Feature Engineering for Motion Classification in Machine Vision

Soumya Shaw¹*, Susan Elias¹ and Sudha Velusamy²

¹ Centre for Advanced Data Science, Vellore Institute of Technology, Chennai, Tamil Nadu, India
² Samsung Research India, Karnataka, India

* soumyashaw.official@gmail.com

Abstract. With the most advanced classification algorithms in the technological platform, the computational power requirement is on the surge. The paper hereby presents computationally trivial algorithms to simplify the process of computational intensive classifications techniques, especially in the Motion Classification arena. The proposed methods prove crucial in acting as a lightweight and computationally fast stepping stone to a fundamentally more significant application of Motion indexing and classification, Action recognition, and predictive analysis of motion energy. The algorithms classify the motions into linear, circular, or periodic motion types by following an appropriate execution order. They consider the tracked motion path of the object of interest as a sequence and use it as a starting point to perform all operations, resulting in a feature that can be classified into separate classes. Using a single parameter for classifying the motion engenders a faster and relatively more straightforward route to motion identification and elicits the algorithm’s uniqueness. Two algorithms are proposed, namely, Angle Derivative Technique and Determinant Method for classifying the motion into two classes (linear & circular). On the other hand, a different algorithm identifies periodic motion using the principle of correlation on the motion sequences. All the algorithms show an average accuracy of over 95%. It also elicited an average processing time of 15.6 ms and 19.86 ms for Angle Derivative Method and Determinant Method, respectively, and 31.2 ms for periodic motion on Intel(R) Core(TM) i3-5005U CPU @ 2.00 GHz and 8GB RAM. A dataset of camera-captured videos consisting of three motion types is used for testing while the proposed methods are trained on a dataset of motion described by mathematical equations with added $3\sigma$ noise levels.

1. Introduction

Machine vision has emerged into a field of advanced technological breakthroughs over the period and aided other fields in optimizing their operation procedures. Be it the flamboyant technology of Tesla’s automated driving system (ADS) [1] [2] or object detection for surveillance, the technology has entered every arena and is proving to be essential to ensure reliable operations. The implementation of the systems commonly uses features and their combinations to perform their specific tasks. The features can individually be a single value or a multidimensional vector based on the application. A feature can be a descriptive segment of information crucial in solving a machine vision problem and is chosen by general understanding, keeping in mind the specific applicability to the problem [3]. Also, the feature must provide a quantitative descriptor, which can be compared to other such cases.
The use of motion classification is yet an unraveled field, and its potential is still underestimated because of its limited usage. Most of the work in this direction is intended to discern Human motion patterns based on key features and their predictive occurrences and consequently connected to the individual’s health. Any other field of application is yet to commence, and this is where this study steps in to introduce broader applicability and, on the other hand, reduce this complicated process into a fundamental approach. The field’s dormant progress, lack of a standard dataset, and hidden potential make it evident why it needs to be in the technological mainstream. Studying an object’s motion gives a new picture of its state and gives an idea of the hidden pattern it is executing. Knowing or predicting a change in motion type can be essential in many industrial and physical tasks. A sudden change in motion type can help detect an anomaly that can prove to be disastrous later. The study of motion patterns can also be thought of as a subset of anomaly detection applications. For instance, a drone entrusted with the task of circling a specific region can be declared an anomaly if it starts to execute a different type of unexpected motion. Motion classification is inherently essential for tasks like Video classification and action recognition and serves as an underlying prerequisite to achieving optimal results. Videos are filled with independent objects and motions, yet collectively putting a tag on it (classifying) needs the knowledge of the individual sub-motions. Tracking individual features to conclude the overall scene may come in handy for the tasks mentioned above. The classification of motion is also applicable in the field of visual surveillance.

The application of Motion indexing runs in parallel to the other uses mentioned [4]. Whenever a live feed or video input is provided, it is compelling to have some features to identify specific desired segments uniquely. Knowing about the motion classes with timestamps will also ensure its reliable indexing for most target cases. A sequel to the proposed work is to use the motion classified output, the density of moving points intertwined with the frame rate of the video, to predict the extent of motion energy associated with the video or the object. Although not too much in use, industrial applications can focus on such algorithms where well-defined behaviors and repetitions are expected, and motion monitoring can serve as a verification mechanism. Therefore, starting from the base, the paper classifies the motions into three buckets, linear, circular, and oscillatory/periodic, as mentioned in Figure 1.

![Figure 1. Target Motion Classes](image_url)

Although the complex solutions seem to be more attractive for some instances, the trivial solutions are always more insightful and easy to visualize, which the study focuses on as it enters a new application domain for the case. Besides, the trivial solutions will also enforce broader applicability and may even work for domains this paper does not specify. Therefore, this paper tries to propose trivial methods for Motion Classification and their applications, aiming for a considerable reduction in complexity and not compromising on accuracy; this algorithm is well suited for lightweight solutions. A lightweight alternative not only makes it more effective in real-time scenarios but also makes it much more affordable.
To brief about the dataset used, the models are trained with an augmented dataset explicitly prepared for this purpose. The augmented dataset is loaded with axial sequences of their respective motion types obtained from the mathematical descriptors of the former. The equation of straight lines is used with varying coefficients of variables depicting the variety of conditions to emulate linear motion. A similar approach is taken to augment the motions of objects in the circular domain. In addition to that, the sequences are merged with noise up to a margin of $3\sigma$ to augment the dataset closer to real-life conditions. However, the testing dataset consists of camera-captured videos of objects in motion thoroughly explained in upcoming sections. The testing gave a candid view of the algorithm’s performance and applicability.

2. Related Works

While the application of Motion Classification is still pristine, the usage of features mentioned in the last section is already in use for various purposes. The feature of the Histogram of oriented gradients (HOG) is suggested for the application of pedestrian motion detection [5], face recognition [6], and tracking purposes [7]. Another class of local descriptors known as Scale Invariant Feature Transform (SIFT) [8] [9] are in prominent use along with the variants of HOG and SIFT together [10], which can be jointly termed as HoG (Histogram of gradient). These local descriptors are histograms representing the gradient’s orientation in the specific image region, centered at a point. Another algorithm called Histogram of Optical flow and Magnitude (HOFM) is used to detect anomalous events in videos [11]. Other authors have made use of multiple features combining spatial (averaged SIFT) and temporal statistics [12]. Dong et al. [13] were successful in achieving a multi-view extension of HoG with identical memory and run-time complexity compared to its single view equivalent. The distribution of intensity gradients and applied masks as features for classification has helped these traditional algorithms be very applicable for the mentioned tasks and continue to pave the way for future developments. Similarly, Aristidou et al. [14] used 13 independent body motion features to classify the actions into six emotional states using the Laban Movement Analysis (LMA) dataset. Chao et al. [15] implemented the cluster using a set of orthogonal spherical harmonic functions. Peter et al. [16] pointed towards a revved-up computer vision use in the industrial sector in automating workflows. One of the examples mentioned is the use of Automatic Optical Inspection (AOI) in the flat panel display (FPD) industry. They, in turn, explain the importance of numerical features obtained from images and the determination of category suing learning algorithms based on the features.

Researchers have already started exploiting the Deep Learning techniques to classify motion. Gihyum Han et al. [17] classified the motions of Pedestrian walking behaviours using Support vector machine (SVM). Motion analysis and classification-based principal component analysis (PCA) and Extreme Learning Machine Classifier (ELMC), a feed-forward neural network, have also been implemented [18]. Deep Neural networks (DNNs) have also facilitated the use of meta-features to be learned automatically [19]. However, the learned features are not invariably interpretable, and DNNs render ineffective learning for various other application domains [20] as well. Zhang et al. [21] also mentioned the added complexity and involved procedures of using Recurrent Neural Networks (RNN) with counter-intuitive models. This study also believes the hype can be countered considering that Deep Learning techniques require extensive datasets to be at par with others, not to forget the usage of GPUs for the long training process as well. The architecture can be overwhelming, sometimes truly shrouding the intuitive working of the classifier while troubleshooting. Techniques like PCA stretch down to several dimensions to tackle the classification task revving up the computational burden on the way.
3. Proposed Methods

3.1. Angle Derivative Technique

A motion in computational terms can be thought of as a sequence of discrete points sampled in time, which yields a set of two coordinates (at each point) in the \(XY\) plane. Let us take a typical example of a linear motion directed in the top-right direction as shown in Figure 2, the dots representing the sampling points capturing the motion.

### Table 1. Axial Sequences of the Linear Motion

| \(t_1\) | \(t_2\) | \(t_3\) | \(t_4\) | \(t_5\) |
|---|---|---|---|---|
| \(X\) | 2 | 3 | 4 | 5 | 6 |
| \(Y\) | 1 | 3 | 5 | 7 | 9 |

The same motion, when translated to corresponding sequences, will be identical to Table 1. Do notice that the \(X\) coordinate and \(Y\) coordinate are stored as separate sequences.

![Figure 2. Example of Linear Motion](image)

Each point individually gives information about its spatial domain and concurrently as a sequence tells about the Spatio-temporal relationship and direction of motion wrt time, which can be generalized in the form shown below as Equation 1 and Equation 2, and consequently to classify the motion type.

\[
X = x_1, x_2, x_3, x_4, \ldots
\]

\[
Y = y_1, y_2, y_3, y_4, \ldots
\]

The angle derivative method considers every two consecutive points from the motion-tracked sequence, say \((x_1, y_1)\) & \((x_2, y_2)\) and finds the angle between them, using Equation 3, taking \(X\)-axis as the reference. The equation, in turn, uses the resultant sequence to conclude the motion’s direction and property.

\[
\theta_i = \tan^{-1} \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \tan^{-1} \frac{Y_{\text{diff}}}{X_{\text{diff}}}
\]
The approach presumably needs the 2-D tracked path for segmented motion using one of the many algorithms available for use. Some of the options available for the purpose are Background subtraction [22], Temporal differencing [23], and Optical flow [24] that can be effectively used for extracting the motion. The topic of motion Segmentation in itself is a separate topic worth considering for extracting the motion coordinates but not the main focus of the paper. However, it is imperative to cite that this study has employed the use of Optical flow to extract the motion of the specific object of interest, also denoted by Figure 3. Optical flow gave an intriguing option to choose which point to track in a moving object instead of considering the whole object at a time. This feature gives additional freedom to the user to choose a particular object with a mouse click from a case where many independent objects may elicit different categories of motion in a single frame.

Figure 3. Optical flow: The red circle represents the point from where the tracking started and the green dot represents the current position of the tracking point

Figure 4. Theory of Motion Types. (a) Angle projected by linear motion (b) Angle projected by Circular motion

A linear motion corresponds to a motion that progresses in a single direction when viewed spatially. It is visually elicited in Figure 4a, where the constant angle is evident. Application of the discussed angle derivative technique engenders a resultant sequence of constant numerical value (the constant angle it makes wrt to X-axis), which in turn, is distinguishable from a resultant sequence of circular motion where the numerical value is variable shown in Figure 4b and sweeps across the whole \((-180^\circ, 180^\circ]\) range as shown in Figure 5.
Standard Deviation is a statistical tool to determine the degree of the sparseness of the data this study is dealing with, and in this case, it is the resulting sequence of the Angle Derivative Method. Standard Deviation in its mathematical form is calculated by Equation 4, where $\mu$ stands for the mean and $N$ for the length of the sequence.

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$  (4)

The Standard Deviation of the resultant sequence suffices the need for a discernible parameter between the two classes essential for the algorithm. This is because of the contrasting difference in the Standard Deviation in both the motion types, owing to the sparseness of the sequence of angles. While ideal cases align perfectly with the algorithms, the real-world data, known to be predominantly noisy, needs additional corrections. Applying a convolution of the sequence with the unit step function, shown in Equation 5 alleviates the problem of classification misnomer to a great extent, maintaining the characteristics of circular motion as well, demonstrated in Figure 6.

$$u[n] = 1, \text{ where } 0 \leq n \leq 20$$  (5)

A particular case in which the angle derivative resultant sequence elicits a motion in the direction of $180^\circ$, with the slightest of noise will yield a direction that will flicker, let’s say, between $-170^\circ$ and $170^\circ$, engendering a high standard deviation motion type and pointing to a circular motion even after convolutional smoothing. The condition is tackled by reversing both the sequences before angle calculation. Among the two standard deviation values obtained from the versions of the sequence (original and reversed), the lower value is chosen as the parameter for comparing the threshold. This motion reverse feature proves valid because a circular motion, even after reversing, will sweep across the same range of angles identical to the original version.
Figure 6. Convolutional Smoothing

\[
\frac{SD_{\text{max}} + SD_{\text{min}}}{2}
\]  

(6)

Figure 7 demonstrates the spread of Standard Deviation over the number line and paves the way for obtaining a classification threshold. As mentioned before, the dataset formed by mathematical definitions of motion types is used to test the spread of the standard deviation values over the number line. The demarcation of the two classes is shown by the usage of different colors on the graph. Summing up the process, the motion captured into X and Y coordinates is converted into a sequence of discrete directions indicated by angles, and then motion reversed to obtain two sequences for a single motion. The standard deviation values are found for each sequence consequently. The standard Deviation with a smaller magnitude is taken as the sequence’s output parameter value and is subject to thresholding. The threshold is calculated by Equation 6, where \(SD_{\text{max}}\) corresponds to the linear motion type’s maximum Standard Deviation value. In contrast, \(SD_{\text{min}}\) refers to the minimum Standard Deviation value of circular motion type. The dataset used in this study gives a threshold of 21.1397 with \(SD_{\text{max}} = 17.2732\) and \(SD_{\text{min}} = 25.0062\). Although the method used here uses manual threshold calculation, this part can be automated by an ML training process that gives a robust threshold value based upon the specific training dataset.

Figure 7. Distribution of Standard Deviation
3.2. Determinant Method

As it sounds prescient, the determinant method leverages the mathematics of determinants to obtain a unique and reliable technique to classify motion types. The approach also obligates the use of a 2-D tracked path for segmented motion using any of the previously stated algorithms (i.e., Optical flow).

This specific algorithm requires concatenating the 2-D coordinate sequence obtained from tracking the object and its transformation into a square matrix. The final form of the sequence is indicated by $S_{\text{final}} = \hat{X}\hat{Y}$, where the operator denotes the concatenation operation. Nonetheless, the initial concatenated sequence must be made into a series of lengths closest to the ‘perfect square’ to facilitate the mentioned transform, as it is previously known that Determinants can only be calculated for square matrices. Obtaining the determinant of the square matrices from the previous step provides us with the key-value pairs used to identify the motion type, and traditional Machine Learning (ML) algorithms can be applied further.

The determinant method provides us with a probable feature that can be exploited to classify the motion. However, the task is not over yet. Each sequence outputs a determinant value that must be mapped to a specific motion class (i.e., linear and circular). In theory, a sequence with a determinant value as zero should be classified as linear and a non-zero determinant value as circular. To be precise about the concept, the Determinant method is used as a dimensionality reduction algorithm, and further mapping of the Determinants needs to be carried out using the ML algorithms. While there are numerous ML algorithms to choose from, this paper decided on five well known and prudent algorithms to test the viability, and they are:

- Logistic Regression
- K-Nearest Neighbor
- Decision Tree
- Random Forest
- Support Vector Machine

Square matrices and determinants elicit two distinctive properties mentioned by Sharma [25], which can be exploited in this case to obtain a plausible pattern. It is summarized as:

**Property 1:** If any two rows (columns) of a square matrix $A = [a_{ij}]$ of order $n (> 2)$ are identical, then its determinant is zero i.e. $A = 0$.

**Property 2:** Let $A = [a_{ij}]$ be a square matrix of order $n$, and let $B$ be the matrix obtained from $A$ by multiplying each element of a row (column) of $A$ by a scalar $k$, then $B = kA$.

An object in linear motion in any direction follows a stipulated progression in the $XY$ plane, captured by the coordinates. For a strict $X$ or $Y$ axis movement, either of the coordinates will remain a constant, depicting the direction. When compiled into a matrix $A$, it will trigger the Property 1 mentioned, to give a determinant value of zero. Alternatively, a linear motion not in the former domain will be in a proportional increase in constant ratio when projected on both axes. The constant progression ratio, jointly supported by Property 1 and Property 2, will result in a condition where $B = kA$, given $A = 0$ from the previous result, leading to a zero determinant value again. While this algorithm is ideally flawless, the noisy real-life tracked paths provide a determinant value minute in magnitude ($|A| \approx 0$), easily discernible from circular paths, which give higher determinants.
The determinant values are immensely diminutive and sparse in their natural domain. Furthermore, the values dramatically differ in exponent as well as shown in Figure 8. So, it is reasonable to perform logarithmic conversion to restrain the sparseness of the determinant values and engender computational simplification. The effectiveness of this adjustment also reflects the accuracy mentioned in the Results section.

![Figure 8. Distribution of Determinant Values](image)

3.3. Correlation-based periodic motion classifier

Periodic motion can be defined as a recurrent motion in which the intervals of time required to complete each cycle are equal. The motion of a pendulum, the motion of a spring, the vibration of a guitar string, and the Earth’s movement around the Sun are some examples of periodic motion [26]. Repetitive/Periodic motion is a particularly very pivotal feature when the scientific arena is discussed. It is a strong prompt for the tasks like activity recognition and motion classification because of its unique and cyclic nature. Mathematically, it can be stated as:

Let the position \( \vec{r} = \vec{r}(t) \) be a function of time. Then, it is said that \( \vec{r} \) is a periodic function such that \( \vec{r}(t) = \vec{r}(t + T) \) if there exists \( T > 0 \), for any value of \( t \) where \( T_{\text{min}} \) delineates the period of the function.

Conversely, the motion classification applications require a much trivial understanding of the background and usage of the axes for description. This type of motion can be uniquely identified by a pattern repeated in X-axis, Y-axis, or both. Two crucial assumptions are included before proceeding to the proposed method:

- It is presumed that the frame rate is rapid enough to record the periodic motion (at least double the highest frequency of the periodic motion owing to the Nyquist rate).

- It is assumes that the position and apparent scale of the segmented objects do not change substantially over time (or only on a periodic basis).

At a later stage, we will discuss the implications of the violations in the above stated assumptions [26]. It is also imperative to remember that the concept of projections to represent any given complex motion is used. Representing directional motion vectors can be challenging for computation and resorting to its axial coordinates is the most straightforward path possible. Moreover, the frame of reference is another concept that will be essential to keep in mind. The videos are a two-dimensional representation of the three-dimensional world and thus a major factor in understanding the limitations.
A circular motion in \(XY\) axes will appear sinusoidal to and fro motion when viewed from any direction other than \(Z\)-axis. A single-axis motion, similarly, may appear a stationary object from a particular viewing angle, deceiving reality. As explained, the outcomes are inherently dependent on the inputs provided and may not be in perfect harmony with the version the user predicted or felt. Before jumping right into the algorithm, it is better to take a slight detour to understand the fundamental Mathematics bolstering the concept. Correlation\(^1\) is defined to be a measure of the degree to which two signals/sequences are similar [28]. Although the field of correlation is immensely popular for practical applications like digital communications, radar, and sonar utilities [29], we have more propensity towards its subdivision, i.e., Auto-correlation. The Auto-correlation of a sequence is the mathematical description of the degree of association between a given time series and a time-delayed version of itself over the successive time interval. It simply defines how data points in a time series are related to the primary data point, on average [30]. Mathematically, the auto-correlation of a finite sequence \(x(t)\) is defined by Equation 7, in continuous form and by Equation 8, in discrete form for a sequence \(x[n]\).

\[
R_{xx}(\tau) = \int_{t_1}^{t_2} x(t)x(t + \tau)dt
\]  
\[R_{xx}[\tau] = \sum_{n=-\infty}^{\infty} x[n]x[n + \tau]
\]

For context, the auto-correlation decreases as the lag \(\tau\) increases in a structured process where adjacent measurements have similar values but distant points have no relationship. The auto-correlation of unstructured processes such as white noise, on the other hand, is theoretically equal to zero for all values of \(\tau > 0\) since there is no influence from one point of time on another. A positive auto-correlation value can be interpreted to remain above or below the signal’s mean value to measure data points’ persistence separated by this lag \(\tau\). A negative auto-correlation means that the mean value continues to alternate with data points separated by this lag. In terms of a time series motion sequence, the auto-correlation function of an aperiodic motion elicits a single peak resultant pattern as shown in Figure 9a. On the other hand, the function unravels a characteristic, multi-peak pattern that indicates the sequence’s periodicity, as shown in Figure 9b.

**Figure 9.** Auto-correlation of Motion Types. (a) Angle projected by linear motion. (b) Angle projected by Circular motion

Visually counting the peaks is a very trivial thing to comprehend but doing it algorithmically needs some mathematical background. Let \(f\) be a real-valued function and be an interior point within the range. Then, \(c\) is termed as the point of local maxima, if there is an \(h > 0\) such that \(f(c) > f(x)\), for all \(x \in (c-h, c+h)\), \(x \neq c\). It also suggests that \(f'(c) = 0\) at the point.

\(^1\) It is imperative to note that correlation is fundamentally different from convolution in terms of a very subtle mathematical difference [27]
Taking the discrete analogy, the value of $h$ is restricted to 1, i.e., $h = 1$ and gives a definitive point in the sequence.

![Figure 10. Divergence Boosting Implications. (a) Without Divergence Boost. (b) With Divergence Boost](image)

Practically, in some cases, the periodic motion is damped and not enough to cause a local maxima that can be detected and identified as a repetition, also shown in Figure 10a. Looking for the slope change can be an option to find the hidden peaks (maxima) but the option is highly susceptible to noise. To alleviate such issues, the feature of divergence boost is added, which amplifies the minute displacements to engender a more reliable identification. It requires the reference height and width of the displaying screen, depending on the axis, for determining the relative displacement and thus understanding the magnitude, to which the result must be amplified, mathematically relayed using Equation 9 and Equation 10. The normalizing parameter, $\mu$ acts as a check to the overfitting scenario. Moreover, the value is converged based upon the supervised learning that helps the model to adapt to the designated dataset. Finally, the overall divergence boosting feature for an N-lengthed sequence $x$ can be summarized as:

$$
\gamma = \frac{d_{\text{ref}}}{d_{\text{motion}}} \mu 
$$

(9)

$$
x[n] = x'[n] \quad \forall n \in [0, N]
$$

(10)

The implication of the feature can be perceived by Figure 10 that elicits two contrasting cases, where the same motion after auto-correlation 10a is different from an additional operation of divergence boosting, perceived in 10b. The former figure still shows some undulations that are noticeable only to the human eyes but does not fall in accordance with the discrete version of finding the local maxima. The divergence boosted output, on the other hand, clearly elicits the crest and trough being diverged locally, resulting in proper identification of motion periodicity as it comes inside the defined boundary of a local maximum.
4. Experimental Results

In the results section, the algorithms are tested extensively on the video dataset containing motions of similar categories, conveying the efficacy of the proposed algorithms with the mentioned features. The proposal is tested on a dataset of mobile captured videos with a resolution of 720p, primarily to simulate the practical conditions of its usage and real-world noise. Every class of motion categorized videos had 60 videos each and all of them were meant for testing. The dataset videos are of randomized objects in motion and were not exclusively selected as per convenience. For the testing purpose, the ‘object of interest’ is selected by clicking on it, and that specific object is motion-tracked until the video ends or the object moves out of the viewing frame. As mentioned before, the tracking is implemented using the concept of Optical flow.

Two parameters are used to quantitatively conclude the results of the algorithms: accuracy and processing time analysis. These two paradigms cover the crucial functional space the algorithms will be subjected to and are enough to judge its efficacy. The accuracy parameter makes sure that the categorization process is comparable to other pre-existing algorithms in the field, and on the other hand, processing time delineates this paper’s objective to propose a lightweight and fast solution for motion classification. The presented algorithms were run on a system with an Intel(R) Core(TM) i3-5005U CPU @ 2.00 GHz and 1 TB memory with OS support of Windows 10 Pro version 1909 (RAM: 8 GB). Processing times are proclaimed as run on a single thread. The python programming framework is put into use for all the algorithmic implementation and computations. The Machine Learning segment is implemented using the scikit-learn [31] package in the same programming language to obtain the results. The plots are produced using the seaborn package and MATLAB combinedly.

The Figure 11 shows a typical linear motion from the dataset in which a person walks across the viewing range from right to left, proposing a linear motion type.

![Sample video from the Motion Database depicting linear motion.](image)

**Figure 11.** Sample video from the Motion Database depicting linear motion.
4.1. Accuracy
An output is considered a correct classification when the tracked sequence (after motion segmentation) given to the respective algorithms predict the actual motion type, which is already known. It is also assumed that the user is capable of choosing a conforming feature to track in the video and the motion segmentation algorithms work ideally.

Table 2. Results of Determinant Method

| Method                      | Original Determinant | Logarithmic Conversion |
|-----------------------------|----------------------|------------------------|
| Logistic Regression         | 68.42 %              | 96.67 %                |
| K-Nearest Neighbor          | 92.10 %              | 94.74 %                |
| Decision Tree               | 96.67 %              | 100.0 %                |
| Random Forest               | 92.10 %              | 98.33 %                |
| Support Vector Machine      | 57.89 %              | 92.10 %                |

Coming to the quantitative part, the accuracy of individual motion types for the angle derivative method is different, i.e., 100% or linear motion and 95.12% for circular motion. Hence, the collective accuracy stands out at 97.56%. On the other hand, the periodic motion part successfully fetched an accuracy of 97.48% upon testing, and for the Determinant method, the observed accuracy is tabulated in table 2.

4.2. Processing Time Analysis
The computation process for the methods involves a series of steps, and thus the efficiency in processing time is observed to have reliable application support. The processing time is calculated only for the mentioned algorithmic parts, and the motion segmentation procedures are not included. The first algorithm took 15.6 ms on average to classify the given sequence into either of the classes, which turns out to be the reliable responsiveness of the technique. In contrast, the Determinant Method took 19.86 ms for the same. The periodic motion classification took an average processing time of 31.2 ms for computation.

5. Discussion
By interpreting the performance of the proposed algorithms, several characteristics become pristine. Each algorithm provides an average accuracy of about 95%. Given the simplifications and diluted reduced computation compared to Deep-Learning techniques, the results are pretty commending. The Angle Derivative Technique and Determinant Method took a computation time of less than 20 ms, and the periodic motion classifier completed its execution within 31.2 ms. Hence, the results indicate a positive result of achieving a lightweight motion classifier with trustable accuracy and speed. The additional upgrades presented in each algorithm, namely, the convolutional smoothing and flickering data in Angle Derivative Technique, the logarithmic conversion in Determinant Method, and Divergence boosting in Periodic Motion classifier, adds to the robustness and enhance the performance based on real-world conditions. The algorithms are also intuitive and scalable. Talking of all the assets the algorithm presents, the study would like to shift the focus to discuss the limitations as well. A typical fast motion adds only a few points to the tracking sequence, making the motion discontinuous and sparsely spread. The tracked path of a circular motion may appear as a quadrilateral under high speed. Similarly, a video input with a low frame rate will emulate the same condition as high-speed motion. Loss of continuity will eventually result in an inaccurate prediction, impacting the accuracy. Thus it is imperative to know the root causes of any misclassification by the algorithm, and partial knowledge of the application is a concern for the intended results.
Motion segmentation is the most crucial prerequisite for the motion classification algorithm to operate successfully, and the dependence on the former becomes equally imperative to discuss. As indicated previously, algorithms like Background subtraction, temporal differencing, and optical flow can be implemented to track the moving object and extract its positional information. Background Subtraction model finds the pixel difference between the current frame and a specific frame for comparison, generally the standard background image with no objects in it, and then thresholds the intensity to find the moving object. Tracking the object in such a case can be done using the centroid of the bounding box, but things are complicated when operates on noisy real-world data. The current frame may appear different to the computer compared to the standard comparison image because of varying light intensities and occlusions and therefore selecting the wrong moving object or its size, eventually culminating in inaccurate coordinates of the centroid. Such cases also hinder the accuracy of the overall model, whereas the fault is not on the side of motion classification. Similarly, the other tracking algorithms also tend to get affected by the unreliability of the input videos. Optical flow, which is relatively complex, depends heavily on the filters (kernels) and corner detection algorithms to track the significant features throughout the video. These features, once occluded, fail to track the object accurately and may start abnormal tracking of any other feature, once again reaching similar conclusions as described before. Multiple moving objects pose another logical obstacle to deciding which object to follow and is of particular interest, and doing the same with human intervention can be pretty convoluted.

6. Conclusions

This work essentially proposes three algorithms for the applications of Motion Classification, namely, Angle Derivative Technique, Determinant Method, and Correlation-based periodic motion classification. While the algorithms fare very well in accuracy and execution time, all designs are vulnerable to unprecedented inputs that we studied in detail, trying to foresee the algorithmic drawbacks they may face. Starting off, the determinant method elicits a condition that must be a topic of concern. The method relies immensely on the length of the tracked sequences. The longer the sequence, the better the accuracy, as the algorithm gets more rows and columns that follow the determinant’s properties stated. Shorter sequences as inputs may hinder the efficiency and applicability of the algorithm because of the short observation time or low frame rate. While the Angle Derivative technique can even work with small sequences and noisy data, the Determinant Method loses efficacy with either short lengthed sequences or noisy data. Increasing the length for accuracy compromises with the added complexity for determining the determinant of the larger matrix. The Determinant Method also compulsorily needs a square matrix to find the determinant. Thus, the algorithm can lose points in the given sequence with essential information that exceeds the squared length subset sequence obtained from the original, again subjected to a reduction in the accuracy.

Conclusively, this paper suggests using the Angle Derivative Technique for Motion Classification purposes with Correlation-based periodic motion classification. However, the former algorithm can be replaced by the Determinant Method for specific cases that elicit better performance than the Angle Derivative. The final flow of classification wraps up in the following order:

- Motion Segmentation & Tracking.
- Check for periodicity using the mentioned periodic motion classifier.
- If not periodic, go for Angle Derivative Technique for the final classification, i.e.
  - Find Standard Deviation of Sequence.
  - If less than the threshold, it is linear, otherwise circular.
The accuracy and processing time fall well within the target range of the concept, and the solution can be termed lightweight and robust in handling background clutter and deviation from ideal motion patterns. The algorithm’s accuracy is relatively comparable, demonstrating the accuracy of 95.12% for the worst case to other such applications where emotional state classification gave an accuracy of 95.6% for different performers [14]. The proposal of ST-LSTM showed an accuracy of 93.3% on the SBU-Kinect dataset, whereas 95.0% on the UT-Kinect dataset for the task of skeleton-based action recognition. On the other hand, the processing time is a success, showing only 15.6 ms and 19.86 ms for the Angle Derivative Technique and Determinant Method, respectively, considering the low-end specifications of the computer used. The study of Feature Engineering by Dong et al. [13] shows an exponential increase in training time with increasing features for training the model.

References

[1] Yurtsever E, Lambert J, Carballo A and Takeda K (2020) A survey of autonomous driving: common practices and emerging technologies. IEEE Access, 8: pp 58443-58469.
[2] Ingle S and Phute M (2016) Tesla autopilot: semi autonomous driving, an uptick for future autonomy. International Research Journal of Engineering and Technology, 3: pp 369-372.
[3] Taghribi A, Raie A and Shalchian M (2017) A novel fast bio-inspired feature for motion estimation. Iranian Conf. on Mach. Vis. and Img. Processing (MVIP), 10: pp 39-44.
[4] Sahouria E and Zakhor A (1997) Motion indexing of video. Proceedings of Intl. Conf. on Img. Processing, 2: pp 526–529.
[5] Dalal N, Triggs B and Schmid C (2006) Human detection using oriented histograms of flow and appearance. Comp. Vision – ECCV, Springer, Berlin, Heidelberg. pp: 428-441.
[6] Shu C, Ding X and Fang C (2011) Histogram of the oriented gradient for face recognition. Tsinghua Science & Tech, 16: pp 216-224.
[7] Bilinski P, Bremond F and Kaâniche MB (2009) Multiple object tracking with occlusions using HOG descriptors and multi resolution images. Intl. Conf. on Img. for Crime Detection and Prevention, 3: pp 1-6.
[8] Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Intl. Journal of Comp. Vision, 60: pp 91-110.
[9] Soatto S and Chiuso A (2016) Visual representations: defining properties and deep approximations. 6th Intl. Conf. on Learning Representations, 6.
[10] Bay H, Tuytelaira T and Van Gool L (2006) Surf: speeded up robust features. Springer, pp 404-417.
[11] Colque RVHM, Júnior CAC and Schwartz WR (2015) Histograms of optical flow orientation and magnitude to detect anomalous events in videos. 28th SIBGRAPI Conf. on Graphics, Patterns and Img, 27: pp 126-133.
[12] Delpont E, Noceti N, Odone F and Verri A (2017) The importance of continuous views for real-time 3d object recognition. Intl. Conf. on Comp. Vision (ICCV), 2: pp 1-8.
[13] Dong J, Karianakis N, Davis D, Hernandez J, Balzer J and Soatto S (2015) Multi-view feature engineering and learning. Proceedings of the IEEE Conf. on Comp. Vision and Patt. Recog.: pp 3251-3260.
[14] Aristidou A and Chrysanthou Y (2013) Motion indexing of different emotional states using LMA components. SIGGRAPH Asia 2013 Tech. Briefs: pp 1-4.
[15] Chao MW, Lin CH, Assa J and Lee TY (2011) Human motion retrieval from hand-drawn sketch. IEEE Transactions on Visual. and Comp. Graphics, 18: pp 729-740.
[16] Zuvela P, Lovric M, Yousefian-Jazi A and Liu JJ (2020) Ensemble learning approaches to data imbalance and competing objectives in design of an industrial machine vision system. Industrial & Eng. Chem. Research, 59: pp 4636-4645.
[17] Han G, Choi H and Song B (2015) Motion classification of pedestrian walking behaviors on the sidewalk. Seventh Intl. Conf. on Ubiquitous and Future Net.: pp 228-231.
[18] Kim D and Kwak K (2017) A motion analysis and classification based on PCA and ELM classifier. Intl. Conf. on Adv. Info., Concepts, Theory, and App. (ICAICTA): pp 1-4.
[19] Bengio Y, Courville A and Vincent P (2013) Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intel.*, 35: pp 1798-1828.

[20] Nargesian F, Samulowitz H, Khurana U, Khalil EB and Turaga DS (2017) Learning feature engineering for classification. *Proceedings of the Twenty-Sixth Intl. Joint Conf. on Artificial Intell.*: pp 2529-2535.

[21] Zhang S, Liu X and Xiao J (2017) On geometric features for skeleton-based action recognition using multilayer lstm networks. *IEEE Win. Conf. on App. of Comp. Vision (WACV)*: pp 148-157.

[22] Stauffer C and Grimson W (2007) Adaptive background mixture models for real-time tracking. *Proceedings of IEEE Conf. Comp. Vision Patt. Recog.*, 2: pp 246-252.

[23] Lipton A, Fujiyoshi H and Patil R (1998) Moving target classification and tracking from real-time video. *IEEE Workshop on App. of Comp. Vision (WACV’98)*, pp 8-14.

[24] Delpiano J, Jara J, Scheer J, Ramirez O, Hartel S and Solar R dJ (2012) Performance of optical flow techniques for motion analysis of fluorescent point signals in confocal microscopy. *Machine Vision and Applications*, 23: pp 675-89.

[25] Sharma RD (2016) *Determinants of Class – 12*. Dhanpat Rai Publications, pp 6.1–6.98.

[26] Cutler R and Davis L (1999) Real-time periodic motion detection, analysis, and applications. *IEEE Trans. Patt. Anal. Mach. Intell.*, 2: p 332.

[27] Boashash B (2003) Chapter 5 - Design of Time-Frequency Distributions. *Time Frequency Analysis* Oxford: Elsevier Science Boashash B, pp. 159-228.

[28] Babu P and Natarajan R (2006) *Signals and Systems*. Scitech publications, pp 20-45.

[29] Mitra SK (2010) *Digital Signal Processing*. McGraw-Hill Companies, pp 41-87.

[30] Box GEP, Gwilym M. Jenkins , Gregory C. Reinsel and Greta M. Ljung (1994) *Time Series Analysis: Forecasting & Control*, John Wiley & Sons, pp 21-46.

[31] Pedregosa F, Varoquaux G, Gramfort A, et al (2011) Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: pp 2825-2830.