Journal of Science and Technology
ISSN: 2362-1443 (Print) 3059-9490 (Online)
2024, Vol. 4, No. 2 (Special Issue), 51-57

Decoding Facial Recognition: Analyzing Standalone Euclidean and KNN Distance Metrics

Prajwol Chhetri
Kathmandu, Nepal
prajwol.chhhetri@gmail.com

Sunil Raut Kshetri
North Shields, England, United Kingdom
sunil.rautkshetri@gmail.com

Article History:
Received: 31 July 2024
Revised: 8 August 2024
Accepted: 17 December 2024

Keywords— ***Facial recognition, K-Nearest Neighbors (KNN), Distance metrics, Euclidean Distance, Manhattan Distance, Minkowski Distance, DLIB, Computational Efficiency, Accuracy, Real-Time Application, YaleFaces, FERET Dataset, Real -Time Identification***

Abstract— **This study compares the performance of K-Nearest Neighbors(KNN) using different distance metrics—Euclidean, Manhattan, and Minkowski—as well as the standalone Euclidean distance metric in facial recognition tasks. Our objective was to compare these methods in terms of accuracy and computational efficiency across diverse datasets,including Celebrity Faces, Color FERET, Family Faces with and without occlusions, and Yale Faces. Utilizing Dlib's face detection tools, we performed a thorough analysis of each metric's effectiveness. The evaluation involved multiple iterations to ensure robust results. Our results reveal that the Manhattan distance metric generally offers superior computational efficiency,with the lowest average computation time of 0.588 milliseconds, compared to Euclidean and Minkowski metrics. While the Euclidean metric provides high accuracy, particularly in controlled Environments like the Yale Faces dataset the Manhattan metric offers a more balanced tradeoff between accuracy and computational time.The Minkowski distance metric,while versatile, generally falls between the Euclidean and Manhattan metrics in terms of efficiency and accuracy. In scenarios with occlusions, such as those found in the Family Faces dataset, the Manhattan distance metric maintains a competitive edge in both accuracy and efficiency compared to other metrics. For applications that demand both high accuracy and minimal computational overhead, especially in real-world environments with potential occlusions, the Manhattan distance metric is particularly recommended. Conversely, when computational efficiency is less of a concern, the Euclidean metric remains a reliable choice. These insights provide valuable guidance for optimizing facial recognition systems to meet diverse application requirements and computational constraints.**

I. Introduction

Facial recognition involves identifying or verifying a person from a digital image or a video frame [1]. It employs various algorithms to map facial features and compare them with a database of known faces. Facial recognition technology is becoming increasingly integral to various applications, from enhancing security to enabling new social media functionalities [1]. With the advances in deep learning and more sophisticated algorithms, the accuracy and efficiency of facial recognition systems have seen significant improvements. Researchers have explored a variety of techniques, including Eigenfaces, Fisherfaces, and modern deep learning approaches[1]. While these methods have achieved notable success,balancing accuracy with computational efficiency remains a challenging task.

Although many studies have compared different algorithms for facial recognition, the comparison of different distance metrics with classifiers like K-Nearest Neighbors (KNN) and standalone metrics like Euclidean distance has not been extensively explored. This gap led to our central research question: How do KNN with different distance metrics and standalone Euclidean distance metric compare in facial recognition tasks regarding accuracy and computational efficiency? To address this, we conducted a comparative analysis of KNN with different distance metrics and Standalone Euclidean distance metric for facial recognition. Using dlib's face recognition model and a series of rigorous experiments,we aim to provide valuable insights into the strengths and weaknesses of each method, thereby guiding future research and practical applications in the field.

A. Literature Review

Facial recognition under challenging conditions, such as occlusions, has been a focus of several studies. For example,"Face Recognition Under Partial Occlusion: A

Detection and Exclusion of Occluded Face Regions Approach" explores methods to improve recognition accuracy by detecting and excluding occluded regions [2]. This study found that excluding occluded regions can significantly enhance recognition accuracy, particularly in scenarios where occlusions are common [2]. Our study complements this by showing that even when occlusions are not explicitly excluded, KNN and Euclidean distance metrics maintain high accuracy, though slight variations exist depending on the dataset.

Similarly, the survey "A survey on techniques to handle face recognition challenges: occlusion, single sample per subject and expression" highlights the impact of occlusions on recognition accuracy, emphasizing that holistic approaches are more adversely affected by occlusions than local approaches[3]. This aligns with our findings, where the presence of occlusions in the Family Faces dataset led to a slight reduction in accuracy, particularly for the Euclidean and Minkowski metrics.

In "Occluded Face Recognition Using Low-rank Regression with Generalized Gradient Direction" the authors propose a low-rank regression method to improve the robustness of facial recognition systems under occlusions [4]. While our study did not focus on advanced occlusion handling techniques, it contributes to the understanding of how different distance metrics perform in the presence of occlusions, with Manhattan distance showing a slight edge in accuracy under such conditions.

Our study's findings also resonate with the results of "Analysis of euclidean distance and Manhattan Distance measure in face recognition", which found that Manhattan distance can sometimes outperform Euclidean distance in certain scenarios [5]. This aligns with our observation that Manhattan distance was generally faster in computational terms, although Euclidean distance had a slight edge in accuracy for certain datasets.

The paper "An Efficient Face Recognition using PCA and Euclidean Distance Classification" focuses on the use of Euclidean distance in a PCA-based facial recognition system [6]. While it shows strong performance for frontal faces, our study extends this by comparing Euclidean distance with other distance metrics in more varied datasets, including those with occlusions.

Finally, "Facial Recognition using Enhanced Facial Features k-Nearest Neighbor (k-NN) for Attendance System" discusses the use of KNN with enhanced facial features for an attendance system [7]. Our study builds on this by exploring how different distance metrics within KNN affect both accuracy and computational efficiency across different types of facial recognition datasets.

II. METHOD

This study explores the performance of KNN using different distance metrics (Euclidean, ManhaΣan, Minkowski) in facial recognition tasks, compared with the standalone Euclidean distance metric. Using Dlib's tools [8], We focused on evaluating their accuracy and computational efficiency under various conditions.

A. Study Design

To address this research question, we designed a study that involved implementing both KNN with different metrics and Standalone Euclidean metric within a facial recognition system. We then evaluated the performance of each method using a variety of datasets, focusing on both accuracy and the computational time required for recognition.

B. Data

We utilized several datasets for this study:

- Color FERET Dataset: The color FERET database is a dataset for face recognition. It contains 11,338 color images of size 512×768 pixels captured in a semi-controlled environment with 13 different poses from 994 subjects [9].
- Yale Faces Dataset: The Yale Face Database is available publicly for non-commercial use, contains 165 grayscale GIF images of 15 individuals. Each person has 11 images showing different facial expressions and configurations[10].
- Celebrity Face Image Dataset: Available from Kaggle, this dataset contains labeled images of 18 Hollywood celebrities with 100 images of each celebrity [11].
- Family Faces Dataset with Occlusions: A custom dataset containing images of 14 family members with occlusions such as sunglasses. Each family member has 10 images showing different facial expressions and in different angles and lighting conditions.
- Family Faces Dataset without Occlusions: Same custom dataset used previously but without occlusions. The images with occlusions such as sunglasses were removed to check the effect of occlusions in facial recognition

All datasets were split into 80-20 for training and testing to observe the changes in performance. Additionally, 10 iterations for each of the experiments were conducted to ensure the robustness of the results.

C. Methods and Algorithms

a) Face Detection:: We utilized the 'dlib' library's frontal face detector , which is based on the Histogram of Oriented Gradients (HOG) and Linear Support Vector Machine (SVM)Method.

Algorithm:-

- Histogram of Oriented Gradients (HOG): This features descriptor is used to detect objects with images by dividing the image into small regions (cells), computing the gradient orientation in each cell, and creating a histogram of these orientations which helps to capture the structural aspects of the image [12].
- Linear Support Vector Machine: SVMs are supervised learning models used for classification and regression [13]. This machine learning model classifies data by recognizing HOG features corresponding to faces versus non-face[14.

Process:

- Compute HOG Features: HOG Features are calculated for the entire image [8].
- Sliding Window Detection: A window slides across the image, computing HOG features for each window and using the trained SVM to identify the presence of a face[8].
- Bounding Box Prediction: If the SVM classifier detects a face, it returns the coordinates of the bounding box surrounding the face [8].

b) Shape Prediction:: We utilized the 'dlib' library's shape predictor [8] with 'shape predictor 68 face landmarks' model for detecting facial landmarks[15]. It uses an ensemble of regression trees algorithm to localize 68 specific points on the face [16].

Algorithm:

- Ensemble of Regression Trees: This approach combines multiple regression trees trained to predict the positions of facial landmarks, Each tree is trained on a subset of the training data, contributing to the overall prediction accuracy [16].

Process:

- Bounding Box: The process begins with the bounding box of the detected face.
- Initial Landmark Estimation: Initial estimates of the landmark points are placed based on a generic face model [16].
- Feature Extraction: Features (e.g., pixel intensities) from the local neighborhood around each landmark are extracted [16].
- Regression Trees: The ensemble of regression trees iteratively refines the landmark positions by minimizing prediction errors at each stage [16].
- Final Landmarks: The precise locations of the 68 facial landmarks are output [16].

c) Face Descriptor Extraction:: For extracting face descriptors, the dlib face recognition model v1, which is based on a Residual Neural Network (ResNet) architecture, is utilized[8]. This model generates a 128-dimensional vector representing the face's features [8].

Algorithm:

- Residual Neural Network (ResNet): This deep learning model, characterized by residual connections, facilitates the training of deep networks by allowing gradients to flow directly through the network, thus migaθng the vanishing gradient problem [17].

Process:

- Input Image: The face image is aligned, cropped, and resized to a standard size (e.g., 150x150 pixels).
- Feature Extraction: The image is passed through multiple layers of convolution, pooling, and residual connections to extract high-level features [8].
- 128-Dimensional Vector: The final layer produces a 128dimensional vector (face descriptor) encoding the unique features of the face [8].
- Descriptor Comparison: These face descriptors can be compared using distance metrics (e.g., Euclidean distance) to determine the similarity between faces [8].

d) Classification::

- K-Nearest Neighbors (KNN): This algorithm classifies a face by finding the 'k' nearest neighbors in the feature space and assigning the label that is most common among them [18]. KNN inherently uses different distance metrics to measure the similarity between feature vectors [18]. Algorithm:
  1) Calculate Distance: Compute the distance between the test point and all training points [18].
  2) Find Neighbors: Identify the k-nearest neighbors[18].
  3) Voting: Assign the class by majority vote from the k- nearest neighbors [18].

We utilized different distance metrics described below to measure the similarity between feature vectors.

1) Euclidean Distance: It measures the straight-line distance between two points in the feature space [19], [20].
2) Manhattan Distance: It measures the absolute difference between the coordinates of a pair of objects [20].
3) Minkowski Distance: A generalization of Euclidean and Manhattan distances [21].

For the KNN method, we chose k=5, which means that when classifying a new face descriptor, the decision is based on the majority vote of the five nearest neighbors in the feature space.

- **Standalone Euclidean Distance Metric:** This metric measures the straight-line distance between two points in the feature space. For face recognition, the Euclidean distance between the feature vectors of the input face and the stored faces is computed, and the closest match is identified [19], [22]. In the Standalone Euclidean Distance method, we classify based on the closest single descriptor. We set a threshold value of 0.5 to decide a match: if the smallest distance between the input descriptor and any training set descriptor is less than or equal to 0.5, the input is identified as the corresponding person; otherwise, it's marked as 'Not identified.'

Data Management

The datasets were preprocessed to ensure that all images were uniformly resized and in proper format where necessary.For each subject in the dataset, data was split into 80-20 ratio and run through 10 iterations of experiments, to observe performance changes.

E. Analysis

- Accuracy Measurement: We evaluated the accuracy of the KNN and Euclidean distance methods using the sklearn.metrics.accuracy score function. This allowed us to compare the predicted labels with the actual labels in the test dataset.
- Computational Efficiency: We measured the time taken to classify the image for each method. This allowed us to compare the average time taken and standard deviationfor average time taken for each method to classify the image.
- Visualization: We plotted average accuracy and timemetrics to visualize the accuracy and computational efficiency of each method across different datasets.

F. Limitations

One of the primary limitations of our study was the availability of datasets in uncontrolled conditions. While we had access to robust datasets like Yale Faces and the FERET database, both of these were captured in controlled environments, limiting the variability of the data. The Celebrity Face Image dataset, though captured in uncontrolledconditions, was relatively small, which constrained our analysis. To address this limitation, we created customdatasets—Family Faces with Occlusions and Family Faces without Occlusions—designed to mimic more challenging,real-world scenarios. This approach allowed us to test our models under varied conditions, ensuring that our experimental design was both robust and replicable acrossdifferent environments.

G. Ethics

In conducting this research, we adhered to ethical guidelines for data usage and privacy. All datasets used were publicly available and properly cited. No personally identifiable information was used without consent, and the research was conducted with a commitment to advancing the field of facialrecognition responsibly.

III. Result

Our experiments yielded insightful results regarding the performance of KNN with different metrics and standalone Euclidean distance metric in facial recognition tasks. Below are the average outcome of each method from 10 iterations for each dataset spliited into 80-20 ratio, along with the outcomes of each iteration.

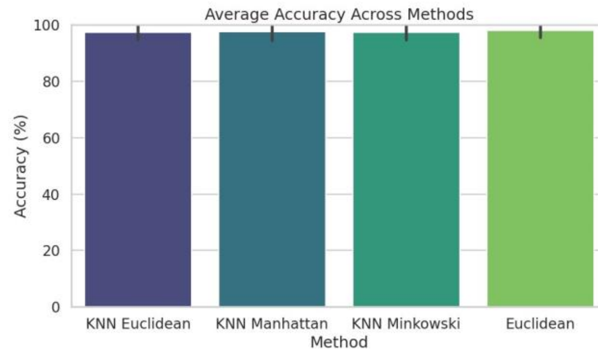

Fig. 1. Average Accuracy and Standard Deviation Across Datasets

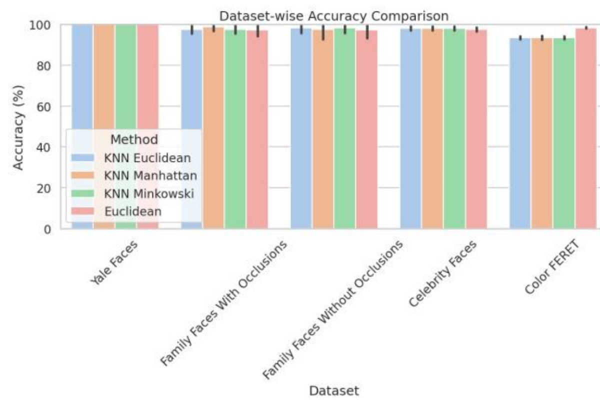

Fig. 2. Dataset Wise Accuracy Comparison

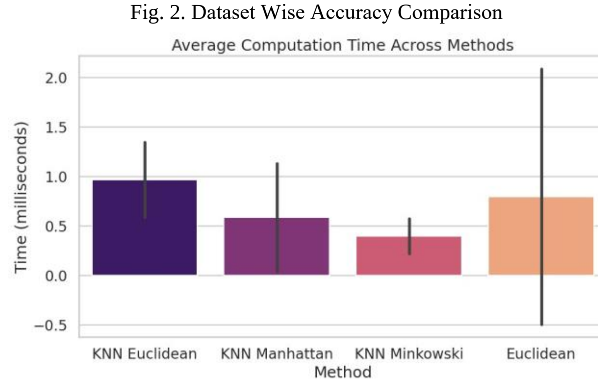

Fig. 3.Average Computation Time (m.s.) and Standard Deviation (m.s.) Across Datasets

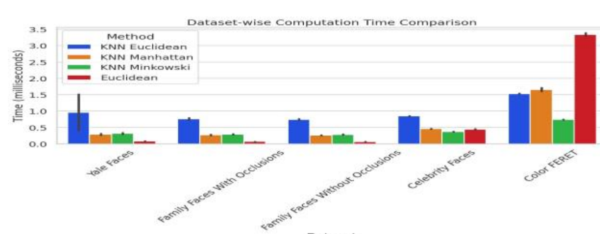

Fig. 4. Dataset Wise Computation Time (m.s.) Comparison

TABLE I: AVERAGE ACCURACY AND STANDARD DEVIATION ACROSS DATASETS

Dataset	Average Accuracy				Average Accuracy S.D			
	Euclidean	KNN			Euclidean	KNN		
		Euclidean	Manhattan	Minkowski		Euclidean	Manhattan	Minkowski
Celebrity Faces	97.567854	98.008114	98.035068	98.008114	0.960162	0.977041	0.975717	0.977041
Color FERET	98.393915	93.435538	93.525528	93.435538	0.41599	0.820855	1.02251	0.820855
Family Faces With Occlusions	97.346113	97.628348	98.812963	97.628348	3.081815	2.044616	1.914505	2.044616
Family Faces Without Occlusions	97.244872	98.429487	97.660256	98.429487	4.12826	2.721529	4.877344	2.721529
Yale Faces	100	100	100	100	0	0	0	0

TABLE II: AVERAGE COMPUTATION TIME (M.S.) AND STANDARD DEVIATION (M.S.) ACROSS DATASETS

Dataset	Average Computation Time				Average Computation Time S.D			
	Euclidean	KNN			Euclidean	KNN		
		Euclidean	Manhattan	Minkowski		Euclidean	Manhattan	Minkowski
Celebrity Faces	0.442354	0.845861	0.45855	0.367868	0.014102	0.008647	0.011574	0.011715
Color FERET	3.341836	1.531798	1.659035	0.742465	0.051019	0.015868	0.057481	0.01131
Family Faces With Occlusions	0.067927	0.757692	0.269077	0.289168	0.006236	0.03466	0.017138	0.014458
Family Faces Without Occlusions	0.065942	0.741838	0.260542	0.281619	0.004476	0.024872	0.011636	0.013014
Yale Faces	0.083805	0.96225	0.292239	0.316343	0.00648	0.554411	0.026807	0.029654

IV. DISCUSSION

A. Accuracy Analysis

The results from our experiments demonstrate that KNearest Neighbors (KNN) using different distance metrics(Euclidean, Manhattan, Minkowski) achieved strong accuracy across most datasets. In controlled environments like the Yale Faces and FERET datasets, where images are uniform and well lit, all methods, including the standalone Euclidean distance metric, consistently achieved perfect or near-perfect accuracy. This indicates that these methods are highly effective in controlled settings.

However, in more complex datasets such as Family Faces with occlusions, the KNN methods using the Manhattan and Minkowski metrics outperformed the standalone Euclidean distance in terms of accuracy. For instance, in the Family Faces dataset with occlusions, KNN Manhattan achieved the highest mean accuracy of 98.81%, whereas the standalone Euclidean method recorded 97.35 %. This suggests that while the Euclidean metric is robust, the Manhattan distance metric may offer better performance in scenarios with occlusions, likely due to its sensitivity to variations in image features.

In the Color FERET dataset, the standalone Euclidean distance achieved a higher accuracy (98.39%) compared to the KNN variants, which saw a significant drop in accuracy,particularly for KNN Euclidean and KNN Manhattan. This variability highlights the importance of dataset characteristics,such as diversity in expressions and poses, which can affect the performance of different methods.

The Celebrity Face Image dataset, which includes more varied images, presented a challenge for all methods, though they still maintained high accuracy. The KNN Minkowski and Manhattan metrics performed slightly better than the standalone Euclidean distance, with a mean accuracy of approximately 98%, indicaϴng that these methods may be better suited for datasets with greater variability.

B. Computational Efficiency

In terms of computational efficiency, the results clearly show that the standalone Euclidean distance metric is generally more efficient across all datasets. For example, in the Celebrity Faces dataset, the average computation time for the Euclidean method was approximately 0.442 milliseconds per image,while KNN Euclidean required almost double that time at 0.846 milliseconds. This pattern of efficiency held true across all datasets, highlighting the simplicity and computational advantage of the Euclidean distance metric.

However, among the KNN methods, the Manhattan and Minkowski distance metric consistently outperformed Euclidean in terms of speed. For instance, in the Family Faces dataset with occlusions, the average computation time for KNN Manhattan was 0.269 milliseconds, 0.289 milliseconds for KNN Minkowski, compared to 0.758 milliseconds for KNN Euclidean. This suggests that while Euclidean may generalize well across different datasets, the Manhattan and Minkowski metrics are more computationally efficient in scenarios requiring faster processing.

In larger and more complex datasets like Color FERET, the computational time for all methods increased, with the standalone Euclidean distance taking 3.342 milliseconds on average. This reflects the impact of dataset size and complexity on processing time, with more diverse facial features and poses contributing to longer computation times.

In real-world scenarios, particularly those involving occlusions, the Manhattan distance metric demonstrated a competitive edge in both accuracy and computational efficiency. The increased computational time observed in the Family Faces dataset with occlusions, especially for the KNN Minkowski and KNN Euclidean metrics, suggests that occlusions add complexity to the recognition process, making computational efficiency a critical consideration.

C. Implications for Facial Recognition Systems

The findings from this study have several important implications for the design and implementation of facial recognition systems. Firstly, the consistent performance of K -Nearest Neighbors (KNN) methods, particularly those using Manhattan and Minkowski metrics, across various datasets underscores their viability in scenarios where accuracy is crucial, such as in controlled environments. However, the slight decline in accuracy observed in more diverse datasets like the Celebrity Faces dataset suggests that systems dealing with a broader range of facial features, such as those found in public spaces or large-scale applications, might benefit from more sophisticated models or hybrid approaches that can better handle the variability in data [23]. Viola and Jones's work on real-time face detection further supports this, emphasizing the need for efficient algorithms in time-sensitive applications[24].

Moreover, the impact of occlusions on both accuracy and computational efficiency, particularly in the Family Faces dataset with occlusions, highlights the challenges that realworld facial recognition systems must overcome. Techniques to effectively manage occlusions, such as incorporating additional preprocessing steps or utilizing more advanced models, may be necessary to maintain high performance. This aspect has been extensively discussed by Zeng and Veldhuis in their paper [25].

While the Euclidean distance metric demonstrated superior computational efficiency across all datasets, the Manhattan distance metric offers a compelling alternative for scenarios where computational efficiency is crucial, delivering a balanced trade-off between accuracy and speed. In time-sensitive scenarios, the Euclidean method's simplicity may offer significant advantages, whereas KNN methods could be more appropriate where accuracy is paramount.

D. Limitations and Future Research

While the study provides valuable insights into the performance of KNN with different distance metrics in facial recognition, it is not without limitations. The relatively small size of some datasets, particularly the Yale Faces dataset, may limit the generalizability of the findings. Additionally, the Family Faces dataset was created to simulate more challenging conditions; access to larger and more varied datasets could provide a broader understanding of the methods' performance. This limitation is also discussed in "Improving Efficiency in Facial Recognition Tasks Through a Dataset Optimization Approach" [26], which underscores the importance of dataset size and diversity in assessing the effectiveness of facial recognition algorithms.

Future research should explore the performance of these methods on larger and more varied datasets, as well as the integration of additional techniques to handle occlusions and other real-world challenges. Additionally, investigating the impact of different feature extraction methods on the performance of these distance metrics could provide a more comprehensive understanding of their applicability in various facial recognition contexts.

REFERENCES

[1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognitton," ACM Compuiing Surveys, vol. 35, no. 4, pp. 399458, 2003, doi: 10.1145/954339.954342.

[2] J. Abiero, M. Kimwele, and G. Chemwa, "Face Recognitton Under Parttal Occlusion: A Deteciion and Exclusion of Occluded Face Regions Approach," in Pattern Recognition and Artificial Intelligence, MedPRAI 2021. Communications in Computer and Information Science, vol. 1543. Springer, Cham, 2022, doi: 10.1007/978-3-031-04112-9 2.

[3] B. Lahasan, S. L. Lulli, and R. SaSegundo, "A survey on techniques to handle face recognition challenges: occlusion, single sample per subject and expression," Artificial Intelligence Review, 2017, doi: 10.1007/s10462 -017-9578-y.

[4] C. Y. Wu and J. J. Ding, "Occluded face recognition using low-rank regression with generalized gradient direction," Pattern Recognition, vol. 80, pp. 256–268, 2018, doi: 10.1016/j.patcog.2018.03.016.

[5] M. D. Malkauthekar, "Analysis of euclidean distance and Manhattan Distance measure in face recognition," Third International Conference on Computational Intelligence and Information Technology (CIIT 2013), Mumbai, 2013, pp. 503-507, doi: 10.1049/cp.2013.26 36.

[6] A. C. Bhensle and R. Raja, "An Efficient Face Recognition using PCA and Euclidean Distance Classification," 2014.

[7] T. X. Tee and H. K. Khoo, "Facial Recognition using Enhanced Facial Features k-Nearest Neighbor (k-NN) for Attendance System," Proceedings of the 2020 2nd International Conference on Information Technology and Computer Communications, 2020, doi: 10.1145/ 3417473.3417475.

[8] D. E. King, ”Dlib C++ Library,” Retrieved from http://dlib.net, 2009.

[9] Phillips, P. J., Hyeonjoon Moon, Rauss, P., & Rizvi, S. A. (n.d.). The FERET evaluation methodology for face-recogni on algorithms. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.1997.609311.

[10] Yale Faces Database, 1997. Retrieved from: http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html.

[11] Celebrity Face Image Dataset, Kaggle. Retrieved from: https://www.kaggle.com/datasets/vishesh1412/celebrity face-imagedataset.

[12] N. Dalal and B. Triggs, ”Histograms of Oriented Gradients for Human Detection,” in Proc. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), 2005, doi: 10.1109/cvpr.2005.177.

[13] B. Scholkopf and A. J. Smola, Learning with kernels: Support vector¨ machines, regularization, optimization, and beyond. MIT Press, 2002.

[14] C. Cortes and V. Vapnik, ”Support-vector networks,” Machine Learning, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/bf00994018.

[15] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, ”300 faces In-the -wild challenge: Database and results,” Image and Vision Computtng (IMAVIS), Special Issue on Facial Landmark Localisatton ”In-The-Wild”, 2016.

[16] V. Kazemi and J. Sullivan, ”One millisecond face alignment with an ensemble of regression trees,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1867-1874.

[17] K. He, X. Zhang, S. Ren, and J. Sun, ”Deep Residual Learning for Image Recogniiion,” 2016 IEEE Conference on Computer Vision and Pattern Recognitton (CVPR), doi: 10.1109/cvpr.2016.90.

[18] T. Cover and P. Hart, ”Nearest neighbor pattern classification,” IEEE Transacttons on Informatton Theory, vol. 13, no. 1, pp. 21-27, 1967.

[19] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern classification. John Wiley & Sons, 2000.

[20] M. Faisal, E. M. Zamzami, and Sutarman, ”Comparattve Analysis of Inter Centroid K-Means Performance using Euclidean Distance, Canberra Distance and Manhattan Distance,” Journal of Physics: Conference Series, vol. 1566, 12112, 2020, doi: 10.1088/17426596/1 566/1/012112.

[21] H. Xu, W. Zeng, X. Zeng, and G. G. Yen, ”An Evolutionary Algorithm Based on Minkowski Distance for Many-ObjecCCve OpCCmizatton,” IEEE Transactions on Cybernetics, vol. 49, no. 11, pp. 3968-3979, Nov. 2019, doi: 10.1109/TCYB.2018.2856208.

[22] C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

[23] Z. Huang, R. Wang, S. Shan, and X. Chen, ”Face recognition on largescale video in the wild with hybrid Euclidean-and-Riemannian metric learning,” Pattern Recognition, vol. 48, no. 10, pp. 3113–3124, 2015, doi: 10.1016/j.patcog.2015.03.011.

[24] P. Viola and M. Jones, ”Rapid object detec33on using a boosted cascade of simple features,” in Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’01), 2001, vol. 1, pp. I-I.

[25] D. Zeng, R. Veldhuis, and L. Spreeuwers, ”A survey of face recognition techniques under occlusion,” IET Biometrics, vol. 10, no. 6, pp. 581-606, 2021, doi: 10.1049/bme2.12029.

[26] L. Vilac¸a, P. Viana, P. Carvalho, and M. Andrade, ”Improving Efficiency in Facial Recognition Tasks Through a Dataset Optimization Approach,” IEEE Access, 2024, doi: 10.1109/ACCESS.2024.3371020.

[27] O. Parkhi, A. Vedaldi, and A. Zisserman, ”Deep Face Recognitton,” in Proc. British Machine Vision Conference, 2015, vol. 1, pp. 41.1-41.12, doi: 10.5244/C.29.41.

[28] R. Brunelli and T. Poggio, ”Face recognition: Features versus templates,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 10, pp. 1042-1052, 1993.

[29] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, ”Eigenfaces vs. Fisherfaces: Recogni..on using class-specific linear projection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711-720, 1997.

[30] D. G. Lowe, ”Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[31] Y. Freund and R. E. Schapire, ”A short introduction to boosting,” Journal-Japanese Society For Artificial Intelligence, vol. 14, no. 771780, p. 1612, 1999.