Human Activity Recognition using Using kernelaized SVM

Akshet Patial[1]* Shivam Kumar[2]* Shreeram Kumar Singh[3]* Raghav Sakhuja[4]* Harshit Giri[5]*
[1]ECE, IIITD [12]CSE, IIITD [3]CSE, IIITD [4]CSAM, IIITD [5]ECE, IIITD
Indraprastha Institute of Information Technology, Delhi
{akshet23155, shivam23004, shreeram23091, raghav21274, harshit21150}@iiitd.ac.in

Abstract

Human Action Recognition (HAR) aims to understand human behavior and assign a label to each action.It has a wide range of applications, and therefore has been attracting increasing attention in the field of computer vision. In this paper we try to recognize of multiple kinds of activities from labelled Images using kernelaized
SVM.

1 Introduction

Human Activity analysis in computer vision involves object detection, tracking and recognition of human activities. Among which,Human activity recognition has a wide range of promising applications in security surveillance. Other than that it has a huge scope in sports as it helps in detecting the human activities which can be done by two ways first by numerical sensor data and other by visuals, former one require a sensor attached to the body but usually require a lot of data to a particular action hence later one has a huge scope nowadays considering recent advances that have been made in digital and mobile phone cameras. Also the unwanted activities like accident, harmful incidents recognition is an important aspect. Action recognition of human group has become a more dynamic area of research work. Based on this topic several approaches have been proposed for activity recognition of human group and detection in surveillance videos. Apart from significant research, action recognition of human group has still a way to go. In this project we plan

[1] .1 Data Overview and Inspection

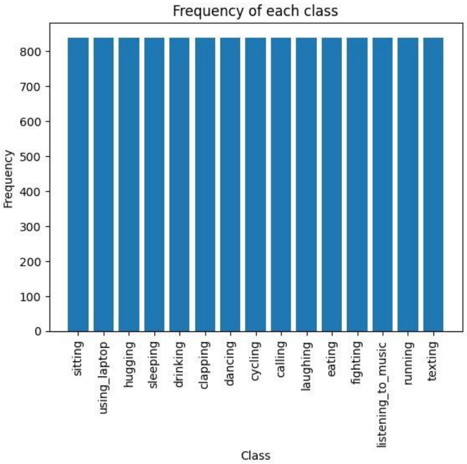
The data being used has a training set of images, each belonging to a particular activity. We are provided with 15 different activities each having 800 images. The activities include calling, clapping, cycling, dancing, drinking, eating, fighting, hugging, laughing, listening to music, running, sitting , sleeping, texting and using laptop. The images provided to us are not of uniform size ranging from 7056p to 50625p in terms of flattened size.

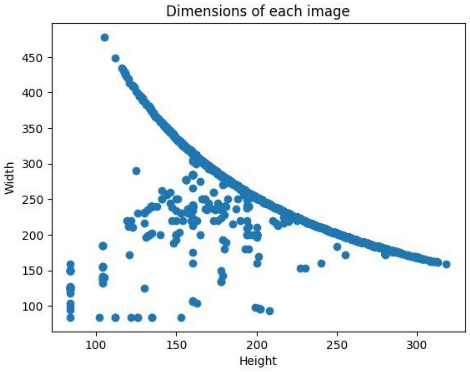
Through this, we notice that the classes are not imbalances and we do not need to unsample or downsample any particular type of images. Further, we need to rescale all the images to a uniform size to get standard sized images. For this we can try and use different dimensions like 64x64, 128x128, or 256x256 by hit and trial.

[2] .2 Possible features and Preprocessing

In this section, we will discuss the methods in which we have decided the importance of different features. We have gone over multiple computer vision features, as well as filters here. In order to tackle this problem. We will try to detect human action from a set of 15 activities using still images.

to find which feature is contribution to the accuracy more, we have plotted multiple graphs and used random forest classifier to do binary classification between each class. The feature importance scores generated by the random forest model enabled me to prioritize and determine which features significantly contribute to the accuracy of the binary classification. This approach facilitates a more comprehensive understanding of the data and aids in feature selection, ultimately enhancing the interpretability and effectiveness of the classification model.



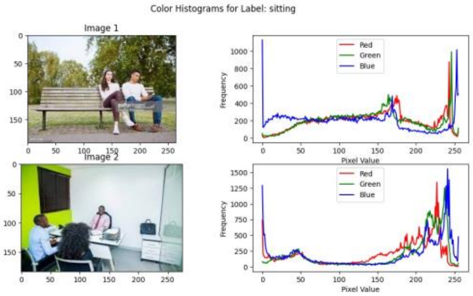


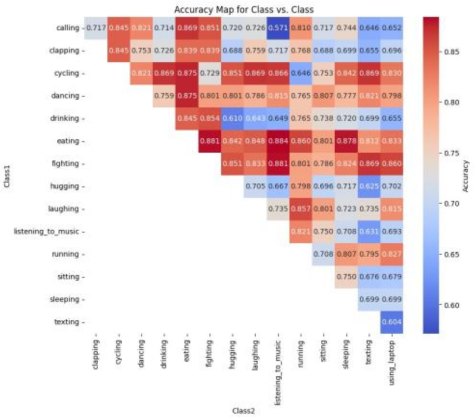
(b) Binary Classification Accuracy using only

(a) color histogram Color Histogram

2.2.1 Colors

In Human activity Recognizition, our initial feeling was that colors might be helpful, but after ploting Color histograms for each class, we notice that the HSV as well as BGR values for all the classes are very similar or very diverse. Our initial assumption was that the importance of color features was not lot.





(b) Binary Classification Accuracy using only

(a) color histogram Color Histogram

But the results were quite surprising as we can see, colors on their do classify certain classes pretty well.

2.2.2 Luminosity and contrast

It was found that the median value of the luminous intensity was ranging between minimum of 80-81 to maximum of 165. Although the interclass median values of light intensity were approximately the same in the plot.

Also the number of outliers were calculated to be 12, which can be considered a major source of error or exceptional cases. Assuming we interpreted 25% of the dataset in the upper quartiles and 75% in Q3 quartiles.

Hence the Parameter of luminosity was contradictory to the assumption that image intensity can be used as a factor for image classification. While luminosity can be a valuable feature in some cases, when it doesn't exhibit significant variation across our HAR dataset, it may not provide much discriminatory power for classification.

The median values of contrast in the HAR dataset are approximately the same, and high and low values are countering the variations in contrast values. The median contrast value ranges between minimum of 55 to maximum of 65, where maximum it suggests that contrast may not be a robust standalone feature for image classification in case.Also the number of outliers were calculated to be 147, which can be considered a major source of error or exceptional cases. Assuming we interpreted25% of the dataset in the upper quartiles and 75%in Q3 quartiles.

2.3 Filters

One of the most common thing while working with image data is to use filters to bring out specific aspects of the image that would help us. We tried to the same by using different filters.

As all our images have quite a lot of noise in the background as well as the activities themselves, entropy in not a filter that should be used.

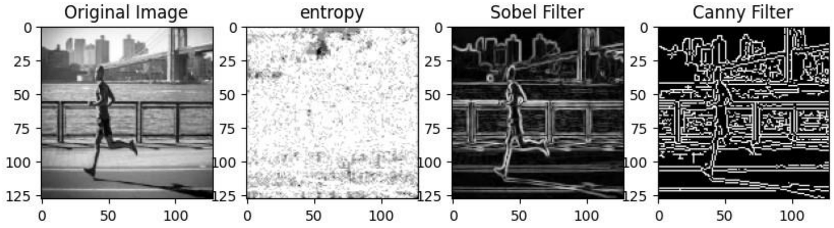
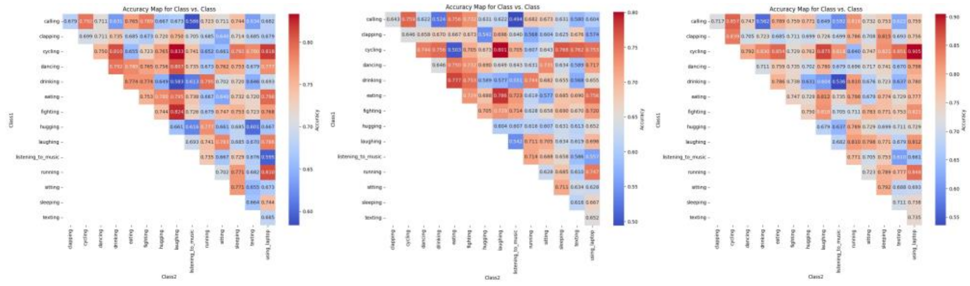


Figure 3: Different filters

Our initial belief was that among sobel and canny filters, we believe it would be better to implement canny filter as it implements Gaussian blur before implementing sobel itself. The use Gaussian blur would be beneficial for us because of high noise that is present in the background.



(a) Sobel (b) Canny (c) HOG

Figure 4: Sobel and Canny

But to our surpise, the accuracy for sobel filter was much better than the one provided by canny filter. This could be due to canny filter overlooking smaller details which were lost in the blur. We also tried out HOG(Histogram of Oriented Gradients). The histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. HOG can be used by us to detect edges and change in intensity to identify gradient variations.

Similar to HOG, LBP (linear Binary Patters) is another form of feature descriptor that works similarly. It is commonly used in computer vision to detect edges and variation in lighting from one part of the image to other.

We also used to the SIFT (Scale-Invariant Feature Transform) algorithm is a computer vision technique used for feature detection and description. It detects distinctive key points or features in an image that are robust to changes in scale, rotation, and affine transformations. SIFT works by identifying keypoints based on their local intensity extrema and computing descriptors that capture the local image information around those keypoints.

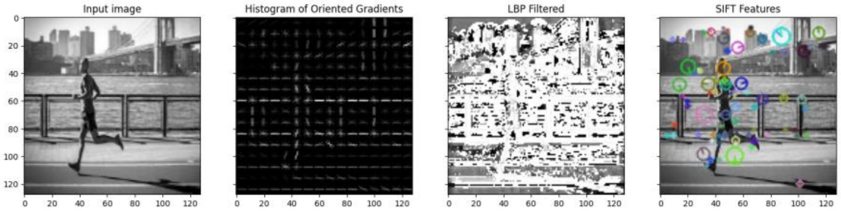


Figure 5: HOG, LBP and SIFT

Apart from above mentioned feature descriptor 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 refers to the comparison of patches within an image to identify regions that are similar or have a high resemblance. The results have been shown in later sections.

2.4 Inferences

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 4 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.

3 Previous Work

Numerous efforts have been made to recognize human activities which involves Human detection, tracking etc. Among which, activity recognition has a wide range of promising applications in security surveillance and sports etc.

Background modelling and subtraction A surveillance video captures the Human in a static background. They are capturing human activity and detect moving objects in an image by subtracting a reference background image from the current image pixel-wise. This is one of the way to detect human from the image other than that there is a non-parameter background model which estimates the probability density function at each pixel using kernel density estimation where the background of the scene is cluttered and not completely static. This is the first step to extract the useful feature from the image.

Feature Selection Local Features: Local features like shape and motion from each frame are getting extracted from the minimum bounding boxes that we have gotten by doing the Background subtraction.

3.1 Challenges in Data Preparation

The Biggest challenge for us seeing the given data set is that all images are entirely different so no two image can have same background therefore background subtraction is not a valid approach to choose. Also above method will completely remove the edges from the background which is valid in this case as they are only focusing on the human subject however in our case the background edge detection is also very important as we have classes which includes activities like calling, cycling, drinking and using laptop etc where capturing edges other than human are important aspect of our feature selection.

3.2 Machine Learning Model and Analysis

In computer vision, person detection and recognition have been dominated by SVM frameworks which surpassed the popular naïve Bayes model. Thus in human activity classification with labelled training samples support vector machines would be superior.

Support vector machines (SVM) classifiers are generally used in distinguishing two classes. But they have been extended to distinguish multi-class by classification by using two popular approaches.

One is to combine a number of two-category classification SVMs in a certain manner to form a multi-class classifier. The other is to directly solve a multi-class classification function with the training samples, former one being more practical. Many algorithms have been derived including the one-against-rest method, the one-against-one method, DAG-SVM, and SVM-BTA etc.

In our case we can make use of these multi class SVM algorithms to train our model initially to find out which one suits better considering our use case.

4 Model and Prepocessing

We resized all images in gray-scale to a uniform size, normalize and flatten them. We tried multiple filters and feature descriptors like Canny, HOG and LBP and decide which of these gives a better results. We did not do any outlier detection, because it does not seem necessary. Also because we could not find any specific outliers by randomly visualising the data or any particular pattern in which that the provided data follows.

We have 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. Accuracies between two classes has been shown below for each kernel.

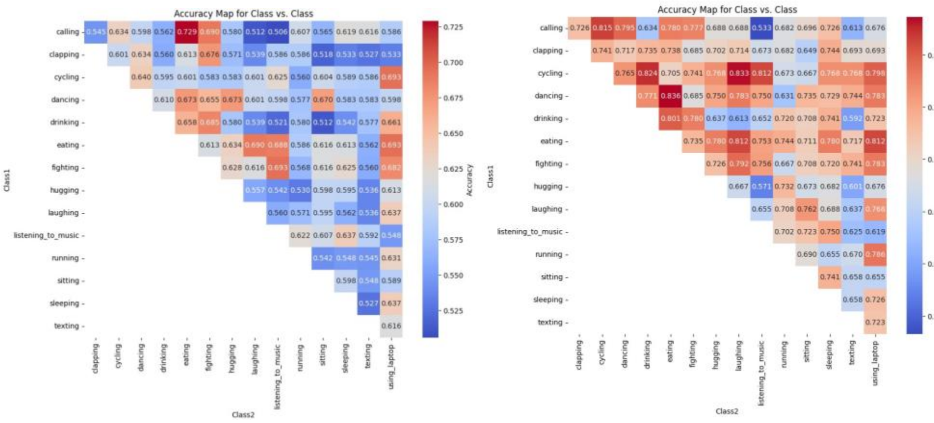
4.1 kernels

As it is evident that our data is not separable, we decide to project our data into higher dimensions using different kernels. For this, we use the following kernels:

RBF kernel: This kernel computes the radial basis function (RBF) between two vectors. This is a widely used kernel having Gaussian distribution. The kernel is defined as:

$$K(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$$

where σ^2 is variance. This kernel is also known as Gaussian kernel.



(a) canny

(b) sobel

Figure 6: Binary Classification Accuracy using rbf kernel

Polynomial Kernel: This kernel computes the degree-d between two vectors. The polynomial kernel represents the similarity between two vectors not only is the same dimension but also across dimensions. The kernel is defined as:

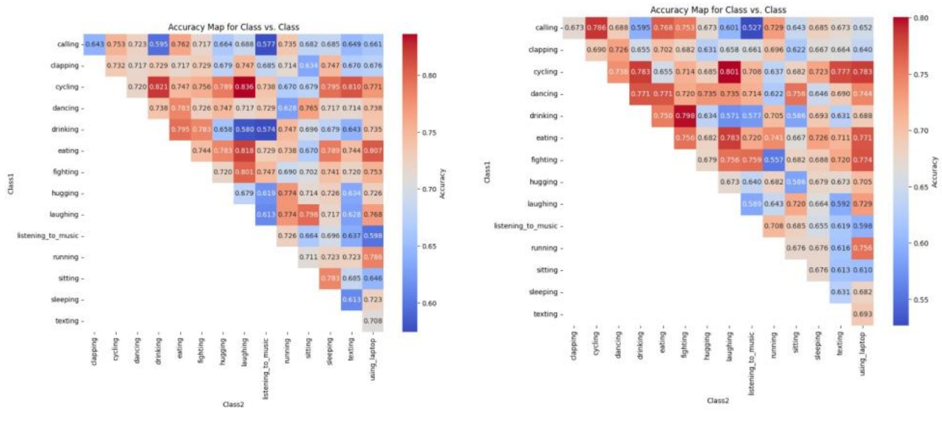
$$K(x,y) = (x^Ty + c)^d$$

where: x, y are the input vectors d is the kernel degree

Laplacian Kernel: Laplacian kernel is a variant of the radial basis function kernel the difference is between the distance, the former uses Manhattan distance which is the absolute difference between the two vectors whereas later one is Norm 2. The kernel is defined as:

$$K(x,y) = \exp\left(-\frac{\|x-y\|}{\sigma}\right)$$

where x and y are the input vectors and $\|x - y\|$ is the Manhattan distance between the input vectors.



(a) polynomial

(b) laplace

Figure 7: Binary Classification Accuracy using Poly and Laplacian kernels

Sigmoid Kernel: The function sigmoid kernel computes the sigmoid kernel between two vectors. The sigmoid kernel is also known as hyperbolic tangent whose value lies between -1 and 1. It is defined as:

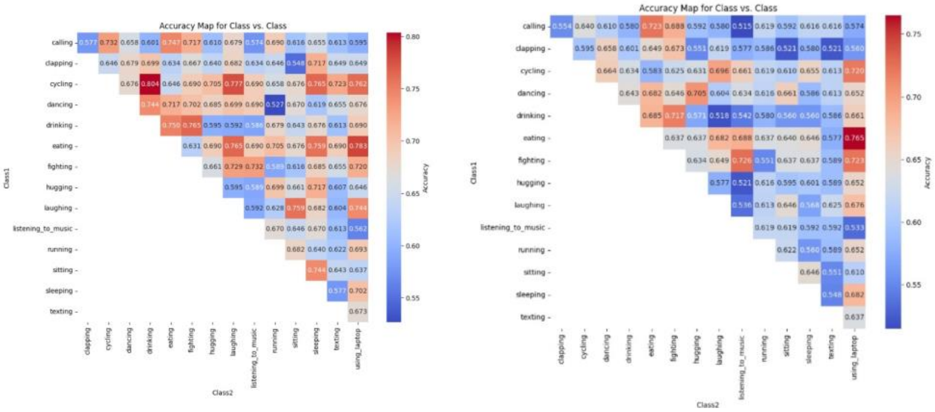
$$K(x,y) = \tanh(\alpha x^T y + c)$$

where: x, y are the input vectors γ is known as slope c is known as intercept chi-Squared

Kernel: The chi squared kernel is given by:

$$K(x, y) = \exp\left(-\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{x_i + y_i}\right)$$

where: x_i and y_i denotes the elements of vectors x and y, respectively.



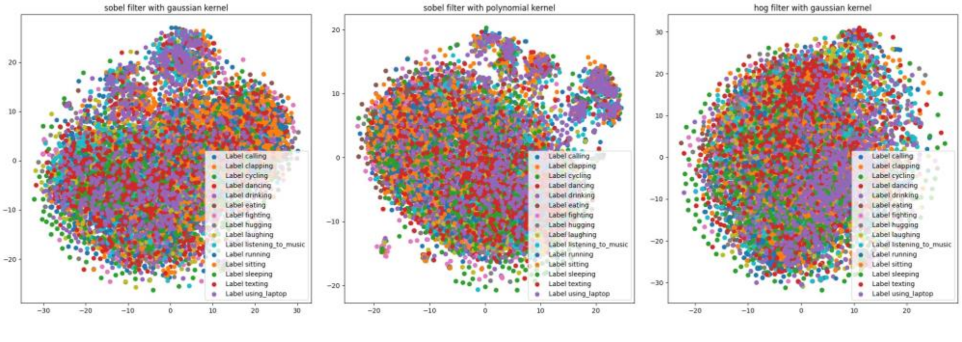
(a) Sigmoid

(b) Chi-sqaured

Figure 8: Binary Classification Accuracy using Sigmoid and Chi-Square kernels

Following are the T-SNE plots

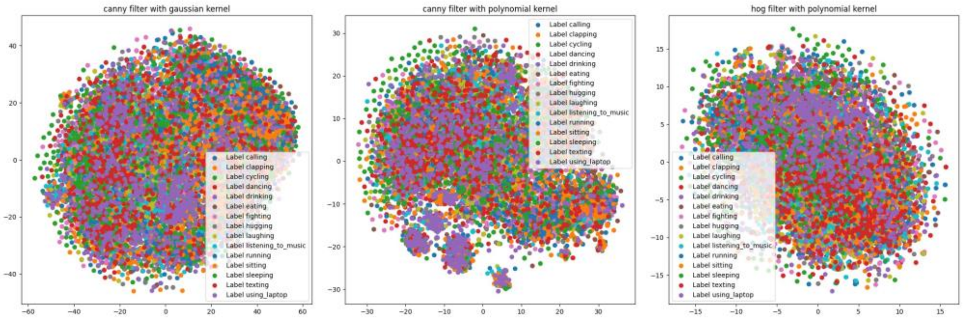
We have used T-SNE (t-distributed Stochastic Neighbor Embedding) for data exploration and visualizing high-dimensional data. It helps to visualize the given dataset into higher dimensions allowing us to understand more about underlying patterns and relationships in the data. We have plotted multiple TSNE plots for different use cases including multiple kernels and edge detection techniques. Some of them are shown below.



(a) sobel+rbf

(b) sobel+poly

(c) hog+tsne



(a) canny+rbf

(b) canny+poly

(c) hog+poly

As it is prominent that the data doesn't seems to be seperable.

4.2 Model description

For the final model, we dicided to use, HOG, LBP, color histogram, and SIFT features. These features are the ones which have the highest accuracy, during out testing.

HOG and LBP are important features, that allows to recognize edges in our images. There features are similar, but provide different aspects of the images. The color histogram allows us to segregate images on the bases of the colors. It easily recognizes classes with specific color compositions, for example the class like cycling is easily separated due its color mainly comprising of outdoor shades. SIFT allows us to find the important keypoints in our image.

We have used PCA and standard scaler on these features. The PCA is a must, as the number of features becomes very large, and the correlation between them increases, as we add more features. We have used features to provide 95% variation of all the features. Along with this we have scaled our data to avoid having too varied of values.

Further, for our model, we have decided to use 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 $1/(n\_features * X.var())$ as value of gamma

4.3 Result and inferences

In this section we shall report the accuracies, and the inferences of different models that we have tried.

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

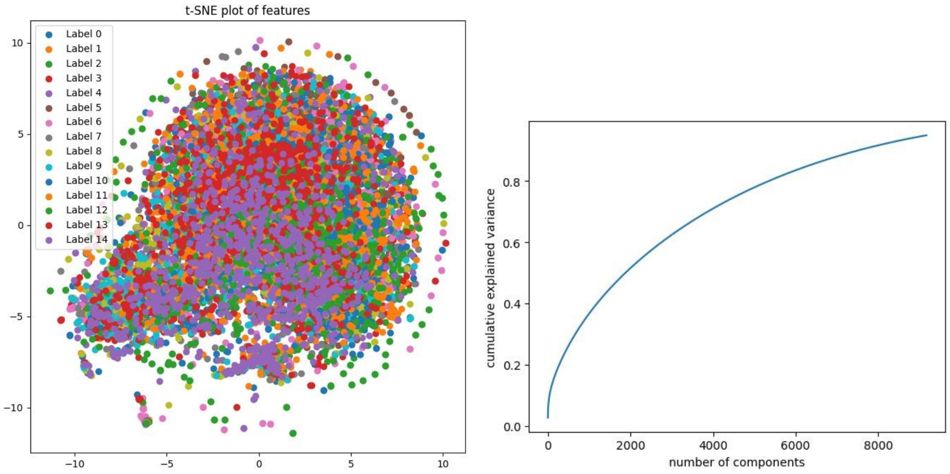
4. Using FFT anf FFTshift we achieved 26.3% accuracy

5. Using HOG+LBP+SIFT+Color Histogram we achieved 35% accuracy

While calculating these accuracies, we tried to use Gaussian blur, and Bilateral Filter, for noise reduction, but it resulted in worse accuracies. This could be due to smoothing of features, which resulted in loss of data.

We also implemented the above models using both ’one vs one’ and ’one vs rest’ modes of the SVM. We noticed that the different between these versions was very minuscule and did not account to much.

The above results have shown as that the different features, do no bring a new dimension to our data, and a lot of information provided by these filters in recurrent. This can be confirmed by the following t-sne plot of the features that provided us the best accuracy. We can see in the t-sne plot that the data is overlapping and not separable at all. This explains the low accuracy that we have achieved. We also notice that out of 31k features that were extracted, only 8000 were actually needed to reach an cummilative variance of 95%.



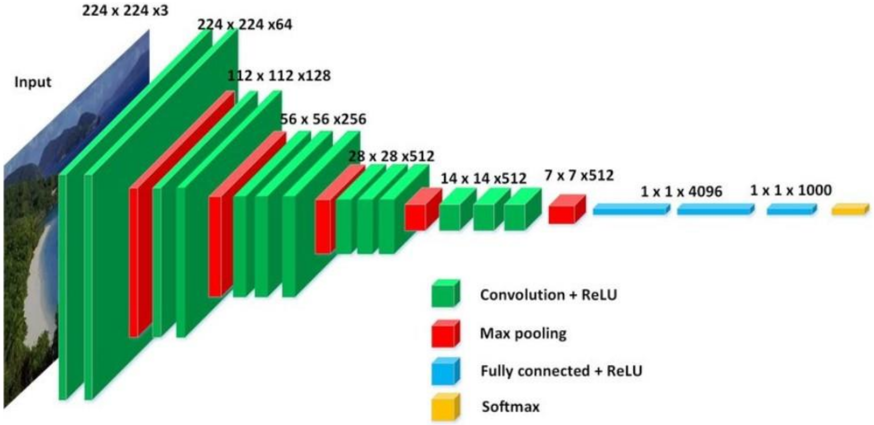
(a) TSNE

(b) PCA

4.3.1 Using Convulation

For Feature extraction purposes, we were not able to get a solid combination of features from our HAR dataset, which could help us in training our SVM model, Therefore while going through some CNN + Kernel SVM image classification based research papers and blogs, we got to know about vgg16.

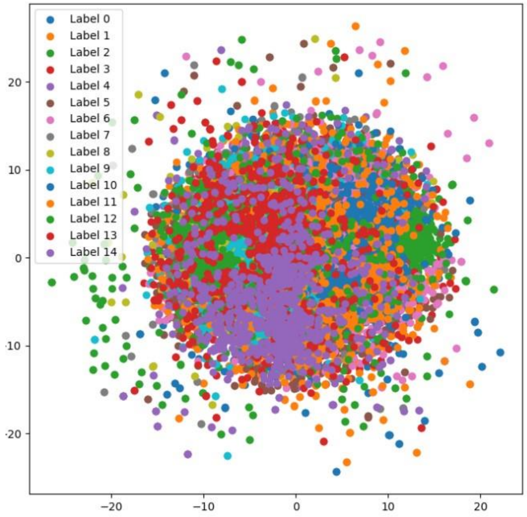
What is vgg 16? VGG16 is a convolutional neural network model that’s used for image feature extraction. It’s unique in that it has only 16 layers that have weights, as opposed to relying on a large number of hyper-parameters. It’s considered one of the best feature extraction computer vision convolution technique.



We only used vgg16 for the purpose of feature extraction, it did'nt played any role in model training or classification. After applying vgg16, we applied PCA on our dataset to reduce dimension of our extracted features. We choosed n_components = 0.95 to preserve 95% variance in our data.

Kernelized SVM model training

After getting suitable amount of extracted features from PCA, we applied GridserachCV on 2 kernels "linear" and "rbf". We tested both on C=1 and 0.1, with default and scaled gamma. The result were favourable with C=1 and default gamma. The accuracies were as follows: 1. C=1, rbf kernel, scaled gamma: 55% accuracy 2. C=1 linear kerbel, default gamma, 67.5% accuracy 3. C=1, rbf kernel, default gamma, 67.10% accuracy. Our aim by using this procedure is to, train a better SVM model, by using computer vision and CNN techniques only for the purpose of Feature Extraction.



We can see in the above figure, that while CNN does not completely separate the data, it does a better job than what out features were doing.

5 Conclusion

In this study, we delved into the challenging domain of human activity recognition using image data. Employing a range of feature extraction techniques including Local Binary Patterns (LBP), Scale-Invariant Feature Transform (SIFT), Histogram of Oriented Gradients (HOG), Color Histogram, and more, we conducted extensive experiments to classify various human activities from images.

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.

6 References:

Huimin Qian *, Yaobin Mao, Wenbo Xiang, Zhiquan Wang Recognition of human activities using SVM multi-class classifier Pattern Recognition Letters 31 (2010) 100–111

Ali, A., Aggarwal, J.K., 2001. Segmentation and recognition of continuous human activity. In: IEEE Workshop on Detection and Recognition of Events in Video, Vancouver, BC, Canada, pp. 28–35

Cheong, S., Oh, S.H., Lee, S.Y., 2004. Support vector machines with binary tree architecture for multi-class classification. Neural Inform. Process.: Lett. Rev. 2 (3), 47–51.

Elgammal, A., Harwood, D., Davis, L., 2000. Non-parametric model for background subtraction. In: ECCV, vol. 1843, pp. 751–767

K. G. Manosha Chathuramali and R. Rodrigo, Faster human activity recognition with SVM, International Conference on Advances in ICT for Emerging Regions (ICTer2012), Colombo, Sri Lanka, 2012, pp. 197-203, doi: 10.1109/ICTer.2012.6421415

Seemanthini Ka, Dr.Manjunath.S.Sb (ICCIDS 2018) Human Detection and Tracking using HOG for Action Recognition International Conference on Computational Intelligence and Data Science

Basly H, Ouarda W, Sayadi FE, Ouni B, Alimi AM. CNN-SVM Learning Approach Based Human Activity Recognition. Image and Signal Processing. 2020 Jun 5;12119:271–81. doi: 10.1007/978-3030-51935-3_29. PMCID: PMC7340932.

Image classification

Classical feature extraction

Image preProcessing

Multi class SVM

VGG16 VGG16