Radar Targets Classification with Manifold Learning

Yoad Pilosof Aviv-Abraham Golan Alon Eilam Pavel Lifshits
Signal and Image Processing Laboratory (SIPL)
Andrew and Erna Viterbi Faculty of Electrical and Computer Engineering
Technion – Israel Institute of Technology
Technion City, Haifa, Israel
https://sipl.eelabs.technion.ac.il

***Abstract*—In this work we propose a new approach for Radar targets classification by using Manifold Learning. We focus on the acute problem of differentiating UAVs from birds. Micro Movements of Radar target, resulting from UAVs rotating propellers or birds' wings flapping cause scattering of the Radar echo signal in the frequency domain, named Micro Doppler. In previous works, the Micro Doppler effect was analyzed with classical methods such as SVM, KNN and Bayes. However, the large dimensionality of the data, as well as the small dataset has limited the performance achieved. The innovation in our work is applying Manifold Learning based preprocessing of the Range-Doppler map prior to SVM classification. Our results show the effectiveness of the proposed method. With real life dataset, we have achieved 82.9% correct classification per single target hit, Vs. 74.4% achieved by classical methods on the same data set.**

***Index Terms*—Birds, Classification, Diffusion Map, Manifold Learning, Micro-Doppler, Radar, SVM, UAV**

I. INTRODUCTION

Due to the increasing use of UAVs for military missions including intelligence and armed attacks, an effective way for differentiating UAVs from birds is essential. Birds cause false alarms since they have velocities, elevation and Radar Cross Section (RCS) similar to UAVs. Micro movements of the target, resulting from UAVs rotating propellors or birds' wings flapping, cause scattering of the Radar echo signal, named Micro Doppler effect [1], [2]. Analyzing the Micro Doppler effect may assist us in distinguishing between UAVs and birds [3]–[7].

Range Doppler Maps are matrices derived from the Radar echo signals, where each cell of the matrix describes the energy of the reflected Radar signals for a given range and velocity (resulting from doppler shift). The x axis describes velocities. The y axis describes the different ranges, and each row is called a Range Gate.

Due to the large dimension of the Range Doppler Map, Principal Component Analysis (PCA) was used for dimension reduction by computing the change in basis transformation, such that the first N components best represent the data. The other components can then be ignored with minimal data loss.

Machine Learning algorithms such as Support Vector Machine (SVM) apply a classification method which constructs the best hyper plane that separates the data to its classes. This hyper plane is then used for targets classification. However, the large dimensionality of the data prevents the construction of an effective hyper plane, and therefore PCA was applied in previous works as preprocessing for dimensionality reduction.

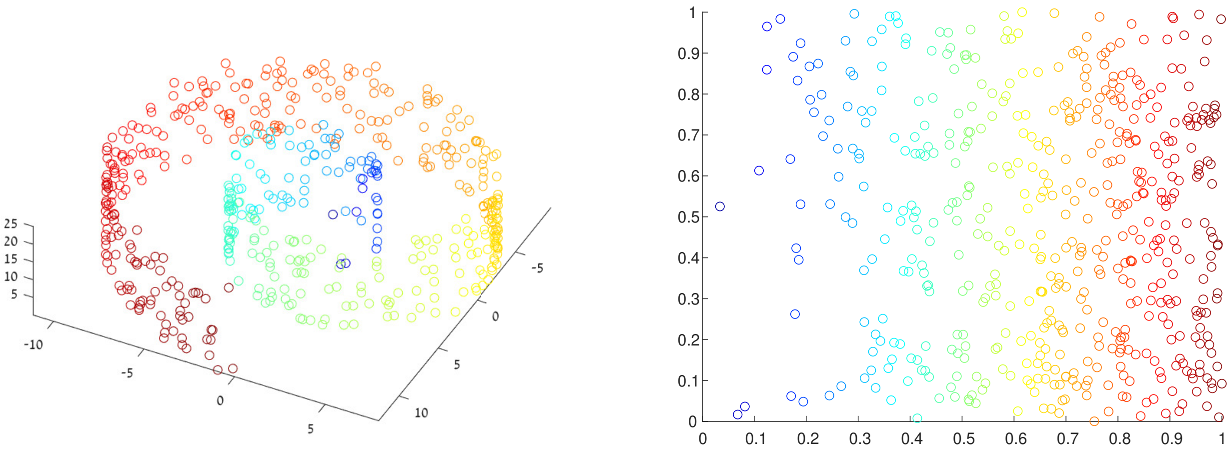
Manifold Learning theory [8] builds on the assumption that the sampled data requires just a few parameters to represent it, yet it resides in a higher dimensional space. Manifold Learning techniques attempt to study and use the structure of the data, such as by calculating distances between points with the constraint that the path lies on the low-dimensional subspace of the data. We expect that using the lower dimensional structure of the data provided by Manifold Learning as input to a classification algorithm, would yield better results than PCA preprocessing.

In this paper we will discuss a Radar targets classification method based on Manifold Learning. Manifold Learning is a non-linear transformation, and this new approach has improved the classifier performance.

Section II discusses the common micro doppler based methods used for classification. Section III provides a description of Manifold Learning, and the Diffusion Map algorithm used in this paper. Section IV presents the problem with the embedding of new data points and offers a solution. Section V describes our algorithm in its entirety. Section VI provides the performance of our solution Vs. the common, PCA based solutions. Section VII concludes this paper.

II. RELATED WORK

Several methods exist that use the Micro Doppler effect for classification and information gathering purposes. One method [3], [4] takes an STFT transform of the time domain radar signal, and extracts its features. This method relies on the signal having a known structure, and the features are problem specific. For example, if the STFT transform is in a sinusoidal form, the amplitude, frequency, DC component and phase features are extracted and used for classification. Other methods [5]–[7] are based on Machine Learning, and can be used in situations where there is no obvious structure of the data. By this method, STFT transform features are extracted using the PCA algorithm. The data is then fed to a classifier, most commonly SVM. Different improvements were offered, such as removing the center of mass velocity from the STFT transform, or using Gabor filters prior to the SVM. Since this method relies on Machine Learning, a training set should be used to find the change of basis transformation for the PCA algorithm and train the classifier.



(a) (b)

Fig. 1. Example of (a) 3D data and (b) its non linear 2D embedding

III. Manifold Learning and Diffusion Map

Points that reside in a high-dimensional space can sometimes be represented as a low-dimensional sub-space. We call this subspace a Manifold. It is advantageous to transform the data to the low-dimensional space that can represent it better. This transformation is not necessarily linear. As an example, Figure 1 shows 3D data and its non-linear 2D embedding. The high-dimensional Euclidean distance ignores the structure of the data while calculating the distance between 2 points. The Diffusion distance constructs a graph where the weight of an edge connecting two points is based on their high-dimensional Euclidean distance (where closer points have higher weights). The Diffusion distance finds paths between two endpoints by traveling along high-weight edges, and uses these paths to calculate a distance function.

Diffusion Map [8], [9] is a Manifold Learning based algorithm that transforms points from the Euclidean space to Diffusion space. It can be shown that the Diffusion distance in the Euclidean space is exactly the same as the Euclidean distance in the Diffusion space. Therefore, we transform the data using the Diffusion Map algorithm, and then may use common classification methods with the Euclidean distance.

Moreover, since in the Diffusion space the elements in the vector that describes the point are mostly sorted by their significance, we can use only the first few elements to calculate the Euclidean distance. This lowers the dimensionality and complexity significantly with minimal distortion.

Algorithm 1 describes the steps to construct the diffusion map.

Algorithm 1 Diffusion Map

1: Choose values for the parameters: $\epsilon > 0, t \geq 1, l \leq N$ where N is the number of points
2: Construct a matrix K such that

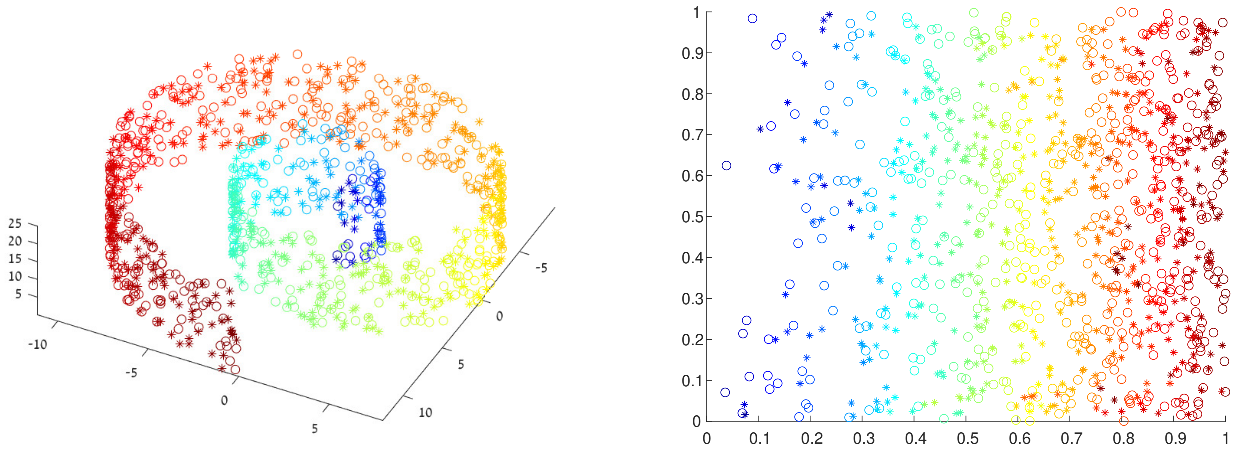
$$K_{ij} = \exp\left(\frac{\|x_i - x_j\|_2^2}{2\epsilon^2}\right)$$

3: Define the matrix P such that $P_{ij} = K_{ij} \cdot \left(\sum_j K_{ij}\right)^{-1}$
4: Consider P as a Transition Matrix of a Markov Chain
5: Calculate the E.V. of $P^t, \{\lambda_i^t, \psi_i\}$
6: Embed each sample $\psi_t^l(x_i) = [\lambda_1^t \psi_1(i), \ldots, \lambda_l^t \psi_l(i)]$
7: Approximate distance between two points by

$$d(x_i, x_j) = \left\|\psi_t^N(x_i) - \psi_t^N(x_j)\right\| \approx \left\|\psi_t^l(x_i) - \psi_t^l(x_j)\right\|$$

IV. Out of Sample Extension

Unlike the PCA method which provides the axes on which the points are cast, the Diffusion Map algorithm doesn't provide us with a way to embed new points. This is because the PCA transformation is linear, and we can calculate the transformation of a new point by interpolation. In practice, we calculate the transformation for a new point by projecting it on the axes found in the PCA algorithm. However, the Diffusion Map transformation is not linear, and thus it is not possible to apply a transformation of a new data point in a similar way. The inability to embed new points is problematic because it means that for each new point we want to classify, a new embedding must be calculated, and a new classifier must be trained from the ground up. This is highly inefficient. The solution is to approximate the embedding for new points using interpolation methods. In this context, the interpolation process is often called Out of Sample Extension (OoSE). This means that we can perform most of the calculation during the training stage, and leave only few steps for classification. We achieve this by using an OoSE algorithm called Laplacian Pyramid Extension (LPE) [10]. In this algorithm we represent the embedded points using other points and their embedded forms, and representation coefficients. We then represent new



(a) (b)

Fig. 2. (a) 3D data and its (b) 2D embedding with extension. The embedding was calculated only for points marked ∘ and was extended for points marked ⋆

points using other points and their embedded forms, and the same representation coefficients. An example of this algorithm is shown in Figure 2.

Given the function **f** to interpolate, The Laplacian Pyramid Extension is performed by the algorithm 2.

Algorithm 2 Laplacian Pyramid Extension

1: Choose values for the parameters $\epsilon, l_{max}, E_{min}$

2: Define $\mathcal{K}_l = \exp\left(\frac{\|x_i - x_j\|_2^2}{2\frac{\epsilon^2}{2^l}}\right)$

3: Calculate $q_i^l = \sum_{j=1}^{N} \mathcal{K}_l\left(x_i, x_j\right)$

4: Iteratively calculate the following until $l \geq l_{max}$ or $\max_{x_k} d_l\left(x_k\right) \leq E_{min}$:

$$s_l\left(x_k\right) = \begin{cases} \sum_{i=1}^{N}\left(q_i^l\right)^{-1} \cdot \mathcal{K}_l\left(x_i, x_k\right) \cdot f\left(x_i\right) & l = 0 \\ \sum_{i=1}^{N}\left(q_i^l\right)^{-1} \cdot \mathcal{K}_l\left(x_i, x_k\right) \cdot d_l\left(x_i\right) & l \geq 1 \end{cases}$$

$d_l\left(x_k\right) = f\left(x_k\right) - \sum_{m=0}^{l-1} s_m\left(x_k\right), l \geq 1$

5: Interpolate the function f on a new point $\tilde{x}$ by

$$f\left(\tilde{x}\right) \triangleq \sum_{m=0}^{l} s_m\left(\tilde{x}\right)$$

V. Algorithm

Figure 3 shows the structure of the algorithm developed in this work. Since the target is detected by the Radar prior to classification, the algorithm inputs are Range Doppler Maps, the corresponding Range and Velocity of the detected targets and the correct classification for the training set. Each map is cropped such that only the target Range Gate vector with about 200 elements in the velocity axis, symmetrical about the target, is kept. Each vector is normalized such that its values are between zero and one. We use this cropping because the micro movements are in the same range as the target, and their velocities in relation to the target are bounded. Since the Micro Doppler effect is confined to this 200-dimensional vector, we should be able to classify each map using only this vector. The Diffusion Map algorithm is used on the training set (where each sample is a 200-dimensional vector) to generate the embedded form of each sample. We use these embedded points to train an SVM classifier, and to find the representation coefficients from the LPE algorithm. This concludes the training phase. To classify a new target, provided its Range Doppler map, we center and crop it as before. To embed a point, we use the LPE algorithm, using the representation coefficient we calculated in the training phase. We then take the embedded form of the range doppler map and feed it to the SVM classifier we trained. The result of the SVM classifier provides the new target classification.

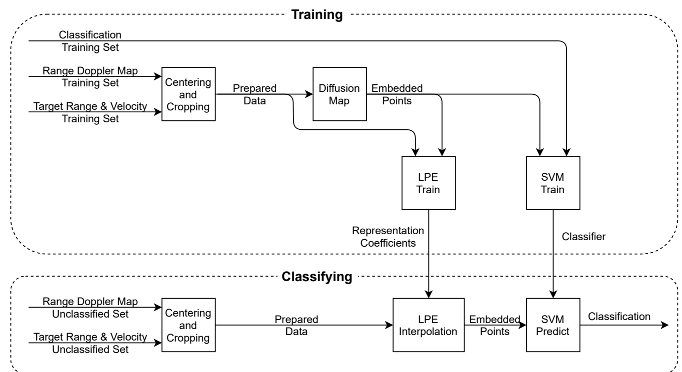
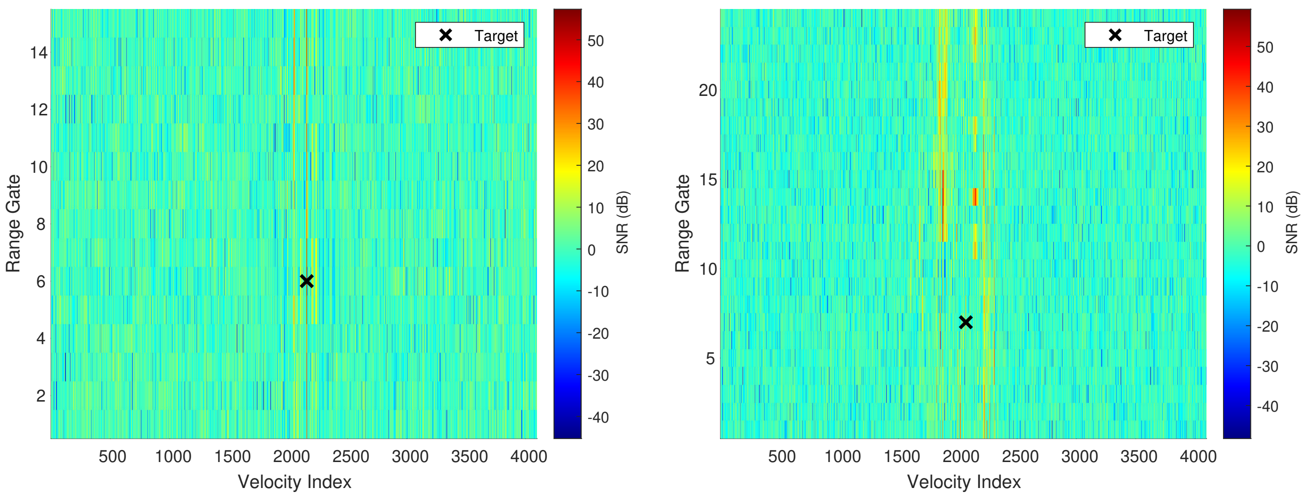


Fig. 3. Flowchart of the classification algorithm

While applying the algorithm on real life sampled data, several hyper parameters shall be tuned to control the pre-processing stage. We define the Crop Size as well as the Eigenvalue Power (t) and Embedded Dimension (l) which control the Diffusion Map stage. Additionally, a cost ratio for



(a) (b)

Fig. 4. Examples of Range Doppler maps of (a) drone and (b) bird, with the target marked

the SVM classifier defines which error is costlier, type I or type II, and by how much. It is required in cases where the training data is biased, i.e., in cases where there are much more samples from one class, or when the classifier is biased, i.e., the difference between the TPR (True Positive Rate, drone classified as a drone) and TNR (True Negative Rate, bird classified as a bird) is too large.

Another hyper parameter we tuned in the Diffusion Map algorithm was ϵ, which is the standard deviation of the Gaussian kernel. We found that it is best to choose a value dynamically, to better match the data. We chose this value as the median Euclidean distance of all pairs of points, multiplied by some constant. This constant was another hyper parameter of the entire algorithm.

The last hyper parameter we used controlled the locality of the data. When calculating the pairwise distance (which is used in the Gaussian kernel), for each point we forced the farthest points to have an infinite distance (which means the Gaussian kernel will be zero). This was done to make sure that the paths found by the Diffusion distance are on high-weighted edges. The percentage of points whose distance was set to infinity is the last hyper parameter we implemented.

TABLE I
FINAL VALUES FOR THE HYPER PARAMETERS OF THE ALGORITHM

Hyper Parameter	Value
Crop Size	201
Eigenvalue Power, t	1
Embedded Dimension, l	100
SVM Cost Ratio	4
Standard Deviation Constant	2
Locality Factor	40%

Different sets of hyper parameters were tested using k-fold validation to evaluate the success rate of the algorithm. We performed a sweep over different values of the Hyper Parameters to choose their optimal values. Table I shows the values chosen for the final algorithm.

To recap, the algorithm is as follows: Crop the data to be centered about the target and normalize it, transform it into the Diffusion space using the Diffusion Map algorithm and train an SVM classifier on a set of tagged samples. Then use the Laplacian Pyramid Extension algorithm to interpolate an embedding for the testing set. Following the training process, the classifier is ready to classify new targets.

VI. PERFORMANCE EVALUATION

We have analyzed a total of 461 samples including 118 Drones and 343 Birds. The data was in the format of Range Doppler maps. Examples of the data set are shown in Figure 4. Manifold Learning preprocessing followed by SVM classifier was compared to a classical solution using PCA [5]–[7] preprocessing, with the same embedded space dimension (100). The results are shown in Table II. Our results show that using Diffusion Map based Manifold Learning has provided a better performance with real data. Other preprocessing methods that were recommended in the literature were implemented and evaluated yet gained a negligible improvement Vs. PCA. Table II shows the results of applying the classical method Vs. Manifold Learning based performance.

TABLE II
PERFORMANCE OF THE PCA BASED SOLUTION VS. THE MANIFOLD LEARNING BASED SOLUTIONS. THE TABLE DESCRIBES THE TPR (TRUE POSITIVE RATE, DRONE CLASSIFIED AS A DRONE), TNR (TRUE NEGATIVE RATE, BIRD CLASSIFIED AS A BIRD) AND BA (BALANCED ACCUARCY) OF THE TWO METHODS

	PCA	Manifold Learning
TPR	77.0%	86.1%
TNR	71.8%	79.8%
BA	74.4%	82.9%

VII. Conclusions

A new approach for Radar targets classification using Manifold Learning was successfully implemented as a nonlinear dimensionality reduction stage followed by SVM classification. With a real-life dataset, a probability of 82.9% correct classification per single target hit was achieved, Vs. 74.4% probability of correct classification rate with classical Machine-Learning preprocessing methods such as PCA.

Acknowledgment

The authors are grateful to RADA Electronic Industries, and in particular to Ilan Wittenberg and Gili Molcho for their kind support of this work. The authors would also like to thank Prof. David Malah, head of the Signal and Image Processing Laboratory (SIPL) at the Andrew and Erna Viterbi Faculty of Electrical and Computer Engineering in the Technion – Israel Institute of Technology, to Prof. Ronen Talmon, to Nimrod Peleg, SIPL's chief engineer, and to the entire SIPL team for their support, advice, and helpful comments.

References

[1] V. C. Chen, F. Li, S.-S. Ho, and H. Wechsler, "Micro-doppler effect in radar: phenomenon, model, and simulation study," *IEEE Transactions on Aerospace and electronic systems*, vol. 42, no. 1, pp. 2–21, 2006.

[2] V. C. Chen, *The micro-Doppler effect in radar*. Artech House, 2019.

[3] S. Björklund, H. Petersson, and G. Hendeby, "Features for micro-doppler based activity classification," *IET radar, sonar & navigation*, vol. 9, no. 9, pp. 1181–1187, 2015.

[4] Y. Kim and H. Ling, "Human activity classification based on micro-doppler signatures using a support vector machine," *IEEE transactions on geoscience and remote sensing*, vol. 47, no. 5, pp. 1328–1337, 2009.

[5] J. Lei and C. Lu, "Target classification based on micro-doppler signatures," in *IEEE International Radar Conference, 2005.* IEEE, 2005, pp. 179–183.

[6] P. Molchanov, R. I. Harmanny, J. J. de Wit, K. Egiazarian, and J. Astola, "Classification of small uavs and birds by micro-doppler signatures," *International Journal of Microwave and Wireless Technologies*, vol. 6, no. 3-4, pp. 435–444, 2014.

[7] Y. Yang, J. Lei, W. Zhang, and C. Lu, "Target classification and pattern recognition using micro-doppler radar signatures," in *Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'06).* IEEE, 2006, pp. 213–217.

[8] R. Talmon, I. Cohen, S. Gannot, and R. R. Coifman, "Diffusion maps for signal processing: A deeper look at manifold-learning techniques based on kernels and graphs," *IEEE signal processing magazine*, vol. 30, no. 4, pp. 75–86, 2013.

[9] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and computational harmonic analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[10] G. Mishne and I. Cohen, "Multiscale anomaly detection using diffusion maps and saliency score," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2014, pp. 2823–2827.