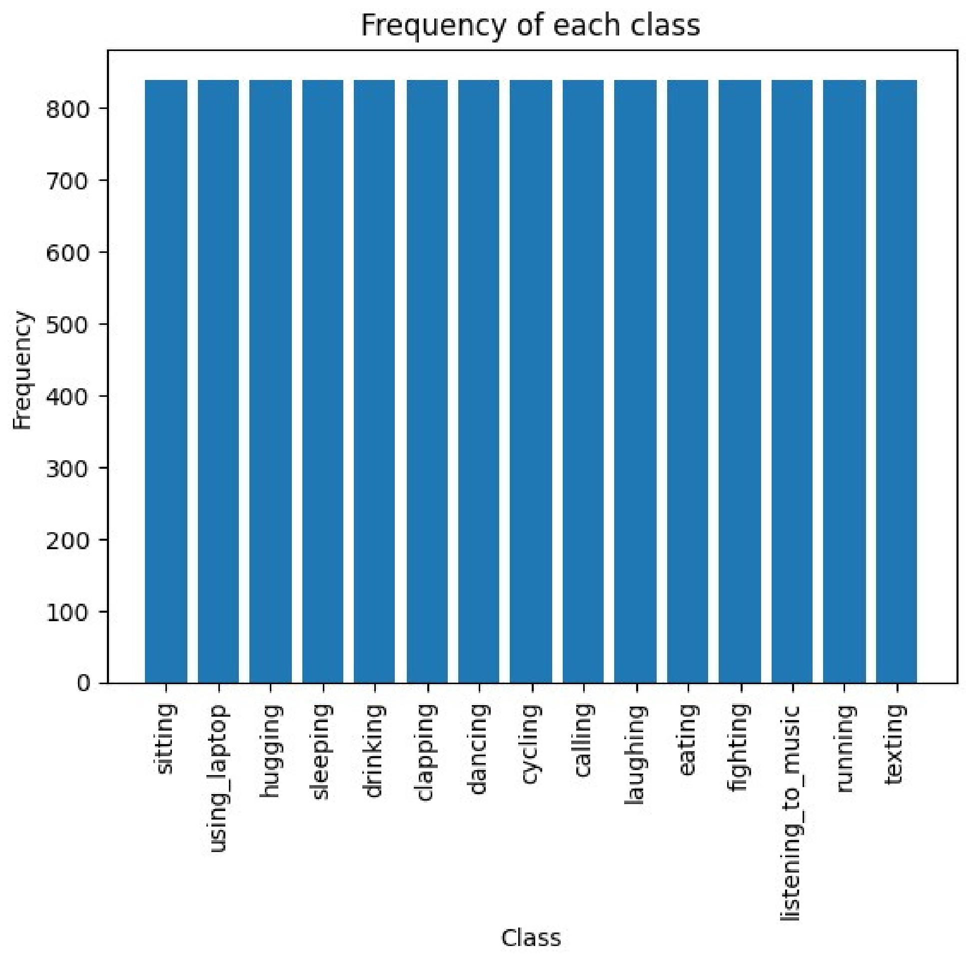
We were tasked with creating a **Human Activity Recognizer (HAR) utilizing Kernelized Support Vector Machines for our Machine Learning Project.**

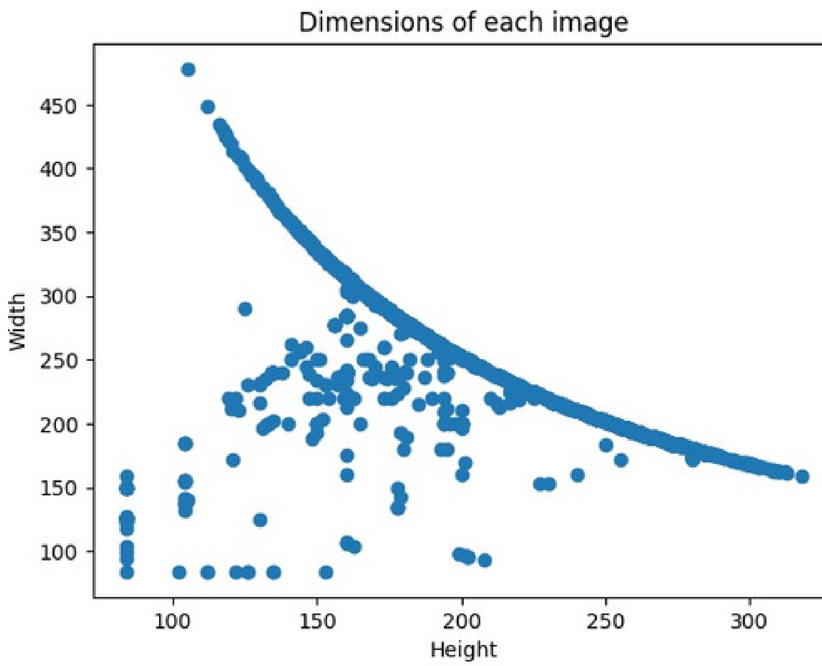
We were provided a training dataset with photos of various human activities such as "**Talking**," "**Sitting**," "**Running**," and "**Using laptop**," etc... .

Using this dataset, we had to train a model that can predict human activity based on the visualization of a 2D image given in the HAR dataset.

Dataset Analysis & Basic EDA

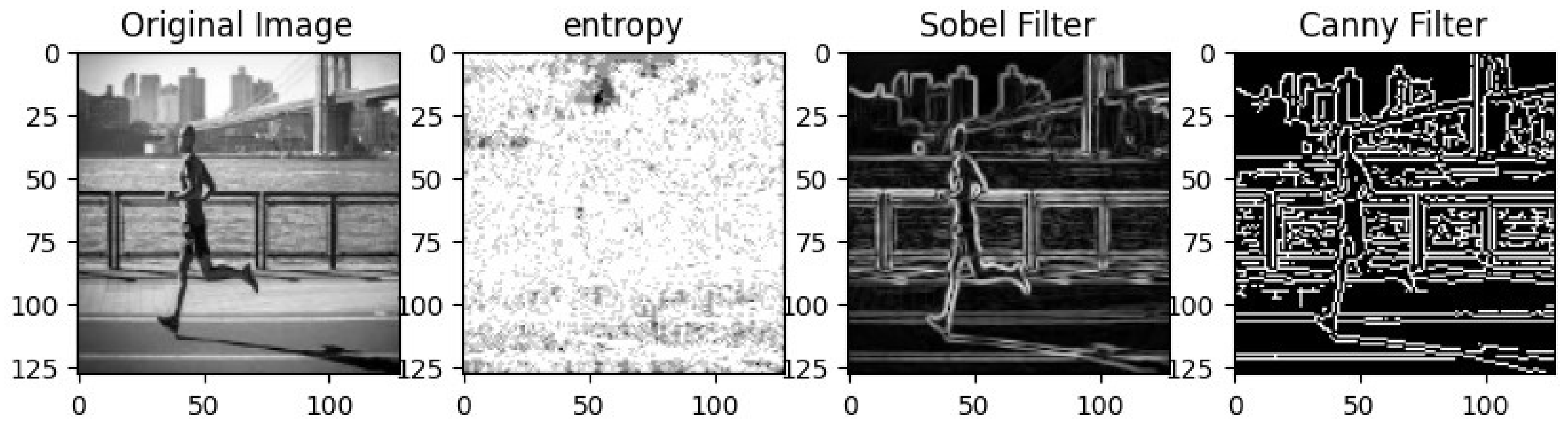
- After exploring the dataset, it was identified that there are a total of 15 unique activities.
- A visual representation of the activity versus image frequency count revealed that each activity is represented by an equal number of images.
- The images in the dataset vary in size, with flattened sizes ranging from 7056 pixels to 50625 pixels.
- Median luminous intensity values were ranging from (80,165) where contrast values ranged from a minimum of 55 to a maximum of 65.



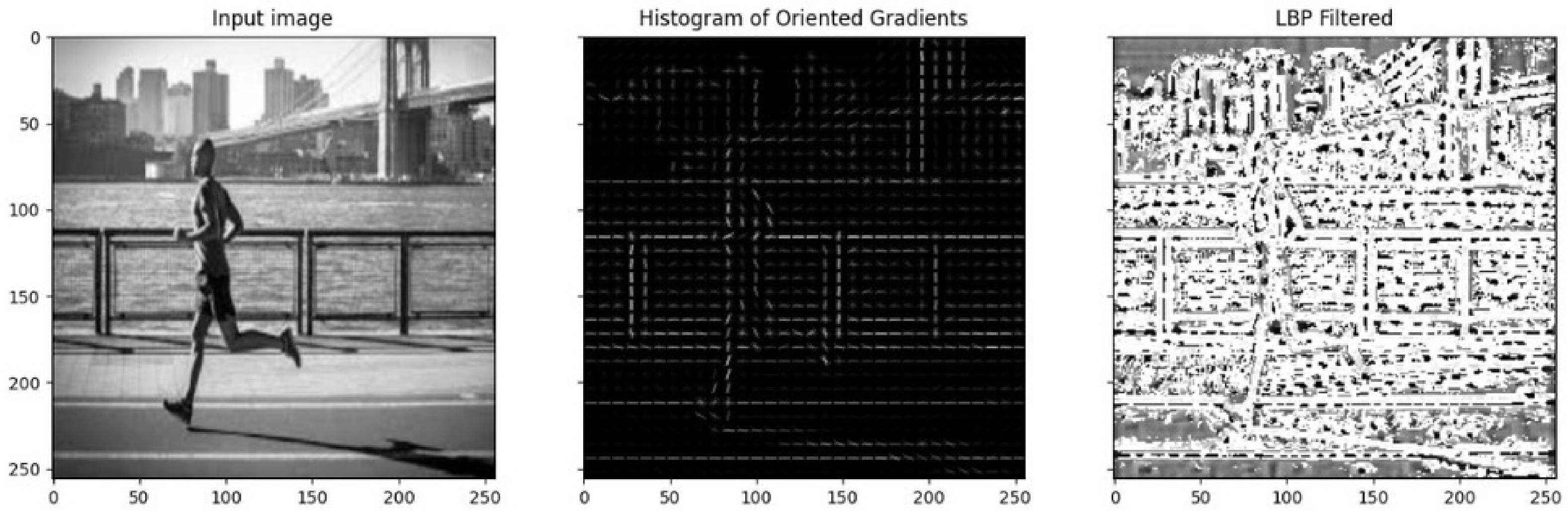


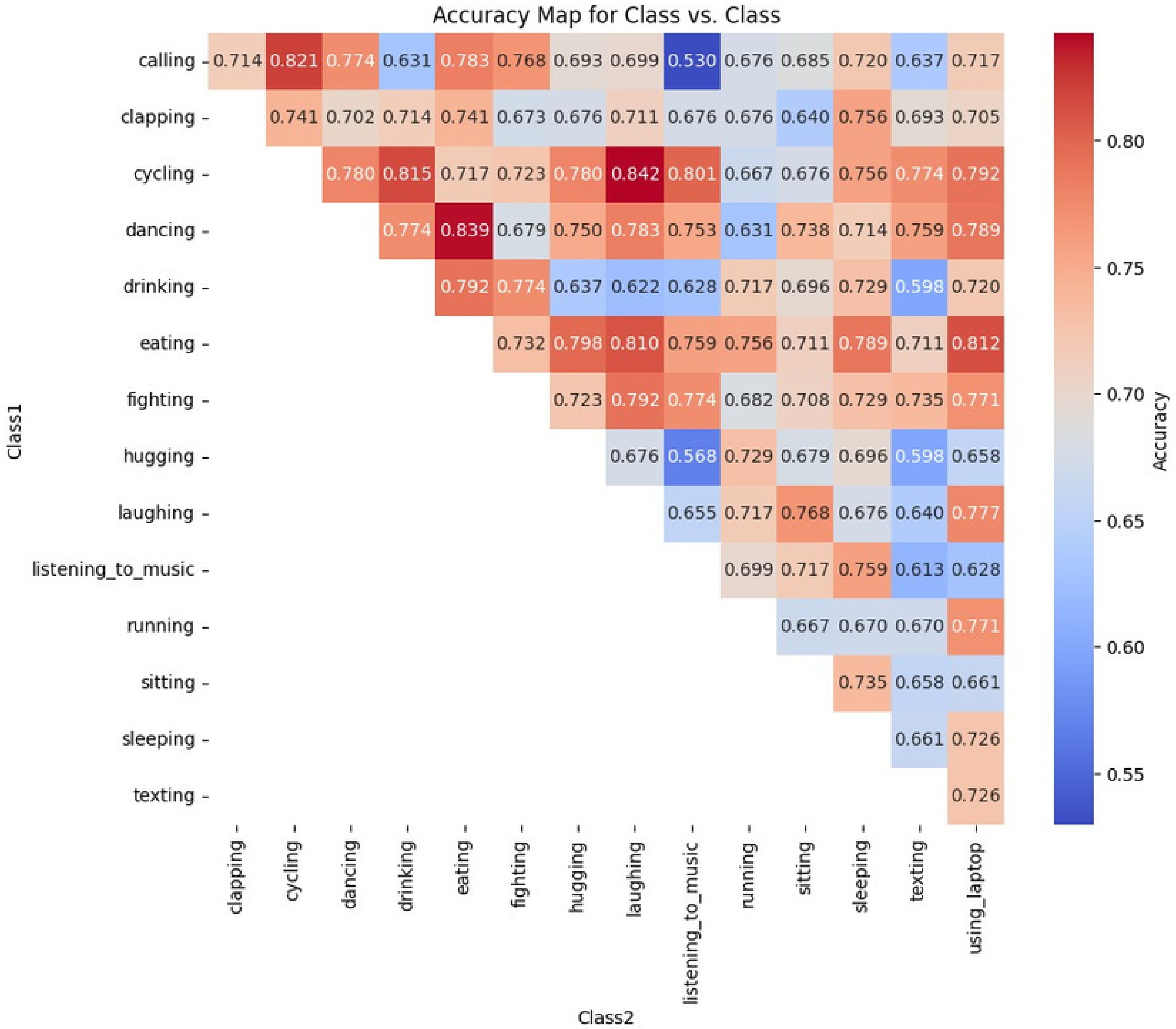
Filter Processing

- As all our images had quite a lot of noise in the background as well as the activities themselves, we inferenced that **entropy in not a filter that should be used**. Also we processed our images through LBP, SOBEL , CANNY and Bilateral etc..
- Among sobel, canny and other filters, we believe it would be **better to implement canny filter** as it implements Gaussian blur before implementing sobel itself



EDA (using filters and feature descriptors)





- Initially we observed that when we plotted before and after plots of an image using Gaussian blur, the noise rate in image textures were very less.
- Interestingly, the accuracy for sobel filter was much better than the one provided by canny filter.
- . HOG was used by us to detect edges and change in intensity to identify gradient variations.
- Similar to HOG, LBP (linear Binary Patters) also helped us to detect edges and variation in lighting from one part of the image to other.

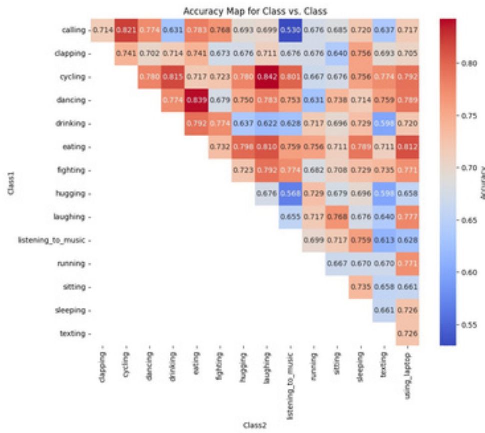
Background Extraction & Concluding Inferences

- we have also used a method called as patch similarity to identify similar patches within an image based on similarity metrics that is mean in our case.
- It helped us to identify regions that are similar or have a high resemblance.
- Through this EDA, we noticed, that out data very varied, and is not well defined by any one feature.
- We also saw that certain classes much easier to classify compared to the other ones. We can see in the heatmaps 3 that classes like running and cycling are much difficult to separate from each other, while those 2 can collectively be separated from other classes relatively easily

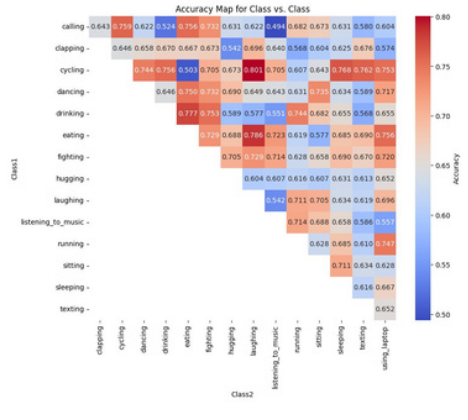
Methodology (Data Pre-processing and model training)

SVM Kernel selection

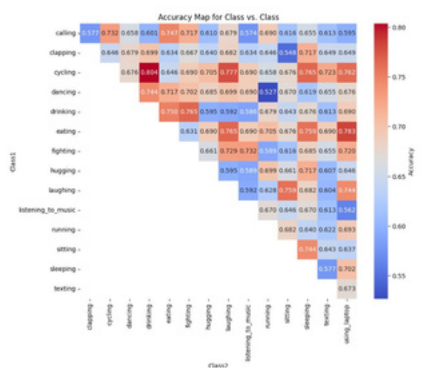
- From our EDA Analysis and its observations, we used multiple filters to test the similarity between the classes which helps us to determine the classes that are easily classified.
- We have plotted heatmaps using random forest classifier to do binary classification between each class after applying kernels such as poly, rbf, Laplacian, sigmoid,chi-square on the features.
- Using these heatmaps and tsne plots, we realized that our data is not separable even in higher dimensions, and decided to use rbf kernel as it was giving the best results



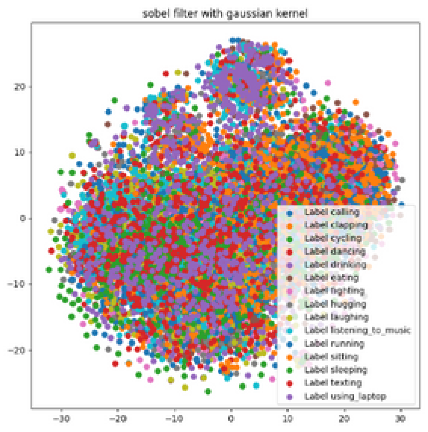
sobel filter+rbf kernel



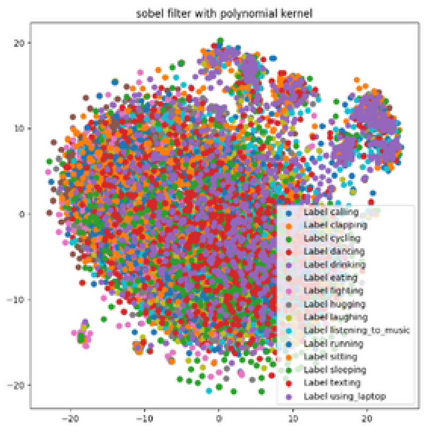
canny filter+rbf kernel



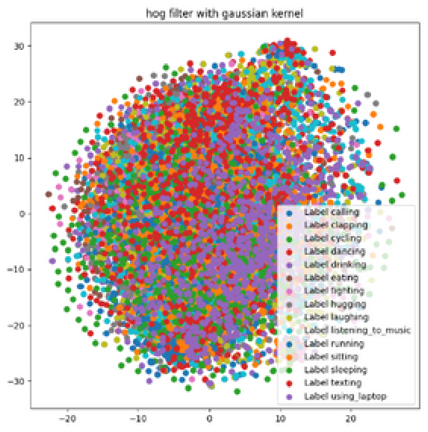
canny filter+sigmoid



(a) sobel+rbf



(b) sobel+poly



(c) hog+tsne

SVM features

- Using the EDA, we had an idea that features like HOG, SIFT were necessary for our data, and decided to add more features using hit and trail method.
- Further, for our model, we have decided to an svm with a rbf kernel. The hyperparameters for it were decided using Grid Search. The Regularization parameter for the svm was taken to be 3, while gamma was set as 'scale' which uses 1/(n_f eatures $\star$ X.var()) as value of gamma

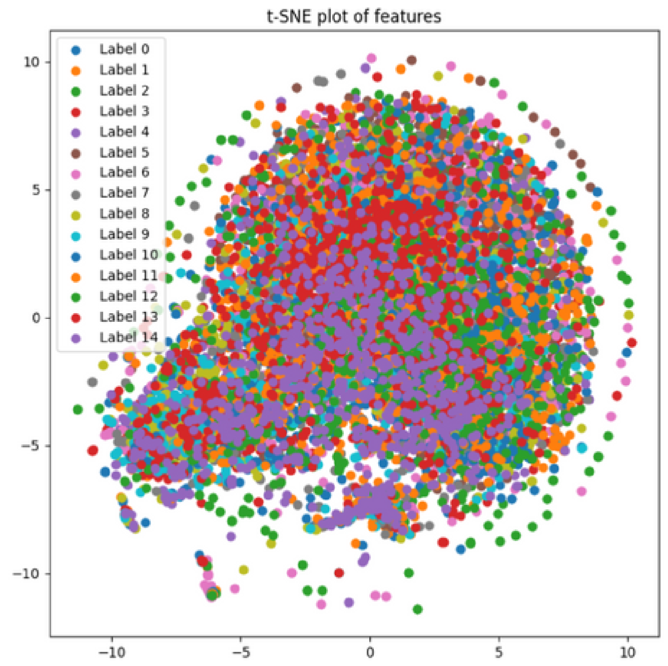
Results

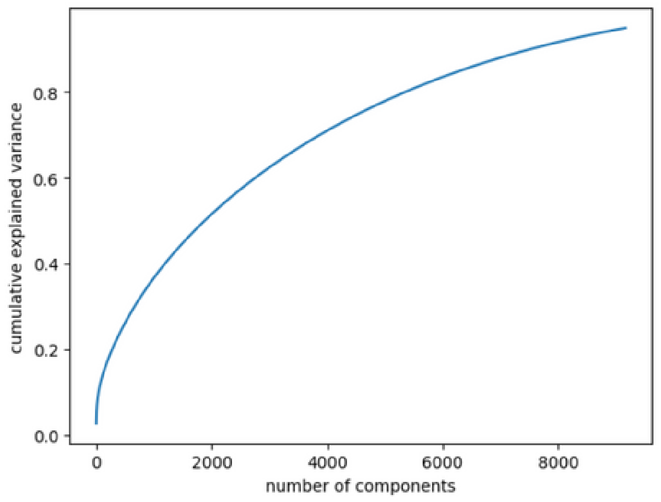
Using purely single features and filters, we acheived the following accuracies.

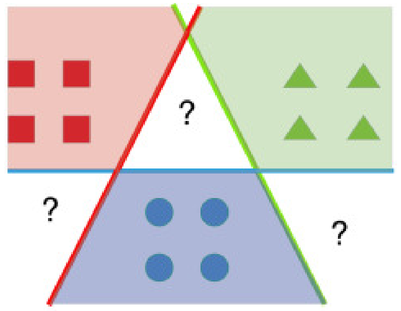
1. Using Histogram of Colors we achieved **25.14%** accuracy
2. Using Sobel Filter we achieved **21.9%** accuracy
3. Using Canny Filter we achieved **15.9%** accuracy
4. Using HOG we achieved **32.1%** accuracy
5. Using Patch Similarity we achieved **32.1%** accuracy

Further, we tried combinations of different features.

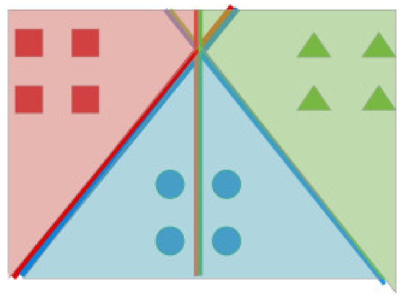
1. Using LBP+HOG we achieved **30.11%** accuracy
2. Using Patch Similarity+LBP we achieved **32.3%** accuracy
3. Using Canny Filter+HOG+LBP we achieved **29.1%** accuracy
4. Using FFT anf FFTshift we achieved **26.3%** accuracy
5. Using HOG+LBP+SIFT+Color Histogram we achieved **35%** accuracy







(a) Separation with OvA.



(b) Separation with OvO.

- We tried to use Gaussian blur, and Bilateral Filter, for noise reduction, but it resulted in worse accuracies
- We also implemented the above models using both 'one vs one' and 'one vs rest' modes of the SVM.and noticed that the different between these versions was minuscule.

Feature Extraction Using Convolution Techniques

- While reviewing some research papers, we learn about some convolution techniques, which can help a lot in feature extraction from image dataset as they extract level by level in images.
- On finding more, we got to known vgg16 and vgg19 feature extractors, which we used for feature extraction from our HAR dataset. On getting features from convolution, we did dimensional reduction using PCA (n_comp=0.95) and then trained our SVM model using gridsearchCV on multiple kernels .

The accuracies are as follows:

- C=1, rbf kernel, scaled gamma: **55%** accuracy
- C=1 linear kerbel, default gamma, **67.5%** accuracy
- C=1, rbf kernel, default gamma, **67.10%** accuracy.

Our aim by using this procedure is to, train a better SVM model, byusing computer vision and CNN techniques only for the purpose of Feature Extraction.

Conclusion

Through this project, we learned about the difficulties in implementing Machine Learning for Computer Vision.We learned about multiple features, kernels, and models that were useful.

Despite our efforts, the achieved accuracy for activity recognition hovers around 35%. This accuracy rate, while indicative of some discrimination capability, falls short of the desired performance level. Furthermore, visualizing the feature space using t-Distributed Stochastic Neighbor Embedding (t-SNE) revealed a disheartening lack of clear separability among different activity classes. This observation raises concerns regarding the efficacy of the selected features in capturing the inherent patterns necessary for robust activity discrimination.
In conclusion, while our study sheds light on the challenges and pitfalls encountered in the realm of human activity recognition from images, it also emphasizes the need for further exploration and innovation in feature extraction and classification methodologies to achieve more accurate and reliable recognition systems in the future.