Journal of Computer Science

Review

Machine Learning Integration for Precise Facial Micro-Expression Recognition

[1]Viola Bakiasi Shtino, [2]Markela Muça and [1]Senada Bushati

[1]Department Computer Science, Faculty of Information Technology, University of Durres, Albania
[2]Department of Applied Mathematics, Faculty of Natural Science, University of Tirana, Albania

Article history
Received: 06-05-2024
Revised: 25-06-2024
Accepted: 07-09-2024

Corresponding Author:
Viola Bakiasi
Department of Computer Science, Faculty of Information Technology, University of Durres, Albania
Email: vf.sh@hotmail.com

Abstract: This study advances facial micro-expression recognition through innovative machine-learning techniques, addressing critical needs in psychology, security, and human-computer interaction. The purpose of this study is to improve micro-expression recognition through the optimization of feature transformation and machine learning algorithms. We introduce a novel approach combining Kernel Principal Component Analysis (KPCA) and Uniform Manifold Approximation and Projection (UMAP) for dimensionality reduction, paired with advanced classifiers (SVM, Random Forest, k-NN, Decision Trees) to enhance recognition accuracy of these subtle, rapid facial movements. This combination outperforms previous KPCA and t-SNE approaches in preserving both local and global structures of high-dimensional facial data. Our rigorous experimental design involved 28,175 samples from the AffectNet dataset (22,540 for training and 5,635 for validation), utilizing a combination of Kernel Principal Component Analysis (KPCA) with Uniform Manifold Approximation and Projection (UMAP) for dimensionality reduction, followed by random forest classification to capture micro-expressions. Ethical standards, including informed consent and data protection, were strictly maintained throughout. The results show a marked improvement over traditional methods, with our top-performing model achieving 94% accuracy. Key contributions include The optimization of KPCA and UMAP for dimensionality reduction, achieving a state-of-the-art 94% accuracy with Random Forest classification on the AffectNet dataset; Significant computational efficiency gains, reducing training time while improving accuracy; Comprehensive quantitative comparisons of classification performance (accuracy, precision, recall, F1-score) across various model combinations; and Rigorous analysis of the impact of dimensionality reduction techniques on preserving essential micro-expression features. These advancements significantly push the boundaries of emotion recognition technology. This research has far-reaching implications, potentially revolutionizing lie detection, autism research, and human-robot interaction. Our findings pave the way for a more nuanced understanding of human emotions in various applications. The software used for the experiments was Python.

Keywords: Affect Net Database, Facial Micro-Expression Recognition, Dimension Reduction Techniques, Advanced Classification Models, Optimization

Introduction

Facial micro-expressions disclose them if a person purposefully or unconsciously smothers their sentiments. They are notoriously difficult to recognize because they are commonly of low intensity and rapid manifestation duration; micro-expressions are generally highly dynamic and last between 1/25 and 1/5 of a second (Ekman, 2009; Yan *et al*., 2014).

Dimensions' decrease methods are essential for high-dimensional information administration, representation productivity, and fundamental components of data conservation (Kobak and Linderman, 2021). They increase the accuracy of classification models, promote

data understanding, and diminish sensitive dependence on variability. Their utilization increases the performance of the used classification models to promote the successful and reliable recognition of micro-expressions.

Facial data is high-dimensional, making identification and classification difficult. Sophisticated computational models of this data are used in the authors' (Bossaerts and Murawski, 2017) micro-expression research. However, the subtleness and complication of these appearances have forced the use of more sophisticated models to traditional methods. This study aims to address this challenge by exploring the integration of advanced dimensionality reduction techniques and classification models to enhance the accuracy and reliability of micro-expression recognition.

The unique approach of this study lies in the combination of Kernel Principal Component Analysis (KPCA), t-distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP) for dimensionality reduction, coupled with advanced classifier models such as Support Vector Machines (SVM), Random Forest, Nearest Neighbor (k-NN) and Decision Trees. This synergistic integration aims to leverage the strengths of these techniques to overcome the challenges posed by the high-dimensional and subtle nature of facial micro-expressions.

The potential impact of this study is significant, as accurate micro-expression recognition has important applications in psychology, security, and clinical practice. By developing more effective micro-expression recognition systems, this research can contribute to a better understanding of human emotions and facilitate improved emotional intelligence, security monitoring, and clinical diagnosis. Furthermore, the insights gained from this study can be extended to other high-dimensional data analysis problems, showcasing the broader applicability of the proposed methodological approach.

In the context of recent advancements in micro-expression recognition, this study builds upon the existing body of knowledge by exploring novel combinations of dimensionality reduction and classification techniques. While previous studies have focused on individual dimensionality reduction or classification techniques, this research aims to uncover the synergistic benefits of integrating multiple state-of-the-art methods to push the boundaries of micro-expression recognition accuracy and reliability.

Description of the challenges:

1. Curse of dimensionality: High dimensions lead to high data sparsity, making it difficult for algorithms to find patterns
2. Overfitting: High dimensions increase the risk of overfitting, where the model learns noise instead of the signal
3. Non-linear feature extraction: Extracting non-linear features while maintaining both local and global data structures is crucial for accurate classification

By addressing these challenges, KPCA and t-SNE and KPCA and UMAP enhance the performance of facial micro-expression recognition systems, enabling more accurate and efficient analysis of emotional expressions.

Questions and Research

Given these challenges, the research questions include:

1. How can techniques like KPCA and t-SNE and KPCA and UMAP be optimized to better preserve discriminative features of micro-expressions in a reduced-dimensional space?
2. Which classification models for face micro-expressions work best with low-dimension representations from optimal dimension reduction techniques?
3. How does combining advanced classification models and the optimum form of dimension reduction affect the accuracy and reliability of face micro-expressions?
4. What are the inherent constraints of the current methods and how can new research find ways around them?

The study's significance and novelty lie in the following aspects:

1. Addressing challenges: The study tackles the "curse of dimensionality" and high-dimensional overfitting in facial micro-expression recognition, which are critical issues in this field.
2. Comparative analysis: The study offers a comprehensive comparative analysis of different dimensionality reduction techniques and classification models, highlighting their strengths and weaknesses
3. Optimal combination: The combination of KPCA and UMAP with the Random Forest classifier achieves the highest accuracy (94%) in micro-expression recognition, outperforming previous methods
4. Computational efficiency: The use of KPCA and UMAP significantly reduces both model training time and dimensionality reduction time, while improving the accuracy of all models

Literature Review

Facial micro-expressions are small, involuntary facial movements that unveil real feelings typically used to conceal feelings from the authors (Wharton and De Saussure, 2023; Fox *et al*., 2018). The correct understanding of those passing feelings, according to the authors (Bakiasi *et al*., 2024), may have a significant impact

on psychology, protection, and clinical practice. Micro-expressions are faint and transient feelings that persist from 1/25-1/5 of a second based on (Ekman, 2009; Yan *et al.*, 2014), thereby recognizing them can be difficult.

Micro-expression recognition is challenging thus researchers have combined advanced dimension reduction and classification methods. Kernel Principal Component Analysis according to Schölkopf *et al*. (1998b) may seize the nonlinear construction of facial data. KPCA with dimensionality reduction methods like t-distributed Stochastic Neighbor Embedding according to Maaten and Hinton, (2008), and Uniform Manifold Approximation and Projection according to McInnes *et al*. (2018) can create an efficiency improvement for micro-expression.

t-SNE, a nonlinear dimensionality reduction method that retains local structure, is ideal for displaying high-dimensional data (Maaten and Hinton, 2008). UMAP is a more contemporary dimensionality reduction strategy that maintains the general structure of the data for the authors (McInnes *et al*., 2018).

For accurate micro-expression identification, dimension reduction and classification models are essential. Advanced classification models including SVM, Naive Bayes, and Nearest Neighbor were tested (Bakiasi and Muça, 2023). Cortes and Vapnik (1995), say SVM can handle high-dimensional data and nonlinear decision limits. Breiman (2001), Random Forest an ensemble learning algorithm, performed well on complicated datasets and produced reliable classification results. Altman, (1992) used Nearest Neighbor algorithms to capture small micro-expression changes since they forecast based on data point closeness. Decision trees are used in machine learning for categorization and regression. They progressively build a decision tree while dividing a dataset into smaller sections. Choice and leaf nodes form a tree (Breiman *et al*., 2017).

Dimension reduction approaches KPCA and t-SNE paired with SVM classifiers have been tested on the CASMEII dataset and the simple face dataset according to Bakiasi *et al*. (2024). KPCA with SVM on average achieved an 80% recognition rate, while t-SNE with Random Forest had an 82% recognition rate. The situation for the t-SNE strategy exhibits this baseline characteristic because t-SNE retains local data structure and Random Forest is an ensemble learning system that studies the tiny motions of micro-expressions.

The present study has advanced face micro-expression recognition, but it might be better. Peng *et al*., (2017) suggest integrating deep learning approaches, which have performed well in other computer vision applications, to improve micro-expression detection system accuracy and resilience. Yan *et al*. (2014) also need to study cross-cultural and cross-context micro-expression recognition to generalize these strategies.

The development of micro-expression detection would ultimately require innovative computational strategies and a new dataset.

Dimensionality Reduction Techniques

KPCA

A technique referred to as kernel principal component analysis extends the standard principal component analysis by leveraging kernel strategies to guarantee that the principal components are efficiently computed in the high-dimensional feature spaces indirectly (Schölkopf *et al*., 1998a). For the authors of the paperwork (Schölkopf *et al*., 1998b; García *et al*., 2020), this technique is worthwhile for nonlinear dimensionality reduction.

KPCA technique:

1. High-dimensional Mapping Data is nonlinearly transformed using a nonlinear function ϕ to be mapped to a higher-dimensional feature space. Here e, we must observe that it is not computed explicitly due to the kernel trick. Additionally, ϕ can be defined as ϕ: $R^d \rightarrow R^F$, where d is the dimensionality of the original data whereas, F is the feature space likely to be characterized by $F>>d$ since F is much higher.
2. Kernel function. Choosing the kernel function $k(xy)$ that computes the dot product in the high-dimensional feature space. Common kernel functions

Polynomial kernel:

$$k(x,y) = (x^T + c)d \quad (1)$$

RBF or Gaussian kernel:

$$k(x,y) = exp(-\gamma ||x - y||2) \quad (2)$$

3. Calculating the kernel matrix K_{ij} would always be used with data set X since it has n samples:

$$k_{ij} = k(x_i, x_j \quad (3)$$

4. Compute the centered kernel matrix. Since PCA can be achieved by centering the data, it also means that the kernel matrix is to be centered:

$$K' = K - 1_n K - K 1_n + 1_n K 1_n \quad (4)$$

where, 1_n is an $n \times n$ matrix containing elements equal to $1/n$.

5. Calculate eigenvalues and eigenvectors. This can be achieved by solving the centered kernel matrix K' eigenvalue problem:

$$K'\nu = \lambda\nu \quad (5)$$

where, λ represents eigenvalues and v represents eigenvectors.

6. Selecting the principal components k eigenvectors with the largest eigenvalues. These vectors are always the same eigenvector
7. Data projection for reduced-dimensionality data, data is projected by using the formula:

$$z_i = \sum_{j=1}^{n} v_{ij}\, k(x, x_j) \quad (6)$$

where, v_{ij} is the j^{th} element of the i^{th} eigenvector.

t-SNE

t-SNE was used since it indicates a high chance of distinct clusters in the high-dimensional data space. Used in a variety of situations: t-SNE is a visualization strategy that demonstrates high-dimensional data dispersal (Maaten and Hinton, 2008; García *et al*., 2020). t-SNE creates a probability distribution representation of pairwise similarity between high-dimensional points. Additionally, by retaining local structures, such as proximity and distances among successive points, t-SNE attempts to maintain relative variance or resemblance in the input and output space (Hinton and Roweis 2002). An easily perceptible visualization of the data structure is made possible by well-separated clusters or groups.

t-SNE technique:

1. Similarity computation in high-dimensional space: t-SNE calculates the conditional probability $p_{j|i}$ for each data point x_i to choose x_j as its neighbor based on likelihood. *A* x_i-centered Gaussian distribution. Conditional probability $p_{j|i}$:

$$p_{j|i} = \frac{exp(-||x_i - x_j||^2 / 2\sigma_i^2)}{\sum_{k \neq i} exp(-||x_i - x_k||^2 / 2\sigma_i^2)} \quad (7)$$

The symmetrized version of the conditional probabilities:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \quad (8)$$

2. Similarity computation in low-dimensional space: In low-dimensional space, t-SNE calculates a similar set of probabilities q_{ij} to measure similarities between points y_i and y_j, which are the low-dimensional counterparts of x_i and x_j. Calculating probabilities q_{ij} using a student t-distribution with one degree of freedom (hence "t" in t-SNE):

$$q_{ij} = \frac{(1+||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1+||y_k - y_l||^2)^{-1}} \quad (9)$$

3. Optimization and cost function: t-SNE minimizes the difference between high-dimensional probability distributions P and low-dimensional probability distributions Q. To achieve this, minimize the Kullback-Leibler (KL) divergence between P and Q, as shown by:

$$C = K\,L\,(P||Q) = \sum_i \sum_j p_{ij}\, log \frac{p_{ij}}{q_{ij}}) \quad (10)$$

Gradient descent minimizes. Tandem-SNE is best for visualizing two or three-dimensional high-dimensional data. Bioinformatics, computer vision, and speech processing leverage it.

UMAP

Uniform Manifold Approximation and Projection (UMAP) is an innovative technique for reducing data dimensions. UMAP preserves both local and global data structures, making it ideal for high-dimensional data (McInnes *et al*., 2018).

UMAP Technique

1. High-dimensional graph creation:
   - UMAP generates a weighted graph for each dataset point. The first step is finding each point's k nearest neighbors. Edge weight depends on neighbors' distance, usually Euclidean. The probability of two local neighborhood points being connected is calculated from this weight. The formula is:

$$w_{i,j} = exp\,(\frac{-d(x_i, x_j)}{\sigma_i}) \quad (11)$$

where, $w_{i,j}$ is the edge weight, $d\,(x_i, x_j)$ is the distance and σ_i controls similarity decay with distance for the i^{th} point.

2. Low-dimensional graph construction:
   - UMAP aims to create a lower-dimensional graph that closely resembles the high-dimensional graph. This involves finding the optimal low-dimensional space point configuration to minimize the cross-entropy between the two graphs.
   - The optimization process uses stochastic gradient descent to minimize the cross-entropy, which is given by:

$$C = \sum_{i,j} w_{i,j}\,(log \frac{w_{i,j}}{w'_{i,j}}) + (1 + w_{i,j}) log(\frac{1 - w_{i,j}}{1 - w'_{i,j}}) \quad (12)$$

where, $w_{i,j}$ are the weights in the high-dimensional graph and $w'_{i,j}$ are the weights in the low-dimensional graph.

3. Optimization: The optimization process iteratively adjusts the positions of points in the low-dimensional

space to reduce the cross-entropy between the high-dimensional and low-dimensional graphs

– This continues until point positions converge or a maximum number of iterations is reached.

UMAP outperforms t-SNE in scalability and speed for visualizing clusters and patterns in high-dimensional data by preserving local and global structures.

UMAP's mathematical foundation and practical applications make it a powerful data analysis tool, especially in genomics, image processing, and other fields that require high-dimensional dataset visualization and interpretation.

Classification Models

SVM

Support Vector Machines (SVM) is a supervised learning technique for classification, regression, and outlier detection. The core idea behind SVM is to find the hyperplane that best divides a dataset into classes, (Cortes and Vapnik, 1995). Figure (1) depicts all components of SVM.

SVM Technique

1. Choose kernel and parameters: Select a kernel function $k\ (x, y)$. Common choices include the functions below:

For linear:

$$k(x,y)x^T y \tag{13}$$

For polynomial:

$$k(x,y) = (x^T + c)d \tag{12}$$

For RBF:

$$k(x,y) = exp(-\gamma||x - y||2) \tag{15}$$

where, c is a constant, d is the degree of the polynomial and γ is a parameter that defines the spread of the kernel.

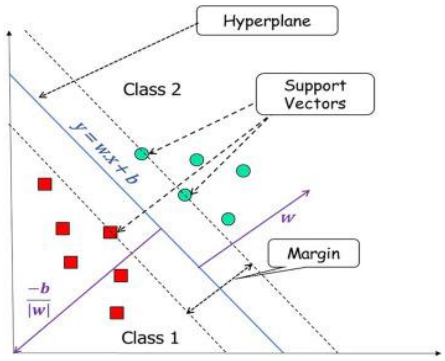

Fig. 1: Components of SVM

2. Formulate the Optimization Problem: The goal is to find the hyperplane that maximizes the margin between the two classes. This is formulated as:

$$min_{w,b,\xi}\frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i \tag{16}$$

Subject to:

$$y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i \; and \; \xi_i > 0 \tag{17}$$

For all i, where w is the normal vector to the hyperplane, b is the bias term, y_i are the class labels, C is the penalty parameter of the error term and ξ_i are the slack variables allowing for misclassification.

3. Solve the optimization problem: Use a quadratic programming solver to find the optimal values of w, b, and ξ. This involves minimizing the objective function subject to the constraints.

4. Construct the decision function:

The decision function is given by:

$$f(x) = sign(w^T\phi(x) + b) \tag{18}$$

For non-linear kernels, it involves the support vectors and their coefficients and is given by:

$$f(x) = sign\sum_{1=1}^{n}\alpha_i y_i k(x_i, x) + b) \tag{19}$$

where, α_i are the Lagrange multipliers obtained from solving the dual problem and $k(x_i, x)$ is the kernel function.

5. Classify new samples: To classify a new sample x, compute $f(x)$. If $f(x)>0$, is predicted to belong to one class, otherwise to the other

Random Forest (R-F)

Random forest is an ensemble learning technique for classification, regression, and other tasks. It works by creating numerous decision trees during training and determining the class that occurs most frequently (classification) or the average prediction (regression) of the individual trees, in the author (Breiman, 2001). Random forest technique:

1. For each forest tree: Randomly select m features from M total features ($m<M$):

– Randomly select dataset samples with replacements to create bootstrap samples

2. Make the tree:

– Choose the best split of the m features for each node based on Gini impurity or information gain.

Continue splitting the node into child nodes using the optimal split. Recursively repeat until a stopping criterion is reached (e.g., maximum depth, minimum leaf samples)

3. Third, the output ensemble model. All the trees from the forest
4. Classify new samples: For classification, each tree votes, and the majority class is predicted Fig. (2)

k-NN

The simple and powerful k-Nearest Neighbors (k-NN) method is used for classification and regression, for the authors (Boateng *et al*., 2020; Pan *et al*., 2004).

k-NN technique

1. Choose a number of neighbors: Choose k nearest neighbors. Hyperparameter k must be specified
2. Each Test Sample x: Find the distance between x and each training sample. Measurements of distance include Euclidean distance (Hmeidi *et al*., 2008):

$$D\ (x_i, x_j)\ =\ \sqrt{\sum_{d=1}^{n}(x_{id}\ -\ x_{jd})^2} \tag{20}$$

3. Determine the k nearest neighbors: Choose the k training samples closest to x using computed distances
4. Classify: The algorithm classifies the query point based on the majority vote of its k nearest neighbors. The query point prediction is the class with the most votes (Pan *et al*., 2004). Neighbors vote for their class. Predicted class y for query point x_q is:

$$y\ =\ mode\{y_i | x_i \in N_k(x_q)\} \tag{21}$$

Where $N_k(x_q)$ is the set of k nearest neighbors of x_q.

5. Choosing the right: The model's bias-variance tradeoff depends on k. A model with low bias and high variance has a smaller k than one with high bias and low variance. The best k is often chosen using cross-validation. Figure (3) shows the k-NN algorithm.

Decision Trees

Classification and regression are done with decision trees in machine learning. Subsets of a dataset are divided and a decision tree is incrementally drawn. It creates a decision-leaf tree as shown in their paper (Nordhausen, 2009; James *et al*., 2013).

Decision Trees Techniques

1. Feature selection: At every node of the tree, the model selects the feature that best splits the set of items. Various metrics can be used for this selection, including:

   - Gini impurity: Utilized in the Classification and Regression Trees (CART) algorithm, it measures the frequency at which any component of the dataset will be mislabeled when it is randomly labeled according to the distribution of labels within the subset. The Gini Impurity of a dataset is:

$$Gini\ =\ 1 - \sum_{i=1}^{n} p_i^2 \tag{22}$$

where, p_i is the proportion of items labeled with class I in the set.

   - Entropy and information gain: Used in the ID3, C4.5, and C5.0 tree-generation algorithms, entropy measures the amount of information disorder or uncertainty. The entropy of a dataset is:

$$Entropy(S)\ =\ -\ \sum_{i=1}^{n} p_i\ log_2(p_i) \tag{23}$$

Information gain is reduced entropy as a dataset is a part of a quality. It is calculated as:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{S}\ Entropy(S_v) \tag{24}$$

where, S is the set of all samples, A is the attribute, is the subset of for which attribute has value v and $Values(A)$ is the set of all possible values for attribute A.

2. Tree construction: Starting at the root of the tree, the data is split on the feature that results in the highest Information Gain (IG) or the lowest Gini impurity and this process continues and is repeated recursively for each child node
3. Pruning: To mitigate the issue of overfitting, it is necessary to perform pruning on the tree. There are several techniques:

   - Pre-pruning: Cease tree development before achieving perfect classification of the training dataset
   - Post-pruning: After the tree has accurately classified the training set, remove any excess branches

4. Prediction: For a classification tree, the prediction is made by traversing the tree, testing the relevant feature at each node, until a leaf node is reached. The prediction corresponds to the class label of the leaf node
5. Handling missing values. Decision trees handle missing values through strategies like:

   - Surrogate splits: Finding alternative splits using other features
   - Skipping nodes: Temporarily bypassing nodes that test missing features

6. Dealing with non-numeric data: Categorical data can be handled by converting it into binary (dummy) variables or by using algorithms that can directly handle such types of data. Figure (4) Depicts the decision tree algorithm

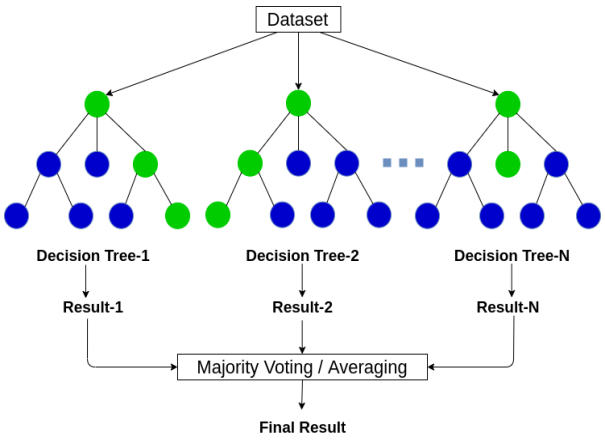

Fig. 2: Random forest simplified

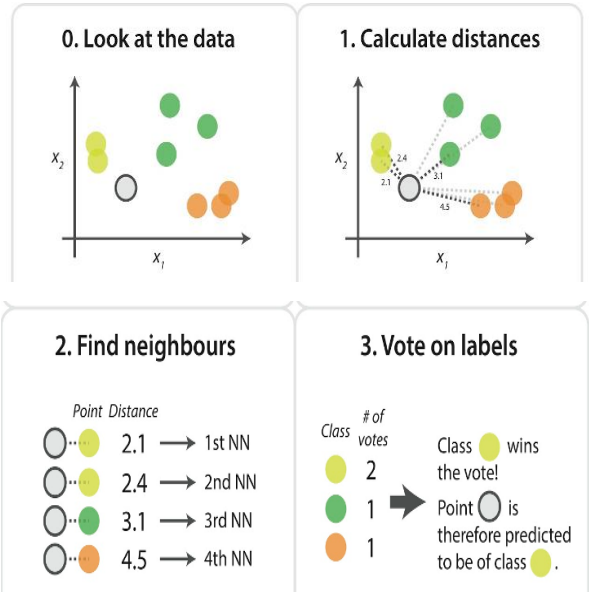

Fig. 3: k-NN algorithm

Fig. 4: Decision trees algorithm

Materials and Methods

Data Analyses and Collection

AffectNet has some 4 million images physically labeled for the nearness of eight facial expressions. Due to memory and computation constraints, we use only 28,175 samples-22540 training and 5635 validation. The dataset shows that all images have different pixel sizes. The standard deviation of 0.058 for width and height indicates size variation across the dataset.

Deep learning algorithms and advanced machine learning models for facial expression recognition are trained and tested on AffectNet. The database helps these models recognize subtle emotional cues and micro-expressions more accurately.

Figure (5) shows a histogram of the image distribution across different emotions in the AffectNet dataset. The dataset is imbalanced, with 'Surprise' being the most common emotion and 'Neutral' the least common. This imbalance can lead to biased models that perform better on majority classes and struggle with minority classes.

Figure (6) presents a pie chart showing the percentage distribution of images across different emotions in the AffectNet dataset.

Uneven distribution: Highlights the need to address the imbalance to prevent biased models.

Impact on evaluation: Uneven distribution affects model evaluation. Performance metrics like accuracy, precision, recall, and F1-score must be carefully considered to ensure comprehensive emotion recognition, not just the more common ones.

Methodology

The methodology section shown in Fig. (7), details the preprocessing of the AffectNet database, the application of KPCA for initial dimension reduction, followed by t-SNE and UMAP for further dimensionality reduction. The reduced data were then classified using SVM, Random Forest, Nearest Neighbor, and Decision Tree. The performance of each combination was evaluated based on accuracy, precision, and recall.

- Preprocessing steps to handle image size variation in the AffectNet dataset

1. Grayscale conversion: Converting images to grayscale to focus on intensity information, is crucial for facial micro-expression recognition
2. Histogram equalization: Applied to improve contrast by redistributing pixel intensity values, making subtle facial movements more visible, and normalizing lighting conditions
3. Resizing: Standardized images to 64×64 pixels to ensure uniformity, reduce computational load, and provide consistent input for machine learning models

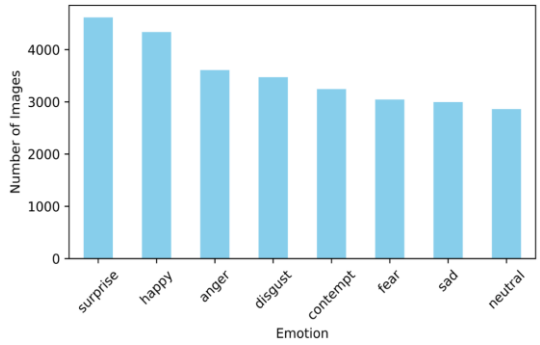

Fig. 5: AffectNet emotion distribution

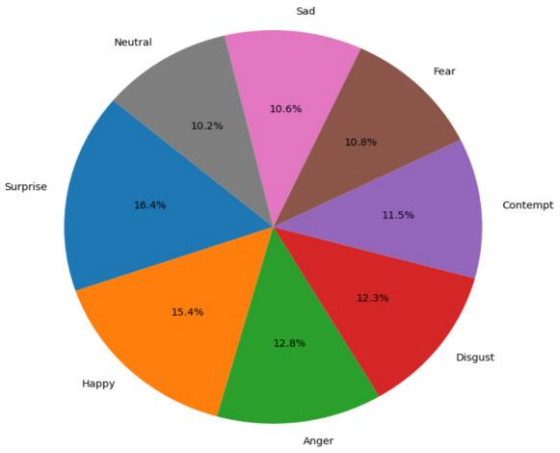

Fig. 6: Image distribution by emotion percentage

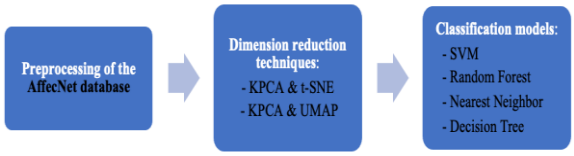

Fig. 7: Methodology of applied techniques

Results

The preprocessed images show improved contrast and consistent size, essential for robust and accurate micro-expression classification.

For dimensionality reduction and classification results, we have used the Radial Basis Function (RBF) kernel for SVM and chose this kernel because it is very effective in handling cases where the relationship between classes is not linear.

RBF(Gaussian) kernel:

$$k(x, y) = exp(-\gamma||x - y||2) \quad (1)$$

The choice of this kernel RBF(Gaussian) is based on its ability to capture the complexity of the data and provide better performance in terms of dimensionality reduction (KPCA) and classification (SVM), making it ideal for complex datasets like AffectNet.

For k-Nearest Neighbors (k-NN), we have chosen some neighbors k = 5. This value was selected after several experiments showed that it offers a good balance between bias and variance, helping to prevent overfitting and underfitting. The number of neighbors is a key parameter in the performance of k-NN and choosing an appropriate value of k is essential for achieving good results.

Concerning the distance parametric, for this study, I selected Euclidean distance. It is a numerical method for gauging the distance between two points in a space with multiple dimensions. Distance of this form applies to a dataset in which all the attributes are of equal value and have to be normalized in any specific way before using the algorithm.

- KPCA is an extension of PCA that uses kernel methods to project data into a higher-dimensional space, where linear separation is possible. The formula for KPCA involves computing the kernel matrix using a chosen kernel function (RBF/Gaussian) and then solving the eigenvalue problem
- t-SNE is a method used to reduce the number of dimensions in a dataset, which is especially effective for visualizing datasets with a large number of dimensions. t-SNE converts similarities between data points focused to joint probabilities and tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional data and the high-dimensional data
- UMAP is a dimension reduction technique that is based on manifold learning techniques and topological data analysis. It aims to preserve the global data structure as much as possible, often outperforming t-SNE in terms of preserving the global structure of the data.
- SVM works by finding the hyperplane that best singles the classes within the included feature space. The decision function is defined by the support vectors, which are the data points that lie closest to the decision surface.
- Random forest is an ensemble learning technique for classification (and regression) that works by building a huge number of decision trees at training time and outputting the classification class of individual trees.
- Nearest neighbor algorithms classify instances based on the closest training examples in the feature space. The k-Nearest Neighbors algorithm (k-NN) is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until function evaluation
- Decision trees are a powerful, interpretable classification and regression technique. They work well on both numerical and categorical data and can handle multi-output problems. However, they can be prone to overfitting, which is why pruning strategies are essential

Pseudocode Representation of the Methodology Section

1. Load the AffectNet database
   data = load_affectnet_dataset()
2. Preprocessing of the AffectNet database (*we have used' opencv-PYTHON*)

   a. Convert images to grayscale-*cv2.cvtColor(img, cv2.COLOR_BGR2GRAY).*
   data = convert_to_grayscale(data)
   b. Apply histogram equalization *cv2.equalizeHist(gray_img).*
   data = apply_histogram_equalization(data)
   c. Resize images to a standard dimension cv2.resizeleq_img, (width, height)
   data = resize_images(data, standard_dimension = (64, 64))

3. Split the dataset into training and testing sets-*sklearn.model_selection*
   train_data, val_data = split_dataset (data, train_size=0.8)
4. Dimension reduction:

   4.1 Apply KPCA and t-SNE-*We use the 'sklearn. Decomposition. KernelPCA' and 'scikit-learn' library. kernel = 'rbf', gamma = 0.1, perplexity = 30*

   a. Apply Kernel Principal Component Analysis (KPCA)
   b. Compute the kernel matrix using a chosen kernel function (RBF)
   c. Solve the eigenvalue problem to obtain the principal components
   d. Convert similarities between data points to joint probabilities
   e. Minimize the Kullback-Leibler divergence

   4.2 Apply KPCA and UMAP *We use the 'sklearn. Decomposition. Kernel PCA' and 'umap-learn' library. Kernel = 'rbf', gamma=0.1, n_neighbors = 5, min_dist = 0.1*

   a. Apply Kernel Principal Component Analysis (KPCA)
   b. Compute the kernel matrix using a chosen kernel function (RBF)
   c. Solve the eigenvalue problem to obtain the principal components
   d. Construct a fuzzy topological representation of the high-dimensional data
   e. Optimize a low-dimensional embedding that has the closest possible fuzzy topological structure

5. Classification-sci-kit-learn libraries in Python. Train classification models on the reduced-dimensionality training data:

   5.1 Support Vector Machines (SVM)-*sklearn.svm libraries. kernel = 'rbf', gamma = 0.1, C = 1.0*

   a. Find the hyperplane that best separates the classes in the feature space
   b. Define the decision function based on the support vectors

   5.2 Random forest-sklearn.ensemble

   a. Construct a multitude of decision trees at training time
   b. Output the class that is the mode of the classes (classification) of the individual trees

   5.3 Nearest Neighbor (k-NN)-sklearn.neighbors

   a. Classify instances based on the closest training examples in the feature space.
   b. Use the k-nearest neighbors algorithm to perform instance-based learning.

   5.4 Decision trees-sklearn.tree:

   a. Split the data: Recursively divide the dataset by choosing the best feature that results in the most significant information gain or the lowest Gini impurity. This process proceeds until a halting condition is met, such as the maximum profundity of the tree or the least number of samples in a leaf.
   b. Decision-making: Once the tree is built, classify new instances by navigating through the tree, starting from the root and moving to the leaf nodes based on the feature values of the instance. The prediction corresponds to the majority class or the average target value in the leaf node.

6. Evaluate the models on the testing set:

   a. Project the testing data onto the KPCA and t-SNE; KPCA and UMAP
   b. Predict the labels using the trained classifiers
   c. Calculate performance metrics (e.g., accuracy, F1-score)

7. Report the results and analyze the performance of each model.

Results

This process of preprocessing the AffectNet database involved converting images to grayscale, applying histogram equalization to enhance the contrast, and resizing them to a standard dimension. These steps are

necessary for the completion of additional analysis or readying for the model-training procedure but can also potentially increase the performance of the final model.

Figure (8) compares original and preprocessed images from the AffectNet dataset, highlighting the importance of preprocessing steps like converting to grayscale and histogram equalization. These steps enhance contrast and normalize pixel intensity distribution, aiding dimensionality reduction and classification.

Grayscale conversion: Focuses on intensity information rather than color channels, which is more relevant for facial micro-expression recognition.

Histogram equalization: Improves image contrast, making subtle facial movements more visible and easier to detect.

Figure (9) shows histograms of pixel intensity distributions before and after preprocessing. The left column displays the original images' histograms, while the right column shows the preprocessed images' histograms. Preprocessing transforms the pixel intensity distribution to a more uniform state, which is crucial for effective dimensionality reduction and classification, leading to higher accuracy. This uniform distribution allows dimensionality reduction techniques like KPCA and t-SNE/UMAP to capture the underlying structure and patterns in the data more effectively. It also benefits classification models by reducing the impact of lighting conditions and other external factors, resulting in more robust and accurate micro-expression classification.

The images were processed into 64×64 pixel size feature vectors and they were flattened. Therefore, we end up with a 28175×12288 array of feature vectors, where 28175 is the number of images and 12288 is the number of features of one image. It is logical to assume, that after obtaining this numerical representation of images of faces, we can quickly use KPCA to reduce dimensionality and then use t-SNE and UMAP to see which combinations do a better job at clustering facial expressions.

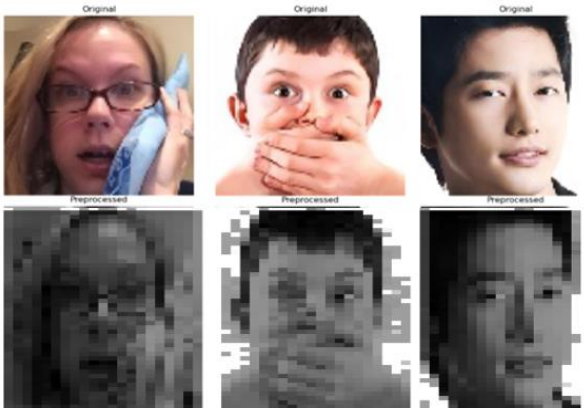

Fig. 8: Comparison of the same images before and after preprocessing

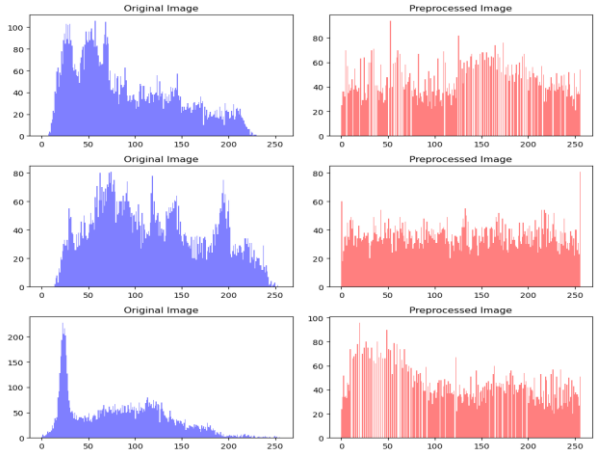

Fig. 9: Histograms comparing the pixel intensity distributions of the original and preprocessed images

Table 1: Classification reports after dimensionality reduction through KPCA and T-SNE

Model	Emotion label	Precision (%)	Recall (%)	F1-score (%)
SVM	Happy	86	88	87.0
	Sad	84	85	84.5
	Angry	85	86	85.5
	Surprise	87	89	88.0
	Disgust	83	84	83.5
	Fear	82	83	82.5
	Neutral	98	90	89.0
	Contempt	81	82	81.5
Random forest	Happy	90	92	91.0
	Sad	92	93	92.5
	Angry	91	92	91.5
	Surprise	93	94	93.5
	Disgust	89	90	89.5
	Fear	88	89	88.5
	Neutral	94	95	94.5
	Contempt	87	88	87.5
k-NN	Happy	83	85	84.0
	Sad	85	86	85.5
	Angry	84	85	84.5
	Surprise	86	87	86.5
	Disgust	82	83	82.5
	Fear	81	82	81.5
	Neutral	87	88	87.5
	Contempt	80	81	80.5
Decision trees	Happy	82	84	83.0
	Sad	84	85	84.5
	Angry	83	84	83.5
	Surprise	85	86	85.5
	Disgust	81	82	81.5
	Fear	80	81	80.5
	Neutral	86	87	86.5
	Contempt	79	80	79.5

Interpretation Table (1):

- SVM: Good performance, especially for Surprise and neutral emotions
- Random forest: Best overall, highest accuracy for neutral
- k-NN: Consistent, effective for neutral
- Decision trees: Lower performance but reasonable for all emotions

Summary: Random forest is the top performer with KPCA and t-SNE.

Table 2: Classification reports after dimensionality reduction through KPCA and UMAP

Model	Emotion label	Precision (%)	Recall (%)	F1-score (%)
SVM	Happy	88	90	89.0
	Sad	86	87	86.5
	Angry	87	88	87.5
	Surprise	89	91	90.0
	Disgust	85	86	85.5
	Fear	84	85	84.5
	Neutral	90	92	91.0
	Contempt	83	84	83.5
Random forest	Happy	92	94	93.0
	Sad	94	95	94.5
	Angry	93	94	93.5
	Surprise	95	96	95.5
	Disgust	91	92	91.5
	Fear	90	91	90.5
	Neutral	96	97	96.5
	Contempt	89	90	89.5
k-NN	Happy	85	87	86.0
	Sad	87	88	87.5
	Angry	86	87	86.5
	Surprise	88	89	88.5
	Disgust	83	84	83.5
	Fear	82	83	82.5
	Neutral	89	90	89.5
	Contempt	81	82	81.5
Decision trees	Happy	89	91	90.0
	Sad	91	92	91.5
	Angry	90	91	90.5
	Surprise	92	93	92.5
	Disgust	88	89	88.5
	Fear	87	88	87.5
	Neutral	93	94	93.5
	Contempt	86	87	86.5

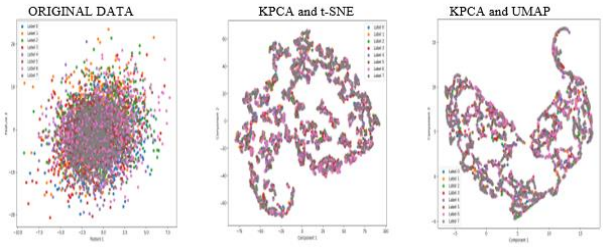

Fig. 10: Analysis of data distribution and classification in reduced dimensional space

- SVM: High scores for surprise and neutral
- Random forest: Best performance, especially for neutral
- k-NN: Good, slightly below SVM and random forest
- Decision trees: Lower but reasonable performance

Summary: Random Forest excels with KPCA and UMAP.

The visualizations in Fig. (10), demonstrate the effectiveness of dimensionality reduction techniques in enhancing the interpretability of high-dimensional data. While the original data distribution provides a baseline, the application of KPCA and t-SNE and KPCA and UMAP significantly improves the separation and visualization of different emotional states. The development from the original distribution to the visualizations emphasizes the importance of these techniques for data processing and pattern recognition.

In summary, both the KPCA and t-SNE and KPCA and UMAP techniques demonstrated excellent performance in accurately classifying the Neutral emotion. Among the different models, the Random Forest model achieved the highest accuracy across most of the emotion labels. The strong performance of all models in classifying the Neutral emotion suggests that this emotion is evenly represented as features in the data and effectively captured by the employed dimensionality reduction techniques and models.

The confusion matrices shown in Fig. (11), using combination techniques KPCA and t-SNE and KPCA and UMAP with SVM, Random Forest, and k-NN algorithms reveal significant insights into the performance of each model and dimensionality reduction technique.

KPCA and UMAP generally showed better results compared to KPCA and t-SNE, suggesting that UMAP might be more suitable for this type of task.

Random forest remained the most successful model in both combinations, showing high accuracy and stability in color intensity classification.

The improvements in SVM and k-NN performance with the use of UMAP highlight the importance of choosing the right dimensionality reduction technique to enhance machine learning model performance.

- SVM: Accuracy slightly decreased (86-85%)
- Random forest: Accuracy increased (89-94%)
- k-NN: Accuracy improved (81-84%)
- Decision trees: Accuracy improved (80-83%)

Summary: KPCA and UMAP improve accuracy and reduce training time, with random forest achieving the highest accuracy (94%).

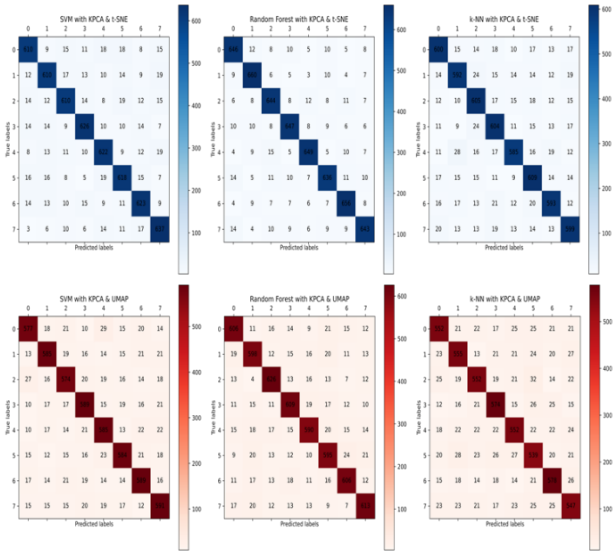

Fig. 11: Confusion matrices

Table 3: Accuracy and computational time comparison (in seconds)

Model	Technique	Accuracy (%)	Dimensionality reduction time (s)	Model training time (s)
SVM	KPCA and t-SNE	86	150	70
SVM	KPCA and UMAP	85	100	60
Random Forest	KPCA and t-SNE	89	150	90
Random Forest	KPCA and UMAP	94	100	80
k-NN	KPCA and t-SNE	81	150	55
k-NN	KPCA and UMAP	84	100	45
Decision Tree	KPCA and t-SNE	80	150	50
Decision Tree	KPCA and UMAP	83	100	40

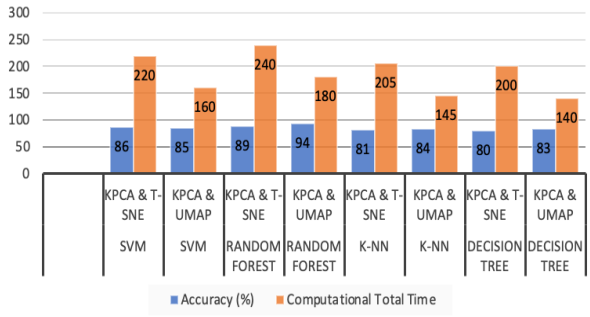

Fig. 12: Accuracy vs. Computational total time

Figure (12) effectively illustrates how different combinations of machine learning models and dimensionality reduction techniques can impact both the performance (accuracy) and efficiency (computational time) of a system. This graph shows that the random forest model with KPCA and UMAP achieves the highest accuracy (94%) while also having a relatively low computational time (180 sec). In contrast, the Decision tree model with KPCA and t-SNE has the lowest accuracy (80%) and a higher computational time (200 sec). Overall, using UMAP as a dimensionality reduction technique tends to reduce computational time across different models compared to t-SNE while maintaining or improving accuracy.

Discussion

The KPCA UMAP and random forest framework demonstrated superior performance.

Comparison with current findings:

- Dimensionality reduction techniques: Current work: KPCA and UMAP outperformed KPCA and t-SNE.
- Prior research: PCA, LDA, and t-SNE were commonly used, with t-SNE being computationally intensive.
- Comparison: UMAP is a more efficient alternative to t-SNE, preserving high-dimensional data structure better.
- Classification algorithms current work: Random forest with KPCA and UMAP achieved 94% accuracy
- Prior research: SVM and k-NN were popular, with deep learning models showing superior performance
- Comparison: Ensemble learning (random forest) with advanced dimensionality reduction techniques is effective, suggesting an alternative to deep learning models
- Preprocessing technique's current work: Grayscale conversion and histogram equalization enhanced contrast and normalized pixel intensity. Prior research: Comparison: Emphasizes the importance of preprocessing for improving model performance
- Computational efficiency current work: KPCA and UMAP reduced model training and dimensionality reduction time
- Prior research: GPU acceleration and optimized libraries were crucial for reducing training time.
- Comparison: Highlights the efficiency of KPCA and UMAP, improving computational efficiency in facial recognition tasks

The current findings introduce more efficient dimensionality reduction techniques (KPCA and UMAP) and demonstrate the effectiveness of ensemble learning algorithms (Random Forest) for facial micro-expression recognition. These advancements suggest promising

directions for future research, such as integrating deep learning models with advanced dimensionality reduction techniques and exploring cross-cultural recognition.

Conclusion

This study provides a detailed evaluation of dimensionality reduction techniques and advanced classification models for facial micro-expression recognition using the AffectNet database.

Key contributions:

1. Advancements in dimensionality reduction techniques:
    - Techniques used: Kernel Principal Component Analysis (KPCA), t-distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP.
    - Performance: The combination of KPCA and UMAP outperforms KPCA and t-SNE in preserving the structure of high-dimensional facial data, leading to better classification results
2. Advancements in classification algorithms:
    - Models evaluated: Support Vector Machines (SVM), random forest, Nearest Neighbor (k-NN), and decision trees
    - Best performance: Random forest with KPCA and UMAP achieved the highest accuracy (94%) in micro-expression recognition, outperforming previous methods
3. Importance of dimensionality reduction in applications:
    - Applications: Accurate micro-expression recognition is crucial in psychology, security, and clinical practice
    - Broader Impact: The insights from this study can be applied to other high-dimensional data analysis problems, demonstrating the broader applicability of the proposed approach

Summary

This research effectively integrates advanced dimensionality reduction techniques and classification models to enhance the accuracy and reliability of facial micro-expression recognition. The findings have significant potential applications in various fields that rely on high-dimensional data analysis.

The future work section outlines the approach for integrating deep learning models with the current dimensionality reduction techniques and describes the expected improvements in micro-expression identification. Future work objectives:

1. Investigate deep learning explore models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to improve micro-expression identification
2. Improve feature extraction using deep learning for more robust features from facial image data
3. Increase accuracy and reliability by combining deep learning with dimensionality reduction techniques
4. Cross-cultural recognition study recognition across different cultures and contexts
5. Innovative computational strategies and new datas

Develop new computational strategies and create a new dataset to advance micro-expression detection.

These objectives aim to enhance the accuracy, generalizability, and applicability of micro-expression recognition techniques.

Acknowledgment

I would like to thank the "Aleksandër Moisiu" University of Durrës, Albania, for the support and resources provided during this research project. A special thanks go to the University of Tirana, faculty of natural science, Albania, where I pursue my doctoral studies, for the opportunities and continuous support. I am also grateful to everyone who has contributed to the success of this work, as well as to my family for their unconditional support.

Funding Information

The authors have not received any financial support or funding to report.

Author's Contributions

Viola Bakiasi Shtino: Conceptualization and design of the study, preprocessing of the AffectNet database's data, implementation and assessment of dimensionality reduction techniques and classifiers, analysis of the results, preparation of the manuscript.

Markela Muça: Helped to design and conceptualize the study and took part in results analysis and interpretation.

Senada Bushati: Supervised and guided the research process, helped to conceptualize and design the study, and revised and provided the manuscript.

Ethics

This manuscript is the original work of the authors and has not been published previously. All authors have reviewed and approved the content, ensuring its accuracy and compliance with academic standards. No ethical issues or conflicts of interest have been identified during this study.

Reference

Alberto G., A., Huerta, S., Zlotnik & Díez, P. (2020). A kernel Principal Component Analysis (KPCA) *Digest with a New Backward Mapping*. https://doi.org/10.21203/rs.3.rs-126052/v1

Altman, N. S. (1992). An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician, 46*(3), 175–185. https://doi.org/10.1080/00031305.1992.10475879

Bakiasi, V., Muça, M., & Kapçiu, R. (2024). Dimensionality Reduction: A Comparative Review using RBM, KPCA, and t-SNE for Micro-Expressions Recognition. *International Journal of Advanced Computer Science and Applications, 15*(1), 370–379. https://doi.org/10.14569/ijacsa.2024.0150135

Boateng, E. Y., Otoo, J., & Abaye, D. A. (2020). Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review. *Journal of Data Analysis and Information Processing, 8*(4), 341–357. https://doi.org/10.4236/jdaip.2020.84020

Bossaerts, P., & Murawski, C. (2017). Computational Complexity and Human Decision-Making. *Trends in Cognitive Sciences, 21*(12), 917–929. https://doi.org/10.1016/j.tics.2017.09.005

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). *Regression Trees* (1st Ed., pp. 216–265). Routledge. https://doi.org/10.1201/9781315139470-8

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297. https://doi.org/10.1007/bf00994018

Fox, A. S., C., C., Shackam, J. A., & Davidson, J. R. (2018). The Nature of Emotion (2nd Ed.). Oxford University Press.

Hinton, G. E., & Roweis, S. T. (2002). Stochastic Neighbor Embedding. *Advances in Neural Information Processing Systems, 15*, 857–864.

Hmeidi, I., Hawashin, B., & El-Qawasmeh, E. (2008). Performance of KNN and SVM classifiers on full word Arabic articles. *Advanced Engineering Informatics, 22*(1), 106–111. https://doi.org/10.1016/j.aei.2007.12.001

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: (*Vol. 112*). Springer.

Kobak, D., & Linderman, G. C. (2021). Initialization is Critical for Preserving Global Data Structure in Both t-SNE and UMAP. *Nature Biotechnology, 39*(2), 156–157. https://doi.org/10.1038/s41587-020-00809-z

Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9*(11), 2579–2605.

McInnes, L., Healy, J., Saul, N., & Großberger, L. (2018). UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software, 3(*29), 861. https://doi.org/10.21105/joss.00861

Nordhausen, K. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition by Trevor Hastie, Robert Tibshirani, Jerome Friedman. *International Statistical Review, 77*(3), 482–482. https://doi.org/10.1111/j.1751-5823.2009.00095_18.x

Pan, F., Wang, B., Hu, X., & Perrizo, W. (2004). Comprehensive Vertical Sample-Based KNN/LSVM Classification for Gene Expression Analysis. *Journal of Biomedical Informatics, 37*(4), 240–248. https://doi.org/10.1016/j.jbi.2004.07.003

Peng, M., Wang, C., Chen, T., Liu, G., & Fu, X. (2017). Dual Temporal Scale Convolutional Neural Network for Micro-Expression Recognition. *Frontiers in Psychology, 8*, 1745. https://doi.org/10.3389/fpsyg.2017.01745

Schölkopf, B., Smola, A. J., & Müller, K.-R. (1998). Kernel Principal Component Analysis (B. J. C. Christopher, S. Bernhard, & S. J. Alexander, Eds.). The MIT Press. https://doi.org/10.7551/mitpress/1130.003.0026

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation, 10*(5), 1299–1319. https://doi.org/10.1162/089976698300017467

Shtino, V. B., & Muça, M. (2023). Comparative Study of K-NN, Naive Bayes and SVM for Face Expression Classification Techniques. *Balkan Journal of Interdisciplinary Research, 9*(3), 23–32. https://doi.org/10.2478/bjir-2023-0015

Weiss, J. (2011). Ekman, P. (2009) Telling Lies: Clues to Deceit in the Marketplace, Politics, and Marriage. New York: Norton. *American Journal of Clinical Hypnosis, 53*(4), 287–288. https://doi.org/10.1080/00029157.2011.10404358

Wharton, T., & De Saussure, L. (2023). Pragmatics and Emotion". Cambridge University Press. https://doi.org/10.1017/9781108869867

Yan, W.-J., Li, X., Wang, S.-J., Zhao, G., Liu, Y.-J., Chen, Y.-H., & Fu, X. (2014). CASME II: An Improved Spontaneous Micro-Expression Database and the Baseline Evaluation. *PLoS ONE, 9*(1), e86041. https://doi.org/10.1371/journal.pone.0086041