Human Activity Recognition Using Kernelized SVM

Milind Jain
2021165

Aman Chauhan
MT23015

Anay Chauhan
2021013

Akash Sharma
MT23012

Dushyant
2020198

Abstract

This report offers a comprehensive analysis of the Human Activity Recognition (HAR) dataset, pivotal for applications in diverse fields. We present an in-depth exploratory data analysis (EDA), highlighting key characteristics and nuances of the dataset. The report then progresses to examine various Support Vector Machine (SVM) techniques, including the application of Kernelized SVM with Radial Basis Function (RBF) kernel, tailored to HAR. Insights from the EDA underpin our methodology in model development, where we extensively test and compare models employing RBF kernel SVM. The outcomes of these models, documented in terms of prediction accuracy and performance metrics, provide valuable contributions to the field of HAR, demonstrating the efficacy of advanced SVM techniques in practical applications.

1 Introduction

Human Activity Recognition (HAR) has seen growing interest due to its relevance in health monitoring and video surveillance applications. Support Vector Machines (SVMs), particularly when enhanced with kernel methods, have demonstrated potential in various classification challenges, including HAR. Kernelized SVMs capture complex patterns in data, making them apt for image-based HAR tasks. In this report, we embark on an exploratory journey of a HAR dataset, emphasizing its unique characteristics and the potential of SVM techniques in its context.

2 Dataset Overview

The HAR dataset is a rich collection of labelled images representing various human activities. A deep dive into the dataset revealed the following characteristics:

- **Activities:** The dataset comprises 15 unique activities.
- **Data Distribution:** Each activity has exactly 840 data points, leading to 12,600 labelled images.
- **Challenges:** Several issues were identified, including imbalanced classes, noisy data, and missing data points.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

3 Exploratory Data Analysis (EDA)

3.1 Unique Activities

The dataset encapsulates a spectrum of human activities. The 15 unique activities include walking, running, jumping, and several others. Each activity is observed precisely 840 times, even distribution. (Figure 1)

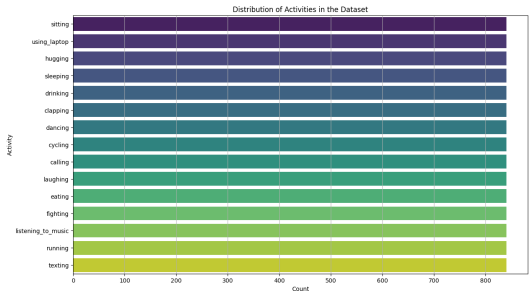

Figure 1: Distribution of Activities

3.2 Image Dimensions Distribution

Analysis of image dimensions showed most images with a resolution of 260x197 pixels, indicating a standardized data collection approach.

Variability: The standard deviation is around 39.92 pixels (width) and 35.28 pixels (height), indicating some variability in image sizes.

Minimum Dimensions: The smallest image size is 84x84 pixels.

Maximum Dimensions: The most prominent image size is 478x318 pixels.

(Figure 2, Figure 3 and Figure 4)

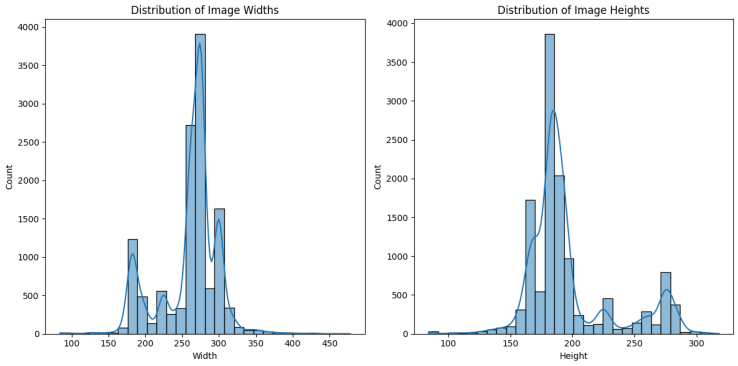

Figure 2: Image Dimensions

3.3 Brightness Distribution

The brightness distribution of images was studied, revealing insights about the various lighting conditions under which the data was collected.

3.3.1 Brightness Range:

The brightness values of all the images fall within the range of 0 to 250. This indicates that there is a diverse set of brightness levels present in the dataset.

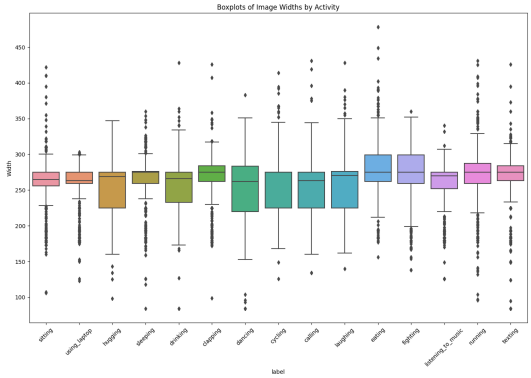

Figure 3: Box Plot (width)

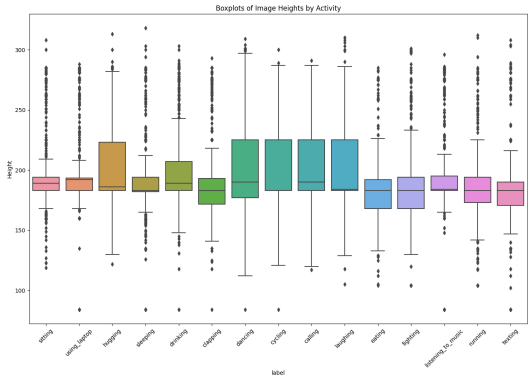

Figure 4: Box Plot (height)

3.3.2 Concentration:

Most images have brightness values concentrated in the range of 110 to 160. This concentration suggests that most of the dataset contains images with similar or moderate brightness levels.

3.3.3 Distribution:

The statement that the brightness data follows a normal Gaussian curve suggests that the distribution of brightness values across the images is approximately normal. This means that there is a balance between images with lower and higher brightness levels, with the majority falling in the middle range.

(Figure 5)

3.4 Color Distribution

Colors significantly influence our understanding of images, especially in the context of activity recognition. Through Exploratory Data Analysis (EDA), we've mapped out the color landscape of our dataset:

General RGB Distribution: This gave us an overview of the common colors across all activities. Such broad insights hint at recurring themes or biases in the dataset.

Activity-specific RGB Distributions: By breaking down color patterns for each activity, we wanted to (if any) observe unique color signatures. For instance, water-based activities lean towards blue hues, while activities in green spaces exhibit more greens.

Gray Intensity Distributions: These helped understand the image's brightness and contrast, offering clues about lighting conditions and possibly the time of day the activity occurred.

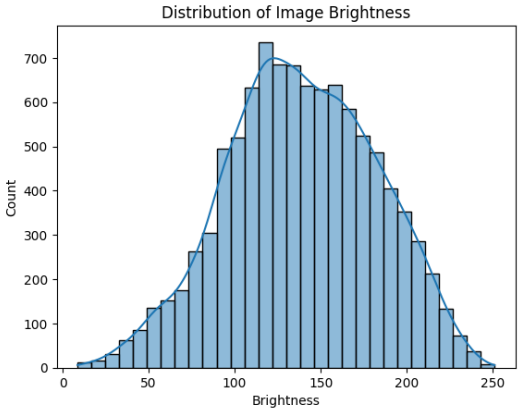

Figure 5: Image Brightness Distribution

In essence, our color analysis has provided a nuanced understanding of the dataset, enhancing our approach towards activity recognition.

(Figure 6, Figure 7 and Figure 8)

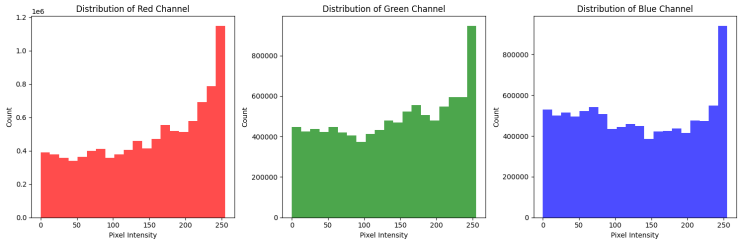

Figure 6: General Color Distribution

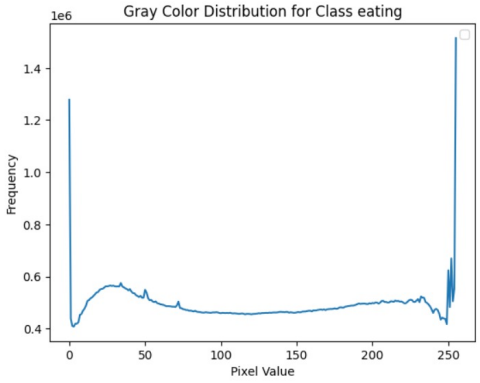

Figure 7: Gray Color Distribution (activity-wise)

3.5 Image Segmentation

Image segmentation involves converting an image into a collection of regions of pixels that are represented by a mask or a labeled image. By dividing an image into segments, you can process only the important segments of the image instead of processing the entire image.

Image segmentation is a commonly used technique in digital image processing and analysis to partition an image into multiple parts or regions, often based on the characteristics of the pixels in the image. Image segmentation could involve separating foreground from background, or clustering

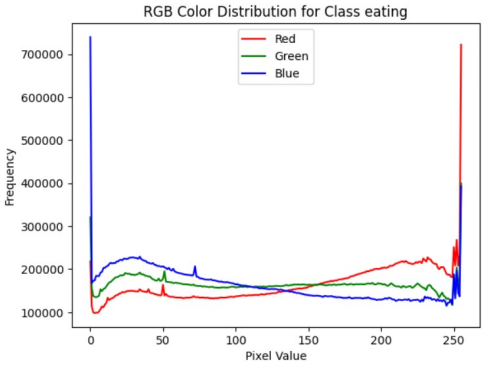

Figure 8: RGB Color Distribution (activity-wise)

regions of pixels based on similarities in color or shape. For example, a common application of image segmentation in medical imaging is to detect and label pixels in an image or voxels of a 3D volume that represent a tumor in a patient's brain or other organs.

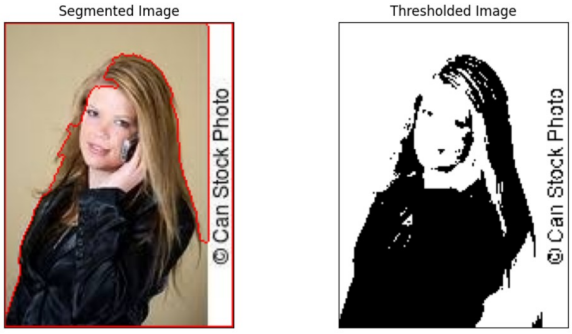

Figure 9: Image Segmentation

4 Existing Analysis of the HAR Dataset

4.1 SVM Techniques Explored

The HAR dataset has been a subject of interest for several researchers. Some of the techniques previously applied to this dataset include:

- **C-Support Vector Machine (C-SVM):** This SVM model is used for distinguishing between two classes in binary classification. Adapted for multi-class classification in HAR using methods like "One-vs-All" or "One-vs-One."
- **Nu-Support Vector Machine (Nu-SVM):** Another SVM model for binary classification in HAR, allowing adjustment between the margin and the model's number of support vectors.
- **One-Class SVM:** Employed for anomaly detection in HAR.
- **Kernel SVM:** Uses kernel functions to capture non-linear patterns in image data.
- **Multi-Class SVM:** Extends binary SVM models to handle more than two classes.
- **SVM with Different Kernels:** Experiments with various kernel functions for different datasets.

4.2 Image Processing

Digital image processing entails the manipulation and analysis of images using computer algorithms to enhance pictorial information for clearer understanding. This area requires extensive experimental

work to validate proposed solutions. The Computer Vision System focuses on recognizing objects of interest from images, aiming to develop machines capable of visual functions akin to human vision. It encompasses image acquisition, pre-processing, segmentation, feature extraction, and presentation or classification.

4.3 SVM in Image Classification

A comprehensive explanation is provided on how SVMs can be utilized for image classification, delving into aspects such as kernel functions, margin maximization, and support vectors. The image dataset underwent various preparation steps, including resizing, normalization, and data augmentation.

5 Future Work

The current analysis represents an initial exploration of the Human Activity Recognition (HAR) dataset, shedding light on its characteristics and the potential of kernelized Support Vector Machines (SVMs) in classification tasks. As we move forward, several avenues for future work emerge.

5.1 Kernelized SVM for HAR

One promising direction is to delve deeper into the application of kernelized SVMs for HAR. While we've provided an overview of SVMs, there is room for more comprehensive experimentation with various kernel functions, including Radial Basis Function (RBF), polynomial, and sigmoid kernels. Tuning hyperparameters, such as the kernel bandwidth and regularization parameter, can further improve the model's performance. We plan to systematically assess the impact of different kernels on the accuracy of HAR predictions.

5.2 Performance Metrics

In our initial analysis, we have primarily focused on accuracy as the performance metric. However, HAR applications often require a more nuanced evaluation. Future work will involve the incorporation of additional metrics such as precision, recall, and F1-score. These metrics provide a more comprehensive understanding of model performance, especially when dealing with imbalanced datasets or when certain activity classes are of particular interest.

6 Models Used

We created multiple models and tested them out on the HAR dataset to check which model was best suited and gave the highest accuracy. We used different features for the RBF kernel SVM that we used. The RBF (Radial Basis Function) kernel is a popular choice for Support Vector Machine (SVM) algorithms, particularly in the context of classification tasks. SVMs are a type of supervised machine learning algorithm used for both classification and regression. The RBF kernel, also known as the Gaussian kernel, is a kernel function that allows SVMs to handle non-linear decision boundaries. The

$$K(X_1, X_2) = \exp(-\frac{\|X_1 - X_2\|^2}{2\sigma^2})$$

Figure 10: RBF kernel equation

RBF kernel allows SVMs to model complex, non-linear decision boundaries in the input space. It achieves this by implicitly mapping the input data into a higher-dimensional feature space where the decision boundary can be a hyperplane.

The various feature extraction techniques (for RBF kernel SVM) that we used are mentioned in detail below.

6.1 Basic Greyscale Pixel Feature Extraction

The first model that we tried out used basic greyscale pixel feature extraction. Extracting features from greyscale pixels involves identifying relevant patterns, textures, or structures in an image that can be used for various tasks such as image classification, object detection, or image segmentation. Feature extraction is crucial because it helps reduce the dimensionality of the data while preserving essential information. Following are some of the techniques we used for feature extraction from greyscale pixels:

- Pixel Intensity Values
- Texture Analysis
- Image Gradients

Using this model for training the dataset gave us an accuracy of **22.53%** on the test dataset.

6.2 HOG Feature Extraction

HOG stands for Histogram of Oriented Gradients. It is a feature descriptor widely used in computer vision and image processing for object detection. It captures information about the distribution of intensity gradients or edge directions in an image. HOG is particularly popular in pedestrian detection and other object recognition tasks. Therefore, it could be beneficial to use HOG to create a model for a Human Action Recognition Dataset. Here's a basic overview of how HOG feature extraction works:

- **Gradient Computation:** Compute the gradient of the image to capture the intensity variations and edges. Common gradient operators include the Sobel or Scharr operators.
- **Gradient Orientation and Magnitude:** For each pixel, compute the gradient magnitude and orientation. The magnitude represents the strength of the gradient, while the orientation indicates the direction.
- **Cell Division:** Divide the image into small cells (e.g., 8x8 pixels). The gradient information within each cell is used to construct histograms.
- **Histograms:** For each cell, create a histogram of gradient orientations. The histogram bins represent different orientation ranges, and the bin values correspond to the frequency of gradients within those ranges.
- **Block Normalization:** Group adjacent cells into blocks (e.g., 2x2 cells). Normalize the histograms within each block to enhance the robustness to lighting variations and contrast.
- **Descriptor Concatenation:** Concatenate the normalized histograms from all the blocks to form the final HOG feature vector for the entire image.

The resulting HOG feature vector was then used as a representation of the image, capturing the local gradient information in different regions. This feature vector was employed as input to the machine learning algorithms which we used, support vector machines (SVMs) in this case, in order to make predictions on the HAR dataset given to us.

Using this model for training the dataset gave us an accuracy of **25%** on the test dataset.

6.3 Scale-Invariant Feature Transform with HOG

Scale-Invariant Feature Transform is another feature extraction algorithm that is used for image processing and is a fundamental tool for various applications such as object recognition. Following are the key aspects Scale-Invariant Feature Transform (SIFT):

- Scale-Invariance
- Key Point Detection
- Scale Space Representation
- Gradient Computation
- Key Point Localization

- Orientation Assignment
- Descriptor Generation
- Matching

A major advantage of SIFT is its robustness and invariance property. Using SIFT (Scale-Invariant Feature Transform) along with HOG (Histogram of Oriented Gradients) can provide complementary information for certain image recogintion tasks. Both SIFT and HOG are feature extraction techniques that capture different aspects of the visual content in images. A few benefits of using HOG along with SIFT are as follows:

- **Robustness to Varied Environments:**SIFT and HOG are both designed to be robust to certain types of variations in images. SIFT's scale invariance and HOG's ability to capture local texture information make them suitable for handling diverse environments and object appearances.
- **Redundancy Reduction:**While SIFT and HOG capture different aspects, their combination can sometimes help reduce redundancy. For example, SIFT keypoints may be concentrated in areas of high local contrast, while HOG can provide a smoother representation of gradients across larger regions.
- **Improved Discriminative Power:**SIFT and HOG focus on different aspects of an image (local features vs. global structure). By combining them, you can potentially enhance the discriminative power of the feature representation. This can be particularly beneficial in tasks like object recognition and image classification.

Although SIFT has a lot of benefits and is a great tool for computer vision as well as image recognition tasks, it has its own disadvantages and shortcomings. This includes sensitivity to changes in viewpoint and the computational cost associated with its key point detection and descriptor generation processes. Using this model for training the dataset gave us an accuracy of **27.06%** on the test dataset.

6.4 Local Binary Pattern with HOG and SIFT

LBP, or Local Binary Pattern, is a texture descriptor used in computer vision for image analysis and pattern recognition. LBP is particularly effective in capturing texture patterns and is often used in tasks such as facial recognition, texture classification, and human action recognition. LBP operates by comparing the intensity of a pixel with the intensity of its neighbors in a local neighborhood. It encodes this comparison result into a binary pattern. The central pixel is assigned a value of 1 if its intensity is greater than or equal to the neighbor's intensity and 0 otherwise. The binary values are then concatenated to form a binary number, representing the local texture pattern. Using LBP in addition to HOG and SIFT can prove to be beneficial in the following ways:

- **Robustness to Different Types of Features:**LBP is robust to changes in illumination and is effective in capturing fine-grained texture details. On the other hand, HOG is robust to variations in object appearance and can capture the overall shape and structure of objects. Combining LBP and HOG features can make the representation more robust across different types of variations in the data.
- **Image-based Classification and Detection:**For image-based classification and object detection tasks, the combination of LBP and HOG features allows the classifier to capture both micro-patterns and macro-patterns in the images. This can be crucial for achieving good performance in tasks like pedestrian detection, face recognition, and scene understanding.
- **Human Action Recognition:**In human action recognition, where actions involve both local motion patterns and global movement characteristics, the combination of LBP and HOG features can be beneficial. LBP can capture local motion details, while HOG can provide information about the overall structure of human poses and movements. SVM can then classify these combined features for recognizing different actions.
- **Effective for Object Recognition:**In object recognition tasks, especially for recognizing objects with distinct texture patterns and shapes, the combination of LBP and HOG features can improve the recognition accuracy. The SVM classifier, trained on this combined feature set, can learn to discriminate between different object classes based on both local texture details and global shape information.

Using this model for training the dataset gave us an accuracy of **28.29%** on the test dataset.

6.5 Using convolution for feature extraction

Convolutional operations involve sliding a small filter (also known as a kernel) over the input data (an image in this case) and computing the dot product of the filter and the local region of the input. This operation is repeated across the entire input to produce a feature map. When we use convolution for feature extraction, we're essentially applying this operation to extract important patterns or features from the input data. These features could represent edges, textures, or more complex structures, depending on the depth and complexity of the convolutional layers.

Simply using convulation for feature extraction is quite different from using a full fledged CNN (which we weren't allowed to use in this case).

Convolutional feature extraction is crucial for human action recognition in image datasets. The steps involved are briefly explained below:

- **Local Pattern Detection:**Convolutional layers are adept at detecting local patterns and features in images. In the context of human action recognition, these patterns could represent distinctive poses, body parts, or motion-related features. Different filters in the convolutional layers can learn to detect various spatial and temporal aspects of human actions.
- **Temporal Information:**For human action recognition, temporal information is crucial. Convolutional layers can be extended into 3D convolutional layers or combined with recurrent layers to capture temporal dependencies in sequences of images. Temporal information helps the model understand the dynamic aspects of actions over time.
- **Reduced Parameter Sharing:**Convolutional layers use parameter sharing, which reduces the number of parameters compared to fully connected layers. This is especially important when dealing with image datasets, where the input size can be large. Reduced parameter sharing makes the model more efficient and helps prevent overfitting.

Using convulation for feature extraction and then applying RBF Kernel SVM on it greatly increases the accuracy of the model giving an accuracy of **64%** on the test dataset.

7 Conclusion

The HAR dataset is a treasure trove of information, and understanding its nuances is crucial for any modeling attempt. Through our EDA, we gained important insights that proved to be instrumental while applying Kernelized SVM for activity recognition. We tested out 5 models mainly and compared the prediction results (Accuracy scores) to the prediction results of what we got by applying CNN. As expected, the accuracy of CNN was greater than the rest of the models. However, since we weren't allowed to use CNN, the next best result was when we simply used convulation for feature extraction instead of a full fledged CNN model. The other models gave much lower accuracies with the highest accuracy being when we applied LBP, HOG and SIFT before applying RBF kernel SVM.

References

1. `https://www.researchgate.net/publication/351023539_Human_Action_Recognition_Using_CNN-SVM_Model`
2. `https://ieeexplore.ieee.org/document/7490795`