Human Activity Recognition using Kernelized SVMs and CNNs

1st Arjit Singh Arora
Student
B.Tech CSE IIIT-D
arjit21452@iiitd.ac.in

2nd Barneet Singh
Student
M.Tech CSE IIIT-D
barneet23028@iiitd.ac.in

3rd Aman Sharma
Student
B.Tech CSE IIIT-D
aman21010@iiitd.ac.in

***Abstract*—The Human Activity Recognition(HAR) model differentiates the different activities a human performs. It has very wide applications, such as in the field of healthcare to monitor the physical activity of patients, in the field of sports and fitness to analyze the activities of athletes to improve their performance, in security, and many more. This research paper tackles the challenge of classifying human activities utilizing a range of machine learning techniques, like Kernelized SVMs and different architectures of Convolutional Neural Networks (CNNs). In this paper, you can see the pre-processing and EDA , alongwith a comparative analysis of these methods applied to the task of recognizing human activities, using a dataset comprising images representing fifteen distinct categories of everyday life activities.**

I. Introduction

In this study, we work with two essential components of our project: Image data and Associated labels, aiming to classify them from a list of 15 actions. These include sitting, clapping, dancing, using laptop, hugging, sleeping, drinking, calling, cycling, laughing, fighting, eating, listening to music, texting and running.

We have two main resources at our disposal—CSV files, 'training.csv' and 'test.csv,' and the raw image data in the 'training' and 'test' directories.

The project is divided into 2 parts:- using Classical ML based preprocessing techniques (for SVM model) and using Deep Learning approaches (i.e. CNNs) for classification. For the first part a train test split of 75:25 is used, and for the second part, a train-val-test split of 70:15:15.

II. EDA and Data Preprocessing

A. Class Imbalance

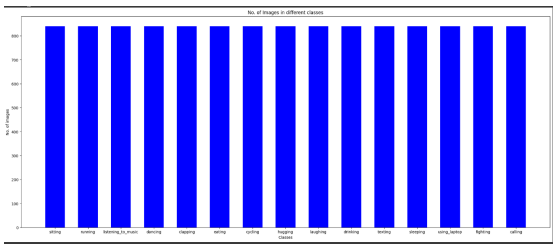

Fig. 1. A Bar Plot depicting class balance.

Class imbalance in machine learning refers to the situation where one class has significantly fewer instances than the others, potentially leading to biased model performance.

All the 15 classes in the HAR dataset have the same number of images, hence, there will be no class imbalance based on the no. of images per class. This will ensure balanced training (which can help the model learn the characteristics of all classes effectively) and reduced bias.

B. Pie chart and Data Distribution

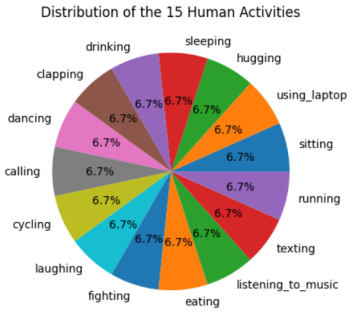

Fig. 2. A Pie chart depicting class distribution.

There are a total of 12600 images in the training folder of the dataset, along with an additional 5400 testing images. Each of the 15 classes in the training set accounts for 6.7% of the total images. This means there are 840 images per class.

C. Histogram plots for all data and per class based (RGB and Grayscale)

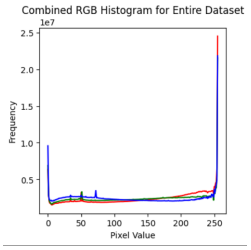

Fig. 3. An RGB Histogram wrt the entire dataset

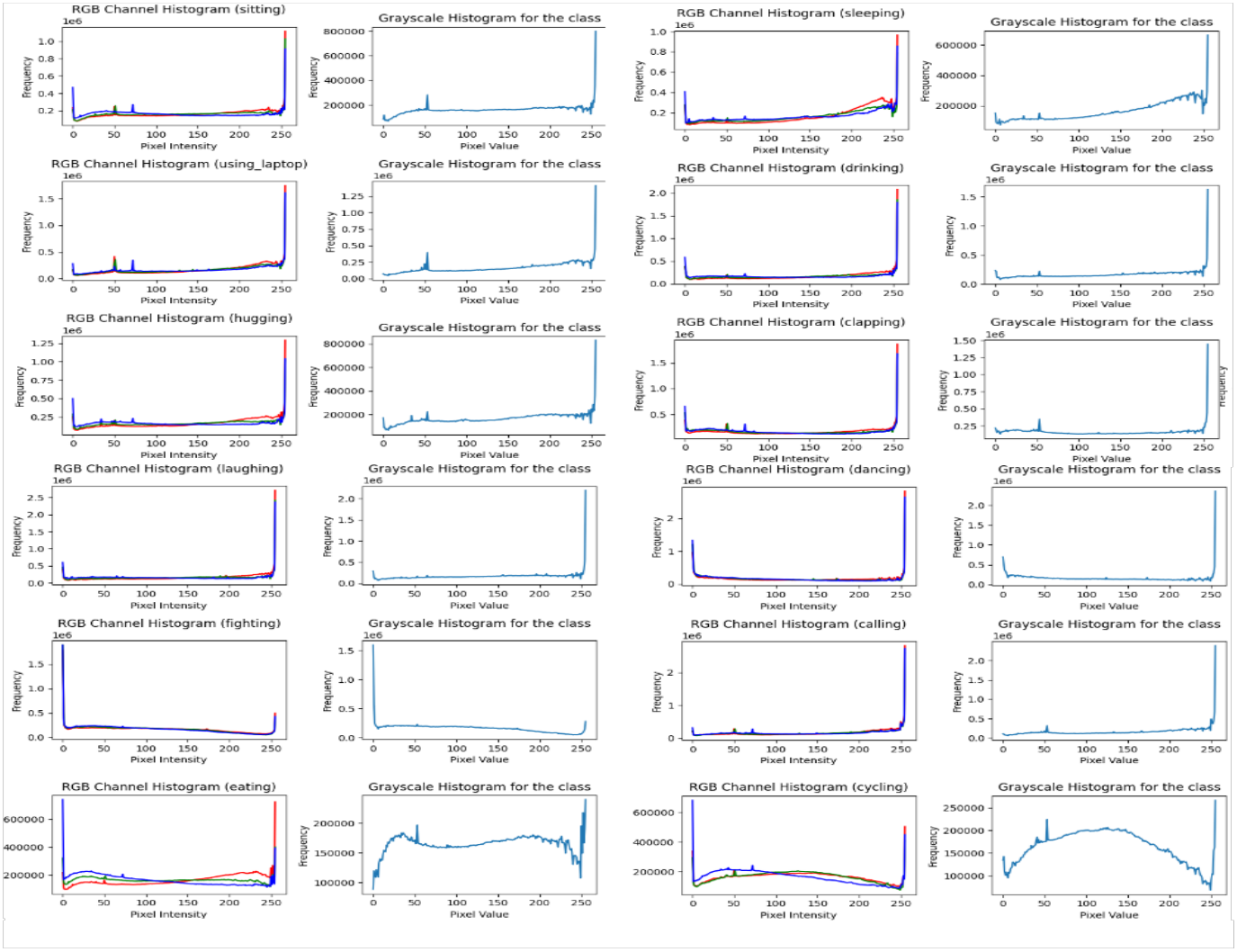

Fig. 4. RGB Histograms for all classes

The dataset clearly doesn't follow a gaussian distribution as evident with strange peaks at the beginning and end (These suggest the presence of ouliers and noise which needs to be removed) . Also on an average the colour distribution is even , evident from the closely grouped peaks and modes.

D. Histogram of oriented Gradients (HOG)

Fig. 5. Gradient features of the image (The silhouette)

Histogram of Oriented Gradients, also known as HOG, is a feature descriptor. It is used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in the localized portion of an image. For the regions of the image, it generates histograms using the magnitude and orientations of the gradient.

E. Edge map using Sobel's filter

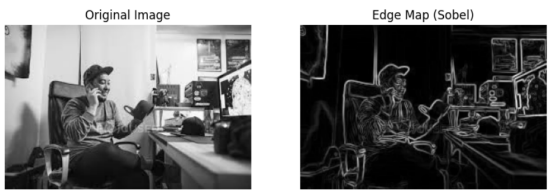

Fig. 6. The Edge map of the image

Sobel's Edge detection filter is used to detect edges in the images. These can be converted to statistical features like

mean, median, etc. to convert into feature vectors.

F. Local Binary Patterns (LBP)

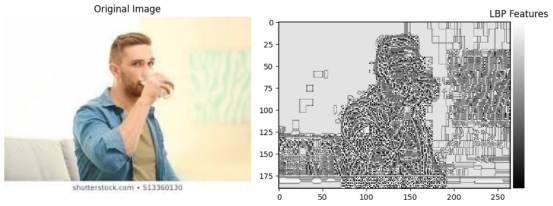

Fig. 7. LBP texture Features of the image

LBPs compute a local representation of texture. This local representation is constructed by comparing each pixel with its surrounding neighborhood of pixels. LBP looks at points surrounding a central point and tests whether the surrounding points are greater than or less than the central point (i.e., gives a binary result).

G. SIFT features

SIFT (Scale-Invariant Feature Transform) features are local descriptors that capture the distinctive appearance of an object at specific interest points in an image. These features are remarkable for their invariance to image scale, rotation, and illumination changes, making them highly robust to variations in image conditions. Additionally, SIFT features exhibit resilience to noise and minor viewpoint alterations, ensuring their effectiveness in a wide range of image-processing tasks.

H. Gaussian and Bilateral Filters

Fig. 8. Effect of bilateral filter on image

A Gaussian filter is used to remove random noise from the image. If applied with a large sigma, it starts to remove edge information beyond removing noise. (i.e. it blurs).

The Bilateral Filter, a good image processing technique, excels in noise reduction while preserving critical image features and edges. It combines spatial proximity and pixel intensity similarity to achieve selective smoothing without compromising edge sharpness. In our case, it worked better than Gaussian for noise because Bilateral filters are often preferred in scenarios where preserving edge details is crucial.

III. Results for Classical ML approaches

Here are the results for classical ML based techniques. The baseline model used is SVMs (experimented with different configurations of kernels and other parameters).

A. HOG + HSV + LBP features

On employing a combination of HOG, HSV, and LBP features. HOG features, extracted from grayscale images, captured shape and texture information. HSV features, extracted from color images, provided color information. LBP features, extracted from both grayscale and color images, captured local texture patterns. These extracted features were concatenated to form a comprehensive feature vector for each image, resulting in approximately 4,500 features per image. This feature matrix was then fed into an SVM classifier with a polynomial kernel of degree 6, leading to optimal classification results. Min-Max scaler boosted the accuracy by 4%

Accuracy on Cross-validation: 35%

B. Bilateral Filter (for noise removal), Sobel filter (for BG separation) then HOG + HSV + LBP features

On removing noise and highlighting the edges of an image would make it easier to extract HOG features, but it actually made the accuracy worse. This might be because the filter removed some of the important edges that HOG needs to work well.]

Accuracy on Cross-validation : 30%

C. Ensemble of HOG, HSV

On combining HOG and HSV features and trained two separate SVMs on different scales of the data. During classification, we calculated the confidence score of each SVM and chose the one with the higher confidence for prediction.

Accuracy on Cross-validation : 31%

D. SIFT Features

Integrating SIFT, HSV, and LBP features augmented our image analysis pipeline, amalgamating distinctive key points, color nuances from HSV, and local texture patterns from LBP. This fusion enriched the model's comprehension, enabling a more detailed and nuanced interpretation of the visual data and enhancing its ability to discern intricate details, colors, and texture variations within the images. Accuracy can be improved if we consider some more features of SIFT, as for now we only took the first row of each feature extracted, which is 1 x 128.

Accuracy on Cross-validation : 25%

E. HSV+LBP

The fusion of HSV (Hue, Saturation, Value) color space with LBP (Local Binary Patterns) descriptors creates a powerful feature extraction technique for image analysis. HSV captures rich color information, while LBP encodes local texture patterns. This fusion enables a comprehensive image representation, facilitating nuanced analysis and

interpretation. The combination leverages both color nuances from HSV and fine-grained texture details from LBP, providing a holistic understanding of the visual content. This fusion finds applications in a wide range of image-processing tasks, including object recognition, image segmentation, and content-based image retrieval.
Accuracy on Cross-validation : 30%

F. Performance Evaluation of Various Kernels for Optimal Model Selection

- Linear Kernel: 28%
- RBF Kernel (Gaussian): 34%
- Polynomial Kernel (Deg-6): 35%
- Sigmoid Kernel: 14%

G. Other Techniques

- To address the dimensionality issue, we augmented the number of HOG features and employed mean and variance calculations to reduce the feature space. This approach proved effective as PCA, our initial choice, failed to yield the desired results.
- Z-score Normalization

IV. Feedback and Issues

Noise in the dataset:
Traditional feature extraction methods involve handcrafted operations on the image. While these methods (filters and thresholds) may capture certain local features, they do not perform noise removal in a learned and adaptive way as they are manual. Ties in the dataset added to the woes.

Imperfect feature extraction:
Handcrafted techniques like HOG, HSV, and LBP have limited capacity to represent and generalize complex and high-dimensional patterns inherent in human activities.

Thus, traditional ML techniques and non-deep learning-based feature engineering do not lead to high accuracy in this task. Existing analysis also showed a majority of DL techniques (like OpenPose, CNNs, and RNNs) being used to solve this problem. CNNs are capable of learning hierarchical representations of data and reducing noise inherently due to multiple layers.

V. DL based techniques

Here the baseline model used is CNN. Different architectures have been used including some pretrained models that were fine tuned on the dataset.

A. Custom CNN based architecture (using modified version of INCEPTION BLOCKS)

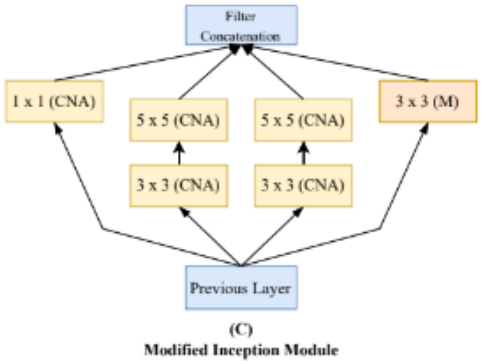

Fig. 9. Modified Inception block (CNA stands for Convolution , Batch Normalization and Activation)

We used 6 such inception blocks followed by a global avg pooling at the end and then a fully connected layer for final classification. After each inception block we did a maxpool having stride=2 (for downsampling the image) . The no. of channels were doubled after each block starting from 32 and all the way upto 512. The Activation function used was ReLU followed by a Cross Entropy Loss.

The following were the parameters used:-

- **Batch Size:** 32
- **No. of Epochs:** 40
- **Learning Rate:** 0.001
- **Image Size:** 160x160
- **Optimizer:** Adam

Accuracy on Cross-validation : 56%

B. Pre-Trained VGG -16

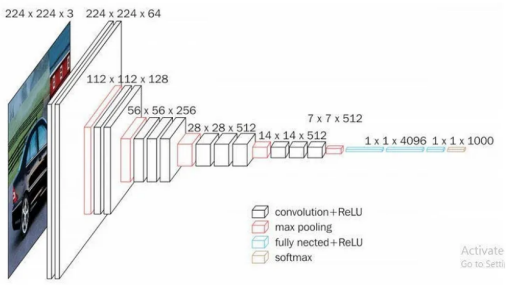

Fig. 10. VGG-16

VGG-16 is a widely recognized convolutional neural network architecture featuring 16 layers, including 13

convolutional layers and 3 fully connected layers. It utilizes small 3x3 convolutional filters throughout the network, enabling it to learn complex image features effectively. Despite its simplicity, VGG-16 achieves strong performance in image classification tasks and serves as a foundational model in deep learning research and applications.

C. Pre-Trained Efficient Net B7

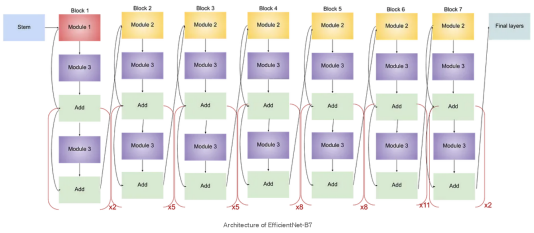

Fig. 11. Efficient Net B7 having ImageNet weights

EfficientNet B7 is a part of the EfficientNet family, renowned for its compound scaling strategy, which systematically adjusts model depth, width, and resolution to achieve optimal performance. As one of the largest variants, B7 boasts deeper, wider, and higher-resolution architecture compared to its counterparts. This design allows it to capture intricate features from images efficiently. By leveraging pre-trained weights from datasets like ImageNet, EfficientNet B7 accelerates training and ensures robust performance across various computer vision applications, making it an invaluable tool in the field.

TABLE I
ACCURACY OF DEEP LEARNING BASED TECHNIQUES

Model Architecture	Accuracy
Modified Inception Net	56%
VGG-16	53%
EfficientNet B7	70%

VI. CONCLUSION

In this paper the research of different machine learning methods used to recognize human activities has been performed. In the first part a lot of different EDA techniques from classical machine learning have been used for feature extraction with the basline model as SVM. Then in the second part different architectures of CNNs were employed to achieve the result. This paper provides the comparison study of the mentioned methods for human activity recognition.
The average accuracy of image classification using SVMs is lower and approaches 30%. The application of different CNN architectures has revealed higher accuracy results, although EfficientNet B7 has reached around 70% average accuracy, which indicates the best score of all applied methods. Considering the obtained results, further studies are needed to analyze the eligibility of different and newly created CNN architectures for the solution of image-based human activity classification problem.