Advances in Artificial Intelligence and Machine Learning; Research 3 (4) 1825-1833 Received 3-10-2023; Accepted 24-12-2023; Published 31-12-2023

A Note on Plant Virus Images for Use in Machine Learning

Senuka D. Abeysinghe senuka.abeysinghe24@ihsd.us
Indian Hill High School
Cincinnati, OH 45243, USA
Ohio's College Credit Plus Program
University of Cincinnati
Cincinnati, OH, 45221-0030, USA

Sarfaraz Ahmed Mohammed mohammsm@mail.uc.edu
Department of Computer Science
University of Cincinnati
Cincinnati, OH 45221-0030, USA

Anca Ralescu ralescal@ucmail.uc.edu
Department of Computer Science
University of Cincinnati
Cincinnati, OH 45221-0030, USA

Corresponding Author: Senuka D. Abeysinghe

Abstract

Plant viruses pose significant threats to agriculture, causing substantial economic losses and affecting food security. Traditional methods of virus detection and classification are often labor-intensive and time-consuming. In this study, we propose a novel approach to distinguish between different plant viruses using image classifiers. We convert the viral genome sequences into images using code generalization, representing nucleotides sequences as pixel intensities. Three popular machine learning algorithms applied to a dataset of plant virus images, namely k-means, k-NN, and Naive Bayes, are employed for clustering and classification. Our initial experimental results suggest that this approach is effective in distinguishing between various plant viruses, offering promising avenues for rapid and automated virus identification and classification.

Keywords: Genome representation, k-means, k-NN, k-Nearest neighbor, Naive Bayes.

1. INTRODUCTION

Plant viruses cause significant damage to crops threatening food security. To address this, we propose a novel approach using image-based machine learning algorithms (k-means, k-NN, and Naive Bayes) to classify plant viruses. The viral genome sequences are transformed into visual representations using a code generalization technique. Our goal is to create an efficient system for automated virus detection and classification to aid agricultural practices. By assembling a

Citation: Senuka D. Abeysinghe, et al. A Note on Plant Virus Images for Use in Machine Learning. Advances in Artificial Intelligence and Machine Learning. 2023;3(4):105.

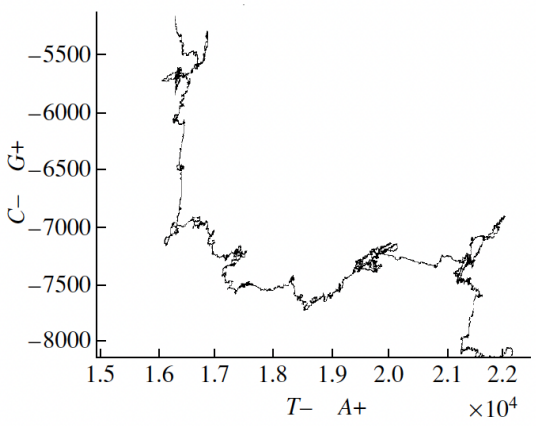

Figure 1: Fragment (1934000–2134000) of the human 22nd chromosome [1].

dataset of plant virus images we aim to enhance plant virology research and plant virus detection and classification which has traditionally relied on labor-intensive methods like serological assays and polymerase chain reaction (PCR). However, recent advances in machine learning and image-based techniques have opened new avenues for virus identification. Researchers have explored code generalization to convert genomic sequences into visual representation. The approach taken in this paper is inspired by the work of [1], and referenced therein, who devised a technique for generating a 2D representation of a genomic sequence. FIGURE 1 taken from [1], shows the 2D walk for a genomic sequence [2, 3]. However, while this is a visual representation which may help the human researcher in making inferences about the structure of that sequence (for instance, to hypothesize the location of telomeres), it is not an image object.

In this study a genomic sequence is represented as a 2D image. Genome visualization is a technique used to convert genomic sequences, which are typically represented as strings of nucleotides (A, T, G, C), into 2D images. The process consists in mapping each nucleotide to specific changes of (x, y) positions on a 2D grid as put forward by [1]. Our study goes a step further, by creating in fact an image corresponding to the 2D walk produced in [1]. Then, each time the pixel (x, y) is re-visited by this 2D walk, its intensity is increased. TABLE 1 shows the directional changes in the pixel (x, y) associated to each of four nucleotides. As the genomic sequence is scanned, the coordinates of a pixel (x, y) are updated according to the rules from TABLE 1.

Table 1: Rules for the pixel (x, y) to obtain the pixel (x', y'), starting from position $(0, 0)$: A" (adenine) shifts the position to the right, ''T'(thymine) shifts it to the left, "G"(guanine) moves it upwards, and "C"(cytosine) moves it downwards.

Nucleotide	x'	y'	Ordered pairs (starting from $(0, 0)$)
A	$x+1$	y	$(1, 0)$
T	$x-1$	y	$(-1, 0)$
C	x	$y+1$	(0.1)
G	x	$y-1$	$(0, -1)$

When represented as an image, rather than a simple 2D walk, after processing the entire genomic sequence, the result is a visually informative representation of the genome. In this image, the pixel intensity corresponds to the frequency with which that pixel has been visited by the 2D walk. This process results in a compact representation of the genomic sequence as an image, which further allows for leveraging the extensive repertoire of machine learning techniques for image data.

A Very Small Illustrative Example

TABLE 2 and FIGURE 2, illustrate this procedure for the sequence 'CAACTG' starting from the position $(0, 0)$.

Table 2: Example walk for the genomic sequence 'CAAGTC'

Nucleotide	Ordered pairs (starting from $(0, 0)$)
C	$(0, 1)$
A	$(1, 1)$
A	$(2, 1)$
G	$(2, 0)$
T	$(1, 0)$
C	$(1, 1)$

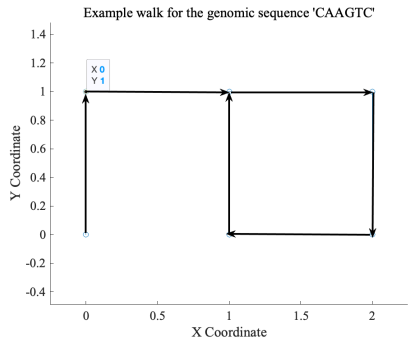

Figure 2: The image of the 2D walk for 'CAAGTC'.

To investigate how meaningful the 2D image representations of genome sequences are, the images obtained are used in machine learning algorithms, including k-Means, k-Nearest Neighbor, and Naive Bayes. The aim is to investigate the effectiveness of this approach to 2D genome visualization.

Brief Description of the Algorithms Used

Images have been a very often used domain of application of machine learning. In this study, three popular image-based machine learning algorithms, namely k-means, k-NN, and Naive Bayes, are used to classify plant virus images. A brief description of the algorithms used is as follows:

- The k-Means algorithm is an unsupervised machine learning algorithm that produces clusters of the data points based on proximity evaluation. The parameter k indicates the number of clusters desired. Each cluster is represented by a cluster representative computed as the mean of the data points in the cluster. A data point is assigned to the cluster with the nearest mean. Once a new data point is assigned to a cluster the cluster mean is recomputed. This process is applied iteratively until all the training data points are assigned to their respective clusters.

  To evaluate the clustering results, we computed cluster silhouettes. The silhouette score of a data point to a cluster, a value in $[-1, 1]$, conveys how good is the cluster for that data point. The higher the silhouette score, the better defined and well-separated the clusters are. By assessing the silhouette scores, one can gain valuable insights into the efficacy of a clustering algorithm.

- The k-Nearest Neighbor (k-NN) is a classification algorithm based on a voting procedure. The assignment of a data point to a class is driven by a vote from its k nearest neighbors. Usually, for a binary classification problem, k is set to an odd value such that a vote for neighbors does not result in a tie.

- The Naive Bayes classifier is a supervised probabilistic algorithm which computes the probability of a data point to a class across all the features. Starting from the training set, the empirical probability of an instance to a class, as well as the class distributions (the prior class probability) are computed. Then, for a data point, its classification is based on the posterior class probability, computed using the Bayes theorem.

2. METHODS

For the experiments carried out in this study, genomes of one 100 viruses were downloaded from the genome database of the National Center for Biotechnology Information (NCBI) [4]. Fifteen virus genomes were selected for training. These belong to one of the following virus classes: Tobacco mosaic virus, Banana bunchy top virus, and Cauliflower mosaic virus. Each of the virus genomes has a genetic sequence that is unique to it. For each virus genome, lengths are obtained and recorded. TABLE 3 shows the names and the length of the genomic sequence for 15 samples (five samples for each of the three types) of the viruses used as training set in this study.

Two-dimensional grayscale images are generated using the genome visualization described above. FIGURE 3 illustrates the images obtained for three different classes, with two samples per class.

3. EXPERIMENTAL RESULTS

The 2D grayscale images generated are of size 500×500 pixels, represented in row major order. Thus, the image data for the $N = 15$ and $N = 100$ viruses considered are combined to create a $N \times 250,000$ matrix.

Table 3: Plant viruses used in this study

Virus code	Virus name	Length of genome sequence
HE818416	Tobacco mosaic virus isolate Fumeng	6,579
HE818417	Tobacco mosaic virus isolate Chuxiong-1	6,579
HE818452	Tobacco mosaic virus isolate Xunyang	6,579
HE818453	Tobacco mosaic virus isolate Xunyang-2	6,577
HE818454	Tobacco mosaic virus isolate Xuyong	6,579
MT433346	Banana bunchy top virus isolate GM_1129006 segment DNA C	1,123
MT433347	Banana bunchy top virus isolate GM_418002 segment DNA C	1,119
MT433348	Banana bunchy top virus isolate GM_519005 segment DNA C	1,117
MT433349	Banana bunchy top virus isolate GM_619001 segment DNA C	1,117
MT433350	Banana bunchy top virus isolate GM_1129006 segment DNA C	1,120
AB863198	Cauliflower mosaic virus DNA, complete genome, isolate: GRC87G	8,260
AB863199	Cauliflower mosaic virus DNA, complete genome, isolate: GRC91B	8,259
AB863200	Cauliflower mosaic virus DNA, complete genome, isolate: GRC92A	8,259
AB863201	Cauliflower mosaic virus DNA, complete genome, isolate: GRC92C	8,258
AB863202	Cauliflower mosaic virus DNA, complete genome, isolate: GRC92D	8,262

k-Means Clustering

The MATLAB function `kmeans` [5], is first applied to a very small sample of 15 samples (five samples of each type of virus) and found that the clusters IDs returned match exactly the type of each virus.

To further evaluate the result of the k-means clustering, cluster silhouettes [6], were computed. The silhouette score of a data point to a cluster, a value in $[-1, 1]$, conveys how good is the cluster for that data point. The higher the silhouette score, the better defined and well-separated the clusters are. By assessing the silhouette scores, one can gain valuable insights into the efficacy of a clustering algorithm. Silhouette graphs (obtained using the MATLAB function `silhouette`) shown in FIGURE 4(a), and FIGURE 4(b). These clusters coincide with the initial selection of the data set of five examples for each of the three different viruses considered. While all the silhouette values are positive, their magnitudes show that those in cluster ID=3 are better clustered than those in clusters IDs=1, 2.

For the whole data set, a 100×250000 matrix, the virus types are as follows: 35 samples of Tobacco mosaic (ID =1), 31 samples of Banana bunchy (ID=2), and 34 samples of Cauliflower mosaic (ID = 3). The corresponding silhouettes are shown in FIGURE 4(b), and TABLE 4. Some data points appear mis-clustered to their original label (their negative silhouette values are shown in bold font in TABLE 4).

k-Nearest Neighbor Classification

The MATLAB function `fitcknn` [7], is used to create a k-NN classifier, $k = 15$. The k-nearest neighbor classification is applied to the 75×250000 data matrix. The MATLAB functions `resubPredict`,

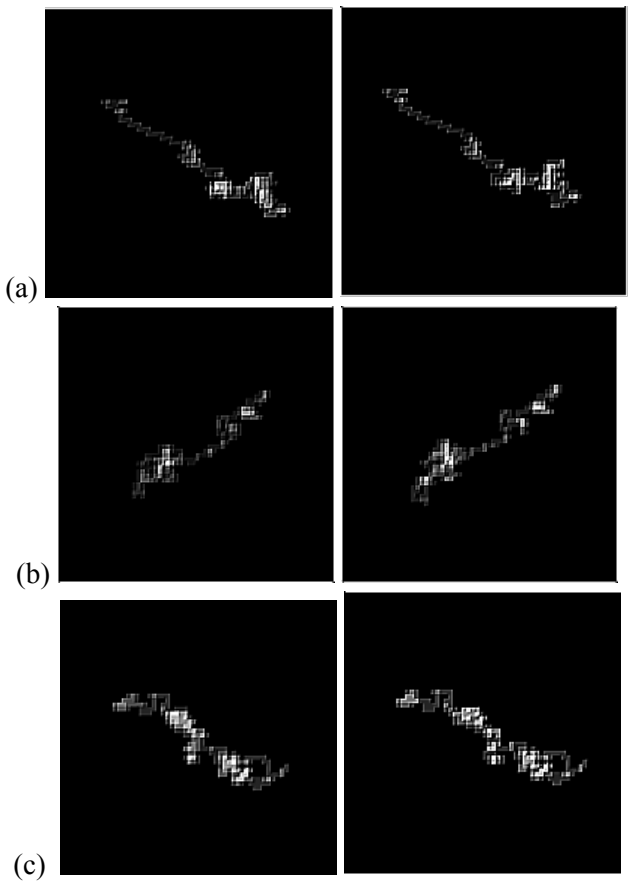

Figure 3: Two samples of each virus: (a) Tobacco Mozaic (b) Banana Bunchy Top (c) Cauliflower Mozaic.

and `confusionchart` are used to access the labels predicted and to display the confusion matrix shown in FIGURE 5(a).

It can be seen from FIGURE 5(a), that each of the 25 samples of the Cauliflower Mosaic and Tobacco Mosaic viruses were correctly predicted. Twenty four of the 25 Banana Bunchy virus were also correctly predicted, with the remaining one to be mis-predicted as Tobacco mosaic virus.

Naive Bayes Classifier

Using the MATLAB `fitcnb` function [8], the Naive Bayes classifier is trained on the feature matrix used for the k-nn classifier, and its corresponding labels. This involves estimating the probability distribution of each feature given the virus class. The aggregation across features is done by assuming independence between features. The MATLAB function `Mdl.ClassNames` displays the names of the virus classes, enabling an understanding of the labels used in the classifier. The MATLAB function `Mdl.Prior` calculates the prior probabilities of each virus class, providing insights into the prevalence of each class in the training data. The confusion matrix generated for the Naive Bayes classifier is shown in FIGURE 5(b).

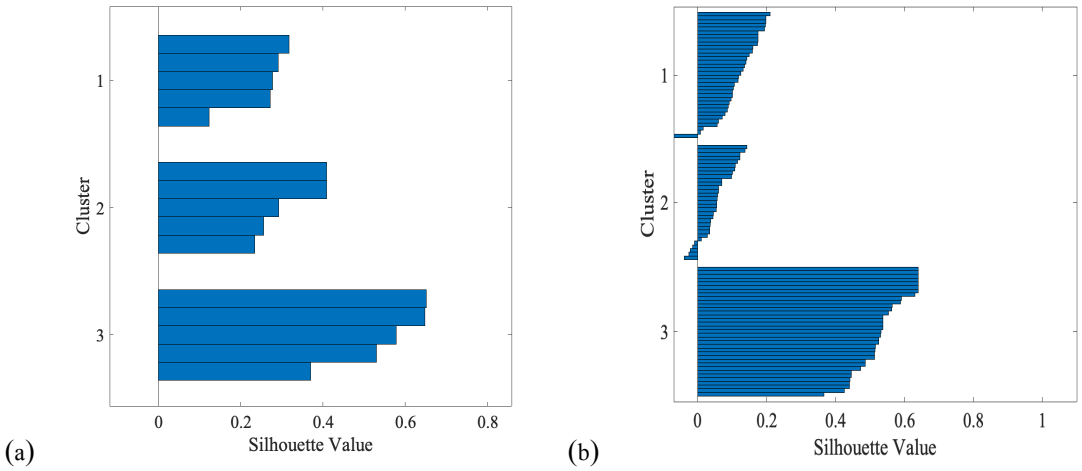

Figure 4: Silhouette values for k-Means clustering (k=3): (a) applied to the small set of 15 genomes, (b) applied to the whole data set.

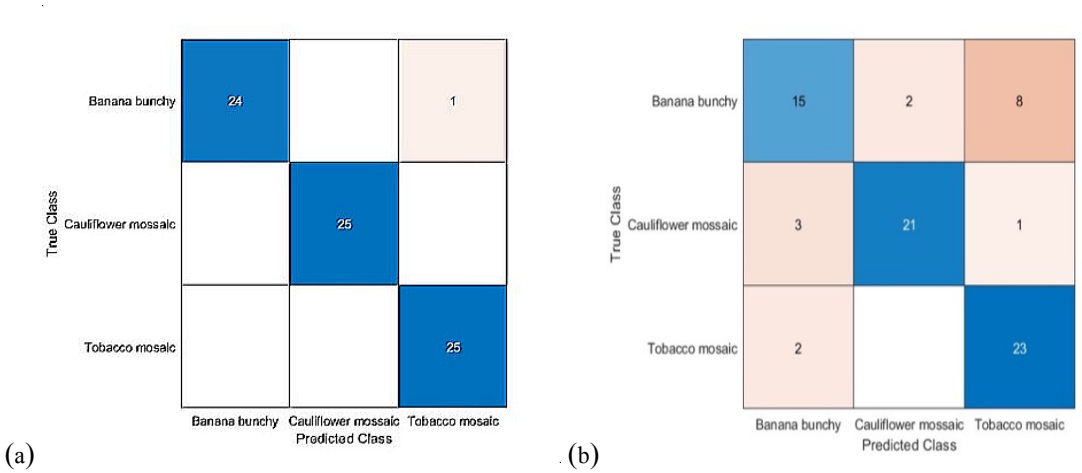

Figure 5: Confusion matrices for the k-nn classifier (a), and for the Naive Bayes classifier (b).

It can be seen that the overall average accuracy is 79%, with the highest accuracy of 92%, obtained for the Tobacco mosaic virus, followed by that of the Cauliflower mosaic, of 84%, and finally the accuracy of the Banana Bunch, of 60%. Qualitatively these results are in agreement with the clustering and the k-nn classifier, in the sense that the cluster for the Tobacco mosaic data is best (highest overall silhouette values), and the highest classification accuracy using both k-nn and Naive Bayes clustering.

Table 4: Silhouette values for the clusters generated by the k-Means algorithm for the the whole data set

	Tobacco Mozaic: Cluster 3		**Banana Bunchy:Cluster 2**		**Cauliflower Mozaic: Cluster 1**	
No.	ID	Silhouette	ID	Silhouette	ID	Silhouette
1	HE818435	0.6406	MT433356	0.1427	AB863183	0.21
2	HE818440	0.6406	MT433357	0.1374	AB863164	0.1976
3	HE818442	0.6406	MT433360	0.1226	AB863165	0.1976
4	HE818445	0.6406	MT433355	0.122	AB863185	0.1965
5	HE818446	0.6406	MT433352	0.1155	AB863191	0.194
6	HE818451	0.6406	MT433361	0.1095	AB863187	0.1753
7	HE818459	0.6406	MT433363	0.1078	AB863172	0.1748
8	HE818447	0.6311	MT433353	0.1006	AB863169	0.1747
9	HE818448	0.5916	MT433354	0.0981	AB863175	0.1734
10	HE818430	0.5893	MT433346	0.0704	AB863168	0.1608
11	HE818444	0.5646	MT433350	0.0704	AB863181	0.1585
12	HE818441	0.5638	MT433349	0.0611	AB863189	0.1492
13	HE818443	0.5539	MT433362	0.0611	AB863171	0.1428
14	HE818449	0.5374	MT433368	0.0579	AB863177	0.1401
15	HE818450	0.5374	MT433347	0.0567	AB863179	0.1365
16	HE818457	0.5374	MT433371	0.0553	AB863167	0.1323
17	HE818458	0.5374	MT433376	0.0553	AB863180	0.1252
18	HE818452	0.5323	MT433348	0.0546	AB863192	0.1193
19	HE818460	0.5311	MT433367	0.0455	AB863186	0.1173
20	HE818433	0.525	MT433366	0.0444	AB863182	0.107
21	HE818434	0.525	MT433351	0.0384	AB863166	0.1036
22	HE818416	0.5158	MT433375	0.0366	AB863176	0.1006
23	HE818454	0.5142	MT433372	0.0356	AB863201	0.1005
24	HE818431	0.5129	MT433358	0.0353	AB863190	0.0976
25	HE818432	0.5129	MT433369	0.029	AB863178	0.0917
26	HE818455	0.4863	MT433359	0.011	AB863174	0.0889
27	HE818456	0.4863	MT433370	**-0.0093**	AB863198	0.0864
28	HE818453	0.4731	MT433373	**-0.0152**	AB863199	0.0792
29	HE818428	0.4458	MT433364	**-0.0219**	AB863202	0.071
30	HE818429	0.4458	MT433365	**-0.0257**	AB863188	0.0614
31	HE818436	0.4417	MT433374	**-0.0387**	AB863200	0.0569
32	HE818438	0.4413			AB863170	0.0159
33	HE818439	0.4413			AB863184	0.0078
34	HE818437	0.4258			AB863173	**-0.0693**
35	HE818417	0.3667				

4. CONCLUSION

This study started from the 2D-walk representation of a genome suggested in [1], and references therein. More specifically, we suggested that while such 2D-walk representation provides a visual tool helpful to researchers in biology, it is not amenable to an automatic processing for obtaining more insight into various types of gnome sequences. Our study converted genome sequences into

image data, where pixels correspond to walk locations and repeated visits of a location to pixel intensity. We applied this approach to a data set of plant virus genome sequence. The resulting data set of images was then used in three popular machine learning algorithms for clustering and classification. Based on these preliminary results, it is reasonable to expect that the image representation of the genome sequence is *meaningful*, and therefore, it offers the opportunity to automatic processing of larger sets of genome sequences.

References

[1] Larionov SA, Loskutov AY, Ryadchenko EV. Enome as a Two-Dimensional Walk. Dokl Phys. 2005;50:634-638.

[2] Arneodo A, Vaillant C, Audit B, Argoul F, d'Aubenton-Carafa Y, Thermes C. Multi-Scale Coding of Genomic Information: From Dna Sequence to Genome Structure and Function. Phys Rep. 2011;498:45-188.

[3] Li C, Fei W, Zhao Y, Yu X. Novel Graphical Representation and Numerical Characterization of DNA Sequences. Appl Sci. 2016;6:63.

[4] https://www.ncbi.nlm.nih.gov/genomes/GenomesGroup.cgi?taxid=10239.

[5] https://www.mathworks.com/help/stats/kmeans.html

[6] Rousseeuw PJ. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. J Comput Appl Math. 1987;20:53-65.

[7] https://www.mathworks.com/help/stats/fitcknn.html

[8] https://www.mathworks.com/help/stats/fitcnb.html